**UNDERGROUND TEST AREA SUBPROJECT
PHASE I DATA ANALYSIS TASK
VOLUME VII**

**TRITIUM TRANSPORT MODEL
DOCUMENTATION PACKAGE**

IT CORPORATION
4330 South Valley View Boulevard, Suite 114
Las Vegas, Nevada 89103

December 1996

DISCLAIMER

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Department of Energy, nor any of their employees, makes a warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Preface

This document represents the documentation for the Phase I Data Analysis Task performed in support of the current Regional Flow Model, Transport Model, and Risk Assessment for the Nevada Test Site Underground Test Area project. Because of the size and complexity of the model area, a considerable quantity of data were collected and analyzed in support of the modeling efforts. The data analysis task was broken into eight subtasks, and descriptions of the subtask activities are contained in the following eight volumes that comprise the Phase I Data Analysis Documentation.

- Volume I: Geologic Model Documentation Package
- Volume II: Potentiometric Data Documentation Package
- Volume III: Groundwater Recharge and Discharge Data Documentation Package
- Volume IV: Hydrologic Parameter Data Documentation Package
- Volume V: Transport Parameter and Source Term Data Documentation Package
- Volume VI: Groundwater Flow Model Documentation Package
- Volume VII: Tritium Transport Model Documentation Package
- Volume VIII: Risk Assessment Documentation Package

Each of the volumes details the results of one or two related subtasks. A summary review of project data and results will be contained in the Phase I Report (in progress).

The Underground Test Area Project Phase I Data Analysis Task was led by IT Corporation; Tetra Tech, Inc.; GeoTrans, Inc.; and Daniel B. Stephens Corporation. The task could not have been completed without the cooperative efforts of many other participants involved in work at the Nevada Test Site. Their organizations are listed in alphabetical order:

- Bechtel Nevada
- Desert Research Institute
- Lawrence Livermore National Laboratory
- Los Alamos National Laboratory
- Raytheon Services Nevada
- U.S. Geological Survey - Geologic Division
- U.S. Geological Survey - Water Resources Division
- U.S. Geological Survey - Yucca Mountain Project

Table of Contents

Preface

List of Figures

List of Tables

List of Acronyms and Abbreviations

1.0 Introduction

1.1 Subproject Description

1.1.1 Subproject Purpose and Scope

1.1.2 Subproject Objectives

1.1.3 Subproject Strategy

1.1.4 Subproject Organization

1.2 Phase I Data Analysis Task Description

1.2.1 Task Purpose and Scope

1.2.2 Task Objectives

1.2.3 Task Approach

1.2.4 Task Data Requirements and Organization

1.2.5 Task Documentation

1.3 Transport Model Subtask Description

1.3.1 Subtask Purpose and Scope

1.3.2 Subtask Objectives

1.3.3 Subtask Approach

1.4 Documentation Organization

2.0 Quality Control

2.1 Data Documentation Evaluation

2.2 Data Quality Evaluation

2.3 Checking Procedures

2.4 Software Quality Assurance

2.5 Standard Methodologies

Table of Contents (Continued)

- 2.6 Technical and Peer Reviews
- 2.7 Corroboration of Data through the Models
- 3.0 Transport Model Description
 - 3.1 Overview
 - 3.2 Solute Transport Governing Equations
 - 3.2.1 Mobile Domain
 - 3.2.2 Immobile Domain
 - 3.3 Implementation of MC_TRANS in a Mixed, Single/Dual Porosity Flow System
 - 3.4 Uncertainty Analysis
 - 3.5 Data Requirements for the Transport Model
 - 3.5.1 Model Geometry and Time Stepping
 - 3.5.2 Transport Parameters
- 4.0 Data Flow
 - 4.1 Groundwater Pathlines
 - 4.2 Selection of Particle Starting Location
 - 4.2.1 Underground Test Locations
 - 4.2.2 Test Yield and Cavity Radius
 - 4.2.3 Selection of Shots Below the Water Table
 - 4.3 MODPATH Simulations
 - 4.4 Calculation of Intermediate Pathline Locations
 - 4.5 Identification of Hydrostratigraphic Units Along a Pathline
 - 4.6 Calculation of Advective Velocity Along a Flow Path
 - 4.7 Selection of Starting Locations for Regional Modeling
 - 4.8 Evaluation of Upgradient and Downgradient Events Along the Same Pathline
- 5.0 Transport Parameters and Uncertainty
 - 5.1 Porosity
 - 5.2 Dispersion Coefficient
 - 5.3 Advective Velocity
 - 5.4 Diffusion Coefficient
 - 5.5 Fracture Spacing

Table of Contents (Continued)

5.6 Initial Tritium Activities

5.7 Correlation of Parameters

5.8 Other Constants

6.0 Transport Calculations

6.1 Pathlines

6.2 Tritium Transport Simulations - General

6.3 Tritium Transport Simulations - BOURBON (Yucca Flat)

6.4 Tritium Transport Simulations - HOUSTON (Central Pahute Mesa)

6.5 Tritium Transport Simulations - TYBO (Western Pahute Mesa)

7.0 Summary of Results

8.0 References

Appendix A - MC_TRANS Documentation

Appendix B - Nuclear Test Data Base

Appendix C - Transport Programs

Appendix D - Summary of Travel Times and Travel Distances for all Starting Locations

Appendix E - Input Data Files to MC_TRANS

List of Figures

<i>Number</i>	<i>Title</i>	<i>Page</i>
1-1	Area of Investigation Location Map	
1-2	Data Utilization in Phase I Modeling	
4-1	Transport Model Data Flow	
4-2	Diagram of a Detonation Below the Water Table	
4-3	Diagram of a Detonation Within Two Cavity Radii Above the Water Table	
4-4	Diagram of a Detonation Greater than Two Cavity Radii Above the Water Table	
4-5	Diagram of a Pathline Segment Through a Flow Model Cell	
4-6	Groundwater Pathlines from Selected Locations	
6-1	Geologic Model Cross Section Along the Pathline from BOURBON	
6-2	Geologic Model Cross Section Along the Pathline from HOUSTON	
6-3	Geologic Model Cross Section Along the Pathline from TYBO	
6-4	Contour Plots of Tritium Concentration Along the Pathline Beginning at KANKAKEE	
6-5	Histogram of Tritium Concentration for the 30-Year Period Around the Peak Concentration in the 95 Percent Level Plot	
6-6	Cumulative Density Plot of Maximum Concentration at 1 km Downgradient of BOURBON	

List of Figures (Continued)

Number	Title	Page
6-7	Cumulative Density Plot of Maximum Concentration at 10 km (6.2 mi) Downgradient of BOURBON	
6-8	Cumulative Density Plot of Maximum Concentration at 20 km (12.4 mi) Downgradient of BOURBON	
6-9	Contour Plots of Tritium Concentration Along the Pathline Beginning at BOURBON Conceptual Model Variation	
6-10	Contour Plots of Tritium Concentration Along the Pathline Beginning at HOUSTON	
6-11	Histogram of Tritium Concentration for the 30-Year Period Around the Maximum Concentration in the 95 Percent Level Contour Plot	
6-12	Cumulative Density Plot of Maximum Tritium Concentration at a Distance of 1 km Downgradient of HOUSTON	
6-13	Cumulative Density Plot of Maximum Tritium Concentration at a Distance of 5 km Downgradient of HOUSTON	
6-14	Cumulative Density Plot of the maximum Tritium Concentration at a Distance of 20 km Downgradient of HOUSTON	
6-15	Contour Plots of Tritium Concentration at the 95 Percent Level - Initial Tritium in Fractures Only	
6-16	Contour Plots of Tritium Concentration Along the Pathline Beginning at PEPATO	
6-17	Histogram of Tritium Concentration for the 30-Year Period Around the Maximum Concentration in the 95 Percent Contour Plot	

List of Figures (Continued)

Number	Title	Page
6-18	Cumulative Density Plot of Maximum Tritium Concentration at 1 km Downgradient of TYBO	
6-19	Cumulative Density Plot of Maximum Tritium Concentration at a Distance of 10 km Downgradient of TYBO	
6-20	Cumulative Density Plot of Maximum Tritium Concentration at a Distance of 30 km Downgradient of TYBO	
6-21	Contour Plots of Tritium Concentration at the 95 Percent Level for Simulations from TYBO Alone	
6-22	Contour Plots of Tritium Concentration Along the TYBO Pathline for the Case of Smaller Specific Discharge	

List of Tables

<i>Number</i>	<i>Title</i>	<i>Page</i>
3-1	Hydrostratigraphic Units Used in the Modeling	
4-1	Effective Porosity Values for Each Hydrostratigraphic Unit	
4-2	Relationship of Hydraulic Conductivity to Fracture Porosity	
4-3	Events Upgradient and Downgradient of BOURBON, HOUSTON, and TYBO	
5-1	Porosity Data and Uncertainty for Each HSU	
5-2	Flow Model Parameters Incorporated in the Transport Model via Sensitivity	
5-3	Fracture Spacing for the Volcanic Aquifers	
5-4	Initial Tritium Concentration Values and Ranges	
5-5	Correlation of Input Parameters	
6-1	Relative Frequency of Tritium Concentration versus Distance for the KANKAKEE/BOURBON Pathline	
6-2	Relative Frequency of Tritium Concentration versus Distance for the HOUSTON Pathline	
6-3	Relative Frequency of Tritium Concentration versus Distance for the PEPATO/TYBO Pathline	
6-4	Input Parameters for the AT123D Simulations	

List of Tables (Continued)

Number	Title	Page
6-5	Additional Dilution Due to Three-Dimensional Dispersion	
7-1	Approximate Distance Beyond Which Tritium Concentration Does Not Exceed 20,000 pCi/L at the 95-Percent Level	

List of Acronyms and Abbreviations

AA	Alluvial aquifer
ASTM	American Society of Testing Materials
BAQ	Basal Aquifer
BC	Boundary conditions
BCF	Block-centered flow
BCU	Basal Confining Unit
BRT	Belted Range Thrust
CAU	Corrective Action Unit(s)
CAS	Corrective Action Site(s)
cm	Centimeter(s)
DDE_F	Data Documentation Evaluation Flags
DIF	Data Information Form
DOE	U.S. Department of Energy
DOE/NV	U.S. Department of Energy, Nevada Operations Office
DRI	Desert Research Institute
EPA	U.S. Environmental Protection Agency
ER	Environmental Restoration
ERDBMS	Environmental Restoration Data Base Management System
ERMA	Environmental Resource Management Applications
ERP	Environmental Restoration Program
ET	Evapotranspiration
FFACO	Federal Facility Agreement and Consent Order
ft	Foot (feet)
GHB	General Head Boundary
g/cm ³	Gram(s) per cubic centimeter
HSU	Hydrostratigraphic unit(s)
I	Intrusives
IT	IT Corporation
kh	Horizontal Hydraulic Conductivity
km	Kilometer(s)
LANL	Los Alamos National Laboratory
LCA	Lower Carbonate Aquifer

List of Acronyms and Abbreviations (Continued)

LCCU	Lower Clastic Confining Unit
LHS	Latin hypercube sampling
LLNL	Lawrence Livermore National Laboratory
m	Meter(s)
m/d	Meter(s) per day
m ² /yr	Square meter(s) per year
NTS	Nevada Test Site
NWIS	National Water Information System
pCi/L	PicoCurie(s) per liter
Pi	Input parameters
QA	Quality Assurance
RSN	Raytheon Services Nevada
SOP	Standard Operating Procedure
SQP	Standard Quality Practice
TBAQ	Welded tuffs above BCU
TC	Tuff Cone
TCBCU	Tuff Cone Basal Confining Unit (nonwelded tuffs)
TMA	Timber Mountain Aquifer
TSDVS	Tertiary Sediments/Death Valley Sediments
UCCU	Upper Clastic Confining Unit
UGTA	Underground Test Area
USGS	U.S. Geological Survey
UTM	Universal Transverse Mercator
VA	Volcanic Aquifer
VCA	Volcanic Confining Unit(s)
VU	Volcanics Undifferentiated

1.0 Introduction

The Nevada Test Site (NTS), located in Nye County in southern Nevada, was the location of 908 nuclear detonations conducted in shafts and tunnels between 1951 and 1992. About one-third of these nuclear underground detonations were conducted near or below the water table (Duncan, 1993). As an unavoidable consequence of these testing activities, radionuclides have been introduced into the subsurface environment, impacting groundwater. The U.S. Department of Energy, Nevada Operations Office (DOE/NV) has been in the process of investigating the effects of these underground nuclear detonations as part of the Underground Test Area (UGTA) Subproject. Descriptions of the project, the data analysis task, and the Risk Assessment Subtask are briefly discussed.

1.1 Subproject Description

This subsection includes descriptions of the purpose, scope, objectives, strategy, and organization of the UGTA Subproject. The documentation that supports this effort is outlined in Section 1.2.5 of this report.

1.1.1 Subproject Purpose and Scope

The purpose of the subproject is to collect data and perform technical evaluations to support decisions about the corrective actions necessary to ensure that the weapons testing areas do not pose a significant risk to human health and the environment.

The scope of the UGTA Subproject includes the following activities:

- New and existing data collection and evaluation
- Quantification of risk to human health and the environment
- Selection of the corrective actions to reduce risk from the sites
- Implementation of corrective actions and closure of the testing areas

1.1.2 Subproject Objectives

The UGTA Subproject objectives are prerequisites to generating the necessary end products to complete the subproject. The objectives provide the framework for subproject strategy. The subproject definition establishes the starting point, and the objectives focus on the end products.

The overall objective of the UGTA Subproject is to evaluate the effects of the weapons testing areas on groundwater and to identify and implement the appropriate corrective actions. This

objective ensures that risks to human health and the environment from impacted groundwater are within protective levels.

Meeting the overall objective requires that several specific project objectives be achieved:

- Effectively plan and scope the UGTA Subproject so that the necessary data are properly collected and relevant evaluations are properly performed.
- Prepare UGTA Subproject plans as guidance and information tools for efficiently conducting the subproject.
- Develop credible predictions of contaminant fate and transport.
- Quantify the risk to human health and the environment, in the absence of any present or future remedial action, to an acceptable level of uncertainty.
- Evaluate an appropriate range of remedial technologies applicable to the various contaminants and impacted media.
- Select appropriate corrective action alternatives.
- Implement selected corrective actions.
- Close the underground test areas.

1.1.3 Subproject Strategy

The Nevada Division of Environmental Protection regulates U.S. Department of Energy's (DOE) Nevada corrective actions through the *Federal Facility Agreement and Consent Order* (FFACO, 1996). Appendix VI of this agreement, "The Corrective Action Strategy," describes the processes that will be used to complete the corrective actions, including those in the UGTA Subproject. The individual sites covered by the agreement are known as Corrective Action Sites (CASs), and they are grouped into Corrective Action Units (CAUs). The UGTA Subproject comprises six CAUs, generally reflecting the geographic locations and geologic and hydrologic environments of the weapons testing areas.

1.1.4 Subproject Organization

Because of the complexity of the problem at hand, the UGTA scope was separated into two major phases. Phase I project activities have focused on a regional investigation. During Phase II, several localized investigations focusing on the CAUs will be conducted.

Phase I has consisted of existing data assessment, groundwater flow and solute transport modeling, and a preliminary risk assessment which has been a key portion of the data assessment activities. The risk assessment has been designed to determine if there is an imminent risk to human health and the environment and to provide input to a value-of-information analysis that will identify and prioritize potential data needs. The results of Phase I will be directly used in developing the work scope for the weapons testing areas and in implementing Phase II. Further information on the Phase I data analysis task is provided in the following subsection.

1.2 Phase I Data Analysis Task Description

This subsection includes a more detailed description of the purpose and scope, objectives, approach, and data requirements and organization of the UGTA Data Analysis Task.

1.2.1 Task Purpose and Scope

The primary purpose of the Data Analysis Task is to compile and analyze existing, as well as new, data. During Phase I, the focus of this analysis task has been a regional investigation. The Phase I Data Analysis Task scope of work includes the evaluation of regional groundwater flow conditions, the transport of tritium from the weapons testing areas, and the determination of the presence or absence of imminent risk to human health and the environment.

1.2.2 Task Objectives

The following were the objectives of the Phase I Data Analysis Task:

- Evaluate the regional groundwater flow conditions using a groundwater flow model.
- Evaluate the transport of tritium in groundwater using a groundwater transport model.
- Develop a preliminary assessment of risk to human health and the environment from the weapons testing areas.

1.2.3 Task Approach

The approach for the Phase I Data Analysis Task has included the following steps:

- Selection of a regional groundwater flow system large enough to encompass the NTS
- Selection of an area of investigation large enough to include the selected regional groundwater flow system
- Collection and analysis of data

- Simulation of regional groundwater flow with a three-dimensional numerical code
- Definition of the major flow paths from the weapons testing areas using a particle-tracking code
- Simulation of the transport of tritium using a numerical, one-dimensional transport code
- Calculations of risk to humans and the environment from exposure to tritiated groundwater

The NTS regional groundwater flow system ([Figure 1-1](#)), which fully encompasses the groundwater flow system underlying the NTS, was selected for this investigation; however, the boundary of this groundwater flow system has not been well defined in some areas. The flow system boundary defined by Waddell et al. (1984) was evaluated at the start of this investigation and is shown on [Figure 1-1](#). The selected area of investigation had to be large enough to allow for a reevaluation and potential expansion of the groundwater flow system boundary. This area covers a large part of Southern Nevada and part of Inyo County in eastern California. The area extends from Death Valley, north to Antelope Valley, and from the Palmetto Mountains east to the Sheep Range, extending over an area of 80,650 square kilometers (31,140 square miles).

Data collection and analysis activities are dictated by the type of data needed. Data types and their methods of analysis are, in turn, dictated by the modeling approach and will not be discussed in this section; however, the documentation of necessary data types and the methods of analysis are discussed in the preceding volumes of this documentation.

The computer code selected to simulate groundwater flow is MODFLOW (McDonald and Harbaugh, 1988). This code was developed by the U.S. Geological Survey (USGS) for the numerical simulation of fully three-dimensional saturated groundwater flow in porous media; and it was also designed to simulate flow under both steady-state and transient conditions. The code will also be used to simulate flow in fractured media by invoking an equivalent porous media assumption. This assumption implies that, at the scale of the model, the hydraulic behavior of fractured geologic units is analogous to that of porous media.

The code selected to simulate pathlines is MODPATH (Pollock, 1989). This code was developed by the USGS to compute and display three-dimensional pathlines based on results from steady-state simulations from MODFLOW (McDonald and Harbaugh, 1988).

The program uses information on layer geometry, boundary conditions, and flux rates to calculate the velocities and positions of particles at different times. MODPATH was used to compute and display pathlines originating from individual underground nuclear testing locations located throughout the weapons testing areas. MODPATH also provides Darcy velocity distributions along the pathlines, which are required by the transport code.

The code used to simulate contaminant transport is MC-TRANS (IT, 1995), and it was developed specifically for this subproject. This finite-element, one-dimensional, radionuclide transport model is capable of simulating advection in a dual-porosity, fractured system with dispersion, sorption, and first-order decay. The code was used to simulate the concentrations of tritium downgradient from selected nuclear test sites under the “no-action alternative.” The code was used in the stochastic mode to evaluate the uncertainties associated with the predicted tritium concentrations.

The code used to estimate human health risk from exposure to tritium-contaminated groundwater is GW-RISK (IT, 1996g), also developed specifically for this subproject. The code calculates impacts from radiological exposures including intake, dose, and risk; it uses factors derived from U.S. Department of Energy (DOE, 1993), International Commission on Radiation Protection (ICRP, 1991), and U.S. Environmental Protection Agency (EPA) guidance (EPA, 1989a, EPA 1989b; and EPA, 1991). The code uses tritium concentrations simulated by MC-TRANS (IT, 1995) as input for three of the most conservative predicted pathlines. The ecological risk assessment determines tritium concentrations below which no adverse effects to ecological receptors should be expected. These concentrations were compared with measured and modeled tritium concentrations in surface water and groundwater. Although no formal guidance exists on how to assess ecological risks associated with exposure to radionuclides, the general format suggested in the EPA *Framework* document (EPA, 1992) was used in conjunction with published radiation dose models (Baker and Soldat, 1992; Blaylock et al., 1993). Both human and ecological risk assessments included examination of data, exposure assessment, toxicity assessment, risk evaluation, and evaluation of uncertainties.

1.2.4 Task Data Requirements and Organization

Task data requirements and organization are briefly described in this section. Data required to achieve the objectives of the Phase I data analysis are as follows:

- Geologic data for regional groundwater flow and transport modeling
- Hydrologic data for regional groundwater flow modeling

- Transport parameter and tritium source data for Tritium transport modeling
- Risk-related data for assessment of risk to human health and the environment

MODFLOW (McDonald and Harbaugh, 1988) data requirements include hydrogeologic framework data, potentiometric data, hydraulic conductivity/transmissivity data, and groundwater recharge and discharge data. Data needs also include estimates of the uncertainties that are used to define the input variable bounds and the output variable target ranges for use in the model calibration process. MODPATH data requirements are MODFLOW data and porosity data. In addition to the Darcy velocities calculated by MODPATH, the transport model (MC-TRANS) requires transport parameter data including bulk and fracture porosity, dispersion, and tritium matrix diffusion data. Data for the tritium source needed for the transport model include the tritium source initial spatial extent and concentration. Data needed for the risk assessment include the peak tritium concentration generated by the transport model from the flowpath at each receptor location. In addition, exposure scenario, tritium dosimetry, and tritium detriment and risk data are required.

The data analysis task was organized into several subtasks based on the approach and the major types of data needs. Data needs of the flow model were fulfilled under four subtasks: Geologic Model, Potentiometric Data, Groundwater Recharge and Discharge Data, and Hydrologic Parameter Data. Porosity data are needed by both MODPATH and MC-TRANS and were collected, together with data on dispersion and matrix diffusion, under the Transport Parameter Subtask. Data needed to define the source term were collected and analyzed under the Source Term Subtask. Data needed for risk assessment were collected and analyzed under the Risk Assessment Subtask.

1.2.5 Task Documentation

Work performed under the data analysis task is documented in the following eight volumes:

- Volume I - *Regional Geologic Model Documentation Package* (IT, 1996a)
- Volume II - *Potentiometric Data Documentation Package* (IT, 1996b)
- Volume III - *Groundwater Recharge and Discharge Data Documentation Package* (IT, 1996c)
- Volume IV - *Hydrologic Parameter Data Documentation Package* (IT, 1996d)

- Volume V - *Transport Parameter and Source Term Data Documentation Package* (IT, 1996e)
- Volume VI - *Groundwater Flow Model Documentation Package* (IT, 1996f)
- Volume VII - *Tritium Transport Model Documentation Package* (this package)
- Volume VIII - *Risk Assessment Documentation Package* (IT, 1996g)

A separate documentation package was not prepared for the Source Term Subtask; however, results of this subtask are reported in Volume V of the task documentation.

The documentation packages are meant to fulfill several subproject needs. The first five packages serve as a repository of available and pertinent data collected for possible inclusion into the modeling process. There is little relationship between these five packages; instead, they are directly related to Volumes VI (Groundwater Flow) and VII (Tritium Transport), the packages which document the flow and transport modeling process and results. Volume VIII, Risk Assessment, is most directly related to Volume VII (Tritium Transport). Concentrations of tritium at specific locations and times required by the risk assessment are provided by the transport model. [Figure 1-2](#) illustrates how information from each of the packages was used and how the packages related to each other.

A second subproject need was to centralize the required information in order to have it readily available for a highly technical review of the modeling process. The intended audience of these eight documents is the technical reviewer. A summary report of the entire project is contained in the *Regional Groundwater Flow and Tritium Transport Modeling and Risk Assessment of the Underground Test Area, Nevada Test Site, Nevada* (DOE, in press). It contains information related to the regulatory and risk management issues. This document is intended to serve as the primary subproject summary that reaches a broader audience.

1.3 Transport Model Subtask Description

The transport model is used to calculate the concentration of tritium in groundwater at selected locations downgradient from major weapons testing areas of the NTS. The model was created to serve as a tool to help guide future decisions with regard to the NTS Environmental Restoration Program.

Figure 1-2
Data Utilization in Phase I Modeling

1.3.1 Subtask Purpose and Scope

The purpose of the transport model is to predict the regional-scale migration of tritium at the scale of kilometers in the groundwater system, away from selected underground test locations. The simulations are limited to tritium because it was created in the greatest abundance during underground nuclear detonations and is mobile in a groundwater environment. CAU scale simulations to be performed later will likely include additional radionuclides.

1.3.2 Subtask Objectives

The transport modeling task has three objectives:

- Calculate the tritium concentration in groundwater downgradient from underground test locations. These concentrations are used in the environmental and human health risk assessment calculations to assess the potential risks over the next 20 to 100 years.
- Serve as a tool that can be used to estimate the extent of groundwater contamination resulting from the underground nuclear testing.
- Assess the impacts of flow and transport parameter uncertainty on the predicted downgradient tritium concentration. Assess the impact of uncertainty in different input parameters on the predicted tritium concentration.

1.3.3 Subtask Approach

The approach used to calculate tritium concentration is presented in this section, followed by a description of the data needs. A general approach, defining the types of analyses, the steps involved, and the parameters needed, is presented first. The transport modeling approach provides the details of the modeling and its descriptions should facilitate an understanding of the approximations inherent in the final results.

A three-step modeling approach was used in this study. First, a three-dimensional groundwater flow model of the groundwater system was developed from which groundwater flow pathlines were identified. Second, advective transport along the pathlines was simulated using a particle tracking technique. This provided an assessment of the relative amount of downgradient migration from each underground test location. Third, tritium transport along three selected pathlines was simulated using a one-dimensional, finite-element model. This provided the tritium concentration data that was used for the risk assessment.

The USGS three-dimensional, finite-difference groundwater flow model, MODFLOW (McDonald and Harbaugh, 1988), was calibrated to groundwater potentiometric and discharge data from the NTS region. Details of the groundwater model and calibration are presented in the *Groundwater Flow Model Documentation Package* (IT, 1996f). Pathlines were determined from the calibrated flow model using the USGS code MODPATH (Pollack, 1989).

The transport model simulates tritium concentration along a one-dimensional pathway defined by the flow model. A one-dimensional, mobile/immobile, finite-element model, MC_TRANS, contained in Appendix A, was developed to account for advection, dispersion, radioactive decay, and diffusion between the mobile and immobile phases. The mobile phase can be conceptualized as the portion of the aquifer through which the water flow occurs. This mobile phase may involve fractures or the more permeable portions of a porous medium. Advection is allowed only in the mobile phase. The immobile phase is conceptualized as the matrix between the fractures or mobile zones. Uncertainty in model parameters is incorporated via either a Monte Carlo or Latin hypercube sampling technique (Iman et al., 1980).

The steps of the transport modeling task are:

- Calculate groundwater pathlines from underground test locations at or below the water table using a three-dimensional groundwater flow model.
- Select three pathlines for transport simulations.
- Identify parameter uncertainty for flow and transport parameters.
- Determine the form and uncertainty of the source term.
- Simulate the transport of tritium along the pathways, accounting for uncertainty with a Monte Carlo or Latin hypercube approach.
- Determine the downgradient distance beyond which tritium concentration does not exceed 20,000 picoCuries per liter (pCi/L).
- Determine how uncertainty in selected input parameters impacts results.
- Provide tritium concentration data at preselected downgradient locations for risk assessment. The downgradient locations are chosen to aid in the presentation of results and to correspond to risk receptor locations.

1.4 *Documentation Organization*

This document is organized in eight sections which have the following content:

Section 1.0 - Introduction

Section 2.0 - Quality Control

Section 3.0 - Transport model description

Section 4.0 - Data flow description

Section 5.0 - Transport parameters and uncertainty discussion

Section 6.0 - Transport calculations, results, and sensitivity

Section 7.0 - Summary of Results

Section 8.0 - References cited

2.0 Quality Control

A Quality Assurance (QA) Program has been developed for all activities performed under the Environmental Restoration (ER) Program including the UGTA Subproject. QA measures taken to control quality while products are generated under the different data analysis subtasks, include data documentation evaluation, data quality evaluation, checking procedures, software quality assurance, use of standard methodologies, technical and peer reviews, and corroboration through the models.

2.1 Data Documentation Evaluation

During the documentation evaluation process of data, flags are assigned. The five levels of Data Documentation Evaluation Flags (DDE_F) are briefly presented here:

- Level 1: Data collected in accordance with Nevada Environmental Restoration Program (ERP) subproject quality assurance plans, approved Nevada procedures, and/or participant-specific procedures. This ranking indicates that all supporting documentation for the data is on file and is available for review by data users.
- Level 2: Data collected in accordance with approved plans and procedures as required for Level 1, with the exception that one or more documentation requirements may be deficient in some way. Examples of data documentation deficiencies may include lost or destroyed field-data collection forms, or data acquired using interim or draft procedures.
- Level 3: Data collected using accepted scientific methodology (e.g., American Society of Testing Materials [ASTM]), U.S. EPA methods, USGS procedures) and are accompanied by supporting and corroborative documentation such as testing apparatus diagrams, field or laboratory notes, and procedures. Documents which have been referenced to qualify submitted data as Level 3 are noted and described in Part II of the Data Information Form (DIF).
- Level 4: Data collected by a participating Nevada ERP organization or another organization not associated with the Nevada ERP prior to the issuance and implementation of project-approved standard policies, procedures, or practices governing data acquisition and qualification. The methods of data collection are documented and traceable; however, the validity and prudence of data use or compliance with referenced procedures is indeterminate. Supporting documentation may or may not exist. Documents which have been referenced to qualify submitted data as Level 4 are noted and described in Part II of the DIF.

Level 5: Data are obtained under unknown, undesirable, or uncertain conditions. When data documentation is unknown, any available supporting or helpful descriptions of the intended use and conditions of data capture should be described and listed in Part II of the DIF.

2.2 Data Quality Evaluation

The criteria used for evaluating different required data were dependent on the type of data and the intended use of the data. Thus, various criteria were used to evaluate data quality. The general procedure assigned one or more flags to each record compiled in the database. These flags indicated the data quality or suitability of the individual data record for the intended usage. Subtask-specific data quality evaluation procedures are described in detail in the corresponding subtask documentation package.

2.3 Checking Procedures

Various checking procedures were designed for quality control purposes. Checking procedures applicable to the UGTA data analysis subtasks include those developed for transcription of data, generation of figures, tables and logs, and performance of calculations. Data compiled by project personnel were subjected to the checking procedures before being added to the ER database. However, the bulk of the ER database is comprised of external digital databases developed by agencies external to the UGTA Subproject, mainly the National Water Information System (NWIS) data from the USGS. Internal procedures do not govern other ER database participants; therefore, their data were not subjected to the checking procedures described here.

2.4 Software Quality Assurance

Various computer codes were developed in-house to aid in the data analysis subtasks. Codes developed specifically for the UGTA Subproject were subjected to software quality assurance requirements such as validation/verification, preparation of operating manuals, and documentation of the theoretical basis for the calculations. Codes developed to load data and perform unit conversion codes used on the Environmental Restoration Data Base Management System (ERDBMS) were also checked.

2.5 Standard Methodologies

Only standard and widely accepted methodologies were used in the development of the interpretive products. The various methodologies used are too numerous to list here; however, they are described and referenced, where appropriate, in the sections discussing their use in the data analysis process.

2.6 *Technical and Peer Reviews*

The review process constitutes an important measure of product quality, and was used throughout the performance of the data analysis activities. The review process consists of both technical and peer reviews. The technical review process is internal and is performed by qualified personnel. The peer review process is intended to complement the technical review process and is usually performed by individuals who are independent of the project.

Products generated under the Data Analysis Task were continuously reviewed to identify any major flaws in the conceptual model of the groundwater flow system at an early stage. The geologic cross sections and maps were subjected to technical and peer reviews several times during their development.

The hydrogeologic cross sections were also subjected to technical and peer reviews during their generation. The peer reviews were conducted during periodic meetings including International Technology (IT) geologists and hydrogeologists and representatives from Raytheon Services Nevada (RSN), the USGS, Desert Research Institute (DRI), Los Alamos National Laboratory (LANL), and Lawrence Livermore National Laboratory (LLNL). Comments were recorded, evaluated, and incorporated into the process of defining the hydrostratigraphy.

The map products were internally reviewed by project personnel and submitted for peer review along with this documentation package. Following receipt of comments from the peer reviewers about the documentation packages and the map products, comments were resolved either by explanation or by modification of the products. Formal responses were prepared.

2.7 *Corroboration of Data through the Models*

This step is completed as the groundwater flow model is calibrated. During the calibration process, interpretations are tested and modified as required. This step is particularly applicable to the geologic model where extensive interpretation is necessary. The geologic model has been modified during the calibration of the flow model because initial interpretations in some areas did not allow for the matching of the predicted and observed hydraulic heads within the existing levels of uncertainties.

3.0 Transport Model Description

3.1 Overview

In choosing a transport model for simulating the migration of tritium from underground test sites, consideration had to be given to the extremely variable nature of the aquifers of the Death Valley groundwater flow system. The heterogeneous nature of the aquifer system is reflected in the Hydrostratigraphic Units (HSUs) mapped in the geologic model which are described in [Table 3-1](#). For more information regarding the HSUs, the reader is directed to the *Regional Geologic Model Data Documentation Package* (IT, 1996, draft).

Within the flow system, some hydrostratigraphic units such as the alluvium can be treated as porous media, while other units, such as the welded tuffs, are dominated by fracture flow. In the fractured units, the matrix between fractures is porous and saturated with water. Diffusion of tritium between the matrix and fracture is a mechanism that has been proposed by other researchers (Grisak and Pickens, 1980; Harrison et al., 1992; and Maloszewski and Zuber, 1991) to account for slower mean velocities of solutes relative to water.

To simulate the transport of tritium, a model was required that allowed for matrix diffusion, advection, radioactive decay, and dispersion. In addition, it was recognized that if other radionuclides were to be considered at a later time, capabilities for adsorption and tracking of daughter products would also be required. A final consideration in the selection of a model was the desire to investigate the impact of parameter uncertainty on the predicted tritium activities using a Monte Carlo or Latin hypercube approach. The Latin hypercube approach was selected for the regional simulations because it had fewer required realizations (thereby saving time) and it also incorporated correlation of parameters which was deemed necessary in some cases.

Although many of the requirements for simulating tritium migration in this complex aquifer system are met by some existing computer codes (e.g., SWIFT or its successors), these codes are typically very large, require large amounts of data, and would be very difficult to implement efficiently in a stochastic framework. The need to simultaneously model the necessary transport mechanisms and provide a computationally efficient tool for uncertainty analysis led to the selection of a one-dimensional model. The choice of a numerical approach over an analytical method was driven by the desire to account for the complexity of the NTS geology and hydrology.

Table 3-1
Hydrostratigraphic Units Used in the Modeling

Layer Number	Abbreviation	Description
20	AA	Alluvial Aquifer
19	TMA	Timber Mountain Aquifer (Uppermost Welded Tuff)
18	TC	Tuff Cone (Variable Welded and Nonwelded Tuffs)
17	TCBCU	Tuff Cone Basal Confining Unit (Nonwelded Tuffs)
16	TBAQ	Belted Range Aquifer
15	BCU	Basal Confining Unit (Nonwelded Tuffs)
14	BAQ	Basal Aquifer (Welded Tuffs)
13	VA	Volcanic Aquifer
12	VCU	Volcanic Confining Unit
11	VU	Volcanics Undifferentiated
10	TSDVS	Tertiary Sediments/Death Valley Sediments
9	LCA3	Lower Carbonate Aquifer (Yucca Flat Upper Plate) Upper Carbonate Aquifer in other NTS areas
8	UCCU	Upper Clastic Confining Unit
7	LCA	Lower Carbonate Aquifer
6	LCCU	Lower Clastic Confining Unit
5	LCA1	Lower Carbonate Aquifer (Upper Plate)
4	LCCU1	Lower Clastic Confining Unit (Upper Plate)
3	LCA2	Lower Carbonate Aquifer (Lower Plate)
2	LCCU2	Lower Clastic Confining Unit (Lower Plate)
1	I	Intrusives

To fulfill the requirements noted above, a one-dimensional model, MC_TRANS, was developed. It simulates the transport of decay-chains of dissolved chemical constituents within this complex groundwater system comprised of both fractured and porous media, where the diffusion of solute mass into matrix or dead-end pore space is an important factor. The model can be used in a deterministic mode or probabilistic mode, incorporating hydrologic uncertainty through either a Monte Carlo or Latin hypercube method. In the deterministic mode, the input parameters are assigned fixed values, and a single solution is determined. In the probabilistic mode, selected parameters are treated as random variables and are allowed to vary within limits defined by the mean, standard deviation, and/or the probability distribution of the parameter. These selected parameters are varied via a Monte Carlo or Latin hypercube sampling approach, and numerous solutions are determined corresponding to the different parameter values.

Chain-decay transport for up to four constituents can be solved for using MC_TRANS. The formulation in MC_TRANS allows for the definition of the dual porosity system geometry using either a parallel fracture/block matrix conceptual model or an interconnected/dead-end pore space conceptual model. The geometry of the parallel fracture/block matrix model is defined by two parameters, the block width and fracture aperture. The geometry of the interconnected/ dead-end pore model is described by the effective porosity and matrix block width. The dual porosity representation of the groundwater system assumes complete mixing across the width of the fracture and in the matrix. A mass transfer coefficient controls interaction between fracture and matrix (or interconnected and dead-end pore space) systems. The solutes are subject to advection, sorption, and first order radioactive decay in the fracture (or interconnected pore space); and sorption and first order radioactive decay in the matrix (or dead-end pore space) domain.

Parameter uncertainty may be described using several types of statistical distributions including normal, lognormal, uniform and user defined distributions. MC_TRANS incorporates flow model sensitivity coefficients when evaluating the uncertainty in simulation results associated with the uncertainty in flow parameters used in the three-dimensional flow model. Additionally, the model contains a sensitivity analysis option, where uncertainty can be considered for user defined subsets of a set of realization parameters.

The Galerkin finite-element approach is used to solve the partial differential transport equations. The source input may be a prescribed concentration, a mass flux boundary condition, or an instantaneous initial condition. The model allows for zonal variations of all input parameters

including deterministic variables and uncertainty parameters. A detailed documentation of the transport model MC_TRANS is given in Appendix A.

Several of the capabilities of MC_TRANS were not used in the simulation of tritium transport. Tritium does not decay into daughter products that must be tracked separately and tritium is not adsorbed to geologic media. Therefore, neither the decay chain nor the sorption capability of MC_TRANS were needed for this work. The dual porosity formulation was chosen in favor of the discrete fracture/matrix formulation. The simulation of fractured geologic units is easily accomplished with either formulation, but the simulation of porous units (such as alluvium) is best accomplished with the dual porosity formulation.

3.2 Solute Transport Governing Equations

Two governing equations are required to describe the transport of tritium in a dual porosity system; one for the mobile domain and a second for immobile domain. The two equations are linked by a first order mass transfer term. The mobile domain represents the fractures or rapid flow conduits of the hydrostratigraphic units, and the immobile domain refers to the matrix between fractures or conduits.

The governing equations as presented in the following text are in the form applied to the transport of tritium. See Appendix A for a presentation of the full governing equations including adsorption and decay chains.

3.2.1 Mobile Domain

The regional simulations presented in this documentation were performed using the dual porosity formulation where the mobile domain is characterized by an effective porosity. In a fractured hydrogeologic unit such as a welded tuff, the effective porosity reflects the ratio of fracture width to the sum of the fracture width and fracture spacing (i.e., ratio of fracture void volume to total volume). In the dead-end pore space model, the effective porosity represents the mobile zone portion of the porosity. For a one-dimensional, dual porosity formulation, the governing equation for mass transport of a dissolved constituent in the mobile zone (the fracture zone or dynamic zone of a dead-end pore space model) can be written in the form:

$$\theta_m \frac{\partial C_m}{\partial t} = D_m \frac{\partial^2 C_m}{\partial x^2} - \frac{\partial(qC_m)}{\partial x} - \theta_m \lambda C_m - \alpha(C_m - C_{im}) \quad (1)$$

Where:

- C_m = activity of the radionuclide in fluids in the mobile zone [Activity/L³];
- C_{im} = activity of the radionuclide in the fluid of the immobile zone [Activity/L³];
- θ_m = effective porosity of the mobile zone [dimensionless];
- D_m = mobile-zone hydrodynamic dispersion coefficient [L²/T];
- q = Darcy flux [L/T];
- α = mobile-immobile zone mass transfer coefficient [1/T];
- λ = first-order constituent decay rate [1/T];
- t = time [T]; and
- x = space coordinate [L]

and

$$D_m = \theta_m \alpha_l v + \theta_m \tau D^\circ \quad (2)$$

In Equation (2), α_l is the longitudinal dispersivity [L]; v is the average linear velocity of groundwater in the mobile zone [L/T]; τ is the mobile zone tortuosity; and D° [L²/T] the molecular diffusion coefficient in free-standing solution. The average linear velocity is defined as:

$$v = \frac{q}{\theta_m} \quad (3)$$

where θ_m , the mobile zone porosity, is also called the effective porosity in this document. The first order radioactive decay constant is defined as:

$$\lambda = \frac{\ln 2}{T^{1/2}} \quad (4)$$

where $T^{1/2}$ is the half life of the radioactive constituent and $\ln 2 = 0.69314718$.

3.2.2 Immobile Domain

The governing equation for each constituent in the immobile zone can be written as:

$$\theta_{im} \frac{\partial C_{im}}{\partial t} = - \theta_{im} \lambda C_{im} + \alpha(C_m - C_{im}) \quad (5)$$

Where:

- C_{im} = average activity of radionuclide in the pore fluid of immobile zone [Activity/L³];
- θ_{im} = effective porosity of the matrix immobile zone [dimensionless]; and
- α = first-order mass transfer coefficient [1/T].

The mass transfer coefficient α is regarded as an empirical constant (Coats and Smith, 1964).

There are several definitions for α including: van Genuchten (1981) and Sudicky (1990). For the dual porosity approach, the formulation of Sudicky is implemented:

$$\alpha = \frac{3 \theta_{im} D_m^*}{B^2} \quad (6)$$

where B is the average length dimension of the immobile domain (i.e., distance between fractures), and D_m^* is the effective diffusion coefficient in the porous matrix. For isotropic media, D_m^* is defined as (Bear, 1972),

$$D_m^* = \tau_{im} D^o \quad (7)$$

where τ_{im} is the tortuosity of the porous matrix. In the remainder of the text, the process of radionuclide movement into the immobile domain will be referred to as matrix diffusion.

3.3 Implementation of MC_TRANS in a Mixed, Single/Dual Porosity Flow System

The groundwater flow system at the NTS is complicated because some formations are characterized as fractured and others as porous. In the fractured formations, the dual porosity approach is implemented with flow in the fractures and diffusion through the fracture walls into

the surrounding matrix. In a fractured formation characterized by parallel fractures, the effective porosity can be calculated as:

$$\theta_m = \frac{b}{B + b} \quad (8)$$

Where:

b = fracture aperture (m); and

B = fracture spacing (m)

Alternatively, the mobile domain can be viewed as a tortuous, interconnected fracture network with an average open area represented by the effective porosity and an average block dimension (B). In the regional modeling, the dual porosity conceptualization is applied to the hydrostratigraphic units that are deemed to be fracture-flow dominated: the carbonate aquifer, the volcanic aquifer units, the intrusives, and the lower clastic confining unit. Other HSUs were also modeled as single porosity: the volcanic confining units, the tuff cones, and the upper clastic confining unit. These units are typically fine grained and could be modeled as porous units with relatively large effective porosity.

Hydrostratigraphic units such as the alluvium, Tertiary and Death Valley Sediments, and the volcanics undifferentiated have been represented by porous units in the transport model. The porous units are modeled as single porosity by defining the porosity of the material as the effective porosity and giving the immobile zone porosity a value of zero. This makes the mass transfer coefficient equal to zero and effectively creates a single porosity unit.

3.4 Uncertainty Analysis

The simulation of advective-dispersive transport of a dissolved species, using MC_TRANS, is based upon a set of user defined parameters describing the medium-specific, chemical-specific and source-specific nature of the modeled system. Values of specified parameters cannot be exactly known because of errors in measurements, spatial and temporal variability in the media and/or practical restrictions in data gathering efforts. In many circumstances it becomes more appropriate to express the parameter values in terms of probability distributions as opposed to

expressing them in terms of deterministic values. Analysis of simulation results can then include the likelihood of events, allowing for an assessment of the impact of data variability on simulation results.

The primary objective of uncertainty analysis is to estimate the uncertainty in spatial distributions and/or temporal changes in distributions of dissolved species given the uncertainty in the input parameters (EPA, 1993). Simulation results reflect the cumulative probability distribution of resultant observation point concentrations. They may also reflect the functions of these concentrations (i.e., maximum concentrations) as determined from known or assumed distributions of one or more of the deterministic model input parameters. Mathematically, observation point concentrations can be viewed as functions of a vector of model input parameters (p_i), where one or more of the parameters p_i may be random variables described by a cumulative probability distribution function. The model can then be used to calculate the cumulative distribution function F_{cobs} of the observation point concentrations. MC_TRANS contains two options for evaluating F_{cobs} , a Monte Carlo option and a Latin hypercube sampling option. The sampling logic for the Monte Carlo option is self-contained within MC_TRANS. Five choices are available in the Monte Carlo option to describe the input parameter probability distributions: (1) constant, (2) normal, (3) log-normal, (4) uniform and (5) user-described histogram. For utilization of the Latin hypercube sampling option, an extra input file must be generated containing a list of all sampled parameters for each realization. A version of Sandia's Latin hypercube sampling (LHS) model (Davenport and Iman, 1982), modified for use in conjunction with MC_TRANS, can be used to generate the realization input. Choices available in the LHS option for describing the input parameter probability distributions include: (1) constant, (2) normal, (3) log-normal, (4) uniform, and (5) log-uniform.

Each sampling option has its own advantage. Monte Carlo analysis has the advantage of being more rigorous. The LHS method has the advantage of being more efficient and it allows for consideration of correlation between parameters.

For the tritium transport analyses described in this study, the LHS option was utilized. The choice of options was based upon the need for the more time-efficient method, and the desire to investigate the influence of correlation between certain parameters.

3.5 Data Requirements for the Transport Model

In MC_TRANS, the transport equations are evaluated using the Galerkin finite-element approach (see Appendix A). The one-dimensional spatial domain (as defined by MODPATH pathlines in

this study) is discretized into a series of equally-spaced one-dimensional finite elements, each delimited by end nodes. The finite elements span from a source inlet to the exit boundary. Properties for each deterministic run (or Monte Carlo or LHS realization) are defined as constant for each element. Initial conditions (mass in place) are defined at nodal points. The temporal terms are evaluated using a centered-in-time finite difference approximation. For each time step, the sets of mobile zone and immobile zone transport equations generated by the Galerkin scheme for each constituent are solved simultaneously.

3.5.1 Model Geometry and Time Stepping

The process of discretizing the one-dimensional domain and choice of finite-element grid spacing is controlled by the grid Peclet Number, $Pe = \Delta x / \alpha$. Huyakorn and Pinder (1983) suggest that the Peclet Number be smaller than 2 to minimize numerical oscillations (overshoot and undershoot). For this work, a criterion of $Pe = 2.0$ was implemented using the minimum value for α of 50 meters (m) (164.1 feet [ft]). This fixed the grid spacing to be no larger than 100 m (328.1 ft). The time step is governed in part by the Courant Number, $Cr = V \Delta t / \Delta x$. Huyakorn and Pinder recommend a value of Cr less than one. However, our experience has shown that larger values of Cr can be used without introducing significant numerical oscillations. The time step was adjusted to keep the Courant Number less than one for most of the simulations.

The maximum length of the flow path was no greater than 99 kilometers (km) (61.5 mi), based on some preliminary scoping simulations. The maximum simulation time of 250 years was selected because it represents approximately 20 tritium half-lives. Assuming an initial cavity activity of 1×10^9 pCi/L, the activity decays below 20,000 pCi/L, which is the regulatory limit, in slightly less than 200 years.

3.5.2 Transport Parameters

The transport model requires the parameters identified in Equations 1-8 to be defined at each node or element. Specific discharges (Darcy velocities) are defined as constant for each element. Values are based upon the Darcy velocity field generated in the three-dimensional MODFLOW finite difference calibrated flow model of the NTS (IT, 1996f). A detailed description of the generation of the elemental velocities used in the transport model is presented in Section 4 of this report. The uncertainties associated with the velocities used in the transport model are defined using flow sensitivity parameters derived from a flow model sensitivity analysis. Uncertainties in flow parameters, such as horizontal hydraulic conductivity, vertical/horizontal conductivity anisotropy, and the coefficient of decrease of horizontal conductivity with depth, in conjunction with the associated sensitivity parameters, are used to describe the stochastic nature of the flow

field. A detailed description of the flow model parameter uncertainties and derivation of the flow model sensitivity parameters is presented in Section 5 of this report. The other transport parameters needed for the tritium analysis include: effective porosity, matrix porosity, longitudinal dispersivity, diffusion coefficient, fracture spacing and initial tritium activities. The probabilistic distributions for these transport parameters and the sources of these distributions are also described in Section 5.

MC_TRANS requires data for options that are not selected. The fracture aperture was given a value of 2, as per instructions in Appendix A. By assuming a fracture aperture of 2 the fracture half-width becomes equal to one, and the model defaults to the dead-end pore space model, where the geometry is described by the effective porosity and fracture spacing.

4.0 Data Flow

The data flow used in the generation of the final results is given in [Figure 4-1](#). The process of generating the tritium breakthrough curves involved four main tasks with several smaller steps within each task. The tasks are:

- Determine groundwater pathlines.
- Identify parameter values and uncertainty.
- Define model grid system and create input files.
- Calculate tritium concentration as a function of distance and time along each path.

A description of each task is contained in the flowchart shown in [Figure 4-1](#). A more detailed chart of the data flow, along with the various computer programs, is given in [Appendix C](#).

4.1 Groundwater Pathlines

Groundwater pathways are determined by tracking the movement of groundwater through the three-dimensional groundwater flow system beginning at a location of interest. For this work, the locations of interest are selected underground test locations. A particle tracking program, MODPATH, takes the output of MODFLOW, the program used to model the groundwater flow system, and calculates the groundwater pathways. In MODPATH, an imaginary particle is followed as it flows through the groundwater system as defined by the flow model. Using the PATHLINE output option of MODPATH, the location coordinates of the particle, in model space, are recorded where the particle crosses a cell boundary. Intermediate locations within cells are identified and compared with the GEOLOGIC MODEL to determine the HSU at each intermediate location. Effective porosity values, appropriate for each hydrostratigraphic unit, are assigned along the pathline, and travel times are computed. The travel times and travel distances from each starting location are summarized for each of the pathlines. Three shot locations (TYBO, HOUSTON, and BOURBON), selected to represent pathlines from Western and Central Pahute Mesa as well as from Yucca Flat, were then discretized for use by MC_TRANS.

4.2 Selection of Particle Starting Location

The regional transport simulations were planned to begin at three subregions, one each in Yucca Flat, Central Pahute Mesa, and Western Pahute Mesa. These starting locations represent the primary testing areas and they are of significant importance in assessing the potential

Figure 4-1
Transport Model Data Flow

impact of tritium transport away from the testing areas. A conservative approach was chosen to ensure that the impacts of underground testing were not underestimated. In each of the three locations, the travel distance of a particle beginning in each of the test locations was determined for fixed times of 100 and 200 years. In each of the three subregions, one of the underground tests with a greater travel distance was chosen to represent the pathway from that subregion.

4.2.1 Underground Test Locations

A database containing underground test location information was provided by LLNL. The database was refined by the DOE and was modified to include estimated water table elevations at each test location as determined by the USGS. The database contains the event name, hole name, date expended, depth of burial, estimated water table elevation, surface elevation, coordinates of the collar, and the working point (if available) in Nevada State Plane coordinates in addition to the stratigraphic and lithologic description of the working point geology.

The database contains 917 entries for devices expended on the NTS. When compared with the DOE list of announced tests through September 1992, (DOE, 1994) it was noted that 19 tests were missing from the LLNL database, and one non-nuclear test had been added. Of the 19 missing tests, 17 were older surface or cratering experiments that were not expected to influence the groundwater.

After the comparison to the DOE document (1994), the database was expanded to include conversion of coordinates from the Nevada State Plane to Universal Transverse Mercator (UTM) and to include an estimate of yield and cavity radius.

4.2.2 Test Yield and Cavity Radius

Before the likelihood of a particular underground test to have contaminated groundwater can be decided, it is first necessary to determine if a test may have encountered the water table. Tests below the water table were automatically included. The decision to include tests above the water table depended on an estimate of the location of the bottom of the shot cavity. To determine the cavity radius, the formula used was recommended by the DOE/NV Containment Evaluation Panel (presented by Smith [1995]), in a personal communication from W. Hawkins, LANL). The formula relates the radius of the cavity to the yield of the device and the depth of burial as:

$$R_c = \frac{70.2 w^{1/3}}{(\rho h)^{1/4}} \quad (9)$$

Where:

R_c = the cavity radius (m);

w = the yield of the shot (kilotons);

ρ = average overburden density (gram(s) per cubic centimeter [g/cm^3]); and

h = depth of burial (m).

The yield of most individual shots is classified information. For the purposes of the regional modeling, the yield was taken as the reported value or, more commonly, as the upper bound of the reported range found in the DOE document (1994). Though the size of the event will be overestimated as a result, this overestimation is necessary to keep the calculations conservative. To determine if a shot cavity may have extended to the water table, the calculated cavity radius was multiplied by 2. The factor of two represents the region surrounding the detonation where significant fractures are created (Smith, 1995). The depth of burial was provided by LLNL in the database. The overburden density was taken to be $2.2 \text{ g}/\text{cm}^3$ ($137 \text{ lb}/\text{ft}^3$). The number is based on a grain density of $2.65 \text{ g}/\text{cm}^3$ ($165 \text{ lb}/\text{ft}^3$) and a porosity of 17 percent. Smith (1995) presents slightly smaller values for *in situ* density of between 1.8 and $1.9 \text{ g}/\text{cm}^3$ (112 and $119 \text{ lb}/\text{ft}^3$). If these smaller density values are used, a cavity radius that is larger by approximately 4 percent will result, but this is not considered significant.

Following the additions, the database was sorted by hole name rather than event name in order to group tests by NTS area. The expanded and sorted database is given in Appendix B.

4.2.3 Selection of Shots Below the Water Table

The first step in the selection process is to identify all the shots that may have encountered the water table. Three vertical locations were identified for each event: the working point. One location twice the cavity radius above and one location twice the radius below the working point (Figure 4-2). Each point was compared to the USGS-estimated water table elevation using the code GWSHOTS (Appendix A). The choice of three vertical locations, even for deeply buried shots well below the water table, is based on an observation that several of the large shots extend across several hydrostratigraphic units and flow model layers. There are expected to be differences in the transport due to vertical placement of the starting point.

Figure 4-2
Diagram of a Detonation Below the Water Table

For tests with working points above the water table, those within two cavity radii of the water table (Figure 4-3) are included for later analysis, while those greater than two cavity radii above the water table (Figure 4-4) are excluded.

4.3 *MODPATH Simulations*

The starting locations defined using the process above were used in the USGS particle tracking program, MODPATH, to simulate the movement of particles through the groundwater flow system. To define these pathlines, the required MODFLOW files from the calibrated flow model were used along with a porosity of 1.0 for all points in the system. The PATHLINE option of MODPATH generates an output file (pathline) giving the location and travel time where a particle crosses a cell boundary. A porosity of 1.0 was used, therefore, the distance traveled across a cell divided by the travel time is the specific discharge, not the velocity. The calculation of solute velocity is part of a later step.

4.4 *Calculation of Intermediate Pathline Locations*

The output of MODPATH, using the PATHLINE option, provides information only at cell boundaries. The numerical discretization for the transport model is finer than for the flow model due to Peclet Number constraints identified above; therefore, it is necessary to provide information at intervals finer than the flow model cell spacing. To provide information at a finer resolution, MODPATH was modified to store additional information in the file pathline. This additional information is used to calculate pathline locations at intermediate locations inside each cell so that the lateral spacing between points is no greater than 100 m (328.1 ft) and so that vertical spacing between points is no greater than 10 m (32.8 ft). These intermediate points were calculated using the same algorithm as in MODPATH and were checked against MODPATH by comparing values at the cell boundaries.

4.5 *Identification of Hydrostratigraphic Units Along a Pathline*

To determine the velocity along a pathline, it is necessary to assign an effective porosity value to each point along the pathway. The program VELOCITY in Appendix B serves two purposes. First, in the initial analysis of the flow system, it is used to determine the velocity along each flowpath in order to select the three paths for transport simulations. Second, it generates the input file used by the transport model preprocessor. In the first case, the effective porosity is used to calculate velocity; in the second case, porosity is taken to be 1.0 everywhere, so that

Figure 4-3
Diagram of a Detonation Within Two Cavity Radii of the Water Table

Figure 4-4
Diagram of a Detonation Above the Water Table

specific discharge is provided to the transport model. Porosity is provided separately to the transport model, thus it was not necessary to determine velocity when creating the transport model input file.

For these regional simulations, the effective porosity of each HSU will be treated as a spatially uniform value. This assumption is used in VELOCITY and in the transport model to be described later. Thus, the porosity at any point along the pathline is uniquely defined by the hydrostratigraphic unit. [Figure 4-5](#) is a diagram of a pathline segment through a model cell in which an HSU boundary is crossed. In this picture, the porosity could differ in each HSU, and the velocity could change within the cell due to HSU changes. To accomplish this, the HSU at each point along the pathway must be known. Using the refined pathlines, the location of each point was compared with the geological model used in the regional groundwater flow simulations. The HSU for each point is assumed to extend to the midpoint between neighboring points. The effective porosity corresponding to each HSU is given in [Table 4-1](#).

The values in [Table 4-1](#) have been optimized to emphasize the most important aquifer units. Several of the VCUs were given smaller effective porosities than the matrix porosity values typically associated with the effective porosity of porous units. These changes were made to reduce the impact of a path that was predominantly in an aquifer unit, but may have crossed a small section of confining unit. As will be discussed further in Section 6, differences in the discretization of the geologic model and the groundwater flow model occasionally led to a situation where a path appeared to enter a confining unit for a short distance and then return to an aquifer. In many cases the flow through the confining unit was not physically realistic. Therefore, using a smaller effective porosity for some of the confining units reduced the impact of the differences in discretization.

Data from the tracer experiment at the Cambric site (Burbey and Wheatcraft, 1986) suggest that effective porosity for the alluvium is on the order of 30 to 35 percent. By analogy, the porosity of other porous units such as the Tertiary and Death Valley Sediments, were also assigned a porosity of near 30 percent. The undifferentiated volcanics were also assumed to be porous units and assigned a porosity of 19 percent, which falls within the range for volcanic units as documented in the *Transport Parameter and Source Term Data Subtask Documentation Package* (IT, 1996e). For porous units such as alluvium, the effective porosity is assumed to be approximately equal to the matrix or bulk porosity.

Flow Model Cell

Figure 4-5
Diagram of a Pathline Segment Through a Flow Model Cell

Table 4-1
Effective Porosity Values for Each Hydrostratigraphic Unit

Hydrostratigraphic Unit	Abbreviation	Effective Porosity	Source
Alluvial Aquifer	AA	0.30	Cambric Expmnt.
Uppermost Welded Tuff	TMA	0.0005	Cubic Law Estimate
Tuff Cones	TC	0.01	Professional Judgement
Non-Welded Tuffs	TCBCU	0.06	Professional Judgement
Welded Tuffs above BCU	TBAQ	0.0005	Cubic Law Estimate
Non-Welded Tuffs	BCU	0.07	Professional Judgement
Welded Tuffs	BAQ	0.0005	Cubic Law Estimate
Volcanic Tuff Aquifer	VA	0.001	Professional Judgement
Volcanic Tuff Confining Unit	VCU	0.03	Professional Judgement
Volcanics Undifferentiated	VU	0.19	Professional Judgement
Tertiary Sediments/Death Valley Section	TSDVS	0.30	Cambric Expmnt.
Lower Carbonate Aquifer (Yucca Flat Upper Plate), Upper Carbonate Aquifer in the NTS Area	LCA3	0.005	Cubic Law Estimate
Upper Clastic Confining Unit	UCCU	0.02	Professional Judgement
Lower Carbonate Aquifer	LCA	0.005	Cubic Law Estimate
Lower Clastic Confining Unit	LCCU	0.01	Professional Judgement
Lower Carbonate Aquifer (Upper Plate)	LCA1	0.005	Cubic Law Estimate
Lower Clastic Confining Unit (Upper Plate)	LCCU1	0.01	Professional Judgement
Lower Carbonate Aquifer (Lower Plate)	LCA2	0.005	Cubic Law Estimate
Lower Clastic Confining Unit (Lower Plate)	LCCU2	0.01	Professional Judgement
Intrusives	I	0.0005	Cubic Law

For units such as the Lower and Upper Carbonate Aquifers and the Timber Mountain, Belted Range and Basal Volcanic Aquifers that are dominated by fracture flow, the effective porosity is more closely approximated by the fracture porosity. The cubic law for estimating the hydraulic conductivity of fractured units was used to provide an initial estimate of the effective porosity of a number of the units. For the welded tuff aquifers, a fracture porosity of 0.0005 is used. This value is determined from published estimates (as presented in the *Transport Parameter and Source Term Data Subtask Documentation Package*) (IT, 1996e) as well as from application of the cubic law to flow in fractures. Assuming parallel fractures, the bulk hydraulic conductivity (K) of the HSU can be related to the fracture aperture via the cubic law of flow in fractures (de Marsily, 1986).

$$K = \frac{\rho g b^3}{12\mu B} \quad (10)$$

Where:

- K = bulk hydraulic conductivity (L/T);
- g = acceleration of gravity (L²/T);
- b = fracture aperture (L);
- ρ = density of fluid (M/L³);
- μ = viscosity (M/(L-T)); and
- B = fracture spacing (L).

The bulk hydraulic conductivity (K) is representative of the value assigned in the three-dimensional flow model. It is recognized that estimating the fracture aperture via the cubic law is an approximation because the concept of parallel fractures extending over many kilometers is unrealistic. Nonetheless, it provides a useful initial estimate of appropriate values for effective porosity in the fractured units. Table 4-2 contains the corresponding effective porosities from Equations 8 and 10 for different values of bulk hydraulic conductivity, using different values of fracture spacing. As will be shown in Section 5, the average fracture spacing for the carbonate rocks is approximately 0.25 m (0.8 ft) with a range of 0.03 to 1.5 m (0.1 to 4.9 ft). For the volcanic rocks, the mean fracture spacing is approximately 3.2 m (10.5 ft) with a range of 0.3 to 9.8 m (1 to 32.2 ft).

Table 4-2
Relationship of Hydraulic Conductivity to Fracture Porosity

Fracture Spacing (m)	Bulk Hydraulic Conductivity Meter(s) per day (m/d)	Fracture Porosity
0.03	0.29	0.002
0.03	18.78	0.007
0.03	63.38	0.01
0.03	293.41	0.02
0.26	0.3	0.0004
0.26	17.33	0.002
0.26	270.84	0.004
0.26	33855.23	0.02
1.0	0.07	0.0001
1.0	8.8	0.0005
1.0	70.42	0.001
1.0	8,802.36	0.005
1.0	70,418.88	0.01
1.6	0.04	0.00006
1.6	22.53	0.0005
1.6	44.01	0.0006
1.6	5,501.48	0.003
5.0	0.11	0.00004
5.0	1.76	0.0001
5.0	14.08	0.0002
5.0	1760.47	0.001
10.	0.88	0.00005
10.	7.04	0.0001
10.	880.24	0.0005

In the flow model, the hydraulic conductivity of the volcanic aquifer units along the TYBO and HOUSTON flow paths is in the range of 3 to 12 m/d (9.8 to 39.4 ft/d), with values on the order of 60 m/d (196.9 ft/d) used for the LCA along the BOURBON pathline. Using the range of fracture spacing and bulk hydraulic conductivity, it can be seen from the data in [Table 4-2](#) that the effective porosity range of 0.0006 to 0.01 is reasonable for the carbonates and the range of 0.0002 to 0.002 is reasonable for the volcanics. The effective porosity values in [Table 4-1](#) for both the carbonates and the volcanic aquifer units fall within the respective ranges.

4.6 Calculation of Advective Velocity Along a Flow Path

The velocity at each point along the flow path was determined by taking the specific discharge divided by porosity at that point. For each pathline as defined in MODPATH, the velocity between each point along the pathline was derived by taking the distance along the path and dividing by the travel time. The cumulative travel time to any downgradient point is determined by summing the travel times along the intermediate points. Program TIMEDIST ([Appendix C](#)) takes the pathvel.dat file and creates individual data files for each pathline. This allows for individual processing of the pathline information and for generation of output plots for selected paths.

4.7 Selection of Starting Locations for Regional Modeling

The purpose of the regional transport modeling is to provide an assessment of the likely travel distance for tritium over a period of approximately 200 years. Using this approach, travel distances over a 100- and 200-year period were calculated for each event below the water table. Up to three starting locations were defined for each shot (top, working point, and bottom) where the top and bottom locations are located two cavity radii above and below the working point, respectively. Only points below the water table were considered, so for some shots, only the bottom of the cavity is used, while for others two or all three of the locations were used. The program PSUMRY ([Appendix C](#)) summarizes all the pathline information into an easily interpreted form. [Appendix D](#) is the output of PSUMRY and lists all the shots potentially in contact with the groundwater system and the travel distances for selected times. These were grouped by NTS area, and three locations were chosen for regional modeling.

The three pathways selected for the regional scale modeling pass through TYBO in Western Pahute Mesa, HOUSTON in Central Pahute Mesa, and BOURBON in Yucca Flat. Shots with greater downgradient travel distance, which were closer to the southern boundary of an area, were given priority. However, other factors such as the geology of the shot or the ability of the shot to represent a general flow area were also considered. The pathlines chosen for these

simulations are given in Figure 4-6 along with several other paths used during sensitivity analyses of the groundwater flow model.

4.8 Evaluation of Upgradient and Downgradient Tests Along the Same Pathline

Prior to selecting the starting locations, an evaluation of upgradient and downgradient events along BOURBON and TYBO paths was performed. If more than one event is along the same pathline, the plume may overlap and lead to greater downgradient migration than if only one event is considered. This analysis is intended to identify two events along the same pathline as BOURBON and TYBO. Preference was given to having one event upgradient and one downgradient. The analysis is limited to two events in order to assess the importance of multiple events in these regional simulations.

Two situations were considered: (1) pathlines from upgradient events that pass through TYBO or BOURBON and (2) downgradient events that are along the pathline from TYBO or BOURBON. The determination of whether a pathline intersected another event is based on the size of plumes originating from each of the events. The size of a plume is approximated by the three dimensional dispersion of a solute in a uniform flow field as described by Freeze and Cherry (1979, p 395). The zone in which 99.7 percent of the contaminant mass occurs is described by an ellipsoid with dimensions measured from the center of mass, of

$$d_x = \sqrt{2D_x t} : d_y = \sqrt{2D_y t} : d_z = \sqrt{2D_z t} \quad (11)$$

where:

- t = is the time since transport began
- d_x , d_y , and d_z = are the distances from the center of mass
- D_x , D_y , and D_z = are the dispersion coefficients in the longitudinal, transverse, and vertical directions.

Recognizing that the dispersion coefficient can be written as the dispersivity times the velocity ($D = \alpha v$) and that the product of velocity and time is travel distance ($d_t = vt$), the distances from the center of mass can also be written as

$$d_x = \sqrt{2\alpha_x d_t} : d_y = \sqrt{2\alpha_y d_t} : d_z = \sqrt{2\alpha_z d_t} \quad (12)$$

where:

α = is the dispersivity

d_t = is the travel distance downgradient from the source.

For this analysis, the emphasis was placed on the y and z direction interaction because dispersion in the transverse directions is generally less than in the longitudinal direction. The lateral dispersivity was taken as 50 m (164.1 ft) (longitudinal dispersivity/10) and the vertical dispersivity as 5 m (16.4 ft). The actual distance from the center of mass to the 99.7 percent distance will be larger by a distance approximately equal to the cavity radius (r_d). Therefore, if a pathline passes within a distance equal to $2d_y + 2r_d$ or $2d_z + 2r_d$ of another event working point, the two plumes are taken to be partially overlapping.

Two plumes that overlap at the maximum distance of $2d_y + 2r_d$ or $2d_z + 2r_d$ would not cause a significant change in either plume. If, however, the distance between the center of mass of the two plumes was significantly less than the maximum distance, then greater impacts would be expected. For example, approximately 68 percent of the mass of a plume lies within a distance of $d/3$. Two plumes that overlap at the $d/3$ distance would have a significant impact on the other. The criteria set for selection of overlapping plumes was $0.5(2d_y + 2r_d)$ or $0.5(2d_z + 2r_d)$; this will reduce the number of potential events and make the choice of two events more straight forward. Table 4-3 identifies: (1) pathlines from upgradient events that pass near BOURBON and TYBO, and (2) downgradient events that are along the pathlines from BOURBON and TYBO. The events in the table are only those for which both the vertical and horizontal distances were less than 50 percent of either $2d_y + 2r_d$ or $2d_z + 2r_d$.

For event TYBO, no events were downgradient; therefore, two events upgradient (KASH and PEPATO) were found to have paths that passed very close to the TYBO working point. Event KASH is 7 km (4.3 mi) upgradient of TYBO and PEPATO is 7.8 km (4.8 mi) upgradient. The paths from these two events passed closer to TYBO than the other paths. For event BOURBON, one event upgradient (KANKAKEE, 9 km [5.6 mi]) and one pair of events downgradient (MICKEY AND TORRIDO, 2.6 km [1.6 mi]) were determined as likely interacting plume sources. The event KANKAKEE was chosen as the upgradient plume because it passed closest to the BOURBON event. The choice of KANKAKEE over SHUFFLE was based the closer vertical position of KANKAKEE. Vertical dispersion is more limited than horizontal dispersion, therefore plumes are more likely to overlap laterally than vertically. The choice of a downgradient event associated with BOURBON is difficult. One event, PINSTRIPE, met the 50 percent criteria, but was rejected because the VCU will likely separate the two plumes and the speed of migration from the PINSTRIPE event is very slow. Two other nearby, downgradient events,

Table 4-3
Detonation Upgradient and Downgradient of BOURBON, HOUSTON, and TYBO

Primary Detonation	Multiple Detonation	Distance Along Pathline (km)	Direction	Vertical Distance from the Pathline to the Working Point (m)	Horizontal Distance from the Pathline to the Working Point (m)
BOURBON	STACCATO	7.9	Upgradient	126	874
BOURBON	CORNICE	7.5	Upgradient	97	304
BOURBON	KLICKITAT	6.6	Upgradient	149	309
BOURBON	CORDUROY	8.5	Upgradient	237	816
BOURBON	KANKAKEE	9.0	Upgradient	117	70
BOURBON	SHUFFLE	6.6	Upgradient	137	53
BOURBON	PINSTRIFE	24.8	Downgradient	434	527
BOURBON	MICKEY	2.5	Downgradient	166	824
BOURBON	TORRIDO	2.8	Downgradient	143	977
TYBO	PEPATO	7.8	Upgradient	36	59
TYBO	KASH	7.0	Upgradient	78	221
TYBO	CABRA	8.8	Upgradient	31	689
TYBO	JORUM	10.2	Upgradient	297	913
TYBO	BOXCAR	8.4	Upgradient	238	205

MICKEY and TORRIDO were outside the 50 percent distance criteria, but were included as being the most representative of downgradient events. These two events were combined into one hypothetical event located along the BOURBON path at a distance of 2.6 km (1.6 mi).

An approximate assessment of the interaction of the plumes was performed by allowing the TYBO path to originate at PEPATO, with KASH and TYBO both included along that same pathline. Similarly, the BOURBON path originated at KANKAKEE, with BOURBON and MIDKEY/TORRIDO included downgradient. A more complete analysis of the interaction of plumes will need to be part of CAU scale modeling.

The three pathlines chosen for these simulations beginning at PEPATO/TYBO, HOUSTON, and KANKAKEE/BOURBON are given in [Figure 4-6](#). The receptor locations at which breakthrough curves will be calculated and the locations of the other events (BOURBON, MICKEY/TORRIDO, KASH, and TYBO) are also shown on [Figure 4-6](#).

5.0 *Transport Parameters and Uncertainty*

Concurrent with the selection of the pathline starting locations, parameter values specific to the transport model such as porosity, dispersion coefficients, and tritium activities were tabulated and uncertainties assigned. Three pieces of information are required for each parameter: the probability distribution representing the parameter variability, the mean value, and the standard deviation.

The version of Sandia's Latin hypercube sample model (Davenport and Iman, 1982), used to generate the realization parameter file for MC_TRANS simulations contains five options for describing parameter probability distributions:

- Constant
- Normal
- Log-normal
- Uniform
- Log-uniform

A parameter that is constant is defined by a single value. An example is the radioactive decay coefficient which is assumed to be the same at all locations and all times for a specific radionuclide ($5.635 \times 10^{-2} \text{ y}^{-1}$ for tritium). Normal distributions are defined by a bell-shaped probability curve which is symmetrical about a mean value. The normal distribution for parameter x is defined by two parameters, the mean, m_x , and the standard deviation, S_x . A log-normal distribution is a skewed distribution with the log of the parameter x being normally distributed. If the $\log_{10}$ (or natural log) of the parameter x is y , the probability distribution of the transformed variable y is normal and can be described by the mean, m_y , and the standard deviation, S_y , of the transformed parameter. The relationships between the mean, m_y , and standard deviation, S_y , of the transformed parameter and the mean and standard deviation of the parameter x can be found in Appendix A. For a log-normally distributed parameter, MC_TRANS requires the mean of the $\log_{10}$ transformed parameter and the standard deviation of the $\log_{10}$ transformed parameter. A uniform distribution is a probability distribution, defined by a minimum value and a maximum value, in which all values within this range have an equal probability of occurring. A log-uniform distribution is a probability distribution, defined by a minimum transformed value and a maximum transformed value, in which all transformed values within this range have an equal probability of occurring.

5.1 Porosity

The effective (or mobile phase) porosity, and the matrix (or immobile phase porosity) are required for each HSU. As noted earlier, porous units are characterized by an effective porosity only. The mobile and immobile zone porosity information for each HSU is given in [Table 5-1](#).

These values differ somewhat from the values in [Table 4-1](#) for several reasons. The values in [Table 4-1](#) were optimized to accentuate the influence of aquifer units, where as in [Table 5-1](#) the effects of high porosity confining units were included explicitly. The effective porosity of the fractured volcanic units were set to the same value because the similarities in the fracture porosity of those units.

The effective porosity of the fractured units was assumed to be log-normally distributed, to avoid the problem of negative values during LHS, while allowing for a rather large range of variation. In addition, the effective porosity in fractured media is a function of the fracture aperture, which has been assumed to be log-normally distributed by other investigators such as Long and Billaux (1987). The $\log_{10}$ standard deviation of 0.21 to 0.25 is based on the criterion that the lower bound of the distribution not produce a value so low as to be physically unrealistic. The chosen range of variability is based in part on the sensitivity of the calculated hydraulic conductivity to aperture where a factor of 25 change in aperture (and effective porosity if the spacing is held constant at 1 m [3.3 ft]) produces more than a three-order of magnitude change in bulk hydraulic conductivity. The three-order-of-magnitude range in hydraulic conductivity is consistent with measured data presented in both the *Hydrologic Parameter Data Documentation Package* (IT, 1996d) and the *Groundwater Flow Model Documentation Package* (IT, 1996f).

The effective porosity of the porous units and the matrix porosity of the fractured units were characterized by a uniformly distributed porosity. The uniform distribution was chosen to provide a greater emphasis to values at the ends of the range of matrix porosity values.

5.2 Dispersion Coefficient

The dispersion coefficient in the mobile zone is the product of the dispersivity and the solute velocity, where the parameter required as input for the transport model is the dispersivity. Information gathered in and around the NTS and documented in the *Transport Parameter Data Documentation Package* (IT, 1996e) gave values up to 30 m (98.4 ft) for scales on the order of 120 m (393.7 ft) or less. Gelhar et al. (1992) summarized dispersivity data from many tracer

Table 5-1
Porosity Data and Uncertainty for Each HSU

	Mobile	Phase	Porosity	Immobile	Phase	Porosity
HSU	Distribution	Mean ^a	Standard Deviation ^a	Distribution	Mean	Standard Deviation
AA	Uniform	0.25	0.35	None	0	0
TMA	Log-normal	-2.87	0.21	Uniform	0.08	0.50
TC	Log-normal	-2.87	0.21	Uniform	0.12	0.45
TCBCU	Uniform	0.07	0.47	None	0	0
TBAQ	Log-normal	-2.87	0.21	Uniform	0.19	0.39
BCU	Uniform	0.33	0.45	None	0	0
BAQ	Log-normal	-2.87	0.21	Uniform	0.09	0.27
VA	Log-normal	-2.87	0.21	Uniform	0.08	0.45
VCU	Uniform	0.07	0.47	None	0	0
VU	Uniform	0.01	0.36	None	0	0
TSDVS	Uniform	0.25	0.35	None	0	0
LCA3	Log-normal	-2.46	0.25	Uniform	0.01	0.15
UCCU	Uniform	0.01	0.23	None	0	0
LCA	Log-normal	-2.46	0.25	Uniform	0.01	0.15
LCCU	Log-normal	-4.50	0.39	Uniform	0.01	0.09
LCA1	Log-normal	-2.46	0.25	Uniform	0.01	0.15
LCCU1	Log-normal	-4.50	0.39	Uniform	0.01	0.09
LCA2	Log-normal	-2.46	0.25	Uniform	0.01	0.15
LCCU2	Log-normal	-4.50	0.39	Uniform	0.01	0.09
I	Log-normal	-4.50	0.39	Uniform	0.001	0.01

^aFor a log-normal distribution, the log₁₀ mean and log₁₀ standard deviation are presented. For a uniform distribution, the mean and standard deviation represent the lower and upper bounds, respectively.

tests around the world and showed that dispersivity appears to be a scale-related parameter in that larger experiments tend to produce larger dispersivity values. A rule of thumb often applied is that longitudinal dispersivity is ten percent of the travel distance. For large travel distances on the order of tens of kilometers, the dispersivity could be as large as several thousand meters. However, there are no good tracer tests at large scales, and it is unlikely that the rule-of-thumb applies at such large scales. Choosing a dispersivity that is too large will make early breakthrough more likely, but will also decrease peak activity significantly.

For these regional simulations, a mean dispersivity of 550 m (1,804.6 ft) and a range of 100 to 1,000 m (328.1 to 3,281 ft) was chosen for the carbonate aquifer and the volcanic aquifers. For the clastic and volcanic confining units the range of dispersivity values were 100 to 500 m (328.1 to 1,640.5 ft) with a mean of 300 m (984.3 ft). The Alluvial Aquifer and the Death Valley Sediments were given a range of 50 to 500 m (164.1 to 1,640.5 ft). The larger dispersivities of the aquifer units reflects the larger fluxes in those units when compared with the confining units and alluvium. The range of values is consistent with the range reported in the literature and documented in the *Transport Parameters Documentation Package* (IT, 1996e). Despite the wide range of values chosen, the model predictions are not very sensitive to the value of the dispersivity as will be shown later when the results of sensitivity analyses are presented.

5.3 Advective Velocity

The advective velocity is determined in the transport model as the specific discharge divided by the effective porosity. The specific discharge is calculated from the calibrated regional groundwater flow model, as described in Section 4. The uncertainty associated with the specific discharge was incorporated in the transport model using a sensitivity coefficient approach where the specific discharge varies in the flow model due to input hydraulic parameter and recharge uncertainty. Primary parameters in the flow model considered were the hydraulic conductivity parameters of each HSU and a coefficient reflecting the proportional increase or decrease in recharge.

The hydraulic conductivity and the coefficient to decrease the conductivity with depth of each HSU or zone within each HSU, were considered variable and were included in the flow parameter sensitivity. A total of 117 parameters (58 HSUs or zones times two parameters plus the recharge parameter) were included in the transport model. The mean value of each parameter is taken directly from the calibrated flow model. From the *Hydrologic Parameter Data Documentation Package* (IT, 1996d), it is known that variations in hydraulic conductivity can be quite large, up to four orders of magnitude in many HSUs. The large variation in measured values is

representative of local scale variation and does not reflect the range of variation at the HSU scale as used in the flow model. Van Marcke (1983) addresses the question of averaging spatially variable processes, and shows that the variation of the averaged parameter is smaller than the variation of the measured values. The range of variation of the hydraulic conductivity is conservatively assumed to be about a 1 order of magnitude for the transport calculations. A corresponding $\log_{10}$ standard deviation of 0.25 was assigned for the $\log_{10}$ standard deviation of the horizontal hydraulic conductivity for all HSUs. The coefficient describing the reduction of bulk hydraulic conductivity (K) with depth was assigned a uniform distribution with a range of values consistent with the uncertainty in the slope of the log hydraulic conductivity with depth as given in the *Hydrologic Parameter Data Documentation Package* (IT, 1996d). During calibration of the groundwater flow model, it was observed that some of the hydraulic parameters could be varied by a factor of three and still achieve calibration. However, this was possible only when compensating changes to other parameters were also implemented. A list of the flow sensitivity parameters and the values chosen for simulation are given in Table 5-2.

Finally, it is recognized that recharge is also uncertain. As recharge increases, more water will flow through the groundwater system. To keep the model in calibration, the calibrated hydraulic conductivity would need to increase a proportional amount to keep the hydraulic gradient the same. Therefore, the dominant effect of changes in recharge can be approximated by multiplying the specific discharge by a recharge factor.

The uncertainty in hydraulic conductivity parameters is incorporated into the transport model via the specific discharge. To do so, the sensitivity of the three components of the specific discharge vector at each node in the flow model to changes in the hydraulic conductivity parameters of each HSU (or zone) was determined. The sensitivities of the three orthogonal components of the specific discharge vector at each node (I,J,K) can be represented by:

$$\begin{aligned} \text{SEN}_{n,x,I,J,K} &= \frac{\Delta q_{x,I,J,K}}{\Delta K_n} \\ \text{SEN}_{n,y,I,J,K} &= \frac{\Delta q_{y,I,J,K}}{\Delta K_n} \\ \text{SEN}_{n,z,I,J,K} &= \frac{\Delta q_{z,I,J,K}}{\Delta K_n} \end{aligned} \quad (13)$$

Table 5-2
Flow Model Parameters Incorporated in the Transport Model via Sensitivity
(Page 1 of 4)

Flow Model Parameter ^a	Mean Value from the Calibrated Flow Model	Distribution	Mean	Standard Deviation
Kh - I	0.001	log-normal	-3.07	0.25
Kh - LCCU - Zone 1	0.02	log-normal	-1.77	0.25
Kh - LCCU - Zone 2	0.2	log-normal	-0.77	0.25
Kh - LCCU - Zone 3	0.1	log-normal	-1.07	0.25
Kh - LCA - Zone 1	110.	log-normal	1.97	0.25
Kh - LCA - Zone 2	110.	log-normal	1.97	0.25
Kh - LCA - Zone 3	0.01	log-normal	-2.07	0.25
Kh - LCA - Zone 4	8.	log-normal	0.83	0.25
Kh - LCA - Zone 5	0.8	log-normal	-0.17	0.25
Kh - LCA - Zone 6	0.4	log-normal	-0.47	0.25
Kh - LCA - Zone 7	2.	log-normal	0.23	0.25
Kh - LCA - Zone 8	0.8	log-normal	-0.17	0.25
Kh - LCA - Zone 9	5.	log-normal	0.63	0.25
Kh - UCCU	0.002	log-normal	-2.77	0.25
Kh - LCA3	0.05	log-normal	-1.37	0.25
Kh - TSDVS - Zone 1	0.5	log-normal	-0.37	0.25
Kh - TSDVS - Zone 2	0.2	log-normal	-0.77	0.25
Kh - VU - Zone 1	1.5	log-normal	0.10	0.25
Kh - VU - Zone 2	0.2	log-normal	-0.77	0.25
Kh - VU - Zone 3	0.4	log-normal	-0.47	0.25
Kh - VCU - Zone 1	0.12	log-normal	-0.99	0.25
Kh - VCU - Zone 2	0.12	log-normal	-0.99	0.25
Kh - VA - Zone 1	2.	log-normal	0.23	0.25
Kh - VA - Zone 2	1.	log-normal	-0.07	0.25
Kh - VA - Zone 3	8.	log-normal	0.83	0.25
Kh - BAQ - Zone 1	35.	log-normal	1.47	0.25
Kh - BAQ - Zone 3	2.	log-normal	0.23	0.25
Kh - BAQ - Zone 4	2.	log-normal	0.23	0.25
Kh - BAQ - Zone 5	2.	log-normal	0.23	0.25
Kh - BAQ - Zone 6	0.2	log-normal	-0.77	0.25
Kh - BCU - Zone 1	0.3	log-normal	-0.59	0.25
Kh - BCU - Zone 3	0.3	log-normal	-0.59	0.25
Kh - BCU - Zone 4	0.3	log-normal	-0.59	0.25
Kh - BCU - Zone 5	0.3	log-normal	-0.59	0.25
Kh - BCU - Zone 6	0.2	log-normal	-0.77	0.25

Table 5-2
Flow Model Parameters Incorporated in the Transport Model via Sensitivity
(Page 2 of 4)

Flow Model Parameter ^a	Mean Value from the Calibrated Flow Model	Distribution	Mean	Standard Deviation
Kh - TBAQ - Zone 1	35.	log-normal	1.47	0.25
Kh - TBAQ - Zone 3	2.	log-normal	0.23	0.25
Kh - TBAQ - Zone 4	2.	log-normal	0.23	0.25
Kh - TBAQ - Zone 6	0.2	log-normal	-0.77	0.25
Kh - TCBCU - Zone 1	0.3	log-normal	-0.59	0.25
Kh - TCBCU - Zone 4	0.03	log-normal	-1.59	0.25
Kh - TCBCU - Zone 6	0.02	log-normal	-1.77	0.25
Kh - TC - Zone 1	35.	log-normal	1.47	0.25
Kh - TC - Zone 3	16.	log-normal	1.13	0.25
Kh - TC - Zone 4	20.	log-normal	1.23	0.25
Kh - TC - Zone 5	4.	log-normal	0.53	0.25
Kh - TC - Zone 6	0.2	log-normal	-0.77	0.25
Kh - TMA - Zone 1	35.	log-normal	1.47	0.25
Kh - TMA - Zone 2	12.	log-normal	1.00	0.25
Kh - TMA - Zone 3	16.	log-normal	1.13	0.25
Kh - TMA - Zone 4	20.	log-normal	1.23	0.25
Kh - TMA - Zone 6	0.2	log-normal	-0.77	0.25
Kh - AA - Zone 1	11.	log-normal	0.97	0.25
Kh - AA - Zone 2	4.4	log-normal	0.57	0.25
Kh - AA - Zone 3	0.01	log-normal	-2.07	0.25
Kh - AA - Zone 4	11.	log-normal	0.97	0.25
Kh - AA - Zone 5	11.	log-normal	0.97	0.25
Kh - AA - Zone 6	11.	log-normal	0.97	0.25
Id- I	0.0015	uniform	0.0	0.0030
Id- LCCU - Zone 1	0.0012	uniform	0.0	0.0024
4d- LCCU - Zone 2	0.0012	uniform	0.0	0.0024
Id- LCCU - Zone 3	0.0012	uniform	0.0	0.0024
Id- LCA - Zone 1	0.001	uniform	0.0	0.002
Id- LCA - Zone 2	0.001	uniform	0.0	0.002
Id- LCA - Zone 3	0.001	uniform	0.0	0.002
Id- LCA - Zone 4	0.001	uniform	0.0	0.002
Id- LCA - Zone 5	0.001	uniform	0.0	0.002
Id- LCA - Zone 6	0.001	uniform	0.0	0.002
Id- LCA - Zone 7	0.0026	uniform	0.0	0.0052
Id- LCA - Zone 8	0.001	uniform	0.0	0.002

Table 5-2
Flow Model Parameters Incorporated in the Transport Model via Sensitivity
(Page 3 of 4)

Flow Model Parameter ^a	Mean Value from the Calibrated Flow Model	Distribution	Mean	Standard Deviation
ld- LCA - Zone 9	0.001	uniform	0.0	0.002
ld- UCCU	0.0015	uniform	0.0	0.003
ld- LCA3	0.001	uniform	0.0	0.002
ld- TSDVS - Zone 1	0.004	uniform	0.0	0.008
ld- TSDVS - Zone 2	0.004	uniform	0.0	0.008
ld- VU - Zone 1	0.0026	uniform	0.0007	0.0045
ld- VU - Zone 2	0.0026	uniform	0.0007	0.0045
ld- VU - Zone 3	0.0026	uniform	0.0007	0.0045
ld- VCU - Zone 1	0.0026	uniform	0.0007	0.0045
ld- VCU - Zone 2	0.0026	uniform	0.0007	0.0045
ld- VA - Zone 1	0.0026	uniform	0.0007	0.0045
ld- VA - Zone 2	0.0026	uniform	0.0007	0.0045
ld- VA - Zone 3	0.0026	uniform	0.0007	0.0045
ld- BAQ - Zone 1	0.0026	uniform	0.0007	0.0045
ld- BAQ - Zone 3	0.0026	uniform	0.0007	0.0045
ld- BAQ - Zone 4	0.0026	uniform	0.0007	0.0045
ld- BAQ - Zone 5	0.0026	uniform	0.0007	0.0045
ld- BAQ - Zone 6	0.0026	uniform	0.0007	0.0045
ld- BCU - Zone 1	0.0026	uniform	0.0007	0.0045
ld- BCU - Zone 3	0.0026	uniform	0.0007	0.0045
ld- BCU - Zone 4	0.0026	uniform	0.0007	0.0045
ld- BCU - Zone 5	0.0026	uniform	0.0007	0.0045
ld- BCU - Zone 6	0.0026	uniform	0.0007	0.0045
ld- TBAQ - Zone 1	0.0026	uniform	0.0007	0.0045
ld- TBAQ - Zone 3	0.0026	uniform	0.0007	0.0045
ld- TBAQ - Zone 4	0.0026	uniform	0.0007	0.0045
ld- TBAQ - Zone 6	0.0026	uniform	0.0007	0.0045
ld- TCBCU - Zone 1	0.0026	uniform	0.0007	0.0045
ld- TCBCU - Zone 4	0.0026	uniform	0.0007	0.0045
ld- TCBCU - Zone 6	0.0026	uniform	0.0007	0.0045
ld- TC - Zone 1	0.0026	uniform	0.0007	0.0045
ld- TC - Zone 3	0.0026	uniform	0.0007	0.0045
ld- TC - Zone 4	0.0026	uniform	0.0007	0.0045
ld- TC - Zone 5	0.0026	uniform	0.0007	0.0045
ld- TC - Zone 6	0.0026	uniform	0.0007	0.0045

Table 5-2
Flow Model Parameters Incorporated in the Transport Model via Sensitivity
(Page 4 of 4)

Flow Model Parameter ^a	Mean Value from the Calibrated Flow Model	Distribution	Mean	Standard Deviation
ld- TMA - Zone 1	0.0026	uniform	0.0007	0.0045
ld- TMA - Zone 2	0.0026	uniform	0.0007	0.0045
ld- TMA - Zone 3	0.0026	uniform	0.0007	0.0045
ld- TMA - Zone 4	0.0026	uniform	0.0007	0.0045
ld- TMA - Zone 6	0.0026	uniform	0.0007	0.0045
ld- AA - Zone 1	0.0037	uniform	0.0	0.0074
ld- AA - Zone 2	0.0037	uniform	0.0	0.0074
ld- AA - Zone 3	0.0037	uniform	0.0	0.0074
ld- AA - Zone 4	0.0037	uniform	0.0	0.0074
ld- AA - Zone 5	0.0037	uniform	0.0	0.0074
ld - AA - Zone 6	0.0037	uniform	0.0	0.0074
Recharge Coefficient	1.0	log-normal	0	0.25

^aKh is the horizontal hydraulic conductivity (m/day) at the land surface in the conductivity versus depth equation, and ld is the coefficient for the decrease in Kh with depth. The HSU designations are defined in Table 3-1. For a uniform distribution, the mean is the lower bound and the standard deviation is the upper bound.

Where:

$SEN_{n,x,I,J,K}$ = the sensitivity of the x component of specific discharge in node I,J,K to flow parameter n;

$SEN_{n,y,I,J,K}$ = the sensitivity of the y component of specific discharge in node I,J,K to flow parameter n; and

$SEN_{n,z,I,J,K}$ = the sensitivity of the z component of specific discharge in node I,J,K to flow parameter n.

The perturbed quantities are given by:

$$\Delta q_i = q_i - \overline{q_i} \quad (14)$$

and

$$\Delta K_i = \frac{K_i - \overline{K}}{\overline{K}} \quad (15)$$

where q_i and K_i are the perturbed values, and the overbar variables are the base case values from the final calibrated model.

In the Latin hypercube portion of the transport model, each of the 116 different hydraulic conductivity parameters were allowed to vary simultaneously, and the resulting change in specific discharge was calculated as:

$$\begin{aligned} q_{x,I,J,K} &= \overline{q_{x,I,J,K}} + \Delta K_1 * SEN_{1,x,I,J,K} + \Delta K_2 * SEN_{2,x,I,J,K} + \dots + \Delta K_N * SEN_{N,x,I,J,K} \\ q_{y,I,J,K} &= \overline{q_{y,I,J,K}} + \Delta K_1 * SEN_{1,y,I,J,K} + \Delta K_2 * SEN_{2,y,I,J,K} + \dots + \Delta K_N * SEN_{N,y,I,J,K} \\ q_{z,I,J,K} &= \overline{q_{z,I,J,K}} + \Delta K_1 * SEN_{1,z,I,J,K} + \Delta K_2 * SEN_{2,z,I,J,K} + \dots + \Delta K_N * SEN_{N,z,I,J,K} \end{aligned} \quad (16)$$

where N is the total number of parameters. The specific discharge in the direction of flow is the magnitude of the specific discharge vector components and is given by:

$$|q_{I,J,K}| = \sqrt{q_{x,I,J,K}^2 + q_{y,I,J,K}^2 + q_{z,I,J,K}^2} \quad (17)$$

An additional sensitivity parameter relating the influence of recharge on specific discharge may also be considered in Latin hypercube (or Monte Carlo) simulations. This assumes a linear relationship between infiltration and specific discharge. When this option is used, the specific discharge is multiplied by a factor reflecting the ratio of sampled infiltration rate and the calibrated infiltration rate.

$$|q_{I,J,K}| = |q_{I,J,K}| \frac{\varepsilon}{\varepsilon} \quad (18)$$

Where:

$\frac{\varepsilon}{\varepsilon}$ is the ratio of the recharge value to the calibrated recharge value.

The recharge ratio was allowed to vary between 0.17 and 5.92, with 90 percent of the values between 0.4 and 2.6. This range is expected to encompass all recharge rates that have been suggested for the NTS region.

5.4 Diffusion Coefficient

The diffusion coefficient is an important parameter controlling the rate of mass transfer between the mobile and immobile regions. Limited site-specific data as well as values from the literature are available in the *Transport Parameter and Source Term Data Documentation Package* (IT, 1996e). Data available from NTS site measurements (Triay et al., 1993) have a range of values from 3.1×10^{-3} to 1.1×10^{-2} square meters per year (m^2/yr) (0.03 to 0.1 square feet per year [ft^2/yr]). These results are from diffusion experiments into fresh rock surfaces and probably overestimate values that might occur under natural conditions. Natural fracture surfaces have mineral coatings that may reduce the efficiency of diffusion into the rock matrix. Other published data noted in the *Transport Parameter and Source Term Data Documentation Package* (IT, 1996e) are somewhat lower with a range of 1.0×10^{-4} to $3.0 \times 10^{-3} \text{ m}^2/\text{yr}$. The range of values of 2.6×10^{-4} to $7.9 \times 10^{-3} \text{ m}^2/\text{yr}$ (0.003 to 0.1 ft^2/yr) used in the transport simulations is intended to represent a balance between NTS-specific data and published values from other sites. A log-normal distribution was chosen to describe the variation in the diffusion coefficient with a $\log_{10}$ mean of -2.84 and a $\log_{10}$ standard deviation of 0.24.

5.5 Fracture Spacing

The spacing of water conducting fractures has been estimated for both the carbonate aquifers and several of the volcanic units. Two recent reports, one for the carbonates (IT, 1996h) and one for the volcanics (IT, 1996i) summarize data on conducting fractures. The reports provide information on apparent fracture spacing and fracture dip angle along the borehole. The true fracture spacing is determined from the equation:

$$B = B' \sin(90 - \alpha) \quad (19)$$

Where:

- B = the true fracture spacing
- B' = apparent fracture spacing along the borehole
- α = the dip angle of the fracture where 0 is horizontal and 90 is vertical.

If $\alpha = 90$, then B is undefined because B' is indeterminate.

Using ranges for the spacing and the dip angle from the above cited reports, a range of values for the true fracture spacing was defined. For the carbonates (IT, 1996h), 268 m (880 ft) of core from Well ER-6-2 were examined. The number of open fractures per 1.52 m (5 ft) interval were recorded as was the average dip angle per 1.52-m (5-ft) interval. A total of 189 fractures were observed over the entire length of core. The average apparent fracture spacing is 1.42 m (4.7 ft) (268 m/189 fractures [880 ft/189 fractures]). A mean dip angle, weighted by the number of fractures, is 80.0 degrees. The true mean fracture spacing using the equation above is 0.22 m (0.7 ft). When individual intervals were examined the minimum and maximum true fracture spacings were 0.03 to 2.9 m (0.1 to 9.5 ft). Of the 105 intervals examined, only five had spacing greater than 1.0 m (3.3 ft). The largest spacing for simulation was chosen to be 1.5 m (4.9 ft) to avoid skewing the results too much toward large values. There were 20 intervals with spacing less than 0.07 m (0.2 ft), therefore the low end of the range was kept near the lower bound of observed values.

For the fractured volcanics (IT, 1996i), the range of fracture spacing was determined from summaries of fracture characteristics of the core from the seven holes that were examined. In IT, 1996i, the number of fractures per vertical foot was summarized per HSU. In addition, the number of fractures per dip angle interval of 0 to 30 degrees, 30 to 60 degrees, and 60 to 90 degrees. The mean true fracture spacing is calculated with the equation above, the mean

apparent fracture spacing, and the weighted mean fracture dip angle. The range of fracture spacing was obtained by letting the dip angle vary from 15 to 75 degrees while using the mean apparent spacing. Table 5-3 summarizes the results. The range of true fracture spacing for the TMA is from 0.9 to 9.8 m (3 to 315.1 ft). The majority of values (based on core length) are in the range of 2.6 to 9.8 m (8.5 to 32.2 ft). For the tuff cones, the range of 0.3 to 9.8 (0.1 to 32.2 ft) is very similar to that of the TMA. There are too few samples to define a mean spacing for the TBA or BAQ, so these will be assumed to be similar to the TMA. These ranges are all based on measured spacing of open fractures in core and from measured dip angles.

The upper end of the range of values, near 10 m (32.8 ft), is too large to be used in the one-dimensional transport model. Sudicky (1990) showed that actual one-dimensional diffusion into the matrix will be limited to a distance away from the fracture of much less than 10.0 m (32.8 ft). Using diffusion coefficient of 5×10^{-3} , Sudicky found that a spacing of 2.4 m (7.9 ft) was larger than the distance of diffusion over a 26-year time period. For the transport simulations, the fracture spacing for the volcanic units was limited to no greater than 2.5 m (8.0 ft). This differs from the observed data, but preserves a physical limitation of the matrix diffusion.

5.6 Initial Tritium Activities

The initial activities of tritium for individual shots are classified data. However, two averaged values have been calculated: one for Pahute Mesa (8.31×10^8 pCi/L) and one for all other shots (3.28×10^8 pCi/L). The averaged values were determined by summing the total tritium inventory for all the shots in each region and dividing by the sum of the estimated cavity volumes, assuming the cavities to be spherical and the porosity of the cavities to be 1.0.

For the transport modeling, it is assumed the tritium is evenly spread over the entire rock volume occupied by a sphere of radius equal to approximately two times the cavity radius. This assumption allows for a prompt injection zone around the actual cavity. Borg (1972) suggests that the zone of pervasive fracturing around an event is about 3 times the cavity radius in the horizontal direction and two times in the vertical direction. The assumption of two times will result in larger initial concentrations which are believed to be conservative. The volume of the initial region in the model is about eight times the cavity volume. To maintain a consistent representation of the total mass of tritium, the initial tritium activity for the simulations is calculated as:

$$C_{init} = C_o V_c / (V_{2c} \theta) \quad (20)$$

Table 5-3
Fracture Spacing for the Volcanic Aquifers

Hole Name	Core Length (m)	HSU	Apparent Fracture Spacing (m)	Minimum True Fracture Spacing (m)	Mean True Fracture Spacing (m)	Maximum True Fracture Spacing (m)	Number of Fractures 0 to 30 Degrees	Number of Fractures 30 to 60 Degrees	Number of Fractures 60 to 90 Degrees
UE-19x	295	TMA	10.2	2.6	3.9	9.8	49	10	3
	468	TC	3.4	0.9	1.3	3.3	49	10	3
UE-18t	514	TMA	10.2	2.6	5.8	9.8	2	0	1
UE-18r	36	TMA	2.5	NA	0.9	2.5	26	7	0
UE-20e#1	8	TBAQ	0.7	0.2	0.3	0.7	18	3	1
	39	TC	10.2	2.6	3.8	9.8	18	3	1
UE-20bh#1	19	TC	1.3	NA	0.4	1.3	14	2	0
U-20c	109	TC	1.0	0.3	0.5	1.0	40	14	4
UE-20f	54	TC	1.0	0.3	0.4	1.0	55	9	5
UE-20f	5	BAQ	0.5	0.1	0.2	0.5	55	9	5

Where:

- C_{init} = initial activity used in the simulation;
- C_o = averaged activity in a cavity assuming a porosity of 1.0;
- V_c = volume of a cavity;
- V_{2c} = volume of a sphere with radius of two times the cavity radius; and
- θ = sum of the mean fracture and matrix porosities.

The V_{2c} was adjusted slightly in some cases to allow for a radius measure of equal multiples of the model element spacing.

The initial concentration is placed in both the fractures and the matrix at the source. Thus, for the purposes of the modeling, the sum of the fracture and matrix porosities will represent the porosity of the initial contaminant volume. This assumption of initial tritium in the fractures and the matrix will be investigated further as one of the sensitivity runs where all the initial tritium is placed only in the fractures. This results in a much larger initial concentration. Table 5-4 contains the initial conditions for each of the events for the case of initial tritium in the matrix and the fracture and for tritium in the fracture only.

For the case of multiple events along the BOURBON pathline, the initial concentration from the MICKEY/TORRIDO event is assumed to be the same as BOURBON. For the TYBO pathline multiple source case, the initial condition at KASH is assumed to be the same as TYBO. This results in about a factor of two over estimate of the initial concentration at KASH. To account for this increase, the initial condition at event PEPATO was reduced by approximately one half.

A Neumann flux boundary condition is specified upgradient of the source as:

$$qC_m - D_m \frac{\partial C_m}{\partial X} = 0 \quad (21)$$

5.7 Correlation of Parameters

The LHS randomly selects parameter values for each realization. Typically, the selection of one parameter value is statistically independent of another. In some cases, however, it is necessary to correlate several parameters such that the value of one parameter is dependent on the value of another. An example of such a case is the relationship between fracture porosity and fracture spacing. Fracture porosity can be estimated as the fracture aperture divided by the fracture spacing. Therefore, as fracture spacing increases, the fracture porosity tends to decrease. In this case fracture porosity and fracture spacing are inversely correlated. Another example is fracture porosity and hydraulic conductivity. Porosity increases if aperture increases or spacing decreases. Both processes increase the cross-sectional area to flow and hydraulic conductivity also increases. In this case fracture porosity and hydraulic conductivity are positively correlated.

In the transport simulations the pairs of parameters considered correlated are given in Table 5-5. The correlation coefficient for each pair of parameters is 0.80 or -0.80. The 80 percent level is somewhat arbitrary but is based on a desire to have a relatively strong correlation among certain parameters. The importance of correlation was examined as part of sensitivity analyses that are presented later.

5.8 Other Constants

The half-life of tritium is well-known to be about 12.3 years. Using Equation 4, the decay coefficient, λ , is $5.635 \times 10^{-2} \text{ (yr}^{-1}\text{)}$. This value is assumed constant throughout the simulations.

Table 5-4
Initial Tritium Concentration Values and Ranges

Nuclear Detonation	Approximate Cavity Radius (m)	Total Porosity	Average Concentration (pCi/L)	Lower Bound Initial Concentration (pCi/L)	Mean Initial Concentration (pCi/L)	Upper Bound Initial Concentration (pCi/L)
BOURBON	69	0.084	3.28×10^8	5.0×10^6	3.7×10^8	7.0×10^9
TORRIDO/MICKEY*				5.0×10^6	3.7×10^8	7.0×10^9
KANKAKEE	73	0.084	3.28×10^8	7.0×10^6	4.6×10^8	8.0×10^9
HOUSTON	62	0.2815	8.30×10^8	1.0×10^7	2.1×10^8	2.0×10^9
TYBO	110	0.2915	8.30×10^8	2.0×10^7	4.7×10^8	5.0×10^9
PEPATO	60	0.2915	3.28×10^8	2.0×10^6	7.4×10^7	1.0×10^9
KASH**				2.0×10^7	4.7×10^8	5.0×10^9
All Tritium in the Fractures Initially						
BOURBON	69	0.004	3.28×10^8	3.0×10^8	8.6×10^9	1.0×10^{11}
HOUSTON	62	0.0015	8.30×10^8	2.0×10^9	3.4×10^{10}	3.0×10^{11}
TYBO	110	0.0015	8.30×10^8	7.0×10^9	9.3×10^{10}	7.0×10^{11}

*For TORRIDO/MICKEY the initial condition is the same as BOURBON.

**For KASH the initial condition is the same as TYBO.

Table 5-5
Correlation of Input Parameters

Parameter 1	Parameter 2	Correlation Coefficient
Effective porosity	Hydraulic conductivity	0.8
Effective porosity	Fracture spacing	-0.8
Matrix porosity	Diffusion coefficient	0.8
Hydraulic conductivity	Conductivity-depth coeff.	0.8

6.0 *Transport Calculations*

MC_TRANS was used to simulate tritium movement along the three selected pathways defined from the groundwater flow model. Some of those parameters are uncertain and are varied using a Latin hypercube technique. Receptor locations were specified for breakthrough curve calculations at selected distances downgradient from the sources. For all three paths, the first receptor location is within the source area.

The results presented in this chapter are organized with pathlines presented first, followed by the results of transport simulations, which includes sensitivity of the results to different parameters. The input data files to MC_TRANS for the three pathlines are presented in Appendix E.

6.1 *Pathlines*

The three pathlines chosen for this study began at separate shot locations: one each on Western Pahute Mesa (TYBO), Central Pahute Mesa (HOUSTON), and Yucca Flat (BOURBON). The pathlines for these three locations were presented in the plan view in Figure 4-6. [Figures 6-1, 6-2, and 6-3](#) show the vertical position of the pathline and the hydrostratigraphy where the top of each HSU is shown as a function of downgradient distance for shots BOURBON, HOUSTON, and TYBO, respectively.

The BOURBON pathline (Figure 6-1) begins at event KANKAKEE in the Lower Carbonate Aquifer and is dominated by the structural relationship of the LCA to other hydrostratigraphic units. The LCA, being the most permeable HSU, is the primary conduit for flow. The path closely follows the structural changes in the position of the LCA. In Frenchman Flat (down gradient distance of 40 km [24.9 mi]), where the top of the LCA is just above sea level, the pathline also goes near sea level. The pathline rises up and over clastic units that have been uplifted in the vicinity of the Specter Range at the northern edge of Amargosa Desert at a distance of about 70 km (43.5 mi). At 90 km (60 mi) the pathline enters the Alluvial Aquifer of Amargosa Desert and remains in that unit until it discharges in the vicinity of Ash Meadows. The pathline through event BOURBON appears to leave the LCA and enter into the overlying Volcanic Confining Unit at downgradient distances near 40 km (24.9 mi). This apparent skipping of the pathline in and out of a confining unit is artificial and stems from the inclusion of geologic model details at scales smaller than the flow model grids. These artificial conditions were corrected prior to transport modeling.

BOURBON

Figure 6-1
Geologic Model Cross Section Along the Pathline from BOURBON

HOUSTON

Figure 6-2
Geologic Model Cross Section Along the Pathline from HOUSTON

TYBO

Figure 6-3
Geologic Model Cross Section Along the Pathline from TYBO

Recall that the hydraulic conductivities of HSUs have been averaged within a flow model cell (weighted by the thicknesses). In the transport simulations, the particle locations are compared directly with the geologic model and are not averaged over the flow model cell. If two HSU are in the same flow model cell and the contrast in hydraulic conductivity is large, (for example, between an aquifer unit such as the LCA and a confining unit such as the VCU), then the transmissivity of the cell is dominated by the more permeable HSU. In the case of the BOURBON path, the flow path will remain in the LCA and not skip into the VCU, therefore LCA-specific data will be used along the path until very near the discharge location at Death Valley where the path enters the Alluvial Aquifer.

The pathline from event HOUSTON (Figure 6-2) begins in the Tuff Cones, then enters the Timber Mountain Aquifer at a downgradient distance of 6 km (3.7 mi). At 38.5 km (23.9 mi) downgradient, the path leaves the Timber Mountain Aquifer and enters the Tuff cones again for the next 4.1 km. After that the path enters the Volcanic Confining Unit for 11 km (6.8 mi). The remainder of the flow path is through a variety of units on its way to Death Valley.

The pathline through TYBO (Figure 6-3) moves downward from the point of origin at PEPATO to the discharge areas of Oasis Valley. The flow path begins in the Timber Mountain Aquifer and remains in that unit for more than 37 km (23 mi) until it discharges at Oasis Valley.

6.2 Tritium Transport Simulations - General

For each of the three pathlines, 200 Latin hypercube realizations were generated. The transport calculations are presented separately for each of the three paths beginning with BOURBON. The results of the transport simulations are presented in three different ways to summarize the Latin hypercube simulations. First, contour plots of tritium concentration, as a function of downgradient distance and time, are presented for different levels from 5 percent to 95 percent of the realizations. To create these plots, the tritium concentration data for all 200 realizations at each location and time are sorted from smallest to largest. The tritium values for which x% of the realizations were smaller were then contoured. Histograms of all tritium values for the 30-year period surrounding the peak mean tritium concentration and cumulative density plots of maximum tritium concentration at each receptor location are also presented. For the maximum tritium concentration plots, the maximum concentration at each downgradient receptor was recorded, sorted from smallest to largest, and plotted as a cumulative density function to demonstrate the range of possible outcomes. These latter figures were reserved for presentation of the variability in predicted concentration due to uncertainty in individual parameters.

6.3 Tritium Transport Simulations - BOURBON (Yucca Flat)

The primary method for presenting modeling results is to use contour plots of tritium concentration as a function of distance and time. For each of the 200 realizations, the model calculates tritium concentrations at fixed distances and times. Therefore at each distance and time, there are 200 values, each representing a different realization. These 200 values at each location in distance and time are sorted from smallest to largest. The position of a value in the sorted list represents the concentration for which a specified percentage of the realizations are smaller. For example, the concentration between the 100th and 101st value represents the 50 percent level and indicates that 50 percent of the realizations produced a larger concentration and 50 percent produced a smaller concentration. At the 95 percent level, 95 percent of the realizations produced a smaller concentration and 5 percent a larger concentration. By choosing the same level (for example 95 percent) for all points in distance and time, it is possible to contour tritium concentrations for which 95 percent of all realizations yielded a smaller value.

The 5, 50, 85, and 95-percent level contour plots for the pathline that originates at event KANKAKEE and passes through events BOURBON and MICKEY/TORRIDO are shown on [Figure 6-4](#). The 5 percent level ([Figure 6-4a](#)) represents the cases of small initial concentration and smaller groundwater movement. The two source areas are clearly seen on the figure; one for KANKAKEE and the other a combination of BOURBON and MICKEY/TORRIDO. The 50 percent level ([Figure 6-4b](#)) is the median case and represents the case where one half the realizations produced smaller values and one half produced larger. At the 50 percent level, the distance to the 20,000 pCi/L contour is approximately 20 km (12.4 mi) from KANKAKEE or 10.5 km (6.5 mi) downgradient from event BOURBON. At the larger levels, such as 95 percent ([Figure 6-4d](#)), the amount of downgradient movement as represented by the 20,000 pCi/L line is greater than at the 50 percent level, but less than 40 km (24.9 mi). For reference, the 36 km (22.9 mi) distance is a point at the northern edge of Frenchman Flat, well within the boundaries of the NTS. The 20,000 pCi/L contour reaches its maximum extent in approximately 25 years and then begins to recede gradually over the next 125 years.

As the tritium move downgradient in the fracture flow system, some of it diffuses in the matrix. As the main pulse of tritium moves farther downgradient, tritium in the matrix behind the pulse will begin to diffuse back in the fractures. When the mass flux from diffusion into the fractures is less than the mass loss due to radioactive decay, the boundary recedes.

A Contour Plot of Tritium Concentration (pCi/L) at 5% Leak

B Contour Plot of Tritium Concentration (pCi/L) at 5% Leak

C Contour Plot of Tritium Concentration (pCi/L) at 5% Leak

D Contour Plot of Tritium Concentration (pCi/L) at 5% Leak

Figure 6-4
Contour Plots of Tritium Concentration Along the Pathline Beginning at KANKAKEE

A second method of presenting the results of the 200 realizations is with histograms of tritium concentration for the 30 year periods surrounding the peak breakthrough. To support the human health risk assessment, the tritium concentration from all 200 trials was summarized in a non-continuous histogram for the 30-year period (the exposure time of the hypothetical land user) centered on the year when the maximum tritium concentration occurred in the contour plot at the 95 percent frequency level.

From [Figure 6-4](#) it is clear that the time of peak concentration at any downgradient location generally decreases as the level increases. This occurs because the higher level plots depict the more extreme cases with larger tritium velocity. As a result, the tritium migrates further downgradient in a shorter amount of time. Larger concentrations at any downgradient location occur at the 95 percent level than at the lower levels.

To be conservative in the analyses, the time of peak concentration used to choose the 30 year interval for the histogram analysis was defined on the basis of the 95 percent level plots.

However, the interval chosen is not -15 years to +15 years around the year of the maximum tritium concentration. The interval chosen is -5 years to +25 years around the year of the maximum concentration. This interval is chosen because the distribution of tritium concentration is not symmetrical, but skewed. A skewed distribution of contaminants in groundwater is typical in matrix diffusion dominated flow systems. Choosing the -5 to +25 years around the year of the maximum tritium concentration in groundwater will assure that the calculated dose, and risk to a potential does receptor is conservative and bounding. The concentration values from each realization over the 30-year period are used to create a non-continuous histogram of tritium concentrations for each dose location along the flow path.

The histograms are a presentation of the relative frequency of occurrence of tritium concentration in groundwater over the 30-year period surrounding the maximum tritium concentration. The histogram approximates the distribution of tritium activity at a location resulting from inclusion of parameter uncertainty in the groundwater transport model. The histogram lists the fraction of realizations within a specific tritium concentration range. Each tritium concentration range is known as a bin, with 25 bins used in the risk assessment. The smallest bin range is <100 pCi/L; the largest bin range is 4.60E+09 to 1.00E+10 pCi/L.

Figure 6-5 is the histogram of tritium concentration in the 30 year period around the peak for each downgradient distance. The values in Figure 6-5 are presented in Table 6-1. Each tritium concentration range represents approximately 1/3 of an order of magnitude, with the upper bound of the range presented on the axis. The distances increase from back to front in the plot such that the histogram for the 0.1 km (90.1 mi) distance is in the back of the figure. Yearly tritium concentration data over the 30 year period are summarized for all realizations in each histogram. Thus, at each distance, there are 6,200 values summarized; 31 yearly values (including the two end points of the 30 year period) times 200 realizations. The smallest interval in Figure 6-5 is the interval of 100 to 220 pCi/L. Any concentration less than 100 pCi/L is not plotted, but is included in the calculation of the histogram frequencies. The relative frequency represents the proportion of 6,200 values that fell within a particular tritium concentration range. The histogram defines the relative percentage of time that a particular range in tritium concentration will occur over the 30 year time period. This representation captures all the uncertainty in the Latin hypercube simulations and provides a balanced view of the true range of outcomes.

The results shown in the histograms are interesting from several aspects. Near the source, the tritium concentrations are large and have a narrower range of values than at larger distances. Well away from the source, nearly all the tritium concentration values fall below 100 pCi/L. For example, beyond 40 km (24.9 mi), more than 90 percent of the tritium values are less than 100 pCi/L. At intermediate distances, the influence of parameter uncertainty is evident by the wide range of tritium concentration values that occur.

The results presented, thus far, indicate that parameter uncertainty has a significant influence on the predicted transport of tritium. To investigate the role of individual parameters on the uncertainty, a series of sensitivity simulations was performed. These simulations took two forms; one in which only a single variable was allowed to vary and the other where all parameters varied, but a major component of the conceptual model was changed.

In the first case, selected individual parameters were allowed to vary within the Latin hypercube portion of MC_TRANS while all other parameters were set equal to their mean value. In Figure 6-6, the cumulative density of maximum tritium concentration was plotted for each parameter. This figure was developed by retaining the peak concentration that occurred at the 1 km (0.6 mi) downgradient distance for each of the 200 realizations. The 200 peak concentration values were sorted from smallest to largest, assigned a number (n) from 1 to 200, and the cumulative density function was calculated as $[n/(200+1)] \times 100$ to plot the density as a

KANKAKEE/BOURBON

Figure 6-5
Histogram of Tritium Concentration for the 30-Year Period Around the
Peak Concentration in the 95 Percent Level Plot

Table 6-1
Relative Frequency of Tritium Concentration versus Distance for the KANKAKEE/BOURBON Pathline
 (Page 1 of 2)

Concentration Interval (pCi/L)			less than	1.00E+02	2.20E+02	4.60E+02	1.00E+03	2.20E+03	4.60E+03	1.00E+04	2.20E+04	4.60E+04	1.00E+05	2.20E+05	4.60E+05
			1.00E+02	2.20E+02	4.60E+02	1.00E+03	2.20E+03	4.60E+03	1.00E+04	2.20E+04	4.60E+04	1.00E+05	2.20E+05	4.60E+05	1.00E+06
Distance (km)	Time Interval (yrs)														
98	220	250	1	0	0	0	0	0	0	0	0	0	0	0	0
95	220	250	1	0	0	0	0	0	0	0	0	0	0	0	0
90	220	250	1	0	0	0	0	0	0	0	0	0	0	0	0
80	136	166	0.9882	0.006	0.0019	0.0024	0.0015	0	0	0	0	0	0	0	0
75	85	115	0.97	0.005	0.0047	0.0055	0.0024	0.0052	0.0045	0.0027	0	0	0	0	0
70	55	85	0.9531	0.0076	0.0055	0.0008	0.0034	0.0102	0.0058	0.0032	0.005	0.0042	0.0013	0	0
65	50	80	0.95	0.0008	0.0047	0.0056	0.0047	0.0039	0.0082	0.0061	0.0056	0.0071	0.0032	0	0
60	47	77	0.9439	0.0061	0.001	0.009	0.0029	0.0039	0.009	0.0048	0.0071	0.0079	0.0044	0	0
55	43	73	0.9389	0.0061	0.0053	0.0015	0.0085	0.0044	0.0063	0.0081	0.0068	0.0061	0.0061	0.0019	0
50	39	69	0.9331	0.0042	0.0066	0.0063	0.0015	0.0097	0.0052	0.0092	0.0056	0.0074	0.0077	0.0035	0
45	28	58	0.9239	0.0058	0.0032	0.004	0.0065	0.0092	0.0069	0.0058	0.01	0.0073	0.0066	0.0071	0.0037
40	24	54	0.9124	0.0058	0.0073	0.006	0.0044	0.0069	0.0102	0.0066	0.0073	0.0115	0.0076	0.0068	0.0073
35	20	50	0.8842	0.004	0.0156	0.0089	0.0097	0.0079	0.0052	0.0123	0.0098	0.0082	0.0131	0.0082	0.0084
30	16	46	0.8171	0.0094	0.0194	0.01	0.0148	0.0066	0.014	0.0142	0.0192	0.0152	0.0181	0.0144	0.0127
25	14	44	0.6937	0.0197	0.0253	0.0227	0.0245	0.02	0.0226	0.0179	0.0213	0.0158	0.03	0.0405	0.0192
20	5	35	0.4823	0.0239	0.019	0.024	0.036	0.0487	0.0273	0.0368	0.0335	0.0482	0.0402	0.025	0.0435
17.5	6	36	0.2439	0.0229	0.0302	0.035	0.0273	0.0334	0.0377	0.0327	0.0416	0.0639	0.0742	0.0685	0.1023
15	0	30	0.084	0.0076	0.0097	0.0126	0.016	0.0177	0.021	0.0189	0.025	0.0358	0.0519	0.0682	0.1065
12.5	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0.0056
10	0	30	0.0326	0	0	0	0	0	0	0	0	0	0.0055	0.0105	0.0289
7.5	0	30	0.109	0.0103	0.014	0.0232	0.0231	0.0252	0.0329	0.0558	0.0355	0.0665	0.1102	0.1205	0.1353
5	0	30	0.1863	0.0144	0.0169	0.0182	0.0239	0.0229	0.0265	0.0332	0.0374	0.04	0.0644	0.0818	0.1119
4	0	30	0.1118	0.0097	0.0105	0.0126	0.019	0.0215	0.0245	0.0279	0.0315	0.0406	0.0497	0.0713	0.1098
3	0	30	0.0619	0.0058	0.0055	0.0068	0.0079	0.01	0.0121	0.0155	0.0239	0.0431	0.0406	0.0566	0.0965
2	0	30	0.0371	0.0006	0.0015	0.0016	0.0016	0.0026	0.0044	0.0055	0.009	0.0135	0.02	0.0458	0.0634
1	0	30	0.0326	0	0	0.0002	0	0.0002	0.0002	0	0.0015	0.0047	0.0077	0.014	0.0329
0.1	0	30	0	0	0	0	0	0.0002	0.0003	0.0005	0.0016	0.0068	0.0094	0.0152	0.0276

Table 6-1
Relative Frequency of Tritium Concentration versus Distance for the KANKAKEE/BOURBON Pathline
 (Page 2 of 2)

Concentration Interval (pCi/L)			1.00E+06	2.20E+06	4.60E+06	1.00E+07	2.20E+07	4.60E+07	1.00E+08	2.20E+08	4.60E+08	1.00E+09	2.20E+09	4.60E+09
			2.20E+06	4.60E+06	1.00E+07	2.20E+07	4.60E+07	1.00E+08	2.20E+08	4.60E+08	1.00E+09	2.20E+09	4.60E+09	1.00E+10
Distance (km)	Time Interval (yrs)													
98	220	250	0	0	0	0	0	0	0	0	0	0	0	0
95	220	250	0	0	0	0	0	0	0	0	0	0	0	0
90	220	250	0	0	0	0	0	0	0	0	0	0	0	0
80	136	166	0	0	0	0	0	0	0	0	0	0	0	0
75	85	115	0	0	0	0	0	0	0	0	0	0	0	0
70	55	85	0	0	0	0	0	0	0	0	0	0	0	0
65	50	80	0	0	0	0	0	0	0	0	0	0	0	0
60	47	77	0	0	0	0	0	0	0	0	0	0	0	0
55	43	73	0	0	0	0	0	0	0	0	0	0	0	0
50	39	69	0	0	0	0	0	0	0	0	0	0	0	0
45	28	58	0	0	0	0	0	0	0	0	0	0	0	0
40	24	54	0.0002	0	0	0	0	0	0	0	0	0	0	0
35	20	50	0.0045	0	0	0	0	0	0	0	0	0	0	0
30	16	46	0.0095	0.0055	0	0	0	0	0	0	0	0	0	0
25	14	44	0.0156	0.0066	0.004	0.0005	0	0	0	0	0	0	0	0
20	5	35	0.0689	0.0271	0.0097	0.005	0.001	0	0	0	0	0	0	0
17.5	6	36	0.0585	0.0694	0.0426	0.0142	0.0013	0.0005	0	0	0	0	0	0
15	0	30	0.1253	0.1747	0.1153	0.0684	0.0303	0.011	0.0002	0	0	0	0	0
12.5	0	30	0.0334	0.1077	0.1865	0.214	0.1898	0.1318	0.0776	0.0347	0.0148	0.0034	0.0005	0.0002
10	0	30	0.0681	0.1521	0.2289	0.2058	0.144	0.0855	0.0311	0.0065	0.0006	0	0	0
7.5	0	30	0.1255	0.0653	0.0258	0.0158	0.0052	0.001	0	0	0	0	0	0
5	0	30	0.1221	0.1179	0.0466	0.0158	0.0124	0.0071	0.0003	0	0	0	0	0
4	0	30	0.1456	0.1448	0.114	0.0298	0.0119	0.0113	0.0021	0	0	0	0	0
3	0	30	0.1352	0.1921	0.1437	0.1042	0.0216	0.0098	0.0073	0	0	0	0	0
2	0	30	0.1302	0.1689	0.2003	0.1677	0.0953	0.021	0.0068	0.0032	0	0	0	0
1	0	30	0.0661	0.1342	0.1989	0.1885	0.1766	0.1003	0.0353	0.0048	0.0013	0	0	0
0.1	0	30	0.0698	0.1163	0.1497	0.1779	0.1561	0.1245	0.079	0.0423	0.0176	0.0045	0.0006	0.0002

Bourbon - (distance = 1 km)

Figure 6-6
Cumulative Density Plot of Maximum Concentration at 1 km Downgradient of BOURBON

percentage. Each curve in [Figure 6-6](#) represents the peak concentration for each of 200 realizations when only a single parameter was varied. The source term, dispersivity, effective porosity, matrix porosity, block width, and diffusion coefficient were the transport parameters that were varied individually.

Only the most significant flow sensitivity parameters were chosen to be included in the sensitivity analyses. Of the 116 flow sensitivity parameters, the most sensitive were identified in [Table 5-5](#) of the *Groundwater Flow Model Documentation Package* (IT, 1996f). For the BOURBON path, the selected flow sensitivity parameter was K_h for the Volcanics Undifferentiated. The final variable to be included in the sensitivity analyses is the recharge coefficient.

The results of the sensitivity runs are presented in [Figures 6-6, 6-7, and 6-8](#) for the 1-km, 10-km, and 20-km (0.6-mi, 6.2-mi, and 12.4-mi) distances. To simplify the interpretation, these results were obtained from transport simulations from event BOURBON alone. The source term dominates the uncertainty near the event. Further downgradient, the most important parameters are matrix diffusion related (matrix porosity, block width, and diffusion coefficient) and the recharge multiplier. These results indicate that the uncertainty in the total amount of water flowing through Yucca Flat and the diffusion properties of the Lower Carbonate Aquifer are important.

Additional considerations in the interpretation of the results revolve around several key aspects of the conceptual model. Three aspects were investigated: (1) the importance of multiple events along the same pathline, (2) the influence of parameter correlation on the results, and (3) the resulting concentrations if all the initial tritium were in the fractures only.

The impact of multiple sources was investigated by simulating the transport from the BOURBON event alone and comparing the results to the previous simulations with multiple events. [Figure 6-9a](#) is the contour plot of tritium concentration from the BOURBON event alone at the 95 percent level. The maximum extent of the 20,000 pCi/L contour is about 22 km (13.7 mi). Recall that the BOURBON event is 9.5 km (5.9 mi) along the path given in [Figure 6-4d](#). If that distance is added to 22 km (13.7 mi), the distance is 31.5 km (19.6 mi). This is about 4.5 km (2.8 mi) less travel distance than in the multiple source case. Part of the difference is made up by the third source placed 2.6 km (1.6 mi) downgradient of BOURBON in

Bourbon - (distance = 10 km)

Figure 6-7
Cumulative Density Plot of Maximum Concentration at 10 km (6.2 mi) Downgradient of BOURBON

Bourbon - (distance = 20 km)

Figure 6-8
Cumulative Density Plot of Maximum Concentration at 20 km (12.4 mi) Downgradient of BOURBON

A. Base Case - Tritium in Matrix and Fractures

B. No Correlation of Parameters - Tritium in Matrix and Fractures

C. Tritium Source in Fractures Only

Note: Contours of Tritium Concentration are pCi/L

Figure 6-9
Contour Plots of Tritium Conception Along the Pathline Beginning at BOURBON Conceptual Model Variation

the original simulation (Figure 6-4d). This would suggest that the additive effects multiple tritium sources may not be a significant factor in the estimation of concentrations downgradient in the carbonate.

Not all parameters were allowed to vary independently of the other parameters. This second case examined a scenario where all correlation among parameters was zero, thus allowing all parameters to vary independently. At the 95 percent level (Figure 6-9b), this result produces a maximum distance of about 17 km (10.6 mi), 5 km (3.1 mi) less than when correlation was included (Figure 6-9a). This suggests that an increase in one parameter is offset by a decrease in another when parameters are uncorrelated. This reduction is significant and suggests that correlation of parameters cannot be ignored.

The final simulation examined the assumption that the initial tritium mass was distributed between the fracture and matrix porosity. The matrix porosity is generally much larger than the fracture porosity and serves as a storage reservoir for tritium. If all the initial tritium mass were to be in the fractures alone, the concentration (mass/volume) would be much greater. To account for the tritium mass in the fractures alone, the range of initial tritium concentration was set to 3.0×10^8 to 1.0×10^{11} pCi/L. The large values were required to maintain the same initial tritium mass as in the base case. Figure 6-9c is the 95 percent level for the case of tritium in fractures alone. The maximum downgradient extent is 25 km (15.5 mi), about 3 km (1.9 mi) larger than the base case. Additionally the 20,000 pCi/L line persists over approximately a 50 year period at that distance whereas in the base case, the distance to the 20,000 pCi/L concentration decreases shortly after reaching its maximum extent.

6.4 Tritium Transport Simulations - HOUSTON (Central Pahute Mesa)

Event HOUSTON is located on Central Pahute Mesa and was chosen to represent expected migration of tritium from many of the nearby shots that follow a path around the east side of Timber Mountain (Figure 4-6). Recall from Figure 6-2 that the pathline from the HOUSTON event began in the tuff cones (TC), then in succession entered the welded tuff aquifer (TMA), the TC again, and then into the Volcanic Confining Unit (VCU). Receptor locations were defined at distances of 0.1 to 10, 15, 20, 25, 30, 35, 40, 45, and 50 km (31.1 mi) downgradient. (0.06, 0.6 to 6.2, 9.3, 12.4, 15.5, 18.6, 21.7, 24.9, 28, and 31.1 mi).

The contours of the tritium concentration as a function of distance and time are given in Figure 6-10, for the 5, 50, 85, and 95 percent levels, respectively. Again the distance

A. Contours of Tritium Concentration (pCi/L) at the 5% Level

B. Contours of Tritium Concentration (pCi/L) at the 50% Level

C. Contours of Tritium Concentration (pCi/L) at the 85% Level

D. Contours of Tritium Concentration (pCi/L) at the 95% Level

Figure 6-10
Contour Plots of Tritium Concentration Along the Pathline Beginning at HOUSTON

downgradient of the 20,000 pCi/L contour increases and the time of peak decreases as the percentage level increases. At the 95 percent level, the rapid transport in the fractured TMA aquifer has tritium at 20,000 pCi/L, 42 km (26.1 mi) downgradient in less than 25 years. At approximately 45 km (28 mi), the pathline enters the VCU and slows down substantially. This reduction in velocity, caused by the larger effective porosity of the confining units, allows radioactive decay to completely remove the tritium before much transport takes place.

The histogram of tritium concentration along the pathline from event HOUSTON is presented in [Figure 6-11](#). As before, these data represent all the concentration data for a 30-year period which is determined from the 95-percent level tritium contour plot in [Figure 6-10](#). [Table 6-2](#) contains all the data that are plotted in [Figure 6-11](#). The relative proportion of tritium concentration values that are above 20,000 pCi/L decreases substantially as the distance from the event increases.

The sensitivity of the results to variability in different parameters is presented in [Figures 6-12](#), [6-13](#), and [6-14](#) for distances of 1, 5, and 20 km (0.6, 3.1, and 12.4 mi), respectively. The parameters that were varied are source term concentration, dispersivity, effective porosity, matrix porosity, block width, diffusion, recharge coefficient, four flow sensitivity parameters, the hydraulic conductivity Zone 1 of the TC and TMA, and the coefficient describing the decrease of conductivity with depth for Zone 1 of the TC and TMA. Near the source the most important parameter is the source term concentration. Further downgradient, at 5 km (3.1 mi), the importance of the source term concentration is similar to the diffusion parameters (matrix porosity, block width, and diffusion coefficient) and the recharge coefficient. At the 20 km (12.4 mi) distance, the variability in the diffusion parameters and the recharge dominate the uncertainty in predicted values.

[Figure 6-15](#) is a contour plot of tritium concentration for the case where the initial mass of tritium was placed in the fractures only. This result is only slightly different from the base case in [Figure 6-10d](#) in terms of maximum extent. The primary differences occur at early time near the source where concentrations are much larger than in the base case.

For the pathline from Central Pahute Mesa, the most important aspect is that the paths do not discharge into Oasis Valley, but rather pass through the VCU which substantially slows the rate of migration and allows radioactive decay to remove the tritium.

Figure 6-11

Histogram of Tritium Concentration for the 30-Year Period Around the Maximum Concentration in the 95 Percent Level Contour Plot

Table 6-2
Relative Frequency of Tritium Concentration versus Distance for the HOUSTON Pathline
 (Page 2 of 2)

Concentration Interval (pCi/L)			1.00E+06	2.20E+06	4.60E+06	1.00E+07	2.20E+07	4.60E+07	1.00E+08	2.20E+08	4.60E+08	1.00E+09	2.20E+09	4.60E+09
			2.20E+06	4.60E+06	1.00E+07	2.20E+07	4.60E+07	1.00E+08	2.20E+08	4.60E+08	1.00E+09	2.20E+09	4.60E+09	1.00E+10
Distance (km)	Time Interval (yrs)													
50	220	250	0	0	0	0	0	0	0	0	0	0	0	0
45	188	218	0	0	0	0	0	0	0	0	0	0	0	0
40	16	46	0	0	0	0	0	0	0	0	0	0	0	0
35	13	43	0.0026	0	0	0	0	0	0	0	0	0	0	0
30	10	40	0.0055	0.0016	0	0	0	0	0	0	0	0	0	0
25	5	35	0.0203	0.0071	0.0018	0	0	0	0	0	0	0	0	0
20	4	34	0.024	0.0173	0.0052	0.001	0	0	0	0	0	0	0	0
15	1	31	0.0494	0.0321	0.0111	0.0056	0.0003	0	0	0	0	0	0	0
10	1	31	0.0852	0.0521	0.0302	0.0119	0.0016	0	0	0	0	0	0	0
9	0	30	0.09	0.0558	0.0327	0.0148	0.0032	0	0	0	0	0	0	0
8	0	30	0.0989	0.065	0.0395	0.0194	0.0058	0.0002	0	0	0	0	0	0
7	0	30	0.1234	0.0776	0.0437	0.0263	0.0081	0.0006	0	0	0	0	0	0
6	0	30	0.1369	0.1039	0.054	0.0303	0.0118	0.0015	0.0003	0	0	0	0	0
5	0	30	0.1477	0.13	0.0716	0.0369	0.0165	0.0024	0.0005	0	0	0	0	0
4	0	30	0.1576	0.1558	0.099	0.0494	0.0226	0.0037	0.0005	0	0	0	0	0
3	0	30	0.139	0.1835	0.1371	0.0689	0.0327	0.0082	0.0006	0.0002	0	0	0	0
2	0	30	0.1731	0.1766	0.1865	0.1176	0.0495	0.0171	0.0023	0.0003	0	0	0	0
1	0	30	0.0856	0.1563	0.2242	0.2219	0.1573	0.0585	0.016	0.004	0.0002	0	0	0
0.1	0	30	0.0392	0.0837	0.1358	0.2073	0.2166	0.1561	0.0866	0.046	0.0135	0.0021	0.0003	0

Houston - (distance = 1 km)

Figure 6-12
Cumulative Density Plot of Maximum Tritium Concentration at a Distance of 1 km
Downgradient of HOUSTON

Houston - (distance = 5 km)

Figure 6-13
Cumulative Density Plot of Maximum Tritium Concentration at a Distance of 5 km
Downgradient of HOUSTON

Houston - (distance = 20 km)

Figure 6-14
Cumulative Density Plot of the Maximum Tritium Concentration at
a Distance of 20 km Downgradient of HOUSTON

Figure 6-15
Contour Plots of Tritium Concentration at the 95% Level - Initial Tritium in Fractures Only

A. Contours of Tritium Concentration (pCi/L) at the 5% Level

B. Contours of Tritium Concentration (pCi/L) at the 50% Level

C. Contours of Tritium Concentration (pCi/L) at the 85% Level

D. Contours of Tritium Concentration (pCi/L) at the 95% Level

Figure 6-16
Contour Plots of Tritium Concentration Along the Pathline Beginning at Event PEPATO

6.5 Tritium Transport Simulations - TYBO (Western Pahute Mesa)

The TYBO event pathline begins at the upgradient event PEPATO, passes through event KASH, and then through TYBO at a distance of 7.8 km (4.8 mi). The tritium contour plots for the 5, 50, 85, and 95 percent levels are given in [Figure 6-16](#). The pathway from TYBO is entirely in the TMA and as a result rapid fracture flow and transport dominates the predicted concentrations. The discharge area of Oasis Valley is located at about 37 km (23 mi) on these figures. At the 50 percent level, the distance to the 20,000 pCi/L contours is approximately 30 km (18.6 mi) ([Figure 6-16b](#)). The tritium concentrations at Oasis Valley do not exceed 20,000 pCi/L. For the larger percentage cases (85, 95 percent) tritium concentrations discharging at Oasis Valley are more than 10^6 pCi/L for periods of time spanning as many as 40 years. At the 95 percent level ([Figure 6-16d](#)), the tritium concentration at Oasis Valley exceeds 20,000 pCi/L beginning with just a few years and lasting until 115 years.

The histogram of tritium values that fall within the 30 year period as defined by the peak concentrations in the 95 percent contour plot are given in [Figure 6-17](#). The data in [Figure 6-17](#) are summarized in [Table 6-3](#). The range of tritium concentrations at the downgradient discharge point (Oasis Valley at 37.1 km [23.1 mi]) is quite large and encompasses values larger than 2.2×10^7 pCi/L. From [Table 6-3](#) it can be seen that more than 27.7 percent of the values in the histogram at Oasis Valley are above 20,000 pCi/L.

The sensitivity results for distances of 1, 10, and 30 km (0.6, 6.2, and 18.6 mi) are presented in [Figures 6-18](#), [6-19](#), and [6-20](#). As before, the sensitivity results are present for the pathline beginning at event TYBO alone. Source term concentration dominates the uncertainty at 1 km (0.6 mi), but its impact is similar to matrix diffusion parameters and the recharge coefficient at 10 km (6.2 mi). By 30 km (18.6 mi), the recharge coefficient and matrix diffusion parameters are more important than the source term.

Two other simulations to address the impact of multiple sources and initial tritium mass in the fractures is shown in [Figure 6-21](#). [Figure 6-21a](#) is a contour plot of tritium concentration for the case transport from TYBO alone with no other upgradient event. The shorter pathline means that the Oasis Valley discharge area is located about 30 km (18.6 mi) downgradient. Tritium concentrations for the 95 percent level are nearly as large and remain above 20,000 pCi/L for approximately 100 years. Thus, the comparison of the one-source versus the multiple-source case suggests that the multiple sources are not the controlling factor in the large tritium values

Tybo - (distance = 1 km)

Figure 6-18
Cumulative Density Plot of Maximum Tritium Concentration at 1 km
Downgradient of Event TYBO

Table 6-3
Relative Frequency of Tritium Concentration versus Distance for the PEPATO/TYBO Pathline
 (Page 1 of 2)

Concentration Interval (pCi/L)			less than 1.00E+02	1.00E+02 2.20E+02	2.20E+02 4.60E+02	4.60E+02 1.00E+03	1.00E+03 2.20E+03	2.20E+03 4.60E+03	4.60E+03 1.00E+04	1.00E+04 2.20E+04	2.20E+04 4.60E+04	4.60E+04 1.00E+05	1.00E+05 2.20E+05	2.20E+05 4.60E+05	4.60E+05 1.00E+06
Distance (km)	Time Interval (yrs)														
37.1	8	38	0.4008	0.0116	0.0224	0.03	0.0244	0.0263	0.0326	0.0306	0.0329	0.0552	0.056	0.0647	0.0719
34.3	8	38	0.3761	0.021	0.0158	0.0274	0.03	0.0234	0.0353	0.0305	0.0337	0.051	0.0624	0.0685	0.0753
31.8	4	34	0.4092	0.0119	0.0227	0.0182	0.0319	0.0252	0.0281	0.0311	0.0332	0.0371	0.0608	0.0616	0.0671
29.4	3	33	0.4042	0.0187	0.0115	0.0247	0.0242	0.0313	0.0253	0.0327	0.0345	0.0334	0.0556	0.064	0.0694
24.5	1	31	0.3792	0.0071	0.0229	0.0132	0.0165	0.0216	0.0335	0.0361	0.0347	0.0373	0.0445	0.0584	0.0837
19.6	1	31	0.2029	0.0235	0.02	0.0261	0.026	0.0203	0.0184	0.0263	0.036	0.0447	0.0432	0.065	0.0945
17.1	2	32	0.1027	0.0097	0.0181	0.0226	0.0313	0.0202	0.0415	0.0276	0.0216	0.0431	0.0502	0.0644	0.0974
14.7	0	30	0.0974	0.0048	0.0076	0.0124	0.0073	0.0115	0.0318	0.0356	0.0329	0.0456	0.0352	0.0566	0.0811
12.2	0	30	0.0569	0.001	0.0019	0.0044	0.0035	0.004	0.0063	0.0089	0.0148	0.0181	0.0442	0.0626	0.0792
9.8	0	30	0.0331	0	0.0003	0.0006	0.004	0	0	0	0.0002	0.0016	0.004	0.0068	0.0245
7.3	0	30	0.0323	0	0	0	0	0	0	0	0	0	0.0037	0.0174	0.0392
4.9	0	30	0.0384	0	0.0003	0.0003	0.0005	0.0003	0.0002	0.0002	0.0011	0.004	0.01	0.0235	0.0579
3.9	0	30	0.0327	0.0003	0.0002	0.0045	0.0002	0	0.0002	0	0.0002	0.0005	0.0008	0.0134	0.0515
2.9	0	30	0.0323	0	0	0	0	0.0002	0	0	0.0026	0.0021	0.0005	0.0071	0.034
2	0	30	0.0323	0	0	0	0	0	0	0	0	0	0.0002	0.0068	0.0273
1	0	30	0	0	0	0	0	0	0	0	0	0.0005	0.0021	0.014	0.0402
0.1	0	30	0	0	0	0	0	0	0.0021	0.0023	0.0023	0.0124	0.0432	0.0765	0.1287

Table 6-3
Relative Frequency of Tritium Concentration versus Distance for the PEPATO/TYBO Pathline
 (Page 2 of 2)

Concentration Interval (pCi/L)			1.00E+06	2.20E+06	4.60E+06	1.00E+07	2.20E+07	4.60E+07	1.00E+08	2.20E+08	4.60E+08	1.00E+09	2.20E+09	4.60E+09	1.00E+10
			2.20E+06	4.60E+06	1.00E+07	2.20E+07	4.60E+07	1.00E+08	2.20E+08	4.60E+08	1.00E+09	2.20E+09	4.60E+09	1.00E+10	2.20E+10
Distance (km)	Time Interval (yrs)														
37.1	8	38	0.0653	0.0452	0.0237	0.0061	0.0003	0	0	0	0	0	0	0	0
34.3	8	38	0.0676	0.0481	0.0266	0.0069	0.0003	0	0	0	0	0	0	0	0
31.8	4	34	0.0681	0.0495	0.0313	0.011	0.0019	0	0	0	0	0	0	0	0
29.4	3	33	0.0687	0.0524	0.0344	0.0118	0.0027	0.0003	0.0002	0	0	0	0	0	0
24.5	1	31	0.0815	0.0642	0.0411	0.0181	0.006	0.0003	0	0.0002	0	0	0	0	0
19.6	1	31	0.1347	0.1061	0.0644	0.034	0.0121	0.0015	0.0003	0	0	0	0	0	0
17.1	2	32	0.1437	0.1594	0.0823	0.0426	0.0189	0.0029	0.0002	0	0	0	0	0	0
14.7	0	30	0.1327	0.1816	0.1302	0.0624	0.0256	0.0073	0.0002	0.0002	0	0	0	0	0
12.2	0	30	0.1179	0.2015	0.1968	0.1165	0.0452	0.0148	0.0015	0.0002	0	0	0	0	0
9.8	0	30	0.0669	0.1635	0.2497	0.2269	0.1492	0.0568	0.0115	0.0003	0	0	0	0	0
7.3	0	30	0.0894	0.1898	0.2348	0.2058	0.1313	0.0471	0.0084	0.0008	0	0	0	0	0
4.9	0	30	0.1371	0.229	0.2221	0.1647	0.0805	0.0265	0.0031	0.0003	0	0	0	0	0
3.9	0	30	0.1194	0.2031	0.2216	0.1869	0.119	0.0387	0.0065	0.0005	0	0	0	0	0
2.9	0	30	0.0968	0.1813	0.2127	0.1965	0.1495	0.0669	0.016	0.0015	0.0002	0	0	0	0
2	0	30	0.0768	0.1565	0.1879	0.1892	0.1697	0.1042	0.0387	0.0079	0.0021	0.0006	0	0	0
1	0	30	0.1002	0.1555	0.1658	0.1832	0.1658	0.0971	0.0366	0.0221	0.0123	0.004	0.0005	0.0002	0
0.1	0	30	0.1703	0.1771	0.1713	0.1206	0.0552	0.0247	0.0092	0.0035	0.0006	0	0	0	0

Tybo - (distance = 10 km)

Figure 6-19

Cumulative Density Plot of Maximum Tritium Concentration at a Distance of 10 km Downgradient of Event TYBO

Tybo - (distance = 30 km)

Figure 6-20
Cumulative Density Plot of Maximum Tritium Concentration at a Distance of 30 km
Downgradient of Event TYBO

Figure 6-21
Contour Plots of Tritium Concentration at the 95 Percent Level for Simulations from Event TYBO Alone

predicted at Oasis Valley for the 95 percent level case. The final example is of all the initial tritium mass in fractures case. This result, [Figure 6-21b](#), yields tritium concentrations in excess of 1×10^7 pCi/L at the discharge boundary. Changes in the conceptual model of the source are important considerations in the predictions.

The two simulations for the TYBO only event produce large tritium concentrations at the Oasis Valley groundwater discharge area at 95 percent levels. These results must be viewed in relation to the Latin hypercube methodology that was used. Many of the parameters that govern the transport of tritium are uncertain. The parameter ranges that were used in this study were intentionally quite large to ensure that the actual tritium transport does not fall outside of the range of predicted concentrations. This means that the 5 percent level results should predict less impact than reality and the 95 percent level should predict greater impact than reality.

An additional simulation was performed to see if the most recent version of the groundwater flow model was the source of the large tritium concentration values. In this last simulation, the specific discharge from the flow model was replaced by a value similar to an earlier version of the flow model. This resulted in approximately a factor of 6 reduction in the discharge as it was included in the transport model. The results of this run, at the 95 percent level, are presented in [Figure 6-22](#). The tritium at Oasis Valley is just slightly above 20,000 pCi/L, which is about a factor of 100 less than the current case. The increased recharge and other changes to the groundwater flow model have resulted in the large increase in tritium values at the discharge area.

The results from the PEPATO/KASH/TYBO simulations are an example. At the 95 percent level, the predicted tritium concentrations at Oasis Valley should have exceeded 1×10^6 pCi/L after about two years and remained above 1×10^6 pCi/L for nearly 40 years. The time of travel in the simulations begins when the tritium begins to migrate away from an event. There is a lag time between when an event occurs and the time that groundwater fills the cavity and begins to carry tritium downgradient. This time lag is unknown, but is expected to be less than the 21 years since the TYBO event. If the 95 percent level contours were reality, the concentration of tritium in Oasis Valley should have reached 1×10^6 pCi/L by now. To date, no tritium concentrations above environmental levels (approximately 50 pCi/L) have been detected at any monitoring site in Oasis Valley (DOE, 1995). This suggests that the 85 and 95 percent results for Oasis Valley likely overestimate concentrations that would occur in reality.

Figure 6-22
Contour Plots of Tritium Concentration Along the TYBO Pathline for the Case of
Smaller Specific Discharge

Another consideration in the interpretation of these results are the limitations of the one-dimensional modeling approach. In the real groundwater flow system, dispersion causes a plume to disperse in three dimensions. The one-dimensional approach allows dispersion in one direction only and therefore limits the amount of dilution that can occur. The amount of additional dilution from a three-dimensional approach is estimated using the analytical Model AT123D.

The effect of three-dimensional dispersion was examined with the computer code AT123D. AT123D is a program of analytical solutions for one-, two-, and three-dimensional transport of heat, dissolved chemical or dissolved radionuclides in a homogeneous aquifer subject to a uniform, stationary regional flow field (Yeh, 1981). The code, based on Green's Functions (Huyakorn and Pinder, 1983), combines the product of solution of point and /or integrated line sources in each of the three principle directions to allow for solution of advective-dispersive transport from a point, planar, or rectangular source. The modeled aquifer is infinite in the direction of flow, but can be considered to be finite or infinite in the vertical and transverse horizontal directions. The source may be instantaneous, steady (finite or infinite in length) or variable over time.

Two cases were examined with AT123D. In the first case, a three-dimensional cubic source was placed in an aquifer that resembled the hydrogeology along the flow path simulated with MC_TRANS. The source was allowed to disperse in three dimensions as it migrated downgradient. A second case was constructed with the vertical and lateral boundaries of the flow system corresponding to the source dimension. This effectively prevents lateral and vertical dispersion and produces a one-dimensional geometry similar to the MC_TRANS simulations. For each case, dilution factors were calculated at selected downgradient distances. The dilution factor is the input concentration value divided by the maximum concentration. The dilution factor results are independent of the initial concentration.

The effect of three-dimensional dispersion is calculated as the ratio of the dilution factor from the three-dimensional case to the dilution factor from the one-dimensional case. In this way, the results can be presented in the form of the additional dilution one could expect if three-dimensional dispersion were considered.

Table 6-4 contains the input parameters used in the AT123D simulations. Included in Table 6-4 are the path name, source dimension, depth of the center of the source below the top of the aquifer, and the aquifer thickness. The results, in terms of the amount of additional dilution due to three-dimensional dispersion are given in Table 6-5 for two sets of dispersion coefficient values. The additional dilution is a factor that ranges between 5 and 200 depending on the pathline, the distance downgradient, and the dispersion coefficient.

Table 6-4
Input Parameters for the AT123D Simulations

Path Name	Source Dimension (length of each side of the cube) (m)	Depth of Center of the Source Below the Top of the Aquifer (m)	Aquifer Thickness (m)
BOURBON	175	400	1,200
HOUSTON	175	340	1,000
TYBO	225	300	1,000

Table 6-5
Additional Dilution Due to Three-Dimensional Dispersion

Path Name	Distance (m)	Additional Dilution (small dispersion)¹	Additional Dilution (large dispersion)¹
BOURBON	20,000	15	81
	30,000	22	111
	60,000	39	179
	80,000	47	211
HOUSTON	20,000	15	73
	40,000	27	121
	50,000	31	137
TYBO	10,000	5	27
	20,000	9	43
	30,000	13	60
	38,000	15	70

¹ Small dispersion: $\alpha_L = 100$ m; $\alpha_T = 20$ m; $\alpha_V = 1$ m.
Large dispersion: $\alpha_L = 1000$ m; $\alpha_T = 200$ m; $\alpha_V = 10$ m.

The additional dilution is on the order of a factor of 5 to 200. An additional source of dilution occurs at the spring discharge locations. It is not clear how much dilution may occur at the springs themselves due to different sources of water converging at the discharge locations. Subsequent simulations to be performed as part of later CAU scale investigations will be designed to address questions of three dimensional dilution.

7.0 Summary of Results

One-dimensional tritium transport simulations were performed along three flow paths originating or passing through from underground test locations: BOURBON, on Yucca Flat; HOUSTON, on Central Pahute Mesa; and TYBO, on Western Pahute Mesa. Several observations can be made based on these results:

- The regional geology, as depicted in the geologic model, is the dominant factor controlling the horizontal and vertical position of paths.
- At many downgradient receptor locations, the range of maximum tritium activity was quite large, often exceeding 5 orders of magnitude.
- Matrix diffusion is an important mechanism governing the migration of tritium in fractured volcanic rocks.
- The recharge coefficient, which accounts for the total groundwater flux uncertainty, is as important as matrix diffusion at downgradient locations.
- Source term uncertainty is most important near the event cavities and decreases in importance as travel distance increases.
- The downgradient distance, beyond which the tritium concentration does not exceed 20,000 pCi/L at the 95-percent level, is summarized in Table 7-1.
- The results presented at the 95-percent level are expected to be an overestimate of what will occur in reality.

Table 7-1
Approximate Distance Beyond Which Tritium Concentration
Does Not Exceed 20,000 pCi/L at the 95-Percent Level

Event	Location	Maximum Downgradient Extent: 95 Percent of all Realizations
BOURBON	Yucca Flat	36 km
HOUSTON	Central Pahute Mesa	42 km
TYBO	Western Pahute Mesa	37 km (Oasis Valley)

8.0 References

- Bear, J. 1972. *Dynamics of Fluids in Porous Media*. New York, NY: American Elsevier.
- Blaylock, B.G., M.L. Frank, and B.R. O' Neal. 1993. *Methodology for Estimating Radiation Dose Rates to Freshwater Biota Exposed to Radionuclides in the Environment*, ES/ER/TM-78. Oak Ridge, TN: Oak Ridge National Laboratory.
- Borg, T.Y. 1972. *Extent of Prevasive Fracturing Around Underground Nuclear Explosions*, UCRL-73831. Livermore, CA: Lawrence Livermore Laboratory.
- Burbey, T.J., and S.W. Wheatcraft. 1986. *Tritium and Chlorine 36 Migration from a Nuclear Explosion Cavity*, Desert Research Institute, Publication 45050. Las Vegas, NV.
- Coats, K.H., and D.B. Smith. 1964. "Dead-end pore volume and dispersion on porous media." In *Society of Petroleum Engineers Journal*, 4: 73-84.
- Davenport, J.M., and R.L. Iman. 1982. An iterative algorithm to produce definite correlation matrix from an "approximate correlation matrix" (with a program user's guide). SAND81-1376.
- de Marsily, G. 1986. *Quantitative Hydrogeology*. Orlando, Florida: Academic Press.
- DOE, see U.S. Department of Energy.
- Duncan, D. U.S. Department of Energy. 1993. Personal Communication. Las Vegas, NV.
- FFACO, see *Federal Facility Agreement and Consent Order of 1996*.
- Federal Facility Agreement and Consent Order of 1996*. Agreement with State of Nevada Division of Environmental Protection, U.S. Department of Energy, and U.S. Department of Defense.
- Freeze, R.A., and J.A. Cherry. 1979. *Groundwater*. Englewood Cliffs, NJ: Prentice-Hall.
- Gelhar, L.W., C. Welty, and K.R. Rehfeldt. 1992. "A Critical Review of Data on Field-Scale Dispersion in Aquifers." In *Water Resources Research*, Vol. 28, No. 7: 1955-1974.
- Grisak, G.E., and J.F. Pickens. 1980. "Solute Transport Through Fractured Media 1: The Effect of Matrix Diffusion." In *Water Resources Research*, 16(4): 710-730.
- Harrison, B., E.A. Sudicky, and J.A. Cherry. 1992. "Numerical Analysis of Solute Migration Through Fractured Clayey Deposits Into Underlying Aquifers." In *Water Resources Research*, 28(2): 515-526.

- Huyakorn, P.S., and G.F. Pinder. 1983. *Computational Methods in Subsurface Flow*. Orlando, FL: Academic Press.
- IT Corporation. 1996a. *Geologic Model Documentation Package (Phase I Data Analysis Documentation, Volume I)*. Prepared for DOE/NV. Las Vegas, NV.
- IT Corporation. 1996b. *Potentiometric Data Documentation Package (Phase I Data Analysis Documentation, Volume II)*. Prepared for DOE/NV. Las Vegas, NV.
- IT Corporation. 1996c. *Groundwater Recharge and Discharge Data Documentation Package (Phase I Data Analysis Documentation, Volume III)*. Prepared for DOE/NV. Las Vegas, NV.
- IT Corporation. 1996d. *Hydrologic Parameter Data Documentation (Phase I Data Analysis Documentation, Volume IV)*. Prepared for DOE/NV. Las Vegas, NV.
- IT Corporation. 1996e. *Transport Parameter and Source Term Data Documentation Package (Phase I Data Analysis Documentation, Volume V)*. Prepared for DOE/NV. Las Vegas, NV.
- IT Corporation. 1996f. *Groundwater Flow Model Documentation Package (Phase I Data Analysis Documentation, Volume VI)*. Prepared for DOE/NV. Las Vegas, NV.
- IT Corporation. 1996g. *Risk Assessment Documentation Package (Phase I Data Analysis Documentation, Volume VIII)*. Prepared for DOE/NV. Las Vegas, NV.
- IT Corporation. 1996h. *Wells ER-6-1 and ER-6-2 Core Fracture Analyses and Geophysical Log Comparisons*, ITLV/10972/171. Prepared for the U.S. Department of Energy, Las Vegas, NV.
- IT Corporation. 1996i. *Draft, Underground Test Area Fracture Analysis Report: Analysis of Fractures in Volcanic Cores from Pahute Mesa*. Prepared for the U.S. Department of Energy. Las Vegas, NV.
- IT Corporation. 1996j. *A Stochastic Advective - Dispersive Transport Code for Multiple Constituents in Dual Porosity or Fractured Geological Media*. Prepared for the U.S. Department of Energy. Las Vegas, NV.
- Long, J.C.S., and D.M. Billaux. 1997. "From Field Data to Fracture Network Modeling: An Example Incorporating Spatial Structure." In *Water Resources Research*, 23(7): 1201-1216.
- Maloszewski, P., and A. Zuber. 1991. "Influence of Matrix Diffusion and Exchange Reactions on Radiocarbon Ages in Fissured Carbonate Aquifers." In *Water Resources Research*, 27(8): 1937-1945.

- McDonald, M.G., and A.W. Harbaugh. 1988. "A Modular Three-Dimensional, Finite Difference Ground-Water Flow Model." In *U.S. Geological Survey Techniques of Water Resources Investigations*, Book 6, Chapter A1.
- Pollock, D.W. 1989. *Documentation of Computer Programs to Compute and Display Pathlines using Results from the U.S. Geological Survey Modular Three-Dimensional, Finite-Difference Ground-Water Flow Model*, U.S. Geological Survey Open-File Report, 89-381.
- Smith, D.K. 1995. *Phenomenology of Underground Nuclear Explosions Conducted at the Nevada Test Site with Emphasis on Recent Experience at Baseball (U7ba) and Ingot (U2gg)*, Lawrence Livermore National Laboratory Report. Prepared for the DOE/NV Environmental Restoration Division for the Underground Test Area Operable Unit.
- Sudicky, E.A. 1990. *The Laplace Transform Galerkin Technique for Efficient Time-Continuous Solution of Solute Transport in Double-Porosity Media*, *Geoderma*, 46: 209-232.
- Triay, I.R., K.H. Birdsell, A.J. Mitchell, and M.A. Ott. 1993. "Diffusion of Sorbing and Non-Sorbing Radionuclides." In *Proceedings of the Fourth Annual International High Level Radioactive Waste Management Conference*. Las Vegas, NV: American Nuclear Society, Inc. and American Society of Civil Engineers.
- U.S. Department of Energy. 1994. *United States Nuclear Tests, July 1945 through September 1992*. DOE/NV-209 (Rev. 14).
- U.S. Environmental Protection Agency. 1989a. *Risk Assessment for Superfund, Human Health Evaluation Manual (Part A)*, EPA/5401/1-89/001. Washington, DC: U.S. Government Printing Office.
- U.S. Environmental Protection Agency. 1989b. *Risk Assessment Guidance for Superfund, Volume II: Environmental Evaluation Manual*, EPA/540/1-89/001. Washington, DC: U.S. Government Printing Office.
- U.S. Environmental Protection Agency. 1991. *Risk Assessment Guidance for Superfund, Human Health Evaluation Manual (Part B), Development of Risk-Based Preliminary Remediation Goals*, EPA/540/R-92/003. Washington, DC: U.S. Government Printing Office.
- U.S. Environmental Protection Agency. 1993. Multimedia exposure assessment model of evaluating the land disposal of hazardous wastes, Theory I. Prepared by Woodward-Clyde Consultants. EPA Contract No. 68-03-3513 and No. 68-03-6304. Environmental Research Laboratory. Athens, GA: U.S. Environmental Protection Agency.
- van Genuchten, M.Th. 1981. *Non-Equilibrium Transport Parameters for Miscible Displacement Experiments*, Research Report No. 119. Prepared for the U.S. Department of Agriculture. Riverside, CA: U.S. Salinity Laboratory.

VanMarke, E. 1983. *Random Fields: Analysis and Synthesis*. Cambridge, MA:
The MIT Press.

Yeh, G.T. 1981. *AT123D: Analytical Transient One-, Two-, and Three-Dimensional
Simulation of Waste Transport in the Aquifer System*. Version 1.22. ORNL/5521. Oak
Ridge, TN.

Appendix A

MC_TRANS Documentation

The documentation of the transport code MC_TRANS is presented. This is a stand-alone document that is reproduced here. The appendix includes a detailed presentation of the governing equations and the numerical approximation. A listing of the source code as it was used in the regional flow analysis is included. The end of the appendix includes several examples and verification test problems.

MC_TRANS

A STOCHASTIC ADVECTIVE-DISPERSIVE TRANSPORT CODE FOR MULTIPLE CONSTITUENTS IN DUAL POROSITY OR FRACTURED GEOLOGICAL MEDIA

PREPARED FOR:

THE U.S. DEPARTMENT OF ENERGY
NEVADA OPERATIONS OFFICE
LAS VEGAS, NEVADA

WORK PERFORMED UNDER CONTRACT NO.
DE-AC08-92NV10972

DECEMBER, 1996

TABLE OF CONTENTS

	Page
Acknowledgments	vi
Abstract	vii
1 INTRODUCTION	1
2 THE PHYSICAL SYSTEM	2
3 GOVERNING EQUATIONS	4
3.1 Transport in the Mobile Zone	4
3.2 Transport in the Immobile Zone	8
3.3 Mass Transfer Coefficient	10
3.5 Initial Conditions	11
3.6 Boundary Conditions	11
4 NUMERICAL SOLUTION	14
4.1 Basis Functions	14
4.2 Method of Weighted Residuals	15
4.3. Time Derivative	18
5 UNCERTAINTY ANALYSIS	20
6 PROGRAM STRUCTURE AND DESIGN	29
7 INPUT DATA INSTRUCTIONS AND PROGRAM USE	33
7.1 Data Input Records	34
7.2 Auxiliary ASCII Input Files	41
7.3 Auxiliary Binary Input Files	42
7.4 Program Use	42
7.5 Output Files	43
Unit 6 - Primary Output File	43
Unit 7 - Maximum at Receptors	43
Unit 8 - Time of Arrival Maximums	43
Unit 9 - Summary of Statistics	43
Unit 10 - Breakthrough Based on Mean Parameters	43
Unit 11 - Parameter Realizations	44
Unit 12 - Tabular Maximum Concentrations	44
Unit 20 - Binary Breakthrough Curves	44

TABLE OF CONTENTS (Continued)

	Page
8 MODEL VERIFICATION	45
8.1 Test 1 - Comparison with One-dimensional Transport with First-Type Boundary Condition	47
8.2 Test 2 - Comparison with One-dimensional Transport with Third-Type Boundary Condition	50
8.3 Test 3 - Comparison with One-dimensional Transport with Decaying First-Type Boundary Condition and Decaying Chemical	52
8.4 Test 4 - Comparison with One-dimensional Transport of Multiple Species	54
8.5 Test 5 - Comparison with One-dimensional Dual Porosity Transport with First-Type Boundary Condition	56
8.6 Test 6 - Comparison with One-dimensional Dual Porosity Transport with Third-Type Boundary Condition	58
8.7 Test 7 - Example Monte Carlo Application	59
8.8 Test 8 - Example Latin Hypercube Sample Application	61
9 REFERENCES	65
Appendix A - MC_TRANS Fortran Source Code Listing	
Appendix B - Verification Input Data Files	
Appendix C - Tabular Comparison of Verification Comparisons	
Appendix D - Source Code Listing of Verification Codes	

LIST OF FIGURES

	Page
2.1. Conceptual diagram of fracture and matrix systems.	2
2.2. Conceptual diagram of a dead-end pore space system (adapted from van Genuchten and Wierenga, 1976).	3
5.1. Latin hypercube sampling segments for normal and uniform distributions (from Iman et al., 1980).	25
6.1. Program control logic for Monte Carlo simulation.	30
6.2. Program control logic for Latin Hypercube simulation.	31
8.1 Test 1 Verification	48
8.2 Test 1 verification	49
8.3 Test 2 verification	51
8.4 Test 3 verification	53
8.5 Test 4 verification.	55
8.6 Test 5 verification	57
8.7 Test 6 verification	58
8.8. Maximum concentration frequency.	60
8.9a. Concentration breakthrough curves for mean parameters.	63
8.9b. Maximum concentration levels for Carbon-14.	64

LIST OF TABLES

	Page
8.1 Summary of Verification Problems.	46

Acknowledgments

The MC_TRANS code was originally programmed in the GeoTrans, Inc., Boulder, Colorado office by Scott Brogan. The code was modified, cleaned and reprogrammed by Barry Lester and David Ward of the GeoTrans Inc. office in Sterling, Virginia. Additional code modifications were implemented by Rick Waddell, Chen Zhu and Debra Cave, all of the GeoTrans Inc. Office in Boulder CO. Program testing was performed by David Ward and Barry Lester. Documentation was developed by David Ward and Barry Lester using Word Perfect 6.1 and Excel 5.0.

Abstract

The MC_TRANS model is a one-dimensional solute transport model, capable of simulating the migration of chains of constituents through fractured geologic media or through porous geologic media where diffusion of solute mass into dead-end pore space is an important factor. The model can be used in a deterministic mode or in a stochastic mode incorporating hydrologic parameter uncertainty through either a Monte Carlo approach or a Latin Hypercube Sampling (LHS) approach. The dual porosity representation of the geologic matrix system assumes complete mixing and a mass transfer coefficient controls interaction between fracture and matrix (or interconnected and dead-end pore space) systems. The contaminants are subject to advection, sorption and first-order decay in the fracture (or interconnected pore space) and sorption and first-order decay in the matrix (or dead-end pore space) domain. The chain-decay transport of up to four constituents can be simulated simultaneously. The Galerkin finite-element approach is used to solve the partial differential transport equations. The source may be input as a prescribed concentration or mass flux boundary condition and/or as a spatially variable initial condition. Parameter uncertainty allows for zonal variations of all input data using several types of statistical distributions. The model is implemented using Fortran77 and can be executed using PC and workstations.

1 INTRODUCTION

MC_TRANS is a one-dimensional model developed to simulate the advective-dispersive transport of decay-chains of dissolved chemicals within groundwater contained in fractured media, or in porous media, where diffusion into stagnant pore water plays an important role in the solute transport processes. The model, is designed to address parameter uncertainty using either a Monte Carlo approach or a Latin Hypercube Sampling (LHS) approach. The uncertainty analysis logic allows for consideration of various type of parameters distributions and allows for zonal variation of the distributions. MC_TRANS may also be used to perform sensitivity analyses where multiple runs are performed using a series of user defined subsets of one or more of the uncertain parameters. For each analysis, deterministic or mean values for uncertain parameters, will be used for the other parameters. The model is implemented using Fortran77 and can be executed using PC and workstations. The output files are designed for efficient synthesis and post-processing of simulations results.

2 THE PHYSICAL SYSTEM

The conceptual model of the physical system simulated by MC_TRANS, is based upon process of advective-dispersive transport in a dual porosity media with a mobile and an immobile zone. Two physical media can be represented by the model, a fractured porous medium and a porous medium subject to matrix diffusion into dead-end pore spaces. In the fractured media representation (Figure 2.1), the mobile zone represents the fracture domain and the immobile zone, the porous matrix blocks.

Figure 2.1. Conceptual diagram of fracture and matrix systems.

In the dead-end pore space representation (Figure 2.2) the mobile zone is representative of the interconnected pores and the immobile zone, the dead-end pores containing stagnant water. The transport mechanisms in the model allow for migration into and out of the mobile and immobile zones. Basic assumptions upon which the model is based include:

- Flow of fluid occurs in the fracture domain (or interconnected pore space) only.
- Mass transfer from the mobile domain to the immobile domain is through diffusion only and the diffusion can be approximated by a first-order mass transfer equation.
- Transport processes in the mobile domain include advection and dispersion, and processes in the matrix domain include diffusion only.

Figure 2.2. Conceptual diagram of a dead-end pore space system (adapted from van Genuchten and Wierenga, 1976).

- Local adsorption equilibrium is attained everywhere in the system, and can be described by linear isotherms.
- At any point along the longitudinal coordinate complete mixing occurs across the mobile zone due to transverse diffusion so that average concentrations can be considered for this zone.

3 GOVERNING EQUATIONS

The MC_TRANS code solves for contaminant transport in the mobile and immobile zones, simultaneously. The code addresses each of up to 4 constituents separately and solves the set of decay-chain equations sequentially.

3.1 Transport in the Mobile Zone

In a one-dimensional dual porosity formulation, the governing equation for mass transport of a dissolved constituent in the mobile zone, (the fracture zone or dynamic zone of a dead-end pore space model) can be written in the form:

$$\theta_m \frac{\partial C_m}{\partial t} + V_1 \frac{\partial S_m}{\partial t} = D_m \frac{\partial^2 C_m}{\partial x^2} - \frac{\partial(qC_m)}{\partial x} - \theta_m \lambda C_m - V \lambda S_m - \alpha(C_m - C_{im}) \quad (1)$$

where:

C_m	=	concentration of the contaminant in fluids in the mobile zone [M/L ³],
S_m	=	concentration of contaminant sorbed on fill [M/M] using the mobile-immobile model or fracture surface [M/L ²] for the dead-end pore space model
C_{im}	=	concentration of the contaminant in the fluid of the immobile zone [M/L ³],
θ_m	=	effective porosity of the mobile zone [-],
D_m	=	mobile-zone hydrodynamic dispersion coefficient [L ² /T],
V_1	=	volume conversion term [M/L ³ or 1/L] for the mobile-immobile and dead-end pore space models,
q	=	Darcy flux [L/T],
α	=	mobile-immobile zone mass transfer coefficient [1/T];
λ	=	first-order constituent decay rate [1/T],
t	=	time [T], and
x	=	space coordinate [L];

where:

$$D_m = \theta_m \alpha_1 v + \theta_m D^\circ \quad (2)$$

and α_1 is the longitudinal dispersivity [L_v], v is the average linear velocity of groundwater in the mobile zone [L/T] and D° [L²/T] the molecular diffusion coefficient in free-standing solution. The average linear velocity is defined as:

$$v = \frac{q}{\theta_{\text{eff}}} \quad (3)$$

where θ_{eff} is the effective porosity of the mobile zone. The decay constant is defined as:

$$\lambda \equiv \frac{\ln 2}{T^{1/2}} \quad (4)$$

where $T^{1/2}$ is the half life and $\ln 2 = 0.69314718$.

In a fractured media representation, the fracture width can be imbedded in the effective porosity term (i.e., $\theta_{\text{eff}} = b/B$ where b is the fracture aperture and B is the spacing between fractures) or explicitly considered in the mass balance for each constituent. In the explicit formulation, the mass balance equation can be written as :

$$\frac{b}{2} \theta_m \frac{\partial C_m}{\partial t} + V_2 \frac{\partial S_m}{\partial t} = \frac{b}{2} D_m \frac{\partial^2 C_m}{\partial x^2} - \frac{b}{2} \frac{\partial(qC_m)}{\partial x} - \frac{b}{2} \theta_m \lambda C_m - V \lambda S_m - \alpha(C_m - C_{\text{im}}) \quad (5)$$

where:

$$\begin{aligned} V_2 &= \text{volume conversion term [M/L}^2 \text{ or -] for the mobile-immobile and the dead-end pore space model.} \\ q &= \text{Darcy flux in the fracture [L/T],} \end{aligned}$$

α = fracture-matrix mass transfer coefficient [L/T], and
 θ_m = effective porosity of the fracture [-].

This is the formulation used in MC_TRANS. Note that the typical dead-end pore space model based on Equation 1 can be used by letting $b=2.0$ and assigning the other terms appropriately.

The sorption model in MC_TRANS is based on linear isotherms. For sorption on the mobile zone porous media, the distribution coefficient is written as (Freeze and Cherry, 1979):

$$K_{dm} = \frac{S_m}{C_m} \quad (6)$$

where S_m is defined as the mass of solute of the solid phase per mass of the solid phase [M/M] and can be differentiated over time to yield:

$$\frac{\partial S_m}{\partial t} = K_{dm} \frac{\partial C_m}{\partial t} \quad (7)$$

The retardation coefficient for the mobile stage can then be defined as:

$$R_m = 1 + \frac{\rho_{bm} K_{dm}}{\theta_m} \quad (8)$$

where ρ_{bm} is the bulk density and $\rho_{bm} = \rho (1 - \theta_m)$ where ρ is the grain density.

For a fractured media, there are two types of sorption. First, a dissolved constituent can be sorbed on the porous media within the fracture, referred to as the fracture fill. The second type of sorption can occur along the fracture-matrix face or fracture wall. Both types of sorption lead to the development of a retardation factor which related the speed of transport of the contaminant relative to the groundwater flow in the fracture.

Assuming that the dominant mode of sorption is associated with the fracture fill, the retardation can be defined as:

$$R_f = 1 + \frac{\rho_{bf} K_{df}}{\theta_f} \quad (9)$$

where f refers to the fracture and or mobile zone. For cases where the dominant sorption is on the fracture walls, based on the assumption of a linear isotherm, the fracture distribution coefficient is defined as:

$$K_a = \frac{S_f}{C_f} \quad (10)$$

where S_f is defined as the mass of solute of the solid phase per unit area of the solid phase $[M/L^2]$. For sorption on the fracture walls, Equation (10) differentiated over time yields:

$$\frac{\partial S_f}{\partial t} = K_a \frac{\partial C_f}{\partial t} \quad (11)$$

For cases where the dominant sorption is on the fracture walls, the retardation coefficient for the fracture can be defined as:

$$R_f = 1 + A K_a \quad (12)$$

where A is the surface area to void space ratio $[1/L]$ for the fracture opening (Freeze and Cherry, 1979). For an empty fracture ($\theta_f = 1.0$) Equation (12) becomes:

$$R_f = 1 + \frac{2 K_a}{b} \quad (13)$$

Defining the volume conversion term, V_2 , as:

$$V_2 = \frac{b\rho_{bm}}{2} \quad (14a)$$

for sorption on the mobile zone porous media or as

$$V_2 = 1.0 \quad (14b)$$

for sorption on a fracture wall then substituting Equation (14a) or (14b) along with the consistent choice of either Equation (8), (9), or (13) (where $R_m = R_f$ for equations (9) and (13)) into Equation (5) the general governing equation for mobile zone transport as used in MC_TRANS can be written as:

$$\theta_m R_m \frac{\partial C_m}{\partial t} = D_m \frac{\partial^2 C_m}{\partial x^2} - \frac{\partial(qC_m)}{\partial x} - \theta_m R_m \lambda C_m - \frac{2\alpha}{b} (C_m - C_{im}) \quad (15)$$

3.2 Transport in the Immobile Zone

For the immobile zone, the governing equation for each constituent can be written as:

$$\theta_{im} \frac{\partial C_{im}}{\partial t} + \rho_{bim} \frac{\partial S_{im}}{\partial t} = - \theta_{im} \lambda C_{im} - \rho_{bim} \lambda S_{im} + \alpha(C_m - C_{im}) \quad (16)$$

where:

- C_{im} = average concentration of contaminant in the pore fluid of immobile zone [M/L³],
- S_{im} = average concentration of contaminant sorbed on the solid matrix [M/M],
- ρ_{bim} = bulk density of the dry matrix [M/L³],

θ_{im} = effective porosity of the matrix immobile zone [-], and
 α = first-order mass transfer coefficient [1/T].

For a fractured media porous matrix representation where the matrix block width is explicitly stated, the mass balance for each contaminant can be written as:

$$\frac{B}{2} \theta_{im} \frac{\partial C_{im}}{\partial t} + \frac{B}{2} \rho_{bim} \frac{\partial S_{im}}{\partial t} = - \frac{B}{2} \theta_{im} \lambda C_{im} - \frac{B}{2} \rho_{bim} \lambda S_{im} + \alpha(C_m - C_{im}) \quad (17)$$

where:

B = width of the average spacing between fractures [L].

The migration between the fracture and matrix systems is controlled by α , a first-order mass transfer coefficient (Coats and Smith, 1967; van Genuchten and Wierenga, 1976). This discussed further in Section 3.3. Equation (17) is the general formulation used in MC_TRANS and parallel to the mobile zone equation reduces to Equation (16) when $B = 2.0$.

Based on the assumption of linear adsorption isotherm, the distribution coefficient can be written as:

$$K_{dim} = \frac{S_{im}}{C_{im}} \quad (18)$$

and the time derivative as:

$$\frac{\partial S_{im}}{\partial t} = K_{dim} \frac{\partial C_{im}}{\partial t} \quad (19)$$

Substituting Equations (18) and (19) into Equation (17), the transport equation can be written as:

$$\theta_{im} R_{im} \frac{\partial C_{im}}{\partial t} = -\theta_{im} R_{im} \lambda C_{im} + \frac{2 \alpha (C_m - C_{im})}{B} \quad (20)$$

where:

$$R_{im} = 1 + \frac{\rho_{bim} K_{dim}}{\theta_{im}} \quad (21)$$

is the retardation factor for the porous matrix.

3.3 Mass Transfer Coefficient

The mass transfer coefficient α is regarded as an empirical constant (Coats and Smith, 1964). There are several definitions including: van Genuchten (1981) and Sudicky (1990). For the fracture model, the formulation by Sudicky (1990) as implemented is:

$$\alpha = \frac{2}{B} (3 \theta_{im} D_{im}^*) \quad (22)$$

where D_{im}^* is the effective diffusion coefficient in porous matrix and, for isotropic media, is defined as (Bear, 1972),

$$D_{im}^* = \tau D^0 \quad (23)$$

where τ is the tortuosity of the porous matrix. The form of Equation (22) differs slightly from that presented in Sudicky (1990) which is:

$$\alpha = \frac{3 \theta_{im} D_{im}^*}{B^2} \quad (24)$$

The use of the transfer coefficient given in (22) is as implemented for equations (5) and (20) where Equation (24) is the equivalent for Equations (1) and (16).

3.5 Initial Conditions

The solution of the primary equations requires the specification of initial conditions for both the fracture and the matrix as follows:

$$C_m(x, t=0) = C_{m \text{ init}}(x) \quad (25)$$

and:

$$C_{im}(x, t=0) = C_{im \text{ init}}(x) \quad (26)$$

3.6 Boundary Conditions

In the one-dimensional flow field, the transport equations are subject to boundary conditions at the inlet where the source may be defined using either a first-type or Dirichlet (prescribed concentration) or a third-type or Neuman (prescribed flux) boundary in which the source is also allowed to decay as defined by a first-order decay constant as follows:

For prescribed concentration:

$$C_m(x=0, t) = C_{src} e^{-\gamma t} \quad (27)$$

For flux condition:

$$Q_c (x=0, t) = q C_m - D_m \frac{\partial C_m}{\partial x} = q C_{src} e^{-\gamma t} \quad (28)$$

where C_{src} is the source concentration at $t=0$. Only the boundary conditions for the first component in a chain of constituents is presented here. The code includes complete implementation of the Bateman equations for up to four components (Lester, et al, 1975). These equations allow for temporal decreasing source concentration for component 1. The daughter (component 2) and grand daughter (component 3) and great grand daughter (component 4) boundary concentration are defined by the source decay rates, γ , not the constituent first-order decay rates, λ . The Bateman equations for the boundary condition concentrations are as follows:

$$C_1 (x=0, t) = C_{src1} e^{-\gamma_1 t} \quad (29)$$

$$C_2 (x=0, t) = C_{src2} e^{-\gamma_2 t} + \frac{\gamma_1 C_{src1}}{\gamma_1 - \gamma_2} (e^{-\gamma_2 t} - e^{-\gamma_1 t}) \quad (30)$$

$$C_3 (x=0, t) = C_{src3} e^{-\gamma_3 t} + \frac{\gamma_2 C_{src2}}{\gamma_2 - \gamma_3} (e^{-\gamma_3 t} - e^{-\gamma_2 t}) + \gamma_1 \gamma_2 C_{src1} \left[\frac{e^{-\gamma_1 t}}{(\gamma_1 - \gamma_2)(\gamma_1 - \gamma_3)} - \frac{e^{-\gamma_2 t}}{(\gamma_1 - \gamma_2)(\gamma_2 - \gamma_3)} - \frac{e^{-\gamma_3 t}}{(\gamma_1 - \gamma_3)(\gamma_2 - \gamma_3)} \right] \quad (31)$$

and:

$$\begin{aligned}
C_4(x=0, t) = & C_{src4} e^{-\gamma_4 t} + \frac{\gamma_3 C_{src3}}{\gamma_3 - \gamma_4} (e^{-\gamma_4 t} - e^{-\gamma_3 t}) \\
& + \gamma_2 \gamma_3 C_{src2} \left[\frac{e^{-\gamma_2 t}}{(\gamma_2 - \gamma_3)(\gamma_2 - \gamma_4)} - \frac{e^{-\gamma_3 t}}{(\gamma_2 - \gamma_3)(\gamma_3 - \gamma_4)} + \frac{e^{-\gamma_4 t}}{(\gamma_2 - \gamma_4)(\gamma_3 - \gamma_4)} \right] \\
& + \gamma_1 \gamma_2 \gamma_3 C_{src1} \left[- \frac{e^{-\gamma_1 t}}{(\gamma_1 - \gamma_2)(\gamma_1 - \gamma_3)(\gamma_1 - \gamma_4)} + \frac{e^{-\gamma_2 t}}{(\gamma_1 - \gamma_2)(\gamma_2 - \gamma_3)(\gamma_2 - \gamma_4)} \right. \\
& \quad \left. - \frac{e^{-\gamma_3 t}}{(\gamma_1 - \gamma_3)(\gamma_2 - \gamma_3)(\gamma_3 - \gamma_4)} + \frac{e^{-\gamma_4 t}}{(\gamma_1 - \gamma_4)(\gamma_2 - \gamma_4)(\gamma_3 - \gamma_4)} \right]
\end{aligned} \tag{32}$$

The exit boundary condition for each constituent is specified such that the migrating contaminant passes freely out of the model domain :

$$\frac{\partial C_m}{\partial x} = 0 \tag{33}$$

4 NUMERICAL SOLUTION

The transport equations are evaluated using the Galerkin finite-element approach. The spatial domain is discretized into a series of equally-spaced one-dimensional finite elements, each delimited by end nodes. The finite elements span from the source inlet to the exit boundary. Properties are defined as constant for each element. The temporal terms are evaluated using a centered-in-time finite difference approximation. For each time step, the sets of mobile zone and immobile zone transport equations generated by the Galerkin scheme for each constituent are solved simultaneously. The sets of equations for each constituent are solved sequentially from parent, to daughter to granddaughter to great-granddaughter.

4.1 Basis Functions

In the finite element domain, a trial or basis function (Isotok, 1989) of the form $\hat{C}(x,t)$ may be assumed where:

$$\hat{C}(x,t) \approx \sum_{j=1}^N C_j(t) \phi_j(x) \quad j \in I(1,2,\dots,N-1,N) \quad (34)$$

and N = the number of nodes ranging from 1 to N . The trial function can be used to represent the dependent variable, either C_f or C_m of each constituent for this model.

The trial functions are defined using a linear approximation or chapeau function between the nodes of the mesh such that the function has a value of 1 at j and decreases linearly to 0 at node $j-1$ and $j+1$ and is zero elsewhere:

$$\phi_j(x) = \begin{cases} \frac{(x - x_{j-1})}{\Delta x} & x_{j-1} \leq x \leq x_j \\ \frac{(x_{j+1} - x)}{\Delta x} & x_j \leq x \leq x_{j+1} \\ 0 & x < x_{j-1} \quad \vee \quad x > x_{j+1} \end{cases} \quad (35)$$

Note that φ_j is defined over the domain $x = 0$ to $x = L$ where L is the fracture length. Substituting the trial functions (34) into Equations (15) and (17) and solving for the residual (error), two equations of the form:

$$\varepsilon_m = \theta_m R_m \frac{\partial \hat{C}_m}{\partial t} - D_m \frac{\partial^2 \hat{C}_m}{\partial x^2} + \frac{\partial(q\hat{C}_m)}{\partial x} + \theta_m R_m \lambda \hat{C}_m + \frac{2}{b} \alpha (\hat{C}_m - \hat{C}_{im}) \quad (36)$$

and

$$\varepsilon_{im} = \theta_{im} R_{im} \frac{\partial \hat{C}_{im}}{\partial t} + \theta_{im} R_{im} \lambda \hat{C}_{im} - \frac{2}{B} \alpha (\hat{C}_m - \hat{C}_{im}) \quad (37)$$

are generated. In the Galerkin method the objective is to minimize total residual error in the approximation and distribute the error over the whole domain.

4.2 Method of Weighted Residuals

Residual minimization is accomplished by integration of the product of the error and a weighting function $W_k(x)$ over the whole domain, such that:

$$\int_{\Omega} \varepsilon W_k(x) dx = 0 \quad (38)$$

In the Galerkin method the weighting function is chosen to be the same as the trial function so that Equation (38) becomes:

$$\int_{\Omega} \varepsilon \varphi_k(x) dx = 0 \quad (39)$$

where the domain Ω ranges from $x_1 = 0$ to $x_N = L$. Substituting Equation (36) into (39) and (37) into (39) and utilizing the definitions given by Equations (34) and (35) gives two sets of working equations upon which the model is based:

$$\begin{aligned} & \sum_{j=1}^N \frac{\partial C_{mj}}{\partial t} \int_0^L \theta_m R_m \varphi_j \varphi_k dx - \sum_{j=1}^N C_{mj} \int_0^L D \theta_j'' \varphi_k dx \\ & + \sum_{j=1}^N C_{mj} \int_0^L q \varphi_j' \varphi_k dx + \sum_{j=1}^N C_{mj} \int_0^L \theta_m R_m \lambda \varphi_j \varphi_k dx \\ & + \sum_{j=1}^N C_{mj} \int_0^L \alpha \varphi_j \varphi_k dx - \sum_{j=1}^N C_{im} \int_0^L \alpha \varphi_j \varphi_k dx = 0 \end{aligned} \quad (40)$$

and

$$\begin{aligned} & \sum_{j=1}^N \frac{\partial C_{im}}{\partial t} \int_0^L \theta_{im} R_{im} \varphi_j \varphi_k dx + \sum_{j=1}^N C_{im} \int_0^L \theta_{im} R_{im} \lambda \varphi_j \varphi_k dx \\ & - \sum_{j=1}^N C_{mj} \int_0^L \alpha \varphi_j \varphi_k dx + \sum_{j=1}^N C_{im} \int_0^L \alpha \varphi_j \varphi_k dx = 0 \end{aligned} \quad (41)$$

where

$$\varphi = \varphi(x) \quad (42a)$$

$$\varphi' = \frac{\partial \varphi(x)}{\partial x} \quad (42b)$$

and

$$\varphi'' = \frac{\partial^2 \varphi(x)}{\partial x^2} \quad (42c)$$

To simplify the equation the second order term in Equation (40) is expanded using integration by parts to generate:

$$\begin{aligned} \int_0^L \varphi_j'' \varphi_k \, dx &= \varphi_k \varphi_j' \Big|_0^L - \int_0^L \varphi_j' \varphi_k' \, dx \\ &= \varphi_k(L) \varphi_j'(L) - \varphi_k(0) \varphi_j'(0) - \int_0^L \varphi_j' \varphi_k' \, dx \end{aligned} \quad (43)$$

Note that the above application of integration by parts disregards the spatially variable transport properties. Technically these properties are also included in the integration but for simplicity are disregarded in the above explanation. Since the formulation considers properties to be constant between nodes (for each element) the two non-zero terms of which each $\varphi_j(x)$ is comprised (see Equation (35)) each need to be multiplied by constant properties to derive the full integral.

The fact that Equation (35) implies that each $\varphi_j(x)$ is a function of two non-zero terms means that Equation (40) can thus be rewritten in the form:

$$\begin{aligned} &\sum_{j=1}^N \frac{\partial C_{mj}}{\partial t} \int_0^L \varphi_f R_m \varphi_j \varphi_k \, dx + \sum_{j=1}^N C_{mj} \int_0^L D \varphi_j' \varphi_k' \, dx \\ &+ \sum_{j=1}^N C_{mj} \int_0^L q \varphi_j' \varphi_k \, dx + \sum_{j=1}^N C_{mj} \int_0^L \varphi_m R_m \lambda \varphi_j \varphi_k \, dx \\ &+ \sum_{j=1}^N C_{mj} \int_0^L \alpha \varphi_j \varphi_k \, dx - \sum_{j=1}^N C_{imj} \int_0^L \alpha \varphi_j \varphi_k \, dx \\ &- D_{N-1} \varphi_k(L) \varphi_j'(L) + D_1 \varphi_k(0) \varphi_j'(0) = 0 \end{aligned} \quad (44)$$

where:

$$\begin{aligned} D_1 &= \text{the dispersion coefficient to element number 1, and} \\ \text{AND-1} &= \text{the dispersion coefficient for element number (N-1).} \end{aligned}$$

4.3. Time Derivative

For the finite element formulation, the time derivatives in Equations (44) and (41) are replaced by finite difference approximations of the form:

$$\frac{\partial C}{\partial t} = \frac{C^{t+1} - C^t}{\Delta t} \quad (45)$$

where C^t is the starting concentrations for each time step and C^{t+1} the concentration to be solved for. The formulation uses a central difference in time approximation so that the nodal concentrations in Equations (44) and (41) are:

$$C_{mj}(t) = 1/2 [C_{mj}^{t+1} + C_{mj}^t]$$

and

$$C_{imj}(t) = 1/2 [C_{imj}^{t+1} + C_{imj}^t] \quad (46)$$

Substituting Equations (45) and (46) into Equations (44) and (41) two sets of N simultaneous equations are generated of the form

$$A_1 C_{mj-1}^{t+1} + B_1 C_{mj}^{t+1} + C_1 C_{mj+1}^{t+1} + A_2 C_{imj-1}^{t+1} + B_2 C_{imj}^{t+1} + C_2 C_{imj+1}^{t+1} = D_1 \quad (47)$$

for the fracture and

$$A_3 C_{mj-1}^{t+1} + B_3 C_{mj}^{t+1} + C_3 C_{mj+1}^{t+1} + A_4 C_{imj-1}^{t+1} + B_4 C_{imj}^{t+1} + C_4 C_{imj+1}^{t+1} = D_2 \quad (48)$$

for the block matrix. All known terms are lumped on the right hand side (RHS) of Equations (47) and (48). The 2N equations represented by Equations (47) and (48) are solved simultaneously for the 2N dependent variables.

5 UNCERTAINTY ANALYSIS

As previously described in Sections 2 through 4, the MC_TRANS computer code is designed to simulate the advective-dispersive transport of dissolved species through 1) fractured porous media or 2) porous media where diffusion into and out of dead-end pore space plays an important role in transport processes. The simulation of advective-dispersive transport of a dissolved species is based upon a set of user defined parameters describing the medium-specific, chemical-specific and source-specific nature of the modeled system.

Because of errors in measurements, spatial and temporal variability in the media and/or financial restrictions in data gathering efforts, values of specified parameters cannot be exactly known. In many circumstances it becomes more appropriate to express the parameter values in terms of probability distributions as opposed to in terms of deterministic values. In this way the end-users of the analysis can look at simulation results in terms of likelihood of events and can assess the impact of data variability on simulation results.

The primary objective of uncertainty analysis is to estimate the uncertainty in spatial distributions and/or temporal changes in distributions of dissolved species given the uncertainty in the input parameters (USEPA, 1993). Simulation results reflect the cumulative probability distribution of resultant observation point concentrations or functions of these concentrations (ie. maximum concentrations) as determined from known or assumed distributions of one or more of the deterministic model input parameters. Mathematically, observation point concentrations can be viewed as functions of a vector of model input parameters (p_i), where:

$$C_{obs} = f(p_i) \quad (49)$$

One or more of the parameters p_i may be random variables described by a cumulative probability distribution function. The model can then be used to calculate the cumulative distribution function F_{Cobs} which can be defined as:

$$F_{\text{Cobs}}(C_{\text{obs}}') = P(C_{\text{obs}} \leq C_{\text{obs}}') \quad (50)$$

where P denotes probability. MC_TRANS contains two options for evaluating F_{cobs} , a Monte Carlo option and a Latin Hypercube Sampling option. The sampling logic for the Monte Carlo option is self-contained within MC_TRANS. For utilization of the Latin Hypercube Sampling option, an extra input file must be generated containing a list of all sampled parameters for each realization. A version of Sandia's LHS model (Davenport and Inman, 1982), modified for use in conjunction with MC_TRANS, can be used to generate the realization input. Each option has its own advantage. Monte Carlo analysis has the advantage of being more rigorous while the LHS method has the advantage of being more efficient and allowing for consideration of correlation between parameters.

5.1 MONTE CARLO SAMPLING

The Monte Carlo analysis procedure used in MC_TRANS is based on the generation of a set of realizations or deterministic simulations, with each realization defined by a set of n deterministic input parameters needed to compute solute concentration levels at selected observation points. In the Monte Carlo procedure, at least one of the input parameters, p_i , must be represented by a cumulative probability distribution. For each realization, a pseudo-random value of each uncertain input variable is drawn from its specified probability distribution. A deterministic simulation is performed for each set of input variables and the model responses, in terms of observation point concentrations over time and maximum concentrations, are generated. The model responses are outputted in a form convenient for post processing, so that a cumulative probability distribution of the model responses can be generated along with any other statistical analyses desired to aid in decision making. The number of realizations needed in a Monte Carlo analysis is a function of the uncertainty levels at which accurate results are required and the number of parameters that are described by cumulative probability distributions. Typically an

acceptable convergence at the 95th percentile, when several of the input parameters are described by statistical distributions, may take may take 2,000-4,000 realizations to attain

In a MC_TRANS Monte Carlo simulation, input parameters may be deterministic or uncertain. For uncertain parameters, parameter uncertainty may be defined by normal, lognormal, uniform and user defined distributions. Normal distributions are defined by a bell shaped probability curve which is symmetrical about a mean value. The normal distribution for parameter x is defined by two parameters, the mean, m_x , and the standard deviation, S_x . The probability density function of parameter x is defined as:

$$f_n(x) = \frac{1}{S_x \sqrt{2\pi}} \exp\left[-0.5\left(\frac{x-m_x}{S_x}\right)^2\right] \quad (51)$$

The cumulative probability distribution can then be written in the form:

$$F_n(x) = \int_{-\infty}^x f_n dx' \quad (52)$$

A lognormal distribution is a skewed distribution with the natural log of the parameter x being normally distributed. If the natural log of the parameter x is y , the probability distribution of the transformed variable y is normal and can be described by the mean, m_y , and the standard deviation, S_y , of the transformed parameter. The relationships between the mean, m_y , and standard deviation, S_y , of the transformed parameter and the mean and standard deviation of the parameter x are:

$$m_x = \exp[m_y + 0.5(S_y)^2] \quad (53)$$

and

$$S_x = m_x^2[\exp(S_y^2) - 1] \quad (54)$$

A uniform distribution is a probability distribution, defined by a minimum value and a maximum value, in which all values within this range have an equal probability of occurring. A uniform probability density distribution may be defined as:

$$f_u(x) = \frac{1}{B-A} \quad (55)$$

where A is the lower bound (minimum) and B is the upper bound (maximum). The cumulative probability density distribution function, $F_u(x)$ can then be defined as:

$$F_u(x) = \frac{x-A}{B-A} \quad (56)$$

5.2 LATIN HYPERCUBE SAMPLING

The Latin hypercube sampling (LHS) method is an effective sampling technique developed to generate a reliable set of realizations that will reflect the statistical distributions of the sampled parameters without needing as large a sample set as the Monte Carlo method does. The method is based upon the division of the range of each variable into n intervals of equal probability. By forcing each sampling interval to have a probability of $1/n$, the endpoints of each interval can be defined by the probability distribution of the variable (Figures 5.1a and 5.1b).

For uniform or nonuniform samples, the range is a given. In theory, the range of normal and lognormal distributions are infinite or undefined. In practice, an effective range of these distributions can be defined. For normal and lognormal distributions, an acceptable range for the samples, A to B, is chosen based on an acceptable truncation of the infinite range of the

distribution. For the sampling methodology used here, the lower (A) and upper (B) ranges of the sample population are based on the assumption that A represented the 0.001 quantile and B the 0.999 quantile, or ± 3.09 standard deviations from the mean. In terms of mean and standard deviation, the mean (or mean of the log) is defined by $\mu = (A + B)/2$ and the standard deviation $\sigma = (B - \mu)/3.09 = (B - A)/6.18$. The variable samples are then derived by random selection within each interval with the density of the variable prorated within the interval.

The choice of the Sandia LHS model was also based upon the option to evaluate whether or not the assumption of independence among variables is appropriate. In “reality”, correlations exist among some of the variables used in the model. In a practical sense, if the results of the modeling effort are not unduly biased by neglecting these correlations, it is much more time and resource effective not to develop the correlation matrices needed for the study. The 1982 release of Sandia’s LHS model (Davenport and Inman, 1982), allows the user to import an “approximate correlation matrix”. This model is especially practical in its capability of allowing the modeler to use a sophisticated guess of a set of correlation coefficients. Although such a set of pair-wise correlation coefficients may not assemble into a positive-definite correlation matrix, the Sandia code contains an algorithm that can take sets of pair-wise correlations and iteratively adjust the correlation matrix until it is positive-definite. This allows the user to assume different degrees of correlation among variables and evaluate the need for studying the correlations between variables.

5.3 FLOW SENSITIVITY PARAMETERS

One-dimensional advective-dispersive transport simulations may involve groundwater travel along stream tubes, with pore velocity components derived from fully three-dimensional analyses. Uncertainty in the velocities along these specified pathways can be considered by utilizing the results of fully three-dimensional sensitivity analyses and considering the influence of changes in specific flow parameters, within the three-dimensional model, on the velocity

data/marketing/3706_1.cdr

data/marketing/3706_2.cdr

Figure 5.1. Latin hypercube sampling segments for normal and uniform distributions (from Iman et al., 1980).

components in the streamtube. Typical input parameters, P_i , considered in sensitivity studies include horizontal hydraulic conductivity (K_h), vertical hydraulic conductivity (K_v) and a functional parameter describing the variance of hydraulic conductivity with depth (L_d). Sensitivity parameters, $S_{ij} = \Delta V_j(x,y,z)/\Delta P_i$, describing the change of velocity components, $V_j(x,y,z)$, versus change in each parameter, P_i , can be used in MC_TRANS to evaluate the uncertainty in results due to uncertainty in flow parameters used in the three-dimensional analysis. Assuming a statistical distribution for each of N sensitivity parameters, and using either a Monte Carlo analysis or generating a Latin Hypercube Sample set for each parameter P_i , the fracture-flow velocity for each finite element can be defined as:

$$\hat{V}(x) = \sqrt{V_1^{*2} + V_2^{*2} + V_3^{*2}} \quad (57)$$

where:

$$V_1^* = V_1 + \sum_{i=1}^N S_{i1} \frac{P_i - \bar{P}_i}{\bar{P}_i} \quad (58)$$

$$V_2^* = V_2 + \sum_{i=1}^N S_{i2} \frac{P_i - \bar{P}_i}{\bar{P}_i} \quad (59)$$

$$V_3^* = V_3 + \sum_{i=1}^N S_{i3} \frac{P_i - \bar{P}_i}{\bar{P}_i} \quad (60)$$

and P_i is the sampled parameter.

A simplified sensitivity function relating the sensitivity of the velocity field to changes in recharge due to infiltration can also be considered in a MC_TRANS uncertainty analysis. The simplified function assumes that the velocities are directly related to the infiltration rate, with element velocities being multiplied by the ratio of a pseudo-randomly generated infiltration rate and the average infiltration rate of the area being modeled. Assuming a statistical distribution for probable infiltration rates, and using either a Monte Carlo analysis or generating a Latin Hypercube Sample set for each realization rate, R_i , the fracture-flow velocity for each finite element ($\hat{V}^*$) can be defined as:

$$\hat{V}^* (x) = \hat{V} \frac{R_i}{\bar{R}} \quad (61)$$

where $\bar{R}$ is the mean of the infiltration rate distribution.

5.4 SENSITIVITY ANALYSES

MC_TRANS also allows the user to perform sensitivity analyses, where the user inputs a separate file containing the a list of sensitivity runs he wishes to perform. Each sensitivity run will use the same general input file, but for each sensitivity analysis only the parameters indicated will be varied. Deterministic values or for uncertain variables, mean values will be used for parameters not indicated for that specific sensitivity analysis.

6 PROGRAM STRUCTURE AND DESIGN

The MC_TRANS computer code is written using Fortran 77. It consists of the main program and seventeen subroutines (Listed in Appendix A). The data storage uses labeled common statements, in which array lengths are declared through the parameter statement. The generally common file, MC_TRANS.INC, is embedded in most of the primary subroutines. There are eight user-definable parameters as listed below:

NN	Max number of nodes in the finite element model.
NDMAX	Max number of values in a raw data parameter distribution.
NPARDIM	Max number of parameters for which distributions can be input.
NSTPS	Max number of time steps in the finite element transport simulation.
MCMAX	Max number of Monte Carlo realizations simulated.
NOBS	Max number of receptor/observation locations for which breakthrough concentrations and risk are calculated.
NPARID	Min number of parameters to run the model.
NS	Max number of species in the model (must be less than or equal to 4).
NFSP	Max number of flow model parameters to be varied in Monte Carlo simulations

To increase the array size, the lengths, as specified in each of the parameter strings is increased and the code is recompiled.

The overall program logic is presented in Figure 6.1 for the deterministic or Monte Carlo option and Figure 6.2 for the Latin Hypercube Sample option. The selection of the Monte Carlo or Latin Hypercube option is controlled by user input, and one simulation using the mean variables

Figure 6.1. Program control logic for Monte Carlo simulation.

Figure 6.2. Program control logic for Latin Hypercube simulation.

is always performed. The option for chains of chemicals is user controlled, but currently limited to a maximum of four species. Time step logic is based on uniform time increments and the same time stepping is used for all species. There are several output files subject to user control.

7 INPUT DATA INSTRUCTIONS AND PROGRAM USE

Data input is contained in three ASCII file and 4 BINARY files. The primary ASCII file contains the majority of the input data and may be created or edited using any text editor. The second and third ASCII files required only if a sensitivity analysis and/or Latin Hypercube analysis is performed. One of the optional files contains the LHS sampled values for the uncertain parameters ('LHS.MCT'). The second optional file contains the sets of uncertain parameters to be sampled for each sensitivity run ('TRANSEN.DAT'). These files must be named as noted above. The 4 BINARY files are created by the code, SENSVEC (see Figure 1 of the main text). SENSVEC is an auxillary code that processes sensitivity data from the flow model and extracts sensitivity coefficients and a velocity field along the flow path. This allows the variability of the flow model parameters to be included in the Monte Carlo approach. The MC_TRANS code is a batch oriented processor, requiring up to seven input files and optionally creating one or more output files. One of the input files uses list-directed input format, sometimes referred to as free format. Each variable is separated by blanks or commas and the read may be terminated using a "/". This style allows for a very compact input files, while also allowing for comments or record identifiers to be included. Listed below is an a portion of a sample input file. There are no column positions, only record sequence is important:

```
Test 1 - {T1C1P1.dat} - First-type B.C. [CR1 = 0.05 , PE1 = 0.1]
5 1213 / MC, ISEED
101 50.0 300 .33333333 1.E30 -1.0 / NUMX, DELX, NUMT, DELT, TOFF, SCL_PERM
1 / INBOUN
1 / NUMOBS
11 / NOBSLOC(NUMOBS)
18 1 4 0 0 1 / NPARD, NSPEC,NFSP,NSENS,NLHS,NRC
1 1 1 100 0 20.0 0.0 / C_SRCE (1)
2 1 1 100 0 0.0 0.0 / GAMMA (2)
3 1 1 101 0 0.0 0.0 / CINIT (3)
4 1 1 101 0 0.0 0.0 / SINIT (4)
5 1 1 100 0 3.0 2.0 / GRAD (5)
6 1 1 100 1 20.0 2.0 / DISP (6)
7 1 1 100 0 1.00e-9 2.0 / DM (7)
8 1 1 100 2 1.0 2.0 / TAU (8)
9 1 1 100 1 0.200 2.55 / THETA1 (9)
10 1 1 100 1 0.200 0.02 / THETA2 (10)
11 1 1 100 0 1.00 0.0 / BWIDTH (11)
```

```

12  1  1 100 0   1.00    0.0  / FWIDTH  (12)
13  1  1 100 2   1.6    0.1  / RHOWB   (13)
14  1  1 100 0   0.0    0.0  / RKD1    (14)
15  1  1 100 0   0.0    0.0  / RKD2    (15)
16  1  1 100 0  0.0e+0  0.0  / DIFF    (16)
17  1  1 100 0 0.000000  0.0  / BETA    (17)
18  1  1   1 3   35.0   45.0  / FLOW PARAMETER #1 (ie. HYDRAULIC COND.) (18)
19  1  1   1 1   10     5.0  / FLOW PARAMETER #2  (19)
20  1  1   1 1   15    10.0  / FLOW PARAMETER #3  (20)
21  1  1   1 2   0.0    0.33 / FLOW PARAMETER #4 (Recharge) (21)
1 1 1 1 1 1 1 1 1 1 / IOU6, IOU7, IOU8, IOU9, IOU10, IOU11, IOU12, IOU20

```

The binary files cannot be edited and must be generated with the code SENSVEC if the flow path changes. The program will perform a limited amount of data checking, but the user must follow basic rules of supplying all required records and following the Fortran standard for the specification of integer and real variables.

7.1 Data Input Records

The input instructions detail the contents for each record. All record types (except for the sensitivity parameters) are mandatory for proper execution, but several lines may be used for those records which are used to define the spatially varying properties. The input records are defined in this section.

Record 1 List: TITLE

Variable	Description
----------	-------------

TITLE	Run description, up to 72 characters.
-------	---------------------------------------

Record 2 List: MC, ISEED

Variable	Description
MC	Number of Monte Carlo or LHS realizations: 0 or 1 for deterministic > 1 for Monte Carlo or LHS simulation (Maximum of MCMAX)
ISEED	Value of seed for random number generator. Value should be a positive integer if Monte Carlo option is used (MC > 1).

Record 3 List: NUMX, DELX, NUMT, DELT, TOFF, SCL_PERM

Variable	Description
NUMX	Number of finite element nodes. (Value must be less than NN in parameter statement.)
DELX	Uniform distance between nodes, i.e. element length.
NUMT	Number of time steps. (Value must be less than NSTPS in parameter statement.)
DELT	Uniform time increment. Total simulation time = NUMT x DELT.
TOFF	Elapsed time at which source boundary condition is terminated.
SCL_PERM	Scaling to calculate velocity using the relation $v = \text{grad} * \text{SCL_PERM} * (\text{FWIDTH}^{**2})$. Enter a value less than or equal to zero for direct input of velocity using GRAD, Distribution 5, Record 8.

Record 4 List: IBOUN

Variable	Description
----------	-------------

IBOUN	Inflow node boundary condition specification: = 1 for prescribed concentration (Dirichlet, first-type) = 2 for prescribed flux (Cauchy, third-type)
-------	---

Record 5 List: NUMOBS

Variable	Description
----------	-------------

NUMOBS	Number of observation node at which output files will be written. Value must be less than NOBS in parameter statement.
--------	--

Record 6 List: (NOBSLOC (IO), IO = 1, NUMOBS)

Variable	Description
----------	-------------

NOBSLOC	Array of element nodes for writing observation data. This list may span several records of input. A maximum of NOBS values.
---------	---

Record 7 List: NPARD, NSPEC, NFSP, NSSENS, NLHS, NRC

Variable	Description
NPARD	Number of parameter distribution lines to be read in. To complete the data file, at least equal to or greater than NPARID, a value of 17. The number of distributions will exceed NPARID when spatial variations in transport properties of initial conditions are specified. The purpose of NPARD is to provide the program with a value of Record 8 lines to read from the data file.
NSPEC	Number of chemical species (between 1 and 4). The value must be less than or equal to NS. Note that source decay equations are only programmed for up to 4 species.
NFSP	Number of flow model parameters that will be varied in the Monte Carlo Realizations.
NSSENS	Number of sensitivity runs.
NLHS	Number of Latin Hypercube Sample parameters.
NRC	Recharge sensitivity indicator (0 = no; 1 = yes).

Record 8 One or more records are input for each of the 17 parameter distributions and the NFMP flow system parameters. Below is a list of the parameters and their ID's. The list indicates whether a parameters spatial distribution is defined from each node or element and whether it is the same for each species or needs to be defined for each species. The list is followed by the format for each record to define the distributions. The parameter distributions are as follows.

Dist.	Variable	Definition	Node	Element	Species
1	C_SRCE	Source concentration			x
2	GAMMA	Source first-order decay constant			x
3	CINIT	Initial concentration in fracture	x		x
4	SINIT	Initial concentration in matrix	x		x
5*	GRAD	Velocity gradient along fracture		x	
6	DISPL	Longitudinal dispersivity in fracture zone		x	
7	DM	Free-water diffusivity			
8	TAU	Fracture tortuosity		x	
9	THETA1	Fracture zone porosity		x	
10	THETA2	Matrix porosity		x	
11	BWIDTH	Block width, i.e. fracture separation		x	
12	FWIDTH	Fracture width		x	
13	RHOB	Bulk density		x	
14	RKD1	Distribution coefficient in fracture region		x	x
15	RKD2	Distribution coefficient in matrix region		x	x
16	DIFF	Matrix region diffusion coefficient		x	x
17	LAMBDA	First order decay constant		x	
18	FSP	Flow model parameter 1		x	
-	-	-			

17+NFSP FSP** Flow model parameter NFSP x

*GRAD can be used to specify either velocity or gradient (see SCL_PERM on record 3).

**The final FSP in the list is the recharge sensitivity parameter if NRC = 1.

For each distribution, seven parameters are used to specify the type of distribution, the elements to apply the distribution and the parameters which comprise the distribution. The input for each distribution is listed below. Note that a distribution is required for each model variable, including deterministic simulations.

List: IPARID(IP,1), IPARID(IP,2),IPARID(IP,3), IPARID(IP,4), IDIST, DMEAN, DSTD

Variable	Description
IPARID(IP,1)	Parameter identification index as specified in Record 8. For example 3 corresponds to the model parameter CINIT.
IPARID(IP,2)	Species index number. Used only for: C_SRCE (PARID=1), CINIT (PARID=3), SINIT (PARID=4), RKD1 (PARID=14), DIFF (PARID=16), LAMBDA (PARID=17). Value must be equal or less than NS.
IPARID(IP,3)	Starting element or node number to apply distribution.
IPARID(IP,4)	Ending element or node number to apply distribution.
IDIST	Integer specification for the type of parameter distribution for species IP: = 0 Deterministic, where mean (DMEAN) is used.

- = 1 Normal distribution, using mean (DMEAN) and standard deviation (DSTD).
- = 2 Lognormal distribution, using log mean (DMEAN) and log standard deviation (DSTD).
- = 3 Uniform distribution, using minimum (DMEAN) and maximum (DSTD).
- = 4 User-defined distribution
- = -i Parameter number i in Latin Hypercube Sample file ('LHS.MCT')

DMEAN Defines the mean or minimum value.

DSTD Defines the standard deviation or maximum.

Note: For lognormal distribution, the mean is defined by the normal statistics as applied to the natural logarithm of the variable. In other words, it is the resultant mean obtained from applying a normal distribution fit, after transforming to the log. The user must be careful not to confuse that the log of the mean is not equal to the mean of the log. For additional reading, see Benjamin and Cornell, 1970. When using programs such as Excel or Stanford graphics, these program work with both the mean log and log mean. Also note that this version of MC_TRAN assumes log base 10 is used not the natural log.

Record 9 List: IOU6, IOU7, IOU8, IOU9, IOU10, IOU11, IOU12, IOU20

Variable	Description
IOU6	Primary output file configured for printable summary of simulation.
IOU7	Output of maximum concentration at receptors.

IOU8	Output of time of arrival maximums.
IOU9	Summary of statistics used in Monte Carlo parameter sampling.
IOU10	Breakthrough curve for mean parameters.
IOU11	Parameter realizations.
IOU12	Tabular summary of maximum concentration values.
IOU20	Binary file of breakthrough curves.

For each of the above input control parameters, the following apply:

- = 0 No output is written.
- = 1 Only the mean simulation results are written
- = n Every n'th realization of the Monte Carlo or LHS sampling or time step is written.

7.2 **Auxiliary ASCII Input Files**

Two auxiliary ASCII input files may be required by user chosen options. If the Latin Hypercube Sample option is chosen, an input file ('LHS.MCT') is required. For a set of MC samples, the file contains MC+1 sets of realization parameters, listed one realization at a time in "free format." The extra realization is the first one which contains the mean values of the parameters. Only the sampled parameters are listed in the file. The order that the parameters are expected to be found in, is based on the absolute value of IDIST (see RECORD 8 above). The data file can be generated using a version of Sandia's LHS model (Davenport and Inman, 1982), modified for use in conjunction with MC_TRANS called LHSMC. A sample file can be found in APPENDIX B Problem 8.

If the sensitivity analysis option is used, an input file ('TRANSEN.DAT') is required. The file contains NSENS sets of data written in free format. Each set of data starts with a line with an integer variable indicating the number of parameter records that will be sampled in that analysis. All other parameter values will be either the mean value or deterministic value for that parameter.

After this line, the position number parameter records containing the data to be sampled are listed in free format (see far right of parameter records in file presented above).]

7.3 Auxiliary Binary Input Files

Four binary files are also needed if the flow sensitivity parameter option is utilized. The files contain the sensitivity parameters for the x-, y-, and z directions and the velocity vectors. These files are generated using SENSVEC and must all have the same name with the trailers, SNX, .SNY, .SNZ and .VEL.

7.4 Program Use

The program is a batch-oriented processor requiring the preparation of input data files using an ASCII text editor, or preprocessor, and 4 binary files created with an external program. The binary files contain the data describing the sensitivity of the specific discharge to flow model parameters. This allows the calculation of the impact on specific discharge (and therefore velocity) of flow model uncertainty. Finally, MC_TRANS creates several output files. After installation and possible recompilation, the program is invoked as a stand-alone executable program from the DOS prompt level. The code request the user to provide the root name of the primary and secondary input files and automatically names all output files.

7.5 Output Files

There are eight possible output files created from the execution of each input data file. These include output echo of input, tabular summaries and a binary file for later post-processing. In the following section the contents of each files are given.

Unit 6 - Primary Output File

This file echoes the input variable in a cleanly formatted style, and is intended for report creation. The file includes an error diagnostics identified.

Unit 7 - Maximum at Receptors

The maximum concentration at selected receptors (observation nodes) is written. This provides a summary of the breakthrough curves for use in risk assessment using the acute dose, based on the maximum predicted concentration value. This value of time at which the maximum concentration is predicted to occur is written to unit 8.

Unit 8 - Time of Arrival Maximums

As discussed above, this at which the maximum concentration is predicted to occur is written to this file.

Unit 9 - Summary of Statistics

This file provides a traceable check on the statistical parameters.

Unit 10 - Breakthrough Based on Mean Parameters

For all simulations, one suite of transport calculations are performed using the mean values for all input variables. This output summarizes the concentration versus time, or breakthrough curve for each of the receptor location, also referred to as observation nodes.

Unit 11 - Parameter Realizations

For completeness, the code can create a file of the random sampling of model input parameters. This is useful in developing an appreciation of the actual models used in the transport calculations.

Unit 12 - Tabular Maximum Concentrations

A summary of the maximum concentration can be created.

Unit 20 - Binary Breakthrough Curves

For post-processing a binary file of the complete breakthrough curve can be written. These data must be read with an auxiliary Fortran program and includes a header record.

8 MODEL VERIFICATION

To verify the model, eight problem sets were developed and executed. The results from the first six problems are compared with analytical solutions in order to verify the mathematical solution, the seventh verification performs a mechanical test of some of the Monte Carlo features and the eighth simulation serves to perform both a mechanical and solution verifying test of the Latin Hypercube analysis logic. In the following sections, the problem definition and input data are described, along with graphical display of the model results. In Appendix B, the input data files are included. In Appendix C, tabular summaries for the first six problems are provided.

Problem Number	Transport Media	Simulation	Source Boundary Condition	Number of Components	Verification Reference	Comments
1	Porous	Deterministic	First-type (Constant Concentration)	1	Wexler, 1989	Three Peclet and Courant number cases, each
2	Porous	Deterministic	Third-type (Flux)	1	Wexler, 1989	Effect of flux boundary condition
3	Porous	Deterministic	First-type decaying	1	Wexler, 1989	Equal source and chemical decay
4	Porous	Deterministic	First-type	3	Intera, 1983	Chain decay
5	Dual porosity	Deterministic	First-type	1	van Genuchten, 1981	Pulse source with dual porosity tailing
6	Dual porosity	Deterministic	Third-type	1	van Genuchten, 1981	Effect of third-type boundary condition
7	Porous	Monte Carlo	First-type (Constant Concentration)	1	(none)	Functional verification of Monte Carlo feature
8	Dual porosity	Latin Hypercube Sample	First-type (Constant Concentration)	1	Sudicky, 1990, Sudicky and Frind, 1982	Functional verification of LHS feature and verification of dual porosity option

Table 8.1 Summary of Verification Problems.

8.1 Test 1 - Comparison with One-dimensional Transport with First-Type Boundary Condition

A comparison was performed for the simple case of advective-dispersive transport, subject to a first-type, or constant concentration boundary condition. No decay or adsorption is included and uniform properties are assigned to simulate transport with the mean values with MC_TRANS. The analytical solution is was first presented by Lapidus and Amundson, 1952. The computer program used to solve this is WEXLER as presented by Wexler, 1989. The code for the analytical solution is presented in Appendix D. A summary of the basic transport and model parameters is listed below.

Parameter Definition	Variable	File	Value	Units
Number of elements	NUMX		101	
Element length	DELX		50.0	m
Number of time steps	NUMT	t1c1p2 t1c2p2 t1c3p2	300 30 15	
Time step	DELT	t1c1p2 t1c2p2 t1c3p2	0.33333 (Cr=0.1) 3.3333 (Cr=1.0) 6.6666 (Cr=2.0)	yr
Inflow boundary condition	INBOUN		1	
Darcy velocity	VEL		3.0	m/yr
Dispersivity	DISP	t1c1p1 t1c1p2 t1c1p3	20.0 (Pe=0.1) 100.0 (Pe=0.5) 200.0 (Pe=1.0)	m
Fracture porosity	THETA1		0.20	(-)
Adsorption	KD1		0.0	cm ³ /gm
Chemical decay	GAMMA		0.0	1/yr
Source concentration	C_SRCE		20.0	mg/L

The finite-element grid is composed of 101 nodes, spaced at 50 meters. The Peclet number was varied for three different dispersivities and three time steps were chosen to vary the Courant number. In Figure 8.1, the results for different Peclet numbers shows excellent agreement between the analytical solution and MC_TRANS. When the Peclet exceeds 1.0, the results of MC_TRANS will demonstrate numerical dispersion and deviate from the analytical solution. In Figure 8.2, the results for different Courant numbers demonstrates the effect of the numerical criteria. The courant number should always be less than 1.0 for an accurate solution.

Figure 8.1 Test 1 Verification

Figure 8.2 Test 1 verification.

8.2 Test 2 - Comparison with One-dimensional Transport with Third-Type Boundary Condition

To verify the implementation of the boundary condition, a problem the same as Test 1 was executed for the base values of $Pe=0.50$ and $Cr=0.1$. The analytical solution was first presented by Selim and Mansell, 1976 and again the WEXLER code was used to compare with MC_TRANS. With the third-type boundary condition, the mass flux is specified using the combination of the source concentration, C_{SRCE} , and the Darcy velocity, VEL . For this problem the mass flux is 60 (mg/L)*(m/yr) or 0.06 mg/yr/m^2 .

Parameter Definition	Variable	File	Value	Units
Source concentration	C_{SRCE}	t2c1p2	20.0	mg/L
Inflow boundary condition	INBOUN	t2c1p2	2	

In Figure 8.3, the agreement between the analytical and numerical solution is excellent. In comparing the first-type and third type boundary condition results, one can notice that the first-type boundary condition results in a slightly earlier breakthrough. The effects are more pronounced closer to the source and at earlier times.

Figure 8.3 Test 2 verification.

8.3 Test 3 - Comparison with One-dimensional Transport with Decaying First-Type Boundary Condition and Decaying Chemical

This verification is an extension of Test 1 in which both the source concentration and the chemical are decayed using the same decay constant as listed below

Parameter Definition	Variable	File	Value	Units
Source concentration	C_SRCE		20.0	mg/L
Inflow boundary condition	INBOUN		1	
Chemical decay	GAMMA	t3c1p2	0.0069314	1/yr
Boundary source decay	BETA	t3c1p2	0.0069314	1/yr

In Figure 8.4, the comparison is again excellent and demonstrates the effect of decay on the breakthrough concentration. The decay rate is equivalent to a half-life of 100 years, and expectedly the discharge concentration at 100 years is approximately one-half of the source concentration.

Figure 8.4 Test 3 verification.

8.4 Test 4 - Comparison with One-dimensional Transport of Multiple Species

To verify the implementation of multiple species and adsorption, the results from MC_TRANS are compared with those originally published by Lester, et al, 1975 using the LESTER program contained in the verification package VERTPAK (Intera, 1983).

Parameter Definition	Variable	File	Value	Units
Number of elements	NUMX	t4	1011	
Element length	DELX		6.562	m
Number of time steps	NUMT		200	
Time step	DELT		3000	yr
Inflow boundary condition	INBOUN		1	
Source concentration	C_SRCE-1 C_SRCE-2 C_SRCE-3		1.0 0.0 0.0	mg/L
Source decay	GAMMA-1 GAMMA-2 GAMMA-3		4.39e-6 1.27e-4 2.90e-7	1/yr 1/yr 1/yr
Darcy velocity	VEL		.656	m/yr
Dispersivity	DISP		8.5	m
Fracture porosity	THETA1		0.10	(-)
Adsorption	KD1		935.1	cm ³ /gm
Chemical decay	BETA-1 BETA-2 BETA-3		4.39e-6 1.27e-4 2.90e-7	1/yr 1/yr 1/yr

In Figure 8.5, the comparison with the analytical solution shows very close agreement over six orders of magnitude.

Figure 8.5 Test 4 verification.

8.5 Test 5 - Comparison with One-dimensional Dual Porosity Transport with First-Type Boundary Condition

To verify the dual porosity module, the MCMAIN code is compared with the analytical solution developed by van Genuchten (1981). The input for the two models differs and the footnotes in the data files, helps explain how the two input files for MC_TRANS the analytical solution CFTIM (See Appendix D) were developed. The problem involves a pulse input with diffusion into the matrix or dual porosity region. This results in breakthrough curve with an increasing plateau and extended tail. This results from the matrix blocks being "filled" as the pulse passes by in the fracture. The parameters are chosen such that the matrix blocks not only offer a buffer or sorptive and retarding effect, but are also capacity limited. The agreement between the numeric and analytical codes is excellent (Figure 8.6).

Parameter Definition	Variable	File	Value	Units
Number of elements	NUMX	t5	101	
Element length	DELX		1.0	m
Number of time steps	NUMT		500	
Time step	DELT		0.021	yr
Inflow boundary condition	INBOUN		1	
Darcy velocity	VEL		10.0	m/yr
Dispersivity	DISP		0.78	m
Fracture porosity	THETA1		0.26	(-)
Matrix porosity	THETA2		0.14	(-)
Fracture adsorption	KD1		0.20	cm ³ /gm
Matrix adsorption	KD2		0.30	cm ³ /gm
Chemical decay	GAMMA		0.0	1/yr
Source concentration	C_SRCE		1.0	mg/L

Figure 8.6 Test 5 verification.

8.6 Test 6 - Comparison with One-dimensional Dual Porosity Transport with Third-Type Boundary Condition

Similar to Test 5, the third-type boundary option is tested also using the CFTIM code. As before the comparison is excellent (Figure 8.7). The input change is for INBOUN switching from a value of 1 to 2.

Figure 8.7 Test 6 verification.

8.7 Test 7 - Example Monte Carlo Application

In order to demonstrate the use of MC_TRANS, an example Monte Carlo application was simulated. The problem includes spatial variation in the model parameters subject to fracture and matrix transport. Parameters sampled for uncertainty include the hydraulic gradient using log normal distributions.

Parameter Definition	Variable	File	Value	Units
Number of elements	NUMX	t7a	51	
Element length	DELX		10.0	m
Number of time steps	NUMT		100	
Time step	DELT		3.0	yr
Inflow boundary condition	INBOUN		1	
Darcy velocity	VEL 1-10 VEL 11-51		10.0 120.0	m/yr
Dispersivity	DISP 1-10 DISP 11-51		500.0 1500.0	m m
Fracture porosity	THETA1 1-10 THETA1 11-51		0.10 0.20	(-) (-)
Matrix porosity	THETA2 1-10 THETA2 11-51		0.30 0.40	(-) (-)
Fracture adsorption	KD1 1-10 KD1 11-51		1.5 10.0	cm ³ /gm cm ³ /gm
Matrix adsorption	KD2 1-10 KD2 11-51		1.5 10.0	cm ³ /gm cm ³ /gm
Matrix diffusion	DIFF 1-10 DIFF 11-51		?? ??	? ?
Chemical decay	GAMMA		2.88e-7	1/yr
Source concentration	C_SRCE		7.56e9	mg/L

Figure 8.8. Maximum concentration frequency.

8.8 Test 8 - Example Latin Hypercube Sample Application

In order to demonstrate the use of MC_TRANS, in conjunction with Sandia Laboratories LHS code ((Davenport and Inman, 1982) a set of 200 Latin Hypercube Sample realizations was simulated using MC_TRANS and the with the results compared to results generated using an analytical solution for dual-porosity fracture transport (Sudicky and Frind, 1982 and Sudicky, 1990). Parameters sampled for uncertainty include the dispersivity, effective porosity, matrix porosity, block width, matrix diffusion coefficient and relative recharge rate. Distribution types are noted in the table below.

Parameter Definition	Variable	File	Value	Units
Number of elements	NUMX	t8a	400	
Element length	DELX	t8a	100.0	m
Number of time steps	NUMT	t8a	1000	
Time step	DELT	t8a	10.0	yr
Inflow boundary condition	INBOUN	t8a	1	
Darcy velocity	VEL	t8a	0.1	m/yr
Dispersivity	DISP	t8b	100/500 ¹	m
Effective porosity	THETA1	t8b	-2.44/0.6 ²	(-)
Matrix porosity	THETA2	t8b	0.8/0.36 ¹	(-)
Tortuosity	TAU	t8a	0.1	(-)
Block width	BWIDTH	t8b	0.1/1.5 ¹	m
Matrix diffusion	DIFF	t8b	-2.84/0.35 ²	m ² /y
Chemical decay	GAMMA	t8a	1.21 x 10 ⁻⁴	1/yr
Source concentration	C_SRCE	t8b	4.759/0.5 ²	PiC/L

¹ Uniform distribution (lower-bound/upper-bound)

² Lognormal (base 10) distribution (mean of log/standard deviation of log)

Breakthrough curves comparing the MC_TRANS results and the analytical solution results for a deterministic simulation using the mean parameter values are presented in the first graph below (Figure 8.9a). This allows for a direct comparison of results with the analytical solution. A favorable comparison between analytical and numerical results is found for the deterministic case. To evaluate the models capacity to solve for peak concentrations, peak concentration results for the numerical and analytical models were compared for 200 LHS generated realizations. A comparison of cumulative probability distribution curves for maximum concentration levels at the source and 5 other observation points (1000, 2000, 3000, 4000 and 5000 m) are presented in the second figure (Figure 8.9b). Again, a good match is seen, except at the fringe of the plume.

Figure 8.9a. Concentration breakthrough curves for mean parameters.

Figure 8.9b. Maximum concentration levels for Carbon-14.

9 REFERENCES

- Bear, J., 1972. *Dynamics of Fluids in Porous Media*, American Elsevier, New York, 764 pp.
- Bear, J., 1979. *Hydraulics of Groundwater*, McGraw-Hill, New York.
- Benjamin, J. R. and C. A. Cornell, 1970. *Probability, Statistics and Decision for Civil Engineers*, McGraw-Hill Book Company, New York, NK.
- Coats, K.H., and D.B. Smith, 1964. Dead-end pore volume and dispersion on porous media, *Soc. Pet. Eng. J.* 4: 73-84.
- Davenport, J.M., and R.L. Iman, 1982. An iterative algorithm to produce definite correlation matrix from an "approximate correlation matrix" (with a program user's guide). SAND81-1376.
- Inman, R.L., J.M. Davenport, and D.K. Zeigler, 1980. Latin hypercube sampling (program users guide). SAND79-1473.
- Intera, 1983. VERTPAK-1, Package of Analytical Solutions for Code Verification, *ONWI-451*, prepared for Battelle Memorial Institute, Office of Nuclear Waste Isolation, Columbus, OH.
- Isotok, J., 1989. *Groundwater Modeling by the Finite Element Method*, American Geophysical Union, Water Resources Monograph 13, Washington, DC.
- Freeze, R. A. and J. A. Cherry, 1979. *Groundwater*, Prentice-Hall, Englewood Cliffs, NJ.
- Lapidus, L. and N. R. Amundson, 1952. Mathematics of adsorption in beds, VI. The effects of longitudinal diffusion in ion exchange and chromatographic columns, *J. Phy. Chemistry*, 56:984-988.
- Lester, D. H., G. Jansen and H. C. Burkholder, 1975. Migration of radionuclide chains through adsorbing medium, *AIChE Symposium*, Ser. 71, 152, p. 202.
- Selim, H. M., and R. S. Mansell, 1976. Analytical solutions of the equation for transport of reactive solutes through soils, *Water Resources Research*, 12(3): 528-532.
- Sudicky, E.A., 1990. The Laplace transfer Galerkin technique for efficient time-continuous solution of solute transport in double-porosity media, *Geoderma*, 46:209-232.
- Sudicky, E.A., and E.O. Frind, 1982. Contaminant transport in fractured porous media: analytical solutions for a system of parallel fractures, *Water Resources Research*, Vol. 18, No. 6, pp 1634-1642.

U.S. Environmental Protection Agency, 1993. Multimedia exposure assessment model of evaluating the land disposal of hazardous wastes, Theory I. Prepared by Woodward-Clyde Consultants. EPA Contract No. 68-03-3513 and No. 68-03-6304. Environmental Research Laboratory. U.S. Environmental Protection Agency, Athens, GA.

van Genuchten, M. Th., 1981. Non-equilibrium transport parameters for miscible displacement experiments, US Dept. of Agriculture, *Research Report No. 119*, US Salinity Laboratory, Riverside, CA.

van Genuchten, M.Th., 1976. Mass transfer studies in sorbing porous media, 1: Analytical Solutions, *Soil Sci. Soc. Amer.*, 40(4):473-480.

Wexler, E. J., 1989. Analytical solutions for one-, two-, and three-dimensional solute transport in groundwater-water systems with uniform flow, US Geological Survey, *Open-file Report 89-56*.

APPENDIX A

MC_TRANS FORTRAN SOURCE CODE LISTING

C		MAIN0010	C	5) Simulate the breakthrough curve at all receptor locations and	MAIN0560
C	*****	MAIN0020	C	all species in the decay chain for the realization.	MAIN0570
C	MONTE CARLO ADVECTIVE-DISPERSIVE TRANSPORT OF 4-MEMBER	MAIN0030	C	6) Write out the BTC at each receptor for each specie and for each	MAIN0580
C	DECAY CHAIN WITH STEADY-STATE DUAL POROSITY	MAIN0040	C	realization to a binary file for post-processing.	MAIN0590
C	*****	MAIN0050	C		MAIN0600
C		MAIN0060	C		MAIN0610
C	Name: MC_TRANS	MAIN0070	C		MAIN0620
C	Version: 5.0 Flow Parameter Sensitivity Model	MAIN0080	C	INCLUDE 'MC_TRLHS.INC'	MAIN0630
C		MAIN0090	C		MAIN0640
C	GeoTrans, Inc.	MAIN0100	C	DOUBLE PRECISION RNUM (1)	MAIN0650
C	4330 S. Valley View Lane, Suite 114	MAIN0110	C	CHARACTER*72 TITLE	MAIN0660
C	Las Vegas, NV 89103	MAIN0120	C	CHARACTER*8 FNAME	MAIN0670
C	Ken Rehfeldt, Rick Waddell 702-737-7225	MAIN0130	C	CHARACTER*12 FNAMES	MAIN0680
C		MAIN0140	C		
C	GeoTrans, Inc.	MAIN0150	C	ckrr 3/26/96	
C	4888 Pearl East Circle	MAIN0160	C	character*2 cvar	
C	Suite 300-E	MAIN0170	C	character*1 k1,k2	
C	Boulder, CO 80301	MAIN0180	C	ckrr	
C	Debbie Cave, Chen Zhu 303-440-4556	MAIN0190	C	LOGICAL LEX	MAIN0690
C		MAIN0200	C		MAIN0700
C	GeoTrans, Inc.	MAIN0210	C	Primary arrays and matrices:	MAIN0710
C	46050 Manekin Place	MAIN0220	C		MAIN0720
C	Suite 100	MAIN0230	C		MAIN0730
C	Sterling, VA 20166	MAIN0240	C	C**PSTAT(,) = Array to hold statistic information on each parameter	MAIN0740
C	David Ward, Barry Lester 703-444-7000	MAIN0250	C	distribution:	MAIN0750
C		MAIN0260	C		MAIN0760
C	November 9, 1995	MAIN0270	C	PSTAT(I,1) = Type of distribution for parameter I	MAIN0770
C		MAIN0280	C	= 0 Deterministic (characterized by the mean)	MAIN0780
C	Modified November, 1995 to incorporate flow model sensitivities		C	= 1 Normal distribution (mean and std)	MAIN0790
C	into the Monte Carlo analysis. The sensitivities of flux to		C	= 2 Lognormal distribution (mean and std)	MAIN0800
C	modeling parameters are read in and used to calculate new		C	= 3 Uniform distribution (max and min)	MAIN0810
C	values of specific discharge, as a function of samples of flow		C	= 4 Raw data distribution [** untested **]	MAIN0820
C	model parameters, assuming a first-order response. The		C		MAIN0830
C	modification was developed by Barry Lester and Rick Waddell.		C	PSTAT(I,2) = Mean value for the parameter I; if a uniform	MAIN0840
C			C	distribution, then this is the minimum parameter	MAIN0850
C			C	value.	MAIN0860
C	Modified March, 1996 to incorporate sensitivity analyses on the		C		MAIN0870
C	input parameters including the flow parameters. A list of the		C	PSTAT(I,3) = Standard deviation for the parameter I; if a uniform	MAIN0880
C	parameters to be varied will be read in and each of those		C	distribution, then this is the max parameter value.	MAIN0890
C	parameters will be allowed to be variable one at a time.		C		MAIN0900
C		MAIN0290	C	PSTAT(I,4) = Number of data points in the raw data distribution;	MAIN0910
C	Funding: DOE NTS Project Number 3705-491.	MAIN0300	C	Non-zero only for PSTAT(I,1)=4.	MAIN0920
C		MAIN0310	C		MAIN0930
C	Equations: Advective-dispersive multicomponent transport.	MAIN0320	C		MAIN0940
C		MAIN0330	C	C**PDIST(,) = The matrix for storing the raw distribution data;	MAIN0950
C	Dimensionality: One-dimensional.	MAIN0340	C	PDIST(I,K) = Data point K for parameter i; K=1,...,PSTAT(I,4).	MAIN0960
C		MAIN0350	C		MAIN0970
C	Solution Procedure: Galerkin Finite element.	MAIN0360	C	C**PAR(IMC) = Vector of parameter values for realization IMC.	MAIN0980
C		MAIN0370	C		MAIN0990
C	Decay chain for up to four members.	MAIN0380	C		MAIN1000
C		MAIN0390	C	C**BTC(IO,N,IS) = Breakthrough concentration at receptor IO, at time N,	MAIN1010
C	Dual-porosity transport using steady-state mass transfer	MAIN0400	C	for species IS.	MAIN1020
C	between fracture and matrix (also referred to	MAIN0410	C		MAIN1030
C	as mobile and immobile zones).	MAIN0420	C		MAIN1040
C		MAIN0430	C	C**CMAX(,) = Maximum breakthrough concentration at the receptors.	MAIN1050
C		MAIN0440	C	CMAX(IO) = Max concentration at receptor IO.	MAIN1060
C		MAIN0450	C		MAIN1070
C		MAIN0460	C	C**CTIM(,) = Time of arrival of maximum breakthrough concentration	MAIN1080
C	MAIN Program functions:	MAIN0470	C	at the receptors.	MAIN1090
C		MAIN0480	C		MAIN1100
C	1) Read in the paramters for the simulation.	MAIN0490	C	CTIM(IO) = Time of arrival of max concentration at receptor IO.	MAIN1110
C	2) Calculate statistics, as necessary, for the parameter	MAIN0500	C		MAIN1120
C	distributions.	MAIN0510	C		MAIN1130
C	3) Calculate the BTC's for parent and daughter species	MAIN0520	C		MAIN1140
C	at the receptors using the mean values.	MAIN0530	C		MAIN1150
C	4) Generate random numbers to randomly select parameters from	MAIN0540	C	C**NOBSLOC(IO) = Finite element node number of receptor IO.	
C	each distribution, e.g. create a realization.	MAIN0550			

C		MAIN1160	C		MAIN1800
C		MAIN1170		INQUIRE (FILE = FNAMES, EXIST = LEX)	MAIN1810
C	-----	MAIN1180	C		MAIN1820
C	-----	MAIN1190		IF (.NOT. LEX) THEN	MAIN1830
C		MAIN1200		WRITE (*, 6010) FNAMES	MAIN1840
C		MAIN1210	6010	FORMAT(' Input data file ',A12,' does not exist',/,/,	MAIN1850
C	Open UNIT 5 for input and the appropriate units for output.	MAIN1220	1	' Re-enter filename: ', \$)	MAIN1860
C		MAIN1230		ITRY = ITRY + 1	MAIN1870
C		MAIN1240		IF (ITRY .EQ. 5) STOP	MAIN1880
C	Files are as follows:	MAIN1250		GO TO 1000	MAIN1890
C		MAIN1260		END IF	MAIN1900
C	UNIT 5 = Input data file.	MAIN1270	C		MAIN1910
C		MAIN1280		OPEN (5, FILE = FNAMES, STATUS = 'OLD')	MAIN1920
C	UNIT 6 = Output of input data.	MAIN1290	C		MAIN1930
C		MAIN1300		FNAMES = FNAME//'.OUT'	MAIN1940
C	UNIT 7 = Output of maximum breakthrough concentration, at each	MAIN1310		OPEN (6, FILE = FNAMES)	MAIN1950
C	receptor, for each realization.	MAIN1320	C		MAIN1960
C		MAIN1330		FNAMES = FNAME//'.CMX'	MAIN1970
C	UNIT 8 = Output of the time of arrival of the maximum breakthrough	MAIN1340		OPEN (7, FILE = FNAMES)	MAIN1980
C	concentration, at each receptor, for each realization.	MAIN1350	C		MAIN1990
C		MAIN1360		FNAMES = FNAME//'.ATM'	MAIN2000
C	UNIT 9 = Summary of means and standard deviations of the risk,	MAIN1370		OPEN (8, FILE = FNAMES)	MAIN2010
C	max. arrival time, and max arrival concentration at each	MAIN1380	C		MAIN2020
C	receptor location. [Not used at this time]	MAIN1390		FNAMES = FNAME//'.SMX'	MAIN2030
C		MAIN1400		OPEN (9, FILE = FNAMES)	MAIN2040
C	UNIT 10= Breakthrough concentrations at each receptor location	MAIN1410	C		MAIN2050
C	using the mean parameter values; this file is closed	MAIN1420		FNAMES = FNAME//'.BTC'	MAIN2060
C	after the values are printed and thus can be used	MAIN1430		OPEN (10, FILE = FNAMES)	MAIN2070
C	to examine the mean characteristics of the transport	MAIN1440	C		MAIN2080
C	model while the Monte Carlo run is executing.	MAIN1450		FNAMES = FNAME//'.PAR'	MAIN2090
C		MAIN1460		OPEN (11, FILE = FNAMES)	MAIN2100
C	UNIT 11= Parameter values used for each realization.	MAIN1470	C		MAIN2110
C		MAIN1480		FNAMES = FNAME//'.TXT'	MAIN2120
C	UNIT 12= Tabular output of maximum concentration at receptor	MAIN1490		OPEN (12, FILE = FNAMES)	MAIN2130
C	location for import into spreadsheet. This includes	MAIN1500	C		MAIN2140
C	a header record for legend or field name.	MAIN1510		FNAMES = FNAME//'.BIN'	MAIN2150
C		MAIN1520		OPEN (20, FILE = FNAMES, FORM = 'UNFORMATTED')	MAIN2160
C	UNIT 20= Breakthrough curve summary {Binary Format}.	MAIN1530	C		MAIN2170
C		MAIN1540	C		MAIN2180
C		MAIN1550		C Read in the input data from UNIT 5.	MAIN2190
C		MAIN1560	C		MAIN2200
C		MAIN1570	C	+++++	MAIN2210
C		MAIN1580		CALL MC_INP (TITLE)	MAIN2220
C	WRITE (*, 6000)	MAIN1590	C	+++++	MAIN2230
C	6000 FORMAT(/////,	MAIN1600	C		MAIN2240
C	1 ' ++++++',/,/,	MAIN1610			
C	2 ' MC_MAIN Version 5.0 9/27/95',/,/,	MAIN1620	C	RKW Nov. 1995 modification	
C	3 ' ++++++',/,/,	MAIN1630	C		
C	4 ' Enter name of the primary input file',/,	MAIN1640		C Read in the input data flow sensitivity parameters.	
C	5 ' (do not include the .DAT suffix)',/,/,T20,' ', \$)	MAIN1650	C		
C		MAIN1660	C	+++++	
C		MAIN1670		IF (NRC.LT.1) THEN	
C	ITRY = 0	MAIN1680		NFSPP=NFSPP	
C	1000 CONTINUE	MAIN1690		ELSE	
C	READ (*, '(A8)') FNAME	MAIN1700		NFSPP=NFSPP-1	
C		MAIN1710		ENDIF	
C	Clean-up (trim and convert to uppercase) file name.	MAIN1720		IF (NFSPP .GT. 0) THEN	
C		MAIN1730		WRITE (*, 6001)	
C	CALL UPCASE (FNAME)	MAIN1740	6001	FORMAT(/////,	
C	CALL LENTRM (FNAME, K)	MAIN1750	1	' Enter name of the primary flow sensitivity ',/,/,	
C		MAIN1760	2	' input files (do not include the suffixes)',	
C	CConcatonate file suffixes (predefined).	MAIN1770	3	/,/,T20,' ', \$)	
C		MAIN1780	C		
C	FNAMES = FNAME (1:K)//'.DAT'	MAIN1790	C		
				ITRY = 0	

```

1001  CONTINUE
      READ (*, '(A8)') FNAME
C
C Clean-up (trim and convert to uppercase) file name.
C
      CALL UPCASE (FNAME)
      CALL LENTRM (FNAME, K)
C
C Concatonate file suffixes (predefined).
C
      FNames = FNAME (1:K)//'.VEL'
C
      INQUIRE (FILE = FNames, EXIST = LEX)
C
      IF (.NOT. LEX) THEN
        WRITE (*, 6011) FNames
6011    FORMAT(' velocity data file ',A12,' does not exist',//,
1        ' Re-enter filename: ', $)
        ITRY = ITRY + 1
        IF (ITRY .EQ. 5) STOP
        GO TO 1001
      END IF
C
      OPEN (13, FILE = FNames, STATUS = 'OLD', FORM = 'UNFORMATTED')
C
C
C Concatonate file suffixes (predefined).
C
      FNames = FNAME (1:K)//'.SNX'
C
      INQUIRE (FILE = FNames, EXIST = LEX)
C
      IF (.NOT. LEX) THEN
        WRITE (*, 6012) FNames
6012    FORMAT(' X sensitivity data file ',A12,' does not exist',$)
        STOP
      END IF
C
      OPEN (14, FILE = FNames, STATUS = 'OLD', FORM = 'UNFORMATTED')
C
C
C
      FNames = FNAME (1:K)//'.SNY'
C
      INQUIRE (FILE = FNames, EXIST = LEX)
C
      IF (.NOT. LEX) THEN
        WRITE (*, 6013) FNames
6013    FORMAT(' Y sensitivity data file ',A12,' does not exist',$)
        STOP
      END IF
C
      OPEN (15, FILE = FNames, STATUS = 'OLD', FORM = 'UNFORMATTED')
C
C
C
      FNames = FNAME (1:K)//'.SNZ'
C
      INQUIRE (FILE = FNames, EXIST = LEX)
C

```

```

      IF (.NOT. LEX) THEN
        WRITE (*, 6014) FNames
6014    FORMAT(' Z sensitivity data file ',A12,' does not exist',$)
      STOP
    END IF
C
      OPEN (16, FILE = FNames, STATUS = 'OLD', FORM = 'UNFORMATTED')
C
      CALL SEN_INP
C
      END IF
C
      *****
CRKW END
C
C Write the simulation title out to most output units.
C Do not include spreadsheet or binary files.
C
      IF (MC .GT. 1) WRITE (7, '(A72, //)') TITLE
      IF (MC .GT. 1) WRITE (8, '(A72, //)') TITLE
      IF (MC .GT. 1) WRITE (9, '(A72, //)') TITLE
      WRITE (10, '(A72, //)') TITLE
      IF (MC .GT. 1) WRITE (11, '(A72, //)') TITLE
C
C Check the parameter distributions; sort and calc mean as necessary.
C
      *****
      CALL P_CHECK (PSTAT, PDIST, NPARDIM, NDMAX, NPARD)
      *****
C
C Generate random numbers and select parameters for the
C realization; store the parameters in the vector PAR().
C
C First initialize the random number generator with the seed value.
C
ckrr
      iseed = iseed
ckrr
C
      *****
      CALL MRAND (ISEED, 1, RNUM)
      *****
C
      ISEED = 0
C
C All subsequent calls will be made with a seed of 0.
C
      WRITE (*, ' (//) ')
C
      IF (MC .EQ. 0) MC = 1
      IBTC = 0
      WRITE (*, *)
1      ' CALCULATING BREAKTHROUGH CONCENTRATIONS USING MEAN',
2      ' PARAMETER VALUES.....'
      WRITE (*, *) ' '
C
C Write parameter identification index before starting Monte Carlo.
C
      IF (IOU11 .GE. 1 .AND. MC .GT. 1) THEN
        WRITE (11, 6020) (IPARID (I1, 2), IPARID
1          (I1, 1), I1 = 1, NPARD)
6020    FORMAT('IPARID(IS)', 100('ID(', I1, ')=' , I3, 1X))
      END IF
C

```

```

MAIN2250
MAIN2260
MAIN2270
MAIN2280
MAIN2290
MAIN2300
MAIN2310
MAIN2320
MAIN2330
MAIN2340
MAIN2350
MAIN2360
MAIN2370
MAIN2380
MAIN2390
MAIN2400
MAIN2410
MAIN2420
MAIN2430
MAIN2440
MAIN2450
MAIN2460
MAIN2470
MAIN2480
MAIN2490
MAIN2500
MAIN2510
MAIN2520
MAIN2530
MAIN2540
MAIN2550
MAIN2560
MAIN2570
MAIN2580
MAIN2590
MAIN2600
MAIN2610
MAIN2620
MAIN2630
MAIN2640
MAIN2650
MAIN2660
MAIN2670
MAIN2680
MAIN2690

```

```

C Write header record for spreadsheet/database field names.      MAIN2700
C                                                                MAIN2710
C      WRITE (12, 6030)                                          MAIN2720
C      6030 FORMAT(' NOBSLOC()', ' LOCATION', ' ISPEC', ' IMC',  MAIN2730
C      1      ' MAX_TIM', ' MAX_CONC')                          MAIN2740
C                                                                MAIN2750
ckrr
c additions to the code to allow for a series of sensitivity runs
c
      isens = 0
1005  continue
      if (nsens.lt.1) then
        do 1004 ip = 1,npard
          iflag(ip) = 1
1004  continue
        else
          isens = isens + 1
          ibtc = 0
c initialize the random number generator with the same seed
          call mrand (iseeds, 1, rnum)
          iseed = 0
          write (cvar,'(i2)') isens
          read (cvar,'(2a1)') k1,k2
          if (k1.eq.' ') k1 = '0'
          fnames = fname(4:k)//'_'//k1//k2//'.txt'
          close (12)
          open (12, file=fnames)
          fnames = fname(4:k)//'_'//k1//k2//'.out'
          close (6)
          open (6,file=fnames)
          fnames = fname(1:k)//'.btc'
          open (10,file=fnames)
          read (21,*) nparsens
          read (21,*) (ipar(j),j=1,nparsens)
          write (6,6080) isens,(ipar(j),j=1,nparsens)
          do 1006 ip = 1,npard
            iflag(ip) = 0
            do 1007 j=1,nparsens
              if (ip.eq.ipar(j)) iflag(ip) = 1
1007          continue
1006          continue
6080          format(' THE FOLLOWING PARAMETERS WERE ALLOWED TO VARY',/,
&          ' DURING SENSITIVITY RUN',i5,/,10(4x,i5))
          endif
ckrr
C                                                                MAIN2760
C                                                                MAIN2770
C                                                                MAIN2780
C ===== MAIN2790
C Start of Monte Carlo loop. MAIN2800
C ===== MAIN2810
C                                                                MAIN2820
C      DO 1010 IMC = 1, MC MAIN2830
C                                                                MAIN2840
C      1020 CONTINUE MAIN2850
C                                                                MAIN2860
C IF MC=1 (e.g. deterministic) then get mean values only and put MAIN2870
C into PAR() else get random parameter values; if MC>1, then MAIN2880
C first do a run with the mean parameters and store the BTC in UNIT 10MAIN2890
C                                                                MAIN2900
C First convert mean of the log normal distribution to antilog MAIN2910
C                                                                MAIN2920
C      IF (MC .EQ. 1) THEN MAIN2930
C        DO 1030 IP = 1, NPARD MAIN2940
C          Uniform Distribution MAIN2950
C          IF (NINT (PSTAT (IP, 1)) .EQ. 3) THEN MAIN2960
C
C          PAR (IP) = 0.5D0*(PSTAT (IP, 2) + PSTAT (IP, 3)) MAIN2970
C          Lognormal Distribution MAIN2980
C          ELSE IF (NINT (PSTAT (IP, 1)) .EQ. 2) THEN MAIN2990
C            AVGNEW = PSTAT (IP, 2) MAIN3000
C            SIGNW = PSTAT (IP, 3) MAIN3010
C            VAROLD = LOG (1 + ((SIGNW*SIGNW)/(AVGNEW*AVGNEW))) MAIN3020
C            AVGOLD = LOG (AVGNEW) - 0.5*VAROLD MAIN3030
C            PAR (IP) = EXP (AVGOLD) MAIN3040
C          ELSE MAIN3050
C            Deterministic (Constant value), Lognormal, MAIN3060
C            Normal or user-defined distribution MAIN3070
C            PAR (IP) = PSTAT (IP, 2) MAIN3080
C          END IF MAIN3090
1030  CONTINUE MAIN3100
C                                                                MAIN3110
C          ELSE MAIN3120
C            IF (IBTC .EQ. 0) THEN MAIN3130
C              DO 1040 IP = 1, NPARD MAIN3140
C                IDIST = NINT (PSTAT (IP, 1))
C                IF (IDIST .EQ. 3) THEN MAIN3150
C                  PAR (IP) = 0.5D0*(PSTAT (IP, 2) + PSTAT (IP, 3)) MAIN3160
C                  Lognormal Distribution MAIN2980
C                  ELSE IF (NINT (PSTAT (IP, 1)) .EQ. 2) THEN
C                    AVGNEW = PSTAT (IP, 2)
C                    SIGNW = PSTAT (IP, 3)
C                    VAROLD = LOG (1 + ((SIGNW*SIGNW)/(AVGNEW*AVGNEW)))
C                    AVGOLD = LOG (AVGNEW) - 0.5*VAROLD
C                    PAR (IP) = EXP (AVGOLD)
CBHL
C                    ELSEIF (IDIST .GE. 0) THEN
C                      PAR (IP) = PSTAT (IP, 2) MAIN3180
C                    ELSE
C                      PAR (IP) = PLHS(1, IABS(IDIST))
C                    END IF MAIN3190
1040  CONTINUE MAIN3200
C          ELSE MAIN3210
C            DO 1050 IP = 1, NPARD MAIN3220
ckrr
C            if(iflag(ip).eq.0) go to 1050
ckrr
C            IDIST = NINT (PSTAT (IP, 1)) MAIN3230
C            IF (IDIST .EQ. 4) THEN MAIN3240
C              NDIST = NINT (PSTAT (IP, 4)) MAIN3250
C              CALL G_PAR2 (PDIST, IP, NPARDIM, MAIN3260
C              NDMAX, NDIST, RNUM, PAR (IP)) MAIN3270
C              ***** MAIN3280
C              ***** MAIN3290
C            ELSE IF (IDIST .EQ. 0) THEN MAIN3300
C              PAR (IP) = PSTAT (IP, 2) MAIN3310
CBHL
C            ELSE IF (IDIST .GT. 0) THEN MAIN3320
C              ***** MAIN3330
C              CALL G_PAR1 (PSTAT, IP, NPARDIM, RNUM, PAR (IP)) MAIN3340
C              ***** MAIN3350
C            ELSE
C              *****
C              PAR (IP) = PLHS(IMC+1, IABS(IDIST))
C              *****
C            END IF MAIN3360
CBHL
1050  CONTINUE MAIN3370
C                                                                MAIN3380
C Write out the realization number to the screen so that the user knows MAIN3390

```

C that the program is still executing.	MAIN3400	C	MAXT0030
C	MAIN3410	SUBROUTINE MAX_TIM (IMC, IBTC, ISPEC)	MAXT0040
WRITE (*, *) ' CALCULATING REALIZATION #', IMC	MAIN3420	C	MAXT0050
ckrr		INCLUDE 'MC_TRLHS.INC'	MAXT0060
write (*, *) ' sensitivity run #', isens		C	MAXT0070
ckrr		C Find the maximum value and records its time step number.	MAXT0080
C	MAIN3430	C	MAXT0090
C Write the parameters used for this realization, and the realization #	MAIN3440	IF (MC .EQ. 1 .OR. IBTC .EQ. 0) GO TO 1000	MAXT0100
C to UNIT 11.	MAIN3450	IF (IMC .EQ. 1 .AND. ISPEC .EQ. 1) THEN	MAXT0110
C	MAIN3460	WRITE (8, *) ' TIME OF MAXIMUM CONCENTRATION ARRIVAL AT ',	MAXT0120
IF (IOU11 .GE. 1) THEN	MAIN3470	1 'RECEPTOR LOCATIONS'	MAXT0130
IF (MOD (IMC, IOU11) .EQ. 0) THEN	MAIN3480	WRITE (7, *) ' MAXIMUM SOLUTE CONCENTRATIONS AT RECEPTOR ',	MAXT0140
WRITE (11, 6040) IMC, (PAR (IP), IP = 1, NPARD)	MAIN3490	1 'LOCATIONS'	MAXT0150
FORMAT(I10,1P,(100G10.3))	MAIN3500	WRITE (8, *) ' '	MAXT0160
END IF	MAIN3510	WRITE (8, *) ' '	MAXT0170
END IF	MAIN3520	WRITE (7, *) ' '	MAXT0180
C	MAIN3530	WRITE (7, *) '# OF OBSERVATION [RECEPTOR] LOCATIONS = '	MAXT0190
END IF	MAIN3540	WRITE (7, *) NUMOBS	MAXT0200
END IF	MAIN3550	WRITE (7, *) ' '	MAXT0210
C	MAIN3560	WRITE (8, 6000) (NOBSLOC (IO), IO = 1, NUMOBS)	MAXT0220
C Run the FE transport model to generate the BTC(s) at the receptor(s).	MAIN3570	WRITE (7, 6000) (NOBSLOC (IO), IO = 1, NUMOBS)	MAXT0230
C	MAIN3580	WRITE (8, 6010) (DELX*(NOBSLOC (IO) - 1), IO = 1, NUMOBS)	MAXT0240
*****	MAIN3590	WRITE (7, 6010) (DELX*(NOBSLOC (IO) - 1), IO = 1, NUMOBS)	MAXT0250
CALL FE_CALL (IMC, IBTC)	MAIN3600	WRITE (8, *) ' '	MAXT0260
C	MAIN3610	WRITE (7, *) ' '	MAXT0270
*****	MAIN3620	6000 FORMAT(20X,' NODE # = ',10(I5,12X))	MAXT0280
IF (IBTC .EQ. 0 .AND. IOU6 .GE. 1) THEN	MAIN3630	6010 FORMAT(' REAL. # ', ' SPEC. # ',	MAXT0290
C	MAIN3640	1 ' X = ',10(1P,G10.4,7X))	MAXT0300
*****	MAIN3650	8000 FORMAT(0P,I10,I10,10X,10(2X,1P,G15.7E3))	MAXT0310
CALL PRN_ELE	MAIN3660	END IF	MAXT0320
CALL PRN_SPE	MAIN3670	C	MAXT0330
CALL PRN_OTH	MAIN3680	1001 DO 1010 IO = 1, NUMOBS	MAXT0340
*****	MAIN3690	VALMAX = - 1.0D20	MAXT0350
END IF	MAIN3700	DO 1020 IT = 1, NUMT	MAXT0360
C	MAIN3710	IF (BTC (IO, IT, ISPEC) .GT. VALMAX) THEN	MAXT0370
C If this is the first call of a stochastic run, only the BTC(s)	MAIN3720	VALMAX = BTC (IO, IT, ISPEC)	MAXT0380
C at the receptors are calculated using the mean values; so need	MAIN3730	IVAL = IT	MAXT0390
C to re-start the stochastic run with IMC=1 and start the random	MAIN3740	END IF	MAXT0400
C selection of parameters.	MAIN3750	1020 CONTINUE	MAXT0410
C	MAIN3760	CMAX (IO) = VALMAX	MAXT0420
IF (IBTC .EQ. 0 .AND. MC .NE. 1) THEN	MAIN3770	CTIM (IO) = FLOAT (IVAL - 1)*DELT	MAXT0430
IBTC = 1	MAIN3780	1010 CONTINUE	MAXT0440
GO TO 1020	MAIN3790	C	MAXT0450
END IF	MAIN3800	IF (IOU8 .GE. 1) THEN	MAXT0460
C	MAIN3810	IF (MOD (IMC, IOU8) .EQ. 0) THEN	MAXT0470
C =====	MAIN3820	WRITE (8, 8000) IMC, ISPEC, (CTIM (IO), IO = 1, NUMOBS)	MAXT0480
C End of Monte Carlo loop; repeat if IMC<MC.	MAIN3830	END IF	MAXT0490
C =====	MAIN3840	END IF	MAXT0500
C	MAIN3850	C	MAXT0510
1010 CONTINUE	MAIN3860	IF (IOU7 .GE. 1) THEN	MAXT0520
C	MAIN3870	IF (MOD (IMC, IOU7) .EQ. 0) THEN	MAXT0530
WRITE (6, 6050) IMC - 1,isens	MAIN3880	WRITE (7, 8000) IMC, ISPEC, (CMAX (IO), IO = 1, NUMOBS)	MAXT0540
ckrr the third line of statement 6050 is added	MAIN3890	END IF	MAXT0550
6050 FORMAT('/', ' MC_MAIN simulation has completed ',	MAIN3900	END IF	MAXT0560
1 I6, ' realizations -----',/,	MAIN3910	C	MAXT0570
& ' ***** sensitivity run number ',i6)		1000 CONTINUE	MAXT0580
if (isens.lt.nsens) go to 1005		C	MAXT0590
if (nsens.gt.0) close (21)		C If MC=1 (e.g. Deterministic) or IBTC=0 write out the BTCS	MAXT0600
ckrr		C at the NUMOBS locations to UNIT 10.	MAXT0610
C	MAIN3920	C	MAXT0620
STOP	MAIN3930	IF (MC .EQ. 1 .OR. IBTC .EQ. 0) THEN	MAXT0630
END	MAIN3940	WRITE (10, *) ' SOLUTE CONCENTRATIONS AT RECEPTOR LOCATIONS'	MAXT0640
C	MAXT0010	WRITE (10, *) ' SPECIE NUMBER ', ISPEC	MAXT0650
C	MAXT0020	WRITE (10, *) ' '	MAXT0660
		WRITE (10, 6020) (NOBSLOC (IO), IO = 1, NUMOBS)	MAXT0670

WRITE (10, 6030) (DELX*(NOBSLOC (IO) - 1), IO = 1, NUMOBS)	MAXT0680	ELSE IF (IPARID (IP, 1) .EQ. 3) THEN	FECA0270
WRITE (10, *) ' '	MAXT0690	DO 1010 IX = IPARID (IP, 3), IPARID (IP, 4)	FECA0280
T = 0.0D0	MAXT0700	CINIT (IX, IS) = PAR (IP)	FECA0290
IF (IOU10 .GE. 1) THEN	MAXT0710	CONTINUE	FECA0300
DO 1030 IT = 1, NUMT	MAXT0720	ELSE IF (IPARID (IP, 1) .EQ. 4) THEN	FECA0310
Include first and last time step.	MAXT0730	DO 1020 IX = IPARID (IP, 3), IPARID (IP, 4)	FECA0320
IF (MOD (IT, IOU10) .EQ. 0 .OR. IT .EQ. 1	MAXT0740	SINIT (IX, IS) = PAR (IP)	FECA0330
1 .OR. IT .EQ. NUMT) THEN	MAXT0750	CONTINUE	FECA0340
WRITE (10, 6040) T,	MAXT0760	ELSE IF (IPARID (IP, 1) .EQ. 5) THEN	FECA0350
1 (BTC (IO, IT, ISPEC), IO = 1, NUMOBS)	MAXT0770	DO 1030 IX = IPARID (IP, 3), IPARID (IP, 4)	FECA0360
END IF	MAXT0780	GRAD (IX) = PAR (IP)	FECA0370
T = T + DELT	MAXT0790	CONTINUE	FECA0380
1030 CONTINUE	MAXT0800	ELSE IF (IPARID (IP, 1) .EQ. 6) THEN	FECA0390
END IF	MAXT0810	DO 1040 IX = IPARID (IP, 3), IPARID (IP, 4)	FECA0400
WRITE (10, *) ' '	MAXT0820	DISPL (IX) = PAR (IP)	FECA0410
WRITE (10, *) ' '	MAXT0830	CONTINUE	FECA0420
IF (ISPEC .EQ. NSPEC) THEN	MAXT0840	ELSE IF (IPARID (IP, 1) .EQ. 7) THEN	FECA0430
CLOSE (10)	MAXT0850	DM = PAR (IP)	FECA0440
END IF	MAXT0860	ELSE IF (IPARID (IP, 1) .EQ. 8) THEN	FECA0450
END IF	MAXT0870	DO 1050 IX = IPARID (IP, 3), IPARID (IP, 4)	FECA0460
C	MAXT0880	TAU (IX) = PAR (IP)	FECA0470
6020 FORMAT(10X, ' NODE # = ', 10(I5,12X))	MAXT0890	CONTINUE	FECA0480
6030 FORMAT(' TIME ', ' X = ', 10(1P,G10.4,7X))	MAXT0900	ELSE IF (IPARID (IP, 1) .EQ. 9) THEN	FECA0490
6040 FORMAT(1P,G12.4,8X,10(2X,G15.7E3))	MAXT0910	DO 1060 IX = IPARID (IP, 3), IPARID (IP, 4)	FECA0500
C	MAXT0920	THETA1 (IX) = PAR (IP)	FECA0510
C Output tabular/spreadsheet values.	MAXT0930	CONTINUE	FECA0520
C	MAXT0940	ELSE IF (IPARID (IP, 1) .EQ. 10) THEN	FECA0530
IF (MC .GE. 1 .AND. IBTC .NE. 0 .AND. IOU12 .GE. 1) THEN	MAXT0950	DO 1070 IX = IPARID (IP, 3), IPARID (IP, 4)	FECA0540
IF (MOD (IMC, IOU12) .EQ. 0) THEN	MAXT0960	THETA2 (IX) = PAR (IP)	FECA0550
DO 1040 IO = 1, NUMOBS	MAXT0970	CONTINUE	FECA0560
WRITE (12, 6050) NOBSLOC (IO), DELX*(NOBSLOC (IO) - 1),	MAXT0980	ELSE IF (IPARID (IP, 1) .EQ. 11) THEN	FECA0570
1 ISPEC, IMC, CTIM (IO), CMAX (IO)	MAXT0990	DO 1080 IX = IPARID (IP, 3), IPARID (IP, 4)	FECA0580
6050 FORMAT(I10,1P,E15.7,0P,I10,I10,1P,E15.5E3,E15.5E3)	MAXT1000	CRKW Nov. 1995 Modification to allow fracture implementation of	
1040 CONTINUE	MAXT1010	C alpha with dual porosity model.	
END IF	MAXT1020	BWIDTH (IX) = PAR (IP)	FECA0590
END IF	MAXT1030	BWIDTHF (IX) = PAR (IP)	
C	MAXT1040	BWIDTH (IX) = 2.	
RETURN	MAXT1050	C RKW End	
END	MAXT1060	1080 CONTINUE	FECA0600
	FECA0010	ELSE IF (IPARID (IP, 1) .EQ. 12) THEN	FECA0610
	FECA0020	DO 1090 IX = IPARID (IP, 3), IPARID (IP, 4)	FECA0620
	FECA0030	HYCOND (IX) = PAR (IP)	FECA0630
SUBROUTINE FE_CALL (IMC, IBTC)	FECA0040	CONTINUE	FECA0640
	FECA0050	ELSE IF (IPARID (IP, 1) .EQ. 13) THEN	FECA0650
INCLUDE 'MR_TRLHS.INC'	FECA0060	DO 1100 IX = IPARID (IP, 3), IPARID (IP, 4)	FECA0660
	FECA0070	RHOB (IX) = PAR (IP)	FECA0670
	FECA0080	CONTINUE	FECA0680
	FECA0090	ELSE IF (IPARID (IP, 1) .EQ. 14) THEN	FECA0690
For the parent (ISPEC=1), the	FECA0100	DO 1110 IX = IPARID (IP, 3), IPARID (IP, 4)	FECA0700
c parameters must be input for each of the NPARID (=17) required input	FECA0110	RKD1 (IX, IS) = PAR (IP)	FECA0710
c parameters. For all daughters (ISPEC=2, NSPEC), input is only needed	FECA0120	CONTINUE	FECA0720
c for PARID=1, 2, 3, 4, 14, 15, 16, 17.	FECA0130	ELSE IF (IPARID (IP, 1) .EQ. 15) THEN	FECA0730
	FECA0140	DO 1120 IX = IPARID (IP, 3), IPARID (IP, 4)	FECA0740
	FECA0150	RKD2 (IX, IS) = PAR (IP)	FECA0750
Initialize FE transport model parameter zones to the values in the	FECA0160	CONTINUE	FECA0760
PAR() vector according to the type and locations in IPARID(,).	FECA0170	ELSE IF (IPARID (IP, 1) .EQ. 16) THEN	FECA0770
	FECA0180	DO 1130 IX = IPARID (IP, 3), IPARID (IP, 4)	FECA0780
	FECA0190	ALPHA (IX, IS) = PAR (IP)	FECA0790
NUMXM1 = NUMX - 1	FECA0200	CRKW NOV 17, 1995 CHANGE TO ALLOW BWIDTH TO BE FRACTURE SPACING	
	FECA0210	CRKW IN DUAL-POROSITY MODE, RATHER THAN SETTING BWIDTH TO 2.0	
DO 1000 IP = 1, NPARD	FECA0220		
IS = IPARID (IP, 2)	FECA0230	1 *THETA2 (IX)*3.0/BWIDTHF (IX) **2.	FECA0800
IF (IPARID (IP, 1) .EQ. 1) THEN	FECA0240		
C_SRCE (IS) = PAR (IP)	FECA0250	CRKW END	
ELSE IF (IPARID (IP, 1) .EQ. 2) THEN	FECA0260	1130 CONTINUE	FECA0810
GAMMA (IS) = PAR (IP)			

```

ELSE IF (IPARID (IP, 1) .EQ. 17) THEN
  BETA (IS) = PAR (IP)
CFSP NOVEMBER, 1995 CHANGE FOR FLOW SENSITIVITY ANALYSIS
ELSE IF (IPARID (IP, 1) .GT. 17) THEN
  DO 1135 IX = IPARID (IP, 3), IPARID (IP, 4)
    ICTR = IPARID (IP, 1) - NPARIT
    FSP (IX, ICTR) = PAR (IP)
    IF (IBTC .EQ. 0 ) THEN
      FSPM (IX, ICTR) = PAR (IP)
    ENDIF
  1135 CONTINUE
CFSP END
END IF
1000 CONTINUE
C
C If THETA2=0, then setup problem for no mass storage, or
C If ALPHA=0, then setup problem for no mass storage.
C
  DELX = DELXSAV
  NUMX = NUMXSAV
  DELT = DELTSAV
  NUMXM1 = NUMX - 1
  LDELT = 0
C
  DO 1140 IO = 1, NUMOBS
    NOBSLOC (IO) = NOBSSAV (IO)
  1140 CONTINUE
C
  VMAX = - 1.0E20
  VMIN = 1.0E20
  IF (NRC.LT.1) THEN
    NFSPP=NFSPP
    SENFCT=1.0
  ELSE
    NFSPP=NFSPP-1
    SENFCT=FSP(1,NFSP)/FSPM(1,NFSP)
  ENDIF
C
C Default is to use velocity directly.
C If SCL_PERM is > 0, calculate velocity by incorporating fracture width
C and user-defined permeability (SCL_PERM).
C
  DO 1150 IX = 1, NUMXM1
    IF (SCL_PERM .GT. 0.0) THEN
C Effective porosity
      EFFPOR = FWIDTH (IX)/(FWIDTH (IX) + BWIDTH (IX))
C Velocity = gradient * fracture width**2
      VEL (IX) = GRAD (IX)*SCL_PERM*FWIDTH (IX)*FWIDTH (IX)
      VMAX = MAX (VEL (IX), VMAX)
      VMIN = MIN (VEL (IX), VMIN)
    ELSE IF (NFSPP .EQ. 0) THEN
      VEL (IX) = GRAD (IX)*SENFCT
    FECA1170
CFSP NOVEMBER, 1995 CALCULATE NEW SPECIFIC DISCHARGE
    ELSE IF (NFSPP .GT. 0) THEN
      V1TRM = VVECT (IX,1)
      V2TRM = VVECT (IX,2)
      V3TRM = VVECT (IX,3)
      IF (NRC.LE.0) THEN
        NFSPP=NFSPP
        SENFCT=1.0
      ELSE
        NFSPP=NFSPP-1
        SENFCT=FSP(1,NFSP)/FSPM(1,NFSP)
      ENDIF

```

FECA0820
FECA0830

FECA0840
FECA0850
FECA0860
FECA0870
FECA0880
FECA0890
FECA0900
FECA0910
FECA0920
FECA0930
FECA0940
FECA0950
FECA0960
FECA0970
FECA0980
FECA0990
FECA1000
FECA1010
FECA1020

FECA1030
FECA1040
FECA1050
FECA1060
FECA1070
FECA1080
FECA1090
FECA1100
FECA1110
FECA1120
FECA1130
FECA1140
FECA1150
FECA1160

```

DO 1155 IFP = 1, NFSPP
  DELFP = (FSP (1, IFP) - FSPM (1, IFP))/FSPM (1, IFP)
  V1TRM = V1TRM + DELFP*SENS (IX, IFP, 1)*365.25

  V2TRM = V2TRM + DELFP*SENS (IX, IFP, 2)*365.25

  V3TRM = V3TRM + DELFP*SENS (IX, IFP, 3)*365.25

1155 CONTINUE
  VEL (IX) = SQRT (V1TRM**2 + V2TRM**2 + V3TRM**2)*SENFCT
CFSP END
END IF
1150 CONTINUE
C
C Auto time stepping (when DELT < 0.0)
C
  IF (DELT .LT. 0.0) THEN
C
C Set flag for auto-time stepping
C
    LDELT = 1
    DELT = DELX*1.5/VMAX
    IF (DELT .GT. ABS (DELTSAV)) DELT = ABS (DELTSAV)
    NUMT = TMAX/DELT + 0.5
    IF (NUMT .GT. 10000) THEN
      NUMT = 10001
      DELT = TMAX/10000
    END IF
    WRITE (*, 6000) NUMX, DELX, NUMT, DELT, VMIN, VMAX
6000 FORMAT(/, ' Auto time stepping option selected (DELT<0)',/,/,
1    ' NUMX =', I5, ' DELX =', 1P,G11.3,0P,/,/,
2    ' NUMT =', I5, ' DELT =', 1P,G11.3,/,/,
3    ' VMIN =', G11.3, ' VMAX =', G11.3)
  ELSE
C
C Normal execution ( DELT > 0)
C
    NUMT = TMAX/DELT + 0.5
    NUMT = NUMT + 1
  END IF
C
C Load variables
C
  DO 1160 IX = 1, NUMXM1
    DO 1170 IS = 1, NSPEC
      IF (ALPHA (IX, IS) .EQ. 0.0D0 .OR.
1      THETA2 (IX) .EQ. 0.0D0) THEN
        ALPHA (IX, IS) = 0.0D0
        IF (THETA2 (IX) .LE. 0.0D0) THEN
          THETA2 (IX) = 0.01D0
        END IF
        R1 (IX, IS) = 1.0D0 + RHOB (IX)*RKD1 (IX, IS)/THETA1 (IX)
        R2 (IX, IS) = 1.0D0
      ELSE
        R1 (IX, IS) = 1.0D0 + RHOB (IX)*RKD1 (IX, IS)/THETA1 (IX)
        R2 (IX, IS) = 1.0D0 + RHOB (IX)*RKD2 (IX, IS)/THETA2 (IX)
      END IF
    1170 CONTINUE
  C
  C Check for zero widths.
  C

```

FECA1180
FECA1190
FECA1200
FECA1210
FECA1220
FECA1230
FECA1240
FECA1250
FECA1260
FECA1270
FECA1280
FECA1290
FECA1300
FECA1310
FECA1320
FECA1330
FECA1340
FECA1350
FECA1360
FECA1370
FECA1380
FECA1390
FECA1400
FECA1410
FECA1420
FECA1430
FECA1440
FECA1450
FECA1460
FECA1470
FECA1480
FECA1490
FECA1500
FECA1510
FECA1520
FECA1530
FECA1540
FECA1550
FECA1560
FECA1570
FECA1580
FECA1590
FECA1600
FECA1610
FECA1620
FECA1630
FECA1640
FECA1650
FECA1660
FECA1670

UP11 = - 1.D0/DELX	FEM00710	END IF	FEM01360
UT = DELX/3.0D0	FEM00720	IF (NSPEC .EQ. 4) THEN	FEM01370
UT1 = 0.5D0	FEM00730	CHECK = ABS (GAMMA (1) - GAMMA (4))	FEM01380
UT11 = 1.0D0/DELX	FEM00740	IF (CHECK .GT. CHMIN) THEN	FEM01390
US = UP	FEM00750	IZ14 = 1	FEM01400
US1 = 0.5D0	FEM00760	END IF	FEM01410
US11 = UP11	FEM00770		FEM01420
C	FEM00780	C	FEM01430
C Check source decay rates for zero divide potential.	FEM00790	CHECK = ABS (GAMMA (2) - GAMMA (4))	FEM01440
C	FEM00800	IF (CHECK .GT. CHMIN) THEN	FEM01450
CHMIN = 0.0	FEM00810	IZ24 = 1	FEM01460
IF (NSPEC .GE. 2) THEN	FEM00820	END IF	FEM01470
C	FEM00830	C	FEM01480
CHECK = ABS (GAMMA (1) - GAMMA (2))	FEM00840	CHECK = ABS (GAMMA (3) - GAMMA (4))	FEM01490
IF (CHECK .GT. CHMIN) THEN	FEM00850	IF (CHECK .GT. CHMIN) THEN	FEM01500
IZ12 = 1	FEM00860	IZ34 = 1	FEM01510
END IF	FEM00870	END IF	FEM01520
C	FEM00880	IZ41 = IZ14	FEM01530
IF (IZ12 .EQ. 0) THEN	FEM00890	IZ42 = IZ24	FEM01540
DENB2 = 0.0	FEM00900	IZ43 = IZ34	FEM01550
ELSE	FEM00910		FEM01560
DENB2 = 1.0D0/(GAMMA (1) - GAMMA (2))	FEM00920	IF (IZ34 .EQ. 0) THEN	FEM01570
END IF	FEM00930	DEND2 = 0.0	FEM01580
END IF	FEM00940	ELSE	FEM01590
C	FEM00950	DEND2 = 1.0D0/(GAMMA (3) - GAMMA (4))	FEM01600
IF (NSPEC .GE. 3) THEN	FEM00960	END IF	FEM01610
IZ21 = IZ12	FEM00970	C	FEM01620
CHECK = ABS (GAMMA (1) - GAMMA (3))	FEM00980	IF (IZ32 .EQ. 0 .OR. IZ42 .EQ. 0) THEN	FEM01630
IF (CHECK .GT. CHMIN) THEN	FEM00990	DEND31 = 0.0	FEM01640
IZ13 = 1	FEM01000	ELSE	FEM01650
END IF	FEM01010	DEND31 = 1.0D0/((GAMMA (3) - GAMMA (2))*	FEM01660
CHECK = ABS (GAMMA (2) - GAMMA (3))	FEM01020	(GAMMA (4) - GAMMA (2)))	FEM01670
IF (CHECK .GT. CHMIN) THEN	FEM01030	END IF	FEM01680
IZ23 = 1	FEM01040	C	FEM01690
END IF	FEM01050	IF (IZ23 .EQ. 0 .OR. IZ43 .EQ. 0) THEN	FEM01700
IZ31 = IZ13	FEM01060	DEND32 = 0.0	FEM01710
IZ32 = IZ23	FEM01070	ELSE	FEM01720
C	FEM01080	DEND32 = 1.0D0/((GAMMA (2) - GAMMA (3))*	FEM01730
IF (IZ23 .EQ. 0) THEN	FEM01090	(GAMMA (4) - GAMMA (3)))	FEM01740
DENC2 = 0.0	FEM01100	END IF	FEM01750
ELSE	FEM01110	C	FEM01760
DENC2 = 1.0D0/(GAMMA (2) - GAMMA (3))	FEM01120	IF (IZ24 .EQ. 0 .OR. IZ34 .EQ. 0) THEN	FEM01770
END IF	FEM01130	DEND33 = 0.0	FEM01780
C	FEM01140	ELSE	FEM01790
IF (IZ21 .EQ. 0 .OR. IZ31 .EQ. 0) THEN	FEM01150	DEND33 = 1.0D0/((GAMMA (2) - GAMMA (4))*	FEM01800
DENC31 = 0.0	FEM01160	(GAMMA (3) - GAMMA (4)))	FEM01810
ELSE	FEM01170	END IF	FEM01820
DENC31 = 1.0D0/((GAMMA (2) - GAMMA (1))*	FEM01180	C	FEM01830
(GAMMA (3) - GAMMA (1)))	FEM01190	IF (IZ21 .EQ. 0 .OR. IZ31 .EQ. 0 .OR. IZ41 .EQ. 0) THEN	FEM01840
END IF	FEM01200	DEND41 = 0.0	FEM01850
C	FEM01210	ELSE	FEM01860
IF (IZ12 .EQ. 0 .OR. IZ32 .EQ. 0) THEN	FEM01220	DEND41 = 1.0D0/((GAMMA (2) - GAMMA (1))*	FEM01870
DENC32 = 0.0	FEM01230	(GAMMA (3) - GAMMA (1))*	FEM01880
ELSE	FEM01240	(GAMMA (4) - GAMMA (1)))	FEM01890
DENC32 = 1.0D0/((GAMMA (1) - GAMMA (2))*	FEM01250	END IF	FEM01900
(GAMMA (3) - GAMMA (2)))	FEM01260	C	FEM01910
END IF	FEM01270	IF (IZ12 .EQ. 0 .OR. IZ32 .EQ. 0 .OR. IZ42 .EQ. 0) THEN	FEM01920
C	FEM01280	DEND42 = 0.0	FEM01930
IF (IZ13 .EQ. 0 .OR. IZ23 .EQ. 0) THEN	FEM01290	ELSE	FEM01940
DENC33 = 0.0	FEM01300	DEND42 = 1.0D0/((GAMMA (1) - GAMMA (2))*	FEM01950
ELSE	FEM01310	(GAMMA (3) - GAMMA (2))*	FEM01960
DENC33 = 1.0D0/((GAMMA (1) - GAMMA (3))*	FEM01320	(GAMMA (4) - GAMMA (2)))	FEM01970
(GAMMA (2) - GAMMA (3)))	FEM01330	END IF	FEM01980
END IF	FEM01340	C	FEM01990
C	FEM01350	IF (IZ13 .EQ. 0 .OR. IZ23 .EQ. 0 .OR. IZ43 .EQ. 0) THEN	FEM02000

	DEND43 = 0.0	FEMO2010		BP = BETA (1)	FEMO2660
	ELSE	FEMO2020	C		FEMO2670
	DEND43 = 1.0D0/((GAMMA (1) - GAMMA (3))*	FEMO2030		ELSE IF (IS .EQ. 3) THEN	FEMO2680
1	(GAMMA (2) - GAMMA (3))*	FEMO2040	C		FEMO2690
2	(GAMMA (4) - GAMMA (3)))	FEMO2050		CF = C_SRCE (3)*EXP (- GAMMA (3)*T) +	FEMO2700
	END IF	FEMO2060	1	GAMMA (2)*C_SRCE (2)*DENC2*(EXP (- GAMMA (3)*T) -	FEMO2710
C		FEMO2070	2	EXP (- GAMMA (2)*T)) +	FEMO2720
	IF (IZ14 .EQ. 0 .OR. IZ24 .EQ. 0 .OR. IZ34 .EQ. 0) THEN	FEMO2080	3	GAMMA (1)*GAMMA (2)*C_SRCE (1)*	FEMO2730
	DEND44 = 0.0	FEMO2090	4	(EXP (- GAMMA (1)*T)*DENC31 +	FEMO2740
	ELSE	FEMO2100	5	EXP (- GAMMA (2)*T)*DENC32 + EXP	FEMO2750
1	DEND44 = 1.0D0/((GAMMA (1) - GAMMA (4))*	FEMO2110	6	(- GAMMA (3)*T)*DENC33)	FEMO2760
1	(GAMMA (2) - GAMMA (4))*	FEMO2120		IF (T .GT. TOFF) CF = 0.0	FEMO2770
2	(GAMMA (3) - GAMMA (4)))	FEMO2130		SF = CF	FEMO2780
	END IF	FEMO2140		BP = BETA (2)	FEMO2790
C		FEMO2150	C		FEMO2800
	END IF	FEMO2160		ELSE IF (IS .EQ. 4) THEN	FEMO2810
C		FEMO2170	C		FEMO2820
C		FEMO2180		CF = C_SRCE (4)*EXP (- GAMMA (4)*T) +	FEMO2830
C	Initialize concentration values.	FEMO2190	1	GAMMA (3)*C_SRCE (3)*DEND2*(EXP (- GAMMA (4)*T) -	FEMO2840
C		FEMO2200	2	EXP (- GAMMA (3)*T)) +	FEMO2850
	DO 1000 IS = 1, NSPEC	FEMO2210	3	GAMMA (2)*GAMMA (3)*C_SRCE (2)*(EXP (- GAMMA (2)*T)*	FEMO2860
	CFOLD (IS) = C_SRCE (IS)	FEMO2220	4	DEND31 + EXP (- GAMMA (3)*T)*DEND32 +	FEMO2870
C		FEMO2230	5	EXP (- GAMMA (4)*T)*DEND33) +	FEMO2880
	DO 1000 I = 1, NUMX	FEMO2240	6	GAMMA (1)*GAMMA (2)*GAMMA (3)*	FEMO2890
	COLD (I, IS) = CINIT (I, IS)	FEMO2250	7	C_SRCE (1)*(EXP (- GAMMA (1)*T)*	FEMO2900
	SOLD (I, IS) = SINIT (I, IS)	FEMO2260	8	DEND41 + EXP (- GAMMA (2)*T)*DEND42 +	FEMO2910
	COLD2 (I, IS, 2) = 0.0D0	FEMO2270	9	EXP (- GAMMA (3)*T)*DEND43 +	FEMO2920
	SOLD2 (I, IS, 2) = 0.0D0	FEMO2280	A	EXP (- GAMMA (4)*T)*DEND44)	FEMO2930
	IF (IS .GT. 1) THEN	FEMO2290		IF (T .GT. TOFF) CF = 0.0	FEMO2940
	COLD2 (I, IS, 1) = CINIT (I, IS - 1)	FEMO2300		SF = CF	FEMO2950
	SOLD2 (I, IS, 1) = SINIT (I, IS - 1)	FEMO2310		BP = BETA (3)	FEMO2960
	ELSE	FEMO2320		END IF	FEMO2970
	COLD2 (I, IS, 1) = 0.0D0	FEMO2330	C		FEMO2980
	SOLD2 (I, IS, 1) = 0.0D0	FEMO2340		DO 1020 IX = 2, NUMXM1	FEMO2990
	END IF	FEMO2350		IF (IS .EQ. 1) THEN	FEMO3000
1000	CONTINUE	FEMO2360		R1P1 = 0.0D0	FEMO3010
C		FEMO2370		R1P2 = 0.0D0	FEMO3020
C		FEMO2380		R2P1 = 0.0D0	FEMO3030
C		FEMO2390		R2P2 = 0.0D0	FEMO3040
C	=====	FEMO2400		ELSE	FEMO3050
C	Start time loop.	FEMO2410		R1P1 = R1 (IX - 1, IS - 1)	FEMO3060
C	=====	FEMO2420		R1P2 = R1 (IX, IS - 1)	FEMO3070
C		FEMO2430		R2P1 = R2 (IX - 1, IS - 1)	FEMO3080
	T = 0.0D0	FEMO2440		R2P2 = R2 (IX, IS - 1)	FEMO3090
C		FEMO2450		END IF	FEMO3100
	DO 1010 IT = 2, NUMT	FEMO2460	C		FEMO3110
C		FEMO2470		BW11 = 2.0D0/BWIDTH (IX - 1)	FEMO3120
	T = T + DELT	FEMO2480		BW12 = 2.0D0/BWIDTH (IX)	FEMO3130
C		FEMO2490		FW11 = 2.0D0/FWIDTH (IX - 1)	FEMO3140
C	IF (MOD(IT,25).EQ.0) WRITE (*,*) ' T = ', T			FW12 = 2.0D0/FWIDTH (IX)	FEMO3150
C	DO 1010 IS = 1, NSPEC	FEMO2510	C		FEMO3160
C		FEMO2520		PP = ALPHA (IX - 1, IS)*FW11*UP + BETA (IS)*	FEMO3170
	IF (IS .EQ. 1) THEN	FEMO2530	1	THETA1 (IX - 1)*R1 (IX - 1, IS)*UP +	FEMO3180
	CF = C_SRCE (1)*EXP (- GAMMA (1)*T)	FEMO2540	2	DH (IX - 1)*UP11 + VEL (IX - 1)*UP1	FEMO3190
	IF (T .GT. TOFF) CF = 0.0	FEMO2550		PT = (ALPHA (IX - 1, IS)*FW11 + ALPHA (IX,	FEMO3200
	SF = CF	FEMO2560	1	IS)*FW12)*UT + BETA (IS)*	FEMO3210
	BP = 0.0D0	FEMO2570	2	(THETA1 (IX - 1)*R1 (IX - 1, IS) +	FEMO3220
C		FEMO2580	3	THETA1 (IX)*R1 (IX, IS))*UT +	FEMO3230
	ELSE IF (IS .EQ. 2) THEN	FEMO2590	4	(DH (IX - 1) + DH (IX))*UT11 + (VEL	FEMO3240
C		FEMO2600	5	(IX - 1) - VEL (IX))*UT1	FEMO3250
	CF = C_SRCE (2)*EXP (- GAMMA (2)*T) +	FEMO2610		PS = ALPHA (IX, IS)*FW12*US + BETA (IS)*	FEMO3260
1	GAMMA (1)*C_SRCE (1)*DENB2*(EXP (- GAMMA (2)*T) -	FEMO2620	1	THETA1 (IX)*R1 (IX, IS)*US +	FEMO3270
2	EXP (- GAMMA (1)*T))	FEMO2630	2	DH (IX)*US11 + VEL (IX)*US1	FEMO3280
	IF (T .GT. TOFF) CF = 0.0	FEMO2640		P2P = - ALPHA (IX - 1, IS)*FW11*UP	FEMO3290
	SF = CF	FEMO2650		P2T = - (ALPHA (IX - 1, IS)*FW11 +	FEMO3300

1	ALPHA (IX, IS)*FWI2)*UT	FEMO3310		B2 (1) = 0.0D0	FEMO3960
	P2S = - ALPHA (IX, IS)*FWI2*US	FEMO3320		B3 (1) = 0.0D0	FEMO3970
C		FEMO3330		B4 (1) = 1.0D0	FEMO3980
	FAC1 = 2.0D0*THETA1 (IX - 1)*R1 (IX - 1, IS)*UP	FEMO3340		C1 (1) = 0.0D0	FEMO3990
	FAC2 = 2.0D0*(THETA1 (IX - 1)*R1 (IX - 1, IS) +	FEMO3350		C2 (1) = 0.0D0	FEMO4000
1	THETA1 (IX)*R1 (IX, IS))*UT	FEMO3360		C3 (1) = 0.0D0	FEMO4010
	FAC3 = 2.0D0*THETA1 (IX)*R1 (IX, IS)*US	FEMO3370		C4 (1) = 0.0D0	FEMO4020
C		FEMO3380		D1 (1) = CF	FEMO4030
	A1 (IX) = FAC1 + DELT*PP	FEMO3390		D2 (1) = SF	FEMO4040
	B1 (IX) = FAC2 + DELT*PT	FEMO3400		ELSE	FEMO4050
	C1 (IX) = FAC3 + DELT*PS	FEMO3410	C		FEMO4060
	A2 (IX) = DELT*P2P	FEMO3420	C	Adjust coefficients for the inflow flux boundary condition.	FEMO4070
	B2 (IX) = DELT*P2T	FEMO3430	C		FEMO4080
	C2 (IX) = DELT*P2S	FEMO3440		IF (IS .EQ. 1) THEN	FEMO4090
	A3 (IX) = - ALPHA (IX - 1, IS)*BWI1*UP*DELT	FEMO3450		R1P2 = 0.0D0	FEMO4100
	B3 (IX) = - (ALPHA (IX - 1, IS)*BWI1 +	FEMO3460		R2P2 = 0.0D0	FEMO4110
1	ALPHA (IX, IS)*BWI2)*UT*DELT	FEMO3470		ELSE	FEMO4120
	C3 (IX) = - ALPHA (IX, IS)*BWI2*US*DELT	FEMO3480		R1P2 = R1 (1, IS - 1)	FEMO4130
C		FEMO3490		R2P2 = R2 (1, IS - 1)	FEMO4140
	FAC4 = 2.0D0*THETA2 (IX - 1)*R2 (IX - 1, IS)*UP	FEMO3500		END IF	FEMO4150
	FAC5 = 2.0D0*(THETA2 (IX - 1)*R2 (IX - 1, IS) +	FEMO3510	C		FEMO4160
1	THETA2 (IX)*R2 (IX, IS))*UT	FEMO3520		BWI2 = 2.0D0/BWIDTH (1)	FEMO4170
	FAC6 = 2.0D0*THETA2 (IX)*R2 (IX, IS)*US	FEMO3530		FWI2 = 2.0D0/FWIDTH (1)	FEMO4180
C		FEMO3540	C		FEMO4190
	A4 (IX) = FAC4 - A3 (IX) + BETA (IS)*THETA2 (IX -	FEMO3550		PT = ALPHA (1, IS)*FWI2*UT + BETA (IS)*	FEMO4200
1	1)*R2 (IX - 1, IS)*UP*DELT	FEMO3560	1	THETA1 (1)*R1 (1, IS)*UT +	FEMO4210
	B4 (IX) = FAC5 - B3 (IX) + BETA (IS)*	FEMO3570	2	DH (1)*UT11 - VEL (1)*UT1	FEMO4220
1	(THETA2 (IX - 1)*R2 (IX - 1, IS) +	FEMO3580	C		FEMO4230
2	THETA2 (IX)*R2 (IX, IS))*UT*DELT	FEMO3590		PS = ALPHA (1, IS)*FWI2*US + BETA (IS)*	FEMO4240
	C4 (IX) = FAC6 - C3 (IX) + BETA (IS)*THETA2	FEMO3600	1	THETA1 (1)*R1 (1, IS)*US +	FEMO4250
1	(IX)*R2 (IX, IS)*US*DELT	FEMO3610	2	DH (1)*US11 + VEL (1)*US1	FEMO4260
	D1 (IX) = (FAC1 - PP*DELT)*COLD (IX - 1, IS) +	FEMO3620	C		FEMO4270
1	(FAC2 - PT*DELT)*COLD (IX, IS) +	FEMO3630		P2T = - ALPHA (1, IS)*FWI2*UT	FEMO4280
2	(FAC3 - PS*DELT)*COLD (IX + 1, IS) - A2 (IX)	FEMO3640		P2S = - ALPHA (1, IS)*FWI2*US	FEMO4290
3	*SOLD (IX - 1, IS) - B2 (IX)*	FEMO3650	C		FEMO4300
4	SOLD (IX, IS) - C2 (IX)*SOLD (IX + 1, IS) +	FEMO3660		A1 (1) = 0.0D0	FEMO4310
5	BP*THETA1 (IX - 1)*R1P1*UP*	FEMO3670		B1 (1) = 2.0D0*THETA1 (1)*R1 (1, IS)*UT + DELT*PT +	FEMO4320
6	DELT*(COLD2 (IX - 1, IS, 1) + COLD2 (IX - 1,	FEMO3680	1	VEL (1)*DELT	FEMO4330
7	IS, 2)) + BP*(THETA1 (IX - 1)*	FEMO3690		C1 (1) = 2.0D0*THETA1 (1)*R1 (1, IS)*US + DELT*PS	FEMO4340
8	R1P1 + THETA1 (IX)*R1P2)*UT*DELT*(COLD2 (IX, IS, 1) +	FEMO3700	C		FEMO4350
9	COLD2 (IX, IS, 2)) + BP*THETA1 (IX)*R1P2*US*DELT*	FEMO3710		A2 (1) = 0.0	FEMO4360
A	(COLD2 (IX + 1, IS, 1) + COLD2 (IX + 1, IS, 2))	FEMO3720		B2 (1) = DELT*P2T	FEMO4370
C		FEMO3730		C2 (1) = DELT*P2S	FEMO4380
	D2 (IX) = (2.0D0*FAC4 - A4 (IX))*SOLD (IX - 1,	FEMO3740	C		FEMO4390
1	IS) + (2.0D0*FAC5 - B4 (IX))*	FEMO3750		D1 (1) = (2.0D0*THETA1 (1)*R1 (1, IS)*UT -	FEMO4400
2	SOLD (IX, IS) + (2.0D0*FAC6 - C4 (IX))*	FEMO3760	1	DELT*PT)*COLD (1, IS) +	FEMO4410
3	SOLD (IX + 1, IS) - A3 (IX)*	FEMO3770	2	(2.0D0*THETA1 (1)*R1 (1, IS)*US -	FEMO4420
4	COLD (IX - 1, IS) - B3 (IX)*COLD (IX, IS)	FEMO3780	3	DELT*PS)*COLD (2, IS) -	FEMO4430
5	- C3 (IX)*COLD (IX + 1, IS) +	FEMO3790	4	(DELT*P2T)*SOLD (1, IS) - (DELT*P2S)*SOLD (2, IS) +	FEMO4440
6	BP*THETA2 (IX - 1)*R2P1*UP*DELT*	FEMO3800	5	VEL (1)*DELT*(CFOLD (IS) + CF - COLD (1, IS)) +	FEMO4450
7	(SOLD2 (IX - 1, IS, 1) + SOLD2 (IX - 1, IS,	FEMO3810	6	BP*THETA1 (1)*R1P2*UT*DELT*(COLD2	FEMO4460
8	2)) + BP*(THETA2 (IX - 1)*	FEMO3820	7	(1, IS, 1) + COLD2 (1, IS, 2)) +	FEMO4470
9	R2P1 + THETA2 (IX)*R2P2)*UT*DELT*(SOLD2 (IX, IS, 1) +	FEMO3830	8	BP*THETA1 (1)*R1P2*US*DELT*(COLD2	FEMO4480
A	SOLD2 (IX, IS, 2)) + BP*THETA2 (IX)*R2P2*US*DELT*	FEMO3840	9	(2, IS, 1) + COLD2 (2, IS, 2))	FEMO4490
B	(SOLD2 (IX + 1, IS, 1) + SOLD2 (IX + 1, IS, 2))	FEMO3850	C		FEMO4500
1020	CONTINUE	FEMO3860		A3 (1) = 0.0D0	FEMO4510
C		FEMO3870		B3 (1) = - ALPHA (1, IS)*BWI2*UT*DELT	FEMO4520
C	Set coefficients for the inflow boundary condition.	FEMO3880		C3 (1) = - ALPHA (1, IS)*BWI2*US*DELT	FEMO4530
C		FEMO3890	C		FEMO4540
	IF (INBOUN .LE. 1) THEN	FEMO3900		A4 (1) = 0.0D0	FEMO4550
	A1 (1) = 0.0D0	FEMO3910		B4 (1) = 2.0D0*THETA2 (1)*R2 (1, IS)*UT +	FEMO4560
	A2 (1) = 0.0D0	FEMO3920	1	ALPHA (1, IS)*BWI2*UT*DELT +	FEMO4570
	A3 (1) = 0.0D0	FEMO3930	2	BETA (IS)*THETA2 (1)*R2 (1, IS)*UT*DELT	FEMO4580
	A4 (1) = 0.0D0	FEMO3940		C4 (1) = 2.0D0*THETA2 (1)*R2 (1, IS)*US +	FEMO4590
	B1 (1) = 1.0D0	FEMO3950	1	ALPHA (1, IS)*BWI2*US*DELT +	FEMO4600

C	2	BETA (IS)*THETA2 (1)*R2 (1, IS)*US*DELT	FEMO4610	C		FEMO5260
			FEMO4620		B4 (NX) = 2.0D0*THETA2 (NX - 1)*R2 (NX - 1, IS)*UT +	FEMO5270
		D2 (1) = (2.0D0*THETA2 (1)*R2 (1, IS)*UT -	FEMO4630	1	ALPHA (NX - 1, IS)*BW11*UT*	FEMO5280
	1	(ALPHA (1, IS)*BW12 + BETA (IS)*	FEMO4640	2	DELT + BETA (IS)*THETA2 (NX - 1)*R2 (NX - 1, IS)*UT*DELT	FEMO5290
	2	THETA2 (1)*R2 (1, IS))*UT*DELT)*	FEMO4650	C		FEMO5300
	3	SOLD (1, IS) + (2.0D0*THETA2 (1)*R2 (1, IS)	FEMO4660		C4 (NX) = 0.0D0	FEMO5310
	4	*US - (ALPHA (1, IS)*BW12 +	FEMO4670	C		FEMO5320
	5	BETA (IS)*THETA2 (1)*R2 (1, IS))*	FEMO4680		D2 (NX) = (2.0D0*THETA2 (NX - 1)*R2 (NX - 1, IS)*	FEMO5330
	6	US*DELT)*SOLD (2, IS) +	FEMO4690	1	UP - (ALPHA (NX - 1, IS)*BW11 +	FEMO5340
	7	(ALPHA (1, IS)*BW12*UT*DELT)*COLD (1, IS) +	FEMO4700	2	BETA (IS)*THETA2 (NX - 1)*R2 (NX - 1, IS))*	FEMO5350
	8	(ALPHA (1, IS)*BW12*US*DELT)*COLD (2, IS) +	FEMO4710	3	UP*DELT)*SOLD (NX - 1, IS) + (2.0D0*	FEMO5360
	9	BP*THETA2 (1)*R2P2*UT*DELT*(SOLD2	FEMO4720	4	THETA2 (NX - 1)*R2 (NX - 1, IS)*UT -	FEMO5370
A		(1, IS, 1) + SOLD2 (1, IS, 2)) +	FEMO4730	5	(ALPHA (NX - 1, IS)*BW11 + BETA (IS)*	FEMO5380
B		BP*THETA2 (1)*R2P2*US*DELT*(SOLD2	FEMO4740	6	THETA2 (NX - 1)*R2 (NX - 1, IS))*	FEMO5390
C		(2, IS, 1) + SOLD2 (2, IS, 2))	FEMO4750	7	UT*DELT)*SOLD (NX, IS) + (ALPHA (NX -	FEMO5400
		END IF	FEMO4760	8	1, IS)*BW11*UP*DELT)*	FEMO5410
C			FEMO4770	9	COLD (NX - 1, IS) + (ALPHA (NX - 1, IS)*BW11*	FEMO5420
C		Adjust coefficients for the outflow boundary condition.	FEMO4780	A	UT*DELT)*COLD (NX, IS) +	FEMO5430
C			FEMO4790	B	BP*THETA2 (NX - 1)*R2P1*UP*DELT*	FEMO5440
		IF (IS .EQ. 1) THEN	FEMO4800	C	(SOLD2 (NX - 1, IS, 1) + SOLD2 (NX - 1, IS, 2))	FEMO5450
		R1P1 = 0.0D0	FEMO4810	D	+ BP*THETA2 (NX - 1)*	FEMO5460
		R2P1 = 0.0D0	FEMO4820	E	R2P1*UT*DELT*(SOLD2 (NX, IS, 1) + SOLD2 (NX, IS, 2))	FEMO5470
		ELSE	FEMO4830	C		FEMO5480
		R1P1 = R1 (NX - 1, IS - 1)	FEMO4840	C	Solve the linear system using the bi-tridiagonal matrix solver.	FEMO5490
		R2P1 = R2 (NX - 1, IS - 1)	FEMO4850	C		FEMO5500
		END IF	FEMO4860	C	*****	FEMO5510
C			FEMO4870		CALL BITRI (CVEC, SVEC, NX)	FEMO5520
		BW11 = 2.0D0/BWIDTH (NX - 1)	FEMO4880	C	*****	FEMO5530
		FW11 = 2.0D0/FWIDTH (NX - 1)	FEMO4890	C		FEMO5540
C			FEMO4900	C	Expand the solution and refresh COLD(I,IS),SOLD(I,IS).	FEMO5550
		PP = ALPHA (NX - 1, IS)*FW11*UP + BETA (IS)*	FEMO4910	C		FEMO5560
1		THETA1 (NX - 1)*R1 (NX - 1, IS)*UP +	FEMO4920		DO 1030 IX = 1, NX	FEMO5570
2		DH (NX - 1)*UP11 + VEL (NX - 1)*UP1	FEMO4930		COLD (IX, IS) = CVEC (IX)	FEMO5580
C			FEMO4940		SOLD (IX, IS) = SVEC (IX)	FEMO5590
		PT = ALPHA (NX - 1, IS)*FW11*UT + BETA (IS)*	FEMO4950		IF (NS .GT. 1) THEN	FEMO5600
1		THETA1 (NX - 1)*R1 (NX - 1, IS)*UT +	FEMO4960		IF (IS .LT. NS) THEN	FEMO5610
2		DH (NX - 1)*UT11 + VEL (NX - 1)*UT1	FEMO4970		COLD2 (IX, IS + 1, 2) = CVEC (IX)	FEMO5620
C			FEMO4980		SOLD2 (IX, IS + 1, 2) = SVEC (IX)	FEMO5630
		P2P = - ALPHA (NX - 1, IS)*FW11*UP	FEMO4990		END IF	FEMO5640
		P2T = - ALPHA (NX - 1, IS)*FW11*UT	FEMO5000		IF (IS .GT. 1) THEN	FEMO5650
C			FEMO5010		COLD2 (IX, IS, 1) = COLD2 (IX, IS, 2)	FEMO5660
		A1 (NX) = 2.0D0*THETA1 (NX - 1)*R1 (NX - 1, IS)*UP + DELT*PP	FEMO5020		SOLD2 (IX, IS, 1) = SOLD2 (IX, IS, 2)	FEMO5670
		B1 (NX) = 2.0D0*THETA1 (NX - 1)*R1 (NX - 1, IS)*UT + DELT*PT	FEMO5030		END IF	FEMO5680
		C1 (NX) = 0.0D0	FEMO5040		END IF	FEMO5690
C			FEMO5050	1030	CONTINUE	FEMO5700
		A2 (NX) = DELT*P2P	FEMO5060	C		FEMO5710
		B2 (NX) = DELT*P2T	FEMO5070	C	Store the results for the specified observation/receptor locations.	FEMO5720
		C2 (NX) = 0.0D0	FEMO5080	C		FEMO5730
C			FEMO5090		DO 1040 IO = 1, NUMOBS	FEMO5740
		D1 (NX) = (2.0D0*THETA1 (NX - 1)*R1 (NX - 1, IS)*UP -	FEMO5100		BTC (IO, IT, IS) = COLD (NOBSLOC (IO), IS)	FEMO5750
1		DELT*PP)*COLD (NX - 1, IS) +	FEMO5110	1040	CONTINUE	FEMO5760
2		(2.0D0*THETA1 (NX - 1)*R1 (NX - 1, IS)*UT -	FEMO5120	C		FEMO5770
3		DELT*PT)*COLD (NX, IS) -	FEMO5130	C	Save old source concentration.	FEMO5780
4		(DELT*P2P)*SOLD (NX - 1, IS) - (DELT*P2T)*SOLD (NX, IS) +	FEMO5140	C		FEMO5790
5		BP*THETA1 (NX - 1)*R1P1*UP*DELT*(COLD2 (NX - 1, IS, 1) +	FEMO5150		CFOLD (IS) = CF	FEMO5800
6		COLD2 (NX - 1, IS, 2)) + BP*THETA1 (NX - 1)*R1P1*UT*DELT*	FEMO5160	C		FEMO5810
7		(COLD2 (NX, IS, 1) + COLD2 (NX, IS, 2))	FEMO5170	C	Next time step.	FEMO5820
C			FEMO5180	C		FEMO5830
		A3 (NX) = - ALPHA (NX - 1, IS)*BW11*UP*DELT	FEMO5190		1010 CONTINUE	FEMO5840
		B3 (NX) = - ALPHA (NX - 1, IS)*BW11*UT*DELT	FEMO5200	C		FEMO5850
		C3 (NX) = 0.0D0	FEMO5210	C	=====	FEMO5860
C			FEMO5220	C	End of time loop.	FEMO5870
		A4 (NX) = 2.0D0*THETA2 (NX - 1)*R2 (NX - 1, IS)*UP +	FEMO5230	C	=====	FEMO5880
1		ALPHA (NX - 1, IS)*BW11*UP*	FEMO5240	C		FEMO5890
2		DELT + BETA (IS)*THETA2 (NX - 1)*R2 (NX - 1, IS)*UP*DELT	FEMO5250	C		FEMO5900

C	RETURN	FEMO5910	END	BITR0630
	END	FEMO5920		GPAR0010
C		FEMO5930		GPAR0020
C		BITR0010		GPAR0030
C		BITR0020	SUBROUTINE G_PAR1 (PSTAT, IP, NPARDIM, RNUM, PARVAL)	GPAR0040
C		BITR0030		GPAR0050
C	SUBROUTINE BITRI (U, V, N)	BITR0040	IMPLICIT DOUBLE PRECISION (A - H, O - Z)	GPAR0060
C		BITR0050	IMPLICIT INTEGER*4 (I - N)	GPAR0070
C	Bi-tridiagonal matrix solver. code based on the solution of VON	BITR0060		GPAR0080
C	ROSENBERG (____). Solves two linear systems of equations of the	BITR0070	DIMENSION PSTAT (NPARDIM, 4)	GPAR0090
C	form:	BITR0080	REAL RNUM (1)	GPAR0100
C		BITR0090	DOUBLE PRECISION RNUM (1)	GPAR0110
C	A1*U(K-1) + B1*U(K) + C1*U(K+1) +	BITR0100		GPAR0120
C	A2*V(K-1) + B2*V(K) + C2*V(K+1) = D1(K)	BITR0110	C Generate the random number using the seed of 0.	GPAR0130
C		BITR0120		GPAR0140
C	A3*U(K-1) + B3*U(K) + C3*U(K+1) +	BITR0130	C Check to see if the value is 0 or 1, if it is then re-call to	GPAR0150
C	A4*V(K-1) + B4*V(K) + C4*V(K+1) = D2(K)	BITR0140	C get a different number. if the value is zero or one then there	GPAR0160
C		BITR0150	C is too large a probability value generated by the inverse normal	GPAR0170
C		BITR0160	C routine.	GPAR0180
C		BITR0170		GPAR0190
C	INCLUDE 'MC_TRLHS.INC'	BITR0180	1000 CONTINUE	GPAR0200
C		BITR0190		GPAR0210
C	DIMENSION U (NN), V (NN)	BITR0200		GPAR0220
C		BITR0210	CALL MRAND (0, 1, RNUM)	GPAR0230
C	Calculate D1,F1 as a function of [A,B,C,D]{1-4}.	BITR0220		GPAR0240
C		BITR0230		GPAR0250
C	DEN = B1 (1)*B4 (1) - B2 (1)*B3 (1)	BITR0240	IF (RNUM (1) .LE. 0.0 .OR. RNUM (1) .GE. 1.0) GO TO 1000	GPAR0260
C		BITR0250		GPAR0270
C	E1 (1) = (B4 (1)*C1 (1) - B2 (1)*C3 (1))/DEN	BITR0260	C If the distribution is normal (IDIST=1), transform to normal	GPAR0280
C	E2 (1) = (B4 (1)*C2 (1) - B2 (1)*C4 (1))/DEN	BITR0270	C and scale the value to the mean and std.	GPAR0290
C	E3 (1) = (B1 (1)*C3 (1) - B3 (1)*C1 (1))/DEN	BITR0280	C If the distribution is lognormal (IDIST=2), transform to lognormal	GPAR0300
C	E4 (1) = (B1 (1)*C4 (1) - B3 (1)*C2 (1))/DEN	BITR0290	C and scale the value to the (log) mean and (log) std.	GPAR0310
C	F1 (1) = (B4 (1)*D1 (1) - B2 (1)*D2 (1))/DEN	BITR0300	C If the distribution is uniform (IDIST=3), scale the value according	GPAR0320
C	F2 (1) = (B1 (1)*D2 (1) - B3 (1)*D1 (1))/DEN	BITR0310	C to the max and min.	GPAR0330
C		BITR0320		GPAR0340
C	DO 1000 I = 2, N	BITR0330	IDIST = NINT (PSTAT (IP, 1))	GPAR0350
C	BT1 = B1 (I) - A1 (I)*E1 (I - 1) - A2 (I)*E3 (I - 1)	BITR0340	DMEAN = PSTAT (IP, 2)	GPAR0360
C	BT2 = B2 (I) - A1 (I)*E2 (I - 1) - A2 (I)*E4 (I - 1)	BITR0350	DSTD = PSTAT (IP, 3)	GPAR0370
C	BT3 = B3 (I) - A3 (I)*E1 (I - 1) - A4 (I)*E3 (I - 1)	BITR0360	IF (IDIST .EQ. 1) THEN	GPAR0380
C	BT4 = B4 (I) - A3 (I)*E2 (I - 1) - A4 (I)*E4 (I - 1)	BITR0370	PARVAL = GINV (RNUM (1))	GPAR0390
C	DT1 = D1 (I) - A1 (I)*F1 (I - 1) - A2 (I)*F2 (I - 1)	BITR0380	PARVAL = DMEAN + DSTD*PARVAL	GPAR0400
C	DT2 = D2 (I) - A3 (I)*F1 (I - 1) - A4 (I)*F2 (I - 1)	BITR0390	ELSE IF (IDIST .EQ. 2) THEN	GPAR0410
C		BITR0400	PARVAL = GINV (RNUM (1))	GPAR0420
C	DEN = BT1*BT4 - BT2*BT3	BITR0410		GPAR0430
C		BITR0420	C The mean and standard deviation entered for the lognormal	GPAR0440
C	E1 (I) = (BT4*C1 (I) - BT2*C3 (I))/DEN	BITR0430	C distribution are real values, as read from the y vs x plot with	GPAR0450
C	E2 (I) = (BT4*C2 (I) - BT2*C4 (I))/DEN	BITR0440	C the x-axis being logarithmic.	GPAR0460
C	E3 (I) = (BT1*C3 (I) - BT3*C1 (I))/DEN	BITR0450		GPAR0470
C	E4 (I) = (BT1*C4 (I) - BT3*C2 (I))/DEN	BITR0460	C TMEAN and TVAR are the transformed mean and variance respectively.	GPAR0480
C	F1 (I) = (BT4*DT1 - BT2*DT2)/DEN	BITR0470	C They are the parameters describing the normal distribution plot	GPAR0490
C	F2 (I) = (BT1*DT2 - BT3*DT1)/DEN	BITR0480	C of the log of the parameter values (plot of y vs ln(x), with	GPAR0500
C		BITR0490	C the x-axis linear.	GPAR0510
C	1000 CONTINUE	BITR0500		GPAR0520
C		BITR0510	TVAR = LOG (1.0D0 + (DSTD*DSTD)/(DMEAN*DMEAN))	GPAR0530
C	Solve for U(I) and V(I) in terms of U(I+1),V(I+1),	BITR0520	TMEAN = LOG (DMEAN) - 0.5D0*TVAR	GPAR0540
C	E1(I), E2(I), E3(I), E4(I), F1(I), and F2(I).	BITR0530		GPAR0550
C		BITR0540	C The parameter value is then the exponential of the value after	GPAR0560
C	U (N) = F1 (N)	BITR0550	C it has been scaled and shifted by the transformed mean and variance.	GPAR0570
C	V (N) = F2 (N)	BITR0560		GPAR0580
C		BITR0570	PARVAL = EXP (TMEAN + SQRT (TVAR)*PARVAL)	GPAR0590
C	DO 1010 I = 1, N - 1	BITR0580	ELSE IF (IDIST .EQ. 3) THEN	GPAR0600
C	II = N - I	BITR0590	PARVAL = DMEAN + RNUM (1)*(DSTD - DMEAN)	GPAR0610
C	U (II) = F1 (II) - E1 (II)*U (II + 1) - E2 (II)*V (II + 1)	BITR0600	END IF	GPAR0620
C	V (II) = F2 (II) - E3 (II)*U (II + 1) - E4 (II)*V (II + 1)	BITR0610		GPAR0630
C		BITR0620	RETURN	GPAR0640
C	1010 CONTINUE			
C	RETURN			

C	END	GPAR0650	C		PMEA0020
C		GPAR0010	C		PMEA0030
C		GPAR0020	C	SUBROUTINE P_MEAN (PD, PS, IP, N, NPARDIM, NDMAX)	PMEA0040
C		GPAR0030	C		PMEA0050
C	SUBROUTINE G_PAR2 (PDIST, IP, NPARDIM, NDMAX, NDIST, RNUM, PARVAL)	GPAR0040	C	IMPLICIT DOUBLE PRECISION (A - H, O - Z)	PMEA0060
C		GPAR0050	C	IMPLICIT INTEGER*4 (I - N)	PMEA0070
C	IMPLICIT DOUBLE PRECISION (A - H, O - Z)	GPAR0060	C		PMEA0080
C	IMPLICIT INTEGER*4 (I - N)	GPAR0070	C	DIMENSION PD (NPARDIM, NDMAX), PS (NPARDIM, 4)	PMEA0090
C		GPAR0080	C		PMEA0100
C	DIMENSION PDIST (NPARDIM, NDMAX)	GPAR0090	C	PMEAN = 0.D+0	PMEA0110
C	REAL RNUM (1), V_FRAC	GPAR0100	C	DO 1000 I = 1, N	PMEA0120
C	DOUBLE PRECISION RNUM (1), V_FRAC	GPAR0110	C	PMEAN = PMEAN + PD (IP, I)	PMEA0130
C		GPAR0120	C	1000 CONTINUE	PMEA0140
C	Generate the random number using the seed of 0.	GPAR0130	C	PS (IP, 2) = PMEAN/N	PMEA0150
C		GPAR0140	C		PMEA0160
C	*****	GPAR0150	C	RETURN	PMEA0170
C	CALL MRAND (0, 1, RNUM)	GPAR0160	C	END	PMEA0180
C	*****	GPAR0170	C		PSOR0010
C		GPAR0180	C		PSOR0020
C	Scale the result by the number of values in the distribution (NDIST).	GPAR0190	C		PSOR0030
C		GPAR0200	C	SUBROUTINE P_SORT (P, IP, N, NPARDIM, NDMAX)	PSOR0040
C	V_FRAC = RNUM (1)*FLOAT (NDIST)	GPAR0210	C		PSOR0050
C		GPAR0220	C	IMPLICIT DOUBLE PRECISION (A - H, O - Z)	PSOR0060
C	Interpolate the value between the two closest data points.	GPAR0230	C	IMPLICIT INTEGER*4 (I - N)	PSOR0070
C		GPAR0240	C		PSOR0080
C	NVAL1 = NINT (V_FRAC)	GPAR0250	C	DIMENSION P (NPARDIM, NDMAX)	PSOR0090
C	IF (FLOAT (NVAL1) .LT. V_FRAC) THEN	GPAR0260	C		PSOR0100
C	NVAL2 = NVAL1 + 1	GPAR0270	C	DO 1000 I = 1, N - 1	PSOR0110
C	ELSE	GPAR0280	C	IMIN = I	PSOR0120
C	NVAL2 = NVAL1	GPAR0290	C	PMIN = P (IP, I)	PSOR0130
C	NVAL1 = NVAL2 - 1	GPAR0300	C	DO 1010 J = I + 1, N	PSOR0140
C	END IF	GPAR0310	C	IF (P (IP, J) .LT. PMIN) THEN	PSOR0150
C		GPAR0320	C	IMIN = J	PSOR0160
C	V_PER = V_FRAC - FLOAT (NVAL1)	GPAR0330	C	PMIN = P (IP, J)	PSOR0170
C	PARVAL = PDIST (IP, NVAL1) + V_PER*(PDIST (IP,	GPAR0340	C	END IF	PSOR0180
C	1 NVAL2) - PDIST (IP, NVAL1))	GPAR0350	C	1010 CONTINUE	PSOR0190
C		GPAR0360	C	IF (IMIN .NE. I) THEN	PSOR0200
C	RETURN	GPAR0370	C	P (IP, IMIN) = P (IP, I)	PSOR0210
C	END	GPAR0380	C	P (IP, I) = PMIN	PSOR0220
C		PCHE0010	C	END IF	PSOR0230
C		PCHE0020	C	1000 CONTINUE	PSOR0240
C		PCHE0030	C	RETURN	PSOR0250
C	SUBROUTINE P_CHECK (PSTAT, PDIST, NPARDIM, NDMAX, NPARD)	PCHE0040	C	END	PSOR0260
C		PCHE0050	C		MCIN0010
C	IMPLICIT DOUBLE PRECISION (A - H, O - Z)	PCHE0060	C		MCIN0020
C	IMPLICIT INTEGER*4 (I - N)	PCHE0070	C		MCIN0030
C		PCHE0080	C	SUBROUTINE MC_INP (TITLE)	MCIN0040
C	DIMENSION PSTAT (NPARDIM, 4), PDIST (NPARDIM, NDMAX)	PCHE0090	C		MCIN0050
C		PCHE0100	C	INCLUDE 'MC_TRLHS.INC'	MCIN0060
C	First go through and sort all raw distribution data; then calc	PCHE0110	C		MCIN0070
C	the mean for the distribution to use for deterministic runs.	PCHE0120	C	CHARACTER*72 TITLE	MCIN0080
C		PCHE0130	C		MCIN0090
C	DO 1000 IP = 1, NPARD	PCHE0140	C	Routine to read in the data for the Monte-Carlo transport simulation.	MCIN0100
C	IF (NINT (PSTAT (IP, 1)) .EQ. 4) THEN	PCHE0150	C		MCIN0110
C	NDIST = NINT (PSTAT (IP, 4))	PCHE0160	C	The following lists the parameter id for inputting distributions	MCIN0120
C	*****	PCHE0170	C	and spatially blocked parameters; PARID=1, 2, 7, and 17 are	MCIN0130
C	CALL P_SORT (PDIST, IP, NDIST, NPARDIM, NDMAX)	PCHE0180	C	fixed in space. The data are input on element basis, except for	MCIN0140
C	CALL P_MEAN (PDIST, PSTAT, IP, NDIST, NPARDIM, NDMAX)	PCHE0190	C	the initial concentrations CINIT and SINIT which are node-based.	MCIN0150
C	*****	PCHE0200	C		MCIN0160
C	END IF	PCHE0210	C	PARID=1 C_SRCE(IS), Source term concentration (pCI/L)	MCIN0170
C	1000 CONTINUE	PCHE0220	C	PARID=2 GAMMA(IS), Source decay constant (1/T)	MCIN0180
C		PCHE0230	C	PARID=3 CINIT(IN,IS), Initial mobile solute concentrations	MCIN0190
C	RETURN	PCHE0240	C	(pCI/L)	MCIN0200
C	END	PCHE0250	C	PARID=4 SINIT(IN,IS), Initial immobile solute concentrations	MCIN0210
C		PMEA0010	C	(pCI/L)	MCIN0220

```

C   PARID=5  GRAD(IE), Mobile zone hydraulic gradient (L/L)          MCIN0230
C   PARID=6  DISPL(IE), Mobile zone longitudinal dispersivity (L)    MCIN0240
C   PARID=7  DM, Mobile zone free-water diffusion coefficient (L**2/T) MCIN0250
C   PARID=8  TAU(IE), Mobile zone tortuosity                        MCIN0260
C   PARID=9  THETA(IE), Mobile zone porosity                        MCIN0270
C   PARID=10 THETA2(IE), Immobile zone porosity                    MCIN0280
C   PARID=11 BWIDTH(IE), Block width (L)                            MCIN0290
C   PARID=12 HYCOND(IE), Hydr. Cond. (L/T)                          MCIN0300
C   PARID=13 RHOB(IE), Bulk density (M/L**3)                        MCIN0310
C   PARID=14 RKD1(IE,IS), Distribution coefficient for mobile zone  MCIN0320
C                                     (L**3/M)                        MCIN0330
C   PARID=15 RKD2(IE,IS), Distribution coefficient for immobile zone MCIN0340
C                                     (L**3/M)                        MCIN0350
C   PARID=16 ALPHA(IE,IS), 1st-order mass transfer coefficient (1/T) MCIN0360
C   PARID=17 BETA(IS), 1st-order decay constant (1/T)               MCIN0370
C                                                                    MCIN0380
C                                                                    MCIN0390
C                                     <----- Record 1 MCIN0400
C   READ (5, '(A72)') TITLE                                         MCIN0410
C   WRITE (6, 6000) TITLE                                           MCIN0420
6000 FORMAT('          ++++++',//, MCIN0430
1  '          MC_MAIN      Version 5.0      9/27/95',//, MCIN0440
2  '          ++++++',//, MCIN0450
3  '10X,*** Record 1 ***',//, MCIN0460
4  '6X,A72) MCIN0470
C   TITLE is the title for the run, to be used for output. MCIN0480
C   C MCIN0490
C   C MCIN0500
C   C Read in fixed Monte Carlo and FE transport parameters. MCIN0510
C   C MCIN0520
C   C MCIN0530
C                                     <----- Record 2 MCIN0540
C   READ (5, *) MC, ISEED MCIN0550
C   WRITE (6, 6010) MC, ISEED MCIN0560
6010 FORMAT(//,10X,*** Record 2 ***',//, MCIN0570
1  ' Number of Monte Carlo or LHS realizations..... (MC)..=',I6,/, MCIN0580
2  ' Seed used for random number generator ..... (ISEED).=',I6) MCIN0590
C   MCIN0600
C   MC is the number of realizations to generate and run1: MCIN0610
C   MC = 0 Deterministic, do one run using mean values MCIN0620
C   MC = 1 Deterministic, do one run using mean values MCIN0630
C   MC > 1 Stochastic, do "mc" monte carlo runs, max = 50,000 MCIN0640
C   MCIN0650
C   ISEED is the seed used for the random number generator MCIN0660
C   used if MC > 1; Values should be positive integers. MCIN0670
C   MCIN0680
C   MCIN0690
C   CALL ERR_CHK (MC, MCMAX, 'MCMAX') MCIN0700
C   MCIN0710
C   MCIN0720
C   MCIN0730
C                                     <----- Record 3 MCIN0740
C   READ (5, *) NUMX, DELX, NUMT, DELT, TOFF, SCL_PERM MCIN0750
CRKW MCIN0760
C   DELXSAV = DELX MCIN0770
C   NUMXSAV = NUMX MCIN0780
C   DELTSAV = DELT MCIN0790
C   TMAX = ABS (DELT)*NUMT MCIN0800
CRKW MCIN0810
C   END MCIN0820
C   IF (TOFF .LE. 0.0) TOFF = 1.0E+30 MCIN0830
C   WRITE (6, 6020) NUMX, DELX, NUMT, DELT, TOFF, SCL_PERM MCIN0840
6020 FORMAT(//,10X,*** Record 3 ***',//, MCIN0850
1  ' Number of finite element nodes .....(NUMX).=',I6,/, MCIN0860
2  ' Uniform distance between nodes .....(DELX).=', MCIN0870
3  '1P,G10.3,/, MCIN0880
4  ' Number of time steps .....(NUMT).=',I6,/, MCIN0890
5  ' Time step size .....(DELT).=', MCIN0900
6  '1P,G10.3,/, MCIN0910
7  ' Time for source shut-off.....(TOFF).=', MCIN0920
8  ' G10.3,/, MCIN0930
9  ' Scaling factor for hydraulic conductivity..(SCL_PERM).=', MCIN0940
A  ' G10.3) MCIN0950
C   IF (DELT .LT. 0) WRITE (6, 6030) MCIN0960
6030 FORMAT(' DELT < 0 INDICATES THAT DELT AND NUMT WILL BE MCIN0970
1 / RECALCULATED TO PROVIDE A COURANT NUMBER = 1') MCIN0980
C   NUMX = The number of FE nodes in the domain. MCIN0990
C   DELX = The uniform spacing between nodes MCIN1000
C   Note: L = Length of domain = DELX*(NUMX-1). MCIN1010
C   NUMT = Number of time steps. MCIN1020
C   DELT = Duration of time steps. MCIN1030
C   Note: TMAX = Duration of simulation = DELT*(NUMT). MCIN1040
C   Because of time looping, first step is initial T=0. thus, MCIN1050
C   increment total number of calculational time steps. MCIN1060
C   Set DELT < 0 for code to calculate DELT and NUMT for Courant = 1 MCIN1070
C   ++++++ MCIN1080
C   CALL ERR_CHK (NUMX, NN, 'NN') MCIN1090
C   ++++++ MCIN1100
C   MCIN1110
C   ++++++ MCIN1120
C   CALL ERR_CHK (NUMT, NSTPS, 'NSTPS') MCIN1130
C   ++++++ MCIN1140
C   MCIN1150
C   WRITE (6, 6040) (NUMX - 1)*ABS (DELX), DABS (NUMT*DELT) MCIN1160
6040 FORMAT(/, MCIN1170
1  ' Total grid distance .....(NUMX-1 * DELX).=', MCIN1180
2  '1P,G10.3,/, MCIN1190
3  ' Total simulation time .....(NUMT * DELT).=', MCIN1200
4  ' G10.3) MCIN1210
C   MCIN1220
C   MCIN1230
C                                     <----- Record 4 MCIN1240
C   READ (5, *) INBOUN MCIN1250
C   WRITE (6, 6050) INBOUN MCIN1260
6050 FORMAT (//,10X,*** Record 4 ***',//, MCIN1270
1  ' Inflow node boundary condition type:',//, MCIN1280
2  ' INBOUN = 1 , Constant concentration',//, MCIN1290
3  ' INBOUN = 2 , Constant flux VC-DDC/DX = VC_SRCE',//, MCIN1300
4  ' Index inflow boundary ..... (INBOUN).=',I6) MCIN1310
C   MCIN1320
C   MCIN1330
C   INBOUN = Inflow node boundary condition type: MCIN1340
C   INBOUN = 1 , Constant concentration MCIN1350
C   INBOUN = 2 , Constant flux VC-DDC/DX = VC_SRCE MCIN1360
C   MCIN1370
C   MCIN1380
C                                     <----- Record 5 MCIN1390
C   READ (5, *) NUMOBS MCIN1400
C   WRITE (6, 6060) NUMOBS MCIN1410
6060 FORMAT(//,10X,*** Record 5 ***',//, MCIN1420
1  ' Number of observation nodes ( >= 1).....(NUMOBS).=',I6) MCIN1430
C   MCIN1440
C   ++++++ MCIN1450
C   CALL ERR_CHK (NUMOBS, NOBS, 'NOBS') MCIN1460
C   ++++++ MCIN1470
C   MCIN1480
C   IF (NUMOBS .EQ. 0) THEN MCIN1490
C   NUMOBS = 1 MCIN1500
C   NOBSLOC (1) = NUMX MCIN1510
C   ELSE MCIN1520

```

C		<----- Record 6	MCIN1530	CALL ERR_CHK (NSPEC, NS, 'NS')	MCIN2010
	READ (5, *) (NOBSLOC (IO), IO = 1, NUMOBS)		MCIN1540	*****	MCIN2020
	WRITE (6, 6070) (NOBSLOC (IO), IO = 1, NUMOBS)		MCIN1550		MCIN2030
6070	FORMAT(//,10X,'*** Record 6 ***',//,		MCIN1560		MCIN2040
1	' List of observation nodes:',//,(20I5))		MCIN1570	DO 1010 IP = 1, NPARD	MCIN2050
END IF			MCIN1580	*****	MCIN2060
C			MCIN1590	CALL PDREAD (IP, PSTAT, PDIST, NDMAX, NPARDIM,	MCIN2070
C	NUMOBS = The number of receptor/observation locations to simulate		MCIN1600	CFSP NOVEMBER 1995 NPARD CHANGED TO NPARDIT	
C	if numobs=0 then numobs is set to 1 and NOBSLOC=NUMX.		MCIN1610	1 IPARID, NPARDID, NUMX, NPARDIT)	MCIN2080
C	NOBSLOC() = The node number(s) where the BTC(S) is(are)		MCIN1620	CFSP END	
C	generated. if NOBSLOC()=0, Then NOBSLOC is reset to NUMX.		MCIN1630	*****	MCIN2090
C			MCIN1640	1010 CONTINUE	MCIN2100
	DO 1000 IO = 1, NUMOBS		MCIN1650		MCIN2110
	IF (NOBSLOC (IO) .EQ. 0 .OR. NOBSLOC (IO) .GT.		MCIN1660	C Check input data/distributions:	MCIN2120
1	NUMX) NOBSLOC (IO) = NUMX		MCIN1670	C 1) All parameters must have values for all model nodes.	MCIN2130
CRKW			MCIN1680	C 2) All species must have input for PARID=1, 2, 5, 9, 11, 12.	MCIN2140
	NOBSSAV (IO) = NOBSLOC (IO)		MCIN1690	C 3) The parent must have input for PARID=1 to 12.	MCIN2150
CRKW			MCIN1700		MCIN2160
	1000 CONTINUE		MCIN1710	CFSP NOVEMBER, 1995 NPARD CHANGED TO NPARDIT	
C			MCIN1720	DO 1020 ITYPE = 1, NPARDIT	MCIN2170
C	Read in species-specific parameter distributions.		MCIN1730	CFSP END	
C			MCIN1740		MCIN2180
C		<----- Record 7	MCIN1750	IF (ITYPE .EQ. 3 .OR. ITYPE .EQ. 4) THEN	MCIN2190
CFSP NOVEMBER, 1995	NFSP ADDED TO READ AND WRITE			NX = NUMX	MCIN2200
	READ (5, *) NPARD, NSPEC, NFSP, nsens, NLHS, NRC			ELSE	MCIN2210
MCIN1760				NX = NUMX - 1	MCIN2220
	WRITE (6, 6080) NPARD, NSPEC, NFSP, nsens, NLHS, NRC			END IF	MCIN2230
MCIN1770					MCIN2240
CFSP END				IF (ITYPE .EQ. 1 .OR. ITYPE .EQ. 2 .OR. ITYPE .EQ. 3	MCIN2250
	6080 FORMAT(//,10X,'*** Record 7 ***',//,		MCIN1780	.OR. ITYPE .EQ. 4 .OR. ITYPE .EQ. 14	MCIN2260
	1 ' Number of parameter distributions read(NPARD).='I6,/,		MCIN1790	.OR. ITYPE .EQ. 15 .OR. ITYPE .EQ. 16	MCIN2270
	2 ' Number of chemical species(NSPEC).='I6,/,		MCIN1800	.OR. ITYPE .EQ. 17) THEN	MCIN2280
	3 ' Number of flow sensitivity parameters.....(NFSP).='I6,/,		MCIN1800		MCIN2290
	4 ' Number of sensitivity runs(NSENS).='I6,/,			DO 1030 ISPEC = 1, NSPEC	MCIN2300
	5 ' Number of variables sampled using LHS.....(NLHS).='I6,/,			DO 1040 IX = 1, NX	MCIN2310
	6 ' Recharge sensitivity control (0=no, 1=yes).....(NRC).='I6,			NODECHK (IX) = 0	MCIN2320
ckrr				CONTINUE	MCIN2330
	if (nsens.ge.1) open(21,file='transen.dat')				MCIN2340
	IF (NLHS.GE.1) open(22,file='LHS.MCT')			DO 1050 IP = 1, NPARD-NFSP	MCIN2350
C				IF (IPARID (IP, 1) .EQ. ITYPE .AND.	MCIN2360
CFSP NOVEMBER, 1995				IPARID (IP, 2) .EQ. ISPEC) THEN	MCIN2370
	NPARDID = NPARDIT + NFSP			DO 1060 IX = IPARID (IP, 3), IPARID (IP, 4)	MCIN2380
CFSP END				NODECHK (IX) = 1	MCIN2390
C			MCIN1810	CONTINUE	MCIN2400
C	NPARD = The number of parameter distributions to read in.		MCIN1820	END IF	MCIN2410
C			MCIN1830	CONTINUE	MCIN2420
C	NSPEC = The number of species to simulate.		MCIN1840		MCIN2430
C			MCIN1850	ISUM = 0	MCIN2440
C	Currently, NPARD must be .GE. NPARDID.		MCIN1860	DO 1070 IX = 1, NX	MCIN2450
C			MCIN1870	ISUM = ISUM + NODECHK (IX)	MCIN2460
	IF (NPARD .LT. NPARDID) WRITE (*, 6090) NPARDID, NPARD, NPARDIM		MCIN1880	CONTINUE	MCIN2470
	IF (NPARD .LT. NPARDID) WRITE (6, 6090) NPARDID, NPARD, NPARDIM		MCIN1890		MCIN2480
6090	FORMAT(//,' ERROR - Insufficient number of distributions',//,		MCIN1900	IF (ISUM .NE. NX) THEN	MCIN2490
1	' The data files must supply at least 'I3,' distributions.',//,		MCIN1910	WRITE (*, *) '*****',	MCIN2500
2	' Currently the data file only contains 'I3,' distributions.',//,		MCIN1920	*****	MCIN2510
3	' There can be as many as 'I3,' distributions.',//,		MCIN1930	WRITE (*, *) '*****FATAL***',	MCIN2520
4	' Be sure to specify at one of each type required.')		MCIN1940	*****	MCIN2530
C			MCIN1950	WRITE (*, *) ' '	MCIN2540
C	*****		MCIN1960	WRITE (*, *) ' '	MCIN2550
C	CALL ERR_CHK (NPARD, NPARDIM, 'NPARDIM')		MCIN1970	' ALL SPATIAL LOCATIONS ARE NOT COVERED FOR:'	MCIN2560
C	*****		MCIN1980	WRITE (*, *) ' PARAMETER TYPE = ', ITYPE	MCIN2570
C			MCIN1990	WRITE (*, *) ' SPECIES NUMBER = ', ISPEC	MCIN2580
C	*****		MCIN2000	WRITE (*, *) ' '	MCIN2590
				WRITE (*, *) ' EXECUTION HALTED.....'	MCIN2600
				WRITE (*, *) '*****',	MCIN2610

1	*****	MCIN2620	6 ' UNIT 6 - Primary output file.....(IOU6-out)..=',I5,,	MCIN3240
	WRITE (*, *) '*****FATAL***',	MCIN2630	7 ' UNIT 7 - Maximum at receptors.....(IOU7-cmx)..=',I5,,	MCIN3250
1	*****	MCIN2640	8 ' UNIT 8 - Time of arrival of maximums.....(IOU8-atm)..=',I5,,	MCIN3260
	STOP	MCIN2650	9 ' UNIT 9 - Summary of statistics.....(IOU9-smx)..=',I5,,	MCIN3270
	END IF	MCIN2660	A ' UNIT 10 - BTC using mean parameter(IOU10-btc)..=',I5,,	MCIN3280
1030	CONTINUE	MCIN2670	B ' UNIT 11 - Parameter realizations.....(IOU11-par)..=',I5,,	MCIN3290
	ELSE	MCIN2680	C ' UNIT 12 - Tabular maximum concs.....(IOU12-txt)..=',I5,,	MCIN3300
C		MCIN2690	D ' UNIT 20 - BTC (binary file).....(IOU20-bin)..=',I5)	MCIN3310
	DO 1080 IX = 1, NX	MCIN2700		MCIN3320
	NODECHK (IX) = 0	MCIN2710	C All data in main input file read in, return to	
1080	CONTINUE	MCIN2720	C main program unless Latin Hypercube Sampling (LHS) is utilized.	
C		MCIN2730		
CFSP NOVEMBER, 1995 NPARD CHANGED TO NPARD-NFSP TO DISABLE ERROR				
C CHECKING WHERE NOT APPROPRIATE FOR FLOW PARAMETERS				
DO 1090 IP = 1, NPARD-NFSP	MCIN2740			
CFSP END				
	IF (IPARID (IP, 1) .EQ. ITYPE .AND.	MCIN2750		
1	IPARID (IP, 2) .EQ. 1) THEN	MCIN2760	IF (NLHS.GE.1) THEN	
	DO 1100 IX = IPARID (IP, 3), IPARID (IP, 4)	MCIN2770	DO 6200 ISAMP=1,MC+1	
	NODECHK (IX) = 1	MCIN2780	READ(22,*) (PLHS(ISAMP,ILHS),ILHS=1,NLHS)	
1100	CONTINUE	MCIN2790	6200 CONTINUE	
	END IF	MCIN2800	write (6,*) ' file lhs.mct read'	
1090	CONTINUE	MCIN2810	ENDIF	MCIN3320
C		MCIN2820		
	ISUM = 0	MCIN2830	C All LHS data in input file read in, return to	
	DO 1110 IX = 1, NX	MCIN2840	C main program.	
	ISUM = ISUM + NODECHK (IX)	MCIN2850	C	
1110	CONTINUE	MCIN2860	1120 CONTINUE	MCIN3350
C		MCIN2870	RETURN	MCIN3370
	IF (ISUM .NE. NX) THEN	MCIN2880	END	MCIN3380
	WRITE (*, *) '*****	MCIN2890		PDRE0030
1	*****	MCIN2900	SUBROUTINE SEN_INP	
	WRITE (*, *) '*****FATAL***',	MCIN2910		
1	*****	MCIN2920	INCLUDE 'MC_TRLHS.INC'	
	WRITE (*, *) ' '	MCIN2930		
	WRITE (*, *)	MCIN2940		
1	' ALL SPATIAL LOCATIONS ARE NOT COVERED FOR:'	MCIN2950	CFSP NOVEMBER 1995	
	WRITE (*, *) ' PARAMETER TYPE = ', ITYPE	MCIN2960	C Read flow sensitivity data from units 13 - 16	
	WRITE (*, *) ' SPECIES NUMBER = ', ISPEC	MCIN2970	NX = NUMX -1	
	WRITE (*, *) ' '	MCIN2980	IF (NRC.LE.0) THEN	
	WRITE (*, *) ' EXECUTION HALTED....'	MCIN2990	NFSPP=NFSP	
1	*****	MCIN3000	ELSE	
	WRITE (*, *) '*****FATAL***',	MCIN3010	NFSPP=NFSP-1	
1	*****	MCIN3020	ENDIF	
	STOP	MCIN3030	DO 1130 IDIR = 1,3	
	END IF	MCIN3040	READ (13) (VVECT(IE,IDIR), IE=1, NX)	
	END IF	MCIN3050	1130 CONTINUE	
C		MCIN3060	write (6,*) ' file *.vel read'	
1020	CONTINUE	MCIN3070	IU = 13	
C		MCIN3080	DO 1150 IDIR = 1,3	
C		MCIN3090	IU = IU + 1	
C	Read output control variables.	MCIN3100	DO 1140 IE = 1,NX	
C		MCIN3110	READ (IU) (SENS (IE, IFSP, IDIR), IFSP = 1, NFSPP)	
C		MCIN3120	1140 CONTINUE	
C		MCIN3130	write (6,*) ' sens file number ',idir, ' read'	
	----- Record 9	MCIN3140	1150 CONTINUE	
	READ (5, *, END = 1120) IOU6, IOU7, IOU8,	MCIN3150	CFSP END	
1	IOU9, IOU10, IOU11, IOU12,	MCIN3160	RETURN	
2	IOU20	MCIN3170	END	
	WRITE (6, 6100) IOU6, IOU7, IOU8, IOU9, IOU10, IOU11, IOU12, IOU20	MCIN3180		PDRE0010
6100	FORMAT (//,10X,'*** Record 9 ***',//,	MCIN3190		PDRE0020
1	' Output control:',//,	MCIN3200		PDRE0030
2	0 for no output',//,	MCIN3210	SUBROUTINE PDREAD (IP, PSTAT, PDIST, NDMAX, NPARDIM, IPARID,	PDRE0040
3	1 to write just mean simulation',//,	MCIN3220	1 NPARDIM, NUMX, NPARIT)	PDRE0050
4	n to write every nth Monte Carlo realization',//,	MCIN3230		PDRE0060
5	or time step',//,			

	IMPLICIT DOUBLE PRECISION (A - H, O - Z)	PDRE0070	1	I6,	I6,	I6,	I7,	I8, 3X,	PDRE0560
	IMPLICIT INTEGER*4 (I - N)	PDRE0080	2	1PG10.3, 3X,	G10.3,3X,A65)				PDRE0570
C		PDRE0090	6011	FORMAT (					PDRE0550
	DIMENSION PSTAT (NPARDIM, 4), PDIST (NPARDIM, NDMAX)	PDRE0100	1	I6,	I6,	I6,	I7,	I8, 3X,	PDRE0560
	DIMENSION IPARID (NPARDIM, 4)	PDRE0110	CFSP	NOVEMBER 1995					
	CHARACTER*65 TPARID (17)	PDRE0120	2	1PG10.3, 3X,	G10.3,3X,A42, I2)				PDRE0570
CFSP	NOVEMBER, 1995		CFSP	END					
	CHARACTER*42 FPARID		C						PDRE0580
CFSP	END		C						PDRE0590
	DATA TPARID/	PDRE0130	C	IPARID(,)	Stores information on the type of parameter and its				PDRE0600
1	'C_SRCE(IS): Source term concentration (pCI/L)	' ,PDRE0140	C	spatial location:					PDRE0610
2	'GAMMA(IS): Source decay constant (1/T)	' ,PDRE0150	C						PDRE0620
3	'CINIT(IN, IS)	PDRE0160	C	IPARID(IP,1) =	Parameter id for parameter IP; the parameter id				PDRE0630
4	: Initial mobile solute concentration (pCI/L) ' ,	PDRE0170	C		corresponds to the parid types listed at the				PDRE0640
5	'SINIT(IN, IS)	PDRE0180	C		beginning of subroutine MC_INP.				PDRE0650
6	: Initial immobile solute concentration (pCI/L) ' ,	PDRE0190	C						PDRE0660
7	'GRAD(IE): Mobile zone hydraulic gradient (L/L) or velocity ' ,PDRE0200		C	IPARID(IP,2) =	Species number for the parameter; only CINIT				PDRE0670
8	'DISPL(IE): Mobile zone longitudinal dispersivity (L)	' ,PDRE0210	C	(PARID=1), SINIT (PARID=2), RKD (PARID=5),					PDRE0680
9	'DM: Mobile zone free-water diffusion coefficient (L**2/T)	' ,PDRE0220	C	ALPHA (PARID=9), BETA (PARID=11), and					PDRE0690
A	'TAU(IE): Mobile zone tortuosity ' ,PDRE0230		C	C_SRCE are specified for species other than					PDRE0700
B	'THETA1(IE): Mobile zone porosity ' ,PDRE0240		C	species number 1 (the parent).					PDRE0710
C	'THETA2(IE): Immobile zone porosity ' ,PDRE0250		C						PDRE0720
D	'BWIDTH(IE): Block width (L)	' ,PDRE0260	C	IPARID(IP,3) =	FE model starting node or element for the				PDRE0730
E	'HYCOND(IE): Average hydraulic conductivity (L/T)	' ,PDRE0270	C	parameter block					PDRE0740
F	'RHOB(IE): Bulk density (M/L**3)	' ,PDRE0280	C	defined by parameter IP; note that flow is in					PDRE0750
G	'RKD1(IE, IS)	PDRE0290	C	the ascending order and the source					PDRE0760
H	: Distribution coefficient mobile zone (L**3/M) ' ,	PDRE0300	C	location is at node or element 1.					PDRE0770
I	'RKD2(IE, IS)	PDRE0310	C						PDRE0780
J	: Distribution coefficient immobile zone (L**3/M) ' ,	PDRE0320	C	IPARID(IP,4) =	FE model ending node or element for the				PDRE0790
&	'ALPHA(IE, IS):1st-	PDRE0330	C	parameter block defined by parameter IP.					PDRE0800
&	order mass transfer coefficient (1/T) ' ,	PDRE0340	C						PDRE0810
&	'BETA(IS), 1st-order decay constant (1/T)	' /PDRE0350	C	IDIST Is the type of distribution flag					PDRE0820
CFSP	NOVEMBER, 1995		C						PDRE0830
	DATA FPARID/'FSP(IX,IFP): Flow sensitivity parameter #' /		C	IDIST = 0	Deterministic, value is "DMEAN".				PDRE0840
CFSP	END		C						PDRE0850
C		PDRE0360	C	IDIST = 1	Normal distribution with mean "DMEAN" and standard				PDRE0860
C		PDRE0370	C	deviation "DSTD"					PDRE0870
C		PDRE0380	C						PDRE0880
	READ (5, *) IPARID (IP, 1), IPARID (IP, 2),	PDRE0390	C	IDIST = 2	Lognormal distributin with (log) mean "DMEAN" and (log)				PDRE0890
1	IPARID (IP, 3), IPARID (IP, 4),	PDRE0400	C	standard deviation of "DSTD".					PDRE0900
2	IDIST, DMEAN, DSTD	PDRE0410	C						PDRE0910
C		PDRE0420	C	IDIST = 3	Uniform distribution with min value of "DMEAN" and max				PDRE0920
	IF (IP .EQ. 1) WRITE (6, 6000)	PDRE0430	C	value of "DSTD".					PDRE0930
CFSP	NOVEMBER, 1995 MODIFICATIONS ON OUTPUT		C						PDRE0940
	IF (IPARID (IP, 1) .LE. NPARIT) THEN		C	IDIST = 4	Distribution consists of raw data, to be entered in				PDRE0950
	WRITE (6, 6010) IPARID (IP, 1), IPARID (IP, 2),	PDRE0440	C	the following records; does not have to be in					PDRE0960
1	IPARID (IP, 3), IPARID (IP, 4),	PDRE0450	C	any order.					PDRE0970
2	IDIST, DMEAN, DSTD,	PDRE0460	C						PDRE0980
3	TPARID (IPARID (IP, 1))	PDRE0470	C	Make sure the parameter type is within range.					PDRE0990
	ELSE		C						PDRE1000
	IFPAR = IPARID (IP, 1) - NPARIT								

	IPARID (IP, 4) = NUMX - 1	PDRE1160	INCLUDE 'MC_TRLHS.INC'	PRNE0060
	END IF	PDRE1170		PRNE0070
C			C Echo the values for each element.	PRNE0080
CBHL		PDRE1180		PRNE0090
C	IF (IDIST .LT. -3 .OR. IDIST .GT. 4) THEN	PDRE1190		PRNE0100
CBHL	IF (IDIST .GT. 4) THEN		WRITE (6, 6000)	PRNE0110
	WRITE (*, *) ' '	PDRE1200	6000 FORMAT(///,10X,'Summary List of Element-based Parameters',/,	PRNE0120
	WRITE (*, *) ' ERROR IN SPECIFYING THE TYPE OF PARAMETER ',	PDRE1210	1 10X,'*****',/,	PRNE0130
1	'DISTRIBUTION'	PDRE1220	2 T2 ,'Element' ,T10,'Distance to', T25,	PRNE0140
	WRITE (*, *) ' PARAMETER NUMBER ', IP	PDRE1230	3 'VEL' ,',DISPL' ,',	PRNE0150
	WRITE (*, *) ' '	PDRE1240	4 'TAU' ,',THETA1' ,',THETA2' ,',BWIDTH' ,',	PRNE0160
	WRITE (*, *) ' EXECUTION HALTED.....'	PDRE1250	5 'FWIDTH' ,',RHOB' ,',/,	PRNE0170
	STOP	PDRE1260	6 T2 ,'Number' ,',T10,'Centroid' ,', T25,/,	PRNE0180
	END IF	PDRE1270	7 T2 ,'_____' ,',T10,'_____' ,', T24,	PRNE0190
C		PDRE1280	8 '_____' ,',_____' ,',	PRNE0200
C	Conversion for lognormal input, where input is in the	PDRE1290	9 '_____' ,',_____' ,',_____' ,',_____' ,',	PRNE0210
C	log-10 transformed range.	PDRE1300	A '_____' ,',_____' ,',/,)	PRNE0220
C	IF (IDIST .EQ. 2) THEN	PDRE1310	DO 1000 IX = 1, NUMX - 1	PRNE0230
C	Multiply to convert from Log-10 to Natural Log.	PDRE1320	XL = (REAL (IX) + 0.5)*DELX	PRNE0240
	DMEAN = DMEAN*2.30259	PDRE1330	WRITE (6, 6010) IX, XL, VEL (IX), DISPL (IX),	PRNE0250
	DSTD = DSTD*2.30259	PDRE1340	1 TAU (IX), THETA1 (IX), THETA2 (IX),	PRNE0260
C	Convert the log of the mean to the mean of the log.	PDRE1350	2 BWIDTH (IX), FWIDTH (IX), RHOB (IX)	PRNE0270
C	DMEAN = EXP (DMEAN + 0.5D0*DSTD*DSTD)	PDRE1360	1000 CONTINUE	PRNE0280
C	Convert the log of the standard deviation to the	PDRE1370	6010 FORMAT(I5,T7,1P,E15.3,8E10.2)	PRNE0290
C	standard deviation of the log.	PDRE1380	RETURN	PRNE0300
	DSTD = SQRT (DMEAN*DMEAN*(EXP (DSTD*DSTD) - 1))	PDRE1390	END	PRNE0310
	END IF	PDRE1400		PRNS0010
C		PDRE1410		PRNS0020
C	User-defined.	PDRE1420		PRNS0030
	IF (IDIST .NE. 4) THEN	PDRE1430	SUBROUTINE PRN_SPE	PRNS0040
	PSTAT (IP, 1) = FLOAT (IDIST)	PDRE1440		PRNS0050
	PSTAT (IP, 2) = DMEAN	PDRE1450	INCLUDE 'MC_TRLHS.INC'	PRNS0060
	PSTAT (IP, 3) = DSTD	PDRE1460		PRNS0070
	PSTAT (IP, 4) = 0.0	PDRE1470	C Echo the values for each species.	PRNS0080
	ELSE	PDRE1480		PRNS0090
	READ (5, *) NDIST	PDRE1490	WRITE (6, 6000)	PRNS0100
	WRITE (6, *) NDIST	PDRE1500	6000 FORMAT(///,10X,'Summary List of Species-based Parameters',/,	PRNS0110
	IF (NDIST .LE. 10) THEN	PDRE1510	1 10X,'*****',/,	PRNS0120
	WRITE (*, *) ' '	PDRE1520	2 T2 ,'Species' ,', T14,	PRNS0130
	WRITE (*, *) ' NUMBER OF PARAMETER S MUST EXCEED 10'	PDRE1530	3 'C_SRCE' ,',GAMMA' ,',BETA' ,',/,	PRNS0140
	STOP	PDRE1540	4 T2 ,'_____' ,', T12,	PRNS0150
	ELSE IF (NDIST .GT. NDMAX) THEN	PDRE1550	5 '_____' ,',_____' ,',_____' ,',/,)	PRNS0160
	WRITE (*, *) ' '	PDRE1560		PRNS0170
	WRITE (*, *) ' NUMBER OF VALUES IN DISTRIBUTION FOR ',	PDRE1570	DO 1000 IS = 1, NSPEC	PRNS0180
1	' PARAMETER ', IP, ' IS GREATER THAN THE MAXIMUM ',	PDRE1580	WRITE (6, 6010) IS, C_SRCE (IS), GAMMA (IS), BETA (IS)	PRNS0190
2	' DIMENSION ', NDMAX	PDRE1590	1000 CONTINUE	PRNS0200
	WRITE (*, *) ' '	PDRE1600	6010 FORMAT(I5,T10,3E10.2)	PRNS0210
	WRITE (*, *) ' EXECUTION HALTED.....'	PDRE1610	RETURN	PRNS0220
	STOP	PDRE1620	END	PRNS0230
	ELSE	PDRE1630		PRNO0010
	PSTAT (IP, 1) = FLOAT (IDIST)	PDRE1640		PRNO0020
	PSTAT (IP, 4) = FLOAT (NDIST)	PDRE1650	SUBROUTINE PRN_OTH	PRNO0030
	READ (5, *) (PDIST (IP, ID), ID = 1, NDIST)	PDRE1660		PRNO0040
	END IF	PDRE1670	INCLUDE 'MC_TRLHS.INC'	PRNO0050
	END IF	PDRE1680		PRNO0060
C		PDRE1690	C Echo the values for species and spatial variable values.	PRNO0070
	RETURN	PDRE1700		PRNO0080
	END	PDRE1710		PRNO0090
C		PRNE0010	WRITE (6, 6000)	PRNO0100
C		PRNE0020	6000 FORMAT(///,	PRNO0110
C	SUBROUTINE PRN_ELE	PRNE0030	1 10X,'Summary List of Species and Element-based Parameters',/,	PRNO0120
		PRNE0040	2 10X,'*****',/,)	PRNO0130
C		PRNE0050		PRNO0140
			WRITE (6, 6010)	PRNO0150
			6010 FORMAT(/,	PRNO0160

1 T2 , 'Element' , T10, 'Distance to' , T25,	PRNO0170	C value returned by the subroutine. Best results are obtained by	MRAN0210
2 'RKD1 (I), I=1, NSPEC)' ,	PRNO0180	C making the initial call with a seed of your choice and then setting	MRAN0220
3 T2 , 'Number' , T10, 'Centroid' , T25, /,	PRNO0190	C the seed to '0' for all subsequent calls.	MRAN0230
4 T2 , '_____ ' , T10, '_____ ' , T24,	PRNO0200		MRAN0240
5 ' _____ ' , /)	PRNO0210		MRAN0250
C	PRNO0220		MRAN0260
DO 1000 IX = 1, NUMX - 1	PRNO0230	C Test to see if 55 odd integers must be generated.	MRAN0270
XL = (REAL (IX) + 0.5)*DELX	PRNO0240		MRAN0280
WRITE (6, 6020) IX, XL, (RKD1 (IX, IS), IS = 1, NSPEC)	PRNO0250	IF ((SEED .LE. 0) .AND. (NSEED .GE. 1)) GO TO 1000	MRAN0290
1000 CONTINUE	PRNO0260	NSEED = SEED	MRAN0300
C	PRNO0270	IF (SEED .LE. 0) NSEED = 5731	MRAN0310
WRITE (6, 6030)	PRNO0280		MRAN0320
6030 FORMAT(/,	PRNO0290	C Generate 55 odd integers in array ITAB().	MRAN0330
1 T2 , 'Element' , T10, 'Distance to' , T25,	PRNO0300		MRAN0340
2 'RKD2 (I), I=1, NSPEC)' ,	PRNO0310	DO 1010 I = 1, 55	MRAN0350
3 T2 , 'Number' , T10, 'Centroid' , T25, /,	PRNO0320	RN1 = MOD (NSEED*9069, 32768)	MRAN0360
4 T2 , '_____ ' , T10, '_____ ' , T24,	PRNO0330	IF (MOD (RN1, 2) .EQ. 0) RN1 = RN1 - 1	MRAN0370
5 ' _____ ' , /)	PRNO0340	ITAB (I) = RN1	MRAN0380
C	PRNO0350	NSEED = RN1	MRAN0390
DO 1010 IX = 1, NUMX - 1	PRNO0360	1010 CONTINUE	MRAN0400
XL = (REAL (IX) + 0.5)*DELX	PRNO0370		MRAN0410
WRITE (6, 6020) IX, XL, (RKD2 (IX, IS), IS = 1, NSPEC)	PRNO0380	C Generate "N" random components for the vector "VV".	MRAN0420
1010 CONTINUE	PRNO0390		MRAN0430
C	PRNO0400	N1 = 0	MRAN0440
WRITE (6, 6040)	PRNO0410	N2 = 24	MRAN0450
6040 FORMAT(/,	PRNO0420	1000 CONTINUE	MRAN0460
1 T2 , 'Element' , T10, 'Distance to' , T25,	PRNO0430	DO 1020 I = 1, N	MRAN0470
2 'ALPHA (I), I=1, NSPEC)' ,	PRNO0440	ITAB (55 - N1) = MOD (ITAB (55 - N2)*ITAB (55 - N1), 32768)	MRAN0480
3 T2 , 'Number' , T10, 'Centroid' , T25, /,	PRNO0450	NSEED = ITAB (55 - N1)/2	MRAN0490
4 T2 , '_____ ' , T10, '_____ ' , T24,	PRNO0460	VV (I) = FLOAT (NSEED)/FLOAT (15384)	MRAN0500
5 ' _____ ' , /)	PRNO0470	N1 = MOD (N1 + 1, 55)	MRAN0510
C	PRNO0480	N2 = MOD (N2 + 1, 55)	MRAN0520
DO 1020 IX = 1, NUMX - 1	PRNO0490	1020 CONTINUE	MRAN0530
XL = (REAL (IX) + 0.5)*DELX	PRNO0500	IF (SEED .GT. 0) SEED = NSEED	MRAN0540
WRITE (6, 6020) IX, XL, (ALPHA (IX, IS), IS = 1, NSPEC)	PRNO0510	RETURN	MRAN0550
1020 CONTINUE	PRNO0520	END	MRAN0560
C	PRNO0530		GINV0010
6020 FORMAT(I5,T7,1P,E15.3,4E10.2)	PRNO0540		GINV0020
WRITE (6, 6050) DM	PRNO0550		GINV0030
6050 FORMAT(///,	PRNO0560	DOUBLE PRECISION FUNCTION GINV (P)	GINV0040
1 ' Mobile zone free-water diffusivity (DM).= ' ,	PRNO0570		GINV0050
2 1P,G10.3)	PRNO0580	C Function to compute the inverse of standard gaussian cumulative	GINV0060
C	PRNO0590	distribution function. Uses numerical approximation from:	GINV0070
RETURN	PRNO0600		GINV0080
END	PRNO0610	Abramovitz and Stegun, 1972. Handbook of Mathematical Functions	GINV0090
C	MRAN0010	10th Printing, National Bureau of Standards, p. 933.	GINV0100
C	MRAN0020		GINV0110
C	MRAN0030		GINV0120
SUBROUTINE MRAND (SEED, N, VV)	MRAN0040	C Coefficients of approximation.	GINV0130
C	MRAN0050		GINV0140
IMPLICIT DOUBLE PRECISION (A - H, O - Z)	MRAN0060	IMPLICIT DOUBLE PRECISION (A - H, O - Z)	GINV0150
IMPLICIT INTEGER*4 (I - N)	MRAN0070	DATA C0/2.515517/, C1/.802853/, C2/.010328/	GINV0160
C	MRAN0080	DATA D1/1.432788/, D2/.189269/, D3/.001308/	GINV0170
DIMENSION VV (N)	MRAN0090		GINV0180
COMMON /MRANDC/ ITAB (55), N1, N2, NSEED	MRAN0100	C Values for + and - infinity.	GINV0190
INTEGER*4 RN1, SEED	MRAN0110		GINV0200
C	MRAN0120	DATA GNEG/ - 1E32/, GPOS/1E32/	GINV0210
C MARSAGLIAS random number generator. This random number generator	MRAN0130	DATA GNEG/-1E10/, GPOS/1E10/	GINV0220
generates random numbers in]0,1[. Note that if the seed value is	MRAN0140		GINV0230
zero on the first call, a default value of 5731 will be used in a	MRAN0150	C Check for probability = 0 or 1.	GINV0240
linear congruential generator to generate 55 odd integers for the	MRAN0160		GINV0250
array 'ITAB()'. These values are preserved by the common statement,	MRAN0170	IF (P .GT. 0.) GO TO 1000	GINV0260
so that they may be used in subsequent calls by setting the seed to	MRAN0180	GINV = GNEG	GINV0270
zero. If the value of 'SEED' is greater than zero in a call to the	MRAN0190	GO TO 1010	GINV0280
subroutine, then the array 'ITAB' will be initialized and a new seed	MRAN0200	1000 CONTINUE	GINV0290

IF (P .LT. 1) GO TO 1020	GINV0300	6000 FORMAT(//,	ERRC0140
GINV = GPOS	GINV0310	1 '*****',//,	ERRC0150
GO TO 1010	GINV0320	2 ' FATAL ERROR: ',//,	ERRC0160
C	GINV0330	3 ' The size of ', (A), ' in the PARAMETER',/,	ERRC0170
C 0<P<1 ==> Approximate the function.	GINV0340	4 ' STATEMENT in file MC_TRNSF.INC has been exceeded',//,	ERRC0180
C	GINV0350	5 ' The size is currently set at ',I8,/,	ERRC0190
1020 CONTINUE	GINV0360	6 ' Edit and increase this to ',I8,/,	ERRC0200
PP = P	GINV0370	7 '*****')	ERRC0210
IF (PP .GE. 0.5) PP = 1. - PP	GINV0380	C STOP	ERRC0220
T = SQRT (LOG (1./(PP*PP)))	GINV0390	C END	ERRC0230
T2 = T*T	GINV0400		ERRC0240
T3 = T2*T	GINV0410		
GINV = T - (C0 + C1*T + C2*T2)/(1 + D1*T + D2*T2 + D3*T3)	GINV0420		
IF (P .EQ. PP) GINV = - GINV	GINV0430		
1010 CONTINUE	GINV0440		
RETURN	GINV0450		
END	GINV0460		
C	LENT0010		
C	LENT0020		
SUBROUTINE LENTRM (STR, K)	LENT0030		
C	LENT0040		
C Trim a string by removing all trailing blanks.	LENT0050		
C	LENT0060		
CHARACTER*(*) STR	LENT0070		
INTEGER*4 K	LENT0080		
C	LENT0090		
IL = LEN (STR)	LENT0100		
K = 1	LENT0110		
DO 1000 I = 1, IL	LENT0120		
IC = ICHAR (STR (I:I))	LENT0130		
IF (IC .EQ. 32) GO TO 1010	LENT0140		
K = K + 1	LENT0150		
1000 CONTINUE	LENT0160		
1010 CONTINUE	LENT0170		
K = K - 1	LENT0180		
RETURN	LENT0190		
END	LENT0200		
C	UPCA0010		
C	UPCA0020		
SUBROUTINE UPCASE (STR)	UPCA0030		
C	UPCA0040		
C Convert a mixed string to all uppercase.	UPCA0050		
C	UPCA0060		
CHARACTER*(*) STR	UPCA0070		
C	UPCA0080		
IL = LEN (STR)	UPCA0090		
DO 1000 I = 1, IL	UPCA0100		
IC = ICHAR (STR (I:I))	UPCA0110		
IF (IC .GE. 97 .AND. IC .LE. 122) STR (I:I) = CHAR (IC - 32)	UPCA0120		
1000 CONTINUE	UPCA0130		
RETURN	UPCA0140		
END	UPCA0150		
C	ERRC0010		
C	ERRC0020		
C	ERRC0030		
SUBROUTINE ERR_CHK (N, NMAX, CODE)	ERRC0040		
C	ERRC0050		
INCLUDE 'MC_TRLHS.INC'	ERRC0060		
C	ERRC0070		
CHARACTER*(*) CODE	ERRC0080		
C	ERRC0090		
IF (N .LE. NMAX) RETURN	ERRC0100		
C	ERRC0110		
WRITE (6, 6000) CODE, NMAX, N	ERRC0120		
WRITE (*, 6000) CODE, NMAX, N	ERRC0130		

Appendix B

Nuclear Test Data Base

The Nuclear Test Database contains information provided Lawrence Livermore National laboratory for most of the nuclear tests conducted at the Nevada Test Site. The column headings are defined as:

Local ID#	Sequential number assigned by IT for ease of tracking
Hole Name	Hole name or location identifier
Hole Type	Unknown
Event Name	Name of the event
Date	Date expended
DOB	Depth to the working point (m)
Max Yield	Yield of the shot, if unclassified, or the maximum of the reported range (kilotons)
Radius	Cavity radius calculated at IT using DOB and Max Yield (m)
Wlv_est	Estimated water table elevation at the shot location (m) - USGS
WP strat	Stratigraphic unit present at the working point
WP hsu	Hydrostratigraphic unit at the working (added by IT)
WP rock type	Lithologic identifier of the working point
UTM - east	Easting coordinate of the collar - UTM (m)
UTM - north	Northing coordinate of the collar - UTM (m)
UTM - east	Easting coordinate of the WP - UTM (m)
UTM - north	Northing coordinate of the WP - UTM (m)
Surface Elevation	Elevation of the land surface (m)

sequence	Hole name	Hole type	Event name	Date	DOB (m)	Max yiel	Radius (WF strat	WP hsu	WP rock	Collar E	cCollar N	coUTM - east	UTM - north	Bottom E	cBottom N	coUTM - east	UTM - north	Surface Elevation
										(meters)	(meters)	(meters)	(meters)	(meters)	(meters)	(meters)	(meters)	(meters)
1	U 1 A.01	H	LEDOUX	27-Sep-90	291	20	38	QTA	AA	AL	206547.83	250698.5	583788.29	4095987.57	-99.9	-99.9	-99.9	1217.98
2	U 1 C	H	YERBA	14-Dec-71	331.99	20	37	QTA	AA	AL	206502	252373.98	583786.41	4097662.46	-99.9	-99.9	-99.9	1228.68
3	U 1 D	H	SUNDOWN	20-Sep-90	256	20	39	QTM	AA	AL	206654.73	254036.07	583883.47	4099324.66	206655.56	254035.98	583884.31	4099324.58
4	U 1 D	H	SUNDOWN	20-Sep-90	270.4	20	39	QTM	AA	AL	206654.73	254036.07	583883.47	4099324.66	206655.56	254035.98	583884.31	4099324.58
5	U 2 A	H	ALPACA	12-Feb-65	224.94	0.33	10	QTM	AA	AL	204811.06	268041.7	581991.13	4113320.23	204811.97	268041.91	581992.04	4113320.45
6	U 2 AA	H	CLUB	30-Jan-64	180.75	20	43	QTM	AA	AL	205353.52	264902.18	582544.48	4110183.43	205353.55	264902.05	582544.51	4110183.29
7	U 2 AB	H	TEE	7-May-65	184.4	7	30	QTM	AA	AL	205706.56	265365.46	582895.81	4110647.83	205706.84	265365.17	582896.09	4110647.54
8	U 2 AD	H	CASHMERE	4-Feb-65	233.17	20	40	QTM	AA	AL	206162.43	264300.78	583355.30	4109585.02	206162.55	264300.33	583355.42	4109584.57
9	U 2 AF	H	KENNEBEC	25-Jun-63	226.16	20	40	QTM	AA	AL	205587.32	264369.66	582780.09	4109651.86	205587.49	264369.36	582780.26	4109651.56
10	U 2 AG	H	MULLETT	17-Oct-63	60.35	20	56	QTM	AA	AL	205685.93	264304.82	582878.90	4109587.39	205685.58	264303.53	582878.56	4109586.10
11	U 2 AH	H	PONGEE	22-Jul-65	134.72	20	46	QTM	AA	AL	205694.55	264414.37	582887.14	4109696.94	205694.29	264413.78	582886.88	4109696.35
12	U 2 AI	H	DRILL	5-Dec-64	188.37	3.4	23	QTM	AA	AL	205435.43	264688.97	582627.12	4109970.56	205435.35	264688.81	582627.04	4109970.40
13	U 2 AI	H	DRILL	5-Dec-64	218.85	20	41	QTM	AA	AL	205435.43	264688.97	582627.12	4109970.56	205435.35	264688.81	582627.04	4109970.40
14	U 2 AK	H	CENTAU	27-Aug-65	173.74	20	43	QTM	AA	AL	205404.97	265024.05	582595.49	4110305.45	205404.72	265023.92	582595.24	4110305.31
15	U 2 AL	H	EMERSON	16-Dec-65	260.6	20	39	QTM	AA	AL	206014.76	265420.05	583203.73	4110703.49	206014.88	265419.65	583203.86	4110703.09
16	U 2 AM	H	COMMODORE	20-May-67	745.24	250	69	TN	VCU	TU	205953.77	264262.2	583146.83	4109545.72	205954.16	264262	583147.21	4109545.52
17	U 2 AN	H	TAPESTRY	12-May-66	248.72	20	39	QTM	AA	AL	205313.9	264688.52	582505.62	4109969.68	205313.87	264688.28	582505.59	4109969.44
18	U 2 AO	H	FLORIST	16-Aug-77	275	20	38	QTM	AA	AL	206016.46	266072.68	583203.14	4111355.95	206016.65	266073.01	583203.33	4111355.29
19	U 2 AP	H	EFFENDI	27-Apr-67	220.98	20	41	QTM	AA	AL	206014.26	265191.68	583204.04	4110475.17	206014.89	265191.24	583204.67	4110474.73
20	U 2 AR	H	ASTAGO	21-Dec-76	330.7	20	37	QTA	AA	AL	205646.46	263539.27	582842.13	4108821.89	205646.55	263538.25	582842.23	4108820.87
21	U 2 AS	H	CLARKSMOBILE	17-May-68	472.44	200	72	TMR	VA	TU	206415.89	263115.32	583612.85	4108400.75	206415.74	263113.95	583612.71	4108399.38
22	U 2 AT	H	KNOX	21-Feb-68	644.8	200	67	TBG	VCU	TU	206874.53	262730.87	584072.73	4108018.01	206874.19	262728.87	584072.39	4108016.01
23	U 2 AU	H	ILDRIM	16-Jul-69	410.26	200	75	TMR	VA	TU	206746.2	263049.12	583943.31	4108335.73	206746.71	263049.19	583943.82	4108335.80
24	U 2 AV	H	CALABASH	29-Oct-69	624.84	110	55	TN	VCU	TU	205953.72	265694.81	583141.74	4110977.96	205956.08	265693.57	583144.11	4110977.63
25	U 2 AW	H	STANYAN	26-Sep-74	572.99	200	69	TMR	VA	TU	205557.6	264505.9	582749.90	4109787.96	205556.48	264506.36	582748.78	4109788.42
26	U 2 AX	H	PORTMANTEAU	30-Aug-74	655.29	200	67	TX	VA	TU	204216.32	266700.62	581401.26	4111977.41	204217.35	266698.58	581402.30	4111975.37
27	U 2 AY 1	H	YANNIGAN	26-Feb-70	391.97	200	76	QTM	AA	AL	206186.22	262704.81	583384.68	4107989.54	206183.34	262700.43	583381.82	4107985.15
28	U 2 AY 2	H	YANNIGAN	26-Feb-70	394.72	200	76	QTM	AA	AL	205740.34	262889.31	582938.27	4108172.43	205740.87	262883.92	582938.82	4108167.04
29	U 2 AY 3	H	YANNIGAN	26-Feb-70	363.63	200	77	QTM	AA	AL	205801	262418.49	583000.57	4107701.94	205804.04	262417.57	583003.61	4107701.03
30	U 2 AZ 1	H	FLASK	26-May-70	528.52	105	57	TMR	VA	TU	206109.36	262372.08	583309.01	4107656.63	206109.53	262371.76	583309.18	4107656.31
31	U 2 AZ 2	H	FLASK	26-May-70	335.28	200	37	QTM	AA	AL	205996.41	262899.87	583194.24	4108183.89	205996.49	262899.57	583194.32	4108183.59
32	U 2 AZ 3	H	FLASK	26-May-70	152.4	20	45	QTM	AA	AL	205719.06	262683.29	582917.72	4107966.39	205719.01	262683.15	582917.67	4107966.25
33	U 2 B	H	ST LAWRENCE	9-Nov-62	166.73	20	44	QTM	AA	AL	205100.41	267964.01	582280.68	4113243.58	205101.09	267964.15	582281.36	4113243.72
34	U 2 BC	H	PARNASSIA	30-Nov-71	330.71	20	37	TN	VCU	TU	205374.26	267592.09	582555.77	4112872.72	205374.59	267591.98	582556.10	4112872.61
35	U 2 BD	H	VULCAN	25-Jun-66	322.78	25	40	QTM	AA	AL	205207.15	267020.35	582390.71	4112300.54	205207.6	267019.7	582391.16	4112299.89
36	U 2 BE	H	NOOR	10-Apr-68	381	200	76	QTM	AA	AL	204612.83	266913.57	581796.92	4112191.70	204613.69	266913.53	581797.78	4112191.66
37	U 2 BF	H	GOURD	24-Apr-69	181.3	20	43	QTM	AA	AL	204536.31	267977.75	581716.68	4113255.33	204536.23	267977.42	581716.60	4113255.00
38	U 2 BG	H	THROW	10-Apr-68	228.6	20	40	QTM	AA	AL	204907.97	267167.05	581491.25	4112444.04	204907.14	267166.71	581490.42	4112443.70
39	U 2 BH	H	SCUTTLE	13-Nov-69	164.59	1.7	19	QTM	AA	AL	204963.25	268053.69	582143.24	4113332.75	204964.09	268053.82	582144.08	4113332.89
40	U 2 BI	H	OAKLAND	4-Apr-67	165.51	20	44	QTM	AA	AL	204308.28	267755.22	581489.49	4113032.06	204308.19	267754.88	581489.40	4113031.72
41	U 2 BJ	H	IMP	9-Aug-68	182.88	20	43	QTM	AA	AL	204755.81	267736.76	581936.97	4113015.18	204755.51	267736.86	581936.67	4113015.28
42	U 2 BL	H	GOURD	24-Apr-69	226.8	20	40	QTM	AA	AL	204429.82	267532.46	581611.78	4112809.79	204430.82	267531.93	581612.78	4112809.26
43	U 2 BM	H	LEXINGTON	24-Aug-67	226.47	20	40	QTM	AA	AL	205023.97	267843.29	582204.69	4113122.62	205024.34	267843.39	582205.05	4113122.72
44	U 2 BN	H	CHATTY	18-Mar-69	195.07	20	42	QTM	AA	AL	204871.33	267791.72	582052.26	4113070.53	204871.36	267791.58	582052.30	4113070.39
45	U 2 BO 1	H	BOWL	26-Jun-69	198.12	20	42	QTM	AA	AL	204627.93	267813.28	581808.85	4113091.23	204627.91	267813.27	581808.83	4113091.22
46	U 2 BO 2	H	BOWL	26-Jun-69	228.6	20	40	QTM	AA	AL	204582.16	267624.14	581763.76	4112901.98	204582.2	267624.14	581763.80	4112901.84
47	U 2 BP 1	H	SPIDER	14-Aug-69	213.36	20	41	QTM	AA	AL	205963.11	267573.06	583144.53	4112855.76	205963.34	267572.79	583144.76	4112855.49
48	U 2 BP 2	H	SPIDER	14-Aug-69	227.69	20	40	QTM	AA	AL	205948.51	267344.89	583130.74	4112627.60	205948.67	267344.55	583130.90	4112627.26
49	U 2 BQ 1	H	KYACK	20-Sep-69	185.93	20	42	QTM	AA	AL	205598.08	267404.71	582780.19	4112686.17	205598.17	267404.73	582780.28	4112686.19
50	U 2 BQ 2	H	KYACK	20-Sep-69	192.02	20	42	QTM	AA	AL	205737.28	267258.38	582919.87	4112540.37	205737.31	267258.23	582919.90	4112540.22
51	U 2 BR	H	HAREBELL	24-Jun-71	518.77	200	71	TX	VA	TU	205685.56	266067.73	582872.34	4111349.84	205685.8	266067.34	582872.59	4111349.45
52	U 2 BS	H	STARWORT	26-Apr-73	563.88	90	53	TX	VA	TU	206441.89	263448.29	583637.68	4108733.73	206443	263448.24	583638.79	4108733.69
53	U 2 BU	H	MINIATA	8-Jul-71	528.83	83	52	TX	VA	TU	207081.55	262018.59	584282.19	4107306.64	207081.71	262018.38	584282.36	4107306.43
54	U 2 BV	H	PORTULACA	28-Jun-73	466.34	200	73	QTTA	AA	AL	203995.03	266249.1	581181.61	4111525.23	203995.32	266248.66	581181.90	4111524.79
55	U 2 BW	H	SUTTER	21-Dec-76	200	20	42	QTM	AA	AL	205984.24	266647.31	583168.91	4111930.32	205984.34	266647.13	583169.01	4111930.14
56	U 2 BX	H	HULSEA	14-Mar-74	195.01	20	42	QTM	AA	AL	205923.38	267005.4	583106.81	4112288.11	205923.35	267005.19	583106.78	4112287.90
57	U 2 BY	H	POLYGONUM	2-Oct-73	213.36	20	41	QTM	AA	AL	205130.73	267406	582312.95	4112685.82	205130.81	267406.09	582313.04	4112685.91
58	U 2 BZ	H	WALLER	2-Oct-73	310.9	20	37	QTM	AA	AL	205603.24	266868.11	582787.23	4112149.73	205603.21	266868.28	582787.20	4112149.90
59	U 2 C	H	KERMET	23-Nov-65	196.29	20	42	QTM	AA	AL	205312.52	267752.14	582493.48	4113032.51	205312.66	267751.45	582493.63	4113031.82
60	U 2 CA	H	STUTZ	6-Apr-66	226.16	20	40	TN	VCU	TU	199113.21	265232.76	576304.63	4110492.00	199113			

62	U	2	CD	H	TRAVELER	4-May-66	197.51	20	42	QTMA	AA	AL	199446.74	264963.24	576639.02	4110223.73	199446.68	264963.23	576638.964110223.72	1457.38
63	U	2	CE	H	NASH	19-Jan-67	365.15	39	45	CBK	LCA	DO	199614.18	265694.33	576803.85	4110955.21	199614.26	265694.43	576803.934110955.32	1452.13
64	U	2	CG	H	HEILMAN	6-Apr-67	152.71	20	45	QTMA	AA	AL	199842.52	266993.18	577034.59	4110255.05	199842.62	266993.31	577034.694110255.18	1432.2
65	U	2	CH	H	POD	29-Oct-69	248.72?#VALUE!		0	TN	VCU	TU	199027.15	265335.05	576218.23	4110593.97	199027.85	265335.4	576218.934110594.32	1491.54
66	U	2	CI	H	POD	29-Oct-69	170.69?#VALUE!		0	QTMA	AA	AL	199195.86	264879.12	576388.50	4110138.75	199196.31	264878.58	576388.954110138.21	1474.56
67	U	2	CJ	H	POD	29-Oct-69	312.42?#VALUE!		0	TU	VCU	TU	199552.98	264771.12	576745.90	4110032.03	199554.17	264770.65	576747.094110031.56	1451.16
68	U	2	CK	H	POD	29-Oct-69	266.77?#VALUE!		0	TN	VCU	TU	200034.35	265313.93	577225.24	4110576.39	200035.06	265313.39	577225.964110575.85	1423.63
69	U	2	CM	S	STODDARD	17-Sep-68	467.87	200	72	TN	VCU	TU	200316.17	263058.08	577514.91	4108322.12	200315.41	263058.08	577514.154108322.11	1396.87
70	U	2	CN	H	CRUET	29-Oct-69	263.65	11	32	TX	VA	TU	200290.56	263240.97	577488.66	4108504.87	200290.38	263240.88	577488.494108504.78	1398.71
71	U	2	CO	H	KRYDDOST	6-May-82	335	20	37	TX	VA	TU	200368.33	262722.83	577568.23	4107987.14	200368.01	262722.65	577567.914107986.96	1390.53
72	U	2	CP	H	CABOC	16-Dec-81	335	20	37	TX	VA	TU	200727.51	262468.02	577928.21	4107733.65	200727.94	262468.42	577928.644107734.05	1374.59
73	U	2	CQ	H	GORBEA	31-Jan-84	388	150	69	TX	VCU	TU	200833.3	262265.63	578034.69	4107531.69	200833.44	262265.49	578034.834107531.55	1371.32
74	U	2	CR	H	WEXFORD	30-Aug-84	314	20	37	TN	VCU	TU	200497.84	265725.18	577687.17	4110989.16	200497.13	265725.44	577686.464110989.42	1403.3
75	U	2	CS	H	MARIBO	26-Jun-85	381	20	35	TN	VCU	TU	200800.69	263530.73	577997.64	4108796.34	200801	263529.97	577997.964108795.59	1379.38
76	U	2	CU	H	KAMICH	24-Feb-89	370	20	36	TN	VCU	TU	200800.85	264018.32	577996.09	4109283.81	-99.9	-99.9	577997.964108795.59	1379.44
77	U	2	CU	H	KAMICH	24-Feb-89	431	20	34	TN	VCU	TU	200800.85	264018.32	577996.09	4109283.81	-99.9	-99.9	577997.964108795.59	1379.44
78	U	2	DB	H	CREW	4-Nov-68	359.66?#VALUE!		0	QTMA	AA	AL	203949.08	264257.19	581142.67	4109533.67	203949.02	264255.78	581142.614109532.26	1314.38
79	U	2	DB	H	CREW	4-Nov-68	603.57?#VALUE!		0	TMA	VA	TU	203949.08	264257.19	581142.67	4109533.67	203949.02	264255.78	581142.614109532.26	1314.38
80	U	2	DC	1	TYG	12-Dec-68	197.82	20	42	QTMA	AA	AL	204839.77	263088.47	582037.23	4108368.38	204839.88	263088.38	582037.344108368.29	1296.63
81	U	2	DC	2	TYG	12-Dec-68	206.96	20	41	QTMA	AA	AL	204690.68	263241.57	581887.64	4108520.91	204691.35	263241.62	581888.314108520.97	1298.66
82	U	2	DC	3	TYG	12-Dec-68	228.3	20	40	QTMA	AA	AL	204781.81	262883.42	581980.01	4108163.18	204781.98	262883.31	581980.184108163.07	1294.94
83	U	2	DC	4	TYG	12-Dec-68	228.3	20	40	QTMA	AA	AL	204484.27	263189.18	581681.47	4108467.81	204484.85	263188.74	581682.054108467.38	1300.01
84	U	2	DC	5	TYG	12-Dec-68	251.16	20	39	QTMA	AA	AL	204575.09	262830.99	581773.53	4108110.04	204575.2	262830.91	581773.644108109.96	1296.69
85	U	2	DC	6	TYG	12-Dec-68	264.87	20	39	QTMA	AA	AL	204381.97	262972.63	581579.96	4108250.96	204382.04	262972.39	581580.034108250.72	1299.61
86	U	2	DD	1	CAN	21-Apr-70	274.32	20	38	QTMA	AA	AL	204353.65	262250.37	581554.18	4107528.79	204353.75	262249.98	581554.284107528.40	1293.14
87	U	2	DD	2	ARNICA	26-Jun-70	309.37	20	37	QTMA	AA	AL	203993.93	262421.17	581193.95	4107698.28	203994.35	262420.97	581194.374107698.08	1296.93
88	U	2	DD	3	ARNICA	26-Jun-70	263.65	20	39	QTMA	AA	AL	204167.7	262777.77	581366.43	4108055.40	204167.68	262777.55	581366.414108055.18	1298.66
89	U	2	DD	4	CAN	21-Apr-70	399.29	200	75	TN	VCU	TU	204525.77	262607.71	581725.00	4107886.64	204525.45	262606.74	581724.694107885.67	1295.1
90	U	2	DE	H	COFFER	21-Mar-69	464.82	100	58	QTMA	AA	AL	203933.91	264561.37	581126.44	4109837.65	203934.26	264560.76	581126.794109837.11	1317.53
91	U	2	DF	H	HUTCH	16-Jul-69	548.64	200	70	QTMA	AA	AL	203860.18	265259.57	581050.27	4110535.48	203861.1	265258.6	581051.194110534.51	1327.04
92	U	2	DG	H	CARPETBAG	17-Dec-70	661.7	220	69	TMA	VA	TU	204262.25	264109.76	581456.28	4109387.38	204262.56	264109.67	581456.594109386.74	1311.18
93	U	2	DE	2	SAPPHO	23-Mar-72	197.82	20	42	QTMA	AA	AL	204466.55	262326.29	581666.78	4107605.08	204466.58	262326.43	581666.814107605.22	1292.66
94	U	2	DE	3	KARA	11-May-72	259.08	20	39	QTMA	AA	AL	204140.11	262262.7	581340.65	4107540.36	204140.05	262262.81	581340.594107540.47	1294.47
95	U	2	DI	H	CHANTILLY	29-Sep-71	330.71	20	37	QTMA	AA	AL	203911.71	263591.53	581107.64	4108868.05	203911.67	263591.28	581107.614108867.80	1306.83
96	U	2	DJ	H	FLAX	21-Dec-72	435.86?#VALUE!		0	QTMA	AA	AL	204231.9	265310.94	581421.72	4110588.14	204231.37	265311.03	581421.194110588.23	1323.08
97	U	2	DJ	H	FLAX	21-Dec-72	688.37?#VALUE!		0	TMR	VA	TU	204231.9	265310.94	581421.72	4110588.14	204231.37	265311.03	581421.194110588.23	1323.08
98	U	2	DK	H	ZINNIA	17-May-72	322.78	20	37	QTMA	AA	AL	203832.71	263155.54	581030.20	4108431.89	203832.85	263155.66	581030.344108432.02	1304.15
99	U	2	DL	H	CHAENACTIS	14-Dec-71	330.71	200	79	QTMA	AA	AL	203682.79	263530.98	580879.09	4108806.71	203682.02	263530.94	580878.234108806.67	1307.9
100	U	2	DM	H	LONGCHAMPS	19-Apr-72	326.44	20	51	QTMA	AA	AL	204192.27	263311.17	581389.12	4108588.75	204192.15	263311.07	581389.004108588.65	1302.93
101	U	2	DN	H	MERIDA	7-Jun-72	204.22	20	41	QTMA	AA	AL	204064.15	262639.05	581263.39	4107916.35	204064.18	262639.02	581263.424107916.32	1297.87
102	U	2	DO	H	GAZOOK	23-Mar-73	326.14	20	37	QTMA	AA	AL	203911.55	262791.4	581110.29	4108068.13	203911.59	262791.34	581110.334108068.07	1300.83
103	U	2	DP	H	DELPHINIUM	26-Sep-72	295.66	15	34	QTMA	AA	AL	204024.31	263247.21	581221.43	4108524.21	204024.37	263247.14	581221.484108524.14	1303.22
104	U	2	DQ	H	SATZ	7-Jul-78	315	20	37	TMR	VA	TU	204795.31	262189.38	581995.94	4107469.36	204795.05	262189.16	581995.684107469.14	1289.92
105	U	2	DR	H	CABRILLO	7-Mar-75	600.5	200	68	QTMA	AA	AL	204147.87	264650.71	581340.03	4109927.79	204147.86	264650.49	581340.024109927.57	1314.67
106	U	2	DS	H	GROVE	22-May-74	313.94	20	37	TMR	VA	TU	205008.9	262524.84	582208.30	4107805.49	205008.96	262524.98	582208.364107805.63	1290.98
107	U	2	DT	H	TANYA	30-Jul-68	381	200	76	QTMA	AA	AL	204338.07	262738.17	581536.89	4108016.41	204338.66	262738.05	581537.484108016.29	1297.58
108	U	2	DU	H	ALVISO	11-Jun-75	183	20	43	QTMA	AA	AL	205100.3	262189.44	582300.85	4107470.49	205100.18	262189.32	582300.734107470.37	1287.87
109	U	2	DV	H	FALLON	23-May-74	466.34	200	73	QTMA	AA	AL	204630.91	263597.64	581826.64	4108876.68	-99.9	-99.9	581827.194108289.45	1301.11
110	U	2	DW	H	CRESTLAKE	18-Jul-74	271.61	20	39	QTMA	AA	AL	204079.31	263012.48	581277.23	4108289.74	204079.26	263012.19	581277.194108289.45	1301.11
111	U	2	DW	H	CRESTLAKE	18-Jul-74	373.7	20	36	QTMA	AA	AL	204079.31	263012.48	581277.23	4108289.74	204079.26	263012.19	581277.194108289.45	1301.11
112	U	2	DY	H	EDAM	24-Apr-75	411.5	200	75	TBG	VCU	TU	203880.92	262615.61	581080.29	4107892.27	203881.16	262615.65	581080.534107892.31	1299.24
113	U	2	DZ	H	BANON	26-Aug-76	536.4	150	64	QTMA	AA	AL	204353.73	263652.74	581549.34	4108930.80	204353.88	263652.91	581549.494108930.97	1302.32
114	U	2	E	H	CUMBERLAND	11-Apr-63	226.47	20	40	QTMA	AA	AL	205312.24	267173.18	582495.24	4112453.70	205312.2	267172.67	582495.204112453.19	1326.4
115	U	2	EA	H	SEAFOAM	13-Dec-73	198.12	20	42	QTMA	AA	AL	205123.22	267695.74	582304.43	4112975.46	205123.25	267695.71	582304.464112975.43	1333.14
116	U	2	EB	H	POTRERO	23-Apr-74	210.31	20	41	QTMA	AA	AL	204825.93	267535.6	582007.78	4112814.32	204825.99	267535.51	582007.544112814.22	1337.01
117	U	2	EF	H	GOUDA	6-Oct-76	200	20	42	QTMA	AA	AL	206099.57	264724.02	583290.97	4110007.93	206099.52	264724.11	583290.924110008.02	1298.48
118	U	2	EG	H	RIVOLI	20-May-76	200	20	42	QTMA	AA	AL	205715.45	265014.28	582905.93	4110296.77	205715.63	265014.6	582906.1141	

125	U	2	EO	H	KLOSTER	15-Feb-79	536.4	150	64	TX	VA	TU	205237.88	266654.79	582422.71	4111935.18	205238.11	266653.89	582422.954111934.28	1323.93
126	U	2	EP	H	NESSEL	29-Aug-79	464	150	66	QTMA	AA	AL	205725.68	263241.03	582922.38	4108524.01	-99.9	-99.9	1286.5	
127	U	2	EQ	H	RIOLA	25-Sep-80	424	1.07	13	QTMA	AA	AL	205908.99	262647.73	583107.73	4107931.50	-99.9	-99.9	1281.2	
128	U	2	ER	H	ISLAY	27-Aug-81	294	20	38	TN	VCU	TU	205703.86	267590.59	582885.29	412872.38	205703.99	267590.57	582885.424112872.36	1323.58
129	U	2	ES	H	AKAVI	3-Dec-81	494	150	65	TX	VA	TU	205328.95	266258.43	582515.15	411539.24	205329.87	266257.78	582516.084111538.60	1320.43
130	U	2	ET	H	CHEEDAM	17-Feb-83	343	20	36	TN	VCU	TU	205984.3	267858.85	583164.71	4113141.55	205984.18	267858.59	583164.604113141.29	1320.27
131	U	2	EU	H	DANABLU	9-Jun-83	320	20	37	QTMA	AA	AL	203689.02	267267.4	580872.10	4112542.19	-99.9	-99.9	1353.47	
132	U	2	EV	H	AGRINI	31-Mar-84	320	20	37	QTMA	AA	AL	204155.81	266029.89	581343.12	4111306.64	204156.18	266029.94	581343.494111306.69	1330.67
133	U	2	EW	H	BRANCO	21-Sep-83	293	20	38	QTMA	AA	AL	206703.43	263262.38	583899.80	4108548.79	206703.91	263262.12	583900.284108548.53	1282.91
134	U	2	EW	H	HEKKIMER	21-Sep-83	427	20	34	TMR	VA	TU	206703.43	263262.38	583899.80	4108548.79	206703.91	263262.12	583900.284108548.53	1282.91
135	U	2	EX	H	ROMANO	16-Dec-83	515	150	64	TMA	VA	TU	205225.33	265371.73	582414.68	4110652.41	205225.33	265371.58	582414.684110652.26	1314.09
136	U	2	EY	H	NIGHTINGALE	22-Jun-88	237.7	150	78	QTMA	AA	AL	205191.77	268224.54	582371.10	4113504.36	205191.71	268224.55	582371.044113504.37	1336.28
137	U	2	EY	H	RHYOLITE	22-Jun-88	207.3	150	81	QTMA	AA	AL	205191.77	268224.54	582371.10	4113504.36	205191.71	268224.55	582371.044113504.37	1336.28
138	U	2	F	H	NARRAGUAGUS	27-Sep-83	150.27	20	45	QTMA	AA	AL	205100.6	266961.12	582284.39	4112240.95	205100.48	266960.61	582284.284112240.44	1329.02
139	U	2	FA	H	FARALLONES	14-Dec-77	668	150	60	QTMA	AA	AL	203987.99	264856.02	581179.47	4110132.48	203988.28	264855.91	581179.764110132.37	1317.29
140	U	2	FB	H	QUARGEL	18-Nov-78	542	150	63	TMR	VA	TU	204185.9	263858.31	581380.83	4109135.72	204186.11	263857.89	581381.044109135.30	1301.9
141	U	2	FC	H	FAJY	28-Jun-79	536	150	64	QTMA	AA	AL	203850.99	265664.47	581039.66	4110940.24	203851.04	265664.32	581039.714110940.09	1330.21
142	U	2	FD	H	TARKO	28-Feb-80	369	20	36	QTMA	AA	AL	203774.43	263820.17	580969.60	4109096.15	-99.9	-99.9	1306.8	
143	U	2	FE	H	CROWDIE	5-May-83	390	20	35	QTMA	AA	AL	203698.2	265938.58	580885.95	4111213.74	203698.35	265938.58	580886.104111213.74	1336.03
144	U	2	FF	H	LABAN	3-Aug-83	326	20	37	QTMA	AA	AL	203744	262981.91	580942.12	4108258.00	203744.2	262981.79	580942.324108257.88	1303.15
145	U	2	G	H	SATOP	15-Aug-63	226.16	20	40	QTMA	AA	AL	204810.91	266883.25	581995.05	4112162.08	204810.99	266883.22	581995.134112162.05	1332.99
146	U	2	GA	S	CORNUCOPIA	24-Jul-86	381	20	35	QTMA	AA	AL	205308.27	265627.45	582496.70	4110908.35	205309.37	265628.29	582497.794110909.19	1314.18
147	U	2	GB	H	PANAMINT	21-May-86	480	20	33	TMR	VA	TU	206273.81	263667.75	583468.87	4108952.54	206274.18	263667.38	583469.244108952.18	1285.82
148	U	2	GE	H	BORATE	23-Oct-87	542.5	150	63	QTMA	AA	AL	204635.21	265528.88	581824.16	4110807.44	204635.28	265528.59	581824.234110807.15	1321.31
149	U	2	GF	H	SCHELLBOURNE	13-May-88	463	150	66	QTMA	AA	AL	205231.4	263594.62	582426.98	4108875.77	-99.9	-99.9	1295.1	
150	U	2	GG	H	INGOT	9-Mar-89	500	150	65	TMR	VA	TU	205679.49	265639.75	582867.78	4110921.95	-99.9	-99.9	1307.42	
151	U	2	GH	H	METROPOLIS	10-Mar-90	469.4	150	66	TMR	VA	TU	206745.94	262280.34	583945.75	4107567.15	-99.9	-99.9	1273.21	
152	U	2	H	H	CARMEL	21-Feb-63	164.29	20	44	QTMA	AA	AL	204521.26	266960.95	581705.21	4112238.74	204522.3	266960.37	581706.254112238.17	1338.2
153	U	2	J	H	ALVA	19-Aug-64	166.12	4.4	26	QTMA	AA	AL	204232.8	267425.99	581415.19	4112702.65	-99.9	-99.9	1347.13	
154	U	2	L	H	AHTANUM	13-Sep-63	226.16	20	40	QTMA	AA	AL	204448.14	267910.54	581628.77	4113187.83	204448.53	267910.02	581629.164113187.31	1346.88
155	U	2	M	1	FENTON	23-Apr-66	167.34	1.4	18	QTMA	AA	AL	204251.77	267595.11	581433.56	4112871.79	-99.9	-99.9	1348.1	
156	U	2	N	H	ACE	11-Jun-64	263.35	20	39	QTMA	AA	AL	204870.81	266274.3	582057.08	4111553.50	204870.82	266274.27	582057.094111553.47	1326.86
157	U	2	P	H	PAR	9-Oct-64	405.69	38	43	QTMA	AA	AL	204780.16	266578.47	581965.38	4111857.27	204780.75	266578.08	581965.974111856.88	1331.43
158	U	2	Q	H	CREPE	5-Dec-64	403.86	200	75	TMR	VA	TU	206898.72	262494.21	584097.74	4107781.50	206898.95	262493.74	584097.974107781.03	1278.58
159	U	2	R	H	PLAID II	3-Feb-66	268.53	200	39	QTMA	AA	AL	205466.13	263804.98	582660.92	4109086.90	205466.41	263804.14	582661.204109086.06	1298.54
160	U	2	T	H	DUMONT	19-May-66	670.87	200	66	TX	VA	TU	206502.28	262128.58	583702.69	4107414.57	206502.33	262128.41	583702.744107414.40	1278.64
161	U	2	U	H	PACKARD	15-Jan-69	246.89	10	31	QTMA	AA	AL	205789.51	266204.06	582975.79	4111486.50	205789.85	266203.75	582976.134111486.19	1315.22
162	U	2	V	H	AGILE	23-Feb-67	733.35	200	65	TX	VA	TU	205740.45	263965.96	582934.95	4109148.83	205740.79	263864.02	582935.304109146.89	1297.35
163	U	2	X	H	LANPHER	18-Oct-67	715.06	200	65	TX	VA	TU	206525.37	262620.07	583724.04	4107906.61	206525.77	262620.03	583724.454107905.98	1281.93
164	U	2	Y	H	HUPMOBILE	18-Jan-68	246.89	7.4	28	QTMA	AA	AL	205789.47	265945.03	582976.66	4111227.54	205788.93	265945.18	582976.124111227.69	1313.69
165	U	3	AA	H	BOOMER	1-Oct-61	100.58	20	49	QTMA	AA	AL	208630.05	255172.42	585854.31	4100467.65	-99.9	-99.9	1227.58	
166	U	3	AB	H	ERMINE	6-Mar-62	73.15	20	54	QTMA	AA	AL	208705.9	255179.16	585930.12	4100474.65	-99.9	-99.9	1227.65	
167	U	3	AC	H	SHERW	16-Sep-61	98.15	20	50	QTMA	AA	AL	208781.77	255185.97	586005.94	4100481.72	-99.9	-99.9	1227.28	
168	U	3	AD	H	PLATYPUS	24-Feb-62	57.91	20	57	QTMA	AA	AL	208861.94	255173.34	586086.14	4100469.38	-99.9	-99.9	1227.1	
169	U	3	AE	H	MINK	29-Oct-61	192.02	20	42	QTMA	AA	AL	208933.48	255199.5	586157.57	4100495.78	-99.9	-99.9	1227.74	
170	U	3	AF	H	COYPU	10-Apr-63	74.68	20	53	QTMA	AA	AL	209002.97	255226.85	586226.94	4100523.37	209002.8	255227.54	586226.774100524.06	1227.58
171	U	3	AG	H	CHINCHILLA 1	19-Feb-62	150.08	1.9	20	QTMA	AA	AL	209072.37	255254.3	586296.23	4100551.06	-99.9	-99.9	1228.29	
172	U	3	AH	H	FISHER	3-Dec-61	363.72	13.4	31	QTMA	AA	AL	209239.84	254900.73	586464.90	4100198.16	-99.9	-99.9	1225.3	
173	U	3	AI	H	HOGNOSE	15-Mar-62	240.34	20	40	QTMA	AA	AL	208942.29	254688.49	586168.17	4099984.93	-99.9	-99.9	1224.93	
174	U	3	AJ	S	RACCOON	1-Jun-62	164.29	20	44	QTMA	AA	AL	208636.45	254869.61	585861.77	4100164.93	-99.9	-99.9	1225.73	
175	U	3	AK	H	RINGTAIL	17-Dec-61	363.02	20	36	QTMA	AA	AL	209451.92	254602.87	586677.97	4099901.12	-99.9	-99.9	1223.56	
176	U	3	AL	H	PAMPAS	1-Mar-62	363.02	9.5	28	QTMA	AA	AL	209154	254390.66	586380.87	4099687.92	-99.9	-99.9	1222.8	
177	U	3	AM	S	AARDVARK	12-May-62	434.04	40	43	TX	VA	TU	208984.95	257044.88	586202.56	4102340.88	-99.9	-99.9	1240.94	
178	U	3	AN	H	WAGTAIL	3-Mar-65	749.6	200	64	TU	VCU	TU	208380.71	256965.37	585598.75	4102259.27	208382.37	256965.62	585600.414102259.52	1237.37
179	U	3	AO	H	AGOUTI	18-Jan-62	260.91	6.4	27	QTMA	AA	AL	208644.09	255050.96	585868.77	4100346.27	-99.9	-99.9	1227.46	
180	U	3	AP	H	STOAT	9-Jan-62	302.36	5.1	24	QTMA	AA	AL	208583.58	254761.19	585809.29	4100056.36	-99.9	-99.9	1225.57	
181	U	3	AQ	H	DORMOUSE	30-Jan-62	363.02	20	36	QTMA	AA	AL	208188.74	255003.39	585413.70	4100297.11	-99.9	-99.9	1227.25	
182	U	3	AR	H	ARMADILLO	9-Feb-62	239.57	7.1	28	QTMA	AA	AL	208238.02	254641.2	585464.24	4099935.19	-99.9	-99.9	1225.12	
183	U	3	AS	H	CHINCHILLA II	31-Mar-62	136.55	20	46	QTMA	AA	AL	208418.27	255012.08	585643.15	4100306.61	-99.9	-99.9	1227.16	
184	U	3	AT	H	JERBOA	1-Mar-63	301.14	20	38	QTMA	AA	AL	209345.79	254752.09	586571.34	4100049.93	209346.25	254752.37	586571.804100050.21	1223.93
185	U	3	AU	S	HAYMAKER	27-Jun-62	408.43	67	52	QTMA	AA	AL	208569.49	254423.39	585796.39	4099718.60	-99.9	-9		

188	U	3	AX	H	PACA	7-May-62	258.32	20	39	QTMA	AA	AL	209476.52	254980.95	586701.23	4100279.19	-99.9	-99.9	1225.7	
189	U	3	AY	H	CHIPMUNK	15-Feb-63	59.44	20	56	QTMA	AA	AL	208875.02	255248.52	586098.95	4100544.59	-99.9	-99.9	1227.77	
190	U	3	AZ	H	DORMOUSE PRIM	5-Apr-62	260.91	10.6	32	QTMA	AA	AL	209609.3	254753.23	586834.78	4100052.00	-99.9	-99.9	1224.08	
191	U	3	BA	H	TENDRAC	7-Dec-62	302.67	20	38	QTMA	AA	AL	209090.09	255553.19	586312.90	4100849.93	209088.12	255554.82	586310.924100851.56	1229.38
192	U	3	BB	H	PEBA	20-Sep-62	241.4	20	40	QTMA	AA	AL	209089.91	255919.02	586311.44	4101215.67	-99.9	-99.9	1232.31	
193	U	3	BC	H	HUTIA	6-Jun-63	134.57	20	46	QTMA	AA	AL	208468.34	254721.08	585694.22	410015.85	208468.58	254720.88	585694.464100015.66	1224.9
194	U	3	BD	H	MERRIMAC	13-Jul-62	413.31	200	75	QTMA	AA	AL	208724.42	255918.98	585946.04	4101214.35	208724.29	255918.87	585945.914101214.24	1231.61
195	U	3	BE	H	DAMON 1	21-Jun-62	260.3	20	39	QTMA	AA	AL	209012.75	254589.27	586238.96	4099885.99	209013	254588.78	586239.214099885.50	1224.08
196	U	3	BF	H	FERRIT	8-Feb-63	325.83	20	37	QTMA	AA	AL	209089.4	255285.04	586309.64	4101581.59	209089.51	256283.76	586309.764101580.31	1234.99
197	U	3	BG	H	ACUSHI	8-Feb-63	260.91	20	39	QTMA	AA	AL	209824.88	254935.13	587049.67	4100234.61	209824.62	254935.28	587049.414100234.75	1225.88
198	U	3	BH	H	HYRAX	14-Sep-62	216.71	20	41	QTMA	AA	AL	209824.94	254691.4	587050.58	4099990.94	209824.45	254691.47	587050.094099991.01	1224.35
199	U	3	BJ	H	BANDICOOT	19-Oct-62	241.4	12.5	34	QTMA	AA	AL	209825.38	254203.68	587052.73	4099503.34	209825.74	254203.53	587053.094099503.19	1221.67
200	U	3	BK	H R	MATACO	14-Jun-63	195.68	20	42	QTMA	AA	AL	210069.23	254935.48	587293.95	4100235.81	210069.52	254935.5	587294.244100235.83	1227.22
201	U	3	BL	H	BOBAC	24-Aug-62	206.05	20	41	QTMA	AA	AL	209580.95	254935.02	586805.80	4100233.64	-99.9	-99.9	1225.6	
202	U	3	BM	H	GUNDI	15-Nov-62	241.4	20	40	QTMA	AA	AL	209581.29	254447.43	586807.85	4099746.17	209580.98	254447.21	586807.544099745.95	1222.25
203	U	3	BN	H	CASSOWARY	16-Dec-64	150.14	20	45	QTMA	AA	AL	209580.79	254203.93	586808.20	4099502.74	209580.62	254203.72	586808.034099502.53	1220.85
204	U	3	BO	H	STURGEON	15-Apr-64	149.77	20	45	QTMA	AA	AL	210069.23	254691.28	587294.81	4099991.67	210069.65	254690.67	587295.234099991.06	1224.84
205	U	3	BP	H	GERBIL	29-Mar-63	279.5	20	38	QTMA	AA	AL	210068.37	254447.47	587294.81	4099747.92	210068.48	254446.52	587294.924099746.97	1222.86
206	U	3	BQ	H	ANCHOVY	14-Nov-63	260.25	20	39	QTMA	AA	AL	210068.73	254203.83	587296.02	4099504.35	210068.9	254203.83	587296.194099504.35	1221.37
207	U	3	BR	H	BELEN	4-Feb-70	420.8	200	74	QTMA	AA	AL	208239.77	255910.88	585461.54	4101204.55	208238.61	255968.2	585460.184101261.85	1232.89
208	U	3	BS	H	FUCE	10-Jun-66	485.55	20	33	QTMA	AA	AL	208239.3	256398.44	585459.36	4101691.99	208239.25	256397.86	585459.314101691.41	1235.45
209	U	3	BT	H	BONEFISH	18-Feb-64	300.76	20	38	QTMA	AA	AL	208727.35	256398.63	585947.29	4101693.88	208728	256397.59	585947.944101692.85	1234.29
210	U	3	BU	H	NUMBAT	12-Dec-62	231.95	20	40	QTMA	AA	AL	210313.1	254935.39	587537.76	4100236.58	210311.88	254934.6	587536.554100235.78	1228.38
211	U	3	BV	H	HARKEE	17-May-63	241.4	20	40	QTMA	AA	AL	210312.48	254691.59	587538.00	4099992.84	210312.31	254692.72	587537.824099993.97	1225.76
212	U	3	BW	H	PEKAN	12-Aug-63	302.27	20	38	QTMA	AA	AL	210312.58	254447.46	587538.95	4099748.77	210312.89	254446.48	587539.274099747.79	1223.77
213	U	3	BX	H	BARBEL	16-Oct-64	258.9	20	39	QTMA	AA	AL	210312.27	254203.67	587539.50	4099505.04	210312.46	254203.71	587539.694099505.08	1221.28
214	U	3	BY	H	FERRIT PRIME	5-Apr-63	241.55	20	40	QTMA	AA	AL	209581.01	253959.75	586809.28	4099258.62	209582.06	253959.17	586810.334099258.04	1219.81
215	U	3	BZ	H	GRUNION	11-Oct-63	261.31	20	39	QTMA	AA	AL	209824.68	253959.97	587052.88	4099259.69	209824.8	253959.91	587053.004099259.63	1219.75
216	U	3	CB	H	CARP	27-Sep-63	329.55	20	37	QTMA	AA	AL	210312.25	253959.94	587540.33	4099261.37	210312.32	253939.82	587540.474099261.25	1219.93
217	U	3	CD	H	DOVEKIE	21-Jan-66	333.05	20	37	QTMA	AA	AL	210312.52	253350.1	587542.74	4098651.69	210312.45	253349.95	587542.674098651.53	1216.52
218	U	3	CE	H	HOOPOE	16-Dec-64	70.35	20	54	QTMA	AA	AL	208743.98	255182.42	585968.18	4100478.04	208743.87	255182.03	585968.074100477.65	1226.94
219	U	3	CG	H	TEJON	17-May-63	74.68	20	53	QTMA	AA	AL	208821.71	255173.35	586045.92	4100469.25	208821.88	255173.21	586046.094100469.11	1227.01
220	U	3	CH	H	SARDINE	4-Dec-63	262.01	20	39	QTMA	AA	AL	209093.04	254203.82	586320.58	4099500.91	209092.94	254203.71	586320.484099500.80	1221.34
221	U	3	CJ	H	SIENNA	18-Jan-66	275.01	20	38	QTMA	AA	AL	210068.38	253959.69	587296.52	4099260.26	210068.64	253959.87	587296.784099260.44	1220.12
222	U	3	CN	H	BILBY	13-Sep-63	714.3	249	70	TOF	VCU	TU	209763.73	256520.14	586982.98	4101819.00	209764.1	256520.11	586983.354101818.97	1241.76
223	U	3	CO	H	PIPEFISH	29-Apr-64	261.92	20	39	QTMA	AA	AL	209337	254203.49	586564.47	4099501.44	-99.9	-99.9	1221	
224	U	3	CP	H	CANVASBACK	22-Aug-64	447.66	20	34	TX	VA	TU	210312.44	257068.77	587529.62	4102369.41	210312.23	257068.57	587529.414102369.21	1253.49
225	U	3	CR	H	BARRACUDA	4-Dec-63	263.34	20	39	QTMA	AA	AL	210617.13	254691.39	587842.57	4099993.70	210617.16	254691.08	587842.604099993.39	1228.65
226	U	3	CT	H	MERLIN	16-Feb-65	296.14	10.1	30	QTMA	AA	AL	209580.56	255544.8	586803.27	4100843.26	209580.68	255544.81	586803.394100843.27	1230.48
227	U	3	CU	H	BITTERLING	12-Jun-64	192.6	20	42	QTMA	AA	AL	210617.01	254142.93	587844.37	4099445.38	210617.86	254143.17	587845.224099445.62	1223.1
228	U	3	CV	H	MINNOW	15-May-64	241.34	20	40	QTMA	AA	AL	210617.1	254447.31	587843.40	4099749.69	210617.2	254447.42	587843.504099749.80	1225.76
229	U	3	CX	H	CYCLAMEN	5-May-66	304.97	12	32	QTMA	AA	AL	208331.09	255423.1	585554.55	4100717.22	208330.89	255422.77	585554.354100716.88	1230.18
230	U	3	CY	H	PIKE	13-Mar-64	114.5	20	48	QTMA	AA	AL	210678.42	255422.45	587901.29	4100724.79	210677.18	255421.49	587900.054100723.83	1237.55
231	U	3	CZ	H	SOLENDON	12-Feb-64	150.18	20	45	QTMA	AA	AL	209092.74	256093.03	586313.65	4101389.64	209092.72	256093.04	586313.644101389.65	1233.35
232	U	3	D	H	PASCAL-B	27-Aug-57	152.47#VALUE!		0	QTMA	AA	AL	208678.65	255252.14	585902.62	4100547.52	-99.9	-99.9	1227.68	
233	U	3	DA	S H	SCAUP	14-May-65	427.03	20	34	TX	VA	TU	210754.05	256352.17	587973.64	4101654.54	210754.6	256351.4	587974.194101653.77	1249.06
234	U	3	DB	H	GUNDI PRIME	9-May-63	271.73	20	39	QTMA	AA	AL	210312.45	255300.99	587535.83	4100602.08	210312.34	255300.86	587535.724100601.95	1232.31
235	U	3	DD	H	KESTREL	5-Apr-65	446.87	20	34	QTMA	AA	AL	209702.64	252679.72	586935.36	4097979.34	209701.76	252679.36	586934.484097978.97	1214.54
236	U	3	DE	H	TUNA	20-Dec-63	414.35	20	35	QTMA	AA	AL	208727.38	255666.66	585949.88	4100962.10	208727.81	255665.97	585950.324100961.41	1229.86
237	U	3	DF	H	CORMORANT	17-Jul-64	271.64	20	39	QTMA	AA	AL	209093.19	251765.24	586329.27	4097062.95	209093.8	251765.29	586329.884097063.01	1211.58
238	U	3	DG	H	SCREAMER	1-Sep-65	301.75	20	38	QTMA	AA	AL	210921.27	252374.83	588155.38	4097678.80	210921.91	252374.79	588155.394097678.76	1211.64
239	U	3	DH	H	BUFF	16-Dec-65	500.41	200	71	TN	VCU	TU	209093.16	257861.42	586307.87	4103157.59	209093.63	257861.55	586308.344103157.72	1250.3
240	U	3	DI	H	GUANAY	4-Sep-64	260.97	20	39	QTMA	AA	AL	209703.18	251765.19	586939.10	4097065.04	209702.94	251765.07	586938.864097064.92	1211.58
241	U	3	DJ	H	TROGON	24-Jul-64	193.03	20	42	QTMA	AA	AL	210617.45	254935.26	587842.04	4100237.51	210617.77	254935.75	587842.354100238.00	1231.09
242	U	3	DK	H	PARROT	16-Dec-64	180.32	1.3	17	QTMA	AA	AL	210617.42	253685.57	587846.38	4098988.14	210617.21	253685.78	587846.184098988.35	1219.45
243	U	3	DL	H	HADDOK	28-Aug-64	363.66	20	36	TX	VA	TU	209703.31	257251.89	586200.01	4102505.35	209703.08	257251.89	586200.034098952.48	1219.2
244	U	3	DM	H	CINNAMON	7-Mar-66	119.85	20	47	QTMA	AA	AL	208970.94	253655.06	586200.43	4098951.87	208970.54	253655.67	586200.034098952.48	1219.2
245	U	3	DN	H	PERSIMMON	23-Feb-67	299.11	20	38	QTMA	AA	AL	210312.57	251765.51	587548.34					

251	U	3	DU	H	FINFOOT	7-Mar-66	195.54	20	42	QTMA	AA	AL	209093.18	253959.98	586321.57	4099257.14	209093.26	253960.05	586321.654099257.21	1220.27
252	U	3	DW	H	TERN	29-Jan-65	210.65	20	41	QTMA	AA	AL	210525.64	254813.19	587750.68	4100115.15	210525.82	254813.38	587750.864100115.34	1228.87
253	U	3	DX	H	MUSCOVY	23-Apr-65	180.29	20	43	TMR	VA	TU	212141.1	251764.98	589376.41	4097073.37	212141.38	251765.12	589376.694097073.51	1216.58
254	U	3	DY	H	PETREL	11-Jun-65	180.75	1.3	17	QTMA	AA	AL	210190.07	254569.1	587416.05	4099869.95	210190.2	254569.4	587416.184099870.25	1224.05
255	U	3	DZ	H	KNIFE B	15-Nov-68	362.93	20	36	QTMA	AA	AL	208757.65	252710.11	585990.51	4098006.41	208757.38	252709.98	585990.234098006.28	1215.24
256	U	3	E	H	PASCAL-C	6-Dec-57	76.2?	VALUE1	0	QTMA	AA	AL	208953.06	255352.08	586176.61	4100648.39	-99.9	-99.9		1228.62
257	U	3	EB	H	TANGERINE	12-Aug-66	87.88	20	51	QTMA	AA	AL	209093.15	254950.17	586318.07	4100247.08	209093.63	254950.23	586318.554100247.14	1225.63
258	U	3	EC	H	OCBRE	29-Apr-66	126.14	20	47	QTMA	AA	AL	209702.8	254660.79	586928.58	4099959.91	209702.7	254660.81	586928.484099959.93	1223.1
259	U	3	ED	H	MOA	1-Sep-65	193.55	20	42	QTMA	AA	AL	210464.76	253959.94	587692.80	4099261.90	210464.93	253960.03	587692.974099261.99	1220.79
260	U	3	EE	H	POMMARD	14-Mar-68	209.1	1.5	17	QTMA	AA	AL	210738.96	255118.17	587962.88	4100420.80	210738.94	255118.05	587962.854100420.68	1235.36
261	U	3	EF	H	MUSHROOM	3-Mar-67	179.55	20	43	QTMA	AA	AL	210799.85	254203.84	588026.95	4099506.92	210799.99	254203.98	588027.09409507.06	1225.06
262	U	3	EH	H	FUTTOCK	18-Jun-75	185.9	20	42	QTMA	AA	AL	209733.28	257099.47	586950.50	4102398.08	209733.46	257099.58	586950.684102398.19	1247.71
263	U	3	EI	H	MORRONES	21-May-70	482.72	200	72	TX	VA	TU	210525.77	257678.37	587740.77	4102979.61	210527.04	257676.78	587742.044102978.02	1264.31
264	U	3	EJ	H	WISE	30-Jan-69	454.06	200	73	TMR	VA	TU	209093.29	255727.62	586315.49	4101024.33	209093.32	255727.45	586315.524101024.16	1230.63
265	U	3	EK	H	TOMATO	7-Apr-66	226.31	20	40	TX	VA	TU	212415.62	251765.5	589650.86	4097074.85	212415.87	251765.37	589651.114097074.72	1222.34
266	U	3	EL	H	PLANER	21-Nov-69	377.84	20	35	QTMA	AA	AL	209702.77	251460.61	586939.76	4096760.54	209703.12	251460.09	586939.764096760.02	1210.21
267	U	3	EM	H	UMBER	29-Jun-67	310.24	10	30	QTMA	AA	AL	209702.81	252984.39	586934.47	4098283.93	209703.77	252981.84	586935.434098281.38	1215.57
268	U	3	EN	H	SEPIA	12-Nov-65	241.19	20	40	QTMA	AA	AL	209733.27	255361.84	586956.58	4100660.88	209733.19	255361.76	586956.564100660.81	1229.19
269	U	3	EO	H	FAWN	7-Apr-67	271.06	20	39	QTMA	AA	AL	209718.04	255849.46	586939.65	4101148.33	209718.11	255849.57	586939.724101148.44	1233.53
270	U	3	EP	H	ABSINTHE	26-May-67	118.65	20	47	QTMA	AA	AL	210007.5	254813.28	587232.67	4100113.42	210007.82	254813.25	587232.994100113.40	1225.79
271	U	3	EQ	H	BRUSH	24-Jan-68	118.3	20	44	QTMA	AA	AL	210464.78	254569.24	587690.69	4099871.05	210464.8	254569.26	587690.714099871.07	1225.63
272	U	3	ER	H	KNIFE C	3-Oct-68	301.44	20	38	TMR	VA	TU	212354.56	252710.2	589586.50	4098019.10	212354.46	252710.29	589586.404098019.19	1228.41
273	U	3	ES	H	CHOCOLATE	21-Apr-67	240.45	20	38	QTMA	AA	AL	208391.98	251948.35	585627.60	4097243.56	208391.79	251948.39	585627.414097243.60	1212.8
274	U	3	ET	H	KEAKI	15-Oct-66	232.55	20	40	QTMA	AA	AL	210190.56	255042.11	587414.88	4100342.84	210190.4	255042.1	587414.724100342.83	1228.59
275	U	3	EU	H	CERISE	18-Nov-66	211.09	20	41	QTMA	AA	AL	210784.97	254569.23	588010.80	4099872.16	210784.7	254569.59	588010.524099872.52	1228.65
276	U	3	EV2S	H	SNUBBER	21-Apr-70	343.5	12.7	31	TU	VCU	TU	212750.86	255927.43	589971.43	411236.91	212750.77	255927.39	589971.34411236.87	1279.74
277	U	3	EW	H	GIBSON	4-Aug-67	240.76	20	40	QTMA	AA	AL	210617.47	252527.35	587850.49	4097830.21	210617.47	252527.7	587850.494097830.56	1213.11
278	U	3	EX	H	GILROY	15-Sep-67	240.63	20	40	QTMA	AA	AL	209855.5	253685.64	587084.66	4098985.54	209855.88	253685.66	587085.044098985.56	1218.62
279	U	3	EY	H	WEMBLEY	5-Jun-68	238.09	20	40	QTMA	AA	AL	210281.82	253685.54	587510.87	4098986.93	210281.92	253685.61	587510.974098987.00	1218.59
280	U	3	EZ	H	SIDECAR	13-Dec-66	240.27	20	40	QTMA	AA	AL	211105.1	253685.22	588333.94	4098989.50	211105.09	253685.2	588333.934098989.48	1222.25
281	U	3	FA	H	SAZERAC	25-Oct-67	301.34	20	38	QTMA	AA	AL	209367.73	253319.95	586598.29	4098618.23	209367.62	253319.76	586598.184098618.04	1217.68
282	U	3	FB	H	KNIFE A	12-Sep-68	331.81	20	37	TMR	VA	TU	210678.02	253350.21	587908.14	4098653.08	210677.63	253350.28	587907.754098653.14	1217.22
283	U	3	FC	H	PICCALILLI	21-Nov-69	393.81	200	76	TU	VCU	TU	211531.42	253289.33	588761.54	4098595.20	211531.42	253289.46	588761.544098595.33	1222.27
284	U	3	FD	H	LAGUNA	23-Jun-71	454.92	200	73	TMR	VA	TU	209702.8	252522.58	586937.02	4097552.53	209703.02	252522.69	586937.244097552.42	1213.36
285	U	3	FE	H	LOVAGE	17-Dec-69	378.05	20	35	QTMA	AA	AL	209702.54	250546.08	586942.73	4095846.24	209701.11	250547.43	586941.304095847.58	1207.96
286	U	3	FF	H	FLORA	1-May-74	149.41	20	45	TX	VA	TU	212994.4	253289.5	590224.15	4098600.50	212995.16	253290.86	590224.914098601.86	1253.97
287	U	3	FH	H	STILT	15-Dec-67	332.43	20	37	TW	VCU	TU	211531.72	253898.77	588759.71	4099204.49	211531.39	253898.87	588759.384099204.59	1230.79
288	U	3	FJ	H	TORCH	21-Feb-68	240.58	20	40	TMR	VA	TU	211531.39	254447.44	588757.45	4099753.02	211531.33	254447.55	588757.404099753.13	1237.19
289	U	3	FK	H	SEVILLA	25-Jun-68	358.65	20	36	TX	VA	TU	212385.3	254447.6	589611.15	4099756.17	212388.2	254448.58	589614.054099757.16	1253.1
290	U	3	FM	H	COGNAC	25-Oct-67	240.39	20	40	QTMA	AA	AL	208178.5	255330.82	585402.32	4100624.42	208179.08	255331.27	585402.904100624.88	1229.57
291	U	3	FN	H	BEEBALM	1-May-70	390.21	20	35	TMR	VA	TU	209184.52	256383.1	586404.40	4101679.96	209184.48	256383.45	586404.354101680.31	1235.63
292	U	3	FQ	H	CANJILON	16-Dec-70	302.19	20	38	TX	VA	TU	209458.74	257861.38	586673.36	4103158.83	209458.26	257861.52	586672.884103158.97	1254.42
293	U	3	FR	H	FIZZ	10-Mar-67	117.73	20	47	QTMA	AA	AL	209052.67	254929.18	586277.67	4100225.95	209052.95	254929.1	586277.964100225.87	1225.76
294	U	3	FS	H	WELDER	3-Oct-68	117.65	20	48	QTMA	AA	AL	209055.59	254973.58	586280.44	4100270.35	209053.43	254973.57	586260.284100270.27	1225.91
295	U	3	FU	H	BEVEL	4-Apr-68	240.75	20	40	QTMA	AA	AL	209839.89	255621.01	587062.27	4100920.36	209839.89	255620.99	587062.274100920.34	1232.31
296	U	3	FV	H	MALLET	31-Jan-68	240.27	20	40	QTMA	AA	AL	210921.99	249936.79	588164.01	4095241.37	210922.14	249936.6	588164.164095241.18	1204.15
297	U	3	FW	H	ADZE	28-May-68	240.03	20	40	TX	VA	TU	212140.98	250790	589379.70	4096098.64	212140.85	250789.89	589379.574096098.53	1215.12
298	U	3	FX	H	AUGER	15-Nov-68	240.62	20	40	TX	VA	TU	211714.48	255118.12	588938.15	4100424.17	211714.24	255118.29	588937.914100424.34	1248.76
299	U	3	FY	H	SPUD	17-Jul-68	240.29	20	40	TMR	VA	TU	211836.42	249936.26	589078.21	4095244.04	211837.04	249936.75	589078.834095244.54	1205.64
300	U	3	FZ	H	HATCHET	3-May-68	240.53	20	40	QTMA	AA	AL	209946.77	252984.35	587178.36	4098284.75	209946.48	252984.11	587178.074098284.50	1215.19
301	U	3	GA	H	FUNNEL	25-Jun-68	118.7	20	47	QTMA	AA	AL	209014.21	254953.81	586239.14	4100250.44	209014.37	254954.06	586239.304100250.69	1226.09
302	U	3	GB	H	FILE	31-Oct-68	228.95	20	40	QTMA	AA	AL	208544.49	251704.23	585780.92	4097000.04	208544.32	251704.16	585780.754096999.97	1212.16
303	U	3	GC	H	BARSAC	20-Mar-69	304.13	20	37	QTMA	AA	AL	209032.3	252252.99	586266.69	4097550.37	209031.91	252253.24	586266.304097550.62	1213.73
304	U	3	GD	H	AJO	30-Jan-70	304.11	20	37	QTMA	AA	AL	208620.44	253228.37	585851.51	4098524.06	-99.9	-99.9		1218.44
305	U	3	GE	H	SAPELLO	12-Apr-74	180.69	20	43	QTMA	AA	AL	207797.75	251460.38	585035.23	4096753.64	207798.21	251460.41	585035.694096753.67	1215.76
306	U	3	GF	H	WINCH	4-Feb-69	240.63	20	40	QTMA	AA	AL	207965.58	258205.15	585014.49	4096144.93	207965.78	250851	585205.354096145.00	1212.19
307	U	3	GG	H	TORTUGAS	1-Mar-84	640.1	150	61	TX	VA	TU	207569.19	257099.12	584786.97	4102390.14	-99.9	-99.9		1243.25

314	U	3	GN	H	PLIERS	27-Aug-69	238.87	20	40	QTMA	AA	AL	208331.36	252192	585566.14	4097486.94	208331.2	252192.29	585565.984097487.23	1214.2
315	U	3	GO	H	TAPPER	12-Jun-69	303.01	20	38	QTMA	AA	AL	209032.36	250789.94	586271.87	4096087.69	209032.26	250790.75	586271.774096088.50	1208.57
316	U	3	GQ	H	BAY LEAF	12-Dec-68	130.16	20	46	QTMA	AA	AL	209014.12	255030.06	586238.78	4100326.67	209014.09	255029.88	586238.754100326.49	1226.21
317	U	3	GR	H	MANZANAS	21-May-70	240.65	20	40	TX	VA	TU	212444.47	251186.36	589681.73	4096495.96	212444.59	251186.26	589681.854096495.86	1221.49
318	U	3	GS	H	APODACA	21-Jul-71	241.4	20	40	TX	VA	TU	212444.61	251429.97	589681.02	4096739.51	212444.37	251429.47	589680.784096739.01	1222
319	U	3	GT	H	BIT A	31-Oct-68	148.31	20	45	QTMA	AA	AL	209079.66	255030.07	586304.30	4100326.91	209079.44	255030.29	586304.084100327.13	1226.02
320	U	3	GT	H	BIT B	31-Oct-68	117.83	20	47	QTMA	AA	AL	209079.66	255030.07	586304.30	4100326.91	209079.44	255030.29	586304.084100327.13	1226.02
321	U	3	GU	H	MESCALERO	5-Jan-72	120.24	20	47	QTMA	AA	AL	209073.62	254877.37	586298.80	4100174.23	209073.77	254877.46	586298.954100174.32	1225.28
322	U	3	GV	H	BONARDA	25-Sep-80	381	150	69	THR	VA	TU	207416.82	256032.47	584638.37	4101323.22	207417.23	256032.37	584638.784101323.13	1236.61
323	U	3	GX	H	ABEYAS	5-Nov-70	393.46	200	76	TX	VA	TU	210663.02	253091.19	587894.06	4098394.07	210662.93	253090.97	587893.974098393.85	1215.37
324	U	3	GZ	H	CUMARIN	25-Feb-70	408.51	200	75	TU	VCU	TU	211744.97	253898.76	588972.91	4099205.22	211744.65	253898.28	588972.594099204.74	1232.84
325	U	3	HA	H	CORAZON	3-Dec-70	241.12	20	40	QTMA	AA	AL	208361.54	250027.99	585603.89	4095323.58	208361.37	250027.8	585603.724095323.39	1207.32
326	U	3	HB	H	JIB	8-May-74	179.8	20	43	THR	VA	TU	211592.36	249936.69	588834.21	4095243.62	211592.93	249936.62	588834.784095243.55	1204.21
327	U	3	HC	H	SPRIT	10-Nov-76	182.9	20	43	QTMA	AA	AL	210159.8	253807.41	587388.46	4099108.34	210159.82	253807.27	587388.474099108.20	1219.1
328	U	3	HD	H	EMBUDO	16-Jun-71	303.06	20	37	QTMA	AA	AL	210495.3	253502.65	587724.94	4098804.84	210495.06	253502.95	587724.704098805.13	1217.88
329	U	3	HE	H	BARRANCA	4-Aug-71	270.74	20	39	QTMA	AA	AL	209977.04	252710.32	587209.58	4098010.89	209977.21	252710.45	587209.754098011.02	1214.19
330	U	3	HF	H	FRIJOLES	22-Sep-71	257.03	20	39	QTMA	AA	AL	210358.09	252527.31	587591.18	4097829.26	210358.72	252527.14	587591.814097829.10	1213.03
331	U	3	HG	H	PEDERNAL	29-Sep-71	378.68	20	35	TPT	VCU	TU	211074.36	251048.99	588312.44	4096353.82	211073.46	251048.61	588311.554096353.44	1207.26
332	U	3	HH	H	JAL	19-Mar-70	301.39	20	38	QTMA	AA	AL	209702.88	249936.61	586945.21	4095236.92	209702.41	249936.47	586944.744095236.78	1205.47
333	U	3	HI	A	CULANTRO	10-Dec-69	134.13	20	46	QTMA	AA	AL	211470.6	251460.44	588707.14	4096766.56	211470.8	251460.22	588707.354096766.34	1208.24
334	U	3	HI	B	CULANTRO	10-Dec-69	149.44	20	45	QTMA	AA	AL	211623.16	251483.19	588859.59	4096789.84	211623.13	251483.02	588859.564096789.67	1209.12
335	U	3	HJ	H	SCUPPER	19-Aug-77	449.6	20	34	QTMA	AA	AL	208910.44	251765.38	586146.57	4097062.45	208910.96	251765.62	586147.084097062.70	1211.95
336	U	3	HK	A	IPECAC	27-May-69	124.18	20	47	QTMA	AA	AL	211531.79	251491.06	588768.21	4096797.38	211531.77	251490.98	588767.994096797.30	1208.39
337	U	3	HK	B	IPECAC	27-May-69	124.22	20	47	QTMA	AA	AL	211714.59	251475.88	588951.02	4096782.85	211714.75	251475.91	588951.184096782.88	1209.56
338	U	3	HK	C	SEAWEEED	1-Oct-69	124.48	20	47	QTMA	AA	AL	211851.48	251353.84	589088.30	4096661.32	211850.92	251353.79	589087.744096661.27	1211.31
339	U	3	HK	D	SEAWEEED	16-Oct-69	118.55	20	47	QTMA	AA	AL	211882.21	251225.7	589119.47	4096533.32	211882.12	251225.65	589119.384096533.27	1211.89
340	U	3	HK	E	SEAWEEED	1-Oct-69	118.55	20	47	QTMA	AA	AL	211851.5	251097.78	589089.22	4096405.32	211851.4	251097.77	589089.124096405.31	1211.61
341	U	3	HK	F	SEAWEEED	1-Oct-69	118.43	20	47	QTMA	AA	AL	211767.67	250995.77	589005.77	4096302.75	211767.51	250996.02	589005.614096303.30	1210.23
342	U	3	HL	H	PENASCO	19-Nov-70	270.92	20	39	QTMA	AA	AL	210312.53	251460.39	587549.37	4096762.45	210311.81	251460.31	587548.654096762.37	1209.86
343	U	3	HM	H	TRUCHAS	28-Oct-70	266.4	20	39	QTMA	AA	AL	210007.61	251460.49	587244.53	4096761.49	210007.27	251460.62	587244.184096761.61	1209.94
344	U	3	HN	H	TRUCHAS	28-Oct-70	118.45	20	47	QTMA	AA	AL	210129.65	251373.41	587366.84	4096674.85	210129.85	251373.35	587367.044096674.80	1209.44
345	U	3	HO	H	TRUCHAS	28-Oct-70	118.38	20	47	QTMA	AA	AL	210190.64	251373.48	587427.81	4096675.14	210190.3	251373.44	587427.474096675.10	1209.44
346	U	3	HP	H	JARA	6-Jun-74	377.95	20	35	QTMA	AA	AL	209641.81	250241.28	586883.08	4095541.30	209641.19	250242.09	586882.464095542.11	1206.95
347	U	3	HQ	H	PRATT	25-Sep-74	313.3	20	37	QTMA	AA	AL	209032.18	251155.77	586270.41	4096453.43	209031.76	251156.18	586269.994096453.84	1210.04
348	U	3	HR	H	CARRIZOZO	3-Dec-70	279.26	20	38	QTMA	AA	AL	208178.72	250485.22	585419.51	4096780.06	208179.14	250484.95	585419.934095779.79	1210.06
349	U	3	HS	H	DEXTER	23-Jun-71	119.94	20	47	QTMA	AA	AL	210266.19	251285.21	587503.65	4096587.15	210266.5	251285.19	587503.964096587.14	1209.35
350	U	3	HT	H	ATARQUE	25-Jul-72	294.44	20	38	QTMA	AA	AL	210388.39	251206.18	587626.10	4096508.57	210388.11	251206.37	587625.824096508.76	1209.2
351	U	3	HU	H	KEEL	16-Dec-74	305.01	20	37	QTMA	AA	AL	210159.87	251072.04	587398.11	4096373.67	210160.24	251072.03	587398.484096373.66	1208.73
352	U	3	HV	H	COLMOR	26-Apr-73	245.8	20	40	QTMA	AA	AL	209950.78	251164.97	587188.74	4096465.84	209951.61	251165.61	587189.574096466.48	1209.44
353	U	3	HX	H	COWLES	3-Feb-72	301.78	20	38	QTMA	AA	AL	210007.64	249936.88	587249.89	4095238.26	210007.54	249936.97	587249.794095238.35	1205.4
354	U	3	HY	H	ELIDA	19-Dec-73	381	20	35	QTMA	AA	AL	208940.9	249936.6	586183.42	4095234.25	208940.79	249937.41	586183.314095235.06	1205.62
355	U	3	HZ	H	FRIJOLES	22-Sep-71	229.17	20	40	QTMA	AA	AL	210037.94	252253.16	587272.07	4097554.06	210037.63	252253.28	587271.764097554.18	1213.1
356	U	3	J	H	PASCAL-A	26-Jul-57	152.47#VALUE!	0	QTMA	AA	AL	208724.33	255553.42	585947.23	4100848.88	-99.9	-99.9		1229.2	
357	U	3	JA	H	ESTACA	17-Oct-74	320.31	20	37	THA	VA	TU	210410.19	250530.21	587650.26	4095832.85	210410.27	250530.06	587650.344095832.70	1206.73
358	U	3	JB	H	BOBSTAY	26-Oct-77	381	20	35	THR	VA	TU	210244.69	250662.15	587484.34	4095964.18	210245.05	250662.33	587484.704095964.36	1207.41
359	U	3	JC	H	CEBOLLA	9-Aug-72	286.7	20	38	QTMA	AA	AL	210038.1	250615.02	587277.97	4095916.34	210037.71	250615.54	587277.584095916.85	1207.52
360	U	3	JD	H	MESITA	9-May-73	149.35	20	45	QTMA	AA	AL	210301.66	250514.69	587541.81	4095816.95	210301.93	250514.67	587542.084095816.93	1206.89
361	U	3	JE	H	HOSPAN	14-Dec-71	302.03	20	38	QTMA	AA	AL	209123.79	252679.8	586356.66	4097977.39	209123.45	252679.73	586356.324097977.32	1215.15
362	U	3	JF	H	SHALLOWS	26-Feb-76	243.8	20	40	QTMA	AA	AL	210312.34	252984.46	587543.84	4098286.14	210312.09	252984.32	587543.594098286.00	1215.15
363	U	3	JG	H	ANGUS	25-Apr-73	452.93	20	34	QTMA	AA	AL	209199.8	250348.01	586440.81	4095644.46	-99.9	-99.9		1207.47
364	U	3	JH	H	BACKGAMMON	29-Nov-79	228.6	20	40	QTMA	AA	AL	209596.05	249143.89	586841.18	4094444.03	209595.83	249144.29	586840.964094444.43	1202.57
365	U	3	JI	H	PAJARA	12-Dec-73	277.98	20	38	QTMA	AA	AL	209596.01	248869.58	586842.10	4094169.79	209595.94	248868.82	586842.034094169.03	1202.44
366	U	3	JJ	H	CAPITAN	28-Jun-72	134.48	20	46	QTMA	AA	AL	209771.21	248995.21	587016.81	4094296.00	209770.84	248995.05	587016.454094295.84	1202.69
367	U	3	JK	H	VELARDE	25-Apr-73	276.76	20	38	QTMA	AA	AL	209868.87	249112.04	587114.04	4094413.14	209868.46	249111.98	587113.634094413.08	1203.19
368	U	3	JL	H	FUYE	14-Aug-74	430.01	20	34	QTMA	AA	AL	208483.64	252405.09	585717.63	4097700.51	208484.01	252401.28	585717.984097700.51	1215.21
369	U	3	JM	H	JICARILLA	19-Apr-72	148.11	20	45	QTMA	AA	AL	210254.23	250552.77	587494.26	4095854.86	210254.1	250552.43	587494.134095854.52	1206.88
370	U	3	JN	H	ALGODONES	18-Aug-71	527.61	200	70	THA	VA	TU	208461.63	256153.25	585682.49	4101447.64	208462.26	2561		

377	U	3	JV	H	RIB	14-Dec-77	213.4	20	41	QTMA	AA	AL	210129.83	251765.69	587365.64	4097067.04	210130.72	251765.63	587366.534097066.98	1210.91
378	U	3	JW	H	FRIJOLAS	22-Sep-71	149.86	20	45	QTMA	AA	AL	210278.76	252216.69	587512.96	4097518.44	210278.42	252216.63	587512.624097518.38	1212.34
379	U	3	JX	H	SOLANO	9-Aug-72	133.81	20	46	QTMA	AA	AL	210248.5	250173.98	587489.86	4095476.14	-99.9	-99.9		1205.99
380	U	3	JY	H	BERNAL	28-Nov-73	283.46	20	38	QTMA	AA	AL	209550.54	251033.79	586789.06	4096333.29	209550.54	251033.71	586789.064096333.21	1209.75
381	U	3	K	H	COLFAX	5-Oct-58	106.7	0.0055	3	QTMA	AA	AL	208659.76	255228.1	585883.82	4100523.42	-99.9	-99.9		1227.93
382	U	3	KB	H	MARSH	6-Sep-75	426.7	20	34	QTMA	AA	AL	209199.88	252436.02	586433.58	4097733.94	209198.84	252438.28	586432.534097736.19	1214.08
383	U	3	KC	H	BILGE	19-Feb-75	319	20	37	QTMA	AA	AL	209549.39	250058.52	586791.33	4095358.27	-99.9	-99.9		1205.93
384	U	3	KD	H	DECK	18-Nov-75	326.1	20	37	QTMA	AA	AL	209885.63	252070.07	587120.44	4097370.48	209885.58	252072.56	587120.384097372.97	1212.53
385	U	3	KF	H	FOREFOOT	2-Jun-77	193.5	20	42	QTMA	AA	AL	209474.21	255910.65	586695.67	4101208.65	209473.89	255910.66	586695.354101208.66	1233.59
386	U	3	KG	H	PUDDLE	26-Nov-74	183	20	43	QTMA	AA	AL	210739.15	249936.43	587981.22	4095240.37	-99.9	-99.9		1196.37
387	U	3	KI	H	COVE	16-Feb-77	335.3	20	37	QTMA	AA	AL	208940.65	250546.53	586181.03	4095844.02	208938.7	250548.05	586179.084095845.53	1207.95
388	U	3	KJ	H	JACKPOTS	1-Jun-78	304.8	20	37	QTMA	AA	AL	208880.06	252100.77	586115.02	4097397.65	208879.24	252103.73	586114.194097400.61	1212.92
389	U	3	KK	H	CERNADA	24-Sep-81	213	20	41	QTMA	AA	AL	209612.66	250759.76	586852.13	4096059.55	209612.22	250759.44	586851.694096059.23	1208.3
390	U	3	KM	H	OARLOCK	16-Feb-77	320	20	37	QTMA	AA	AL	209184.65	251308.1	586422.31	4096606.25	-99.9	-99.9		1210.08
391	U	3	KN	H	CONCENTRATION	1-Dec-78	243.8	20	40	QTMA	AA	AL	209567.23	253106.57	586798.49	4098405.60	209566.45	253106.08	586797.714098405.11	1216.31
392	U	3	KP	H	SEAMOUNT	17-Nov-77	371.9	20	36	QTMA	AA	AL	209489.92	252100.11	586724.73	4097399.13	209486.71	252098.83	586721.524097397.84	1212.83
393	U	3	KQ	H	MEMORY	14-Mar-79	365.8	20	36	QTMA	AA	AL	208178.83	252893.01	585411.19	4098187.24	208178.57	252891.35	585410.944098185.58	1217.59
394	U	3	KR	H	CLAIRETTE	5-Feb-81	353.6	20	36	QTMA	AA	AL	208864.83	251018.39	586103.58	4096315.50	208864.41	251018.46	586103.164096315.56	1209.51
395	U	3	KS	H	OFFSHORE	8-Aug-79	396.2	150	69	TPT	VCU	TU	211013.43	251460.53	588250.09	4096765.05	211013.56	251461.3	588250.224096765.82	1208.69
396	U	3	KT	H	EBBTIDE	15-Sep-77	381	20	35	QTMA	AA	AL	209874.02	253441.71	585104.54	4098734.73	207867.68	253441.71	585098.204098734.71	1220.67
397	U	3	KU	H	VERDELLLO	31-Jul-80	365.8	20	36	QTMA	AA	AL	209702.77	251262.45	586940.45	4096562.43	-99.9	-99.9		1209.91
398	U	3	KV	H	VICTORIA	19-Jun-92	243.9	20	40	QTA	AA	AL	210830.58	250424.17	588070.92	4095728.31	210830.77	250424.86	588071.104095729.00	1205.88
399	U	3	KW	H	FREZEOUT	11-May-79	335.3	20	37	QTMA	AA	AL	210160.07	249616.5	587403.40	4094918.50	210159.93	249616.27	587403.264094918.27	1204.36
400	U	3	KX	H	CANFIELD	2-May-80	350.5	20	36	TMA	VA	TU	210007.55	256032.62	587228.45	4101332.46	210007.82	256032.28	587228.714101332.12	1237.55
401	U	3	KY	H	HURON KING	24-Jun-80	320	20	37	QTMA	AA	AL	208681.71	252393.05	585915.69	4097689.17	-99.9	-99.9		1214.64
402	U	3	KZ	H	ALEMAN	11-Sep-86	502.9	20	33	TMR	VA	TU	207264.79	257464.64	584481.36	4102754.50	-99.9	-99.9		1245.11
403	U	3	LA	H	BOUSCHET	7-May-82	563.9	150	63	TX	VA	TU	207640.75	257465.03	584857.23	4102756.20	207641.09	257463.4	584857.574102754.58	1244.5
404	U	3	LB	H	NAVATA	29-Sep-83	183	20	43	QTMA	AA	AL	209885.51	255743.02	587107.45	4101042.50	-99.9	-99.9		1233.8
405	U	3	LC	H	SABADO	11-Aug-83	320	20	37	TMR	VA	TU	211501.56	249570.84	588744.71	4094877.54	211501.9	249570.18	588745.054094876.88	1202.62
406	U	3	LD	H	VILLITA	10-Nov-84	373.4	20	36	QTMA	AA	AL	210190.44	249828.72	587433.02	4095131.77	210189.92	249829.87	587432.504095131.91	1204.63
407	U	3	LE	H	CERRO	2-Sep-82	228.6	20	40	QTMA	AA	AL	210327.73	252009.22	587562.64	4097311.20	210327.88	252009.13	587562.794097311.11	1211.43
408	U	3	LG	H	FLORA	22-May-80	335.3	20	37	QTMA	AA	AL	208940.86	250149.88	586182.63	4095447.47	-99.9	-99.9		1206.35
409	U	3	LH	H	TENAGA	17-Apr-82	357	20	36	TX	VA	TU	210846.07	251689.01	588081.97	4096992.89	210845.31	251688.89	588081.214096992.76	1204.57
410	U	3	LI	H	MOGOLLON	20-Apr-86	259.1	20	39	TMA	VA	TU	207630.17	251094.8	584868.97	4096387.56	-99.9	-99.9		1214.63
411	U	3	LJ	H	TREBBIANO	4-Sep-81	304.8	20	37	TMA	VA	TU	207416.63	256245.82	584637.43	4101536.52	207416.72	256245.97	584637.534101536.67	1237.89
412	U	3	LK	H	MONAHANS A	9-Nov-88	289.6	20	38	QTMA	AA	AL	209867.38	248838.95	587113.51	4094140.12	209867.62	248838.92	587113.754094140.09	1202.29
413	U	3	LL	H	TORNERO	11-Feb-87	298.4	20	38	TMR	VA	TU	207752.1	250988.06	584991.24	4096281.28	-99.9	-99.9		1213.41
414	U	3	LM	H	SEYVAL	12-Nov-82	365.8	20	36	QTMA	AA	AL	208864.49	252435.99	586098.28	4097732.73	-99.9	-99.9		1214.36
415	U	3	LO	H	COALORA	11-Feb-83	274	20	38	TMA	VA	TU	207660.72	256032.45	584882.21	4101324.06	207660.65	256032.21	584882.144101323.82	1237.02
416	U	3	LP	H	WHITEFACE A	20-Dec-89	197.1	20	42	QTMA	AA	AL	208940.94	252679.83	586173.85	4097976.78	-99.9	-99.9		1214.7
417	U	3	LP	H	WHITEFACE B	20-Dec-89	182.9	20	43	QTMA	AA	AL	208940.94	252679.83	586173.85	4097976.78	-99.9	-99.9		1214.7
418	U	3	LR	H	VAUGHN	15-Mar-85	426.7	150	67	TMR	VA	TU	207660.66	256245.84	584881.40	4101537.39	-99.9	-99.9		1238.07
419	U	3	LS	H	MUGGINS	9-Dec-83	244	20	40	TMA	VA	TU	207660.69	251216.72	584899.06	4096509.56	207660.69	251216.54	584899.064096509.38	1215.22
420	U	3	LT	H	MINERO	20-Dec-84	243.8	20	40	TMA	VA	TU	207749.58	251127.7	584988.23	4096420.87	207749.16	251128.01	584987.814096421.18	1214.62
421	U	3	LU	H	WACO	1-Dec-87	182.9	20	43	QTMA	AA	AL	211342.65	249427.79	588586.34	4094733.97	-99.9	-99.9		1202.78
422	U	3	LV	H	DUORO	20-Jun-84	381	150	69	TMA	VA	TU	207904.5	249845	585147.61	4095139.04	-99.9	-99.9		1207.3
423	U	3	LW	H	CORREO	2-Aug-84	335.3	20	37	TMR	VA	TU	211043.94	251689.12	588279.79	4096993.69	-99.9	-99.9		1208.98
424	U	3	LZ	H	CHAMITA	17-Aug-85	335.3	20	37	TMA	VA	TU	207904.41	250058.36	585146.77	4095352.35	-99.9	-99.9		1208.21
425	U	3	M	H	LUNA	21-Sep-58	147.5	0.0015	2	QTMA	AA	AL	208701.25	255272.64	585925.14	4100568.09	-99.9	-99.9		1227.69
426	U	3	MC	H	ABO	30-Oct-85	198.1	20	42	QTMA	AA	AL	208483.59	255422.88	585707.01	4100717.53	-99.9	-99.9		1228.95
427	U	3	ME	H	KINIBITO	5-Dec-85	579.1	150	62	TX	VA	TU	207660.56	255712.5	584883.17	4101004.19	-99.9	-99.9		1234.97
428	U	3	MF	H	TAROKA	13-Aug-87	640.1	150	61	TN	VCU	TU	207660.37	256566.53	584879.99	4101858.00	-99.9	-99.9		1239.5
429	U	3	MG	H	PANCHUELA	30-Jun-87	320	20	37	QTMA	AA	AL	207904.54	249464.93	585148.34	4094941.02	207904.54	249464.9	585148.344094940.99	1206.09
430	U	3	MH	H	LAREDO	21-May-88	350.5	150	71	TN	VCU	TU	212841.78	253441.62	590071.04	4098752.04	212841.77	253441.71	590071.034098752.13	1247.12
431	U	3	MK	H	BOWIE	6-Apr-90	213.4	20	41	TC	VCU	TU	212385.14	257343.13	598600.84	4102650.97	-99.9	-99.9		1297.63
432	U	3	ML	H	DIVIDER	23-Sep-92	339.9	20	36	TC	VCU	TU	212796.86	252131.3	590030.72	4097441.90	-99.9	-99.9		1234.76
433	U	3	MN	H	ABILENE	7-Apr-88	243.8	20	40	TMR	VA	TU	207782.58	251262.38	585020.75	4096555.63	-99.9	-99.9		1214.63
434	U	3	MT	H	LUBBOCK	18-Oct-91	457.2	150	66	TF	VCU	TU	207660.35	256840.54	584879.01	4102131.94	-99.9	-99.9		1239.71
435	U	3	N	H	BERNALILLO	17-Sep-58	140	0.015	4	QTMA	AA	AL	208750.93	255308.31	585974.68	4100603.93	-99.9	-99.9		1227.78
436	U	3	P	H	SAN JUAN	20-Oct-58	71.3	0	0	QTMA	AA	AL	208806.42	255333.15	586030.07	4100628.96	-99.9	-99.9		1227.97
437	U	3	Q	H	OTERO	12-Sep-58	146.3	0.038	6	QTMA	AA	AL	208865.84	255346.41	586089.43	4100642.42	-99.9	-99.9		1228.79
438	U	3																		

440	U	4	AA	H	TRUMBULL	26-Sep-74	262.71	20	39	QTTA	AA	AL	203240.88	260116.77	580449.18	4105391.83	203240.71	260116.79	580449.014105391.85	1295.59
441	U	4	AB	H	TEMESCAL	2-Nov-74	262.71	20	39	QTTA	AA	AL	203515.39	260116.83	580723.62	4105392.85	203515.34	260116.91	580723.574105392.94	1290.4
442	U	4	AC	H	BELLOW	16-May-84	207.3	20	41	QTMA	AA	AL	203378.14	260040.64	580586.68	4105316.21	203378.28	260040.46	580586.814105316.03	1292.75
443	U	4	AF	H	CARNELIAN	28-Jul-77	208	20	41	QTMA	AA	AL	203576.4	260604.49	580782.90	4105880.60	203576.01	260604.42	580782.524105880.53	1291.68
444	U	4	AH	H	KARAB	16-Mar-78	331	20	37	TN	VCU	TU	204414.68	259232.63	581625.78	4104512.04	204414.48	259232.55	581625.584104511.96	1274.52
445	U	4	AI	H	BURZET	3-Aug-79	450	150	66	QTTA	AA	AL	205458.48	259103.41	582669.77	4104386.51	205459.36	259103.07	582670.654104386.18	1262.42
446	U	4	AJ	H	MONTREY	29-Jul-82	400	150	68	TX	VA	TU	204993.65	261137.52	582197.92	4106418.47	-99.9	-99.9		1280.16
447	U	4	AK	H	TILCI	11-Nov-81	445	150	67	QTTA	AA	AL	205587.96	258257.6	582802.18	4103541.37	205588.52	258257	582802.744103540.78	1258.98
448	U	4	AL	H	MANTECA	10-Dec-82	413	150	68	QTMA	AA	AL	205283.14	258684.34	582495.94	4103966.93	205283.05	258684.33	582495.854103966.93	1262.97
449	U	4	AM	H	VILLE	12-Jun-85	293.2	20	38	TX	VA	TU	204206.43	259579.58	581416.37	4104858.39	204206.27	259579.54	581416.214104858.39	1276.84
450	U	4	AN	H	COSO	8-Mar-91	475	20	34	TBG	VCU	TU	205076	261366.57	582279.44	4106647.75	205076.34	261366.09	582279.794106647.27	1281.23
451	U	4	AN	H	COSO	8-Mar-91	442	20	34	TC	VCU	TU	205076	261366.57	582279.44	4106647.75	205076.34	261366.09	582279.794106647.27	1281.23
452	U	4	AN	H	COSO	8-Mar-91	333	20	37	TMR	VA	TU	205076	261366.57	582279.44	4106647.75	205076.34	261366.09	582279.794106647.27	1281.23
453	U	4	AR	H	BRETON	13-Sep-84	483	150	65	TMA	VA	TU	205341.99	259406.31	582552.24	4104688.93	205341.94	259406.71	582552.194104688.93	1264.59
454	U	4	AS	H	ROQUEFORT	16-Oct-85	415	150	68	TN	VCU	TU	200863.63	261960.82	578066.08	4107227.06	-99.9	-99.9		1367.82
455	U	4	AT	H	PALISADE 1	15-May-89	335.3	20	37	TC	VCU	TU	200909.23	261701.72	578112.58	4106968.19	-99.9	-99.9		1365.13
456	U	4	AT	H	PALISADE 2	15-May-89	390	20	35	TC	VCU	TU	200909.23	261701.72	578112.58	4106968.19	-99.9	-99.9		1365.13
457	U	4	AT	H	PALISADE 3	15-May-89	404	20	35	TC	VCU	TU	200909.23	261701.72	578112.58	4106968.19	-99.9	-99.9		1365.13
458	U	4	AU	H	BULLFROG	30-Aug-88	489.2	150	65	TMR	VA	TU	205578.84	259324.2	582789.32	4104607.67	-99.9	-99.9		1263.4
459	U	4	AV	H	BRISTOL	26-Nov-91	457.2	20	34	TC	VCU	TU	205474.32	260494.72	582680.72	4105777.50	-99.9	-99.9		1273.22
460	U	4	B	H	MACKEBEL	18-Feb-64	333.71	20	37	TX	VA	TU	207154.19	260409.93	584360.46	4105698.65	207155	260410	584361.274105698.72	1263.37
461	U	4	C	H	ZAZA	27-Sep-67	667	200	66	TN	VCU	TU	206929.18	260756.86	584134.29	4106044.70	206930.69	260756.65	584135.804106044.49	1267.41
462	U	4	D	H	LATIR	27-Feb-74	640.99	200	67	TN	VCU	TU	206959.58	261366.41	584162.54	4106654.20	206960.66	261364.8	584163.634106652.59	1270.03
463	U	4	E	H	TOGALLANT	28-Feb-75	713.2	200	65	TN	VCU	TU	206654.84	261580.04	583857.13	4104866.71	206655.57	261579.72	583857.864104866.39	1272.07
464	U	4	F	H E/U4N	10-May-78	640	0	0	TX	VA	TU	206990.12	259537.59	584199.49	4106826.05	206989.86	259535.11	584199.244104823.47	1258.49	
465	U	4	G	H	ICEBERG	23-Mar-78	640	150	61	TN	VCU	TU	207111.71	261092.1	584315.60	4106380.49	207111.79	261093.04	584315.674106381.43	1265.67
466	U	4	H	H	SCANTLING	19-Aug-77	701	150	60	TN	VCU	TU	206807.03	262006.47	584007.78	4107293.56	206807.81	262006.04	584008.574107293.13	1272.48
467	U	4	I	H	GLENCOE	22-Mar-86	609.6	29	36	TX	VCU	TU	205801.28	259000.55	583012.84	4104284.88	205801	259000.86	583012.564104285.19	1260.53
468	U	4	J	H	JORNADA	28-Jan-82	640	139	59	TN	VCU	TU	207111.79	259934.12	584319.74	4105222.81	-99.9	-99.9		1260.11
469	U	4	L	H	QUINELLA	8-Feb-79	579.1	150	62	TX	VA	TU	206782.79	261168.34	583986.49	4106455.56	206782.21	261169.09	583985.914106456.31	1267.64
470	U	4	N	H	HEARTS	6-Sep-79	640	140	60	TX	VA	TU	206976.11	259574.95	584185.35	4104863.25	206976.84	259572.35	584186.094104860.66	1258.89
471	U	4	O	H	TECHADO	22-Sep-83	533.4	150	64	TN	VCU	TU	207264.51	261518.55	584466.86	4106807.37	207264.86	261518.39	584467.214106807.21	1267.59
472	U	4	P	H	ROUSANNE	12-Nov-81	518.2	150	64	TBG	VCU	TU	207300.13	261800.89	584501.48	4107089.76	207300.84	261801.18	584502.194107090.06	1269.8
473	U	4	Q	H	CAPROCK	31-May-84	600	150	62	TN	VCU	TU	207386.32	261244.54	584589.60	4105533.86	-99.9	-99.9		1263.71
474	U	4	R	H	VERMEJO	2-Oct-84	350.5	20	36	TMA	VA	TU	206976.28	259254.71	584186.65	4104543.10	-99.9	-99.9		1255.6
475	U	4	S	H	TULIA	26-May-89	396.24	20	35	TMA	VA	TU	206769.11	259324.36	583979.29	4104612.00	206769.54	259324.12	583979.724104611.77	1257.3
476	U	4	T	H	GASCON	14-Nov-86	593.14	150	62	TN	VCU	TU	207386.31	260944.24	584590.65	4106233.63	-99.9	-99.9		1263.09
477	U	4	U	H	DALHART	13-Oct-88	640.1	150	61	TN	VCU	TU	207288.55	259675.24	584497.36	4104954.62	-99.9	-99.9		1255.62
478	U	5	A	H	WISHBONE	18-Feb-65	174.96	20	43	QTA	AA	AL	216408.15	229667.19	593719.61	4074996.01	216408.32	229667.4	593719.784074996.23	940.52
479	U	5	B	H	DILUTED WATER	16-Jun-65	192.63	20	42	QTA	AA	AL	215798.57	229667.35	593110.18	4074994.05	215798.84	229667.19	593110.454074993.89	943.42
480	U	5	E	H	CAMBRIC	14-May-65	294.74	0.75	13	QTA	AA	AL	214832.97	230252.04	592142.78	4075575.23	214833.32	230252.02	592143.134075575.21	956.1
481	U	5	I	H	DERRINGER	12-Sep-66	255.12	7.8	29	QTA	AA	AL	216235.08	236207.06	593523.77	4081533.66	-99.9	-99.9		1039.98
482	U	5	K	H	MILKSHAKE	25-Mar-68	264.57	20	39	QTA	AA	AL	217976.85	235639.47	595267.09	4080972.29	217977.1	235639.79	595267.334080972.61	1020.63
483	U	6	A	H	RUSSET	5-Mar-68	119.79	20	47	QTMA	AA	AL	206792.18	246458.16	584047.42	4091749.17	206806.32	246462.63	584061.544091753.69	1197.41
484	U	6	D	H	PRESIDIO	22-Apr-87	320	20	37	TX	VA	TU	211342.58	247955.24	588591.43	4093261.79	-99.9	-99.9		1198.51
485	U	6	E	H	AUSTIN	21-Jun-90	350.5	20	36	TMR	VA	TU	211342.68	249031.21	588587.76	4094337.49	211342.5	249030.97	588587.584094337.25	1201.52
486	U	6	G	H	HARLINGEN A	23-Aug-88	289.6	20	38	QTMA	AA	AL	210141.53	248839.17	587387.59	4094141.30	-99.9	-99.9		1202.32
487	U	6	H	H	HARLINGEN B	23-Aug-88	289.6	20	38	QTMA	AA	AL	210141.64	248564.82	587388.66	4093867.02	-99.9	-99.9		1201.87
488	U	6	I	H	MONAHANS B	9-Nov-88	289.6	20	38	QTMA	AA	AL	209867.42	248564.76	587114.51	4093866.00	-99.9	-99.9		1201.46
489	U	7	A	H	FOREST	31-Oct-64	386.79	20	35	TX	VA	TU	208788.07	261706.16	585989.37	4107000.28	208788.29	261707.45	585989.594107001.57	1296.77
490	U	7	AA	H	TAJIQUE	28-Jun-72	332.31	20	37	TN	VCU	TU	212384.93	257556.49	589599.88	4102864.28	212385.29	257557.19	589600.244102864.98	1300.72
491	U	7	AB	H	REDMUD	8-Dec-76	426.7	20	34	TN	VCU	TU	211531.47	258623.57	588742.90	4103928.10	211531.7	258624.12	588743.124103928.65	1295.66
492	U	7	AC	H	ESCAPOS	10-Jul-74	639.99	200	67	TN	VCU	TU	208849.29	258135.48	586063.10	4103430.72	208849.24	258135.03	586063.064103430.27	1250.4
493	U	7	AD	H	MIRRA	8-Mar-73	568.76	200	69	TN	VCU	TU	209280.57	261307.87	586483.15	4106603.82	209281.47	261306.78	586484.054106602.73	1206.21
494	U	7	AE	H	STRAKE	4-Aug-77	518.2	150	64	TN	VCU	TU	211059.29	259431.08	588268.00	4104733.75	211059.53	259431.5	588268.244104734.17	1200.18
495	U	7	AF	H	POFRILLO	21-Jun-73	568.45	200	69	TC	VCU	TU	209245.51	260025.26	586452.60	4105321.41	209245.54	260025.17	586452.634105321.32	1282.31
496	U	7	AG	H	OBAR	30-Apr-75	569	200	69	TC	VCU	TU	209092.82	261891.49	586293.40	4107186.63	209091.55	261891.43	586292.134107186.57	1306.35
497	U	7	AH	H	MIZZEN	3-Jun-75	637	200	67	TN	VCU	TU	208453.89	260330.3	585660.11	4105623.60	208454.06	260330.17	585660.284105623.47	1274
498	U	7	AI	H	KEELSON	4-Feb-76	640	200	67	TN	VCU	TU	209001.41	257495.78	586217.43	4102791.72	209000.89	257495.15	586216.914102791.09	1244.78
499	U	7	AJ	S	RUDDER	28-Dec-76	6													

503	U	7	AN	H	BILLET	27-Jul-76	635.5	150	61	TN	VCU	TU	207782.59	258166.13	584996.57	4103457.62	207782.96	258166.32	584996.944103457.81	1249.52
504	U	7	AO	H	PINEAU	16-Jul-81	204.2	20	41	TMA	VA	TU	209946.56	259659.66	587154.75	4104958.36	209934.62	259676.39	587142.764104975.05	1286.38
505	U	7	AP	H	CREWLINE	25-May-77	563.9	150	63	TN	VCU	TU	207675.98	260269.19	584882.61	4105559.77	207677.35	260269.88	584883.984105560.47	1264.28
506	U	7	AQ	H	SANDREEF	9-Nov-77	701	150	60	TBG	VCU	TU	207233.99	257800.36	584449.39	4103090.02	207233.6	257800.67	584449.004103090.33	1247.98
507	U	7	AT	H	CHES	20-Jun-79	335.3	20	37	TN	VCU	TU	210312.37	261762.67	587513.09	4107062.12	210312.28	261764.65	587512.994107064.11	1336.23
508	U	7	AU	H	RUMMY	27-Sep-78	640	150	61	TX	VA	TU	207112.05	258653.84	584324.49	4103942.86	207113.34	258652.99	584325.784103942.01	1252.91
509	U	7	AV	H	LOWBALL	12-Jul-78	563.9	150	63	TN	VCU	TU	207782.94	258532.23	584995.64	4103823.63	207783.08	258531.53	584995.784103822.93	1252.43
510	U	7	AX	H	BACCARAT	24-Jan-79	326	20	37	TN	VCU	TU	210617.19	261518.9	587818.69	4106819.49	210617.56	261518.96	587819.064106819.55	1337.83
511	U	7	AY	H	TOPMAST	23-Mar-78	457.2	20	34	TN	VCU	TU	209916.09	260787.41	587120.33	4106085.72	209915.92	260787.44	587120.164106085.75	1309.33
512	U	7	B	H	AUK	2-Oct-64	452.38	20	34	TN	VCU	TU	210921.96	258470.94	588134.07	4103773.37	210921.73	258470.64	588133.854103773.07	1280.94
513	U	7	BA	H	BASEBALL	15-Jan-81	563.9	150	63	TN	VCU	TU	207691.18	259466.11	584900.62	4104756.95	-99.9	-99.9	-99.9	1258.61
514	U	7	BD	H	PALIZA	1-Oct-81	472.4	150	66	TN	VCU	TU	210895.54	258880.45	588106.22	4104182.68	210895.93	258879.62	588106.624104181.85	1287.32
515	U	7	BE	H	PYRAMID	16-Apr-80	579.1	150	62	TN	VCU	TU	208944.7	261031.34	586148.33	4106326.18	208944.83	261031.65	586148.464106326.49	1292.75
516	U	7	BG	H	ALIGOTE	29-May-81	320	20	37	TOS	VCU	TU	211296.91	261128.73	588499.60	4106431.80	211297.4	261128.64	588500.104106431.71	1337.47
517	U	7	BH	H	FARADA	26-May-83	384	20	35	TN	VCU	TU	211150.59	261244.6	588352.92	4106547.13	211150.5	261244.44	588352.834106546.97	1338.82
518	U	7	BI	H	DOLCETTO	30-Aug-84	365.8	20	36	TN	VCU	TU	211723.62	259793.82	588930.89	4105098.72	211723.62	259794.27	588930.894105099.18	1317.88
519	U	7	BK	H	MULESHOE	15-Nov-89	243.8	20	40	TN	VCU	TU	210464.94	261640.62	587666.05	4106940.64	-99.9	-99.9	-99.9	1338.24
520	U	7	BL	H	TADO	5-Jun-86	518.2	150	64	TN	VCU	TU	210282	260726.38	587486.36	4106025.99	-99.9	-99.9	-99.9	1315.76
521	U	7	BM	H	DUTCHES	24-Oct-80	427	20	34	TN	VCU	TU	211744.82	258104.77	588958.01	4103410.18	211744.4	258104.74	588957.594103410.14	1291.75
522	U	7	BO	H	MUNDO	1-May-84	567	150	63	TN	VCU	TU	209665.21	261602.51	586866.66	4106899.73	209665.49	261602.85	586866.934106900.08	1319.06
523	U	7	BP	H	ATRISCO	5-Aug-82	640.1	138	59	TOY	VCU	TU	211089.72	259168.87	588299.34	4104471.71	-99.9	-99.9	-99.9	1294.77
524	U	7	BR	H	BORREGO	29-Sep-82	563.9	150	63	TN	VCU	TU	207675.91	259933.74	584883.72	4105224.41	207675.7	259933.74	584883.514105224.41	1261.09
525	U	7	BS	H	HERMOSA	2-Apr-85	640.1	150	61	TN	VCU	TU	208788.41	260330.08	585994.54	4105624.55	208788.1	260330.26	585994.234105624.73	1277.78
526	U	7	BU	H	TURQUOISE	14-Apr-83	533	150	64	TX	VA	TU	207594.15	257883.74	584809.17	4103174.64	207594.56	257884.03	584809.584103174.94	1246.15
527	U	7	BV	H	POMIL	27-Sep-85	365.8	20	36	TN	VCU	TU	211510.38	259793.6	588717.71	4105097.76	211510.1	259793.51	588717.434105097.66	1310.64
528	U	7	BY	H	MIDLAND	16-Jul-87	487.7	150	65	TN	VCU	TU	209580.95	261305.68	586783.46	4106602.68	209580.65	261305.25	586783.164106602.25	1310.58
529	U	7	CA	H	TOKARANA	10-Feb-89	502.9	150	65	TN	VCU	TU	211623.05	258348.93	588835.42	4103653.85	-99.9	-99.9	-99.9	1293.58
530	U	7	CB	H	FLOYDADA	15-Aug-91	502.9	20	33	TOY	VCU	TU	211510.19	259519.12	588718.48	4104823.34	211510.53	259519.33	588718.824104823.56	1306.52
531	U	7	E	H	PIRANHA	13-May-66	548.72	200	70	TBG	VCU	TU	208696.82	259446.34	585906.08	4104740.71	208695.79	259446.6	585905.044104741.17	1263.71
532	U	7	F	H	BROWN	23-Jul-65	530.81	200	70	TX	VA	TU	208727.42	260665.23	585932.39	4105959.40	208727.74	260664.65	585932.724105958.82	1284.23
533	U	7	G	H	CHARCOAL	10-Sep-65	455.37	200	73	TX	VA	TU	210190.42	258470.66	587402.72	4103770.52	210188.56	258470.68	587400.864103770.53	1271.61
534	U	7	H	H	CABRESTO	24-May-73	197.82	20	42	QTMA	AA	AL	208940.68	258470.93	586153.30	4103766.41	208940.67	258470.75	586153.294103766.33	1255.15
535	U	7	I	H	LAMPBLACK	18-Jan-66	561.48	200	69	TN	VCU	TU	210005.22	259993.44	587212.22	4105292.27	210004.04	259992.94	587211.054105291.76	1294.07
536	U	7	J	H	LINE	1-Apr-66	561.44	20	32	TN	VCU	TU	209885.54	261213.96	587088.29	4106512.06	209885.75	261214.26	587088.504106512.36	1318.4
537	U	7	K	H	TAN	3-Jun-66	560.68	200	69	TX	VA	TU	208544.53	257404.25	585760.99	4102698.61	208545.57	257404.67	585762.034102699.03	1240.41
538	U	7	M	H	MICKEY	10-May-67	499.65	200	71	TN	VCU	TU	212095.37	258470.82	589307.19	4103777.36	212095.31	258470.63	589307.134103777.17	1304.68
539	U	7	N	H	BOURBON	20-Jan-67	559.72	200	69	OP	LCA	DO	211318.2	260909.34	588521.66	4106212.54	211319.25	260909.4	588522.714106212.60	1333.27
540	U	7	O	H	DAQUIRI	23-Sep-66	561.15	20	32	TN	VCU	TU	208483.65	261275.35	585686.55	4106568.51	-99.9	-99.9	-99.9	1287.3
541	U	7	P	H	BLENTON	30-Apr-69	557.73	200	69	TN	VCU	TU	210434.33	258867	587645.18	4104167.62	-99.9	-99.9	-99.9	1281.69
542	U	7	R	H	SHAPER	23-Mar-70	560.44	200	69	TN	VCU	TU	209794.55	259385.01	587003.74	4104683.25	209794.01	259384.51	587003.214104682.75	1278.67
543	U	7	S	H	GRAPE A	17-Dec-69	550.67	200	70	TN	VCU	TU	211531.76	259080.52	588741.58	4104384.93	211533.64	259081.8	588743.464104386.22	1300.9
544	U	7	T	H	THISTLE	30-Apr-69	560.47	200	69	TN	VCU	TU	211165.7	259842.27	588372.94	4105145.21	211165.82	259846.03	588373.054105148.96	1308.01
545	U	7	U	H	COBBLER	8-Nov-67	667.12	20	31	TN	VCU	TU	208483.34	259994.73	585690.73	4105288.28	208483.86	259995.36	585691.254105288.85	1269.44
546	U	7	V	H	GRAPE B	4-Feb-70	554.03	200	69	QTMA	AA	TU	209306.5	260696.02	586511.22	4105992.21	209305.73	260696.62	586510.454105992.81	1296.31
547	U	7	W	H	TORRIDO	27-May-69	514.72	200	71	TN	VCU	TU	212095.51	258165.95	589308.40	4103472.57	212095.81	258165.68	589308.704103472.30	1297.31
548	U	7	X	H	ARTESIA	16-Dec-70	484.83	200	72	TN	VCU	TU	210952.33	260939.51	588155.78	4106241.42	210952.35	260939.26	588155.804106241.17	1328.99
549	U	7	Y	H	TIGERAS	14-Oct-70	560.62	200	69	TC	VCU	TU	211226.51	257666.11	588441.37	4102969.81	211225.9	257667.81	588440.754102971.50	1276.14
550	U	7	Z	H	OSCURO	21-Sep-72	560.22	200	69	TX	VA	TU	208422.4	258912.69	585633.60	4104206.24	208422.42	258909.3	585633.634104202.85	1251.99
551	U	8	A	H	DISCUS THROW	27-May-66	337.06	22	38	TX	VA	TU	202917.57	269567.86	580092.77	4114839.34	202917.75	269567.45	580092.954114838.93	1406.5
552	U	8	B	H	CYATHUS	6-Mar-70	293.88	8.7	29	TX	VA	TU	203454.43	268986.27	580631.53	4114259.79	203454.21	268985.88	580631.314114259.40	1379.68
553	U	8	C	H	NORBO	8-Mar-80	271	20	39	TO	VCU	TU	204216.54	269748.51	581390.77	4115024.51	204216.47	269748.82	581390.704115024.82	1376.06
554	U	8	D	H	BANEBERRY	18-Dec-70	277.98	10	30	TMA	VA	TU	202823.25	268983.82	580000.52	4114255.13	202823.13	268983.44	580000.414114254.74	1394.01
555	U	8	E	H	CARRPHILLY	27-Sep-78	420	20	35	TN	VCU	TU	203863.4	268755.02	581041.21	4114030.04	203863.43	268755.25	581041.244114030.27	1367.86
556	U	8	E	H	CREMINO	27-Sep-78	210	20	41	QTMA	AA	AL	203863.4	268755.02	581041.21	4114030.04	203863.43	268755.25	581041.244114030.27	1367.86
557	U	8	J	H	COTTAGE	23-Mar-85	515	150	64	TOF	VCU	TU	203698.16	269478.53	580872.52	4115022.71	203698.14	269478.89	580872.504115023.07	1388.76
558	U	8	K	H	VIDE	30-Apr-81	323	20	37	TO	VCU	TU	204072.5	269457.79	581247.79	4114733.36	204072.54	269457.66	581247.834114733.23	1373.37
559	U	8	L	H	SECO	25-Feb-81	200	20	42	TN	VCU	TU	204114.81	269964.5	581288.30	4115240.09	204114.81	269964.59	581288.304115240.18	1383.04
560	U	8	M	H	FRISCO	23-Sep-82	451	150	66	TOF	VCU	TU	203805	269169.39	58					

566	U	9	AC	H	TOYAH	15-Mar-63	130.45	20	46	QTMA	AA	AL	207660.57	263765.07	584854.93	4109054.71	207660.45	263764.69	584854.824109054.33	1279.1
567	U	9	AD	H	MISSISSIPPI	5-Oct-62	493.78	115	59	TMR	VA	TU	207157.8	265268.03	584347.01	4110555.52	207158.29	265267.86	584347.514110555.35	1290.65
568	U	9	AE	H	STONES	22-May-63	393.19	200	76	TX	VA	TU	208178.72	262128.25	585378.70	4107420.12	208179.12	262127.95	585379.104107419.83	1284.09
569	U	9	AF	H	MANATEE	14-Dec-62	59.74	20	56	QTMA	AA	AL	208089.71	263591.49	585284.57	4108882.68	208089.85	263591.63	585284.714108882.82	1280.68
570	U	9	AH	H	PLEASANT	29-May-63	210.92	20	41	QTMA	AA	AL	207864.56	264015.05	585057.99	4109305.34	-99.9	-99.9		1279.49
571	U	9	AI	H	APSHAPA	6-Jun-63	89	20	51	QTMA	AA	AL	208087.5	263637.33	585282.20	4108928.50	208087.66	263637.66	585282.364108928.83	1281.23
572	U	9	AJ	H	GARDEN	23-Oct-64	149.66	20	45	TMR	VA	TU	208878.14	262797.09	586075.59	4108091.25	208877.87	262797.37	586075.324108091.53	1301.19
573	U	9	AK	H	KOHOCTON	23-Aug-63	254.81	20	39	TMR	VA	TU	208495.41	263674.21	585689.88	4108966.80	208495.61	263673.72	585690.084108966.31	1300.52
574	U	9	AK	1	NATCHEES	23-Aug-63	59.13	20	56	QTMA	AA	AL	208507.6	263674.04	585702.07	4108966.67	208507.85	263674.65	585702.324108967.29	1288.36
575	U	9	AL	H	AJAX	11-Nov-66	239.88	20	40	QTMA	AA	AL	207203.12	264719.34	584394.25	4110007.13	207203.63	264719.25	584394.764110007.04	1280.77
576	U	9	AO	H	FORE	16-Jan-64	491.03	200	72	TX	VA	TU	207264.26	265588.06	584452.32	4110875.84	207264.61	265587.47	584452.684110875.25	1289.55
577	U	9	AP	H	RACK	15-Aug-68	199.64	20	42	QTMA	AA	AL	207355.82	263530.3	584551.09	4108818.93	207355.77	263530.25	584551.044108818.88	1280.53
578	U	9	AQ	H	TORNILLO	11-Oct-63	149.66	20	45	QTMA	AA	AL	208635.77	262984.86	585832.63	4108278.12	208635.75	262984.94	585832.604108278.20	1293.51
579	U	9	AR	H	DRIVER	7-May-64	149.96	20	45	QTMA	AA	AL	208062.79	263149.71	585259.21	4108440.92	208062.87	263149.54	585259.294108440.75	1280.25
580	U	9	AT	H	MUSTANG	15-Nov-63	165.81	20	44	QTMA	AA	AL	207468.41	264481.2	584660.31	4109769.98	207468.72	264480.8	584660.624109769.58	1283.73
581	U	9	AU	H	BOGEY	17-Apr-64	119.18	20	47	QTMA	AA	AL	208632.34	263067.46	585828.90	4108360.69	208632.55	263067.84	585829.114108361.07	1291.78
582	U	9	AV	H	EAGLE	12-Dec-63	164.9	5.3	28	QTMA	AA	AL	207736.54	264341.23	584928.86	4109630.99	207736.59	264340.95	584928.914109630.71	1281.2
583	U	9	AW	H	BACKSWING	14-May-64	160.63	20	44	QTMA	AA	AL	208184.77	262827.6	585382.29	4108119.32	208184.75	262827.44	585382.274108119.16	1284.19
584	U	9	AX	H	GREYS	22-Nov-63	301.14	200	81	QTMA	AA	AL	207630.2	263042.63	584827.11	4108332.34	207629.83	263042.67	584826.804108332.39	1274.74
585	U	9	AY	H	OCOMTO	23-Jan-64	264.87	10.5	31	TX	VA	TU	208413.67	263835.44	585607.59	4109127.70	208413.76	263835.28	585607.694109127.54	1286.59
586	U	9	AZ	H	TINDERBOX	22-Nov-68	441.96	20	34	TX	VA	TU	207873.92	265328.95	585062.74	4110618.94	207874.67	265328.77	585063.494110618.76	1287.63
587	U	9	B	H	WHITE	25-May-62	192.63	20	42	QTMA	AA	AL	207020.15	263653.11	584215.07	4108940.53	207021.06	263651.28	584215.994108938.70	1280.31
588	U	9	BA	H	HANDICAP	12-Mar-64	143.56	20	45	QTMA	AA	AL	208407.31	264225.49	585599.87	4109517.63	208407.37	264225.58	585599.934109517.72	1287.75
589	U	9	BB	H	BUNKER	13-Feb-64	227.08	20	40	TN	VCU	TU	208760.99	264458.82	585952.64	4109752.15	208760.88	264459	585952.534109752.33	1296.47
590	U	9	BC	H	HOOK	14-Apr-64	204.22	20	41	TN	VCU	TU	208986.56	264117.21	586179.35	4109411.41	208986.49	264117.13	586179.284109411.33	1302.72
591	U	9	BD	H	SPOON	11-Sep-64	179.83	20	43	TMR	VA	TU	209382.67	262433.01	586581.27	4107729.03	209382.93	262432.62	586581.534107728.64	1315.61
592	U	9	BE	H	FADE	25-Jun-64	205.13	20	41	TMR	VA	TU	209093.26	262143.81	586292.95	4107438.89	209093.5	262143.45	586293.194107438.53	1306.82
593	U	9	BF	H	LINKS	23-Jul-64	120.4	20	47	QTMA	AA	AL	208803.59	262433.16	586002.34	4107727.15	208803.38	262433.31	586002.634107727.30	1300.1
594	U	9	BG	H	CHENILLE	22-Apr-65	140.82	20	45	QTMA	AA	AL	208940.83	262174.7	586140.45	4107469.24	208940.62	262174.98	586140.244107469.51	1303.3
595	U	9	BH	H	WOOL	14-Jan-65	216.1	20	41	TN	VCU	TU	209443.53	263021.57	586640.05	4108317.66	209443.8	263021.06	586640.324108317.15	1313.6
596	U	9	BI	1	TERRINE	18-Dec-69	457.2	200	73	TN	VCU	TU	208553.47	263177.1	585749.67	4108470.02	208554.35	263176.19	585750.554108469.11	1290.7
597	U	9	BI	2	TERRINE	18-Dec-69	417.58	200	75	TN	VCU	TU	209101.27	263203.76	586297.24	4108498.60	209100.8	263203.34	586296.774108498.18	1304.06
598	U	9	BJ	H	TICKING	21-Aug-65	208.48	20	41	TMR	VA	TU	209336.94	262311.5	586535.98	4107607.39	209337.59	262311.02	586536.634107606.91	1314.82
599	U	9	BK	H	SUDE	20-Mar-65	143.26	20	45	TMR	VA	TU	209292.76	262621.12	586490.73	4107916.84	209292.76	262621.1	586490.734107916.76	1312.65
600	U	9	BM	H	SEERSUCKER	19-Feb-65	144.17	20	45	TMR	VA	TU	208892.45	262824.28	586089.80	4108118.32	208892.21	262823.96	586089.574108118.16	1301.16
601	U	9	BN	H	TWED	21-May-65	284.38	20	38	TN	VCU	TU	209184.7	262966.67	586381.48	4108261.86	209184.6	262966.88	586381.384108262.07	1308.14
602	U	9	BO	H	ORGANDY	11-Jun-65	168.86	20	43	TX	VA	TU	209580.56	262661.91	586778.31	4107958.57	209580.53	262661.91	586778.284107958.57	1320.49
603	U	9	BP	H	IZZER	16-Jul-65	163.68	20	44	QTMA	AA	AL	208788.54	262570.62	585986.81	4107864.52	208788.43	262570.36	585986.704107864.26	1299.52
604	U	9	BR	H	MAXWELL	13-Jan-66	182.88	20	43	TX	VA	TU	209199.87	262707.93	586397.55	4108003.24	209200.05	262707.87	586397.734108003.18	1309.94
605	U	9	BS	H	ELKHART	17-Sep-65	220.37	20	41	TMR	VA	TU	208574.91	262128.58	585774.79	4107421.84	208575.25	262128.36	585775.134107421.63	1293.91
606	U	9	BT	H	TEMPLAR	24-Mar-66	149.96	20	45	TMR	VA	TU	208862.92	262386.63	586061.82	4107680.84	208862.92	262386.93	586061.824107681.14	1301.68
607	U	9	BU	H	HULA	29-Oct-68	200.56	20	42	QTMA	AA	AL	208011.2	262372.27	585210.36	4107663.49	208011.34	262372.31	585210.504107663.53	1280.62
608	U	9	BV	H	SWITCH	22-Jun-67	302.06	20	38	TN	VCU	TU	209093.24	263743.99	586287.31	4109038.66	209093.37	263743.98	586287.444109038.65	1303.48
609	U	9	BK	H	NOGGIN	6-Sep-68	582.17	200	69	TX	VA	TU	207441.36	264899.33	584631.80	4110187.91	207441.97	264899.26	584632.414110187.84	1286.34
610	U	9	BY	H	VALISE	18-Mar-69	91.44	20	51	QTMA	AA	AL	208000.23	265244.21	585189.31	4110534.66	207999.89	265244.13	585188.984110534.58	1287.87
611	U	9	BZ	H	BIGGIN	30-Jan-69	243.84	20	40	TMR	VA	AL	208056.84	264597.1	585248.18	4109887.92	208057	264597.35	585248.344109888.17	1283.76
612	U	9	C	H	STILLWATER	8-Feb-62	181.36	3.07	23	QTMA	AA	AL	206959.54	263917.88	584153.55	4109205.02	206959.84	263917.88	584153.854109205.02	1286.26
613	U	9	CB	H	CUP	26-Mar-65	538.89	200	70	TN	VCU	TU	207812.22	266182.13	584998.06	4111471.68	207812.22	266181.81	584998.064111471.36	1293.97
614	U	9	CC	H	PLAYER	27-Aug-64	90.22	20	51	QTMA	AA	AL	208032.43	262827.48	585229.99	4108118.66	208032.39	262826.76	585229.954108117.94	1280.47
615	U	9	CE	H	CLYMER	12-Mar-66	398.07	20	35	QTMA	AA	AL	206959.53	265740.59	584147.14	4111027.26	206959.31	265740.15	584146.924111026.82	1292.93
616	U	9	CF	H	VAT	10-Oct-68	192.02	20	42	QTMA	AA	AL	207874.08	264596.94	585065.47	4109887.12	207874.28	264596.22	585065.674109886.39	1282.63
617	U	9	CG	H	GRADINO	16-Aug-77	320	20	47	QTMA	AA	AL	207264.34	265908.04	584451.28	4111195.74	207264.35	265908.11	584451.294111195.81	1289.93
618	U	9	CH	H	GRUYERE	16-Aug-77	207	20	41	QTMA	AA	AL	207264.34	265908.04	584451.28	4111195.74	207264.35	265908.11	584451.294111195.81	1289.93
619	U	9	CG	H	CATHAY	8-Oct-71	377.95	20	35	TX	VA	AL	208331.24	262433.34	585530.11	4107725.67	208329.99	262434.19	585528.864107726.52	1288.21
620	U	9	CI	H	ARSENATE	9-Nov-72	250.24	20	39	TX	VA	TU	208770.15	263285.84	585965.92	4108579.50	208770.25	263285.88	585966.024108579.54	1295.01
621	U	9	CJ	H	ALUMROOT	14-Feb-73	182.88	20	43	QTMA	AA	AL	207712.89	266121.42	584359.10	4111408.74	207172.51	266121.6	584358.724111408.92	1291.38
622	U	9	CK	H	SILENE	28-Jun-73	198.12	20	42	QTMA	AA	AL	208011.16	262555.18	585209.68	4107846.36	208011			

629	U	9	CS	H	ARMADA	22-Apr-83	265	20	39	TX	VA	TU	209657.12	262189.6	586856.51	4107486.65	-99.9	-99.9	1322.53	
630	U	9	CV	H	GALENA	23-Jun-92	290	20	38	TN	VCU	TU	208864.63	263561.06	586059.40	4108854.98	-99.9	-99.9	1295.62	
631	U	9	CV	H	GALENA	23-Jun-92	381	20	35	TN	VCU	TU	208864.63	263561.06	586059.40	4108854.98	-99.9	-99.9	1295.62	
632	U	9	CV	H	GALENA	23-Jun-92	401	20	35	TN	VCU	TU	208864.63	263561.06	586059.40	4108854.98	-99.9	-99.9	1295.62	
633	U	9	CW	H	CERRERO	14-Aug-85	183	20	43	TC	VCU	TU	210373.04	262146.39	587572.40	4107445.96	210373.73	262146.49	587573.094107446.06	1342.64
634	U	9	D	H	BRAZOS	8-Mar-62	256.34	8.4	29	QTMA	AA	AL	207298.58	263355.83	584494.47	4108644.30	207298.58	263355.23	584494.484108643.70	1280.56
635	U	9	E	H	HATCHIE	8-Feb-63	60.96	20	56	QTMA	AA	AL	208205.01	263782.99	585399.17	4109074.54	208205.09	263782.76	585399.264109074.30	1282.72
636	U	9	F	H	TOGA	18-Oct-62	59.44	20	56	QTMA	AA	AL	208061.15	264069.41	585254.34	4109360.38	208061.13	264069.49	585254.324109360.46	1280.96
637	U	9	G	H	CODSAW	19-Feb-62	212.14	20	41	TMR	VA	TU	208339.67	263949.65	585533.21	4109241.62	208339.67	263949.35	585533.214109241.33	1285.44
638	U	9	H	H	CMARRON	23-Feb-62	304.8	11.9	31	QTMA	AA	AL	207344.65	264098.72	584537.92	4109387.16	207344.65	264098.41	584537.924109386.85	1282.57
639	U	9	I	H	ANACOSTIA	27-Nov-62	226.77	20	40	TU	VCU	TU	209063.11	263439.11	586258.25	4108733.75	209062.19	263439.11	586257.344108733.75	1301.01
640	U	9	J	H	HOOSIC	28-Mar-62	186.84	3.4	23	TU	VCU	TU	208630.14	263603.7	585824.83	4108896.78	208629.52	263604.26	585824.204108897.34	1290.95
641	U	9	K	H	DEAD	21-Apr-62	193.55	20	42	QTMA	AA	AL	208843.16	263015.48	586039.85	4108309.46	208843.16	263015.79	586039.854108309.77	1298.73
642	U	9	L	H	PASSAIC	6-Apr-62	233.48	20	40	QTMA	AA	AL	207734.35	262860.55	584931.87	4108150.68	207734.35	262860.55	584931.874108150.68	1274.89
643	U	9	M	H	EEL	19-May-62	217.63	4.5	25	QTMA	AA	AL	207446.63	263409.28	584642.30	4108698.26	207445.41	263407.75	584641.084108696.72	1280.07
644	U	9	N	H	HUDSON	12-Apr-62	150.88	20	45	QTMA	AA	AL	207651.6	263914.66	584845.44	4109204.23	207651.3	263915.27	584845.444109204.84	1280.19
645	U	9	P	H	BLACK	27-Apr-62	217.63	20	41	QTMA	AA	AL	208288.54	262951.25	585485.60	4108243.30	208288.54	262951.86	585485.604108243.31	1285.47
646	U	9	Q	H	ROANOKE	12-Oct-62	176.78	20	43	QTMA	AA	AL	207126.76	263418.25	584322.48	4108706.10	207127.16	263417.71	584322.884108705.56	1279.4
647	U	9	R	H	ARIKAREE	10-May-62	166.42	20	44	QTMA	AA	AL	207346.29	263958.02	584540.06	4109246.50	207345.99	263958.63	584539.764109247.11	1281.35
648	U	9	U	H	RARIYAN	6-Sep-62	157.28	20	44	QTMA	AA	AL	207660.91	264262.37	584853.53	4109551.88	207660.6	264261.85	584853.224109551.36	1281.41
649	U	9	V	H	SACRAMENTO	30-Jun-62	149.05	20	45	QTMA	AA	AL	207432.02	262829.44	584629.73	410818.51	207432.19	262829.77	584629.90410818.85	1273.24
650	U	9	W	H	KOOTANAI	24-Apr-63	181.97	20	43	QTMA	AA	AL	208419.5	263189.25	585615.69	4108481.70	208419.57	263189.23	585615.764108481.68	1287.75
651	U	9	W	1	PAISANO	24-Apr-63	57	20	57	QTMA	AA	AL	208428.83	263181.55	585625.05	4108474.03	208427.82	263182.6	585624.034108475.08	1287.81
652	U	9	X	H	ALLEGHENY	29-Sep-62	210.92	20	41	QTMA	AA	AL	208733.41	262756.5	585931.04	4108050.16	208732.57	262756.51	585930.204108050.17	1297.72
653	U	9	Y	H	WICHITA	27-Jul-62	150.27	20	45	QTMA	AA	AL	206617.68	264189.22	583810.82	4109475.09	206617.38	264188.92	583810.524109474.79	1291.93
654	U	9	Z	H	YORK	24-Aug-62	226.47	20	40	QTMA	AA	AL	208136.05	262969.71	585333.09	4108261.22	208136.21	262969.5	585333.254108261.01	1282.72
655	U	9	IAA25	H	PITON	28-May-70	100.58	20	49	TN	VCU	TU	208971.6	265176.6	586160.67	4110470.48	208971.74	265176.56	586160.814110470.44	1305.86
656	U	9	IS 25	H	BALTIC	6-Aug-71	411.48	20	35	TN	VCU	TU	207995.82	265176.63	585185.14	4110467.09	207995.85	265176.51	585185.174110466.96	1286.87
657	U	9	IT 28	H	AVENS	16-Dec-70	379.48	20	35	TN	VCU	TU	208117.77	265542.15	585305.78	4110832.94	208117.73	265541.79	585305.744110832.58	1291.14
658	U	9	IU 24	H	AVENS	16-Dec-70	306.02	20	37	TN	VCU	TU	208239.68	265054.84	585429.37	4110346.18	208239.59	265054.88	585429.284110346.22	1288.45
659	U	9	IV 24	H	FOB	23-Jan-70	265.79	20	39	TN	VCU	TU	208361.47	265054.91	585551.13	4110346.68	208362.17	265057.47	585551.824110349.24	1289.86
660	U	9	IV 26	H	ARABIS	6-Mar-70	249.91	20	39	TN	VCU	TU	208361.52	265298.52	585550.32	4110590.23	208361.5	265298.75	585550.304110590.46	1291.87
661	U	9	IV 27	H	FOB	23-Jan-70	244.45	20	40	TN	VCU	TU	208361.48	265240.51	585549.85	4110712.19	208361.26	265420.4	585549.634110712.08	1292.93
662	U	9	IW 21	H	AVENS	16-Dec-70	306.32	20	37	TN	VCU	TU	208483.45	264689.02	585674.36	4109981.31	208483.41	264688.88	585674.324109981.17	1290.62
663	U	9	IW 22	H	HAPLOPAPPUS	28-Jun-72	184.4	20	42	TX	VA	TU	208498.79	264810.52	585689.27	4110102.84	208499.12	264810.54	585689.604110102.85	1291.29
664	U	9	IW 24	H	SOLANUM	14-Dec-72	201.17	20	42	TN	VCU	TU	208544.65	265130.94	585733.99	4110423.34	208544.62	265131.08	585733.964110423.47	1294.31
665	U	9	IX 20	H	HOD	1-May-70	265.18	20	39	TN	VCU	TU	208605.71	264566.95	585797.02	4109859.70	208605.98	264566.84	585797.294109859.59	1292.93
666	U	9	IX 23	H	HOD	1-May-70	240.79	20	40	TN	VCU	TU	208605.53	264932.86	585795.55	4110225.52	208605.67	264933.08	585795.694110225.74	1294.76
667	U	9	IX 24	H	SCREE	13-Oct-70	249.94	20	39	TN	VCU	TU	208605.61	265060.66	585795.19	4110353.29	208605.28	265060.36	585794.854110352.98	1295.98
668	U	9	IX 27	H	PITON	28-May-70	228.6	20	40	TN	VCU	TU	208606.03	265420.1	585794.34	4110712.64	208606	265420	585794.314110712.53	1296.88
669	U	9	IX 28	H	ARABIS	6-Mar-70	258.71	20	39	TN	VCU	TU	208605.6	265542.29	585793.48	4110834.79	208605.67	265542.04	585793.554110834.56	1297.27
670	U	9	IX 29	H	AVENS	16-Dec-70	294.13	20	38	TN	VCU	TU	208605.71	265664.03	585793.17	4110956.50	208605.73	265664.13	585793.184110956.60	1298.4
671	U	9	IXY31	H	NAMA	5-Aug-71	272.8	20	38	TN	VCU	TU	208666.48	265908.05	585853.06	4111200.67	208666.41	265908	585852.994111200.62	1300.34
672	U	9	IY 27	H	FOB	23-Jan-70	100.58	20	49	TN	VCU	TU	208727.77	265420.19	585916.05	4110713.15	208727.74	265420.07	585916.024110713.03	1299.65
673	U	9	IY 30	H	PITON	28-May-70	236.22	20	40	TN	VCU	TU	208727.42	265786.1	585914.42	4111078.97	208727.88	265786.76	585914.874111079.63	1300.77
674	U	9	IYZ26	H	CANNA	17-Nov-72	182.88	20	43	TN	VCU	TU	208753.4	265298.47	585942.10	4110591.55	208753.13	265298.33	585941.834110591.42	1299.98
675	U	9	IYZ26	H	CANNA	17-Nov-72	213.42	20	41	TN	VCU	TU	208753.4	265298.47	585942.10	4110591.55	208753.13	265298.33	585941.834110591.42	1299.98
676	U	9	IYZ 21	H	SCREE	13-Oct-70	192.02	20	42	TN	VCU	TU	208849.39	264689.03	586040.21	4109982.61	208849.85	264688.83	586040.674109982.41	1300.04
677	U	9	IZ 24	H	SCREE	13-Oct-70	100.58	20	49	TN	VCU	TU	208849.49	265054.04	586039.03	4110347.52	208849.28	265054.45	586038.814110347.93	1301.76
678	U	9	IZ 25	H	HOD	1-May-70	100.58	20	49	TN	VCU	TU	208849.38	265176.59	586038.49	4110470.04	208849.28	265176.52	586038.394110469.97	1302.54
679	U	9	IZ 26	H	ARABIS	6-Mar-70	100.58	20	49	TN	VCU	TU	208849.07	265298.48	586037.75	4110591.90	208849.1	265298.44	586037.784110591.86	1302.95
680	U	9	IZ 27	H	NAMA	5-Aug-71	243.84	20	40	TN	VCU	TU	208849.5	265420.22	586037.75	4110713.61	208849.58	265419.98	586037.834110713.37	1302.99
681	U	10	A	H	DUB	30-Jun-64	259.08	20	39	QTMA	AA	AL	206593.63	269151.22	583769.35	4114435.73	-99.9	-99.9	1308.48	
682	U	10	AA	H	RIVET 1	18-Jan-67	151.79	20	45	QTMA	AA	AL	207474.72	268124.05	584653.82	4113411.92	207474.71	268123.66	584653.814113411.53	1302.38
683	U	10	AB	H	VIPO	14-Jul-67	96.62	20	50	QTMA	AA	AL	207618.14	268139.22	584797.15	4113427.59	207617.94	268139.5	584796.954113427.87	1304.64
684	U	10	AD	H	VIGIL	22-Nov-66	91.44	20	51	QTMA	AA	AL	207314.64	268596.09	584592.12	4113883.28	207314.62	268596.44	584492.104113883.63	1304.79
685	U	10	AF	H	YARD	7-Sep-67	521.21	200	71	TX	VA	TU	206929.46	266792.8	584113.38	4112079.10	206929.09	266793.23	584113.014112079.52	1297.38
686	U	10	AG	H																

692	U10	AJ D H	TUB	6-Jun-68	273.1	20	38	QTMA	AA	AL	207627.89	268331.87	584806.22	4113620.23	207627.73	268331.8	584806.064113620.16	1305.03
693	U10	AJ F H	TUB	6-Jun-68	188.98	20	42	QTMA	AA	AL	207995.61	268157.81	585174.46	4113447.50	207995.99	268158.2	585174.844113447.90	1310.13
694	U10	AK H	CROCK	8-May-68	181.66	20	43	QTMA	AA	AL	208513.53	267265.76	585695.38	4112557.50	208513.2	267265.56	585695.054112557.30	1315.81
695	U10	AM 1 H	TUN	10-Dec-69	199.64	20	42	QTMA	AA	AL	204566.73	268392.32	581745.63	4113669.90	204566.94	268392.14	581745.844113669.73	1349.98
696	U10	AM 2 H	TUN	10-Dec-69	193.85	20	42	TN	VCU	TU	204566.73	268590.34	581744.94	4113867.87	204566.68	268590.23	581744.894113867.76	1351.38
697	U10	AM 3 H	TUN	10-Dec-69	193.7	20	42	TBG	VCU	TU	204764.85	268391.79	581943.70	4113670.07	204765.46	268391.49	581944.324113669.77	1345.64
698	U10	AM 4 H	TUN	10-Dec-69	256.34	20	39	TN	VCU	TU	204765.39	268589.95	581943.55	4113868.18	204764.47	268589.18	581942.634113867.41	1347.34
699	U10	AN H	LABIS	5-Feb-70	441.96	25	37	TN	VCU	TU	208160.81	267999.94	585340.17	4113290.26	208161.1	267999.75	585340.464113290.07	1313
700	U10	AP 1 H	CORNICE	15-May-70	390.14	200	76	TN	VCU	TU	208456.16	268219.87	585634.68	4113511.17	208456.07	268219.97	585634.584113511.27	1319.36
701	U10	AP 3 H	CORNICE	15-May-70	443.48	200	73	TN	VCU	TU	208160.66	267773.29	585340.82	4113063.66	208161.96	267774.57	585342.114113064.95	1312.41
702	U10	AQ H	BRACKEN	9-Jul-71	304.8	20	37	TN	VCU	TU	208702.17	268057.72	585881.19	4113349.92	208702.46	268057.97	585881.484113350.17	1323.48
703	U10	AR H	LAGOON	14-Oct-71	304.8	20	37	QTMA	AA	AL	206883.01	269747.06	584056.56	4115032.44	206883.01	269746.75	584056.564115032.13	1312.62
704	U10	AS H	PINEDROPS	10-Jan-74	213.36	20	41	QTMA	AA	AL	207090.67	269120.59	584266.37	4114406.86	207090.56	269120.37	584266.264114406.64	1309.88
705	U10	AS H	PINEDROPS	10-Jan-74	281.94	20	38	QTMA	AA	AL	207090.67	269120.59	584266.37	4114406.86	207090.56	269120.37	584266.264114406.64	1309.88
706	U10	AS H	PINEDROPS	10-Jan-74	342.9	20	36	TN	VCU	TU	207090.67	269120.59	584266.37	4114406.86	207090.56	269120.37	584266.264114406.64	1309.88
707	U10	AT H	DIANTHUS	17-Feb-72	304.8	20	37	QTMA	AA	AL	206613.54	268182.99	583792.65	4113467.82	206613.12	268182.88	583792.244113467.71	1305.95
708	U10	AV H	KASHAN	24-May-73	265.18	20	39	QTMA	AA	AL	206639.63	267797.75	583820.09	4113082.77	206639.02	267798.05	583819.484113083.07	1303.79
709	U10	AW H	NATOMA	5-Apr-73	243.84	20	40	QTMA	AA	AL	206813.28	269535.19	583987.59	4114820.38	-99.9	-99.9	583987.594114820.38	1309.85
710	U10	AX H	AKBAR	9-Nov-72	266.7	20	39	TN	VCU	TU	208666.42	267790.27	585846.39	4113082.41	208666.58	267789.96	585846.554113082.10	1321.98
711	U10	AY H	CHEVRE	23-Nov-76	317	20	37	QTMA	AA	AL	206926.09	268851.15	584102.78	4114136.91	206926.13	268851.15	584102.814114136.91	1304.97
712	U10	B H	HANDCAR	5-Nov-64	403.25	12	29	CBK	LCA	DO	205649.01	269138.92	582825.01	4114420.12	205650.81	268831.28	582826.814114419.47	1333.41
713	U10	BA H	DOPINO	8-Mar-77	183	20	43	QTMA	AA	AL	206897.79	269349.22	584072.73	4114634.75	206897.93	269349.32	584072.874114634.85	1309.55
714	U10	BA H	DOPINO	8-Mar-77	282	20	38	QTMA	AA	AL	206897.79	269349.22	584072.73	4114634.75	206897.93	269349.32	584072.874114634.85	1309.55
715	U10	BB H	LARKIN	6-Feb-75	274.3	20	38	QTMA	AA	AL	207020.46	269595.48	584194.50	4114881.38	207020.46	269595.42	584194.504114881.32	1312.29
716	U10	BB H	PORTOLA	6-Feb-75	198.1	20	42	QTMA	AA	AL	207020.46	269595.48	584194.50	4114881.38	207020.46	269595.42	584194.504114881.32	1312.29
717	U10	BC H	ASCO	25-Apr-78	183	20	43	TN	VCU	TU	208529.31	266959.65	585712.23	4112251.52	208529.68	266959.8	585712.604112251.68	1312.19
718	U10	BD H	PERA	8-Sep-79	200	20	42	TX	VA	TU	208224.48	267005.39	585407.32	4112296.18	208224.64	267005.61	585407.484112296.40	1307.23
719	U10	BE H	ORKNEY	2-May-84	210	20	41	QTMA	AA	AL	206791.92	271821.15	583958.20	4117105.68	-99.9	-99.9	583958.204117105.68	1378.34
720	U10	BF H	QUESO	11-Aug-82	216	20	41	QTMA	AA	AL	207355.87	270861.12	584525.38	4116147.88	207355.36	270860.21	584524.874116146.96	1337.16
721	U10	BG H	HAVARTI	5-Aug-81	200	20	42	TN	VCU	TU	208502.04	266868.17	585685.29	4112159.97	-99.9	-99.9	585685.294112159.97	1309.91
722	U10	BH H	HAZE BROOK	3-Feb-87	186	20	42	QTMA	AA	AL	207294.95	269900.89	584467.85	4115187.68	-99.9	-99.9	584467.854115187.68	1317.38
723	U10	BH H	HAZE BROOK	3-Feb-87	226	20	40	QTMA	AA	AL	207294.95	269900.89	584467.85	4115187.68	-99.9	-99.9	584467.854115187.68	1317.38
724	U10	BB H	HAZE BROOK	3-Feb-87	262	20	39	QTMA	AA	AL	207294.95	269900.89	584467.85	4115187.68	-99.9	-99.9	584467.854115187.68	1317.38
725	U10	C H	TURF	24-Apr-64	506.27	200	71	TMR	VA	TU	206700.65	266395.85	583886.02	4111681.44	206700.63	266395.25	583886.004111680.84	1298.45
726	U10	CA H	JARLSBERG	27-Aug-83	200	20	42	TOS	VCU	TU	208564.98	271218.22	585732.93	4116509.14	208565.06	271218.3	585733.014116509.22	1344.42
727	U10	CB H	NORMANNA	12-Jul-84	200	20	41	TOS	VCU	TU	208535.63	271112.33	585703.95	4116403.17	-99.9	-99.9	585703.954116403.17	1342.34
728	U10	CC H	BRIE	18-Jun-87	203	20	41	TOS	VCU	TU	208478.04	271285.89	585645.77	4116576.48	208478.16	271285.81	585645.894116576.40	1345.19
729	U10	D S H	DUFFER	18-Jun-64	446.84	20	34	TN	VCU	TU	208196.61	268225.7	585375.17	4113516.08	208197.06	268225.47	585375.624113515.85	1314.39
730	U10	D S1 H	MARVEL	21-Sep-67	175.87	2.2	21	QTMA	AA	AL	208196.25	268228.18	585374.80	4113518.56	-99.9	-99.9	585374.804113518.56	1314.39
731	U10	E H	KLICKITAT	20-Feb-64	492.56	200	72	TN	VCU	TU	208056.73	266548.04	585241.22	4111838.36	208056.24	266547.88	585240.734111838.20	1300.34
732	U10	F H	SANTEE	27-Oct-62	318.52	20	37	QTMA	AA	AL	206868.31	266365.27	584053.74	4111651.46	206868.31	266365.27	584053.744111651.46	1296.47
733	U10	G H	CASSELMAN	8-Feb-63	302.97	20	38	QTMA	AA	AL	207013.12	266318.07	584198.68	4111604.78	207013.26	266317.9	584198.824111604.61	1294.61
734	U10	H H	SEDAN	6-Jul-62	193.55	104	73	QTMA	AA	AL	207569.17	269443.76	584743.61	4114731.63	207570.12	269441.94	584744.564114729.81	1315.79
735	U10	I H	BYE	16-Jul-64	390.75	200	76	TN	VCU	TU	207568.81	270020.94	584741.22	4115308.66	207568.93	270020.39	584741.344115308.11	1321.4
736	U10	K H	CORDUROY	3-Dec-65	678.79	200	66	TN	VCU	TU	206959.65	268078.34	584139.04	4113364.41	206959.54	268078.23	584138.934113364.30	1302.29
737	U10	M H	REO	22-Jan-66	208.18	20	41	QTMA	AA	AL	208178.41	267250.24	585360.40	4112540.81	208178.44	267250.08	585360.434112540.65	1309.45
738	U10	N H	MUDPACK	16-Dec-64	152.1	2.7	23	TN	VCU	TU	205648.8	269519.98	582823.46	4114801.08	205649.07	269519.81	582823.734114801.01	1336.83
739	U10	P H	KANKAKEE	15-Jun-66	454.76	200	73	CBK	LCA	DO	207263.65	268834.32	584440.31	4114121.27	207263.79	268834.01	584440.454114121.46	1308.09
740	U10	Q H	STANLEY	27-Jul-67	483.72	200	72	TX	VA	TU	207310.26	266304.18	584495.80	4111591.94	207310.35	266304.57	584495.894111591.83	1291.52
741	U10	R H	WASHER	10-Aug-67	466.34	20	34	TU	VCU	TU	207416.43	267188.24	584598.83	4112476.15	207416.48	267188.39	584598.884112476.30	1297.65
742	U10	S H	ROVENA	10-Aug-66	195.07	20	42	QTMA	AA	AL	207363.97	268516.91	584541.72	4113804.29	207363.74	268517.32	584541.494113804.70	1304.7
743	U10	T H	SHUFFLE	18-Apr-68	493.17	200	72	TOF	VCU	TU	208330.81	266730.73	585514.59	4112021.97	208331.29	266729.2	585515.074112020.44	1306.18
744	U10	U H	NEWARK	29-Sep-66	229.21	20	40	QTMA	AA	AL	207516.46	268516.88	584694.17	4113804.80	207514.87	268517.41	584692.584113805.32	1306
745	U10	W H	SIMMS	5-Nov-66	198.73	20	42	QTMA	AA	AL	207412.33	268660.01	584589.56	4113947.53	207412.33	268660.01	584589.564113947.53	1306.31
746	U10	X H	WARD	8-Feb-67	259.99	20	39	QTMA	AA	AL	207421.18	268390.39	584599.36	4113678.01	207420.57	268390.39	584598.754113678.00	1303.79
747	U10	Y H	RIVET 3	2-Mar-67	273.71	20	38	QTMA	AA	AL	207282.95	268206.05	584461.81	4113493.23	207282.04	268206.35	584460.904113493.52	1301.56
748	U10	Z H	RIVET 2	26-Jan-67	197.51	20	42	QTMA	AA	AL	207340.76	268087.32	584520.03	4113374.73	207341.07	268087.01	584520.334113374.42	1300.65
749	U11	B H	PINSTRIPE	25-Apr-66	295.66	20	38	TX	VA	TU	217109.35	237378.67	594393.73	4082708.03	-99.9	-99.9	594393.734082708.03	1093.02
750	U11	C H	NEWPOINT	13-Dec-66	239.27	20	40	QTMA	AA	AL	217367.66	236249.15	594555.92	4081579.69	217367.74	236249.04	594556.004081579.58	1030.47
751	U11	E H	DIANA MOON	27-Aug-68	242.01													

755	U12	B.02	H	TAMALPIAS	8-Oct-58	100.58	0.072	8	TX	TC,TCBCUTU	193789	271395	570960.16	4116633.91	-99.9	-99.9	-99.9	
756	U12	B.04	H	EVANS	29-Oct-58	256.03	0.055	5	TX	TC,TCBCUTU	193408	271340	570579.46	4116577.59	-99.9	-99.9	-99.9	
757	U12	B.08	H	FEATHER	22-Dec-61	247.5	150	77	TX	TC	TU	193093.83	271350.5	570265.34	411586.98	-99.9	-99.9	2268.63
758	U12	B.09	H	CHENA	10-Oct-61	255.42	20	39	TX	TC,TCBCUTU	193206.89	271279.73	570378.61	4116516.63	-99.9	-99.9	2277.34	
759	U12	B.10	H	YUBA	5-Jun-63	242.62	3.1	21	TX	TC,TCBCUTU	193014.53	271532.61	570185.42	4116768.76	-99.9	-99.9	2267.11	
760	U12	C.01	H	URANUS	14-Mar-58	34.75	0.001	2	TN	BCU	TU	-99.9	-99.9	-99.9	-99.9	-99.9	-99.9	-99.9
761	U12	C.02	H	SATURN	9-Aug-57	39.01	0	0		TU		-99.9	-99.9	-99.9	-99.9	-99.9	-99.9	-99.9
762	U12	C.03	H	NEPTUNE	14-Oct-58	30.02	115	120	TN	BCU	TU	193858	271235	571029.71	4116474.20	-99.9	-99.9	-99.9
763	U12	D	H	MARS	28-Sep-58	38.1	0.013	5	TN	BCU	TU	193792	271146	570964.04	4116384.99	-99.9	-99.9	-99.9
764	U12	D.01	H	VENUS	22-Feb-58	30.48	0.001	2	TN	BCU	TU	-99.9	-99.9	-99.9	-99.9	-99.9	-99.9	-99.9
765	U12	E.02	H	LOGAN	16-Oct-58	283.46	5	24	TN	BCU	TU	193731	270145	570906.57	4115384.04	-99.9	-99.9	2155.25
766	U12	E.03	H	ANTLER	15-Sep-61	402.03	2.6	18	TN	BCU	TU	193144	270577	570318.21	4115813.86	-99.9	-99.9	2280.95
767	U12	E.05	H	BLANCA	30-Oct-58	301.14	22	39	TN	BCU	TU	193656	270355	570830.86	4115593.72	-99.9	-99.9	2172.01
768	U12	E.10	H	DORSAL FIN	29-Feb-68	409.96	20	35	TN	BCU	TU	192819.74	270204.82	569995.35	4115440.64	-99.9	-99.9	2287.53
769	U12	E.11	H	DIESEL TRAIN	5-Dec-69	419.13	20	35	TN	BCU	TU	192873.45	269690.13	570050.85	4114926.28	-99.9	-99.9	2233.88
770	U12	E.12	H	HUDSON MOON	26-May-70	422.45	20	35	TN	BCU	TU	192646.89	269980.68	569823.33	4115215.95	-99.9	-99.9	2301.25
771	U12	E.14	H	DIDO QUEEN	5-Jun-73	391.36	20	35	TN	BCU	TU	192494.7	270248.54	569670.24	4115483.21	-99.9	-99.9	2274.02
772	U12	E.18	H	DINING CAR	5-Apr-75	383	20	35	TN	BCU	TU	192601.35	270569.64	569775.73	4115804.60	-99.9	-99.9	2264.71
773	U12	E.20	H	HYBLA GOLD	1-Nov-77	385	20	35	TN	BCU	TU	192683.03	270555.07	569857.44	4115790.32	-99.9	-99.9	2266
774	U12	F.01	H	MERCURY	23-Sep-58	55.787#VALUE!	0	0	TN	BCU	TU	-99.9	-99.9	-99.9	-99.9	-99.9	-99.9	-99.9
775	U12	G.01	H	MADISON	12-Dec-62	402.34	20	35	TN	BCU	TU	193670	268829	570850.22	4114068.18	-99.9	-99.9	2279
776	U12	G.06	H	RED HOT	5-Mar-66	405.38	20	35	TN	BCU	TU	193099	269082.54	570278.48	4114319.64	-99.9	-99.9	2326.36
777	U12	G.07	H	DOOR MIST	31-Aug-67	445.95	20	34	TN	BCU	TU	193045.52	269422.56	570223.82	4114659.38	-99.9	-99.9	2322.28
778	U12	G.09	H	CYPRESS	12-Feb-69	411.48	20	35	TN	VCU	TU	192892.26	268477.46	570073.92	4113713.99	-99.9	-99.9	2291.8
779	U12	G.10	H	CAMPBOR	29-Jun-71	423.67	20	34	TN	BCU	TU	192819.04	269334.96	569997.70	4114571.01	-99.9	-99.9	2316.49
780	U12	J.01	H	DES MOINES	13-Jun-62	201.17	2.9	22	TX	TC,TCBCUTU	197175	274400	574334.70	4119650.03	-99.9	-99.9	1722.73	
781	U12	K.01	H	PLATTE	14-Apr-62	191.41	1.85	19	TX	TC,TCBCUTU	197592	274380	574751.66	4119631.50	-99.9	-99.9	1722.12	
782	U12	N.02	H	MIDMIST	26-Jun-67	377.04	20	36	TN	BCU	TU	193130.76	272146.98	570299.45	4117383.38	-99.9	-99.9	2226.87
783	U12	N.04	H	HUDSON SEAL	24-Sep-68	344.42	20	36	TN	BCU	TU	193256.22	272442.37	570423.84	4117679.13	-99.9	-99.9	2194.56
784	U12	N.05	H	MISTY NORTH	2-May-72	376.08	20	36	TN	BCU	TU	193047.1	272762.67	570213.65	4117998.61	-99.9	-99.9	2226.02
785	U12	N.06	H	DIANA MIST	11-Feb-70	402.03	20	35	TN	BCU	TU	193362.71	272057.23	570531.66	4117294.47	-99.9	-99.9	2255.83
786	U12	N.07	H	HUSKY ACE	12-Oct-73	413.31	20	35	TN	BCU	TU	193543.44	271958.18	570712.69	4117196.08	-99.9	-99.9	2264.69
787	U12	N.08	H	MING BLADE	19-Jun-74	388.99	20	35	TN	BCU	TU	193177.36	273057.14	570342.84	4118293.46	-99.9	-99.9	2238.55
788	U12	N.09	H	HYBLA FAIR	28-Oct-74	404.01	20	35	TN	BCU	TU	193476	272041	570644.98	4117278.64	-99.9	-99.9	2253.61
789	U12	N.10	H	MIGHTY EPIC	12-May-76	398	20	35	TN	BCU	TU	192716.38	272916.41	569882.48	4118151.15	-99.9	-99.9	2250.65
790	U12	N.10AH	H	DIABLO HAWK	13-Sep-78	388	20	35	TN	BCU	TU	192866.11	272888.28	570032.27	4118123.55	-99.9	-99.9	2239.13
791	U12	N.11	H	MINERS IRON	31-Oct-80	389.5	20	35	TN	BCU	TU	193343.43	273163.5	570508.49	4118400.38	-99.9	-99.9	2239.41
792	U12	N.12	H	MINI JADE	26-May-83	379.2	20	35	TN	BCU	TU	193375.6	272882.4	570541.64	4118119.46	-99.9	-99.9	1854.97
793	U12	N.15	H	DIAMOND ACE	23-Sep-82	407.26	20	35	TN	BCU	TU	193219.76	273247.7	570384.56	4118484.12	-99.9	-99.9	-99.9
794	U12	N.15	H	HURON LANDING	23-Sep-82	408.54	20	35	TN	BCU	TU	193219.76	273247.7	570384.56	4118484.12	-99.9	-99.9	-99.9
795	U12	N.17	H	MISTY RAIN	6-Apr-85	388.6	20	35	TN	BCU	TU	193189.76	272005.53	570358.94	4117242.17	-99.9	-99.9	2238.67
796	U12	N.18	H	TOMME/MIDNIGH	21-Sep-83	404.8	20	35	TN	BCU	TU	193000.13	272993.71	570165.88	4118229.42	-99.9	-99.9	-99.9
797	U12	N.19	H	DIAMOND BEECH	9-Oct-85	404.5	20	35	TN	BCU	TU	192926.23	272985.19	570092.03	4118220.64	-99.9	-99.9	-99.9
798	U12	N.20	H	MILL YARD	9-Oct-85	375	20	36	TN	BCU	TU	193367.58	272931.27	570533.45	4118168.29	-99.9	-99.9	2230.36
799	U12	N.21	H	MIDDLE NOTE	18-Mar-87	398.7	20	35	TN	BCU	TU	193060.71	273048.92	570226.25	4118284.83	-99.9	-99.9	2250.34
800	U12	N.22	H	MINERAL QUARR	25-Jul-90	389.4	20	35	TN	BCU	TU	192557.63	272672.72	569724.63	4117906.96	-99.9	-99.9	2243.05
801	U12	N.22	H	RANDSBURG	25-Jul-90	389.4	20	35	TN	BCU	TU	192557.63	272672.72	569724.63	4117906.96	-99.9	-99.9	2243.05
802	U12	N.23	H	MISTY ECHO	10-Dec-88	400.2	150	68	TN	BCU	TU	192989.38	271805.94	570159.31	4117041.93	-99.9	-99.9	2259.39
803	U12	N.24	H	HUNTERS TROPH	18-Sep-92	385.3	20	35	TN	BCU	TU	192937.17	272680.6	570104.04	4117916.17	-30.45	-30.45	1853.86
804	U12	N.25	H	NPE	22-Sep-93	389.87#VALUE!	0	0	TN	BCU	TU	-30.45	-30.45	-30.45	-30.45	-30.45	-30.45	-99.9
805	U12	P.02	H	MISSION CYBER	2-Dec-87	270.6	20	39	TX	TC,TCBCUTU	197057.81	275778.41	574212.69	4121027.66	-99.9	-99.9	1952.95	
806	U12	P.03	H	DISKO ELM	14-Sep-89	261.2	20	39	TX	TC,TCBCUTU	197099	275916	574253.38	4121165.36	-99.9	-99.9	-99.9	
807	U12	P.04	H	DISTANT ZENIT	19-Sep-91	263.8	20	39	TP	TC	TU	196785.81	275896.37	573940.35	4121144.64	-99.9	-99.9	1948.19
808	U12	P.05	H	DIAMOND FORTU	30-Apr-92	236	20	40	TP	TC	TU	197592.61	275729.9	574747.52	4120981.05	-30.45	-30.45	1682.37
809	U12	Q	H	CLEARWATER	16-Oct-63	544.98	200	70	TN	BCU	TU	191207.98	271701.35	568378.75	4116931.10	191207.98	271701.66	568378.754116931.42
810	U12	R	H	WINEKIN	15-Jan-69	518.16	200	71	TN	VCU	TU	191567.17	272918.75	568733.57	4118149.44	191566.82	272921.18	568733.214118151.87
811	U12	T.01	H	MINT LEAF	5-May-70	405.38	20	35	TN	BCU	TU	195235.76	273758.4	572398.23	4119001.78	-99.9	-99.9	2120.5
812	U12	T.02	H	DIAMOND SCULL	20-Jul-72	423.98	20	34	TN	BCU	TU	195294.9	273536.63	572458.13	4118780.27	195279.97	273541.2	572443.194118784.79
813	U12	T.03	H	HUSKY POP	24-Oct-75	348	20	36	TN	BCU	TU	195617.75	274326.56	572778.12	4119571.13	-99.9	-99.9	2063.23
814	U12	T.04	H	MIDAS MYTH/MI	15-Feb-84	360.9	20	36	TN	BCU	TU	195493	274297	572653.50	4119541.14	195541.22	274284.3	572701.764119528.61
815	U12	T.08	H	MIGHTY OAK	10-Apr-86	394.4	20	35	TN	BCU	TU	195331.5	273959.4	572493.24	4119203.06	-99.9	-99.9	2110.7
816	U12	T.09	H	MISSION GHOST	20-Jun-87	321.3	20	37	TN	BCU	TU	195787.15	274140.72	572948.13	4119385.93	-99.9	-99.9	2043.91
817	U15	A	H	HARDHAT	15-Feb-62	287.12	5.7	25	KQM	I	QM	206297.38	274901.47	583452.95	4120183.47	-99.9	-99.9	1558.75

818	U15	A.01	H	PILEDRIIVER	2-Jun-66	462.69	62	49	KGD	I	GD	206631.62	274993.5	583786.78	4120276.65	-99.9	-99.9	1551.44
819	U15	E	H	TINYTOT	17-Jun-65	110.95	20	48	KQM	I	QM	206507.57	274587.05	583664.19	4119869.87	-99.9	-99.9	1530.71
820	U16	A	H	MARSEMALLOW	28-Jun-62	310.9	20	37	TN	VCU	TU	193838.34	250732.54	571081.99	4095977.10	-99.9	-99.9	2268.33
821	U16	A.02	H	GUMDROP	21-Apr-65	304.8	20	37	TN	VCU	TU	193747.16	250516.24	570991.59	4095760.54	-99.9	-99.9	1958.34
822	U16	A.03	H	DOUBLE PLAY	15-Jun-66	327.66	20	37	TN	BCU	TU	193670.86	250793.47	570914.34	4096037.43	-99.9	-99.9	1980.9
823	U16	A.04	H	MING VASE	20-Nov-68	307.85	20	37	TN	VCU	TU	193358.68	250811.04	570602.19	4096053.91	-99.9	-99.9	1958.34
824	U16	A.05	H	DIAMOND DUST	12-May-70	270.11	20	39	TN	BCU	TU	193762	250882	571005.15	4096126.26	-99.9	-99.9	1926.65
825	U16	A.06	H	DIAMOND MINE	1-Jul-71	266.15	20	39	TN	BCU	TU	193627.5	251001.49	570870.27	4096245.24	-99.9	-99.9	1923.29
826	U18	A	H	DANNY BOY	5-Mar-62	33.53	0.43	18	QTB	AA	BA	179215.55	261986.98	556423.71	4107177.24	179215.14	261987.09	556423.304107177.35
827	U18	D	H	SULKY	18-Dec-64	27.13	92	114	QTB	AA	BA	181216.38	258848.59	558435.00	4104046.73	181216.02	258848.88	558434.644104047.02
828	U19	A	S	SCOTCH	23-May-67	977.39	155	55	TCF	TC	TU	178712.48	280187.26	555856.78	4125370.73	178712.3	280187.21	555856.604125370.68
829	U19	AA	H	SHEEPSHEAD	26-Sep-79	640	150	61	TX	TC,TCBCUTU		179253.53	275082.7	556415.65	4120269.48	179252.32	275083.77	556414.444120270.55
830	U19	AB	H	TOWANDA	2-May-85	664.5	150	60	TCB	TCBCU	TU	182690.75	277795.28	559842.38	4122993.42	182690.98	277795.66	559842.614122993.80
831	U19	AC	H	TIERRA	15-Dec-84	640	150	61	TCF	TCBCU	RH	184434.94	280904.33	561575.15	4126107.76	184435.72	280903.89	561575.934126107.33
832	U19	AD	H	CHANCELLOR	1-Sep-83	625	143	60	TPE	TC	RH	180038.08	279934.35	557182.91	4125122.56	180038.72	279934.94	557183.554125123.15
833	U19	AE	H	NEBBIOLO	24-Jun-82	640.1	20	61	TPR	TC	RH	178707.73	275874.53	555867.22	4121059.17	178707.87	275873.24	555867.364121057.88
834	U19	AF	H	GALVESTON	4-Sep-86	487.1	20	33	TPE	TC	RH	178920.96	276265.3	556079.01	4121450.59	-99.9	-99.9	2045.21
835	U19	AI	H	SERPA	17-Dec-80	573	150	63	TBD	TBA	TU	183551.46	285720.06	560674.93	4130919.07	183551.38	285720.62	560674.854130919.63
836	U19	AJ	H	HARZER	6-Jun-81	637	150	61	TB	TBA	TU	182636.27	283342.74	559768.37	4128539.17	182636.04	283341.95	559768.154128538.38
837	U19	AK	H	HOSTA	12-Feb-82	640.1	150	61	TBD	TBA	RH	183459.55	288292.67	560573.97	4133490.65	-99.9	-99.9	2102.48
838	U19	AN	H	LABQUARK	30-Sep-86	616	150	61	TCF	TC	RH	184248.31	282980.85	561381.25	4128183.06	184248.18	282980.21	561381.124128182.42
839	U19	AQ	H	LOCKNEY	24-Sep-87	615.7	150	61	TPE	TC	RH	178308.51	274960.74	555471.32	4120144.23	-99.9	-99.9	2072.34
840	U19	AR	H	CYBAR	17-Jul-86	627.9	119	57	TPB	TC	RH	179984.82	280589.99	557127.35	4125777.83	-99.9	-99.9	2044.23
841	U19	AU	H	ALAMO	7-Jul-88	621.8	150	61	TAC	TC	RH	178125.17	277673.66	555278.49	4122855.76	-99.9	-99.9	1991.56
842	U19	AX	H	KEARSARGE	17-Aug-88	615.7	150	61	TCF	TC	RH	184328.35	282656.66	561462.41	4127859.24	-99.9	-99.9	2129.18
843	U19	AY	H	AMARILLO	27-Jun-89	640.1	150	61	TPR	TC	RH	180167.64	280233.99	557311.38	4125422.57	-99.9	-99.9	2045.82
844	U19	AZ	H	HOUSTON	14-Nov-90	594.4	150	62	TPE	TC	TU	178616.36	274897.77	555779.31	4120082.36	-99.9	-99.9	2058.31
845	U19	B	H	HALFBEAK	30-Jun-66	819.3	365	77	TOSQ	BCU	RH	184990.77	284728.61	562117.35	4129932.96	184990.69	284728.89	562117.274129933.24
846	U19	BA	H	BEZAR	4-Apr-91	629.4	150	61	TCPK	TC	LA	183764.49	282534.8	560899.13	4127735.42	183763.81	282534.74	560898.454127735.36
847	U19	BG	H	JUNCTION	26-Mar-92	621.8	150	61	TPR	TC	RH	179617.38	279898.61	556762.45	4125085.35	-99.9	-99.9	2040.33
848	U19	C	H	RICKY	15-Jun-68	683.28	200	66	TX	TC,TCBCUTU		183622.63	279075.47	560769.50	4124276.54	183623.69	279076.06	560770.564124277.13
849	U19	D	H	CHARTREUSE	6-May-66	666.75	73	47	TBD	TBA	RH	182941.97	288292.74	560056.53	4133488.89	182942.19	288293.01	560056.754133489.16
850	U19	E	H	MUNSTER	3-Jan-76	1452.4	1000	93	TBD	TBA	RH	181965.98	282800.68	559100.96	4127774.95	181966.28	282580.46	559101.254127774.74
851	U19	F	H	INLET	20-Nov-75	819	1000	108	TAI	TC	RH	178943.56	274625.53	556107.38	4119811.35	178942.73	274625.17	556106.554119810.98
852	U19	G	H	ESTUARY	9-Mar-76	868.1	500	84	TOSQ	BCU	RH	179209.87	284059.88	556340.39	4129244.03	179210.09	284059.91	556340.614129244.06
853	U19	I	H	SLED	29-Aug-68	728.88	200	65	TX	TC,TCBCUTU		180768.69	277446.35	557922.08	4122637.81	180767.84	277445.34	557921.234122636.80
854	U19	L	H	STINGER	22-Mar-68	667.76	200	66	TOSQ	BCU	RH	183947.5	286588.2	561067.80	4131788.37	183947.61	286588.24	561067.914131788.41
855	U19	N	H	SCROLL	23-Apr-68	224.03	20	40	TAC	TC	TU	178194.05	287135.66	555314.00	4132315.38	178194.54	287135.1	555314.494132314.82
856	U19	P	H	POOL	17-Mar-76	879.3	500	84	TCB	TCBCU	TU	182390.87	278069.96	559541.62	4123266.97	-99.9	-99.9	2102.7
857	U19	Q	H	CAMENBERT	26-Jun-75	1310.6	1000	96	TCB	TCBCU	TU	178833.91	280614.56	555976.67	4125798.34	178832.25	280614.49	555975.014125798.26
858	U19	T	H	EMMENTHAL	2-Nov-78	576.1	20	32	TCF	VCU	RH	185134.13	281637.57	562271.56	4126843.27	185133.67	281638.45	562271.104126844.14
859	U19	U	H	MAST	19-Jun-75	911.3	1000	105	TBD	TBA	RH	183093.8	288554.84	560207.40	4133751.46	183094.4	288556.35	560207.994133752.97
860	U19	V	H	ALMENDRO	6-Jun-73	1063.75	1000	101	TBD	TBAQ	RH	180847.74	276863.09	558003.16	4122054.99	180849.05	276862.8	558004.474122054.71
861	U19	X	H	BACKBEACH	11-Apr-78	672.1	150	60	TP	TC	RH	178860	275572.52	556020.51	4120757.78	178860.1	275572.78	556020.614120758.04
862	U19	Y	S	PANIR	31-Aug-78	681	150	60	TPR	TC	RH	179832.21	280285	556975.86	4125472.39	179832.12	280285.33	556975.774125472.72
863	U19	Z	S	FONDUTTA	11-Apr-78	633	150	61	TB	TBA	TU	182539.82	282924.1	559673.43	4128120.30	182539.61	282924.17	559673.214128120.37
864	U20	A	H	BUREO	12-May-65	695.55	20	30	TAC	TC	TU	173322	276574.69	550480.52	4121740.18	173321.37	276576.84	550479.894121742.33
865	U20	A	1	DURYEA	14-Apr-66	543.76	70	49	TAC	TC	RH	173322	276574.7	550480.52	4121740.19	-99.9	-99.9	1987.21
866	U20	AA	H	COLBY	14-Mar-76	1273.4	1000	96	TAC	TC	TU	169702.66	283594.46	546837.48	4128745.23	169703.02	283594.92	546837.834128745.69
867	U20	AB	H	FARM	16-Dec-78	689	150	60	TAC	TC	TU	175138.37	279989.66	552284.36	4125160.59	175138.81	279989.67	552284.804125160.61
868	U20	AC	H	COLWICK	26-Apr-80	633	150	61	TAC	TC	RH	174070.31	277216.49	551226.37	4122384.44	-99.9	-99.9	1972.95
869	U20	AD	H	PEPATO	11-Jun-79	681	150	60	TAC	TC	TU	171145.53	281788.23	548286.30	4126944.59	-99.9	-99.9	1940.48
870	U20	AE	H	TAPI	25-Jul-80	680	150	60	TX	TC,TCBCUTU		169189.01	278081.49	546343.39	4123232.01	169189.36	278081.64	546343.744123232.16
871	U20	AF	H	KASH	12-Jun-80	645	150	61	TAC	TC	RH	171271.75	280897.18	548415.63	4126054.24	171272.3	280896.67	548416.184126053.73
872	U20	AG	H	MOLBO	12-Feb-82	638	150	61	TPB	TC	RH	170505.43	274533.93	547671.93	4119690.08	170506.33	274534.75	547672.824119690.91
873	U20	AH	H	GIBNE	25-Apr-82	570	150	63	TX	TC,TCBCUTU		174071.76	278038.95	551224.92	4123206.67	174070.83	278038.19	551223.994123205.91
874	U20	AI	H	JEFFERSON	22-Apr-86	609	150	62	TAC	TC	RH	172486.81	278953.58	549637.19	4124115.46	172486.52	278953.28	549636.904124115.16
875	U20	AJ	H	CABRA	26-Mar-83	542.5	150	63	TAC	TC	RH	170718.79	283007.51	547855.39	41228162.02	170718.97	283007.38	547855.574128161.89
876	U20	AK	H	SALUT	12-Jun-85	608	150	62	TPB	TC	RH	168157.17	277139.67	545315.16	4122866.82	-99.9	-99.9	1900.49
877	U20	AL	H	EGMONT	9-Dec-84	546	150	63	TX	TC,TCBCUTU		167396.42	279604.13	544545.95	4124747.90	167396.61	279604.21	544546.144124747.98
878	U20	AM	H	KAPPELLI	25-Jul-84	640	150	61	TAC	TC	RH	175107.74	279365.15	552555.94	4124536.15	175107.41	279365.94	552555.614124535.94
879	U20	AN	H	SERENA	25-Jul-85	597	150	62	TAC	TC	RH	172666.9	282630.43	549804.28	4127791.92	-99.9	-99.9	1969.5
880	U20	AO	H	GOLDSTONE	28-Dec-85	549	150	63	TPB	TC	RH	169606.34	276027.49	546767.84	4121180.06	-99.9	-99.9	1913.83

881	U20	AP	H	BODIE	13-Dec-86	635	150	61	TAC	TC	TU	175016.72	278831.46	552166.83	4124002.29	175016.34	278831.22	552166.45	4124002.05	2018.09
882	U20	AQ	H	DARWIN	25-Jun-86	549	150	63	TX	TC,TCBCUTU		167244.25	278994.64	544395.97	4124138.05	167244	278994.73	544395.72	4124138.14	1875.93
883	U20	AR	H	KERNVILLE	15-Feb-88	541.6	150	63	TX	TC,TCBCUTU		169700.67	284531.37	546832.18	4129681.86	169700.68	284531.44	546832.19	4129681.93	1926.03
884	U20	AS	H	BELMONT	16-Oct-86	605	150	62	TPB	TC	TU	170596.63	274077.14	547764.71	4119233.75	-99.9	-99.9	-99.9	-99.9	1897.98
885	U20	AT	H	DELAMAR	18-Apr-87	544.1	150	63	TX	TC,TCBCUTU		166375.32	277139.99	543533.82	4122280.87	-99.9	-99.9	-99.9	-99.9	1902.05
886	U20	AV	H	HARDIN	30-Apr-87	625	150	61	TX	TC,TCBCUTU		174010.8	275509.36	551172.88	4120677.58	-99.9	-99.9	-99.9	-99.9	1970.14
887	U20	AW	H	CONTACT	22-Jun-89	544.1	150	63	TAC	TC	RH	174955.51	281041.33	552097.85	4126211.33	174955.87	281041.4	552098.21	4126211.40	2007.11
888	U20	AY	H	COMSTOCK	2-Jun-88	620.3	150	61	TAC	TC	RH	172410.47	278511.56	549562.43	4123673.30	-99.9	-99.9	-99.9	-99.9	1987.45
889	U20	AZ	H	BARNWELL	8-Dec-89	600.8	150	62	TAC	TC	TU	175229.87	275295.91	552392.36	4120468.47	-99.9	-99.9	-99.9	-99.9	2058.27
890	U20	B	H	PIPKIN	8-Oct-69	623.62	1000	115	TAC	TC	RH	172441.18	278132.07	549594.47	4123294.02	172441.11	278132.58	549594.40	4123294.53	1991.7
891	U20	BB	H	TENABO	12-Oct-90	600	150	62	TM	TMA	TU	167699.87	277139.67	544857.99	4122285.21	-99.9	-99.9	-99.9	-99.9	1897.77
892	U20	BC	H	HORNITOS	31-Oct-89	563.9	150	63	TX	TC,TCBCUTU		168006.23	278831.53	545158.31	4123977.67	-99.9	-99.9	-99.9	-99.9	1873.3
893	U20	BD	H	BULLION	13-Jun-90	673.9	150	60	TAC	TC	TU	174269.44	278679.23	551420.29	4123847.47	-99.9	-99.9	-99.9	-99.9	1976.93
894	U20	BE	H	HOYA	14-Sep-91	658	150	60	TAC	TC	TU	173568.31	274686.18	550733.41	4119853.07	-99.9	-99.9	-99.9	-99.9	1978.64
895	U20	BF	H	MONTELLLO	16-Apr-91	641.6	150	61	TAC	TC	TU	172364.81	276880.69	549522.53	4122042.73	-99.9	-99.9	-99.9	-99.9	1987.99
896	U20	C	H	BENHAM	19-Dec-68	1402.08	1150	99	TAC	TC	TU	169534.53	275325.13	546698.52	4120477.64	169533.38	275324.47	546697.37	4120476.98	1914.54
897	U20	D	H	KNICKERBOCKER	26-May-67	630.63	76	49	TPB	TC	RH	168944.77	277151.06	546102.50	4122300.98	168944.5	277151.02	546102.23	4122300.94	1905.65
898	U20	E	H	JORUM	16-Sep-69	1160.89	1000	99	TAC	TC	TU	170657.93	284501.04	547789.28	4129654.92	170658.83	284501.8	547790.18	4129655.68	1925.21
899	U20	F	H	FONTINA	12-Feb-76	1219	1000	98	TAC	TC	TU	168206.38	279753.65	545355.15	4124900.23	168207.12	279753.94	545355.89	4124900.53	1864.37
900	U20	G	H	GREBLEY	20-Dec-66	1216.46	870	93	TCB	TC	TU	175305.5	283172.98	552440.23	4128343.61	175305.68	283173.16	552440.41	4128343.80	1972.02
901	U20	H	H E	REX	24-Feb-66	671.17	20	31	TAC	TC	TU	173049.37	279811.44	550196.57	4124975.07	173077.41	279808.32	550224.62	4124972.04	1998.58
902	U20	I	H	BOXCAR	26-Apr-68	1165.86	1300	108	PSC		RH	171104.48	282424.95	548243.02	4127580.99	171104.28	282425.61	548242.82	4127581.65	1941.63
903	U20	K	H	PALANQUIN	14-Apr-65	85.65	4.3	31	TTC	TMA	TR	165091.14	280743.51	542237.32	4125878.84	165091.64	280744.27	542237.81	4125879.60	1887.88
904	U20	L	H	CABRIOLET	26-Jan-68	51.82	2.3	28	TTC	TMA	TR	165898.65	280797.49	543044.40	4125935.65	165898.71	280798.08	543044.46	4125936.24	1888.85
905	U20	M	H	HANDLEY	26-Mar-70	1209	1000	98	TC	TC	TU	164150.96	282972.98	541289.55	4128104.35	164149.4	282972.68	541287.99	4128104.05	1799.25
906	U20	N	H	CHESHIRE	14-Feb-76	1167	500	78	TAC	TC	RH	174266.1	276574.19	551424.36	4121743.01	174266	276573.25	551424.27	4121742.06	1974.08
907	U20	P	H	STILTON	3-Jun-75	731.5	200	65	TOSQ	BCU	RH	165140.96	287366.41	542263.77	4132500.02	165140.96	287366.1	542263.77	4132499.71	1694.54
908	U20	T	H	CHATEAUGAY	28-Jun-68	607.23	200	68	TPB	TC	RH	168707.47	276880.79	545866.22	4122029.95	168707.89	276880.58	545866.64	4122029.74	1903.59
909	U20	U	H	SCHOONER	8-Dec-68	111.25	30	55	TBG	TBA	TU	161330.96	287734.63	538453.57	4132854.69	161330.96	287734.94	538453.57	4132855.00	1695.42
910	U20	V	H	PURSE	7-May-69	598.78	200	68	TX	TC,TCBCUTU		167122.28	281026.27	544266.88	4126168.67	167123.64	281026.48	544268.24	4126168.89	1855.58
911	U20	Y	H	TYBO	14-May-75	765	1000	110	TPT	VA	TU	169483.11	274138.18	546651.29	4119290.85	169481.51	274137.65	546649.69	4119290.31	1907.05
912	U20	Z	H	KASSERI	28-Oct-75	1265	1000	97	TCB	TCBCU	TU	175020.72	281837.81	552160.23	4127007.81	175020.83	281838.69	552160.34	4127008.69	1984
913	U30	A	H	BUGGY	12-Mar-68	41.15	1.08	23	TDM		BA	178805.47	250493.75	556054.03	4095685.75	-99.9	-99.9	-99.9	-99.9	1587.43
914	U30	B	H	BUGGY	12-Mar-68	41.15	1.08	23	TDM		BA	178762.68	250509.9	556011.19	4095701.74	-99.9	-99.9	-99.9	-99.9	1588.04
915	U30	C	H	BUGGY	12-Mar-68	41.15	1.08	23	TDM		BA	178719.86	250526.02	555968.33	4095717.71	-99.9	-99.9	-99.9	-99.9	1588.76
916	U30	D	H	BUGGY	12-Mar-68	41.15	1.08	23	TDM		BA	178677.13	250542.12	555925.55	4095733.65	-99.9	-99.9	-99.9	-99.9	1587.92
917	U30	E	H	BUGGY	12-Mar-68	41.15	1.08	23	TDM		BA	178634.35	250558.22	555882.73	4095749.60	-99.9	-99.9	-99.9	-99.9	1587.43

Appendix C

Transport Programs

Appendix C contains a listing of computer codes used in the analysis. The relationship of the codes to the data flow and final output is summarized in the process diagram also included in this appendix. The listing of the MC_TRANS code is not included here, but is part of [Appendix A](#). The codes are listed in the order of usage.

CODE NAME

GWSHOT02

MODP

PATHHSU

VELOC02

TIMEDIST

PSUMRY02

MCCSTNTS

QSENS02

SENSVEC

PRELHS

LHSMC

READBIN4

BTCHIST

CREATION OF PATHLINE FILES

CALCULATION OF VELOCITY ALONG PATHLINES

CALCULATE TRAVEL TIMES AND SUMMARIZE

PREPARE INPUT FILES FOR LATIN HYPERCUBE SAMPLING AND TRANSPORT MODELING

PREPARE FLOW SENSIVITY FILES FOR THE TRANSPORT MODEL

USE LATIN HYPERCUBE SAMPLING TO CREATE INPUT PARAMETER VALUES

```

C          ***** GWSHOT02.FOR *****
C WRITTEN BY K. R. REHFELDT, (KRR), 10/15/95, GeoTrans, Inc.
c modified 3/13/96 to use a new shots data base
c modified 10/21/96 to accomodate the new flow model
C PROGRAM TO COMPARE THE HEADS AT EACH SHOT WITH THE ELEVATION
C OF THE BOTTOM OF THE SHOT CAVITY TO SEE IF THE SHOT MIGHT IMPACT
C THE GROUNDWATER. IF A SHOT IS WITHIN 2 CAVITY RADII, IT IS USED.
C THE FOLLOWING ARE CALCULATED
C   1. ELEVATION OF THE BOTTOM OF TWICE THE CAVITY RADIUS
C   2. DIFFERENCE BETWEEN 1. AND THE CALCULATED GROUNDWATER LEVEL
C   3. A FILE WITH INFORMATION ABOUT SHOTS IN THE UNSATURATED ZONE
C      ARE ALSO GENERATED FOR FUTURE USE
C
C KEY DATA SET INPUTS
C   LOCATION INFORMATION FOR EACH SHOT          NTSSHOTS.DAT
C   MODEL GRID INFORMATION                      M_ELEV.DAT
C   MODEL GRID INFORMATION                      GRIDS.DAT
C KEY OUTPUT
C   LOCATION INFORMATION FOR SHOTS NEAR WATER TABLE SHOTSWT.DAT
C   PATHLINE STARTING LOCATIONS                 PATHS.PTS
C   PATHLINES BEGINNNING AT THE WATER TABLE
C       FOR UNSATURATED ZONE SHOTS              PATHSWT.PTS
C
C   implicit double precision (a-h,o-z)
C   character*4 d1
C   character*50 cvar
C   character*48 v1
C   character*8 v2
C   character*14 sname
C   character*2 sloc
C   character*45 v3
C   character*21 v4
C   real*8 x,y,xp,yp,xs,ys
C   dimension delx(68),dely(76),ibound(68,76,20),k_coord(3),
&   elev(21),hobs(3),zlocal(3)
C   open (unit=23,file='ntsshots.dat')
C   open (unit=24,file='ibound.rst')
C   open (unit=25,file='shotswt.dat')
C   open (unit=26,file='m_elev.dat')
C   open (unit=27,file='grids.dat')
C   open (unit=28,file='paths.pts')
C   open (unit=29,file='debug.out')
C   open (unit=30,file='numpaths.dat')
C   open (unit=31,file='partid.dat')
C   open (unit=32,file='pathswt.pts')
C   open (unit=33,file='partidu.dat')
C   np = 0
C   npu = 0
C   read (27,1000) d1
C   read (27,*) alfa,nrow,ncol,x0ul,y0ul,nlay
c   read grid spacings
C   rad = alfa/57.295779
C   read (27,*) (delx(j),j=1,ncol)
C   write (6,*) ' read delx'
C   read (27,*) (dely(i),i=1,nrow)
C   write (6,*) ' read dely'
C   close (27)
C   ytot = 0.0
C   do 878 i=1,nrow
C   ytot = ytot + dely(i)
878   continue
C   y0 = y0ul - ytot*cos(rad)
C   x0 = x0ul + ytot*sin(rad)
C   write (29,*) x0,y0
c   read model layer information
C   read (26,1003) nmlay

```



```

        read (26,1004) (elev(k),k=1,nmlay+1)
        close (26)
        write (6,*) ' read elevation data'
c      read ibound from the last model run
        do 10 k=1,nlay
        do 11 I=1,nrow
            read (24,1002) (ibound(j,i,k),j=1,ncol)
11          continue
10        continue
        write (6,*) ' ibound read'
c      call hread (dh,ncol,nrow,nlay)
c      sort the heads to put the largest value into an array dhm
c      do 30 i=1,nrow
c      do 31 j=1,ncol
c      dhm(j,i) = 0.0
c      do 32 k=1,nlay
c      if (dh(j,i,k).gt.dhm(j,i).and.dh(j,i,k).lt.2000.)
c      & dhm(j,i) = dh(j,i,k)
c32    continue
c31    continue
c30    continue
c      read the shot information
        read (23,1000) d1
        read (23,1000) d1
        do 20 nk=1,1000
            read (23,1006,end=601) v1,dob,v2,radius,head,v3,x,y,
            & v4,xp,yp,elevls
            head = head * 0.3048
            write (6,*) ' reading shot number ',nk
            write (cvar,'(a48)') v1
            read (cvar,'(25x,a14)') sname
            if (x.lt.1.or.elevls.lt.0.) go to 20
            if (xp.lt.1.) then
                xs = x
                ys = y
            else
                xs = xp
                ys = yp
            end if
            hobs(1) = elevls - dob - 2.0*radius
            hobs(2) = elevls - dob
            hobs(3) = elevls - dob + 2.0*radius
            xs = xs - x0
            ys = ys - y0
            xs1 = xs*cos(rad) + ys*sin(rad)
            ys1 = ys*cos(rad) - xs*sin(rad)
            sumx = 0
            sumy = 0
            i_coord = 0
            j_coord = 0
            do 37 m=1,3
                k_coord(m) = 0
37          continue
            do 40 j=1,ncol
                sumx = sumx + delx(j)
                if(xs1.le.sumx) then
                    coff = ((xs1-(sumx-delx(j)))/delx(j))-0.5
                    j_coord = j
                    xlocal = coff + 0.5
                    go to 41
                end if
40          continue
41          do 52 i=1,nrow
                sumy = sumy + dely(nrow-i+1)
                if(ys1.le.sumy) then
                    roff = (((ys1-(sumy-dely(nrow-i+1)))/dely(nrow-i+1))-0.5)*(-1.)

```

```

        i_coord = nrow-i+1
        ylocal = roff*(-1.) + 0.5
        go to 53
    endif
52     continue
53     do 57 m=1,3
        do 55 k=1,nlay
            if (hobs(m).gt.elev(k)) then
                k_coord(m) = k-1
                zlocal(m) = (hobs(m)-elev(k))/(elev(k-1) - elev(k))
                go to 57
            end if
        55     continue
        57     continue
        do 59 n=1,3
            c      write (6,*) ' call to SSEN1I, nk = ',nk
            c      CALL SSEN1I (k_coord(n),i_coord, j_coord,COFF,ROFF,DELY,DELX,
            c      1      IBOUND,NCOL,NROW,NLAY,RINT,JOFF,IOFF,nlay)
            c      jo = joff
            c      io = ioff
            ii = i_coord
            jj = j_coord
            kk = k_coord(n)
            if (ibound(jj,ii,kk).le.0) go to 58
            if (n-2) 62,63,64
            62     sloc = ' B'
            go to 83
            63     sloc = 'WP'
            go to 83
            64     sloc = ' T'
c65     if (ii+io.le.0.or.ii+io.ge.ncol.or.jj+jo.le.0.or.
c      & jj+jo.ge.nrow) then
c      h = dhm(jj,ii)
c      go to 83
c      end if
c      interpolate the calculated heads to a point in space that
c      corresponds with the measured well
c      H = ( RINT(1)*dhm(jj,ii) +
c      1      RINT(2)*dhm(jj+jo,ii) + RINT(3)*dhm(jj,ii+io)
c      1      + RINT(4)*dhm(jj+jo,ii+io) )
            83     continue
            c      if (h.gt.3000) h = dhm(jj,ii)
            c      if (h.gt.3000) go to 59
            write (29,1008) v1,dob,hobs(n),head
            resid = head - hobs(n)
            if (resid.le.0.) go to 58
            if (resid.gt.0.) then
                if (n.eq.1) write(25,1009) v1,dob,v2,radius,head,v3,x,y,v4,
                & xp,yp,elevls
                write(28,1007) j_coord,i_coord,k_coord(n),xlocal,ylocal,
                & zlocal(n),sname,sloc
                np = np + 1
                write (31,1110) np,sname,sloc
            end if
            go to 59
        58     if (n.gt.1) go to 59
            do 74 k=1,nlay
                if (ibound(jj,ii,k).gt.0) then
                    kk = k
                    go to 68
                end if
            74     continue
        68     npu = npu+1
            write (33,1110) npu,sname,'WT'
            zlocun = 1
            write (32,1007) jj,ii,kk,xlocal,ylocal,zlocun,sname,'WT'

```

```

59    continue
20    continue
1000  format (a4)
1001  format (8(i5,f5.3))
1002  format (40i2)
1003  format (4i5)
1004  format (f8.2)
1005  format (14f4.0)
1006  format (a48,f8.0,a8,f8.0,f5.0,a46,f11.0,f11.0,a21,f11.0,f11.0,
&      f9.0)
1007  format (3i5,3f10.4,10x,a14,2x,a2)
1008  format(a50,2(1x,f8.2),2(1x,f12.2))
1009  format (a48,f8.2,a8,f9.0,f5.0,a46,f11.2,2x,f11.2,a21,2f13.2,
&      f9.2)
1110  format (i5,1x,a14,2x,a2)
601  write (30,1003) np
      CLOSE (23)
      CLOSE (24)
      CLOSE (25)
      CLOSE (26)
      CLOSE (27)
      CLOSE (28)
      close (29)
      close (30)
      close (31)
      STOP
      END
C ----- HREAD
C      Function: To read head/concentration data from model output file
C      SUBROUTINE HREAD (dh,ncol,nrow,nlay)
C
C      CHARACTER*4 TEXT
C      CHARACTER*12 FILLER
C      real*4 tlft,t2ft
C      dimension dh(ncol,nrow,nlay)
C      Set return status (0:normal, -1:error).
C      open (7,file='base.hds',form='unformatted')
C----- UNFORMATTED READ
C      do 6101 k=1,nlay
C      READ (7) LFT, KPERFT, TLFT, T2FT,
C      1      FILLER, TEXT, NXFT, NYFT, KFT
C      READ (7) ((DH(I,J,k), I=1, NCOL), J=1, NROW)
c6101  continue
C      close(7)
C      RETURN
C      end
C=====
C      SUBROUTINE SSEN1I(ider1,ider2,
C      1      ider3,COFF,ROFF,DELR,DELC,IBOUND,
C      2      NCOL,NROW,NLAY,RINT,JOFF,IOFF,MLAY)
C-----VERSION 1000 01FEB1992
C      *****
C      CALCULATE INTERPOLATION COEFFICIENTS FOR LOCATING OBSERVED HEADS.
C      *****
C      SPECIFICATIONS:
C      -----
C      DIMENSION NDER(3),DELR(NCOL),DELC(NROW),IBOUND(NCOL,NROW,NLAY),
C      1      RINT(4)
C      -----
C
C      nder(1) = ider1
C      nder(2) = ider2
C      nder(3) = ider3
C      K=NDER(1)
C      IF(K.LT.0) K=MLAY
C      I=NDER(2)

```

```

C      J=NDER(3)
C      I1=I+1
C      J1=J+1
C      IOFF=1
C      JOFF=1
C      IF(ROFF.LT.0.) THEN
C          I1=I-1
C          IOFF=-1
C      ENDIF
C      IF(COFF.LT.0.) THEN
C          J1=J-1
C          JOFF=-1
C      ENDIF
C      IF(I1.GE.1.AND.I1.LE.NROW) IBI=IBOUND(j,i1,K)
C      IF(J1.GE.1.AND.J1.LE.NCOL) IBJ=IBOUND(j1,i,K)
C      IF(I1.GE.1.AND.I1.LE.NROW.AND.J1.GE.1.AND.J1.LE.NCOL)
C  A      IBIJ=IBOUND(j1,i1,K)
C      IF(I1.LT.1.OR.I1.GT.NROW) THEN
C          ROFF=0.
C          IBI=0
C      ENDIF
C      IF(J1.LT.1.OR.J1.GT.NCOL) THEN
C          COFF=0.
C          IBJ=0
C      ENDIF
C      IF(I1.LT.1.OR.I1.GT.NROW.OR.J1.LT.1.OR.J1.GT.NCOL) IBIJ=0
C
C      IF((ABS(ROFF).LT..001.AND.ABS(COFF).LT..001).OR.
C  1      (ABS(ROFF).LT..001.AND.IBJ.EQ.0).OR.
C  1      (ABS(COFF).LT..001.AND.IBI.EQ.0).OR.
C  1      (IBI.EQ.0.AND.IBJ.EQ.0)) THEN
C          IOFF=0
C          JOFF=0
C          DO 100 IR=1,4
C  100      RINT(IR)=.25
C          RETURN
C      ENDIF
C
C---CALCULATE CONSTANTS
C      IF(ABS(ROFF).GT..001) THEN
C          DC=(DELC(I)+DELC(I1))/2.
C          DCF=ABS(ROFF)*DELC(I)
C      ENDIF
C      IF(ABS(COFF).GT..001) THEN
C          DR=(DELR(J)+DELR(J1))/2.
C          DRF=ABS(COFF)*DELR(J)
C      ENDIF
C      IF(ABS(ROFF).GT..001.AND.ABS(COFF).GT..001) A=1/(DC*DR)
C
C---LINEAR INTERPOLATION
C      IF(ABS(ROFF).LT..001.OR.
C  1      (IBI.EQ.0.AND.IBIJ.EQ.0)) THEN
C          IOFF=0
C          RINT(1)=0.5*(1.-DRF/DR)
C          RINT(2)=0.5*DRF/DR
C          RINT(3)=RINT(1)
C          RINT(4)=RINT(2)
C          RETURN
C      ENDIF
C
C      IF(ABS(COFF).LT..001.OR.
C  1      (IBJ.EQ.0.AND.IBIJ.EQ.0)) THEN
C          JOFF=0
C          RINT(1)=0.5*(1.-DCF/DC)
C          RINT(2)=RINT(1)
C          RINT(3)=0.5*DCF/DC

```

```

C      RINT(4)=RINT(3)
C      RETURN
C  ENDIF
C
C---CALCULATE BASIS FUNCTIONS FOR INTERPOLATION ON A RECTANGLE
C  IF(IBJ.NE.0.AND.IBI.NE.0.AND.
C  1  IBIJ.NE.0) THEN
C      RINT(3)=A*(DR-DRF)*DCF
C      RINT(4)=A*DRF*DCF
C      RINT(2)=A*DRF*(DC-DCF)
C      RINT(1)=A*(DR-DRF)*(DC-DCF)
C      RETURN
C  ENDIF
C
C---CALCULATE BASIS FUNCTIONS FOR INTERPOLATION ON A TRIANGLE
C  IF(IBJ.EQ.0) THEN
C      RINT(1)=A*(DR*DC - DR*DCF)
C      RINT(2)=0.
C      RINT(3)=A*(DR*DCF - DC*DRF)
C      RINT(4)=A*(DC*DRF)
C      RETURN
C  ENDIF
C
C  IF(IBI.EQ.0) THEN
C      RINT(1)=A*(DR*DC - DC*DRF)
C      RINT(4)=A*(DR*DCF)
C      RINT(2)=A*(DC*DRF - DR*DCF)
C      RINT(3)=0.
C      RETURN
C  ENDIF
C
C  IF(IBIJ.EQ.0) THEN
C      RINT(1)=A*(DR*DC - DC*DRF - DR*DCF)
C      RINT(3)=A*(DR*DCF)
C      RINT(2)=A*(DC*DRF)
C      RINT(4)=0.
C      RETURN
C  ENDIF
C  END
C

```

```

C---Version 1.2   April 27, 1990                                     # A00020
C*****                                                    # A00030
C                      MODPATH                                     # A00040
C                                                                # A00050
C  A three-dimensional particle tracking post-processing program for # A00060
C    the USGS modular three-dimensional ground-water flow model    # A00070
C                                                                # A00080
C                      by                                         # A00090
C                                                                # A00100
C                      David W. Pollock                           # A00110
C                                                                # A00120
C*****                                                    # A00130
C                                                                # A00140
C----- REDIMENSION MASTER ARRAY BY CHANGING THE NEXT TWO STATEMENTS ----# A00150
C----- MAKE SURE THAT LENA IS SET EQUAL TO THE LENGTH OF THE A ARRAY ----# A00160

c  A array and LENA reset to 750,000 from 60,000 by IGMWC 1/2/91

      COMMON A(1150000)                                           # A00170
      LENA=1150000                                              # A00180
C-----                                                    # A00190
C                                                                # A00200
C  SET UNIT NUMBERS FOR SCREEN I/O AND FILES THAT ARE SPECIFIED BY # A00210
C  INTERACTIVE I/O                                              # A00220
C                                                                # A00230
C  I0 IS THE UNIT NUMBER OF FILE CONTAINING FILE NAMES AND UNIT NUMBERS # A00240
C    I0=101                                                    # A00250
C  I2 IS THE UNIT NUMBER FOR THE "PATHLINE" FILE                # A00260
C    I2=102                                                    # A00270
C  I3 IS THE UNIT NUMBER FOR THE "ENDPOINT" FILE                 # A00280
C    I3=103                                                    # A00290
C  I4 IS THE UNIT NUMBER FOR THE "TIMESERS" FILE                 # A00300
C    I4=104                                                    # A00310
C  I5 IS THE UNIT NUMBER FOR A SCRATCH FILE                      # A00320
C    I5=105                                                    # A00330
C  I6 IS THE UNIT NUMBER FOR THE STARTING LOCATIONS FILE         # A00340
C    I6=106                                                    # A00350
C  I7 IS THE UNIT NUMBER FOR THE "SUMMARY.PTH" FILE              # A00360
C    I7=107                                                    # A00370
C  I8 IS THE UNIT NUMBER FOR THE TIME STEP FILE                  # A00380
C    I8=108                                                    # A00390
C  I9 IS THE UNIT NUMBER FOR THE FILE TO WHICH SCREEN INPUT IS   # A00400
C    ECHOED IF THE RUN IS INTERACTIVE (IBATCH=0). IF THE RUN IS # A00410
C    BATCH (IBATCH NOT EQUAL TO 0), "SCREEN" INPUT IS READ FROM  # A00420
C    A FILE OPENED TO UNIT I9                                    # A00430
C                                                                # A00440
C    I9=109                                                    # A00450
C                                                                # A00460
C                                                                # A00470
C  "IBATCH" IS A FLAG INDICATING WHETHER THE RUN HAS INTERACTIVE # A00480
C  SCREEN INPUT OR IS "BATCH"                                    # A00490
C                                                                # A00500
C  IBATCH = 0 --> INPUT AT SCREEN                                # A00510
C  IBATCH NOT EQUAL TO 0 --> ALL INPUT READ FROM FILES           # A00520
C                                                                # A00530
C  THE VALUE OF IBATCH IS SET BY A CALL TO SUBROUTINE IOTYPE WHICH # A00540
C  OPENS A FILE NAMED "TERMIN.PTH". IF TERMIN.PTH DOES NOT EXIST, IT IS # A00550
C  CREATED BY IOTYPE. IOTYPE READS THE FIRST LINE OF TERMIN.PTH. IBATCH # A00560
C  IS SET TO 0 IF THE FIRST LINE CONTAINS "SCREEN". IT IS SET TO 1 IF # A00570
C  THE FIRST LINE CONTAINS "BATCH". IF THE FILE HAS JUST BEEN CREATED, # A00580
C  IBATCH IS SET TO 0 AND THE WORD "SCREEN" IS WRITTEN IN THE FIRST # A00590
C  LINE OF THE FILE.                                           # A00600
C                                                                # A00610
C    CALL IOTYPE(IBATCH,I9,I7)                                   # A00620
C                                                                # A00630
C    IF (IBATCH.EQ.0) THEN                                       # A00640

```

```

        WRITE(*,*) ' -----' # A00650
        WRITE(*,*) ' MODPATH -- Version 1.2 -- April 27, 1990 ' # A00660
        WRITE(*,*) ' -----' # A00670
        END IF # A00680
C----- # A00690
C # A00700
C OPEN THE DATA FILES CONTAINING THE BASIC INFORMATION ABOUT THE FLOW # A00710
C SYSTEM. ALSO OPEN STANDARD OUTPUT AND SCRATCH FILES. # A00720
C # A00730
C     CALL FILES (I0,I1,I3,I5,I7,I9,IBATCH) # A00740
C # A00750
C WRITE VERSION NUMBER AND DATE TO FILE SUMMARY.PTH # A00760
C # A00770
C     WRITE(I7,*) ' ' # A00780
C     WRITE(I7,*) 'MODPATH -- Version 1.2 -- April 27, 1990 ' # A00790
C     WRITE(I7,*) ' ' # A00800
C # A00810
C ALLOCATE SPACE FOR ARRAYS # A00820
C # A00830
C     CALL SPACE(LENA,NCOL,NROW,NLAY,NCBL,IGRID,NLPOR,NZDIM,NUNIT,LCQX, # A00840
C     1LCQY,LCQZ,LCPOR,LCIBOU,LCXMAX,LCDX,LCYMAX,LCDY,LCZBOT,LCZTOP, # A00850
C     2LCDZ,LCHAD,LCBUFF,LCLAYC,LCNCON,LCDZCB,LCIBUF,LCIUN,LCXLC,LCQSS, # A00860
C     3LCYLC,LCZLC,LCZLL,LCTOT,LCJLC,LCILC,LCKLC,NPART,NCP1,NRP1,NLP1, # A00870
C     4I1,I7,IBATCH) # A00880
C # A00890
C EXECUTE TRACKING PROGRAM # A00900
C # A00910
C     CALL DRIVER (NCOL,NROW,NLAY,NCBL,IGRID,NLPOR,NZDIM,A(LCQX), # A00920
C     1A(LCQY),A(LCQZ),A(LCPOR),A(LCIBOU),A(LCXMAX),A(LCDX),A(LCYMAX), # A00930
C     2A(LCDY),A(LCZBOT),A(LCZTOP),A(LCDZ),A(LCHAD),A(LCBUFF), # A00940
C     3A(LCLAYC),A(LCNCON),A(LCDZCB),A(LCIBUF),A(LCIUN),A(LCXLC), # A00950
C     4A(LCYLC),A(LCZLC),A(LCZLL),A(LCTOT),A(LCJLC),A(LCILC),A(LCKLC), # A00960
C     5A(LCQSS),NPART,NCP1,NRP1,NLP1,NUNIT,MODE,I1,I2,I3,I4, # A00970
C     6I5,I6,I7,I8,I9,IBATCH) # A00980
C # A00990
C     STOP # A01000
C     END # A01010
C-----END OF ROUTINE----- # A01020

C # B00010
C---Version 1.2 April 27, 1990 # B00020
C***** # B00030
C # B00040
C     DRIVER # B00050
C # B00060
C THIS IS THE MAIN SUBROUTINE THAT CONTROLS THE OVERALL SEQUENCE OF # B00070
C COMPUTATIONS FOR THE PARTICLE TRACKING ANALYSIS. # B00080
C # B00090
C***** # B00100
C # B00110
C     SUBROUTINE DRIVER (NCOL,NROW,NLAY,NCBL,IGRID,NLPOR,NZDIM,QX, # B00120
C     1QY,QZ,POR,IBOUND,XMAX,DELR,YMAX,DELC,ZBOT,ZTOP,DELZ,HEAD,BUFF, # B00130
C     2LAYCON,NCON,DELZCB,IBUFF,IUNIT,XLC,YLC,ZLC,ZLLC,TOT,JLC,ILC,KLC, # B00140
C     3QSS,NPART,NCP1,NRP1,NLP1,NUNIT,MODE,I1,I2,I3,I4,I5, # B00150
C     4I6,I7,I8,I9,IBATCH) # B00160
C # B00170
C     DIMENSION TOT(NPART),XLC(NPART),YLC(NPART),ZLC(NPART), # B00180
C     1ZLLC(NPART),JLC(NPART),ILC(NPART),KLC(NPART),QX(NCP1,NROW,NLAY), # B00190
C     2QY(NCOL,NRP1,NLAY),QZ(NCOL,NROW,NLP1),POR(NCOL,NROW,NLPOR), # B00200
C     3IBOUND(NCOL,NROW,NLAY),XMAX(NCOL),DELR(NCOL),YMAX(NROW), # B00210
C     4DELC(NROW),ZBOT(NZDIM),ZTOP(NZDIM),DELZ(NLAY),DELZCB(NLAY), # B00220
C     5IBUFF(NCOL,NROW,NLAY),IUNIT(NUNIT),HEAD(NCOL,NROW,NLAY), # B00230
C     6LAYCON(NLAY),NCON(NLAY),BUFF(NCOL,NROW,NLAY), # B00240
C     7QSS(NCOL,NROW,NLAY) # B00250
C     CHARACTER*79 MES # B00260
C # B00270

```

```

        CALL DATIN (LAYCON,NCON,HEAD,DELT,XMAX,YMAX,DELR,DELC,DELZ,      # B00280
        1DELZCB,ZTOP,ZBOT,IBOUND,POR,QX,QY,QZ,IUNIT,NCOL,NROW,NLAY,NLPOR, # B00290
        2NCP1,NRP1,NLP1,NUNIT,NZDIM,IRP,NPART,NPRT,MODE,FRAC,BUFF,IREV,    # B00300
        3IBUFF,IGRID,JLC,ILC,KLC,XLC,YLC,ZLLC,NCBL,ITMS,QSS,ISNK,         # B00310
        4ITREAD,I1,I2,I4,I5,I6,I7,I8,I9,IZSTOP,IPEP,                     # B00320
        5IBATCH,MAXSTP)                                                    # B00330
C                                                                           # B00340
C CHECK MASS BALANCES                                                    # B00350
C                                                                           # B00360
        MES= 'DO YOU WANT TO CHECK THE MASS BALANCE FOR ALL CELLS ?'      # B00370
        CALL YESNO (IBATCH,I9,MES,IAN)                                     # B00380
        IF (IAN.EQ.1) THEN                                                 # B00390
        IF (IBATCH.EQ.0)                                                  # B00400
        1WRITE (*,*) 'SPECIFY AN ERROR TOLERANCE (IN PERCENT):'          # B00410
        IF (IBATCH.EQ.0) THEN                                             # B00420
        READ (*,*) ETOL                                                  # B00430
        WRITE(I9,*) ETOL                                                 # B00440
        ELSE                                                              # B00450
        READ(I9,*) ETOL                                                  # B00460
        END IF                                                            # B00470
        CALL BALNCE (QX,QY,QZ,QSS,IBOUND,HEAD,NCOL,NROW,NLAY,          # B00480
        1NCP1,NRP1,NLP1,KOUNT,I7,IBATCH,ETOL)                          # B00490
        IF (KOUNT.GT.0) THEN                                              # B00500
        MES= 'DO YOU WANT TO CONTINUE ?'                                  # B00510
        CALL YESNO (IBATCH,I9,MES,IAN)                                    # B00520
        IF (IAN.EQ.0) STOP                                                # B00530
        END IF                                                            # B00540
        END IF                                                            # B00550
C                                                                           # B00560
C CHECK DATA CELL BY CELL                                              # B00570
C                                                                           # B00580
        MES= ' DO YOU WANT TO CHECK DATA CELL BY CELL ?'               # B00590
        CALL YESNO (IBATCH,I9,MES,IAN)                                     # B00600
        IF (IAN.EQ.1) THEN                                                 # B00610
        CALL CELDAT (XMAX,YMAX,DELR,DELC,ZBOT,ZTOP,                     # B00620
        1DELZCB,POR,IBOUND,HEAD,QX,QY,QZ,LAYCON,NCOL,NROW,NLAY,NZDIM,    # B00630
        2IGRID,NLPOR,NCP1,NRP1,NLP1,NCON,QSS,I9,IBATCH)                # B00640
C                                                                           # B00650
        MES= ' DOES THE DATA APPEAR TO BE CORRECT?'                     # B00660
        CALL YESNO (IBATCH,I9,MES,IAN)                                    # B00670
        IF (IAN.EQ.0) RETURN                                              # B00680
        END IF                                                            # B00690
C                                                                           # B00700
C                                                                           # B00710
C SWITCH SIGNS OF FLOWS IF REVERSE TRACKING IS USED                     # B00720
C                                                                           # B00730
        IF (IREV.EQ.1) THEN                                              # B00740
        DO 10 K=1,NLAY                                                    # B00750
        DO 10 I=1,NROW                                                    # B00760
        DO 10 J=1,NCP1                                                    # B00770
10      QX(J,I,K)= -QX(J,I,K)                                           # B00780
        DO 20 K=1,NLAY                                                    # B00790
        DO 20 I=1,NRP1                                                    # B00800
        DO 20 J=1,NCOL                                                    # B00810
20      QY(J,I,K)= - QY(J,I,K)                                           # B00820
        DO 30 K=1,NLP1                                                    # B00830
        DO 30 I=1,NROW                                                    # B00840
        DO 30 J=1,NCOL                                                    # B00850
30      QZ(J,I,K)= -QZ(J,I,K)                                           # B00860
        END IF                                                            # B00870
C                                                                           # B00880
        IOP=0                                                            # B00890
        IF (MODE.NE.2) THEN                                               # B00900
        MES= 'DO YOU WANT TO WRITE ENDPOINT INFORMATION TO THE SCREEN ?' # B00910
        CALL YESNO (IBATCH,I9,MES,IOP)                                   # B00920
        IF (IOP.EQ.1.AND.NPRT.GT.100) THEN                               # B00930

```



```

      IF (IBATCH.EQ.0) WRITE(*,5040) NPRT                                # B00940
5040  FORMAT(I6,' LINES OF TEXT WILL BE WRITTEN TO THE SCREEN.')          # B00950
      MES='DO YOU REALLY WANT TO WRITE THAT MANY LINES TO THE SCREEN ?'    # B00960
      CALL YESNO (IBATCH,I9,MES,IOP)                                       # B00970
      END IF                                                                # B00980
      END IF                                                                # B00990
C                                     # B01000
      IF (IBATCH.EQ.0) THEN                                                # B01010
      WRITE (*,*) ' '                                                       # B01020
      WRITE (*,*) 'NOW COMPUTING PARTICLE PATHS...'                       # B01030
      WRITE (*,*) ' '                                                       # B01040
      END IF                                                                # B01050
C                                     # B01060
      KOUNT=0                                                              # B01070
      KSTP=0                                                                # B01080
      NSTP=0                                                                # B01090
      KPER=0                                                                # B01100
      TIME=0.0E+0                                                          # B01110
      TMAX=0.0E+0                                                          # B01120
      DO 40 N=1,NPART                                                      # B01130
40      TOT(N)=TIME                                                         # B01140
      IF (ITMS.EQ.1.AND.ITREAD.EQ.1) THEN                                # B01150
      READ(I8,'(I10)') NPER                                               # B01160
      IF (NPER.EQ.0) THEN                                                  # B01170
      WRITE (I7,*) 'TIME DATA FILE SAYS NPER = 0. RUN STOPPED.'          # B01180
      IF (IBATCH.EQ.0)                                                    # B01190
      1 WRITE (*,*) 'TIME DATA FILE SAY NPER = 0. RUN STOPPED.'          # B01200
      STOP                                                                # B01210
      END IF                                                                # B01220
      END IF                                                                # B01230
C                                     # B01240
C  IF TIME SERIES MODE, WRITE INITIAL CONDITIONS IN TIMESERIES FILE      # B01250
C                                     # B01260
      IF (MODE.EQ.2) THEN                                                  # B01270
      CALL ZCALC (IGRID,NCOL,NROW,NLAY,ZLC,ZLLC,ZTOP,ZBOT,                # B01280
      1 HEAD,LAYCON,ILC,JLC,KLC,NZDIM,NPART,NPRT,I7)                     # B01290
      DO 50 N=1,NPART                                                      # B01300
      WRITE (I4,5020) KOUNT,N,JLC(N),ILC(N),KLC(N),XLC(N),YLC(N),ZLC(N), # B01310
      1ZLLC(N),TMAX                                                         # B01320
50      CONTINUE                                                            # B01330
      END IF                                                                # B01340
C                                     # B01350
C----- START TIME LOOP -----                                         # B01360
C                                     # B01370
60      CONTINUE                                                            # B01380
      TOLD=TMAX                                                            # B01390
      KOUNT=KOUNT+1                                                         # B01400
      KSTP=KSTP+1                                                           # B01410
C                                     # B01420
C  INCREMENT TMAX                                                         # B01430
C                                     # B01440
C  ITMS = 0 -- LOCATIONS ARE NOT COMPUTED AT SPECIFIC POINTS IN TIME      # B01450
C  ITMS = 1 -- LOCATIONS ARE COMPUTED AT SPECIFIC POINTS IN TIME          # B01460
C                                     # B01470
      IF (ITMS.EQ.1) THEN                                                  # B01480
C                                     # B01490
C  ITREAD = 0 -- CONSTANT TIME STEP LENGTH                                # B01500
C  ITREAD = 1 -- TIME STEP DATA READ FROM A FILE                         # B01510
C                                     # B01520
      IF (ITREAD.EQ.0) THEN                                                # B01530
      TMAX=TMAX+DELT                                                        # B01540
      IF (KOUNT.GT.MAXSTP) TMAX=1.0E+30                                    # B01550
      ELSE                                                                  # B01560
      IF (KSTP.GT.NSTP.OR.KOUNT.EQ.1) THEN                                # B01570
      IF (KPER.LT.NPER) THEN                                                # B01580
      READ (I8,5000) PERLEN,NSTP,TSMULT,TFAC                              # B01590

```

```

      KPER=KPER+1                                # B01600
      KSTP=1                                      # B01610
      DELT= TFAC*PERLEN/FLOAT(NSTP)              # B01620
      IF(TSMULT.NE.1.)                           # B01630
1      DELT=TFAC*PERLEN*(1.-TSMULT)/(1.-TSMULT**NSTP) # B01640
      TMAX=TMAX+DELT                             # B01650
      ELSE                                        # B01660
      TMAX=1.0E+30                               # B01670
      ITMS=0                                      # B01680
      END IF                                     # B01690
      ELSE                                        # B01700
      DELT=TSMULT*DELT                           # B01710
      TMAX=TMAX+DELT                             # B01720
      END IF                                     # B01730
      IF (KOUNT.EQ.1) WRITE (I7,*) 'PARTICLES COORDINATES WERE COMPUTED' # B01740
1AT THE FOLLOWING POINTS IN TIME:'              # B01750
      IF(TMAX.LT.1.0E+30) WRITE (I7,5005) KOUNT,TMAX,KSTP,KPER # B01760
5005  FORMAT(I5,':  TIME =',1PE12.5,' (STEP',I4,',  PERIOD',I4,')') # B01770
      END IF                                     # B01780
5000  FORMAT(F10.0,I10,2F10.0)                 # B01790
      ELSE                                        # B01800
      TMAX= 1.0E+30                              # B01810
      END IF                                     # B01820
C                                             # B01830
C----- START PARTICLE LOOP -----          # B01840
C                                             # B01850
      IEND=0                                      # B01860
      DO 70 N=1,NPRT                             # B01870
      IF(TOT(N).GT.0.0E+0) GO TO 70              # B01880
C                                             # B01890
C SET START TIME FOR PARTICLE                 # B01900
C                                             # B01910
      TIME=TOLD                                  # B01920
C                                             # B01930
      CALL FLOLIN (N,IGRID,MODE,TIME,TMAX,IDSCH, # B01940
1JLC(N),ILC(N),KLC(N),XLC(N),YLC(N),ZLC(N),ZLLC(N),IBOUND,LAYCON, # B01950
2ZBOT,ZTOP,XMAX,YMAX,QX,QY,QZ,DELC,DEL,R,POR,HEAD,NCON,NCOL,NROW, # B01960
3NLAY,NLPOR,NZDIM,NCP1,NRP1,NLP1,ISNK,IREV,FRAC,IZSTOP,I2, # B01970
4I7,QSS,1)                                     # B01980
C                                             # B01990
C ASSIGN TIME OF TRAVEL TO TOT ARRAY IF PARTICLE HAS DISCHARGED # B02000
C                                             # B02010
C IDSCH = 0 -- PARTICLE HAS DISCHARGED         # B02020
C IDSCH = 1 -- PARTICLE HAS NOT DISCHARGED     # B02030
C                                             # B02040
      IF(IDSCH.EQ.0) TOT(N)=TIME                # B02050
      IEND=IEND+IDSCH                          # B02060
C                                             # B02070
C WRITE DISCHARGE INFO TO SCREEN IF IOP=1      # B02080
C                                             # B02090
C                                             # B02100
      IF (IDSCH.EQ.0) THEN                      # B02110
      IF(IOP.EQ.1.AND.IBATCH.EQ.0)             # B02120
1WRITE (*,5010) N,ILC(N),JLC(N),KLC(N),TOT(N) # B02130
5010  FORMAT(' PARTICLE',I5,' STOPS IN (I=',I3,', J=',I3,', K=', # B02140
1I3,') TRAVEL TIME = ',1PE12.4)              # B02150
      END IF                                   # B02160
C                                             # B02170
C DON'T WRITE RESULTS IN TIMESERIES FILE FOR TIME=TMAX IF PARTICLE # B02180
C DISCHARGED AT TIME < TMAX.                  # B02190
C                                             # B02200
C      IF(IDSCH.EQ.0.AND.TIME.LT.TMAX) GO TO 70 # B02210
C                                             # B02220
C WRITE RESULTS TO TIMESERIES FILE            # B02230
C                                             # B02240
C      IF (MODE.EQ.2) THEN                    # B02250

```

```

        WRITE(I4,5020) KOUNT,N,JLC(N),ILC(N),KLC(N),XLC(N),YLC(N),ZLC(N), # B02260
1ZLLC(N),TMAX # B02270
5020   FORMAT(2(I4,1X),3(I3,1X),4(E20.12,1X),E20.12) # B02280
        END IF # B02290
C # B02300
70   CONTINUE # B02310
C # B02320
C----- END PARTICLE LOOP ----- # B02330
C # B02340
C CHECK TO SEE IF ALL PARTICLES HAVE DISCHARGED. IF NOT GO THROUGH # B02350
C TIME LOOP AGAIN. # B02360
C # B02370
        IF(IEND.GT.0) GO TO 60 # B02380
C # B02390
C----- END TIME LOOP ----- # B02400
C # B02410
C WRITE DISCHARGE DATA TO ENDPOINT FILE # B02420
C # B02430
        REWIND(I5) # B02440
        DO 80 N=1,NPRT # B02450
        READ(I5,*) NN,XSTRT,YSTRT,ZLSTRT,JFRST,IFRST,KFRST # B02460
        IZONE=IBOUND(JLC(N),ILC(N),KLC(N)) # B02470
        IF(IZONE.LT.0) IZONE=-IZONE # B02480
        IF(IZONE.GE.1000) IZONE=IZONE/1000 # B02490
        IZONE2=IBOUND(JFRST,IFRST,KFRST) # B02500
        IF(IZONE2.LT.0) IZONE2=-IZONE2 # B02510
        IF(IZONE2.GE.1000) IZONE2=IZONE2/1000 # B02520
        IF (IPEP.EQ.0) THEN # B02530
        WRITE(I3,5030) IZONE,JLC(N),ILC(N),KLC(N),XLC(N),YLC(N),ZLC(N), # B02540
1ZLLC(N),TOT(N),XSTRT,YSTRT,ZLSTRT,JFRST,IFRST,KFRST,IZONE2 # B02550
        ELSE # B02560
        IF (IZONE.EQ.IZSTOP) WRITE (I3,5030) IZONE,JLC(N),ILC(N),KLC(N), # B02570
1 XLC(N),YLC(N),ZLC(N),ZLLC(N),TOT(N),XSTRT,YSTRT,ZLSTRT,JFRST, # B02580
2 IFRST,KFRST,IZONE2 # B02590
        END IF # B02600
80   CONTINUE # B02610
5030   FORMAT(4I4,8E12.4,4I4) # B02620
C # B02630
        RETURN # B02640
        END # B02650
C # B02660
C-----END OF ROUTINE----- # B02670
C # C00010
C---Version 1.2   April 27, 1990 # C00020
C***** # C00030
C # C00040
C # C00050
C # C00060
C THIS SUBROUTINE COMPUTES THE PATH OF A PARTICLE FROM AN INITIAL # C00070
C STARTING LOCATION AND STARTING TIME TO A DISCHARGE POINT OR A # C00080
C SPECIFIED POINT IN TIME. IF THE LINE MODE IS USED (MODE=1), THE # C00090
C COORDINATES OF EACH COMPUTED POINT ARE WRITTEN TO A FILE NAMED # C00100
C "PATHLINE" # C00110
C # C00120
C***** # C00130
C # C00140
        SUBROUTINE FLOLIN (IPART,IGRID,MODE,TIME,TMAX, # C00150
1IDSCH,JP,IP,KP,XP,YP,ZP,ZLOC,IBOUND,LAYCON,ZBOT,ZTOP,XMAX,YMAX, # C00160
2QX,QY,QZ,DELC,DELR,POR,HEAD,NCON,NCOL,NROW,NLAY,NLPOR,NZDIM, # C00170
3NCP1,NRP1,NLP1,ISNK,IREV,FRAC,IZSTOP,I2,I7,QSS,ISS) # C00180
C # C00190
        DIMENSION IBOUND(NCOL,NROW,NLAY),ZTOP(NZDIM),ZBOT(NZDIM), # C00200
1LAYCON(NLAY),XMAX(NCOL),YMAX(NROW),NCON(NLAY),QSS(NCOL,NROW,NLAY), # C00210
2POR(NCOL,NROW,NLPOR),QX(NCP1,NROW,NLAY),QY(NCOL,NRP1,NLAY), # C00220
3QZ(NCOL,NROW,NLP1),DELC(NROW),DELR(NCOL),HEAD(NCOL,NROW,NLAY) # C00230
C # C00240

```

```

C THE VARIABLE "ISS" IS A FLAG INDICATING STEADY STATE OR TRANSIENT # C00250
C CONDITIONS. IN THIS PROGRAM, ISS IS SET EQUAL TO 1 FOR STEADY STATE. # C00260
C # C00270
C KOLD=KP # C00280
C NSEGS= NCOL*NROW*NLAY # C00290
C IDSCH=1 # C00300
C IBND=IBOUND(JP,IP,KP) # C00310
C HED=HEAD(JP,IP,KP) # C00320
C # C00330
C RETURN IF PARTICLE IS IN INACTIVE CELL # C00340
C # C00350
C IF (IBND.EQ.0.OR.HED.GT.1.0E+29) THEN # C00360
C IDSCH=0 # C00370
C RETURN # C00380
C END IF # C00390
C # C00400
C RETURN IF PARTICLE IS IN THE AUTOMATIC DISCHARGE ZONE # C00410
C # C00420
C IBSTOP= IBOUND(JP,IP,KP) # C00430
C IF (IBSTOP.LT.0) IBSTOP= -IBSTOP # C00440
C IF (IBSTOP.GE.1000) IBSTOP=IBSTOP/1000 # C00450
C IF (IBSTOP.EQ.IZSTOP) THEN # C00460
C IDSCH=0 # C00470
C RETURN # C00480
C END IF # C00490
C # C00500
C INITIALIZE VARIABLES AND COMPUTE INITIAL VALUE OF # C00510
C NORMALIZED Z COORDINATE # C00520
C # C00530
C ILAST=0 # C00540
C IF (IGRID.EQ.1) THEN # C00550
C NK=KP # C00560
C NKP=NK+1 # C00570
C ELSE # C00580
C NK = (KP-1)*NCOL*NROW + (IP-1)*NCOL + JP # C00590
C NKP= NK + (NCOL*NROW) # C00600
C END IF # C00610
C IF (KP.EQ.NLAY.AND.ZLOC.LT.0.0) THEN # C00620
C WRITE(I7,*) ' RUN STOPPED. INCONSISTENT CONFINING BED LOCATION' # C00630
C STOP # C00640
C END IF # C00650
C ZMX=ZTOP(NK) # C00660
C HED=HEAD(JP,IP,KP) # C00670
C IF (LAYCON(KP).EQ.1) ZMX=HED # C00680
C IF (LAYCON(KP).GT.1.AND.HED.LT.ZMX) ZMX=HED # C00690
C IF (ZLOC.GE.0.0) THEN # C00700
C ZP=ZLOC*ZMX + (1.0-ZLOC)*ZBOT(NK) # C00710
C ELSE # C00720
C ZL= 1.0+ZLOC # C00730
C ZP=ZL*ZBOT(NK) + (1.0-ZL)*ZTOP(NKP) # C00740
C END IF # C00750
C # C00760
C WRITE STARTING COORDINATES # C00770
C # C00780
CKRR10/30/95) ADDED OUTPUT FOR CALCULATION OF LOCATION IN CELLS
C IF (MODE.EQ.1) WRITE(I2,170) VX,DVXDX,VX1,VX2,DT,XP,XMN,XXM,IVXFLG,
C & IX
C IF (MODE.EQ.1) WRITE(I2,170) VY,DVYDY,VY1,VY2,DT,YP,YN,XXM,IVYFLG,
C & IY
C IF (MODE.EQ.1) WRITE(I2,170) VZ,DVZDZ,VZ1,VZ2,DT,ZP,ZMN,ZMX,IVZFLG,
C & IZ
CKRR10/30/95
C IF (MODE.EQ.1) WRITE(I2,160) IPART,XP,YP,ZLOC,ZP,TIME, # C00790
C 1JP,IP,KP # C00800
C # C00810
C CHECK TO SEE IF PARTICLE IS LOCATED IN CONFINING BED. # C00820

```

```

C IF IT IS, MOVE PARTICLE TO NEXT ACTIVE LAYER. # C00830
C # C00840
IF(ZLOC.GT.0.0) GO TO 50 # C00850
IF(ZLOC.GE.0.0.AND.QZ(JP,IP,KP+1).GE.0.0) GO TO 50 # C00860
LCON=NCON(KP) # C00870
KPOR=NLAY+LCON # C00880
IF(LCON.EQ.0) GO TO 30 # C00890
IF(KP.EQ.NLAY) GO TO 30 # C00900
V= QZ(JP,IP,KP+1)/POR(JP,IP,KPOR)/DELC(IP)/DELR(JP) # C00910
IF(V.EQ.0.0) THEN # C00920
IDSCH=0 # C00930
RETURN # C00940
END IF # C00950
ZCB= ZTOP(NKP) # C00960
ZMN= ZBOT(NK) # C00970
IF(V.GT.0.0) DT= -ZLOC*(ZMN-ZCB)/V # C00980
IF(V.LT.0.0) DT= -(1.0+ZLOC)*(ZMN-ZCB)/V # C00990
TIM=TIME+DT # C01000
C # C01010
C CHECK TO SEE IF MAXIMUM TIME HAS BEEN REACHED OR EXCEEDED. # C01020
C IF SO, SET TIME EQUAL TO TMAX, CALCULATE PARTICLE LOCATION, & # C01030
C EXIT SUBROUTINE. # C01040
C # C01050
IF(TMAX.GT.TIM) GO TO 10 # C01060
DTT=TMAX-TIME # C01070
ZLC= ZLOC + V*DTT/(ZMN-ZCB) # C01080
ZP= -ZLC*ZCB + (1.0+ZLC)*ZMN # C01090
TTM=-TMAX # C01100
IF(MODE.EQ.1) WRITE(I2,160) IPART,XP,YP,ZLC,ZP,TTM, # C01110
1JP,IP,KP # C01120
TIME=TMAX # C01130
ZLOC=ZLC # C01140
RETURN # C01150
10 CONTINUE # C01160
C # C01170
TIME=TIME+DT # C01180
ZP=ZMN # C01190
ZLOC=0.0 # C01200
IF(V.GE.0.0) GO TO 20 # C01210
KP=KP+1 # C01220
ZP=ZCB # C01230
ZLOC=1.0 # C01240
20 IF(MODE.EQ.1) WRITE(I2,160) IPART,XP,YP,ZLOC,ZP,TIME,JP,IP,KP # C01250
GO TO 40 # C01260
30 WRITE(I7,*) ' RUN STOPPED. INCONSISTENT CONFINING BED LOCATION.' # C01270
STOP # C01280
40 CONTINUE # C01290
C # C01300
C THE PARTICLE IS NOW IN AN ACTIVE MODEL LAYER. CHECK TO SEE IF IT IS # C01310
C IN AN INACTIVE CELL WITHIN THE MODEL LAYER. IF SO, RESET # C01320
C KP AND ZLOC TO THE VALUES AT THE EXIT POINT OF PREVIOUS ACTIVE CELL. # C01330
C # C01340
IBND=IBOUND(JP,IP,KP) # C01350
HED=HEAD(JP,IP,KP) # C01360
IF(IBND.EQ.0.OR.HED.GT.1.0E+29) THEN # C01370
IDSCH=0 # C01380
KP=KOLD # C01390
ZLOC= -1.0 # C01400
RETURN # C01410
END IF # C01420
50 CONTINUE # C01430
C # C01440
C START TRACKING LOOP # C01450
C # C01460
DO 150 N=1,NSEGS # C01470
C # C01480

```

C	STORE CURRENT CELL LOCATION AND LOCAL Z COORDINATE	# C01490
C		# C01500
	JOLD=JP	# C01510
	IOLD=IP	# C01520
	KOLD=KP	# C01530
	ZLOLD=ZLOC	# C01540
C		# C01550
C	ASSIGN SCALAR VARIABLES FOR CONVENIENCE	# C01560
C		# C01570
	XMN=0.0	# C01580
	IF(JP.GT.1) XMN=XMAX(JP-1)	# C01590
	XX=XMAX(JP)	# C01600
	YMN=0.0	# C01610
	IF(IP.LT.NROW) YMN=YMAX(IP+1)	# C01620
	YY=YMAX(IP)	# C01630
	IF(IGRID.EQ.1) THEN	# C01640
	NK=KP	# C01650
	ELSE	# C01660
	NK= (KP-1)*NCOL*NROW + (IP-1)*NCOL + JP	# C01670
	END IF	# C01680
	ZMX=ZTOP(NK)	# C01690
	HED=HEAD(JP,IP,KP)	# C01700
	IF(LAYCON(KP).EQ.1) ZMX=HED	# C01710
	IF(LAYCON(KP).GT.1.AND.HED.LT.ZMX) ZMX=HED	# C01720
	ZMN= ZBOT(NK)	# C01730
	IF(ZMX.EQ.ZMN) ZMX=ZMX+0.01	# C01740
	ZP= ZLOC*ZMX + (1.0-ZLOC)*ZMN	# C01750
	DZ= ZMX-ZMN	# C01760
C		# C01770
C	ASSIGN FACE VELOCITIES	# C01780
C		# C01790
	PHI=POR(JP,IP,KP)	# C01800
	VX1= QX(JP,IP,KP)/PHI/DELC(IP)/DZ	# C01810
	VX2= QX(JP+1,IP,KP)/PHI/DELC(IP)/DZ	# C01820
	VY1= QY(JP,IP+1,KP)/PHI/DELR(JP)/DZ	# C01830
	VY2= QY(JP,IP,KP)/PHI/DELR(JP)/DZ	# C01840
	VZ1= QZ(JP,IP,KP+1)/PHI/DELC(IP)/DELR(JP)	# C01850
	VZ2= QZ(JP,IP,KP)/PHI/DELC(IP)/DELR(JP)	# C01860
C		# C01870
C	COMPUTE CELL TRANSIT TIMES IN X, Y, AND Z DIRECTIONS	# C01880
C		# C01890
	CALL DTCALC (VX1,VX2,VX,DVXDX,XMN,XX,XP,DTX,IVXFLG)	# C01900
	CALL DTCALC (VY1,VY2,VY,DVYDY,VMN,YY,YP,DTY,IVYFLG)	# C01910
	CALL DTCALC (VZ1,VZ2,VZ,DVZDZ,ZMN,ZMX,ZP,DTZ,IVZFLG)	# C01920
C		# C01930
C	DETERMINE EXIT FACE AND SET FLAGS	# C01940
C		# C01950
	IX=0	# C01960
	IY=0	# C01970
	IZ=0	# C01980
	DT=DTX	# C01990
	IX=1	# C02000
	IF(DTY.GE.DT) GO TO 60	# C02010
	DT=DTY	# C02020
	IX=0	# C02030
	IY=1	# C02040
	IZ=0	# C02050
60	IF(DTZ.GE.DT) GO TO 70	# C02060
	DT=DTZ	# C02070
	IY=0	# C02080
	IX=0	# C02090
	IZ=1	# C02100
70	CONTINUE	# C02110
	IF(VX.LT.0.0.AND.IX.GT.0) IX=-IX	# C02120
	IF(VY.LT.0.0.AND.IY.GT.0) IY=-IY	# C02130
	IF(VZ.LT.0.0.AND.IZ.GT.0) IZ=-IZ	# C02140

C		# C02150
C	CHECK TO SEE IF THIS CELL IS A DISCHARGE POINT	# C02160
C		# C02170
C	CHECK TO SEE IF THERE IS NO OUTFLOW FROM THIS CELL.	# C02180
C	IF SIMULATION IS STEADY STATE, DISCHARGE PARTICLE AND RETURN.	# C02190
C		# C02200
	IXYZ=0	# C02210
	IF(IVXFLG.GT.1) IXYZ=IXYZ+1	# C02220
	IF(IVYFLG.GT.1) IXYZ=IXYZ+1	# C02230
	IF(IVZFLG.GT.1) IXYZ=IXYZ+1	# C02240
	IF(IXYZ.EQ.3.AND.ISS.EQ.1) THEN	# C02250
	IDSCH=0	# C02260
	RETURN	# C02270
	END IF	# C02280
C		# C02290
C	RETURN IF PARTICLE IS IN THE AUTOMATIC DISCHARGE ZONE	# C02300
C		# C02310
	IF(IZSTOP.GT.0) THEN	# C02320
	IBABS=IBOUND(JP,IP,KP)	# C02330
	IF(IBOUND(JP,IP,KP).LT.0) IBABS= -IBOUND(JP,IP,KP)	# C02340
	IF(IBABS.GE.1000) IBABS=IBABS/1000	# C02350
	IF(IBABS.EQ.IZSTOP) THEN	# C02360
	IDSCH=0	# C02370
	RETURN	# C02380
	END IF	# C02390
	END IF	# C02400
C		# C02410
	IF(IBOUND(JP,IP,KP).GT.-1000.AND.IBOUND(JP,IP,KP).LT.1000)	# C02420
1	GO TO 80	# C02430
	IBD=0	# C02440
	IF(IBOUND(JP,IP,KP).LE.-1000.OR.IBOUND(JP,IP,KP).GE.1000) IBD=1	# C02450
	IF(IBD.EQ.1.AND.ISNK.GE.1) THEN	# C02460
	IF(TIME.LE.0.0.AND.IREV.EQ.1) GO TO 80	# C02470
	IF (ISNK.EQ.1) THEN	# C02480
	IDSCH=0	# C02490
	RETURN	# C02500
	END IF	# C02510
	END IF	# C02520
C		# C02530
C	DO A SECOND CHECK TO SEE IF PARTICLE CANNOT GET OUT OF CELL.	# C02540
C	IF IT CANNOT, DISCHARGE IT.	# C02550
C		# C02560
	IF(TIME.LE.0.0.AND.IREV.EQ.1) GO TO 80	# C02570
	DXDY= DELR(JP)*DELC(IP)	# C02580
	DXDZ= DELR(JP)*(ZMX-ZMN)	# C02590
	DYDZ= DELC(IP)*(ZMX-ZMN)	# C02600
	VSIGN=1.0	# C02610
	IF(IREV.EQ.1) VSIGN= -1.0	# C02620
	VLX1=VSIGN*VX1	# C02630
	VLX2=VSIGN*VX2	# C02640
	VLY1=VSIGN*VY1	# C02650
	VLY2=VSIGN*VY2	# C02660
	VLZ1=VSIGN*VZ1	# C02670
	VLZ2=VSIGN*VZ2	# C02680
	QINOUT= -QSS(JP,IP,KP)	# C02690
	IF(QINOUT.LE.0.0) GO TO 80	# C02700
	QI=0.0	# C02710
	IF(VLX1.GT.0.0) QI=QI+VLX1*DYZ	# C02720
	IF(VLX2.LT.0.0) QI=QI-VLX2*DYZ	# C02730
	IF(VLY1.GT.0.0) QI=QI+VLY1*DXDZ	# C02740
	IF(VLY2.LT.0.0) QI=QI-VLY2*DXDZ	# C02750
	IF(VLZ1.GT.0.0) QI=QI+VLZ1*DXDY	# C02760
	IF(VLZ2.LT.0.0) QI=QI-VLZ2*DXDY	# C02770
	IF(QI.LE.0.0) GO TO 80	# C02780
	F=QINOUT/QI/POR(JP,IP,KP)	# C02790
	IF(F.GT.FRAC) THEN	# C02800

```

        IDSCH=0                                # C02810
        RETURN                                # C02820
        END IF                                # C02830
C      # C02840
80    CONTINUE                                # C02850
C      # C02860
C    CHECK TO SEE IF TIME IS GREATER THAN OR EQUAL TO TMAX. # C02870
C    IF SO, SET TIME EQUAL TO TMAX, CALCULATE LOCATION & RETURN # C02880
C      # C02890
        TIM=TIME+DT                            # C02900
        IF(TMAX.GT.TIM) GO TO 90                # C02910
        DTT= TMAX-TIME                        # C02920
        CALL NEWXYZ (VX,DVXDX,VX1,VX2,DTT,XP,XMN,XXM,XPP,IVXFLG,0) # C02930
        CALL NEWXYZ (VY,DVYDY,VY1,VY2,DTT,YP,YMN,XXM,YPP,IVYFLG,0) # C02940
        CALL NEWXYZ (VZ,DVZDZ,VZ1,VZ2,DTT,ZP,ZMN,ZMX,ZPP,IVZFLG,0) # C02950
C      # C02960
        ZLOC= (ZPP-ZMN)/(ZMX-ZMN)              # C02970
C      # C02980
C    WRITE COORINATES OF INTERMEDIATE POINTS # C02990
C      # C03000
        TTM= -TMAX                             # C03010
        IF(MODE.EQ.1) WRITE(I2,160) IPART,XPP,YPP,ZLOC,ZPP,TTM, # C03020
        1JP,IP,KP                             # C03030
        XP=XPP                                 # C03040
        YP=YPP                                 # C03050
        ZP=ZPP                                 # C03060
        TIME=TMAX                             # C03070
        RETURN                                # C03080
90    CONTINUE                                # C03090
C      # C03100
C    COMPUTE PARTICLE COORDINATES AT THE EXIT POINT OF CELL (JP,IP,KP) # C03110
C      # C03120
CKRR 10/30/95 CODE ADDED TO ALLOW CALCULATE OF COORDINATES OUTSIDE MODPATH
        if(mode.eq.1)write(i2,170)vx,dvxdx,vx1,vx2,dt,xp,xmn,xxm,ivxflg,ix
        if(mode.eq.1)write(i2,170)vy,dvydy,vy1,vy2,dt,yp,ymn,xxm,ivyflg,iy
        if(mode.eq.1)write(i2,170)vz,dvzdz,vz1,vz2,dt,zp,zmn,zmx,ivzflg,iz
CKRR 10/30/95
        CALL NEWXYZ (VX,DVXDX,VX1,VX2,DT,XP,XMN,XXM,XNEW,IVXFLG,IX) # C03130
        CALL NEWXYZ (VY,DVYDY,VY1,VY2,DT,YP,YMN,XXM,YNEW,IVYFLG,IY) # C03140
        CALL NEWXYZ (VZ,DVZDZ,VZ1,VZ2,DT,ZP,ZMN,ZMX,ZNEW,IVZFLG,IZ) # C03150
C      write (i2,171)vx,dvxdx,vx1,vx2,dt,xp,xmn,xxm,xnew,ivxflg,ix
C      write (i2,171)vy,dvydy,vy1,vy2,dt,yp,ymn,xxm,ynew,ivyflg,iy
C      write (i2,171)vz,dvzdz,vz1,vz2,dt,zp,zmn,zmx,znew,ivzflg,iz
c 171  format (9e12.4,2i5)
        XP=XNEW                                # C03160
        YP=YNEW                                # C03170
        ZP=ZNEW                                # C03180
        TIME=TIME+DT                            # C03190
C      # C03200
        ZLOC= (ZP-ZMN)/(ZMX-ZMN)              # C03210
        IF(ZLOC.GT.1.0) ZLOC=1.0               # C03220
        IF(ZLOC.LT.0.0) ZLOC=0.0               # C03230
C      # C03240
        IF(IX.LT.0) JP=JP-1                    # C03250
        IF(IX.GT.0) JP=JP+1                    # C03260
        IF(IY.LT.0) IP=IP+1                    # C03270
        IF(IY.GT.0) IP=IP-1                    # C03280
        ICB=0                                  # C03290
        IF(IZ.LT.0.AND.NCON(KP).NE.0) ICB=1   # C03300
        IF(IZ.LT.0.AND.NCON(KP).EQ.0) KP=KP+1 # C03310
        IF(IZ.GT.0) KP=KP-1                    # C03320
C      # C03330
C    DISCHARGE PARTICLE IF IT REACHES BOUNDARY OF THE ACTIVE GRID # C03340
C      # C03350
C      IEXIT=0                                # C03360
        IF(JP.LT.1.OR.JP.GT.NCOL) IEXIT=1     # C03370

```



```

      IF(IP.LT.1.OR.IP.GT.NROW) IEXIT=1 # C03380
      IF(KP.LT.1.OR.KP.GT.NLAY) IEXIT=1 # C03390
      IF(IEXIT.EQ.1) GO TO 100 # C03400
      IBND=IBOUND(JP,IP,KP) # C03410
      HED=HEAD(JP,IP,KP) # C03420
      IF(IBND.EQ.0.OR.HED.GT.1.0E+29) IEXIT=1 # C03430
100  CONTINUE # C03440
      IF(IEXIT.EQ.1) THEN # C03450
      IDSCH=0 # C03460
      JP=JOLD # C03470
      IP=IOLD # C03480
      KP=KOLD # C03490
      IF(MODE.EQ.1) WRITE(I2,160) IPART,XP,YP,ZLOC,ZP,TIME,JP,IP,KP # C03500
      RETURN # C03510
      END IF # C03520
C # C03530
C SWITCH ZLOC FROM 0 TO 1 OR 1 TO 0 IF PARTICLE CHANGES LAYERS # C03540
C # C03550
      IF(IZ.LT.0.AND.NCON(KOLD).EQ.0) ZLOC=1.0 # C03560
      IF(IZ.GT.0.AND.NCON(KP).EQ.0) ZLOC=0.0 # C03570
      IF(IZ.GT.0.AND.NCON(KP).GT.0) THEN # C03580
      ZLOC= -1.0 # C03590
      ICB=1 # C03600
      END IF # C03610
C # C03620
C WRITE COORDINATES OF EXIT POINT # C03630
C # C03640
      IF(MODE.EQ.1) WRITE(I2,160) IPART,XP,YP,ZLOC,ZP,TIME,JP,IP,KP # C03650
C # C03660
C CHECK TO SEE IF PARTICLE HAS ENTERED CONFINING BED. # C03670
C IF SO, MOVE PARTICLE THROUGH CONFINING BED TO AN ACTIVE MODEL LAYER. # C03680
C OTHERWISE, GO TO END OF CELL LOOP. # C03690
C # C03700
      IF(ICB.EQ.0) GO TO 150 # C03710
      KOLD=KP # C03720
      LCON=NCON(KP) # C03730
      KPOR=NLAY+LCON # C03740
      IF(LCON.EQ.0) GO TO 130 # C03750
      IF(KP.EQ.NLAY) GO TO 130 # C03760
      V= QZ(JP,IP,KP+1)/POR(JP,IP,KPOR)/DELC(IP)/DELR(JP) # C03770
      IF(V.EQ.0.0) THEN # C03780
      IDSCH=0 # C03790
      RETURN # C03800
      END IF # C03810
      IF(IGRID.EQ.1) THEN # C03820
      NK= KP # C03830
      NKP=NK+1 # C03840
      ELSE # C03850
      NK= (KP-1)*NCOL*NROW + (IP-1)*NCOL + JP # C03860
      NKP=NK + (NCOL*NROW) # C03870
      END IF # C03880
      ZCB= ZTOP(NKP) # C03890
      ZMN= ZBOT(NK) # C03900
      IF(V.GT.0.0) DT= -ZLOC*(ZMN-ZCB)/V # C03910
      IF(V.LT.0.0) DT= -(1.0+ZLOC)*(ZMN-ZCB)/V # C03920
C # C03930
C CHECK TO SEE IF TIME IS GREATER THAN OR EQUAL TO TMAX. # C03940
C IF SO, SET TIME EQUAL TO TMAX, CALCULATE LOCATION, & # C03950
C EXIT SUBROUTINE. # C03960
C # C03970
      TIM=TIME+DT # C03980
      IF(TMAX.GT.TIM) GO TO 110 # C03990
      DTT=TMAX-TIME # C04000
      ZLC= ZLOC + V*DTT/(ZMN-ZCB) # C04010
      ZP= -ZLC*ZCB + (1.0+ZLC)*ZMN # C04020
      TTM=-TMAX # C04030

```

```

        IF(MODE.EQ.1) WRITE(I2,160) IPART,XP,YP,ZLC,ZP,TTM,JP,IP,KP      # C04040
        TIME=TMAX                                                         # C04050
        ZLOC=ZLC                                                         # C04060
        RETURN                                                            # C04070
110    CONTINUE                                                           # C04080
C                                          # C04090
        TIME=TIME+DT                                                     # C04100
        ZP=ZMN                                                           # C04110
        ZLOC=0.0                                                         # C04120
        IF(V.GE.0.0) GO TO 120                                           # C04130
        KP=KP+1                                                           # C04140
        ZP=ZCB                                                           # C04150
        ZLOC=1.0                                                         # C04160
120    IF(MODE.EQ.1) WRITE(I2,160) IPART,XP,YP,ZLOC,ZP,TIME,JP,IP,KP    # C04170
        GO TO 140                                                         # C04180
130    WRITE(I7,*) ' RUN STOPPED. INCONSISTENT CONFINING BED LOCATION.' # C04190
        STOP                                                             # C04200
140    CONTINUE                                                           # C04210
C                                          # C04220
C THE PARTICLE IS NOW IN AN ACTIVE MODEL LAYER. CHECK TO SEE IF IT IS    # C04230
C IN AN INACTIVE CELL WITHIN THE MODEL LAYER. IF SO, RESET              # C04240
C KP AND ZLOC TO THE VALUES AT THE EXIT POINT OF PREVIOUS ACTIVE CELL.  # C04250
C                                          # C04260
        IBND=IBOUND(JP,IP,KP)                                           # C04270
        HED=HEAD(JP,IP,KP)                                               # C04280
        IF(IBND.EQ.0.OR.HED.GT.1.0E+29) THEN                             # C04290
            IDSCH=0                                                       # C04300
            KP=KOLD                                                        # C04310
            ZLOC= -1.0                                                    # C04320
            RETURN                                                         # C04330
        END IF                                                            # C04340
150    CONTINUE                                                           # C04350
C                                          # C04360
        RETURN                                                            # C04370
160    FORMAT(I5,1X,5(E20.12,1X),2(I3,1X),I3)                          # C04380
CKRR 10/21/95
170    FORMAT(8E20.9,2I5)
CKRR
        END                                                              # C04390
C-----END OF ROUTINE-----# C04400
C                                          # D00010
C---Version 1.2   April 27, 1990                                       # D00020
C*****# D00030
C                                          # D00040
C                          NEWXYZ                                         # D00050
C                                          # D00060
C THIS SUBROUTINE COMPUTES A NEW X, Y, OR Z PARTICLE COORDINATE GIVEN    # D00070
C FACE VELOCITIES AND A TIME INTERVAL.                                   # D00080
C                                          # D00090
C*****# D00100
C                                          # D00110
        SUBROUTINE NEWXYZ (V,DVDX,V1,V2,DT,XP,XMIN,XMAX,XNEW,           # D00120
1  IVFLG,ISIDE)                                                         # D00130
C                                          # D00140
        IF(IVFLG.EQ.1) GO TO 10                                           # D00150
        IF(IVFLG.EQ.2) GO TO 20                                           # D00160
        XNEW= XP + V*(EXP(DVDX*DT) - 1.0E+0)/DVDX                       # D00170
        GO TO 30                                                         # D00180
10     XNEW= XP + V1*DT                                                  # D00190
        GO TO 30                                                         # D00200
20     XNEW= XP                                                          # D00210
        GO TO 40                                                         # D00220
30     IF(ISIDE.GT.0) XNEW=XMAX                                           # D00230
        IF(ISIDE.LT.0) XNEW=XMIN                                           # D00240
40     CONTINUE                                                           # D00250
        IF(XNEW.LE.XMIN.AND.V1.EQ.0.0E+0) XNEW=XMIN + 1.0E-3*(XMAX-XMIN) # D00260

```

```

        IF(XNEW.GE.XMAX.AND.V2.EQ.0.0E+0) XNEW=XMAX - 1.0E-3*(XMAX-XMIN) # D00270
        RETURN # D00280
        END # D00290
C-----END OF ROUTINE----- # D00300
C # E00010
C---Version 1.2 April 27, 1990 # E00020
C***** # E00030
C # E00040
C          DTCALC # E00050
C # E00060
C THIS SUBROUTINE CALCULATES THE TRANSIT TIME (DT) FROM THE CURRENT # E00070
C LOCATION TO A POTENTIAL DISCHARGE FACE IN A GIVEN DIRECTION. # E00080
C IT ALSO DETERMINES IF THERE IS NO POTENTIAL FOR DISCHARGE ACROSS # E00090
C EITHER OF THE FACES PERPEDICULAR TO THE GIVEN COORDINATE DIRECTION. # E00100
C IT SETS AND RETURNS A FLAG "IVFLG" TO INDICATE WHAT CONDITION EXISTS # E00110
C FOR THE GIVEN COORDINATE DIRECTION. # E00120
C # E00130
C      IVFLG = 0 -- "NORMAL" CONDITIONS. OUTFLOW CAN OCCUR ACROSS ONE # E00140
C                  OR BOTH OF THE FACES PERPENDICULAR TO THIS # E00150
C                  DIRECTION. # E00160
C      IVFLG = 1 -- A CONSTANT, NON-ZERO VELOCITY COMPONENT EXISTS IN # E00170
C                  THIS DIRECTION THROUGHOUT THE CELL. DT IS COMPUTED # E00180
C                  USING THE CONSTANT VELOCITY INSTEAD OF THE LOG # E00190
C                  EXPRESSION. # E00200
C      IVFLG = 2 -- "MACHINE" ZERO ( $V < 10^{-15}$ ) VELOCITY COMPONENT EXISTS # E00210
C                  THROUGHOUT THE CELL IN THIS DIRECTION. # E00220
C                  DT IS SET TO "MACHINE" INFINITY ( $10^{20}$ ). # E00230
C      IVFLG = 3 -- THERE IS NO POTENTIAL FOR OUTFLOW FROM EITHER FACE # E00240
C                  PERPENDICULAR TO THIS DIRECTION. # E00250
C                  DT IS SET TO "MACHINE" INFINITY ( $10^{20}$ ). # E00260
C # E00270
C***** # E00280
C # E00290
C      SUBROUTINE DTCALC (V1,V2,V,DVDX,X1,X2,XP,DT,IVFLG) # E00300
C # E00310
C          IVFLG=0 # E00320
C          DT= 1.0E+20 # E00330
C          V2A= ABS(V2) # E00340
C          V1A= ABS(V1) # E00350
C          DV= V2-V1 # E00360
C          DVA= ABS(DV) # E00370
C # E00380
C          CHECK FOR A UNIFORM ZERO VELOCITY IN THIS DIRECTION # E00390
C          IF SO, GO TO 30 AND RETURN WITH DT SET TO 1E+20 AND IVFLG=2 # E00400
C # E00410
C              IF(V2A.LE.1.0E-15.AND.V1A.LE.1.0E-15) GO TO 30 # E00420
C # E00430
C          CHECK FOR A CONSTANT UNIFORM NON-ZERO VELOCITY IN THIS DIRECTION. # E00440
C          VARIABLE VVV CAN ONLY BE  $< 0$  IF V1 AND V2 ARE IN THE SAME # E00450
C          DIRECTION. IF  $VVV < 1E-4$ , V1 AND V2 ARE ESSENTIALLY EQUAL. IF # E00460
C          THIS CONDITION EXISTS, GO TO 20 AND SET V=V1 AND COMPUTE TRAVEL # E00470
C          TIME USING CONSTANT VELOCITY. SET IVFLG=1. # E00480
C # E00490
C              VV=V1A # E00500
C              IF(V2A.GT.VV) VV=V2A # E00510
C              VVV= DVA/VV # E00520
C              IF(VVV.LE.1.0E-4) GO TO 20 # E00530
C # E00540
C          COMPUTE VELOCITY CORRESPONDING TO PARTICLE'S POSTION # E00550
C # E00560
C              DX= X2-X1 # E00570
C              DVDX= DV/DX # E00580
C              XF= (XP-X1)/DX # E00590
C              V= (1.0E+0-XF)*V1 + XF*V2 # E00600
C # E00610
C          IF FLOW IS INTO CELL FROM BOTH SIDES THERE IS NO OUTFLOW, SO # E00620

```

```

C GO TO 40 AND SET IVFLG=3 AND RETURN # E00630
C # E00640
C IF (V1.GE.0.0E+0.AND.V2.LE.0.0E+0) GO TO 40 # E00650
C # E00660
C IF FLOW IS OUT OF CELL ON BOTH SIDES, FIND LOCATION OF DIVIDE, # E00670
C COMPUTE TRAVEL TIME TO EXIT FACE AND RETURN WITH IVFLG=0 # E00680
C # E00690
C IF (V1.LE.0.0.AND.V2.GE.0.0) THEN # E00700
C IF (ABS(V).LE.0.0) THEN # E00710
C V= 1.0E-20 # E00720
C IF (V2.LE.0.0) V= -V # E00730
C END IF # E00740
C END IF # E00750
C VR1= V1/V # E00760
C VR2= V2/V # E00770
C VR=VR1 # E00780
C IF (VR.LE.0.0E+0) VR=VR2 # E00790
C V1V2= V1*V2 # E00800
C IF (V1V2.LE.0.0) GO TO 10 # E00810
C IF (V.GT.0.0) VR=VR2 # E00820
C IF (V.LT.0.0) VR=VR1 # E00830
10 CONTINUE # E00840
C DT= ALOG(VR)/DV DX # E00850
C GO TO 50 # E00860
20 CONTINUE # E00870
C IVFLG=1 # E00880
C ZRO= 1.0E-15 # E00890
C ZROM= -ZRO # E00900
C V=V1 # E00910
C IF (V1.GT.ZRO) DT= (X2-XP)/V1 # E00920
C IF (V1.LT.ZROM) DT= (X1-XP)/V1 # E00930
C GO TO 50 # E00940
30 IVFLG=2 # E00950
C GO TO 50 # E00960
40 IVFLG=3 # E00970
50 CONTINUE # E00980
C RETURN # E00990
C END # E01000
C-----END OF ROUTINE----- # E01010

C # F00010
C---Version 1.2 April 27, 1990 # F00020
C***** # F00030
C # F00040
C DATIN # F00050
C # F00060
C THIS SUBROUTINE CONTROLS DATA INPUT TO MODPATH. # F00070
C # F00080
C***** # F00090
C # F00100
C SUBROUTINE DATIN (LAYCON,NCON,HEAD,DELT,XMAX,YMAX,DELR,DELC,DELZ, # F00110
1DELZCB,ZTOP,ZBOT,IBOUND,POR,QX,QY,QZ,IUNIT,NCOL,NROW,NLAY,NLPOR, # F00120
2NCP1,NRP1,NLP1,NUNIT,NZDIM,IRP,NPART,NPRT,MODE,FRAC,BUFF,IREV, # F00130
3IBUFF,IGRID,JLC,ILC,KLC,XLC,YLC,ZLLC,NCBL,ITMS,QSS,ISNK, # F00140
4ITREAD,I1,I2,I4,I5,I6,I7,I8,I9,IZSTOP,IPEP, # F00150
5IBATCH,MAXSTP) # F00160
C # F00170
C CHARACTER*80 FLPART,FIL,OUT1,OUT2,OUT3 # F00180
C CHARACTER*79 MES # F00190
C CHARACTER*16 TEXT # F00200
C # F00210
C DIMENSION LAYCON(NLAY),NCON(NLAY),DELR(NCOL),DELC(NROW),DELZ(NLAY) # F00220
1,DELZCB(NLAY),XMAX(NCOL),YMAX(NROW),ZTOP(NZDIM),ZBOT(NZDIM), # F00230
2HEAD(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY),POR(NCOL,NROW,NLPOR), # F00240
3QX(NCP1,NROW,NLAY),QY(NCOL,NRP1,NLAY),QZ(NCOL,NROW,NLP1), # F00250
4IUNIT(NUNIT),BUFF(NCOL,NROW,NLAY),IBUFF(NCOL,NROW,NLAY), # F00260

```

```

5JLC(NPART), ILC(NPART), KLC(NPART), XLC(NPART), YLC(NPART),
6ZLLC(NPART), QSS(NCOL, NROW, NLAY)
C
IU=I1
NPRT=NPART
OUT1= 'PATHLINE'
OUT2= 'ENDPOINT'
OUT3= 'TIMESERS'
C
C WRITE NCOL, NROW, NLAY, NCBL, AND IGRID TO SUMMARY.PTH
C
WRITE(I7,5000)
5000 FORMAT(' -----#
1-----')
WRITE(I7,5010) NCOL,NROW,NLAY,NCBL
5010 FORMAT(I5,' COLUMNS',I5,' ROWS',I5,' LAYERS',I5,' CONFINING LAYERS#
1')
WRITE(I7,5020) IGRID
5020 FORMAT(' IGRID (GRID TYPE CODE) IS',I2)
C
C READ UNIT NUMBERS FOR INPUT FILES
C
READ(IU,5030) (IUNIT(N),N=1,NUNIT)
5030 FORMAT(16I5)
WRITE(I7,5000)
WRITE(I7,5040) (IUNIT(N),N=1,8)
5040 FORMAT(' IUNIT ARRAY: ',8I4)
C
C ENTER MODE DATA
C
IF (IBATCH.EQ.0) THEN
WRITE (*,*) 'SELECT THE MODE FOR STORING OUTPUT DATA:'
WRITE (*,*) ' 0 = ENDPOINTS AND STARTING POINTS ONLY'
WRITE (*,*) ' 1 = FLOW LINE COORDINATES'
WRITE (*,*) ' 2 = TIME SERIES DATA FOR SCATTER PLOTS'
END IF
IF (IBATCH.EQ.0) THEN
READ (*,*) MODE
WRITE(I9,*) MODE
ELSE
READ(I9,*) MODE
END IF
C
C ENTER TIME STEP INFORMATION FOR MODES 1 AND 2, AND DUMMY VALUES
C FOR MODE 0
C
IF (MODE.EQ.0) THEN
ITMS=0
END IF
IF (MODE.EQ.1.OR.MODE.EQ.2) THEN
IF (MODE.EQ.1) THEN
MES= 'DO YOU WANT TO COMPUTE LOCATIONS AT INTERMEDIATE TIMES ?'
CALL YESNO (IBATCH,I9,MES,ITMS)
ELSE IF (MODE.EQ.2) THEN
ITMS=1
END IF
IF (ITMS.EQ.1) THEN
IF (IBATCH.EQ.0) THEN
WRITE (*,*) 'HOW SHOULD POINTS IN TIME BE SPECIFIED ?'
WRITE (*,*) ' 0 = WITH A CONSTANT TIME STEP'
WRITE (*,*) ' 1 = VALUES OF TIME READ FROM A FILE'
END IF
IF (IBATCH.EQ.0) THEN
READ (*,*) ITREAD
WRITE(I9,*) ITREAD
ELSE

```

	READ(I9,*) ITREAD	# F00930
	END IF	# F00940
	IF(ITREAD.LT.0.OR.ITREAD.GT.1) THEN	# F00950
10	CONTINUE	# F00960
	IF(IBATCH.EQ.0) WRITE (*,*) 'ENTER 0 OR 1:'	# F00970
	IF(IBATCH.EQ.0) THEN	# F00980
	READ (*,*) ITREAD	# F00990
	WRITE(I9,*) ITREAD	# F01000
	ELSE	# F01010
	READ(I9,*) ITREAD	# F01020
	END IF	# F01030
	IF(ITREAD.LT.0.OR.ITREAD.GT.1) GO TO 10	# F01040
	END IF	# F01050
	IF (ITREAD.EQ.1) THEN	# F01060
	IF (IBATCH.EQ.0)	# F01070
	1WRITE (*,*) 'ENTER THE NAME OF THE DATA FILE:'	# F01080
	IF(IBATCH.EQ.0) THEN	# F01090
	READ (*,280) FIL	# F01100
	WRITE(I9,280) FIL	# F01110
	ELSE	# F01120
	READ(I9,280) FIL	# F01130
	END IF	# F01140
	CALL OPNFIL (I8,FIL,1,I7,IBATCH,1)	# F01150
5050	FORMAT (8F10.0)	# F01160
	ELSE	# F01170
	IF (IBATCH.EQ.0) WRITE (*,*)	# F01180
	1'ENTER: TIME STEP LENGTH, FACTOR FOR CONVERTING UNITS'	# F01190
	IF(IBATCH.EQ.0) THEN	# F01200
	READ (*,*) DELT,TMULT	# F01210
	WRITE(I9,*) DELT,TMULT	# F01220
	ELSE	# F01230
	READ(I9,*) DELT,TMULT	# F01240
	END IF	# F01250
	DELT=DELT*TMULT	# F01260
	IF (IBATCH.EQ.0) THEN	# F01270
	WRITE (*,*) 'ENTER THE MAXIMUM NUMBER OF TIME STEPS ALLOWED'	# F01280
	READ (*,*) MAXSTP	# F01290
	WRITE(I9,*) MAXSTP	# F01300
	ELSE	# F01310
	READ(I9,*) MAXSTP	# F01320
	END IF	# F01330
	END IF	# F01340
	END IF	# F01350
	END IF	# F01360
	IF (IBATCH.EQ.0) THEN	# F01370
	WRITE (*,*) 'HOW ARE STARTING LOCATIONS TO BE ENTERED?'	# F01380
	WRITE (*,*) ' 1 = FROM AN EXISTING DATA FILE'	# F01390
	WRITE (*,*) ' 2 = ARRAYS OF PARTICLES WILL BE GENERATED INTERNALLY'	# F01400
	END IF	# F01410
	IF(IBATCH.EQ.0) THEN	# F01420
	READ (*,*) IRP	# F01430
	WRITE(I9,*) IRP	# F01440
	ELSE	# F01450
	READ(I9,*) IRP	# F01460
	END IF	# F01470
	IF(IRP.LT.1.OR.IRP.GT.2) IRP=2	# F01480
		# F01490
C		# F01500
C	OPEN FILE FOR PARTICLE STARTING LOCATIONS	# F01510
C		# F01520
	IF(IRP.EQ.1) THEN	# F01530
	IF (IBATCH.EQ.0) WRITE (*,*)	# F01540
	1'ENTER NAME OF DATA FILE CONTAINING STARTING LOCATIONS:'	# F01550
	IF(IBATCH.EQ.0) THEN	# F01560
	READ (*,280) FLPART	# F01570
	WRITE(I9,280) FLPART	# F01580

```

ELSE                                                    # F01590
READ(I9,280) FLPART                                    # F01600
END IF                                                  # F01610
CALL OPNFIL (I6,FLPART,1,I7,IBATCH,1)                # F01620
ELSE IF (IRP.EQ.2) THEN                                # F01630
MES= 'DO YOU WANT TO STORE INTERNALLY-GENERATED STARTING LOCATIONS# F01640
1 ON DISK ?'                                           # F01650
CALL YESNO(IBATCH,I9,MES,ILSTOR)                     # F01660
IF(ILSTOR.EQ.1) THEN                                   # F01670
IF (IBATCH.EQ.0) WRITE (*,*) 'ENTER A FILE NAME:'    # F01680
IF(IBATCH.EQ.0) THEN                                   # F01690
READ (*,280) FLPART                                   # F01700
WRITE(I9,280) FLPART                                   # F01710
ELSE                                                    # F01720
READ(I9,280) FLPART                                   # F01730
END IF                                                  # F01740
CALL OPNFIL (I6,FLPART,2,I7,IBATCH,3)               # F01750
END IF                                                  # F01760
END IF                                                  # F01770
C                                                        # F01780
C CREATE AND OPEN 'PATHLINE' AND (OR) 'TIMESERS' FILES # F01790
C                                                        # F01800
IF(MODE.EQ.1) CALL OPNFIL (I2,OUT1,4,I7,IBATCH,3)    # F01810
IF(MODE.EQ.2) CALL OPNFIL (I4,OUT3,4,I7,IBATCH,3)    # F01820
C                                                        # F01830
C CHOOSE BETWEEN FORWARD AND REVERSE TRACKING        # F01840
C                                                        # F01850
IF (IBATCH.EQ.0) THEN                                  # F01860
WRITE (*,*) 'IN WHICH DIRECTION SHOULD PARTICLES BE TRACKED?' # F01870
WRITE (*,*) '      0 = FORWARD IN THE DIRECTION OF FLOW'      # F01880
WRITE (*,*) '      1 = BACKWARDS TOWARD RECHARGE LOCATIONS'   # F01890
END IF                                                  # F01900
IF(IBATCH.EQ.0) THEN                                   # F01910
READ (*,*) IREV                                         # F01920
WRITE(I9,*) IREV                                       # F01930
ELSE                                                    # F01940
READ(I9,*) IREV                                         # F01950
END IF                                                  # F01960
C                                                        # F01970
C SELECT DISCHARGE CRITERION                          # F01980
C                                                        # F01990
IF (IBATCH.EQ.0) THEN                                  # F02000
WRITE (*,*) 'HOW SHOULD PARTICLES BE TREATED WHEN THEY ENTER CELLS# F02010
1 WITH INTERNAL SINKS ?'                                # F02020
WRITE (*,*) '      0 = PARTICLES PASS THROUGH CELLS WITH WEAK SINKS'# F02030
WRITE (*,*) '      1 = PARTICLES ARE STOPPED WHEN THEY ENTER CELLS W# F02040
1ITH INTERNAL SINKS'                                    # F02050
WRITE (*,*) '      2 = PARTICLES ARE STOPPED WHEN THEY ENTER CELLS W# F02060
1HERE DISCHARGE TO SINKS'                              # F02070
WRITE (*,*) '      IS LARGER THAN A SPECIFIED FRACTION OF THE TO# F02080
1TAL INFLOW TO THE CELL'                                # F02090
END IF                                                  # F02100
IF(IBATCH.EQ.0) THEN                                   # F02110
READ (*,*) ISNK                                        # F02120
WRITE(I9,*) ISNK                                       # F02130
ELSE                                                    # F02140
READ(I9,*) ISNK                                       # F02150
END IF                                                  # F02160
FRAC=0.99                                              # F02170
IF(ISNK.EQ.2) THEN                                     # F02180
IF (IBATCH.EQ.0) THEN                                   # F02190
WRITE (*,*) 'ENTER A NUMBER BETWEEN 0 AND 1:'          # F02200
WRITE (*,*) '      (0.0 => NONE OF THE INFLOW TO THE CELL IS DISCHAR# F02210
1GED TO INTERNAL SINKS)'                                # F02220
WRITE (*,*) '      (1.0 => ALL INFLOW TO THE CELL IS DISCHARGED TO I# F02230
INTERNAL SINKS)'                                         # F02240

```

```

END IF                                     # F02250
IF (IBATCH.EQ.0) THEN                     # F02260
READ (*,*) FRAC                           # F02270
WRITE(I9,*) FRAC                           # F02280
ELSE                                       # F02290
READ(I9,*) FRAC                           # F02300
END IF                                     # F02310
IF (FRAC.GE.0.99) FRAC=0.99              # F02320
END IF                                     # F02330
C                                         # F02340
C SPECIFY AUTOMATIC DISCHARGE ZONE        # F02350
C                                         # F02360
IZSTOP=0                                  # F02370
IPEP=0                                    # F02380
MES= 'DO YOU WANT TO SPECIFY ONE ZONE IN WHICH TO STOP PARTICLES W# F02390
1HENEVER THEY ENTER?'                     # F02400
CALL YESNO (IBATCH,I9,MES,IAN$)           # F02410
IF (IAN$EQ.1) THEN                        # F02420
IF (IBATCH.EQ.0)                          # F02430
1WRITE (*,*) 'ENTER THE ZONE NUMBER (MUST BE > 1):' # F02440
IF (IBATCH.EQ.0) THEN                     # F02450
READ (*,*) IZSTOP                         # F02460
WRITE(I9,*) IZSTOP                        # F02470
ELSE                                       # F02480
READ(I9,*) IZSTOP                         # F02490
END IF                                     # F02500
IF (IZSTOP.LT.2) IZSTOP=0                # F02510
IF (IBATCH.EQ.0) THEN                     # F02520
WRITE (*,*) 'SPECIFY WHICH ENDPOINTS TO RECORD:' # F02530
WRITE (*,*) ' 0 = ENDPOINT DATA RECORDED FOR ALL PARTICLES' # F02540
WRITE (*,*) ' 1 = ENDPOINT DATA RECORDED ONLY FOR PARTICLES' # F02550
WRITE (*,5060) IZSTOP                     # F02560
END IF                                     # F02570
5060 FORMAT ('          TERMINATING IN ZONE ',I3) # F02580
IF (IBATCH.EQ.0) THEN                     # F02590
READ (*,*) IPEP                           # F02600
WRITE(I9,*) IPEP                           # F02610
ELSE                                       # F02620
READ(I9,*) IPEP                           # F02630
END IF                                     # F02640
IF (IPEP.LT.0.OR.IPEP.GT.1) IPEP=0        # F02650
END IF                                     # F02660
C                                         # F02670
IF (IBATCH.EQ.0)                          # F02680
1WRITE (*,*) 'READING AND PROCESSING INPUT DATA...' # F02690
C                                         # F02700
C LAYER TYPE CODES                        # F02710
C                                         # F02720
READ(IU,5070) (LAYCON(N),N=1,NLAY)        # F02730
5070 FORMAT(40I2)                         # F02740
WRITE(I7,5000)                            # F02750
WRITE(I7,*) 'LAYCON (LAYER TYPE CODES):' # F02760
WRITE(I7,5080) (LAYCON(N),N=1,NLAY)        # F02770
5080 FORMAT(25I3)                         # F02780
C                                         # F02790
C CONFINING BED CODES                     # F02800
C                                         # F02810
IF (NCBL.GT.0) THEN                       # F02820
READ(IU,5070) (NCON(N),N=1,NLAY)          # F02830
ELSE                                       # F02840
DO 20 N=1,NLAY                            # F02850
20 NCON(N)=0                              # F02860
END IF                                     # F02870
IF (NCBL.EQ.0) THEN                       # F02880
WRITE(I7,5000)                            # F02890
WRITE(I7,*) 'NO CONFINING LAYERS. NCON = 0 FOR ALL LAYERS.' # F02900

```


	ELSE	# F02910
	WRITE(I7,5000)	# F02920
	WRITE(I7,*) 'NCON (CONFINING LAYER CODES):'	# F02930
	WRITE(I7,5080) (NCON(N),N=1,NLAY)	# F02940
	END IF	# F02950
	IF(NCBL.GT.0) THEN	# F02960
	NN=0	# F02970
	DO 30 N=1,NLAY	# F02980
	IF(NCON(N).EQ.0) GO TO 30	# F02990
	NN=NN+1	# F03000
	NCON(N)=NN	# F03010
30	CONTINUE	# F03020
	END IF	# F03030
C		# F03040
C	DELR, GRID SPACING ALONG A ROW (X-DIRECTION)	# F03050
C		# F03060
	WRITE(I7,5000)	# F03070
	WRITE(I7,*) 'DELR ARRAY NOW BEING READ...'	# F03080
	CALL IN1DR (IU,DELR,NCOL,I7)	# F03090
C		# F03100
C	COMPUTE XMAX ARRAY	# F03110
C		# F03120
	XMAX(1)=DELR(1)	# F03130
	DO 40 J=2,NCOL	# F03140
40	XMAX(J)=XMAX(J-1)+DELR(J)	# F03150
C		# F03160
C	DELC, GRID SPACING ALONG A COLUMN (Y-DIRECTION)	# F03170
C		# F03180
	WRITE(I7,5000)	# F03190
	WRITE(I7,*) 'DELC ARRAY NOW BEING READ...'	# F03200
	CALL IN1DR (IU,DELC,NROW,I7)	# F03210
C		# F03220
C	COMPUTE YMAX ARRAY	# F03230
C		# F03240
	YMAX(NROW)=DELC(NROW)	# F03250
	DO 50 I=2,NROW	# F03260
	II=NROW+1-I	# F03270
50	YMAX(II)=YMAX(II+1)+DELC(II)	# F03280
C		# F03290
C	READ VERTICAL COORDINATE DATA	# F03300
C		# F03310
	IF (IGRID.EQ.1) THEN	# F03320
	WRITE(I7,5000)	# F03330
	WRITE(I7,*) 'DELZ ARRAY NOW BEING READ...'	# F03340
	CALL IN1DR (IU,DELZ,NLAY,I7)	# F03350
	IF(NCBL.GT.0) THEN	# F03360
	WRITE(I7,5000)	# F03370
	WRITE(I7,*) 'DELZCB ARRAY NOW BEING READ...'	# F03380
	CALL IN1DR (IU,DELZCB,NLAY,I7)	# F03390
	ELSE	# F03400
	DO 60 N=1,NLAY	# F03410
60	DELZCB(N)=0.0	# F03420
	END IF	# F03430
	READ(IU,5090) ZBL1	# F03440
5090	FORMAT(F10.0)	# F03450
	ZBOT(1)=ZBL1	# F03460
	WRITE(I7,5000)	# F03470
	WRITE(I7,5100) ZBOT(1)	# F03480
5100	FORMAT(' BOTTOM ELEVATION OF LAYER 1 IS',E13.5)	# F03490
	ZTOP(1)= ZBOT(1) + DELZ(1)	# F03500
	DO 70 K=2,NLAY	# F03510
	ZTOP(K)= ZBOT(K-1) - DELZCB(K-1)	# F03520
	ZBOT(K)= ZTOP(K) - DELZ(K)	# F03530
70	CONTINUE	# F03540
	ELSE	# F03550
	DO 80 K=1,NLAY	# F03560

```

      KP= 1 + (K-1)*NCOL*NROW                                # F03570
      IF(LAYCON(K).NE.1) THEN                                  # F03580
      WRITE(I7,5000)                                           # F03590
      WRITE(I7,5110) K                                         # F03600
5110  FORMAT(' TOP ELEVATION OF LAYER',I4,' NOW BEING READ...') # F03610
      CALL IN2DR (IU,ZTOP(KP),NCOL,NROW,I7)                   # F03620
      END IF                                                  # F03630
      WRITE(I7,5000)                                           # F03640
      WRITE(I7,5120) K                                         # F03650
5120  FORMAT(' BOTTOM ELEVATION OF LAYER',I4,' NOW BEING READ...') # F03660
      CALL IN2DR (IU,ZBOT(KP),NCOL,NROW,I7)                   # F03670
80    CONTINUE                                                # F03680
      END IF                                                  # F03690
C                                           # F03700
C IBOUND DATA                                           # F03710
C                                           # F03720
      DO 90 K=1,NLAY                                           # F03730
      WRITE(I7,5000)                                           # F03740
      WRITE(I7,5130) K                                         # F03750
5130  FORMAT(' IBOUND ARRAY FOR LAYER',I4,' NOW BEING READ...') # F03760
      CALL IN2DI (IU,IBOUND(1,1,K),NCOL,NROW,I7)              # F03770
90    CONTINUE                                                # F03780
C                                           # F03790
C POROSITY DATA                                           # F03800
C                                           # F03810
      DO 100 K=1,NLAY                                          # F03820
      WRITE(I7,5000)                                           # F03830
      WRITE(I7,5140) K                                         # F03840
5140  FORMAT(' POROSITY ARRAY FOR LAYER',I4,' NOW BEING READ...') # F03850
      CALL IN2DR (IU,POR(1,1,K),NCOL,NROW,I7)                 # F03860
      LCON=NCOL(K)                                             # F03870
      IF(LCON.GT.0) THEN                                       # F03880
      KCB=NLAY+LCON                                           # F03890
      WRITE(I7,5000)                                           # F03900
      WRITE(I7,5150) K                                         # F03910
5150  FORMAT(' POROSITY ARRAY FOR CONFINING LAYER UNDERLYING MODEL LAYER# F03920
1',I4,' NOW BEING READ...') # F03930
      CALL IN2DR (IU,POR(1,1,KCB),NCOL,NROW,I7)               # F03940
      END IF                                                  # F03950
100   CONTINUE                                                # F03960
C                                           # F03970
C READ HEADS                                           # F03980
C                                           # F03990
      DO 110 K=1,NLAY                                          # F04000
      DO 110 I=1,NROW                                          # F04010
      DO 110 J=1,NCOL                                          # F04020
110   HEAD(J,I,K)=0.                                         # F04030
      IUHED=IUNIT(8)                                           # F04040
      IF (IUHED.NE.0) THEN                                     # F04050
120   CONTINUE                                                # F04060
      READ (IUHED,END=130) KSTP,KPER,PERTIM,TOTIM,TEXT,NC,NR,K # F04070
      IF (TEXT.NE.' HEAD') THEN                                # F04080
      IF (IBATCH.EQ.0)                                         # F04090
1WRITE (*,*) 'HEAD FILE DOES NOT CONTAIN HEAD DATA.'        # F04100
      WRITE (I7,*) 'HEAD FILE DOES NOT CONTAIN HEAD DATA.'   # F04110
      STOP                                                     # F04120
      END IF                                                  # F04130
      WRITE(I7,5000)                                           # F04140
      WRITE(I7,5160) K                                         # F04150
5160  FORMAT(' HEADS NOW BEING READ FOR LAYER',I4)           # F04160
      READ(IUHED) ((HEAD(J,I,K),J=1,NCOL),I=1,NROW)          # F04170
      GO TO 120                                                # F04180
      END IF                                                  # F04190
130   CONTINUE                                                # F04200
      WRITE(I7,*) 'HEADS HAVE BEEN READ'                      # F04210
      WRITE(I7,*) '**** HEADS FOR INDIVIDUAL CELLS CAN BE CHECKED USING # F04220

```

```

      1THE CELL-BY-CELL OPTION *****'                                # F04230
C                                                                    # F04240
C  SET IBOUND = 0 FOR DRY CELLS                                         # F04250
C                                                                    # F04260
      DO 140 K=1,NLAY                                                    # F04270
      DO 140 I=1,NROW                                                    # F04280
      DO 140 J=1,NCOL                                                    # F04290
      IF (HEAD(J,I,K).GT.1.0E+29) IBOUND(J,I,K)=0                      # F04300
140  CONTINUE                                                            # F04310
C                                                                    # F04320
      CALL CELFLO (QX,QY,QZ,BUFF,IUNIT,IBOUND,HEAD,IBUFF,DELR,         # F04330
      1DELCL,NCOL,NROW,NLAY,NCPL,NRP1,NLP1,NUNIT,QSS,I7,IBATCH,I9)    # F04340
C                                                                    # F04350
C  READ PARTICLE LOCATIONS FROM DATA FILE                             # F04360
C                                                                    # F04370
      IF(IRP.EQ.1) THEN                                                  # F04380
      WRITE(I7,5000)                                                    # F04390
      WRITE(I7,*) 'STARTING LOCATIONS NOW BEING READ FROM FILE...'    # F04400
      N=0                                                                # F04410
150  N=N+1                                                                # F04420
      IF(N.GT.NPART) GO TO 190                                           # F04430
160  READ(I6,*,END=190) JLC(N),ILC(N),KLC(N),XLC(N),YLC(N),ZLLC(N)    # F04440
      NPRT=N                                                             # F04450
      JP=JLC(N)                                                         # F04460
      IP=ILC(N)                                                         # F04470
      IF(KLC(N).EQ.0) THEN                                              # F04480
      ZLLC(N)=1.0                                                       # F04490
      DO 170 K=1,NLAY                                                  # F04500
      KLC(N)=K                                                           # F04510
      IF(IBOUND(JP,IP,K).EQ.0) GO TO 170                               # F04520
      IF(HEAD(JP,IP,K).GE.1.0E+29) GO TO 170                           # F04530
      GO TO 180                                                         # F04540
170  CONTINUE                                                            # F04550
      GO TO 160                                                         # F04560
      END IF                                                            # F04570
180  CONTINUE                                                            # F04580
      IF(IBOUND(JP,IP,KLC(N)).EQ.0) GO TO 160                          # F04590
C                                                                    # F04600
C  CONVERT LOCAL X AND Y TO GLOBAL COORDINATES                         # F04610
C                                                                    # F04620
      IF(XLC(N).LE.0.0) XLC(N)=0.001                                    # F04630
      IF(XLC(N).GE.1.0) XLC(N)=0.999                                    # F04640
      IF(YLC(N).LE.0.0) YLC(N)=0.001                                    # F04650
      IF(YLC(N).GE.1.0) YLC(N)=0.999                                    # F04660
      IF(ZLLC(N).LE.0.0) ZLLC(N)=0.001                                  # F04670
      IF(ZLLC(N).GE.1.0) ZLLC(N)=0.999                                  # F04680
      XMN=0.0                                                            # F04690
      XMX=XMAX(JP)                                                       # F04700
      IF(JP.GT.1) XMN= XMAX(JP-1)                                        # F04710
      XLC(N)= (1.0-XLC(N))*XMN + XLC(N)*XMX                             # F04720
      YMN=0.0                                                            # F04730
      YMX=YMAX(IP)                                                       # F04740
      IF(IP.LT.NROW) YMN=YMAX(IP+1)                                      # F04750
      YLC(N)= (1.0-YLC(N))*YMN + YLC(N)*YMX                             # F04760
      WRITE(I5,5170) N,XLC(N),YLC(N),ZLLC(N),JLC(N),ILC(N),KLC(N)    # F04770
5170  FORMAT(I10,3E15.6,3I5)                                           # F04780
      GO TO 150                                                         # F04790
190  CONTINUE                                                            # F04800
      WRITE(I7,*) ' STARTING LOCATIONS HAVE BEEN READ FROM FILE'      # F04810
      END IF                                                            # F04820
      IF(IRP.EQ.2) THEN                                                  # F04830
      MES= ' STARTING LOCATIONS WILL BE GENERATED FOR 3-D SUBREGIONS.' # F04840
      IF (IBATCH.EQ.0) WRITE (*,5180) MES                               # F04850
5180  FORMAT(1X,A)                                                       # F04860
      IF (IBATCH.EQ.0) WRITE (*,*) ' '                                  # F04870
      MES= 'SHOULD PARTICLES BE PLACED IN CONSTANT HEAD CELLS ?'      # F04880

```

```

CALL YESNO (IBATCH,I9,MES,ICHEAD) # F04890
N=0 # F04900
KO=0 # F04910
200 CONTINUE # F04920
KO=KO+1 # F04930
IF (IBATCH.EQ.0) THEN # F04940
WRITE (*,5190) KO # F04950
5190 FORMAT(' ENTER DATA FOR SUBREGION ',I2,' :') # F04960
WRITE (*,*) ' ' # F04970
WRITE (*,*) 'DEFINE THE SUBREGION BY ENTERING THE MINIMUM AND MAXI# F04980
1MUM J,I,K COORDINATES.' # F04990
WRITE (*,*) ' ENTER: MINIMUM J, MAXIMUM J, MINIMUM I, MAXIMUM I, # F05000
1 MINIMUM K, MAXIMUM K' # F05010
END IF # F05020
IF (IBATCH.EQ.0) THEN # F05030
READ (*,*) JMIN,JMAX,IMIN,IMAX,KMIN,KMAX # F05040
WRITE(I9,*) JMIN,JMAX,IMIN,IMAX,KMIN,KMAX # F05050
ELSE # F05060
READ(I9,*) JMIN,JMAX,IMIN,IMAX,KMIN,KMAX # F05070
END IF # F05080
IF (JMAX.LT.1.OR.JMAX.GT.NCOL) JMAX=NCOL # F05090
IF (IMAX.LT.1.OR.IMAX.GT.NROW) IMAX=NCOL # F05100
IF (KMAX.LT.1.OR.KMAX.GT.NLAY) KMAX=NLAY # F05110
KWT=0 # F05120
IF (KMIN.EQ.0) THEN # F05130
KMIN=1 # F05140
KMAX=1 # F05150
KWT=1 # F05160
END IF # F05170
IF (IBATCH.EQ.0) THEN # F05180
WRITE (*,*) 'WHERE SHOULD THE PARTICLES BE LOCATED ?' # F05190
WRITE (*,*) ' 0 = WITHIN A CELL' # F05200
WRITE (*,*) ' 1 = ON ONE OR MORE OF THE CELL FACES' # F05210
END IF # F05220
IF (IBATCH.EQ.0) THEN # F05230
READ (*,*) ILOC # F05240
WRITE(I9,*) ILOC # F05250
ELSE # F05260
READ(I9,*) ILOC # F05270
END IF # F05280
IF (ILOC.EQ.0) THEN # F05290
IF (IBATCH.EQ.0) THEN # F05300
WRITE (*,*) 'ENTER: NJ NI NK' # F05310
WRITE (*,*) ' NJ = NUMBER OF PARTICLES PER CELL IN THE J DIRECTI# F05320
1ON' # F05330
WRITE (*,*) ' NI = NUMBER OF PARTICLES PER CELL IN THE I DIRECTI# F05340
1ON' # F05350
WRITE (*,*) ' NK = NUMBER OF PARTICLES PER CELL IN THE K DIRECTI# F05360
1ON' # F05370
END IF # F05380
IF (IBATCH.EQ.0) THEN # F05390
READ (*,*) NJ,NI,NK # F05400
WRITE(I9,*) NJ,NI,NK # F05410
ELSE # F05420
READ(I9,*) NJ,NI,NK # F05430
END IF # F05440
IF (NJ.EQ.0) NJ=1 # F05450
IF (NI.EQ.0) NI=1 # F05460
IF (NK.EQ.0) NK=1 # F05470
ELSE # F05480
MES= 'DO YOU WANT TO PLACE PARTICLES ON FACE 1 ?' # F05490
CALL YESNO (IBATCH,I9,MES,IF1) # F05500
IF (IF1.EQ.1) THEN # F05510
IF (IBATCH.EQ.0) THEN # F05520
WRITE (*,*) 'ENTER: NI NK' # F05530
WRITE (*,*) ' NI = NUMBER OF PARTICLES IN THE I DIRECTION FOR FA# F05540

```

```

1CE 1'                                     # F05550
WRITE (*,*) '    NK = NUMBER OF PARTICLES IN THE K DIRECTION FOR FA# F05560
1CE 1'                                     # F05570
END IF                                     # F05580
IF (IBATCH.EQ.0) THEN                     # F05590
READ (*,*) NI1,NK1                       # F05600
WRITE(I9,*) NI1,NK1                     # F05610
ELSE                                      # F05620
READ(I9,*) NI1,NK1                       # F05630
END IF                                    # F05640
END IF                                    # F05650
MES= 'DO YOU WANT TO PLACE PARTICLES ON FACE 2 ?' # F05660
CALL YESNO (IBATCH,I9,MES,IF2)           # F05670
IF (IF2.EQ.1) THEN                       # F05680
IF (IBATCH.EQ.0) THEN                   # F05690
WRITE (*,*) 'ENTER:  NI  NK'           # F05700
WRITE (*,*) '    NI = NUMBER OF PARTICLES IN THE I DIRECTION FOR FA# F05710
1CE 2'                                     # F05720
WRITE (*,*) '    NK = NUMBER OF PARTICLES IN THE K DIRECTION FOR FA# F05730
1CE 2'                                     # F05740
END IF                                     # F05750
IF (IBATCH.EQ.0) THEN                   # F05760
READ (*,*) NI2,NK2                     # F05770
WRITE(I9,*) NI2,NK2                   # F05780
ELSE                                    # F05790
READ(I9,*) NI2,NK2                     # F05800
END IF                                    # F05810
END IF                                    # F05820
MES= 'DO YOU WANT TO PLACE PARTICLES ON FACE 3 ?' # F05830
CALL YESNO (IBATCH,I9,MES,IF3)           # F05840
IF (IF3.EQ.1) THEN                     # F05850
IF (IBATCH.EQ.0) THEN                   # F05860
WRITE (*,*) 'ENTER:  NJ  NK'           # F05870
WRITE (*,*) '    NJ = NUMBER OF PARTICLES IN THE J DIRECTION FOR FA# F05880
1CE 3'                                     # F05890
WRITE (*,*) '    NK = NUMBER OF PARTICLES IN THE K DIRECTION FOR FA# F05900
1CE 3'                                     # F05910
END IF                                     # F05920
IF (IBATCH.EQ.0) THEN                   # F05930
READ (*,*) NJ3,NK3                     # F05940
WRITE(I9,*) NJ3,NK3                   # F05950
ELSE                                    # F05960
READ(I9,*) NJ3,NK3                     # F05970
END IF                                    # F05980
END IF                                    # F05990
MES= 'DO YOU WANT TO PLACE PARTICLES ON FACE 4 ?' # F06000
CALL YESNO (IBATCH,I9,MES,IF4)           # F06010
IF (IF4.EQ.1) THEN                     # F06020
IF (IBATCH.EQ.0) THEN                   # F06030
WRITE (*,*) 'ENTER:  NJ  NK'           # F06040
WRITE (*,*) '    NJ = NUMBER OF PARTICLES IN THE J DIRECTION FOR FA# F06050
1CE 4'                                     # F06060
WRITE (*,*) '    NK = NUMBER OF PARTICLES IN THE K DIRECTION FOR FA# F06070
1CE 4'                                     # F06080
END IF                                     # F06090
IF (IBATCH.EQ.0) THEN                   # F06100
READ (*,*) NJ4,NK4                     # F06110
WRITE(I9,*) NJ4,NK4                   # F06120
ELSE                                    # F06130
READ(I9,*) NJ4,NK4                     # F06140
END IF                                    # F06150
END IF                                    # F06160
MES= 'DO YOU WANT TO PLACE PARTICLES ON FACE 5 ?' # F06170
CALL YESNO (IBATCH,I9,MES,IF5)           # F06180
IF (IF5.EQ.1) THEN                     # F06190
IF (IBATCH.EQ.0) THEN                   # F06200

```

```

WRITE (*,*) 'ENTER:  NJ  NI'                                # F06210
WRITE (*,*) '    NJ = NUMBER OF PARTICLES IN THE J DIRECTION FOR FA# F06220
1CE 5'                                                         # F06230
WRITE (*,*) '    NI = NUMBER OF PARTICLES IN THE I DIRECTION FOR FA# F06240
1CE 5'                                                         # F06250
END IF                                                         # F06260
IF (IBATCH.EQ.0) THEN                                         # F06270
  READ (*,*) NJ5,NI5                                          # F06280
  WRITE(I9,*) NJ5,NI5                                          # F06290
ELSE                                                           # F06300
  READ(I9,*) NJ5,NI5                                          # F06310
END IF                                                         # F06320
END IF                                                         # F06330
MES= 'DO YOU WANT TO PLACE PARTICLES ON FACE 6 ?'           # F06340
CALL YESNO (IBATCH,I9,MES,IF6)                                # F06350
IF (IF6.EQ.1) THEN                                           # F06360
  IF (IBATCH.EQ.0) THEN                                       # F06370
    WRITE (*,*) 'ENTER:  NJ  NI'                                # F06380
    WRITE (*,*) '    NJ = NUMBER OF PARTICLES IN THE J DIRECTION FOR FA# F06390
1CE 6'                                                         # F06400
    WRITE (*,*) '    NI = NUMBER OF PARTICLES IN THE I DIRECTION FOR FA# F06410
1CE 6'                                                         # F06420
  END IF                                                         # F06430
  IF (IBATCH.EQ.0) THEN                                         # F06440
    READ (*,*) NJ6,NI6                                          # F06450
    WRITE(I9,*) NJ6,NI6                                          # F06460
  ELSE                                                           # F06470
    READ(I9,*) NJ6,NI6                                          # F06480
  END IF                                                         # F06490
END IF                                                         # F06500
END IF                                                         # F06510
DO 230 K=KMIN,KMAX                                           # F06520
DO 230 I=IMIN,IMAX                                           # F06530
DO 230 J=JMIN,JMAX                                           # F06540
  KKK=K                                                         # F06550
  IF (KWT.EQ.1) THEN                                           # F06560
    DO 210 KK=1,NLAY                                           # F06570
      KKK=KK                                                    # F06580
      IF (IBOUND(J,I,KKK).EQ.0) GO TO 210                      # F06590
      IF (HEAD(J,I,KKK).GE.1.00E+29) GO TO 210                 # F06600
      GO TO 220                                                  # F06610
210  CONTINUE                                                  # F06620
      GO TO 230                                                  # F06630
    END IF                                                       # F06640
    IF (IBOUND(J,I,KKK).EQ.0) GO TO 230                        # F06650
220  CONTINUE                                                  # F06660
    IF (ICHEAD.EQ.0.AND.IBOUND(J,I,KKK).LT.0) GO TO 230      # F06670
    IF (ILOC.EQ.0) THEN                                          # F06680
      CALL VOLDIV(J,I,KKK,JLC,ILC,KLC,XLC,YLC,ZLLC,NI,NJ,NK,NPART,N, # F06690
1IBATCH)                                                         # F06700
    ELSE                                                         # F06710
      IF (IF1.EQ.1) CALL FACDIV (J,I,KKK,JLC,ILC,KLC,XLC,YLC,ZLLC,1, # F06720
1NI1,NK1,NPART,N,IBATCH)                                         # F06730
      IF (IF2.EQ.1) CALL FACDIV (J,I,KKK,JLC,ILC,KLC,XLC,YLC,ZLLC,2, # F06740
1NI2,NK2,NPART,N,IBATCH)                                         # F06750
      IF (IF3.EQ.1) CALL FACDIV (J,I,KKK,JLC,ILC,KLC,XLC,YLC,ZLLC,3, # F06760
1NJ3,NK3,NPART,N,IBATCH)                                         # F06770
      IF (IF4.EQ.1) CALL FACDIV (J,I,KKK,JLC,ILC,KLC,XLC,YLC,ZLLC,4, # F06780
1NJ4,NK4,NPART,N,IBATCH)                                         # F06790
      IF (IF5.EQ.1) CALL FACDIV (J,I,KKK,JLC,ILC,KLC,XLC,YLC,ZLLC,5, # F06800
1NJ5,NI5,NPART,N,IBATCH)                                         # F06810
      IF (IF6.EQ.1) CALL FACDIV (J,I,KKK,JLC,ILC,KLC,XLC,YLC,ZLLC,6, # F06820
1NJ6,NI6,NPART,N,IBATCH)                                         # F06830
    END IF                                                         # F06840
230  CONTINUE                                                  # F06850
    IF (IBATCH.EQ.0) WRITE (*,5200) N,NPART                    # F06860

```

```

5200  FORMAT(' TOTAL NUMBER OF PARTICLES IS ',I6/                # F06870
1' MAXIMUM NUMBER OF PARTICLES IS ',I6)                        # F06880
MES= 'DO YOU WANT TO SPECIFY ANOTHER SUBREGION ?'             # F06890
CALL YESNO(IBATCH,I9,MES,IAN)                                  # F06900
IF(IANS.EQ.1) GO TO 200                                         # F06910
NPRT=N                                                          # F06920
DO 240 N=1,NPRT                                                 # F06930
IF(ILSTOR.EQ.1) WRITE(I6,5210) JLC(N),ILC(N),KLC(N),XLC(N),YLC(N), # F06940
1ZLLC(N)                                                         # F06950
5210  FORMAT(3I10,3F14.7)                                       # F06960
      JP=JLC(N)                                                  # F06970
      IP=ILC(N)                                                  # F06980
      XMN=0.0                                                    # F06990
      XMX=XMAX(JP)                                              # F07000
      IF(JP.GT.1) XMN= XMAX(JP-1)                               # F07010
      XLC(N)= (1.0-XLC(N))*XMN + XLC(N)*XMX                    # F07020
      YMN=0.0                                                    # F07030
      YMX=YMAX(IP)                                              # F07040
      IF(IP.LT.NROW) YMN=YMAX(IP+1)                             # F07050
      YLC(N)= (1.0-YLC(N))*YMN + YLC(N)*YMX                    # F07060
      WRITE(I5,5170) N,XLC(N),YLC(N),ZLLC(N),JLC(N),ILC(N),KLC(N) # F07070
240   CONTINUE                                                  # F07080
      IF(ILSTOR.EQ.1) CLOSE (I6)                                # F07090
      END IF                                                     # F07100
C                                                                    # F07110
C  CHANGE ZONE CODES IN THE IBOUND ARRAY                        # F07120
C                                                                    # F07130
      IF (IZSTOP.NE.0) THEN                                     # F07140
MES= 'DO YOU WANT TO CHANGE ANY OF THE ZONE CODES IN THE IBOUND AR# F07150
1RAY ?'                                                         # F07160
CALL YESNO (IBATCH,I9,MES,IAN)                                  # F07170
IF (IANS.EQ.1) THEN                                             # F07180
250  CONTINUE                                                  # F07190
      IF (IBATCH.EQ.0) THEN                                     # F07200
WRITE (*,*) 'WHAT TYPE OF CHANGE DO YOU WANT TO MAKE ?'       # F07210
WRITE (*,*) '  1 = CHANGE AN ENTIRE LAYER'                     # F07220
WRITE (*,*) '  2 = CHANGE AN INDIVIDUAL CELL'                 # F07230
WRITE (*,*) '  3 = CHANGE ALL CELLS IN A SUBREGION OF THE GRID' # F07240
      END IF                                                    # F07250
      IF(IBATCH.EQ.0) THEN                                     # F07260
READ (*,*) IANS                                                 # F07270
WRITE(I9,*) IANS                                                # F07280
      ELSE                                                       # F07290
READ(I9,*) IANS                                                 # F07300
      END IF                                                    # F07310
      IF(IANS.EQ.1) THEN                                         # F07320
IF (IBATCH.EQ.0) WRITE (*,*) 'ENTER THE LAYER NUMBER:'        # F07330
IF(IBATCH.EQ.0) THEN                                           # F07340
READ (*,*) NLAYER                                              # F07350
WRITE(I9,*) NLAYER                                             # F07360
      ELSE                                                       # F07370
READ(I9,*) NLAYER                                             # F07380
      END IF                                                    # F07390
      IF (IBATCH.EQ.0) WRITE (*,*) 'ENTER THE NEW ZONE CODE:'  # F07400
IF(IBATCH.EQ.0) THEN                                           # F07410
READ (*,*) NUM                                                 # F07420
WRITE(I9,*) NUM                                                # F07430
      ELSE                                                       # F07440
READ(I9,*) NUM                                                 # F07450
      END IF                                                    # F07460
DO 260 I=1,NROW                                                 # F07470
DO 260 J=1,NCOL                                                 # F07480
ITEMP=IBOUND(J,I,NLAYER)                                       # F07490
IF(ITEMP.NE.0) THEN                                             # F07500
IBOUND(J,I,NLAYER)=NUM                                         # F07510
IF(ITEMP.LT.0) IBOUND(J,I,NLAYER)= -IBOUND(J,I,NLAYER)      # F07520

```

	END IF	# F07530
260	CONTINUE	# F07540
	ELSE IF (IANS.EQ.2) THEN	# F07550
	IF (IBATCH.EQ.0)	# F07560
	1WRITE (*,*) 'ENTER THE CELL INDICES: J I K'	# F07570
	IF (IBATCH.EQ.0) THEN	# F07580
	READ (*,*) J,I,K	# F07590
	WRITE(I9,*) J,I,K	# F07600
	ELSE	# F07610
	READ(I9,*) J,I,K	# F07620
	END IF	# F07630
	IF (IBATCH.EQ.0) WRITE (*,*) 'ENTER THE NEW ZONE CODE:'	# F07640
	IF (IBATCH.EQ.0) THEN	# F07650
	READ (*,*) NUM	# F07660
	WRITE(I9,*) NUM	# F07670
	ELSE	# F07680
	READ(I9,*) NUM	# F07690
	END IF	# F07700
	ITEMP=IBOUND(J,I,K)	# F07710
	IF (ITEMP.NE.0) THEN	# F07720
	IBOUND(J,I,K)=NUM	# F07730
	IF (ITEMP.LT.0) IBOUND(J,I,K)= -IBOUND(J,I,K)	# F07740
	END IF	# F07750
	ELSE IF (IANS.EQ.3) THEN	# F07760
	IF (IBATCH.EQ.0) THEN	# F07770
	WRITE (*,*) 'DEFINE THE SUBREGION BY ENTERING THE MINIMUM AND MAXI	# F07780
	1MUM J,I,K COORDINATES.'	# F07790
	WRITE (*,*) ' ENTER: MINIMUM J, MAXIMUM J, MINIMUM I, MAXIMUM I,	# F07800
	1 MINIMUM K, MAXIMUM K'	# F07810
	END IF	# F07820
	IF (IBATCH.EQ.0) THEN	# F07830
	READ (*,*) JJ1,JJ2,II1,II2,KK1,KK2	# F07840
	WRITE(I9,*) JJ1,JJ2,II1,II2,KK1,KK2	# F07850
	ELSE	# F07860
	READ(I9,*) JJ1,JJ2,II1,II2,KK1,KK2	# F07870
	END IF	# F07880
	IF (JJ2.LT.1.OR.JJ2.GT.NCOL) JJ2=NCOL	# F07890
	IF (II2.LT.1.OR.II2.GT.NROW) II2=NROW	# F07900
	IF (KK2.LT.1.OR.KK2.GT.NLAY) KK2=NLAY	# F07910
	IF (IBATCH.EQ.0) WRITE (*,*) 'ENTER THE NEW ZONE CODE:'	# F07920
	IF (IBATCH.EQ.0) THEN	# F07930
	READ (*,*) NUM	# F07940
	WRITE(I9,*) NUM	# F07950
	ELSE	# F07960
	READ(I9,*) NUM	# F07970
	END IF	# F07980
	DO 270 K=KK1,KK2	# F07990
	DO 270 I=II1,II2	# F08000
	DO 270 J=JJ1,JJ2	# F08010
	ITEMP=IBOUND(J,I,K)	# F08020
	IF (ITEMP.NE.0) THEN	# F08030
	IBOUND(J,I,K)=NUM	# F08040
	IF (ITEMP.LT.0) IBOUND(J,I,K)= -IBOUND(J,I,K)	# F08050
	END IF	# F08060
270	CONTINUE	# F08070
	END IF	# F08080
	MES= 'DO YOU WANT TO CHANGE SOME MORE ZONE CODES ?'	# F08090
	CALL YESNO (IBATCH,I9,MES,IANS)	# F08100
	IF (IANS.EQ.1) GO TO 250	# F08110
	END IF	# F08120
	END IF	# F08130
C		# F08140
	RETURN	# F08150
280	FORMAT(A)	# F08160
	END	# F08170
C----	END OF ROUTINE-----	# F08180


```

C                                                    # G00010
C---Version 1.2   April 27, 1990                    # G00020
C*****                                                  # G00030
C                                                    # G00040
C                                VOLDIV                # G00050
C                                                    # G00060
C  THIS SUBROUTINE PLACES A THREE DIMENSIONAL ARRAY OF PARTICLES WITHIN # G00070
C  A FINITE DIFFERENCE CELL.                          # G00080
C                                                    # G00090
C*****                                                  # G00100
C                                                    # G00110
C      SUBROUTINE VOLDIV (J,I,K,JLC,ILC,KLC,XLC,YLC,ZLLC,NDIV,MDIV,LDIV, # G00120
1NPART,NP,IBATCH)                                     # G00130
      DIMENSION JLC(NPART),ILC(NPART),KLC(NPART),XLC(NPART),YLC(NPART), # G00140
      1ZLLC(NPART)                                     # G00150
C                                                    # G00160
C      DELN= 1.0E+0/FLOAT(NDIV)                       # G00170
      S1F= DELN/2.0E+0                                 # G00180
      DELM= 1.0E+0/FLOAT(MDIV)                       # G00190
      S2F= DELM/2.0E+0                                 # G00200
      DELL= 1.0E+0/FLOAT(LDIV)                       # G00210
      S3F= DELL/2.0E+0                                # G00220
      S1=S1F                                           # G00230
      S2=S2F                                           # G00240
      S3=S3F                                           # G00250
      DO 30 L=1,LDIV                                  # G00260
      S1=S1F                                           # G00270
      DO 20 N=1,NDIV                                  # G00280
      S2=S2F                                           # G00290
      DO 10 M=1,MDIV                                  # G00300
      NP=NP+1                                           # G00310
      IF(NP.GT.NPART) THEN                             # G00320
      IF (IBATCH.EQ.0)                                # G00330
      1WRITE (*,*) 'MAXIMUM NUMBER OF PARTICLES EXCEEDED. RUN STOPPED.' # G00340
      STOP                                              # G00350
      END IF                                           # G00360
      JLC(NP)=J                                         # G00370
      ILC(NP)=I                                         # G00380
      KLC(NP)=K                                         # G00390
      XLC(NP)=S2                                         # G00400
      YLC(NP)=S1                                         # G00410
      ZLLC(NP)=S3                                         # G00420
      S2=S2+DELM                                         # G00430
10      CONTINUE                                       # G00440
      S1=S1+DELN                                         # G00450
20      CONTINUE                                       # G00460
      S3=S3+DELL                                         # G00470
30      CONTINUE                                       # G00480
      RETURN                                           # G00490
      END                                              # G00500
C-----END OF ROUTINE-----                        # G00510
C                                                    # H00010
C---Version 1.2   April 27, 1990                    # H00020
C*****                                                  # H00030
C                                                    # H00040
C                                FACDIV                # H00050
C                                                    # H00060
C  THIS SUBROUTINE PLACES A TWO DIMENSIONAL ARRAY OF PARTICLES ON ONE # H00070
C  FACE OF A FINITE DIFFERENCE CELL.                  # H00080
C                                                    # H00090
C*****                                                  # H00100
C                                                    # H00110
C      SUBROUTINE FACDIV (J,I,K,JLC,ILC,KLC,XLC,YLC,ZLLC,IS,NDIV,      # H00120
1MDIV,NPART,NP,IBATCH)                                # H00130
      DIMENSION JLC(NPART),ILC(NPART),KLC(NPART),XLC(NPART),YLC(NPART), # H00140
      1ZLLC(NPART)                                     # H00150

```

```

C                                     # H00160
    DELN= 1.0E+0/FLOAT(NDIV)         # H00170
    S1F= DELN/2.0E+0                 # H00180
    DELM= 1.0E+0/FLOAT(MDIV)        # H00190
    S2F= DELM/2.0E+0                 # H00200
    S1=S1F                           # H00210
    S2=S2F                           # H00220
    DO 20 N=1,NDIV                   # H00230
    S2=S2F                           # H00240
    DO 10 M=1,MDIV                   # H00250
    NP=NP+1                           # H00260
    IF(NP.GT.NPART) THEN              # H00270
    IF (IBATCH.EQ.0)                  # H00280
1WRITE (*,*) 'MAXIMUM NUMBER OF PARTICLES EXCEEDED. RUN STOPPED.' # H00290
    STOP                             # H00300
    END IF                           # H00310
    JLC(NP)=J                         # H00320
    ILC(NP)=I                         # H00330
    KLC(NP)=K                         # H00340
    IF(IS.EQ.1) THEN                  # H00350
    XLC(NP)=0.0001                    # H00360
    YLC(NP)=S1                       # H00370
    ZLLC(NP)=S2                      # H00380
    ELSE IF(IS.EQ.2) THEN             # H00390
    XLC(NP)=0.9999                    # H00400
    YLC(NP)=S1                       # H00410
    ZLLC(NP)=S2                      # H00420
    ELSE IF(IS.EQ.3) THEN             # H00430
    XLC(NP)=S1                       # H00440
    YLC(NP)=0.0001                   # H00450
    ZLLC(NP)=S2                      # H00460
    ELSE IF(IS.EQ.4) THEN             # H00470
    XLC(NP)=S1                       # H00480
    YLC(NP)=0.9999                   # H00490
    ZLLC(NP)=S2                      # H00500
    ELSE IF(IS.EQ.5) THEN             # H00510
    XLC(NP)=S1                       # H00520
    YLC(NP)=S2                       # H00530
    ZLLC(NP)=0.0001                  # H00540
    ELSE IF(IS.EQ.6) THEN             # H00550
    XLC(NP)=S1                       # H00560
    YLC(NP)=S2                       # H00570
    ZLLC(NP)=0.9999                  # H00580
    END IF                           # H00590
    S2=S2+DELM                       # H00600
10  CONTINUE                         # H00610
    S1=S1+DELN                       # H00620
20  CONTINUE                         # H00630
    RETURN                           # H00640
    END                               # H00650
C-----END OF ROUTINE-----# H00660
C                                     # I00010
C---Version 1.2   April 27, 1990    # I00020
C*****# I00030
C                                     # I00040
C                                     # I00050
C                                     # I00060
C THIS SUBROUTINE WRITES A QUESTION TO THE SCREEN THAT REQUIRES A # I00070
C YES-NO RESPONSE. THE YES-NO RESPONSE IS CONVERTED TO AN INTEGER # I00080
C RESPONSE (0 = NO, 1 =YES)          # I00090
C                                     # I00100
C*****# I00110
C                                     # I00120
    SUBROUTINE YESNO(IBATCH,I9,MES,IAN) # I00130
    CHARACTER*79 MES                  # I00140
    CHARACTER*1 IN                     # I00150

```

```

10      CHARACTER*10 ANSWR                                # I00160
        CONTINUE                                         # I00170
        IF (IBATCH.EQ.0) THEN                             # I00180
        WRITE (*,5000) MES                               # I00190
        WRITE (*,*) '      [Y = YES;      N OR <CR> = NO] ' # I00200
        END IF                                           # I00210
        IF (IBATCH.EQ.0) THEN                             # I00220
        READ (*,'(A10)') ANSWR                          # I00230
        IF (ANSWR.EQ.' ') WRITE(I9,*) 'N'                # I00240
        IF (ANSWR.NE.' ') WRITE(I9,'(A10)') ANSWR        # I00250
        ELSE                                             # I00260
        READ(I9,'(A10)') ANSWR                           # I00270
        END IF                                           # I00280
5000    FORMAT(1X,A79)                                    # I00290
        IN=' '                                           # I00300
        DO 20 N=1,10                                     # I00310
        IF (ANSWR(N:N).NE.' ') THEN                     # I00320
        IN= ANSWR(N:N)                                   # I00330
        GO TO 30                                         # I00340
        END IF                                           # I00350
20      CONTINUE                                         # I00360
30      CONTINUE                                         # I00370
        IANS= -1                                         # I00380
        IF (IN.EQ.'Y'.OR.IN.EQ.'y') IANS=1              # I00390
        IF (IN.EQ.'N'.OR.IN.EQ.'n'.OR.IN.EQ.' ') IANS=0 # I00400
        IF (IANS.EQ.-1) GO TO 10                        # I00410
        RETURN                                           # I00420
        END                                              # I00430
C-----END OF ROUTINE-----# I00440
C                                                         # J00010
C---Version 1.2   April 27, 1990                         # J00020
C*****# J00030
C                                                         # J00040
C                                                         # J00050
C               CELFLO                                   # J00060
C THIS SUBROUTINE OVERSEES BUDGET DATA INPUT.           # J00070
C                                                         # J00080
C*****# J00090
C                                                         # J00100
        SUBROUTINE CELFLO (QX,QY,QZ,BUFF,IUNIT,IBOUND,HEAD,IBUFF,
1DELR,DELC,NCOL,NROW,NLAY,NCP1,NRP1,NLP1,NUNIT,QSS,I7,IBATCH,
2I9)                                                     # J00110
C                                                         # J00120
        DIMENSION DELC(NROW),DELR(NCOL),QX(NCP1,NROW,NLAY),
1QY(NCOL,NRP1,NLAY),QZ(NCOL,NROW,NLP1),BUFF(NCOL,NROW,NLAY),
2IUNIT(NUNIT),IBUFF(NCOL,NROW,NLAY),HEAD(NCOL,NROW,NLAY),
3IBOUND(NCOL,NROW,NLAY),QSS(NCOL,NROW,NLAY)             # J00130
        CHARACTER*16 TEXT                                # J00140
C                                                         # J00150
        IUBCF=IUNIT(7)                                   # J00160
C                                                         # J00170
C ZERO ARRAY CONTAINING NET FLOW RATE TO SOURCES & SINKS # J00180
C                                                         # J00190
        DO 1 K=1,NLAY                                    # J00200
        DO 1 I=1,NROW                                    # J00210
        DO 1 J=1,NCOL                                    # J00220
        QSS(J,I,K)=0.0                                   # J00230
1      QSS(J,I,K)=0.0                                     # J00240
C                                                         # J00250
C GENERATE PROCESSED FACE FLOW TERMS                     # J00260
C                                                         # J00270
        WRITE(I7,5500)                                   # J00280
5500    FORMAT(' -----' )                             # J00290
        1-----' )                                     # J00300
        WRITE(I7,*) 'FLOW RATES FROM THE BCF PACKAGE NOW BEING READ...' # J00310
C                                                         # J00320
C READ CONSTANT HEAD FLOWS                               # J00330

```

C	TEXT= ' CONSTANT HEAD'	# J00380
	CALL RDBUDG (BUFF,TEXT,NCOL,NROW,NLAY,IUBCF,I7)	# J00390
C		# J00400
C	POTENTIAL DISCHARGE CELLS ARE FLAGGED IN THE IBOUND ARRAY	# J00410
C	BY MULTIPLYING IBOUND VALUE BY 1000. THIS LOOP FLAGS CH	# J00420
C	CELLS THAT HAVE NET DISCHARGE.	# J00430
C		# J00440
	DO 10 K=1,NLAY	# J00450
	DO 10 I=1,NROW	# J00460
	DO 10 J=1,NCOL	# J00470
	IF(IBOUND(J,I,K).GE.0) GO TO 10	# J00480
	IF(BUFF(J,I,K).LE.0.0) IBOUND(J,I,K)=1000*IBOUND(J,I,K)	# J00490
10	CONTINUE	# J00500
C		# J00510
C	READ X FACE FLOWS	# J00520
C		# J00530
	IF(NCOL.GT.1) THEN	# J00540
	TEXT= 'FLOW RIGHT FACE '	# J00550
	CALL RDBUDG (BUFF,TEXT,NCOL,NROW,NLAY,IUBCF,I7)	# J00560
	ELSE	# J00570
	CALL ZERO (BUFF,NCOL,NROW,NLAY)	# J00580
	END IF	# J00590
	DO 20 K=1,NLAY	# J00600
	DO 20 I=1,NROW	# J00610
	QX(1,I,K)= 0.0	# J00620
	DO 20 J=1,NCOL	# J00630
20	QX(J+1,I,K)= BUFF(J,I,K)	# J00640
C		# J00650
C	READ Y FACE FLOWS	# J00660
C		# J00670
	IF(NROW.GT.1) THEN	# J00680
	TEXT= 'FLOW FRONT FACE '	# J00690
	CALL RDBUDG (BUFF,TEXT,NCOL,NROW,NLAY,IUBCF,I7)	# J00700
	ELSE	# J00710
	CALL ZERO (BUFF,NCOL,NROW,NLAY)	# J00720
	END IF	# J00730
	DO 30 K=1,NLAY	# J00740
	DO 30 J=1,NCOL	# J00750
	QY(J,1,K)= 0.0	# J00760
	DO 30 I=1,NROW	# J00770
30	QY(J,I+1,K)= -BUFF(J,I,K)	# J00780
C		# J00790
C	READ Z FACE FLOWS	# J00800
C		# J00810
	IF(NLAY.GT.1) THEN	# J00820
	TEXT= 'FLOW LOWER FACE '	# J00830
	CALL RDBUDG (BUFF,TEXT,NCOL,NROW,NLAY,IUBCF,I7)	# J00840
	ELSE	# J00850
	CALL ZERO (BUFF,NCOL,NROW,NLAY)	# J00860
	END IF	# J00870
	DO 40 I=1,NROW	# J00880
	DO 40 J=1,NCOL	# J00890
	QZ(J,I,1)= 0.0	# J00900
	DO 40 K=1,NLAY	# J00910
40	QZ(J,I,K+1)= -BUFF(J,I,K)	# J00920
	WRITE(I7,*) ' BCF FLOW RATES HAVE BEEN READ'	# J00930
C		# J00940
C	MODIFY FACE FLOW FOR BOUNDARY FLUXES AND SET IBOUND FLAGS	# J00950
C	FOR THOSE CELLS WITH DISTRIBUTED SINKS AND SOURCES	# J00960
C		# J00970
C	--- <td># J00980</td>	# J00980
	IF(IUNIT(2).NE.0) CALL WELLS (IUNIT(2),QX,QY,QZ,IBOUND,	# J00990
	1NCOL,NROW,NLAY,NCPL,NRPL,NLP1,QSS,DELR,DELC,I7,I9,IBATCH)	# J01000
C	--- <td># J01010</td>	# J01010
	IF(IUNIT(5).NE.0) CALL RDGHB (2,IUNIT(5),QX,QY,QZ,BUFF,IBOUND,	# J01020
		# J01030

```

        INCOL,NROW,NLAY,NCP1,NRP1,NLP1,QSS,IBUFF,DELR,DELC,I7)          # J01040
C---RECHARGE                                                              # J01050
        IF(IUNIT(1).NE.0) CALL RCHRG ( IUNIT(1),QZ,BUFF,IBUFF,HEAD,      # J01060
        1IBOUND,DELR,DELC,NCOL,NROW,NLP1,NLAY,QSS,I7,I9,IBATCH)         # J01070
C---ET                                                                    # J01080
        IF(IUNIT(3).NE.0) CALL EVAPO ( IUNIT(3),QZ,BUFF,IBUFF,NCOL,NROW,  # J01090
        1NLAY,NLP1,QSS,IBOUND,I7,I9,IBATCH)                             # J01100
C---RIVERS                                                                # J01110
        IF(IUNIT(4).NE.0) CALL RDGHB ( 1,IUNIT(4),QX,QY,QZ,BUFF,IBOUND,  # J01120
        1NCOL,NROW,NLAY,NCP1,NRP1,NLP1,QSS,IBUFF,DELR,DELC,I7)         # J01130
C---GENERAL HEAD BOUNDARIES                                              # J01140
        IF(IUNIT(6).NE.0) CALL RDGHB ( 3,IUNIT(6),QX,QY,QZ,BUFF,IBOUND,  # J01150
        1NCOL,NROW,NLAY,NCP1,NRP1,NLP1,QSS,IBUFF,DELR,DELC,I7)         # J01160
        WRITE(I7,*) ' '                                                  # J01170
        WRITE(I7,*) '**** FLOW RATES FROM BCF AND ALL STRESS PACKAGES HAVE# J01180
1 BEEN READ ****'                                                       # J01190
        WRITE(I7,*) '**** INTERCELL FLOW RATES CAN BE CHECKED USING THE CE# J01200
1LL-BY-CELL OPTION ****'                                                # J01210
        WRITE(I7,5500)                                                  # J01220
C                                                                        # J01230
        RETURN                                                            # J01240
        END                                                              # J01250
C-----END OF ROUTINE-----# J01260
C                                                                        # K00010
C---Version 1.2   April 27, 1990                                         # K00020
C*****# K00030
C                                                                        # K00040
C                                ZERO                                     # K00050
C                                                                        # K00060
C    THIS SUBROUTINE FILLS A THREE-DIMENSIONAL ARRAY WITH ZEROS.        # K00070
C                                                                        # K00080
C*****# K00090
C                                                                        # K00100
        SUBROUTINE ZERO(X,NJ,NI,NK)                                       # K00110
        DIMENSION X(NJ,NI,NK)                                           # K00120
        DO 10 K=1,NK                                                     # K00130
        DO 10 I=1,NI                                                     # K00140
        DO 10 J=1,NJ                                                     # K00150
10    X(J,I,K)=0.0                                                       # K00160
        RETURN                                                            # K00170
        END                                                              # K00180
C-----END OF ROUTINE-----# K00190
C                                                                        # L00010
C---Version 1.2   April 27, 1990                                         # L00020
C*****# L00030
C                                                                        # L00040
C                                RCHRG                                     # L00050
C                                                                        # L00060
C    THIS SUBROUTINE READS THE RECHARGE DATA FILE FROM THE USGS MODULAR  # L00070
C    FINITE-DIFFERENCE FLOW MODEL.                                       # L00080
C                                                                        # L00090
C*****# L00100
C                                                                        # L00110
        SUBROUTINE RCHRG (IU,QZ,RECH,IRCH,HEAD,IBOUND,DELR,DELC,        # L00120
        1NCOL,NROW,NLP1,NLAY,QSS,I7,I9,IBATCH)                         # L00130
C                                                                        # L00140
        DIMENSION QZ(NCOL,NROW,NLP1),RECH(NCOL,NROW,NLAY),IRCH(NCOL,NROW),# L00150
        1HEAD(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY),DELR(NCOL),        # L00160
        2DELC(NROW),QSS(NCOL,NROW,NLAY)                                  # L00170
C                                                                        # L00180
        CHARACTER*79 MES                                                 # L00190
C                                                                        # L00200
        WRITE(I7,5000)                                                  # L00210
5000  FORMAT(' -----# L00220
1-----')                                                              # L00230
        WRITE(I7,*) 'RECHARGE DATA NOW BEING READ...'                 # L00240

```

```

READ(IU,5010) NRCHOP,ID1,IFACE # L00250
IF (IFACE.EQ.0.AND.NLAY.GT.1) THEN # L00260
MES= 'DO YOU WANT RECHARGE ASSIGNED TO THE TOP FACE OF CELLS ?' # L00270
CALL YESNO (IBATCH,I9,MES,IAN$) # L00280
IF (IAN$EQ.1) IFACE=1 # L00290
END IF # L00300
5010 READ(IU,5010) ID1,ID2 # L00310
FORMAT(3I10) # L00320
WRITE(I7,*) 'RECHARGE ARRAY NOW BEING READ...' # L00330
CALL IN2DR (IU,RECH(1,1,1),NCOL,NROW,I7) # L00340
IF(NRCHOP.EQ.2) THEN # L00350
WRITE(I7,*) 'RECHARGE LAYER ARRAY NOW BEING READ...' # L00360
CALL IN2DI (IU,IRCH,NCOL,NROW,I7) # L00370
END IF # L00380
IF(NRCHOP.EQ.1) THEN # L00390
DO 10 I=1,NROW # L00400
DO 10 J=1,NCOL # L00410
IF(IBOUND(J,I,1).LE.0.OR.HEAD(J,I,1).GT.1.0E+29) GO TO 10 # L00420
IF(IFACE.GT.0) THEN # L00430
QZ(J,I,1)= QZ(J,I,1) - RECH(J,I,1)*DELR(J)*DELC(I) # L00440
ELSE # L00450
QSS(J,I,1)=QSS(J,I,1) + RECH(J,I,1)*DELR(J)*DELC(I) # L00460
END IF # L00470
10 CONTINUE # L00480
END IF # L00490
IF(NRCHOP.EQ.2) THEN # L00500
DO 20 I=1,NROW # L00510
DO 20 J=1,NCOL # L00520
K= IRCH(J,I) # L00530
IF(IBOUND(J,I,K).LE.0.OR.HEAD(J,I,K).GT.1.0E+29) GO TO 20 # L00540
IF(IFACE.GT.0) THEN # L00550
CALL FACTYP (K,1,J,I,K-1,NBD,IBOUND,NCOL,NROW,NLAY,J,I,K,6,5,I7) # L00560
IF (NBD.EQ.1) THEN # L00570
QZ(J,I,K)= QZ(J,I,K) - RECH(J,I,1)*DELR(J)*DELC(I) # L00580
ELSE # L00590
QSS(J,I,K)=QSS(J,I,K) + RECH(J,I,1)*DELR(J)*DELC(I) # L00600
END IF # L00610
ELSE # L00620
QSS(J,I,K)=QSS(J,I,K) + RECH(J,I,1)*DELR(J)*DELC(I) # L00630
END IF # L00640
20 CONTINUE # L00650
END IF # L00660
IF(NRCHOP.EQ.3) THEN # L00670
DO 50 I=1,NROW # L00680
DO 50 J=1,NCOL # L00690
DO 30 K=1,NLAY # L00700
KK=K # L00710
IF(IBOUND(J,I,K).LT.0) GO TO 50 # L00720
IF(IBOUND(J,I,K).EQ.0.OR.HEAD(J,I,K).GT.1.0E+29) GO TO 30 # L00730
GO TO 40 # L00740
30 CONTINUE # L00750
GO TO 50 # L00760
40 CONTINUE # L00770
IF(IFACE.GT.0) THEN # L00780
QZ(J,I,KK)= QZ(J,I,KK) - RECH(J,I,1)*DELC(I)*DELR(J) # L00790
ELSE # L00800
QSS(J,I,KK)=QSS(J,I,KK) + RECH(J,I,1)*DELC(I)*DELR(J) # L00810
END IF # L00820
50 CONTINUE # L00830
END IF # L00840
C # L00850
RETURN # L00860
END # L00870
C-----END OF ROUTINE----- # L00880
C # M00010
C---Version 1.2 April 27, 1990 # M00020

```

```

C*****# M00030
C# M00040
C# M00050
C# M00060
C THIS SUBROUTINE READS THE WELL PACKAGE DATA FILE FROM THE USGS# M00070
C MODULAR FINITE-DIFFERENCE FLOW MODEL.# M00080
C# M00090
C*****# M00100
C# M00110
C SUBROUTINE WELLS (IU,QX,QY,QZ,IBOUND,NCOL,NROW,NLAY,# M00120
1NCP1,NRP1,NLP1,QSS,DELR,DELC,I7,I9,IBATCH)# M00130
C# M00140
C DIMENSION QX(NCP1,NROW,NLAY),QY(NCOL,NRP1,NLAY),# M00150
1QZ(NCOL,NROW,NLP1),IBOUND(NCOL,NROW,NLAY),# M00160
2QSS(NCOL,NROW,NLAY),DELR(NCOL),DELC(NROW)# M00170
C CHARACTER*79 MES# M00180
C# M00190
C WRITE(I7,5000)# M00200
5000 FORMAT(' -----')# M00210
1-----')# M00220
C WRITE(I7,*) 'WELL DATA NOW BEING READ...'# M00230
C READ(IU,5010) ID1,ID2# M00240
C READ(IU,5010) ITMP# M00250
5010 FORMAT (2I10)# M00260
C IF(ITMP.LE.0) RETURN# M00270
C IERR=0# M00280
C DO 10 N=1,ITMP# M00290
C READ(IU,5050) K,I,J,Q,IFACE# M00300
C IB=IBOUND(J,I,K)# M00310
C IF (IB.LE.0) THEN# M00320
C IF (IB.EQ.0) WRITE (I7,5020) J,I,K# M00330
C IF (IB.LT.0) WRITE (I7,5030) J,I,K# M00340
5020 FORMAT (' A WELL WAS SPECIFIED FOR INACTIVE CELL (J,I,K) = (' ,I3,# M00350
1',' ,I3',' ,I3,')')# M00360
5030 FORMAT (' A WELL WAS SPECIFIED FOR CONSTANT HEAD CELL (J,I,K) = (' ,# M00370
1,I3',' ,I3',' ,I3,')')# M00380
C IERR=IERR+1# M00390
C END IF# M00400
C IF (IFACE.LT.0) THEN# M00410
C CALL FLUXBC (Q,QSS,QX,QY,IBOUND,DELR,DELC,J,I,K,NCOL,NROW,NLAY,# M00420
1 NCP1,NRP1)# M00430
C ELSE IF(IFACE.EQ.0.OR.IFACE.GT.6) THEN# M00440
C IF(Q.LE.0.0.AND.ABS(IB).LT.1000) IBOUND(J,I,K)= 1000*IB# M00450
C QSS(J,I,K)=QSS(J,I,K) + Q# M00460
C ELSE IF(IFACE.EQ.1) THEN# M00470
C CALL FACTYP (J,1,J-1,I,K,NBD,IBOUND,NCOL,NROW,NLAY,J,I,K,IFACE,4,# M00480
1 I7)# M00490
C IF (NBD.EQ.1) THEN# M00500
C QX(J,I,K)= QX(J,I,K) + Q# M00510
C ELSE# M00520
C IF(Q.LE.0.0.AND.ABS(IB).LT.1000) IBOUND(J,I,K)= 1000*IB# M00530
C QSS(J,I,K)=QSS(J,I,K) + Q# M00540
C END IF# M00550
C ELSE IF(IFACE.EQ.2) THEN# M00560
C CALL FACTYP (J,NCOL,J+1,I,K,NBD,IBOUND,NCOL,NROW,NLAY,J,I,K,IFACE,# M00570
1 4,I7)# M00580
C IF (NBD.EQ.1) THEN# M00590
C QX(J+1,I,K)= QX(J+1,I,K) - Q# M00600
C ELSE# M00610
C IF(Q.LE.0.0.AND.ABS(IB).LT.1000) IBOUND(J,I,K)= 1000*IB# M00620
C QSS(J,I,K)=QSS(J,I,K) + Q# M00630
C END IF# M00640
C ELSE IF(IFACE.EQ.3) THEN# M00650
C CALL FACTYP (I,NROW,J,I+1,K,NBD,IBOUND,NCOL,NROW,NLAY,J,I,K,IFACE,# M00660
1 4,I7)# M00670
C IF (NBD.EQ.1) THEN# M00680

```

```

      QY(J,I+1,K)= QY(J,I+1,K) + Q      # M00690
      ELSE                                # M00700
      IF(Q.LE.0.0.AND.ABS(IB).LT.1000) IBOUND(J,I,K)= 1000*IB      # M00710
      QSS(J,I,K)=QSS(J,I,K) + Q          # M00720
      END IF                              # M00730
      ELSE IF(IFACE.EQ.4) THEN            # M00740
      CALL FACTYP (I,1,J,I-1,K,NBD,IBOUND,NCOL,NROW,NLAY,J,I,K,IFACE,4,      # M00750
1 I7)                                     # M00760
      IF (NBD.EQ.1) THEN                  # M00770
      QY(J,I,K)= QY(J,I,K) - Q           # M00780
      ELSE                                # M00790
      IF(Q.LE.0.0.AND.ABS(IB).LT.1000) IBOUND(J,I,K)= 1000*IB      # M00800
      QSS(J,I,K)=QSS(J,I,K) + Q          # M00810
      END IF                              # M00820
      ELSE IF(IFACE.EQ.5) THEN            # M00830
      CALL FACTYP (K,NLAY,J,I,K+1,NBD,IBOUND,NCOL,NROW,NLAY,J,I,K,IFACE,      # M00840
1 4,I7)                                   # M00850
      IF (NBD.EQ.1) THEN                  # M00860
      QZ(J,I,K+1)= QZ(J,I,K+1) + Q       # M00870
      ELSE                                # M00880
      IF(Q.LE.0.0.AND.ABS(IB).LT.1000) IBOUND(J,I,K)= 1000*IB      # M00890
      QSS(J,I,K)=QSS(J,I,K) + Q          # M00900
      END IF                              # M00910
      ELSE IF(IFACE.EQ.6) THEN            # M00920
      CALL FACTYP (K,1,J,I,K-1,NBD,IBOUND,NCOL,NROW,NLAY,J,I,K,IFACE,4,      # M00930
1 I7)                                     # M00940
      IF (NBD.EQ.1) THEN                  # M00950
      QZ(J,I,K)= QZ(J,I,K) - Q           # M00960
      ELSE                                # M00970
      IF(Q.LE.0.0.AND.ABS(IB).LT.1000) IBOUND(J,I,K)= 1000*IB      # M00980
      QSS(J,I,K)=QSS(J,I,K) + Q          # M00990
      END IF                              # M01000
      END IF                              # M01010
10    CONTINUE                           # M01020
      WRITE(I7,*) ' WELL DATA HAS BEEN READ'      # M01030
      IF (IERR.GT.0) THEN                  # M01040
      IF (IBATCH.EQ.0) THEN                # M01050
      WRITE (*,5040) IERR                  # M01060
5040  FORMAT(' WELLS WERE SPECIFIED FOR ',I4,' INACTIVE OR CONSTANT HEAD# M01070
1 CELLS')
      WRITE (*,*) 'A LIST IS PROVIDED IN THE "SUMMARY.PTH" FILE'      # M01090
      END IF                              # M01100
      MES= 'DO YOU WANT TO CONTINUE ?'      # M01110
      CALL YESNO (IBATCH,I9,MES,IAN)        # M01120
      IF(IANS.EQ.0) STOP                    # M01130
      END IF                              # M01140
      RETURN                              # M01150
5050  FORMAT(3I10,F10.0,I10)              # M01160
      END                                  # M01170
C-----END OF ROUTINE-----# M01180
C                                         # N00010
C---Version 1.2   April 27, 1990        # N00020
C*****# N00030
C                                         # N00040
C                                         # N00050
C                                         # N00060
C THIS SUBROUTINE READS THE EVAPOTRANSPIRATION DATA FILE FOR THE USGS      # N00070
C MODULAR FINITE-DIFFERENCE FLOW MODEL. IT ALSO CALLS A SUBROUTINE          # N00080
C TO READ THE BUDGET FILE FOR ET.      # N00090
C                                         # N00100
C*****# N00110
C                                         # N00120
C SUBROUTINE EVAPO (IU,QZ,BUFF,IEVT,NCOL,NROW,NLAY,NLP1,QSS,IBOUND,      # N00130
1 I7,I9,IBATCH)                          # N00140
C                                         # N00150
C DIMENSION QZ(NCOL,NROW,NLP1),BUFF(NCOL,NROW,NLAY),      # N00160

```



```

1 IEVT(NCOL,NROW),QSS(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY)      # N00170
  CHARACTER*16 LABEL                                              # N00180
C                                                                    # N00190
  CHARACTER*79 MES                                              # N00200
C                                                                    # N00210
  LABEL= '                  ET'                                # N00220
  WRITE(I7,5000)                                                # N00230
5000 FORMAT(' -----' )                                         # N00240
1-----' )                                                    # N00250
  WRITE(I7,*) 'EVAPOTRANSPIRATION DATA NOW BEING READ...'      # N00260
  READ(IU,5010) NEVTOP,IEVTCB,IFACE                              # N00270
  IF (IFACE.EQ.0) THEN                                           # N00280
    MES= 'DO YOU WANT TO ASSIGN EVAPOTRANSPIRATION TO THE TOP FACE OF' # N00290
    1 CELLS ?'                                                  # N00300
    CALL YESNO (IBATCH,I9,MES,IAN)                               # N00310
    IF (IAN.EQ.1) IFACE=1                                         # N00320
    END IF                                                       # N00330
    READ(IU,5010) INSURF,INEVTR,INEXDP,INIEVT                   # N00340
    IF (INSURF.GE.0) CALL IN2DR (IU,BUFF,NCOL,NROW,I7)           # N00350
    IF (INEVTR.GE.0) CALL IN2DR (IU,BUFF,NCOL,NROW,I7)           # N00360
    IF (INEXDP.GE.0) CALL IN2DR (IU,BUFF,NCOL,NROW,I7)           # N00370
5010  FORMAT (4I10)                                              # N00380
  WRITE(I7,*) 'ET RATES NOW BEING READ...'                      # N00390
C---READ FLOW ET FLOW RATES INTO BUFFER                          # N00400
  CALL RDBUDG (BUFF,LABEL,NCOL,NROW,NLAY,IEVTCB,I7)             # N00410
C                                                                    # N00420
  WRITE(I7,*) ' ET RATES HAVE BEEN READ'                        # N00430
  IF(NEVTOP.EQ.2) THEN                                           # N00440
    WRITE(I7,*) 'ET LAYER NUMBER ARRAY NOW BEING READ...'      # N00450
    CALL IN2DI (IU,IEVT,NCOL,NROW,I7)                           # N00460
    DO 10 I=1,NROW                                               # N00470
    DO 10 J=1,NCOL                                               # N00480
      K=IEVT(J,I)                                                 # N00490
      IF(IFACE.GT.0) THEN                                          # N00500
        CALL FACTYP (K,1,J,I,K-1,NBD,IBOUND,NCOL,NROW,NLAY,J,I,K,6,6,I7) # N00510
        IF (NBD.EQ.1) THEN                                        # N00520
          QZ(J,I,K)= QZ(J,I,K) - BUFF(J,I,K)                    # N00530
        ELSE                                                       # N00540
          QSS(J,I,K)=QSS(J,I,K) + BUFF(J,I,K)                    # N00550
          IF(BUFF(J,I,K).LE.0.0.AND.IBOUND(J,I,K).LT.1000)       # N00560
            1 IBOUND(J,I,K)= 1000*IBOUND(J,I,K)                 # N00570
        END IF                                                    # N00580
      ELSE                                                         # N00590
        QSS(J,I,K)=QSS(J,I,K) + BUFF(J,I,K)                     # N00600
        IF(BUFF(J,I,K).LE.0.0.AND.IBOUND(J,I,K).LT.1000)       # N00610
          1 IBOUND(J,I,K)= 1000*IBOUND(J,I,K)                 # N00620
        END IF                                                    # N00630
    CONTINUE                                                      # N00640
    ELSE                                                           # N00650
      DO 20 I=1,NROW                                              # N00660
      DO 20 J=1,NCOL                                              # N00670
        IF(IFACE.GT.0) THEN                                        # N00680
          QZ(J,I,1)= QZ(J,I,1) - BUFF(J,I,1)                    # N00690
        ELSE                                                       # N00700
          QSS(J,I,1)=QSS(J,I,1) + BUFF(J,I,1)                    # N00710
          IF(BUFF(J,I,1).LE.0.0.AND.IBOUND(J,I,1).LT.1000)       # N00720
            1 IBOUND(J,I,1)= 1000*IBOUND(J,I,1)                 # N00730
        END IF                                                    # N00740
    CONTINUE                                                      # N00750
    END IF                                                        # N00760
    RETURN                                                         # N00770
    END                                                           # N00780
C-----END OF ROUTINE-----# N00790

C                                                                    # O00010
C---Version 1.2   April 27, 1990                                # O00020

```

```

C*****# 000030
C# 000040
C          RDGHB# 000050
C# 000060
C THIS SUBROUTINE READS THE DATA FILES FOR THE RIVER, DRAIN, AND# 000070
C GENERAL HEAD BOUNDARY PACKAGES FOR THE USGS MODULAR FINITE-DIFFERENCE# 000080
C FLOW MODEL. IT ALSO READS THE BUDGET FILES FOR THOSE STRESS PACKAGES.# 000090
C# 000100
C*****# 000110
C# 000120
C      SUBROUTINE RDGHB (IPK,IU,QX,QY,QZ,BUFF,IBOUND,NCOL,NROW,NLAY,# 000130
      1NCP1,NRP1,NLP1,QSS,IBUFF,DELR,DELC,I7)# 000140
C# 000150
C      DIMENSION QX(NCP1,NROW,NLAY),QY(NCOL,NRP1,NLAY),# 000160
      1QZ(NCOL,NROW,NLP1),IBOUND(NCOL,NROW,NLAY),BUFF(NCOL,NROW,NLAY),# 000170
      2QSS(NCOL,NROW,NLAY),IBUFF(NCOL,NROW,NLAY),DELR(NCOL),DELC(NROW)# 000180
      CHARACTER*16 LABEL# 000190
C# 000200
      WRITE(I7,5000)# 000210
5000  FORMAT(' -----')# 000220
      1-----'# 000230
      IF(IPK.EQ.1) THEN# 000240
      WRITE(I7,*) 'RIVER DATA NOW BEING READ...'# 000250
      LABEL= ' RIVER LEAKAGE'# 000260
      ELSE IF(IPK.EQ.2) THEN# 000270
      WRITE(I7,*) 'DRAIN DATA NOW BEING READ...'# 000280
      LABEL= ' DRAINS'# 000290
      ELSE IF(IPK.EQ.3) THEN# 000300
      WRITE(I7,*) 'GENERAL HEAD BOUNDARY DATA NOW BEING READ...'# 000310
      LABEL= ' HEAD DEP BOUNDS'# 000320
      END IF# 000330
      READ(IU,5010) MX,ICB# 000340
      READ(IU,5010) ITMP# 000350
5010  FORMAT(2I10)# 000360
      IF(ITMP.LE.0) RETURN# 000370
      DO 10 K=1,NLAY# 000380
      DO 10 I=1,NROW# 000390
      DO 10 J=1,NCOL# 000400
10    IBUFF(J,I,K)=0# 000410
      WRITE(I7,*) ' FLOW RATES NOW BEING READ...'# 000420
      CALL RDBUDG (BUFF,LABEL,NCOL,NROW,NLAY,ICB,I7)# 000430
      WRITE(I7,*) ' FLOW RATES HAVE BEEN READ'# 000440
      DO 20 N=1,ITMP# 000450
      IF(IPK.EQ.1) THEN# 000460
      READ(IU,5020) K,I,J,H,C,E,IFACE# 000470
5020  FORMAT(3I10,3F10.0,I10)# 000480
      IF (IBOUND(J,I,K).EQ.0) WRITE (I7,5030) J,I,K# 000490
5030  FORMAT(' A RIVER WAS SPECIFIED FOR INACTIVE CELL (J,I,K) =',# 000500
      12(I4,', '),I4)# 000510
      ELSE# 000520
      READ(IU,5050) K,I,J,H,C,IFACE# 000530
      IF (IBOUND(J,I,K).EQ.0) WRITE (I7,5040) J,I,K# 000540
5040  FORMAT(' A DRAIN OR GHF WAS SPECIFIED FOR INACTIVE CELL (J,I,K) =',# 000550
      12(I4,', '),I4)# 000560
5050  FORMAT(3I10,2F10.0,I10)# 000570
      END IF# 000580
      IF(IBUFF(J,I,K).EQ.1) GO TO 20# 000590
      IBUFF(J,I,K)=1# 000600
      IB=IBOUND(J,I,K)# 000610
      IF (IFACE.LT.0) THEN# 000620
      CALL FLUXBC (BUFF(J,I,K),QSS,QX,QY,IBOUND,DELR,DELC,J,I,K,# 000630
      1 NCOL,NROW,NLAY,NCP1,NRP1)# 000640
      ELSE IF(IFACE.EQ.0.OR.IFACE.GT.6) THEN# 000650
      IF(BUFF(J,I,K).LE.0.0.AND.ABS(IB).LT.1000)IBOUND(J,I,K)=1000*IB# 000660
      QSS(J,I,K)=QSS(J,I,K) + BUFF(J,I,K)# 000670
      ELSE IF(IFACE.EQ.1) THEN# 000680

```

```

JM1=J-1                                # 000690
CALL FACTYP (J,1,J-1,I,K,NBD,IBOUND,NCOL,NROW,NLAY,J,I,K,IFACE, # 000700
1 IPK,I7)                                # 000710
IF (NBD.EQ.1) THEN                      # 000720
QX(J,I,K)= QX(J,I,K) + BUFF(J,I,K)      # 000730
ELSE                                     # 000740
IF(BUFF(J,I,K).LE.0.0.AND.ABS(IB).LT.1000)IBOUND(J,I,K)= 1000*IB # 000750
QSS(J,I,K)=QSS(J,I,K) + BUFF(J,I,K)     # 000760
END IF                                  # 000770
ELSE IF(IFACE.EQ.2) THEN                 # 000780
CALL FACTYP (J,NCOL,J+1,I,K,NBD,IBOUND,NCOL,NROW,NLAY,J,I,K,IFACE, # 000790
1 IPK,I7)                                # 000800
IF (NBD.EQ.1) THEN                      # 000810
QX(J+1,I,K)= QX(J+1,I,K) - BUFF(J,I,K)  # 000820
ELSE                                     # 000830
IF(BUFF(J,I,K).LE.0.0.AND.ABS(IB).LT.1000)IBOUND(J,I,K)= 1000*IB # 000840
QSS(J,I,K)=QSS(J,I,K) + BUFF(J,I,K)     # 000850
END IF                                  # 000860
ELSE IF(IFACE.EQ.3) THEN                 # 000870
CALL FACTYP (I,NROW,J,I+1,K,NBD,IBOUND,NCOL,NROW,NLAY,J,I,K,IFACE, # 000880
1 IPK,I7)                                # 000890
IF (NBD.EQ.1) THEN                      # 000900
QY(J,I+1,K)= QY(J,I+1,K) + BUFF(J,I,K)  # 000910
ELSE                                     # 000920
IF(BUFF(J,I,K).LE.0.0.AND.ABS(IB).LT.1000)IBOUND(J,I,K)= 1000*IB # 000930
QSS(J,I,K)=QSS(J,I,K) + BUFF(J,I,K)     # 000940
END IF                                  # 000950
ELSE IF(IFACE.EQ.4) THEN                 # 000960
CALL FACTYP (I,1,J,I-1,K,NBD,IBOUND,NCOL,NROW,NLAY,J,I,K,IFACE, # 000970
1 IPK,I7)                                # 000980
IF (NBD.EQ.1) THEN                      # 000990
QY(J,I,K)= QY(J,I,K) - BUFF(J,I,K)      # 001000
ELSE                                     # 001010
IF(BUFF(J,I,K).LE.0.0.AND.ABS(IB).LT.1000)IBOUND(J,I,K)= 1000*IB # 001020
QSS(J,I,K)=QSS(J,I,K) + BUFF(J,I,K)     # 001030
END IF                                  # 001040
ELSE IF(IFACE.EQ.5) THEN                 # 001050
CALL FACTYP (K,NLAY,J,I,K+1,NBD,IBOUND,NCOL,NROW,NLAY,J,I,K,IFACE, # 001060
1 IPK,I7)                                # 001070
IF (NBD.EQ.1) THEN                      # 001080
QZ(J,I,K+1)= QZ(J,I,K+1) + BUFF(J,I,K)  # 001090
ELSE                                     # 001100
IF(BUFF(J,I,K).LE.0.0.AND.ABS(IB).LT.1000)IBOUND(J,I,K)= 1000*IB # 001110
QSS(J,I,K)=QSS(J,I,K) + BUFF(J,I,K)     # 001120
END IF                                  # 001130
ELSE IF(IFACE.EQ.6) THEN                 # 001140
CALL FACTYP (K,1,J,I,K-1,NBD,IBOUND,NCOL,NROW,NLAY,J,I,K,IFACE, # 001150
1 IPK,I7)                                # 001160
IF (NBD.EQ.1) THEN                      # 001170
QZ(J,I,K)= QZ(J,I,K) - BUFF(J,I,K)      # 001180
ELSE                                     # 001190
IF(BUFF(J,I,K).LE.0.0.AND.ABS(IB).LT.1000)IBOUND(J,I,K)= 1000*IB # 001200
QSS(J,I,K)=QSS(J,I,K) + BUFF(J,I,K)     # 001210
END IF                                  # 001220
END IF                                  # 001230
20 CONTINUE                             # 001240
IF(IPK.EQ.1) THEN                       # 001250
WRITE(I7,*) ' RIVER DATA HAS BEEN READ' # 001260
ELSE IF(IPK.EQ.2) THEN                  # 001270
WRITE(I7,*) ' DRAIN DATA HAS BEEN READ' # 001280
ELSE IF(IPK.EQ.3) THEN                  # 001290
WRITE(I7,*) ' GENERAL HEAD BOUNDARY DATA HAS BEEN READ' # 001300
END IF                                  # 001310
RETURN                                  # 001320
END                                      # 001330
C-----END OF ROUTINE-----# 001340

```

```

C                                                    # P00010
C---Version 1.2   April 27, 1990                    # P00020
C*****                                                  # P00030
C                                                    # P00040
C                                                    # P00050
C                                                    # P00060
C   THIS SUBROUTINE READS A TWO DIMENSIONAL ARRAY OF REAL NUMBERS.  # P00070
C                                                    # P00080
C*****                                                  # P00090
C                                                    # P00100
C      SUBROUTINE IN2DR (IU,X,JJ,II,I7)              # P00110
C      DIMENSION X(JJ,II)                            # P00120
C      CHARACTER*20 FMT                              # P00130
C                                                    # P00140
C      READ(IU,5000) LOCAT,CNSTNT,FMT,IPRN           # P00150
5000  FORMAT(I10,F10.0,A20,I10)                      # P00160
C      IF(LOCAT.LE.0) THEN                            # P00170
C      DO 10 I=1,II                                   # P00180
C      DO 10 J=1,JJ                                   # P00190
10    X(J,I)= CNSTNT                                # P00200
C      WRITE(I7,5010) CNSTNT                          # P00210
5010  FORMAT('  CONSTANT VALUE OF',E13.5)            # P00220
C      RETURN                                          # P00230
C      END IF                                          # P00240
C                                                    # P00250
C      IF(FMT.EQ.'                                     ' ) THEN # P00260
C      DO 20 I=1,II                                   # P00270
C      READ(LOCAT,*) (X(J,I),J=1,JJ)                  # P00280
20    CONTINUE                                       # P00290
C      ELSE                                           # P00300
C      DO 30 I=1,II                                   # P00310
C      READ(LOCAT,FMT) (X(J,I),J=1,JJ)                 # P00320
30    CONTINUE                                       # P00330
C      END IF                                          # P00340
C      IF (CNSTNT.GT.0.0.OR.CNSTNT.LT.0.0) THEN      # P00350
C      DO 40 I=1,II                                   # P00360
C      DO 40 J=1,JJ                                   # P00370
40    X(J,I)= CNSTNT*X(J,I)                         # P00380
C      END IF                                          # P00390
C      IF(IPRN.GT.0) THEN                             # P00400
C      DO 50 I=1,II                                   # P00410
C      WRITE(I7,5020) (X(J,I),J=1,JJ)                 # P00420
50    CONTINUE                                       # P00430
C      ELSE                                           # P00440
C      WRITE(I7,*) '  DATA WAS READ BUT NOT PRINTED' # P00450
C      END IF                                          # P00460
5020  FORMAT(10E13.5)                               # P00470
C                                                    # P00480
C      RETURN                                          # P00490
C      END                                            # P00500
C-----END OF ROUTINE-----                          # P00510
C                                                    # Q00010
C---Version 1.2   April 27, 1990                    # Q00020
C*****                                                  # Q00030
C                                                    # Q00040
C                                                    # Q00050
C                                                    # Q00060
C   THIS SUBROUTINE READS A ONE DIMENSIONAL ARRAY OF REAL NUMBERS. # Q00070
C                                                    # Q00080
C*****                                                  # Q00090
C                                                    # Q00100
C      SUBROUTINE IN1DR (IU,X,JJ,I7)                 # Q00110
C      DIMENSION X(JJ)                               # Q00120
C      CHARACTER*20 FMT                              # Q00130
C                                                    # Q00140
C      READ(IU,5000) LOCAT,CNSTNT,FMT,IPRN           # Q00150

```

5000	FORMAT(I10,F10.0,A20,I10)	# Q00160
	IF(LOCAT.LE.0) THEN	# Q00170
	DO 10 J=1,JJ	# Q00180
10	X(J)= CNSTNT	# Q00190
	WRITE(I7,5010) CNSTNT	# Q00200
5010	FORMAT(' CONSTANT VALUE OF',E13.5)	# Q00210
	RETURN	# Q00220
	END IF	# Q00230
C		# Q00240
	IF(FMT.EQ.' ') THEN	# Q00250
	READ(LOCAT,*) (X(J),J=1,JJ)	# Q00260
	ELSE	# Q00270
	READ(LOCAT,FMT) (X(J),J=1,JJ)	# Q00280
	END IF	# Q00290
	IF (CNSTNT.GT.0.0.OR.CNSTNT.LT.0.0) THEN	# Q00300
	DO 20 J=1,JJ	# Q00310
20	X(J)= CNSTNT*X(J)	# Q00320
	END IF	# Q00330
C		# Q00340
	IF(IPRN.GT.0) THEN	# Q00350
	WRITE(I7,5020) (X(J),J=1,JJ)	# Q00360
5020	FORMAT(10E13.5)	# Q00370
	ELSE	# Q00380
	WRITE(I7,*) ' DATA WAS READ BUT NOT PRINTED'	# Q00390
	END IF	# Q00400
	RETURN	# Q00410
	END	# Q00420
C-----	END OF ROUTINE-----	# Q00430
C		# R00010
C---Version 1.2	April 27, 1990	# R00020
C*****		# R00030
C		# R00040
C	IN2DI	# R00050
C		# R00060
C	THIS SUBROUTINE READS A TWO DIMENSIONAL ARRAY OF INTEGERS.	# R00070
C		# R00080
C*****		# R00090
C		# R00100
	SUBROUTINE IN2DI (IU,N,JJ,II,I7)	# R00110
	DIMENSION N(JJ,II)	# R00120
	CHARACTER*20 FMT	# R00130
C		# R00140
	READ(IU,5000) LOCAT,ICON,FMT,IPRN	# R00150
5000	FORMAT(2I10,A20,I10)	# R00160
	IF(LOCAT.LE.0) THEN	# R00170
	DO 10 I=1,II	# R00180
	DO 10 J=1,JJ	# R00190
10	N(J,I)= ICON	# R00200
	WRITE(I7,5010) ICON	# R00210
5010	FORMAT(' CONSTANT VALUE OF',I10)	# R00220
	RETURN	# R00230
	END IF	# R00240
C		# R00250
	IF(FMT.EQ.' ') THEN	# R00260
	DO 20 I=1,II	# R00270
	READ(LOCAT,*) (N(J,I),J=1,JJ)	# R00280
20	CONTINUE	# R00290
	ELSE	# R00300
	DO 30 I=1,II	# R00310
	READ(LOCAT,FMT) (N(J,I),J=1,JJ)	# R00320
30	CONTINUE	# R00330
	END IF	# R00340
	IF (ICON.NE.0) THEN	# R00350
	DO 40 I=1,II	# R00360
	DO 40 J=1,JJ	# R00370
40	N(J,I)= ICON*N(J,I)	# R00380

[illegible]

C		# T00110
	SUBROUTINE SPACE (LENA,NCOL,NROW,NLAY,NCBL,IGRID,NLPOR,NZDIM,NUNIT,	# T00120
	1LCQX,LCQY,LCQZ,LCPOR,LCIBOU,LCXMAX,LCDX,LCYMAX,LCDY,LCZBOT,LCZTOP,	# T00130
	2LCDZ,LCHEAD,LCBUFF,LCLAYC,LCNCON,LCDZCB,LCIBUF,LCIUN,LCXLC,LCQSS,	# T00140
	3LCYLC,LCZLC,LCZLL,LCTOT,LCJLC,LCILC,LCKLC,NPART,NCP1,NRP1,NLP1,	# T00150
	4I1,I7,IBATCH)	# T00160
C		# T00170
C		# T00180
	IU=I1	# T00190
	NUNIT=8	# T00200
C		# T00210
C	ENTER DIMENSION DATA	# T00220
C		# T00230
	READ(IU,5000) NCOL,NROW,NLAY,NCBL,IGRID	# T00240
5000	FORMAT(8I10)	# T00250
C		# T00260
	NCRL=NCOL*NROW*NLAY	# T00270
	NCP1=NCOL+1	# T00280
	NRP1=NROW+1	# T00290
	NLP1=NLAY+1	# T00300
	NLPOR= NLAY+NCBL	# T00310
C		# T00320
	IF (IGRID.EQ.1) THEN	# T00330
	NZDIM=NLAY	# T00340
	ELSE	# T00350
	NZDIM=NCRL	# T00360
	END IF	# T00370
C		# T00380
	ISUM=1	# T00390
	LCQX=ISUM	# T00400
	ISUM=ISUM+NCRL+ (NROW*NLAY)	# T00410
	LCQY=ISUM	# T00420
	ISUM=ISUM+NCRL+ (NCOL*NLAY)	# T00430
	LCQZ=ISUM	# T00440
	ISUM=ISUM+NCRL+ (NCOL*NROW)	# T00450
	LCPOR=ISUM	# T00460
	ISUM=ISUM+ (NCOL*NROW*NLPOR)	# T00470
	LCIBOU=ISUM	# T00480
	ISUM=ISUM+NCRL	# T00490
	LCXMAX=ISUM	# T00500
	ISUM=ISUM+NCOL	# T00510
	LCDX=ISUM	# T00520
	ISUM=ISUM+NCOL	# T00530
	LCYMAX=ISUM	# T00540
	ISUM=ISUM+NROW	# T00550
	LCDY=ISUM	# T00560
	ISUM=ISUM+NROW	# T00570
	LCZBOT=ISUM	# T00580
	ISUM=ISUM+NZDIM	# T00590
	LCZTOP=ISUM	# T00600
	ISUM=ISUM+NZDIM	# T00610
	LCDZ=ISUM	# T00620
	ISUM=ISUM+NLAY	# T00630
	LCHEAD=ISUM	# T00640
	ISUM=ISUM+NCRL	# T00650
	LCBUFF=ISUM	# T00660
	ISUM=ISUM+NCRL	# T00670
	LCLAYC=ISUM	# T00680
	ISUM=ISUM+NLAY	# T00690
	LCNCON=ISUM	# T00700
	ISUM=ISUM+NLAY	# T00710
	LCDZCB=ISUM	# T00720
	ISUM=ISUM+NLAY	# T00730
	LCIBUF=ISUM	# T00740
	ISUM=ISUM + NCRL	# T00750
	LCIUN=ISUM	# T00760

```

      ISUM=ISUM + NUNIT
      LCQSS=ISUM
      ISUM=ISUM+NCRL
      ISM1=ISUM-1
      NDIF=LENA-ISM1
      NPART=NDIF/8
      IF (NPART.LT.1) THEN
      IF (IBATCH.EQ.0) THEN
      WRITE (*,*)
1 'THE SIZE OF THE MASTER ARRAY MUST BE INCREASED.'
      WRITE (*,*)
1 'REDIMENSION THE MASTER ARRAY:  A(LENA)'
      WRITE (*,5010) ISM1
5010  FORMAT(' WITH      LENA =',I8,' + 8 X (NUMBER OF PARTICLES)')
      END IF
      WRITE (I7,*)
1 'THE SIZE OF THE MASTER ARRAY MUST BE INCREASED.'
      WRITE (I7,*)
1 'REDIMENSION THE MASTER ARRAY:  A(LENA)'
      WRITE (I7,5010) ISM1
      STOP
      END IF
      LCXLC=ISUM
      ISUM=ISUM+NPART
      LCYLC=ISUM
      ISUM=ISUM+NPART
      LCZLC=ISUM
      ISUM=ISUM+NPART
      LCZLL=ISUM
      ISUM=ISUM+NPART
      LCJLC=ISUM
      ISUM=ISUM+NPART
      LCILC=ISUM
      ISUM=ISUM+NPART
      LCKLC=ISUM
      ISUM=ISUM+NPART
      LCTOT=ISUM
      ISUM=ISUM+NPART
      ISM1=ISUM-1
      IF (IBATCH.EQ.0) THEN
      WRITE (*,5020) NPART
5020  FORMAT (' THE MAXIMUM NUMBER OF PARTICLES FOR THIS RUN IS ',
1 I8)
      END IF
      WRITE (I7,5030) NPART
5030  FORMAT (' MAXIMUM NUMBER OF PARTICLES IS ',I8)
      WRITE (I7,5040) ISM1,LENA
5040  FORMAT (I8,' ELEMENTS OUT OF ',I8,' USED IN THE "A" ARRAY')
      RETURN
      END
C-----END OF ROUTINE-----
C
C---Version 1.2   April 27, 1990
C*****
C
C                      CELDAT
C
C  THIS SUBROUTINE PROVIDES A SUMMARY OF IMPORTANT DATA FOR A SPECIFIED
C  GRID CELL.
C
C*****
C
C      SUBROUTINE CELDAT (XMAX,YMAX,DELR,DELC,ZBOT,ZTOP,DELZCB,POR,
1IBOUND,HEAD,QX,QY,QZ,LAYCON,NCOL,NROW,NLAY,NZDIM,IGRID,NLPOR,
2NCP1,NRP1,NLP1,NCON,QSS,I9,IBATCH)
C

```


	CHARACTER*79 MES	#	U00160
C		#	U00170
	DIMENSION XMAX(NCOL),YMAX(NROW),DELR(NCOL),DELC(NROW),ZBOT(NZDIM),	#	U00180
	1ZTOP(NZDIM),DELZCB(NLAY),POR(NCOL,NROW,NLPOR),	#	U00190
	2IBOUND(NCOL,NROW,NLAY),HEAD(NCOL,NROW,NLAY),QX(NCP1,NROW,NLAY),	#	U00200
	3QY(NCOL,NRP1,NLAY),QZ(NCOL,NROW,NLP1),NCON(NLAY),LAYCON(NLAY),	#	U00210
	4QSS(NCOL,NROW,NLAY)	#	U00220
C		#	U00230
	MES= ' DO YOU WANT TO CHECK DATA FOR ANOTHER CELL ?'	#	U00240
10	IF (IBATCH.EQ.0) WRITE (*,*)	#	U00250
	1'ENTER J I K COORDINATES OF CELL: J I K'	#	U00260
	IF (IBATCH.EQ.0) THEN	#	U00270
	READ (*,*) J,I,K	#	U00280
	WRITE(I9,*) J,I,K	#	U00290
	ELSE	#	U00300
	READ(I9,*) J,I,K	#	U00310
	END IF	#	U00320
	IF (IBATCH.EQ.0) WRITE (*,5000) J,I,K	#	U00330
5000	FORMAT(' DATA FOR CELL (' ,I3,' ,',I3,' ,',I3,')')	#	U00340
	IF (IBOUND(J,I,K).GT.-1000.AND.IBOUND(J,I,K).LT.1000) THEN	#	U00350
	IB=IBOUND(J,I,K)	#	U00360
	ISINK=0	#	U00370
	ELSE	#	U00380
	IB=IBOUND(J,I,K)/1000	#	U00390
	ISINK=1	#	U00400
	END IF	#	U00410
	IF (IBATCH.EQ.0) WRITE (*,5010) IB,HEAD(J,I,K),POR(J,I,K)	#	U00420
5010	FORMAT(' IBOUND=',I7,' HEAD=',E15.5,' POROSITY=',F6.3)	#	U00430
	X1=0.0	#	U00440
	X2=XMAX(J)	#	U00450
	IF (J.GT.1) X1=XMAX(J-1)	#	U00460
	IF (IBATCH.EQ.0) WRITE (*,5020) X1,X2,DELR(J)	#	U00470
5020	FORMAT(' X1=',E15.5,' X2=',E15.5,' (X2-X1)=',E15.5)	#	U00480
	Y1=0.0	#	U00490
	Y2=YMAX(I)	#	U00500
	IF (I.LT.NROW) Y1=YMAX(I+1)	#	U00510
	IF (IBATCH.EQ.0) WRITE (*,5030) Y1,Y2,DELC(I)	#	U00520
5030	FORMAT(' Y1=',E15.5,' Y2=',E15.5,' (Y2-Y1)=',E15.5)	#	U00530
	IF (IGRID.EQ.1) THEN	#	U00540
	N=K	#	U00550
	ELSE	#	U00560
	N= (K-1)*NCOL*NROW + (I-1)*NCOL + J	#	U00570
	END IF	#	U00580
	ZMX=ZTOP(N)	#	U00590
	IF (LAYCON(K).EQ.1) ZMX=HEAD(J,I,K)	#	U00600
	IF (LAYCON(K).GT.1.AND.HEAD(J,I,K).LT.ZTOP(N)) ZMX=HEAD(J,I,K)	#	U00610
	DZ=ZMX-ZBOT(N)	#	U00620
	IF (IBATCH.EQ.0) WRITE (*,5040) ZBOT(N),ZMX,DZ	#	U00630
5040	FORMAT(' Z1=',E15.5,' Z2=',E15.5,' (Z2-Z1)=',E15.5)	#	U00640
	QX1=QX(J,I,K)	#	U00650
	QX2=QX(J+1,I,K)	#	U00660
	IF (ABS(DZ).GT.1.0E-20) THEN	#	U00670
	VX1=QX1/(Y2-Y1)/(ZMX-ZBOT(N))/POR(J,I,K)	#	U00680
	VX2=QX2/(Y2-Y1)/(ZMX-ZBOT(N))/POR(J,I,K)	#	U00690
	ELSE	#	U00700
	VX1=0.0	#	U00710
	VX2=0.0	#	U00720
	END IF	#	U00730
	IF (IBATCH.EQ.0) THEN	#	U00740
	WRITE (*,*)	#	U00750
	1'VOLUMETRIC FLOW (Q1-Q6) AND VELOCITY (V1-V6) AT CELL FACES'	#	U00760
	WRITE (*,5050) QX1,QX2,VX1,VX2	#	U00770
5050	FORMAT(' X (FACES 1&2): Q1=',1PE13.5,' Q2=',E13.5,' V1=',E11.3,	#	U00780
	1 ' V2=',1PE11.3)	#	U00790
	END IF	#	U00800
	OY1=OY(J,I+1,K)	#	U00810

```

QY2=QY(J,I,K) # U00820
IF (ABS(DZ).GT.1.0E-20) THEN # U00830
VY1=QY1/(X2-X1)/(ZMX-ZBOT(N))/POR(J,I,K) # U00840
VY2=QY2/(X2-X1)/(ZMX-ZBOT(N))/POR(J,I,K) # U00850
ELSE # U00860
VY1=0.0 # U00870
VY2=0.0 # U00880
END IF # U00890
IF (IBATCH.EQ.0) WRITE (*,5060) QY1,QY2,VY1,VY2 # U00900
5060 FORMAT(' Y (FACES 3&4): Q3=',1PE13.5,' Q4=',E13.5,' V3=',E11.3, # U00910
1 ' V4=',E11.3) # U00920
QZ1=QZ(J,I,K+1) # U00930
QZ2=QZ(J,I,K) # U00940
VZ1=QZ1/(X2-X1)/(Y2-Y1)/POR(J,I,K) # U00950
VZ2=QZ2/(X2-X1)/(Y2-Y1)/POR(J,I,K) # U00960
IF (IBATCH.EQ.0) WRITE (*,5070) QZ1,QZ2,VZ1,VZ2 # U00970
5070 FORMAT(' Z (FACES 5&6): Q5=',E13.5,' Q6=',E13.5,' V5=',E11.3, # U00980
1 ' V6=',E11.3) # U00990
QI=0.0 # U01000
QO=0.0 # U01010
IF(QX1.GT.0.0) THEN # U01020
QI=QI+QX1 # U01030
ELSE # U01040
QO=QO-QX1 # U01050
END IF # U01060
IF(QX2.LT.0.0) THEN # U01070
QI=QI-QX2 # U01080
ELSE # U01090
QO=QO+QX2 # U01100
END IF # U01110
IF(QY1.GT.0.0) THEN # U01120
QI=QI+QY1 # U01130
ELSE # U01140
QO=QO-QY1 # U01150
END IF # U01160
IF(QY2.LT.0.0) THEN # U01170
QI=QI-QY2 # U01180
ELSE # U01190
QO=QO+QY2 # U01200
END IF # U01210
IF(QZ1.GT.0.0) THEN # U01220
QI=QI+QZ1 # U01230
ELSE # U01240
QO=QO-QZ1 # U01250
END IF # U01260
IF(QZ2.LT.0.0) THEN # U01270
QI=QI-QZ2 # U01280
ELSE # U01290
QO=QO+QZ2 # U01300
END IF # U01310
QINOUT= QI - QO + QSS(J,I,K) # U01320
IF (IBATCH.EQ.0) THEN # U01330
WRITE (*,5080) QI,QO,QSS(J,I,K) # U01340
5080 FORMAT(' TOTAL IN ACROSS CELL FACES =',E13.5/ # U01350
1 ' TOTAL OUT ACROSS CELL FACES =',E13.5/ # U01360
1 ' NET SOURCES =',E13.5) # U01370
WRITE (*,*) 'MASS BALANCE ERROR IN PERCENT IS DEFINED AS:' # U01380
WRITE (*,*) ' 100 X [(IN - OUT + NET SOURCES)/(IN + OUT)/2]]' # U01390
END IF # U01400
QAVE=(QI+QO)/2.0 # U01410
IF(QAVE.GT.1.0E-20) THEN # U01420
ERROR= 100.0*(QINOUT/QAVE) # U01430
IF (IBATCH.EQ.0) WRITE (*,5090) ERROR # U01440
ELSE # U01450
IF (IBATCH.EQ.0) THEN # U01460
WRITE (*,*) 'NO FLOW INTO OR OUT OF THIS CELL' # U01470

```

```

        WRITE (*,*) ' NO MASS BALANCE COMPUTED' # U01480
        END IF # U01490
        END IF # U01500
5090 FORMAT(' MASS BALANCE ERROR =',F10.5,' PERCENT') # U01510
        IF(NCON(K).GT.0) THEN # U01520
            LCON=NCON(K) # U01530
            NLP=NLAY+LCON # U01540
            VZCB=QZ1/(X2-X1)/(Y2-Y1)/POR(J,I,NLP) # U01550
            IF (IGRID.EQ.1) THEN # U01560
                THICK=DELZCB(K) # U01570
            ELSE IF (IGRID.EQ.0.AND.K.LT.NLAY) THEN # U01580
                N= (K-1)*NCOL*NROW + (I-1)*NCOL + J # U01590
                NBELOW= N + (NCOL*NROW) # U01600
                THICK= ZBOT(N) - ZTOP(NBELOW) # U01610
            END IF # U01620
            IF(IBATCH.EQ.0) WRITE (*,5100) THICK,POR(J,I,NLP),VZCB # U01630
5100 FORMAT(' CONFINING BED DATA: '/' THICKNESS =',1PE13.5/ # U01640
1 ' POROSITY =',E13.5/' VERTICAL VELOCITY =',E13.5) # U01650
        END IF # U01660
C # U01670
        CALL YESNO(IBATCH,I9,MES,IAN) # U01680
        IF(IANS.EQ.1) GO TO 10 # U01690
        RETURN # U01700
        END # U01710
C-----END OF ROUTINE----- # U01720
C # V00010
C---Version 1.2 April 27, 1990 # V00020
C***** # V00030
C # V00040
C BALNCE # V00050
C # V00060
C THIS SUBROUTINE CALCULATES VOLUMETRIC BALANCES FOR ALL VARIABLE HEAD # V00070
C CELLS IN THE FINITE DIFFERENCE GRID. # V00080
C # V00090
C***** # V00100
        SUBROUTINE BALNCE (QX,QY,QZ,QSS,IBOUND,HEAD,NCOL,NROW,NLAY, # V00110
1NCP1,NRP1,NLP1,KOUNT,I7,IBATCH,ETOL) # V00120
C # V00130
        DIMENSION QX(NCP1,NROW,NLAY),QY(NCOL,NRP1,NLAY), # V00140
1QZ(NCOL,NROW,NLP1),QSS(NCOL,NROW,NLAY),HEAD(NCOL,NROW,NLAY), # V00150
2IBOUND(NCOL,NROW,NLAY) # V00160
C # V00170
        KOUNT=0 # V00180
        EMAX=0.0 # V00190
        DO 10 K=1,NLAY # V00200
        DO 10 I=1,NROW # V00210
        DO 10 J=1,NCOL # V00220
            IF(IBOUND(J,I,K).LE.0) GO TO 10 # V00230
            IF(HEAD(J,I,K).GT.1.0E+29) GO TO 10 # V00240
            QX1=QX(J,I,K) # V00250
            QX2=QX(J+1,I,K) # V00260
            QY1=QY(J,I+1,K) # V00270
            QY2=QY(J,I,K) # V00280
            QZ1=QZ(J,I,K+1) # V00290
            QZ2=QZ(J,I,K) # V00300
C # V00310
            QI=0.0 # V00320
            QO=0.0 # V00330
            IF(QX1.GT.0.0) THEN # V00340
                QI=QI+QX1 # V00350
            ELSE # V00360
                QO=QO-QX1 # V00370
            END IF # V00380
            IF(QX2.LT.0.0) THEN # V00390
                QI=QI-QX2 # V00400
            ELSE # V00410

```


C	SUBROUTINE FLUXBC (Q,QSS,QX,QY,IBOUND,DELR,DELC,J,I,K,NCOL, 1NROW,NLAY,NCP1,NRP1)	# W00100 # W00110 # W00120 # W00130 # W00140 # W00150 # W00160 # W00170 # W00180 # W00190 # W00200 # W00210 # W00220 # W00230 # W00240 # W00250 # W00260 # W00270 # W00280 # W00290 # W00300 # W00310 # W00320 # W00330 # W00340 # W00350 # W00360 # W00370 # W00380 # W00390 # W00400 # W00410 # W00420 # W00430 # W00440 # W00450 # W00460 # W00470 # W00480 # W00490 # W00500 # W00510 # W00520 # W00530 # W00540 # W00550 # W00560 # W00570 # W00580 # W00590 # W00600 # W00610 # W00620 # W00630 # W00640 # W00650 # W00660 # W00670 # W00680 # W00690 # W00700 # W00710 # W00720 # X00010 # X00020 # X00030
C	DIMENSION DELR(NCOL),DELC(NROW),IBOUND(NCOL,NROW,NLAY), 1 QSS(NCOL,NROW,NLAY),QX(NCP1,NROW,NLAY),QY(NCOL,NRP1,NLAY) DIMENSION QSIDE(4)	
C	SUM=0.0E+0 DO 10 N=1,4 QSIDE(N)=0.0	
10		
C	IF(J.EQ.1) THEN SUM=SUM+DELC(I) QSIDE(1)=DELC(I) ELSE IF (IBOUND(J-1,I,K).EQ.0) THEN SUM=SUM+DELC(I) QSIDE(1)=DELC(I) END IF	
C	IF(J.EQ.NCOL) THEN SUM=SUM+DELC(I) QSIDE(2)=DELC(I) ELSE IF (IBOUND(J+1,I,K).EQ.0) THEN SUM=SUM+DELC(I) QSIDE(2)=DELC(I) END IF	
C	IF(I.EQ.1) THEN SUM=SUM+DELR(J) QSIDE(4)=DELR(J) ELSE IF (IBOUND(J,I-1,K).EQ.0) THEN SUM=SUM+DELR(J) QSIDE(4)=DELR(J) END IF	
C	IF(I.EQ.NROW) THEN SUM=SUM+DELR(J) QSIDE(3)=DELR(J) ELSE IF (IBOUND(J,I+1,K).EQ.0) THEN SUM=SUM+DELR(J) QSIDE(3)=DELR(J) END IF	
C	ASUM= ABS(SUM) IF (ASUM.LT.1.0E-20) THEN QSS(J,I,K)=QSS(J,I,K) + Q IF(Q.LE.0.0.AND.IBOUND(J,I,K).LT.1000) 1 IBOUND(J,I,K)=1000*IBOUND(J,I,K) RETURN END IF	
C	DO 20 N=1,4 QSIDE(N)= Q*(QSIDE(N)/SUM)	
20		
C	QX(J,I,K)= QX(J,I,K) + QSIDE(1) QX(J+1,I,K)= QX(J+1,I,K) - QSIDE(2) QY(J,I+1,K)= QY(J,I+1,K) + QSIDE(3) QY(J,I,K)= QY(J,I,K) - QSIDE(4)	
C	RETURN END	
C	-----END OF ROUTINE-----	
C	C---Version 1.2 April 27, 1990	
C	C*****	

C			#	X00040
C		FILES	#	X00050
C			#	X00060
C	THIS SUBROUTINE OPENS FILES.		#	X00070
C			#	X00080
C	*****		#	X00090
C			#	X00100
	SUBROUTINE FILES (I0,I1,I3,I5,I7,I9,IBATCH)		#	X00110
	CHARACTER*80 FNAME		#	X00120
C			#	X00130
C	CREATE AND OPEN STANDARD OUTPUT AND SCRATCH FILES		#	X00140
C			#	X00150
	FNAME= 'SUMMARY.PTH'		#	X00160
	CALL OPNFIL (I7,FNAME,4,I7,IBATCH,3)		#	X00170
	FNAME= 'ENDPOINT '		#	X00180
	CALL OPNFIL (I3,FNAME,4,I7,IBATCH,3)		#	X00190
	CALL OPNFIL (I5,FNAME,3,I7,IBATCH,3)		#	X00200
C			#	X00210
C	OPEN FILE CONTAINING FILE NAMES AND FORTRAN UNITS		#	X00220
C			#	X00230
	IF (IBATCH.EQ.0) WRITE (*,*)		#	X00240
	1'ENTER NAME OF FILE CONTAINING NAMES AND UNITS OF DATA FILES:'		#	X00250
	IF (IBATCH.EQ.0) THEN		#	X00260
	READ (*,5000) FNAME		#	X00270
	WRITE(I9,5000) FNAME		#	X00280
	ELSE		#	X00290
	READ(I9,5000) FNAME		#	X00300
	END IF		#	X00310
5000	FORMAT (A)		#	X00320
C			#	X00330
	CALL OPNFIL (I0,FNAME,1,I7,IBATCH,1)		#	X00340
C			#	X00350
	KFIRST=0		#	X00360
10	READ (I0,*,END=20) IU,FNAME		#	X00370
	IF(KFIRST.EQ.0) I1=IU		#	X00380
	KFIRST=1		#	X00390
C			#	X00400
C	OPEN DATA FILES		#	X00410
C			#	X00420
	CALL OPNFIL (IU,FNAME,1,I7,IBATCH,1)		#	X00430
C			#	X00440
	WRITE(I7,5010) IU,FNAME		#	X00450
5010	FORMAT(' FORTRAN UNIT ',I3,' = ',A60)		#	X00460
	GO TO 10		#	X00470
20	CONTINUE		#	X00480
	RETURN		#	X00490
	END		#	X00500
C	-----END OF ROUTINE-----		#	X00510
C			#	Y00010
C	---Version 1.2 April 27, 1990		#	Y00020
C	*****		#	Y00030
C			#	Y00040
C		RDBUDG	#	Y00050
C			#	Y00060
C	THIS SUBROUTINE READS AN ENTRY IN A BINARY BUDGET FILE PRODUCED BY		#	Y00070
C	THE USGS MODULAR FINITE-DIFFERENCE FLOW MODEL.		#	Y00080
C			#	Y00090
C	*****		#	Y00100
C			#	Y00110
	SUBROUTINE RDBUDG (BUFF,LABEL,NCOL,NROW,NLAY,IU,IO)		#	Y00120
	DIMENSION BUFF(NCOL,NROW,NLAY)		#	Y00130
	CHARACTER*16 LABEL,TEXT		#	Y00140
C			#	Y00150
	IPASS=0		#	Y00160
10	IPASS=IPASS+1		#	Y00170
C			#	Y00180

```

        IF(IPASS.GT.2) THEN                                # Y00190
        WRITE (IO,5000) LABEL,IU                          # Y00200
5000    FORMAT (1X,A16,' BUDGET TERMS WERE NOT FOUND ON UNIT ',I3) # Y00210
        STOP                                              # Y00220
        END IF                                           # Y00230
C                                                    # Y00240
20    READ (IU,END=30 ) KSTP,KPER,TEXT,NC,NR,NL        # Y00250
        IF (TEXT.EQ.LABEL) THEN                        # Y00260
        READ (IU) (((BUFF(J,I,K),J=1,NCOL),I=1,NROW),K=1,NLAY) # Y00270
        RETURN                                          # Y00280
        ELSE                                           # Y00290
        READ (IU) (((BUFF(J,I,K),J=1,NCOL),I=1,NROW),K=1,NLAY) # Y00300
        END IF                                         # Y00310
        GO TO 20                                       # Y00320
30    REWIND (IU)                                       # Y00330
        GO TO 10                                       # Y00340
        END                                           # Y00350
C-----END OF ROUTINE-----                          # Y00360
C                                                    # Z00010
C---Version 1.2   April 27, 1990                      # Z00020
C*****                                                # Z00030
C                                                    # Z00040
C                                                    # Z00050
C                                                    # Z00060
C THIS ROUTINE READS THE FIRST LINE OF THE FILE NAMED "TERMIN". IT # Z00070
C LOOKS FOR THE FIRST NON-BLANK CHARACTER. IF THAT CHARACTER IS "S", # Z00080
C IBATCH IS SET TO 0 FOR SCREEN INPUT. IF THE CHARACTER IS "B", IBATCH # Z00090
C IS SET TO 1 TO READ "SCREEN" INPUT FROM FILE "TERMIN". FILE "TERMIN" # Z00100
C IS OPENED BY THIS ROUTINE.                          # Z00110
C                                                    # Z00120
C*****                                                # Z00130
C                                                    # Z00140
C        SUBROUTINE IOTYPE (IBATCH,I9,I7)              # Z00150
C                                                    # Z00160
C        CHARACTER*132 LINE1                          # Z00170
C        CHARACTER*6 S                                  # Z00180
C        CHARACTER*5 B                                  # Z00190
C        CHARACTER*80 FNAME                            # Z00200
C                                                    # Z00210
C        S= 'SCREEN'                                    # Z00220
C        B= 'BATCH'                                    # Z00230
C                                                    # Z00240
C        FNAME= 'TERMIN.PTH'                          # Z00250
C        CALL OPNFIL (I9,FNAME,4,I7,IBATCH,3)          # Z00260
C                                                    # Z00270
C        READ (I9,'(A)',END=10 ) LINE1                # Z00280
C        IS= INDEX (LINE1,S)                          # Z00290
C        IB= INDEX (LINE1,B)                          # Z00300
C        IF (IS.NE.0) THEN                             # Z00310
C        IBATCH=0                                       # Z00320
C        ELSE IF (IB.NE.0) THEN                        # Z00330
C        IBATCH=1                                       # Z00340
C        ELSE IF (IS.EQ.0.AND.IB.EQ.0) THEN            # Z00350
C        GO TO 10                                       # Z00360
C        END IF                                         # Z00370
C        RETURN                                         # Z00380
10    CONTINUE                                         # Z00390
        REWIND (I9)                                    # Z00400
        WRITE (I9,'(A)') 'SCREEN'                    # Z00410
        IBATCH=0                                       # Z00420
        RETURN                                          # Z00430
        END                                           # Z00440
C                                                    # Z00450
C-----END OF ROUTINE-----                          # Z00460
C                                                    # AA00010
C---Version 1.2   April 27, 1990                      # AA00020

```

```

C*****#AA00030
C#AA00040
C#AA00050
C#AA00060
C THIS ROUTINE COMPUTES THE GLOBAL Z COORDINATE GIVEN THE LOCAL#AA00070
C Z COORDINATE WITHIN THE CELL#AA00080
C#AA00090
C*****#AA00100
C#AA00110
C SUBROUTINE ZCALC (IGRID,NCOL,NROW,NLAY,ZLC,ZLLC,ZTOP,ZBOT,#AA00120
1 HEAD,LAYCON,ILC,JLC,KLC,NZDIM,NPART,NPRT,I7)#AA00130
C#AA00140
C DIMENSION ZLLC(NPART),ZLC(NPART),ZTOP(NZDIM),ZBOT(NZDIM),#AA00150
1 HEAD(NCOL,NROW,NLAY),LAYCON(NLAY),ILC(NPART),JLC(NPART),#AA00160
2 KLC(NPART)#AA00170
C#AA00180
C DO 10 NPT=1,NPRT#AA00190
IP=ILC(NPT)#AA00200
JP=JLC(NPT)#AA00210
KP=KLC(NPT)#AA00220
ZLOC= ZLLC(NPT)#AA00230
IF(IGRID.EQ.1) THEN#AA00240
NK=KP#AA00250
NKP=NK+1#AA00260
ELSE#AA00270
NK = (KP-1)*NCOL*NROW + (IP-1)*NCOL + JP#AA00280
NKP= NK + (NCOL*NROW)#AA00290
END IF#AA00300
IF(KP.EQ.NLAY.AND.ZLOC.LT.0.0) THEN#AA00310
WRITE(I7,*) ' RUN STOPPED. INCONSISTENT CONFINING BED LOCATION'#AA00320
STOP#AA00330
END IF#AA00340
ZMX=ZTOP(NK)#AA00350
HED=HEAD(JP,IP,KP)#AA00360
IF(LAYCON(KP).EQ.1) ZMX=HED#AA00370
IF(LAYCON(KP).GT.1.AND.HED.LT.ZMX) ZMX=HED#AA00380
IF(ZLOC.GE.0.0) THEN#AA00390
ZP=ZLOC*ZMX + (1.0-ZLOC)*ZBOT(NK)#AA00400
ELSE#AA00410
ZL= 1.0+ZLOC#AA00420
ZP=ZL*ZBOT(NK) + (1.0-ZL)*ZTOP(NKP)#AA00430
END IF#AA00440
ZLC(NPT)=ZP#AA00450
10 CONTINUE#AA00460
RETURN#AA00470
END#AA00480
C#AA00490
C-----END OF ROUTINE-----#AA00500
C#AB00010
C---Version 1.2 April 27, 1990#AB00020
C*****#AB00030
C#AB00040
C#AB00050
C#AB00060
C THIS ROUTINE OPENS A SINGLE FILE.#AB00070
C#AB00080
C*****#AB00090
C#AB00100
C SUBROUTINE OPNFIL (IU,FNAME,NSTAT,IOUT,IBATCH,NACT)#AB00110
CHARACTER*80 FNAME,FMT,FM#AB00120
LOGICAL*4 EX#AB00130
C#AB00140
C#AB00150
C#AB00160
C#AB00170
C IU = FORTRAN UNIT NUMBER OF FILE#AB00180
C FNAME = FILE NAME#AB00180

```



```

C  NSTAT = STATUS OF FILE                                #AB00190
C      1 => OLD FILE                                     #AB00200
C      2 => NEW FILE                                     #AB00210
C      3 => SCRATCH FILE (DELETED AUTOMATICALLY WHEN RUN ENDS) #AB00220
C      4 => UNDETERMINED STATUS (MAY OR MAY NOT EXIST)    #AB00230
C          IF IT DOES NOT EXIST, IT IS CREATED BY OPEN STATEMENT #AB00240
C          IF IT DOES EXIST, IT IS OPENED AS 'OLD' FILE    #AB00250
C                                                         #AB00260
C  IOUT = UNIT NUMBER FOR OUTPUT FILE TO WRITE ERROR MESSAGE TO #AB00270
C      IF NECESSARY                                     #AB00280
C                                                         #AB00290
C  IBATCH = FLAG INDICATING IF THERE IS INTERACTIVE DIALOGUE AT #AB00300
C      TERMINAL                                         #AB00310
C                                                         #AB00320
C      0 => THERE IS INTERACTIVE DIALOGUE                #AB00330
C      1 => THERE IS NOT INTERACTIVE DIALOGUE (BATCH MODE) #AB00340
C                                                         #AB00350
C  NACT = VARIABLE DENOTING IF A FILE IS READ ONLY, WRITE ONLY, OR #AB00360
C      READ AND WRITE.                                  #AB00370
C                                                         #AB00380
C      1 => READ ONLY                                     #AB00390
C      2 => WRITE ONLY                                    #AB00400
C      3 => READ AND WRITE                                #AB00410
C                                                         #AB00420
C  "NACT" IS NOT USED IN THIS SUBROUTINE. ALL FILES ARE OPENED FOR #AB00430
C  READING AND WRITING. OPENING "READ ONLY" AND "WRITE ONLY" FILES #AB00440
C  IS MACHINE DEPENDENT. THE VARIABLE "NACT" IS PASSED TO THIS     #AB00450
C  ROUTINE TO MAKE IT EASY TO MODIFY THE OPEN STATEMENTS TO ALLOW  #AB00460
C  FOR READ AND WRITE ONLY FILES ON ANY GIVEN MACHINE. THE CALLS TO #AB00470
C  THIS SUBROUTINE ARE CURRENTLY SET UP SO THAT ANY FILE THAT IS   #AB00480
C  WRITTEN TO IS GIVEN "NACT = 3" AND ANY FILE THAT IS READ FROM BUT #AB00490
C  NOT WRITTEN TO IS GIVEN "NACT = 1". NO FILES ARE GIVEN "WRITE ONLY" #AB00500
C  STATUS. "READ ONLY" STATUS IS USEFUL IF A NUMBER OF USERS WANT  #AB00510
C  TO SIMULTANEOUSLY SHARE INPUT FILES.                      #AB00520
C                                                         #AB00530
C  IF THE FILE DOES NOT CURRENTLY EXIST, A NEGATIVE UNIT NUMBER IS  #AB00540
C  USED AS A FLAG TO INDICATED THAT A NEW FILE SHOULD BE CREATED AS #AB00550
C  AN UNFORMATTED (BINARY) FILE.                              #AB00560
C                                                         #AB00570
C      IBIN=0                                             #AB00580
C      IF (IU.LT.0) THEN                                  #AB00590
C          IU= -IU                                        #AB00600
C          IBIN=1                                         #AB00610
C      END IF                                             #AB00620
C                                                         #AB00630
C  CHECK TO SEE IF A FILE IS OPENED TO UNIT=IOUT SO THAT ERROR    #AB00640
C  MESSAGES CAN BE WRITTEN.                                     #AB00650
C      INQUIRE (UNIT=IOUT,OPENED=EX)                    #AB00660
C      IO=0                                               #AB00670
C      IF (EX) IO=1                                       #AB00680
C                                                         #AB00690
C  OPEN AN EXISTING FILE                                         #AB00700
C                                                         #AB00710
C      IF (NSTAT.EQ.1) THEN                                  #AB00720
10  INQUIRE (FILE=FNAME,EXIST=EX,UNFORMATTED=FM)          #AB00730
C  CHECK TO SEE IF FILE EXISTS                                  #AB00740
C      IF (EX) GO TO 20                                       #AB00750
C      IF (IBATCH.EQ.0) THEN                                  #AB00760
C          WRITE (*,*) 'THE FOLLOWING FILE DOES NOT EXIST:'    #AB00770
C          WRITE (*,'(1X,A)') FNAME                          #AB00775
C          WRITE (*,*) 'ENTER THE NAME OF AN EXISTING FILE (<CR>=QUIT):' #AB00780
C          READ (*,'(A)') FNAME                              #AB00790
C          IF (FNAME.EQ.' ') STOP                             #AB00800
C          GO TO 10                                           #AB00810
C      ELSE                                                  #AB00820
C          IF (IO.EQ.1) WRITE (IOUT,*) 'FILE DOES NOT EXIST:' #AB00830

```

	IF (IO.EQ.1) WRITE (IOUT,'(A)') FNAME	#AB00840
	STOP	#AB00850
	END IF	#AB00860
20	CONTINUE	#AB00870
	FMT='FORMATTED'	#AB00880
	IF (IBIN.EQ.1.OR.FM.EQ.'YES') FMT='UNFORMATTED'	#AB00890
	OPEN (IU,FILE=FNAME,STATUS='OLD',FORM=FMT,IOSTAT=IERR)	#AB00900
	IF (IERR.GT.0) GO TO 40	#AB00910
	RETURN	#AB00920
	END IF	#AB00930
C		#AB00940
C	OPEN A NEW FILE	#AB00950
C		#AB00960
	IF (NSTAT.EQ.2) THEN	#AB00970
30	INQUIRE (FILE=FNAME,EXIST=EX)	#AB00980
	IF (EX) THEN	#AB00990
	IF (IBATCH.EQ.0) THEN	#AB01000
	WRITE (*,*) 'THE FOLLOWING FILE ALREADY EXISTS:'	#AB01010
	WRITE (*,'(1X,A)') FNAME	#AB01015
	WRITE (*,*) 'ENTER THE NAME OF A NEW FILE (<CR>=QUIT):'	#AB01020
	READ (*,'(A)') FNAME	#AB01030
	IF (FNAME.EQ.' ') STOP	#AB01040
	GO TO 30	#AB01050
	ELSE	#AB01060
	IF (IO.EQ.1) WRITE (IOUT,*) 'FILE ALREADY EXISTS:'	#AB01070
	IF (IO.EQ.1) WRITE (IOUT,'(A)') FNAME	#AB01080
	STOP	#AB01090
	END IF	#AB01100
	END IF	#AB01110
	FMT='FORMATTED'	#AB01120
	IF (IBIN.EQ.1) FMT='UNFORMATTED'	#AB01130
	OPEN (IU,FILE=FNAME,STATUS='NEW',FORM=FMT,IOSTAT=IERR)	#AB01140
	IF (IERR.GT.0) GO TO 40	#AB01150
	RETURN	#AB01160
	END IF	#AB01170
C		#AB01180
C	OPEN A SCRATCH FILE	#AB01190
C		#AB01200
	IF (NSTAT.EQ.3) THEN	#AB01210
	FMT='FORMATTED'	#AB01220
	IF (IBIN.EQ.1) FMT='UNFORMATTED'	#AB01230
	OPEN (IU,STATUS='SCRATCH',FORM=FMT,IOSTAT=IERR)	#AB01240
C	FOR MICROSOFT FORTRAN USE:	#AB01250
C	OPEN (IU,FORM=FMT,IOSTAT=IERR)	#AB01260
	IF (IERR.GT.0) GO TO 40	#AB01270
	RETURN	#AB01280
	END IF	#AB01290
C		#AB01300
C	OPEN A FILE OF UNKNOWN STATUS	#AB01310
C		#AB01320
	IF (NSTAT.EQ.4) THEN	#AB01330
	INQUIRE (FILE=FNAME,EXIST=EX,UNFORMATTED=FM)	#AB01340
	IF (EX) THEN	#AB01350
	FMT='FORMATTED'	#AB01360
	IF (IBIN.EQ.1.OR.FM.EQ.'YES') FMT='UNFORMATTED'	#AB01370
	OPEN (IU,FILE=FNAME,STATUS='OLD',FORM=FMT,IOSTAT=IERR)	#AB01380
	IF (IERR.GT.0) GO TO 40	#AB01390
	ELSE	#AB01400
	FMT='FORMATTED'	#AB01410
	IF (IBIN.EQ.1) FMT='UNFORMATTED'	#AB01420
	OPEN (IU,FILE=FNAME,STATUS='NEW',FORM=FMT,IOSTAT=IERR)	#AB01430
	IF (IERR.GT.0) GO TO 40	#AB01440
	END IF	#AB01450
	RETURN	#AB01460
	END IF	#AB01470
C		#AB01480

```

C WRITE MESSAGE INDICATING PROBLEM OPENING FILE #AB01490
C #AB01500
40 IF (IBATCH.EQ.0) WRITE (*,5000) IU #AB01510
IF (IO.EQ.1) WRITE (IOUT,5000) IU #AB01520
5000 FORMAT (' ERROR OPENING FILE TO UNIT ',I3) #AB01530
STOP #AB01540
END #AB01550
C-----END OF ROUTINE----- #AB01560
C #AC00010
C---Version 1.2 April 27, 1990 #AC00020
C***** #AC00030
C #AC00040
C FACTYP #AC00050
C #AC00060
C THIS ROUTINE DETERMINES WHETHER A CELL FACE IS A BOUNDARY FACE OR #AC00070
C A CONNECTION WITH AN ACTIVE CELL. IF A BOUNDARY, NBD=1; IF A #AC00080
C CONNECTION TO AN ACTIVE CELL, NBD=0. #AC00090
C #AC00100
C***** #AC00110
C #AC00120
SUBROUTINE FACTYP (N,NBOUND,JNXT,INXT,KNXT,NBD,IBOUND,NCOL,NROW, #AC00130
1 NLAY,J,I,K,IFACE,IPAC,IO) #AC00140
DIMENSION IBOUND (NCOL,NROW,NLAY) #AC00150
CHARACTER*3 TEXT(6) #AC00160
DATA TEXT /'RIV','DRN','GHB','WEL','RCH','EVT'/ #AC00170
C #AC00180
NBD=0 #AC00190
IF(N.EQ.NBOUND) THEN #AC00200
NBD=1 #AC00210
ELSE #AC00220
IF (IBOUND(JNXT,INXT,KNXT).EQ.0) THEN #AC00230
NBD=1 #AC00240
ELSE #AC00250
WRITE (IO,5000) IFACE,J,I,K,TEXT(IPAC) #AC00260
5000 FORMAT('FACE',I2,' OF (J,I,K) = ',I4,',',I4,',',I4, #AC00270
1' IS NOT A BOUNDARY FACE. ',A3,' FLOW TREATED AS SOURCE/SINK TERM' #AC00280
2) #AC00290
END IF #AC00300
END IF #AC00310
RETURN #AC00320
END #AC00330
C-----END OF ROUTINE----- #AC00340

```

```

C          ***** PATHHSU.FOR *****
C WRITTEN BY K.R. REHFELDT (KRR) 10/21/95, GeoTrans, Inc
C PROGRAM TO DEFINE INTERMEDIATE LOCATIONS ALONG THE GROUNDWATER
C PATHLINE. A USER DEFINED PARAMETER CALLED STEPSIZE GIVES THE
C DESIRED MAXIMUM LENGTH OF PATH BETWEEN RECORDED PATHS. THIS
C DATA WILL BE USED IN A COMPARISON WITH THE GEOLOGIC MODEL TO
C PROVIDE BETTER DEFINITION OF HSUs ALONG THE PATH. THIS WILL GIVE
C MORE PRECISE CONTROL OF HSU SPECIFIC PARAMETERS SUCH AS POROSITY,
C FRACTURE APERTURE AND SPACING.
C
C KEY INPUT DATA SET
C      PATHLINE OUTPUT FILE                      PATHLINE
C
C KEY OUTPUT DATA SET
C      PATHLINE INFORMATION AT SMALLER INTERVALS      HSUPATH.DAT
C
      IPARTI = 0
      DATA STEPSIZZ/10.0/
      DATA STEPSIZZ/100.0/
      OPEN (11,FILE='PATHLINE')
      OPEN (12,FILE='HSUPATH.DAT')
      open (13,file='hspath.err')
10    READ (11,170,END=100)VX,DVXDX,VX1,VX2,DT,XPOLD,XMN,XX,IVXFLG,IX
      READ (11,170) VY,DVYDY,VY1,VY2,DT,YPOLD,YMN,XX,IVYFLG,IY
      READ (11,170) VZ,DVZDZ,VZ1,VZ2,DT,ZPOLD,ZMN,ZMX,IVZFLG,IZ
      READ (11,160) IPART,XP,YP,ZLOC,ZP,TIME,J,I,K
      IF (IPART.NE.IPARTI) THEN
      WRITE (12,180) IPART,XP,YP,ZP,TIME,J,I,K
      IPARTI = IPART
      GO TO 10
      END IF
      DELT = 0.0
      DELPM = ABS(ZP-ZPOLD)
      DELPX = amax1(ABS(XP-XPOLD),ABS(YP-YPOLD))
      IF (DELPX/STEPSIZZ.GT.DELPM/STEPSIZZ) THEN
      NSTEP = INT(DELPX/STEPSIZZ) + 1
      ELSE
      NSTEP = INT(DELPX/STEPSIZZ) + 1
      END IF
      IF (NSTEP.GT.1) THEN
      DLT = DT/FLOAT(NSTEP)
      ELSE
      DLT = DT
      END IF
C
C      DO 20 M=1,NSTEP                                # C03120
C      DELT = DELT + DLT
C      CALL NEWXYZ (VX,DVXDX,VX1,VX2,DELT,XPOLD,XMN,XX,XNEW,IVXFLG,0) # C03130
C      CALL NEWXYZ (VY,DVYDY,VY1,VY2,DELT,YPOLD,YMN,XX,YNEW,IVYFLG,0) # C03140
C      CALL NEWXYZ (VZ,DVZDZ,VZ1,VZ2,DELT,ZPOLD,ZMN,ZMX,ZNEW,IVZFLG,0) # C03150
C
C      WRITE (12,180) IPART,XNEW,YNEW,ZNEW,DLT,J,I,K      # C03200
20    CONTINUE
      delx = xp - xnew
      dely = yp - ynew
      delz = zp - znew
C      WRITE (13,190) ipart,NSTEP,delx, dely,delz
      IF (ABS(XP-XNEW).GT.100.0.OR.ABS(YP-YNEW).GT.100.0.OR.
&ABS(ZP-ZNEW).GT.10.0) THEN
      WRITE (13,190) ipart,NSTEP,delx, dely,delz
      STOP
      END IF
      GO TO 10
160  FORMAT (I5,1X,5(E20.12,1X),2(I3,1X),I3)
170  FORMAT (8(E20.9),2I5)
180  FORMAT (I5,1X,4(E20.12),2X,4I5)

```

```

190  FORMAT (2i5,3f20.10)
100  CLOSE (11)
      CLOSE (12)
      STOP
      END

C                                          # D00010
C---Version 1.2   April 27, 1990          # D00020
C*****          # D00030
C                                          # D00040
C                      NEWXYZ              # D00050
C                                          # D00060
C  THIS SUBROUTINE COMPUTES A NEW X, Y, OR Z PARTICLE COORDINATE GIVEN # D00070
C  FACE VELOCITIES AND A TIME INTERVAL.    # D00080
C                                          # D00090
C*****          # D00100
C                                          # D00110
C      SUBROUTINE NEWXYZ (V,DV DX,V1,V2,DT,XP,XMIN,XMAX,XNEW,
1  IVFLG,ISIDE)          # D00120
C                                          # D00130
C                                          # D00140
C      IF(IVFLG.EQ.1) GO TO 10              # D00150
C      IF(IVFLG.EQ.2) GO TO 20              # D00160
C      XNEW= XP + V*(EXP(DV DX*DT) - 1.0E+0)/DV DX # D00170
C      GO TO 30                            # D00180
10  XNEW= XP + V1*DT                        # D00190
C      GO TO 30                            # D00200
20  XNEW= XP                                # D00210
C      GO TO 40                            # D00220
30  IF(ISIDE.GT.0) XNEW=XMAX                 # D00230
C      IF(ISIDE.LT.0) XNEW=XMIN             # D00240
40  CONTINUE                               # D00250
C      IF(XNEW.LE.XMIN.AND.V1.EQ.0.0E+0) XNEW=XMIN + 1.0E-3*(XMAX-XMIN) # D00260
C      IF(XNEW.GE.XMAX.AND.V2.EQ.0.0E+0) XNEW=XMAX - 1.0E-3*(XMAX-XMIN) # D00270
C      RETURN                              # D00280
C      END                                # D00290
C-----END OF ROUTINE-----          # D00300

```

```

C          ***** VELOCITY.FOR *****
C WRITTEN BY K.R. REHFELDT, (KRR), 10/18/95, GeoTrans, Inc.
C modified 10/21/96 to run the new version of the flow model
C THIS PROGRAM READS THE GEOLOGIC MODEL,
C LAYER ELEVATION DATA AND SHOT STRATIGRAPHIC INFORMATION TO CALCULATE
C   A. HSUs CROSSED BY EACH SHOT CAVITY
C   C. IDENTIFIES THE FLOW MODELS LAYERS CORRESPONDING TO EACH
C       SHOT CAVITY
C KEY INPUT DATA SETS AND SOURCES
C   MODEL GRID SPACING INFORMATION          GRIDS.DAT
C   MODEL LAYER ELEVATION DATA            M_ELEV.DAT
C   THE GEOLOGIC MODEL - HSU TOPS - NAME READ FROM
C       COND.MDL                          GEOL_15.MDL
C   PARTICLE COORDINATE INFORMATION        HSUPATH.DAT
C   POROSITY DATA FOR EACH HSU            POROSITY.DAT
C
C KEY OUTPUT DATA SETS
C   VELOCITY ALONG THE PATH LINE AFTER ACCOUNTING
C       FOR EFFECTIVE POROSITY
C                                           PATHVEL.DAT
C
C-----problem set up
C   IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C   parameter (ib=1,ie=114,jb=1,je=140,kb=0,ke=20)
C   dimension hsub(ib:ie,jb:je,kb:ke),hsub(ib:ie,jb:je,kb:ke)
C   dimension selev(ib:ie,jb:je),topelev(21),poro(20)
C   dimension delrm(76),delcm(68)
C   character*2 dummy,dummy2
C   open (17,file='pathvel.dat')
C   open (14,file='porosity.dat')
C   read (14,1110) (poro(m),m=1,20)
1110   format (f10.0)
C   close (14)
C   read the model grid coordinate
C   open(11,file='grids.dat')
C   read (11,*) d1, d2, d3, x0g, y0g
C   read (11,*) alfam, nrm, ncm, x0ul, y0ul
C   define the coordinates of the lower left corner of the MODFLOW
C   grid and the upper left of the geologic model
C   rad = alfam/57.295779
C   read (11,*) (delcm(i),i=1,ncm)
C   read (11,*) (delrm(i),i=1,nrm)
C   close (11)
C   calculate the coordinates of the model grid in earth coordinates
C   do 50 i=1,ncm
50     sumdx = sumdx + delcm(i)
C   do 51 j=1,nrm
51     sumdy = sumdy + delrm(j)
C   y0m = y0ul - sumdy*cos(rad)
C   x0m = x0ul + sumdy*sin(rad)
C   read in elevations of model layers,nmlay=actual model layers+1dummy layer
C-----M_ELEV.DAT is the elevation above mean sea level
C
C   open (10,file="m_elev.dat")
C   read (10,1000) nmlay
C   read (10,1001) (topelev(i),i=1,nmlay+1)
C   close (10)
C
C   read in geologic layers
C   open (10,file='geol_15.mdl')
C   read (10,1002) dummy
C
C we have 22 layers in geol model. "ilay" is the layer #, starting from
C 21 for surface layer to 0 for bottom elevation.

```

```

do 200 n=1,22
read (10,1003) ilay
  if (ilay.eq.21) then
    do 90 j=1,140
90      read(10,1004) (selev(i,j),i=1,114)
      go to 200
    endif
    if (ilay.ge.kb.and.ilay.le.ke) then

c here kb and ke are begin and end layers in geologic model,
c read all of layer interested
c into temp. Read outside or unwanted layers into dummy_layer

      do 100 j=1,140
100        read (10,1004) (hsut(i,j,ilay),i=1,114)
        else
          do 110 j=1,140
110          read (10,1004) (dummy_layer,i=1,114)
          endif
200      continue
c      locate top and bottom of HSUs

250      do 300 j=jb,je
        do 300 i=ib,ie
        do 300 k=ke,kb,-1
        if (hsut(i,j,k).gt.9990.) then
          goto 300
        else

c need to use "ki=k-1,kb,-1" to find the 1st non-9999. layer beneath the top
c it will be the bottom of this HSU

          do 290 ki=k-1,kb,-1
            if (hsut(i,j,ki).gt.9990.) then
              goto 290
            else
              hsub(i,j,k)=hsut(i,j,ki)
              goto 300
            endif
290          continue
          endif
300      continue
C
C
c      read the fluid statistics table for information on wells
C
      open(11,file='hsupath.dat')
      rad = alfam/57.295779
c      Read the first line of hsupath.dat
150      read(11,1006,end=400) ipart,xp,yp,zp,dt,jcell,icell,kcell
        x = xp*cos(rad) - yp*sin(rad)
        y = yp*cos(rad) + xp*sin(rad)
        xs = x + x0m
        ys = y + y0m
        lgt = 0
1006      format (i5,1x,4(e20.12),2x,3i5)
c-----find well location in the geological grid
        i_geo=ANINT((xs-x0g)/2000.)+1
        j_geo=ANINT((y0g-ys)/2000.)+1
C
c      loop through model layers at node i_geo,j_geo
c topelev(km)&topelev(km+1) are top & bottom of the specified
c model layer
c
c      loop through HSUs

```

```

53      do 390 kg=ke, kb, -1

c      loop through i,j to get HSUs within model layer
        i = i_geo
        j = j_geo
        if (hsut(i,j,kg).gt.9990.) go to 390
        if (zp.le.hsut(i,j,kg).and.zp.gt.hsub(i,j,kg)) lgt=kg
390      continue
        if (lgt.eq.0) lgt=lgtold
        dt = dt * poro(lgt)
        write(17,1008)ipart,xs,ys,zp,dt,lgt,jcell,icell,kcell
        lgtold = lgt
        go to 150
400      continue
1000      format (4i5)
1001      format (f8.2)
1002      format (a15,f10.2)
1003      format (5x,i2,i3,i5)
1004      format (10f8.2)
1005      format (f8.6)
1007      format (20i2)
1008      format (i5,1x,4(e20.10),i5,5x,3i5)
2001      format (8f12.6)
2002      format (1p8e12.5)
2150      format (114i1)
        close(18)
        stop
        end

```



```

C          ***** TIMEDIST.FOR *****
C WRITTEN BY K. R. REHFELDT, (KRR), 10/22/95, GeoTrans, Inc.
C PROGRAM TO TAKE THE OUTPUT OF PROGRAM VELOCITY.FOR AND CREATE
C INDIVIDUAL FILES FOR EACH PARTICLE WITH CUMMULATIVE TRAVEL
C TIME AND DISTANCE RECORDED ALONG WITH THE CORRESPONDING HSUs.
C
C KEY DATA SET INPUTS
C      PATHLINE INFORMATION CORRECTED FOR POROSITY      VELOCITY.FOR
C
C KEY DATA SET OUTPUTS
C      OUTPUT FILE FOR EACH PARTICLE WITH
C          CUMMULATIVE TRAVEL TIME AND DISTANCE      Pxxxx.PTH
C          SURFER PLOT FILE                          Sxxxx.DAT
C
      character*4 cvar
      character*20 nameout,nameplt
      character*1 k1,k2,k3,k4
      open (12,file='pathvel.dat')
      iparti = 0
10    read (12,160,end = 400) ipart,x,y,z,delt,ihsu
      if (ipart.ne.iparti) then
        close (13)
        dist = 0.0
        time = 0.0
        xold = x
        yold = y
        zold = z
        iparti = ipart
        write (cvar,'(i4)')ipart
        read (cvar,'(4a1)') k1,k2,k3,k4
        if (k1.eq.' ') k1 = '0'
        if (k2.eq.' ') k2 = '0'
        if (k3.eq.' ') k3 = '0'
        nameout = 'p'//k1//k2//k3//k4//'.pth'
        nameplt = 's'//k1//k2//k3//k4//'.dat'
        open (13,file = nameout)
        open (14,file = nameplt)
        write (13,170) ipart,time,dist,ihsu
        write (14,180) x,y,z,time
        go to 10
      end if
      time = time + delt/365.
      deld = sqrt((x-xold)**2 + (y-yold)**2 + (z-zold)**2)
      dist = dist + deld
      write (13,170) ipart,time,dist,ihsu
      write (14,180) x,y,z,time
      xold = x
      yold = y
      zold = z
      go to 10
160   format (i5,1x,4(e20.10),i5,5x,i5)
170   format (i5,1x,2(e20.10),2i5)
180   format (4e20.10)
400   stop
      end

```

```

C          ***** PSUMRY.FOR *****
C WRITTEN BY K.R. REHFELDT, (KRR), GeoTrans, Inc.
C CALCULATE THE GROUNDWATER TRAVEL TIME TO FIXED
C LOCATIONS, THE DISTANCE TRAVELED FOR FIXED TIMES,
C AND THE DISTANCE AT WHICH A PATH CROSSES FROM ONE
C HSU TO ANOTHER.
C
C KEY DATA SET INPUT
C     NUMBER OF PATHS TO BE EXAMINED             NUMPATHS.DAT
C     DISTANCES AND TIMES FOR WHICH
C     A SUMMARY IS REQUESTED                     PSUMRY.IN
C     THE CUMMULATIVE TIME AND DISTANCE
C     FOR EACH PATH                             Pxxxx.PTH
C     LOCATION (X,Y,Z) AND CUMMULATIVE
C     TIME FOR EACH PATH                         Sxxxx.DAT
C     FILE CONTAINING PARTICLE ID #,SHOT
C     NAME AND STARTING POINT LOCATION
C     WITH RESPECT TO THE WP                     PARTID.DAT
C
C KEY DATA SET OUTPUT
C     SUMMARY DISTANCES AT FIXED TIMES           FIXTIME.DAT
C     SUMMRAY TIMES AT FIXED LOCATIOS           FIXDIST.DAT
C     OUTPUT FILES WITH X AND HSU               Hxxxx.DAT
C
C
C     DIMENSION TIME(2000),DIST(2000),IHSU(2000),X(2000),Y(2000)
C     DIMENSION TIMES(10),RECEP(10),TD(10),RT(10),Z(2000)
C     DIMENSION TX(10),TY(10),TZ(10)
C     CHARACTER*1 K1,K2,K3,K4
C     CHARACTER*4 CVAR
C     CHARACTER*14 SNAME
C     CHARACTER*2 SLOC
C     CHARACTER*20 INFILE
C     CHARACTER*20 INFILE2
C     CHARACTER*20 OUTFILE
C     OPEN (11,FILE='PSUMRY.IN')
C     OPEN (15,FILE='FIXTIME.DAT')
C     OPEN (16,FILE='FIXDIST.DAT')
C     OPEN (18,FILE='FIXTLOC.DAT')
C     OPEN (17,FILE='PARTID.DAT')
C     READ (11,100) NT,NR
C     READ (11,*) (TIMES(I),I=1,NT)
C     READ (11,*) (RECEP(I),I=1,NR)
C     OPEN (12,FILE='NUMPATHS.DAT')
C     READ (12,100) NUMPATH
C     WRITE (15,200)(TIMES(I),I=1,NT)
C     WRITE (18,400)(TIMES(I),I=1,NT)
200  FORMAT (' TRAVEL DISTANCE(m) AT FIXED TIMES(yrs)',/,
&         ' SHOTNAME      ',4X,'ID',3X,'SLOC',8F10.0)
C     WRITE (16,300)(RECEP(I),I=1,NR)
300  FORMAT (' TRAVEL TIME(yrs) AT FIXED DISTANCES(m)',/,
&         ' SHOTNAME      ',4X,'ID',3X,'SLOC',8F10.0)
400  FORMAT (' PARTICLE LOCATION AT FIXED TIMES (yrs)',/,
&         ' SHOTNAME      ',4X,'ID',3X,'SLOC',8F10.0)
C     DO 10 M=1,NUMPATH
C     WRITE (CVAR,'(I4)') M
C     READ (CVAR,'(4A1)') K1,K2,K3,K4
C     IF (K1.EQ.' ') K1 = '0'
C     IF (K2.EQ.' ') K2 = '0'
C     IF (K3.EQ.' ') K3 = '0'
C     INFILE = 'P'//K1//K2//K3//K4//'.PTH'
C     INFILE2 = 'S'//K1//K2//K3//K4//'.DAT'
C     OPEN (19,FILE = INFILE2)
C     OPEN (13,FILE = INFILE)

```

```

      READ (17,201) IDPART,SNAME,SLOC
      NL = 0
      DO 20 L=1,2000
      READ (13,160,END=25) IP,TIME(L),DIST(L),IHSU(L)
      READ (19,*) X(L),Y(L),Z(L)
      IF (L.EQ.1) IPART = IP
      IF (IPART.NE.IDPART) WRITE (6,*) ' IPART AND IDPART NOT EQUAL',
&                                IPART,IDPART
      NL = NL + 1
20    CONTINUE
25    CLOSE (13)
      DO 28 J=1,NT
      TD(J) = 0.0
28    CONTINUE
      DO 29 J=1,NR
      RT(J) = 0.0
29    CONTINUE
      DO 30 J=1,NT
      IF (TIMES(J).GT.TIME(NL)) THEN
      TD(J) = -999.
      GO TO 30
      END IF
      DO 31 L=1,NL-1
      IF (TIMES(J).GT.TIME(L).AND.TIMES(J).LE.TIME(L+1)) THEN
      TD(J) = DIST(L) + (DIST(L+1)-DIST(L))*(TIMES(J)-TIME(L))
&          /(TIME(L+1) - TIME(L))
      TX(J) = X(L) + (X(L+1)-X(L))*(TIMES(J)-TIME(L))
&          /(TIME(L+1) - TIME(L))
      TY(J) = Y(L) + (Y(L+1)-Y(L))*(TIMES(J)-TIME(L))
&          /(TIME(L+1) - TIME(L))
      TZ(J) = Z(L) + (Z(L+1)-Z(L))*(TIMES(J)-TIME(L))
&          /(TIME(L+1) - TIME(L))
      END IF
31    CONTINUE
30    CONTINUE
201   FORMAT (I5,1X,A14,2X,A2)
      WRITE (15,202) SNAME,IDPART,SLOC,(TD(J),J=1,NT)
      WRITE (18,203) SNAME,IDPART,SLOC,(TX(J),TY(J),TZ(J),J=1,NT)
      DO 32 K=1,NR
      IF (RECEP(K).GT.DIST(NL)) THEN
      RT(K) = -999.
      GO TO 32
      END IF
      DO 33 L=1,NL-1
      IF (RECEP(K).GT.DIST(L).AND.RECEP(K).LE.DIST(L+1)) THEN
      RT(K) = TIME(L) + (TIME(L+1)-TIME(L))*(RECEP(K)-DIST(L))
&          /(DIST(L+1) - DIST(L))
      END IF
33    CONTINUE
32    CONTINUE
      WRITE (16,202) SNAME,IDPART,SLOC,(RT(J),J=1,NR)
      OUTFILE = 'H'//K1//K2//K3//K4//'.DAT'
      OPEN (14,FILE=OUTFILE)
      DO 34 L = 1,NL-1
      IF (IHSU(L).NE.IHSU(L+1)) THEN
      CHANGED = (DIST(L+1)+DIST(L))/2.0
      CHANGET = (TIME(L+1)+TIME(L))/2.0
      WRITE (14,150) CHANGED,CHANGET,IHSU(L),IHSU(L+1)
100   FORMAT (2I5)
150   FORMAT (2E20.10,2I5)
160   FORMAT (I5,1X,2(E20.10),I5)
202   FORMAT (1X,A14,I5,5X,A2,8F10.0)
203   FORMAT (1X,A14,I5,5X,A2,3(8F10.0))
      END IF
34    CONTINUE
      CLOSE (14)

```

10 CONTINUE
 STOP
 END

```

C
C
C-----
      PARAMETER(NPT=30000,NT=2000,NPP=1400,
2 NPD=20,NPATH=150)
      CHARACTER*80 FNAMEI,DUMMY, TITLE
      CHARACTER*32 LINE6
      CHARACTER*20 LINE(5)
      CHARACTER*12 FNAMEO,LABCOR
      CHARACTER*10 LABEL(NPD),LABELJ
      CHARACTER*5 FNAMEO1
      CHARACTER*4 FNAMEO3,FNAMEO4,FNAMEO5
      CHARACTER*2 LABEL2(4)
      DIMENSION X(NPT),Y(NPT),Z(NPT),
1          DIST(NT),DT(NT),VEL(NT),XX(NPP),TDIST(NT+1),
2          VELD(NT),VXX(NPP),IDIST(126,20,4),
3          DMEAN(126,20,4),DSTD(126,20,4),DELT(NPT)
      DIMENSION II(NPT),JJ(NPT),KK(NPT),IPART(NPT),IZSEN(40,4),
1          IBEG(NPATH),IEND(NPATH),KG(NPT),JDIST(100),
2          IBEGZ(100),IENDZ(100),KGZN(100),NPOBS(100),
3          IZCTR(20),IZINC(20)
      DIMENSION DELX(NT),DELY(NT),DELZ(NT),IIR(NT),JJR(NT),KKR(NT),
1          IIP(NPP),JJP(NPP),KKP(NPP),QX(NPP),QY(NPP),QZ(NPP),
2          KGP(NPP),KGR(NT)
C-----
C
C===== STATEMENTS TO OPEN INPUT/OUTPUT FILES =====
      WRITE(*,5)
5      FORMAT(' Enter MAIN Input Filename: ',%)
      READ(*,15) FNAMEI
15     FORMAT(A80)
      OPEN (UNIT=7,FILE=FNAMEI,FORM='FORMATTED',STATUS='OLD')
      OPEN (UNIT=8,FILE='PATHVEL.DAT',FORM='FORMATTED',STATUS='OLD')
C-----
C..... CARD TYPE 1
      IN=7
      INP=8
      IOUT=12
      IZERO=0
      IONE=1
C***** LABELS
      FNAMEO1='MC_CO'
      FNAMEO3='.DAT'
      FNAMEO4='.VEL'
      FNAMEO5='.IJK'
      LINE(1)='/MC,ISEED'
      LINE(2)='/NX,DX,NT,DT,SCL_PRM'
      LINE(3)='/INBOUN'
      LINE(4)='/NUMOBS'
      LINE(5)='/NOBSLOC'
      LINE6='/NPARD,NSPEC,NFSP.NSENS,NLHS,NRC'
      LABEL(1)='/C_SRCE'
      LABEL(2)='/GAMMA'
      LABEL(3)='/CINIT'
      LABEL(4)='/SINIT'
      LABEL(5)='/VEL'
      LABEL(6)='/DISP'
      LABEL(7)='/DM'
      LABEL(8)='/TAU'
      LABEL(9)='/THETA1'
      LABEL(10)='/THETA2'
      LABEL(11)='/BWIDTH'
      LABEL(12)='/FWIDTH'
      LABEL(13)='/RHOWB'
      LABEL(14)='/RKD1'
      LABEL(15)='/RKD2'

```

```

        LABEL(16)=' /DIFF '
        LABEL(17)=' /LAMBDA1W '
        LABEL(18)=' /LAMBDA1S '
        LABEL(19)=' /LAMBDA2W '
        LABEL(20)=' /LAMBDA2S '
        LABEL2(1)=' KH '
        LABEL2(2)=' KV '
        LABEL2(3)=' LD '
        LABEL2(4)=' RC '
        LABELJ=' /XX_00_0 '
        LABCOR=' CORRMATX.DAT '
C        LINEL=' /IOU6,IOU7,IOU8,IOU9,IOU10,IOU11,IOU12,IOU20 '
C*****
        READ(IN,35) TITLE
        READ(IN,*) MC,ISEED
        READ(IN,*) IBOUN
        READ(IN,*) NPROP,NEL
        READ(IN,*) PECL,COUR,DISTMXI,DELTMXI,TTIME
        READ(IN,*) NSPEC,NFSP,NSENS,NLHS,NRC
        READ(IN,*) NUMOBS
        READ(IN,*) (NPOBS(I),I=1,NUMOBS)
        DO 12 N=1,NPROP
            READ (IN,*) IPAR,ISPEC,IHSU,IDIST(IPAR,IHSU,ISPEC),
1            DMEAN(IPAR,IHSU,ISPEC),DSTD(IPAR,IHSU,ISPEC)
12 CONTINUE
        READ(IN,*) IOU6,IOU7,IOU8,IOU9,IOU10,IOU11,IOU12,IOU20
        READ(IN,*) NZS
        DO 14 IZ=1,NZS
            READ(IN,*) IZCTR(IZ),IZINC(IZ)
            READ(IN,*) (IZSEN(I,IZ),I=1,IZCTR(IZ))
14 CONTINUE
C
        READ(INP,35) DUMMY
        I=0
        IBEG(1)=1
        NPART=1
22 CONTINUE
        I=I+1
        READ(INP,*,END=32) IPART(I),X(I),Y(I),Z(I),DELT(I),KG(I),
1            II(I),JJ(I),KK(I)
        IF (I.GT.1) THEN
            IF (IPART(I).NE.IPART(I-1)) THEN
                NPART=NPART+1
                IEND(NPART-1)=I-1
                IBEG(NPART)=I
            ENDIF
        ENDIF
        GO TO 22
32 CONTINUE
        IEND(NPART)=I-1
        DO 42 IPS=1,NPART
            WRITE(FNAMEO,8) FNAMEO1,IPS,FNAMEO3
            IF(IPS.LT.10) FNAMEO(7:7)='0'
            IF(IPS.LT.100) FNAMEO(6:6)='0'
            OPEN (UNIT=IOUT,FILE=FNAMEO,FORM='FORMATTED',STATUS='UNKNOWN')
            WRITE(IOUT,35) TITLE
            IBEGIP=IBEG(IPS)+1
            IENDIP=IEND(IPS)
            TDIST(1)=0.0
            ILC=0
            VELMAX=-1.0E10
            DO 52 IL=IBEGIP,IENDIP
                DO 52 ISPEC=1,NSPEC
                    ILC=ILC+1
                    DT(ILC)=DELT(IL)
                    IF (ISPEC.EQ.1) THEN

```

```

        DELX(ILC)=X(IL)-X(IL-1)
        DELY(ILC)=Y(IL)-Y(IL-1)
        DELZ(ILC)=Z(IL)-Z(IL-1)
        DIST(ILC)=SQRT(DE LX(ILC)**2+DE LY(ILC)**2+
1          DELZ(ILC)**2)
        IIR(ILC)=II(IL)
        JJR(ILC)=JJ(IL)
        KKR(ILC)=KK(IL)
        TDIST(ILC+1)=TDIST(ILC)+DIST(ILC)
        ENDIF
        KGR(ILC)=KG(IL-1)
        IHSU=KGR(ILC)
ckrr 4/9/96
        effpor = dmean(9,ihsu,ispec)
        if (idist(9,ihsu,ispec).eq.2) then
            effpor = exp(2.303*dmean(9,ihsu,ispec) + 0.5*
&              (2.303*dst d(9,ihsu,ispec)**2))
        end if
ckrr
        RET=1.0+DMEAN(13,IHSU,ISPEC)*DMEAN(14,IHSU,ISPEC)/
1        effpor
        VEL(ILC)=DIST(ILC)*365.25/(DT(ILC)*RET*effpor)
        VELD(ILC)=VEL(ILC)*RET*effpor
        IF (VEL(ILC).GT.VELMAX) VELMAX=VEL(ILC)
52      CONTINUE
        DISP=DMEAN(6,1,1)
C        IF (DMEAN(6,1,1).LT.0.0) DISP=-DMEAN(6,1,1)*TDIST(ILC+1)
        DXMAX=PECL*DISP
        DISTMX=MIN(DISTMXI,TDIST(ILC+1))
        DX=DISTMX/NEL
        IF (DX.GT.DXMAX) THEN
            DX=DXMAX
            NEL=DISTMX/DX
        ENDIF
        NP=NEL+1
        DTRUN=DX*COUR/VELMAX
        DTRUN=MIN(DTRUN,DELTMXI)
ckrr additional constraint to prevent large number of time steps
c        dtrun = max(dtrun,0.0625)
        NTT=2+TTIME/DTRUN
C        IF(NTT.GT.IOU10) THEN
C            NUMB=NTT/IOU10
C        ELSE
C            NUMB=1
C        ENDIF
C        IOU10N=-NUMB
        VD=VELD(1)
        VXX(1)=VD
        QX(1)=VD*DELX(1)/DIST(1)
        QY(1)=VD*DELY(1)/DIST(1)
        QZ(1)=VD*DELZ(1)/DIST(1)
        IIP(1)=IIR(1)
        JJP(1)=JJR(1)
        KKP(1)=KKR(1)
        KGP(1)=KGR(1)
        XX(1)=0
        ILC=1
        DO 72 IP=2,NP
            XX(IP)=XX(IP-1)+DX
CKRR CHANGED IP-1 TO IP
            XCHECK=XX(IP)+0.5*DX
CKRR
11      CONTINUE
            IF (XCHECK.GT.TDIST(ILC+1)) THEN
                ILC=ILC+1
                IF (DT(ILC).LE.0.0) THEN

```

```

        ILC=ILC+1
      ENDIF
      GO TO 11
    ELSE
      VD=VELD( ILC)
      VXX( IP)=VD
      KGP( IP)=KGR( ILC)
      QX( IP)=VD*DELX( ILC)/DIST( ILC)
      QY( IP)=VD*DELY( ILC)/DIST( ILC)
      QZ( IP)=VD*DELZ( ILC)/DIST( ILC)
      IIP( IP)=IIR( ILC)
      JJP( IP)=JJR( ILC)
      KKP( IP)=KKR( ILC)
    ENDIF
  72 CONTINUE
CKRR 4/5/96
      IF (DISTMX.LT.DISTMXI) THEN
        VD=VXX(NP-1)
        VXX(NP)=VXX(NP-1)
        KGP(NP)=KGP(NP-1)
        QX(NP)=QX(NP-1)
        QY(NP)=QY(NP-1)
        QZ(NP)=QZ(NP-1)
        IIP(NP)=IIP(NP-1)
        JJP(NP)=JJP(NP-1)
        KKP(NP)=KKP(NP-1)
      END IF
CKRR
      CALL CONDENSE (1,NP,KGP,KGZN,NZNS,IBEGZ,IENDZ)
      ZERO=0.0
      AMONE=-1.0
CRKW NEXT CALC NOT CORRECT.  WORKS FOR NSPEC=1 ONLY
      ICARD=4*NSPEC*NZNS+NEL+12*NZNS+NFSP
ckrr modified to allow for only two cards for input concentration
      icard = icard - 2*nzns + 2*2
ckrr 3/20/96
      WRITE(IOUT,6) MC,ISEED,LINE(1)
      WRITE(IOUT,16) NP,DX,NTT,DTRUN,ZERO,AMONE,LINE(2)
      WRITE(IOUT,26) IBOUN,LINE(3)
      WRITE(IOUT,26) NUMOBS,LINE(4)
      WRITE(IOUT,27) (NPOBS(I),I=1,NUMOBS)
      WRITE(IOUT,36) ICARD,NSPEC,NFSP,NSENS,NLHS,NRC,LINE6
      KCTR=0
      NPID=0
      DO 91 J=1,4
        CALL OUT(J,IBEGZ,IENDZ,IDIST,DMEAN,DSTD,NSPEC,NZNS,KGZN,
1 IOUT,KCTR,JDIST,NPID,LABEL)
91 CONTINUE
      IFIVE=5
      DO 92 J=1,NEL
        NPID=NPID+1
        WRITE(IOUT,46) IFIVE,IONE,J,J,IZERO,VXX(J),ZERO,LABEL(5),NPID
92 CONTINUE
      DO 93 J=6,17
        CALL OUT(J,IBEGZ,IENDZ,IDIST,DMEAN,DSTD,NSPEC,NZNS,KGZN,
1 IOUT,KCTR,JDIST,NPID,LABEL)
93 CONTINUE
      JEND=17
      ICHECK=0
      DO 194 JJJ=1,NZS
        JBEG=JEND+1
        JEND=JEND+IZCTR(JJJ)
        JCOUNT=0
        JZONE=0
        LABELJ(2:3)=LABEL2(JJJ)
      do 94 J=JBEG,JEND

```



```

JCOUNT=JCOUNT+1
IZ=IZSEN(JCOUNT,JJJ)
IF (IZ.EQ.JZONE) THEN
  INCNT=INCNT+1
  IF (IZINC(JJJ).GT.0) THEN
    KCTR=KCTR+1
    JDIST(KCTR)=IDIST(J,1,1)
  ENDIF
  NKCTR=-KCTR
  LABELJ(8:8)=CHAR(48+INCNT)
ELSE
  KCTR=KCTR+1
  NKCTR=-KCTR
  JDIST(KCTR)=IDIST(J,1,1)
  INT1=48+IZ/10
  INT2=48+MOD(IZ,10)
  LABELJ(5:5)=CHAR(INT1)
  LABELJ(6:6)=CHAR(INT2)
  LABELJ(8:8)=CHAR(49)
  INCNT=1
  JZONE=IZ
ENDIF
NPID=NPID+1
write(iout,46) j,ione,ione,ione,NKCTR,dmean(j,1,1),
1 dstd(j,1,1),LABELJ,NPID
94 continue
194 CONTINUE
WRITE(IOUT,56) IOU6,IOU7,IOU8,IOU9,IOU10,IOU11,IOU12,IOU20
C
CLOSE(IOUT)
WRITE(FNAMEO,8) FNAMEO1,IPS,FNAMEO4
IF(IPS.LT.10) FNAMEO(7:7)='0'
IF(IPS.LT.100) FNAMEO(6:6)='0'
OPEN (UNIT=IOUT,FILE=FNAMEO,FORM='UNFORMATTED',STATUS='UNKNOWN')
WRITE(IOUT) (QX(J),J=1,NEL)
WRITE(IOUT) (QY(J),J=1,NEL)
WRITE(IOUT) (QZ(J),J=1,NEL)
CLOSE (IOUT)
WRITE(FNAMEO,8) FNAMEO1,IPS,FNAMEO5
IF(IPS.LT.10) FNAMEO(7:7)='0'
IF(IPS.LT.100) FNAMEO(6:6)='0'
OPEN (UNIT=IOUT,FILE=FNAMEO,FORM='FORMATTED',STATUS='UNKNOWN')
DO 95 J=1,NEL
95 WRITE(IOUT,'(4I5)') J,IIP(J),JJP(J),KKP(J)
CLOSE (IOUT)
C
LABCOR(8:8)=CHAR(48+IPS)
OPEN (UNIT=9,FILE=LABCOR,FORM='FORMATTED',STATUS='UNKNOWN')
DO 995 IK=1,KCTR
  WRITE(9,996) IK,JDIST(IK)
995 CONTINUE
996 FORMAT(2I5)
CLOSE (9)
C
42 CONTINUE
C
C***** FORMAT STATEMENTS
C
6 FORMAT(2I5,50X,A20)
16 FORMAT(I5,1X,F14.3,1X,I8,1X,F15.5,1X,F3.0,1X,F5.1,5X,A20)
26 FORMAT(I5,55X,A20)
27 FORMAT (10I5)
36 FORMAT(6I5,15X,A32)
C 46 FORMAT(5I5,2F14.5,10X,A20)
46 FORMAT(I3,1X,I1,2(1X,I7),1X,I4,2(1X,G14.5),14X,A10,' (',I4,')')
56 FORMAT(8I5,2X,A44)

```

```

57 FORMAT(4I5,3G14.5)
8  FORMAT(A5,I3,A4)
35 FORMAT(A80)
STOP
END
SUBROUTINE CONDENSE (IBEGIP,IENDIP,KG,KGZN,NZNS,IBEGZ,IENDZ)
DIMENSION KG(1400),IBEGZ(100),IENDZ(100),KGZN(100)
J=1
NZNS=1
IOLD=IBEGIP
KGZN(1)=KG(IBEGIP)
IBEGZ(J)=IBEGIP
DO 10 I=IBEGIP,IENDIP
IF (KG(I).NE.KG(IOLD)) THEN
    IBEGZ(J+1)=I
    IENDZ(J)=I-1
    J=J+1
    KGZN(J)=KG(I)
    IOLD=I
ENDIF
10 CONTINUE
NZNS=J
IENDZ(J)=IENDIP-IBEGIP+1
RETURN
END

SUBROUTINE OUT(IPAR,IBEGZ,IENDZ,IDIST,DMEAN,DSTD,NSPEC,NZNS,KGZN,
1 IOUT,KCTR,JDIST,NPID,LABEL)
DIMENSION IBEGZ(100),IENDZ(100),IDIST(126,20,4),DMEAN(126,20,4),
1 DSTD(126,20,4),KGZN(100),ICTRSV(20,4),KCTRSV(20,4),ICTR(4),
2 JDIST(100),IZNID(4)
CHARACTER*10 LABEL(20)
DO 10 IZN=1,NZNS
DO 10 ISPEC=1,NSPEC
IF (IZN.EQ.1) IZNID(ISPEC)=0
IF (IPAR.EQ.3.or.ipar.eq.4) THEN
    IENDF=6
    NHSU=KGZN(IZN)
    ID=IDIST(IPAR,NHSU,ISPEC)
    NPID=NPID+1
    IF (ID.GT.0) THEN
        KCTR=KCTR+1
        JDIST(KCTR)=ID
        NKCTR=-KCTR
        WRITE(IOUT,46) IPAR,ISPEC,IBEGZ(1),IENDF,
1        NKCTR,DMEAN(IPAR,NHSU,ISPEC),DSTD(IPAR,NHSU,ISPEC),
1        NHSU,LABEL(IPAR),NPID
    ELSE
        WRITE(IOUT,46) IPAR,ISPEC,IBEGZ(1),IENDF,
1        IDIST(IPAR,NHSU,ISPEC),
1        DMEAN(IPAR,NHSU,ISPEC),DSTD(IPAR,NHSU,ISPEC),
1        NHSU,LABEL(IPAR),NPID
    ENDIF
    IDI=0
    dmeano = 0.0
    NPID=NPID+1
    WRITE(IOUT,46) IPAR,ISPEC,IENDF+1,IENDZ(NZNS),
1    IDI,
1    DMEANO,DSTD(IPAR,NHSU,ISPEC),
1    NHSU,LABEL(IPAR),NPID
    RETURN
ENDIF
NHSU=KGZN(IZN)
ID=IDIST(IPAR,NHSU,ISPEC)
IF (ID.GT.0) THEN
    IZNID(ISPEC)=IZNID(ISPEC)+1

```

```

      IF ( IZNID( ISPEC ).EQ.1 ) THEN
        ICTR( ISPEC )=1
        ICT=ICTR( ISPEC )
        ICTRSV( ICT, ISPEC )=NHSU
        KCTR=KCTR+1
        KCTRSV( ICT, ISPEC )=KCTR
        NKCTR=-KCTR
        JDIST( KCTR )=ID
      ELSE
        DO 12 ICK=1, ICTR( ISPEC )
          IF ( NHSU.EQ. ICTRSV( ICK, ISPEC ) ) THEN
            NKCTR=-KCTRSV( ICK, ISPEC )
            GO TO 13
          ENDIF
12      CONTINUE
          ICTR( ISPEC )=ICTR( ISPEC )+1
          ICT=ICTR( ISPEC )
          ICTRSV( ICT, ISPEC )=NHSU
          KCTR=KCTR+1
          KCTRSV( ICT, ISPEC )=KCTR
          NKCTR=-KCTR
          JDIST( KCTR )=ID
13      CONTINUE
        ENDIF
        NPID=NPID+1
        WRITE ( IOUT, 46 ) IPAR, ISPEC, IBEGZ( IZN ), IENDZ( IZN ),
1      NKCTR, DMEAN( IPAR, NHSU, ISPEC ), DSTD( IPAR, NHSU, ISPEC ),
1      NHSU, LABEL( IPAR ), NPID
      ELSE
        NPID=NPID+1
        WRITE ( IOUT, 46 ) IPAR, ISPEC, IBEGZ( IZN ), IENDZ( IZN ),
1      IDIST( IPAR, NHSU, ISPEC ),
1      DMEAN( IPAR, NHSU, ISPEC ), DSTD( IPAR, NHSU, ISPEC ),
1      NHSU, LABEL( IPAR ), NPID
      ENDIF
10    CONTINUE
46  FORMAT( I3, 1X, I1, 2( 1X, I7 ), 1X, I4, 2( 1X, G14.5 ), 7X, I5, 2X, A10, ' ( ', I4,
1      ' ) ' )
      RETURN
      END

```

```

C*****
C
C          qsens02.for
C
C      Written by James R. Erickson, GeoTrans, Inc.
C
C      November 16,1995
C
C      Modified by Ken R. Rehfeldt, March 18, 1996, to allow for a greater
C      number of sensitivity variables and a larger model dimension
C
C      Version 2.0
C
C      Program to read a binary cell by cell darcy flux file from modflow
C      output files and calculate the normalized sensitivity
C
C      Input Files Required:
C
C          grids.dat--Contains modflow grid spacing from BCF file
C          m_eleva.dat--Contains the elevations of the top of each model
C                      layer
C          sens_nam.fil--Contains name of modflow files with cell fluxes
C                      for the sensitivity runs
C          sav.flx--Contains the modflow cell fluxes for the calibrated
C                   model base case
C
C      Output Files Required:
C
C          The output files have an "s" concatenated to the front of the
C          name of the input files
C
C      Key Parameters:
C
C          buff(j,i,k)--Fluxes from modflow files
C          qx(j,i,k)--Fluxes for the right face (ie positive x direction)
C          qy(j,i,k)--Fluxes for the front face (ie positive y direction)
C          qz(j,i,k)--Fluxes for the lower face (ie positive z direction)
C          sensx(j,i,k)--Flux sensitivities in x direction
C          sensy(j,i,k)--Flux sensitivities in y direction
C          sensz(j,i,k)--Flux sensitivities in z direction
C
C      RKW 3/14/96 DIMENSIONED INCREASED FOR FINER GRID
C      RKW 10/18/96 DIMENSIONED FOR 68 COLUMNS, WAS FOR 67 "QSENS03.FOR"
C*****
C
C      Main Program
C
C
C          dimension qx(69,76,20),qy(68,77,20),qz(68,76,21),
C          1buff(68,76,20),qxbase(69,76,20),qybase(68,77,20),
C          2qzbase(68,76,20)
C          dimension sensx(69,76,20),sensy(68,77,20),sensz(68,76,21)
C          dimension delr(68),delc(76),elev(21)
C          dimension filein(120),propor(120)
C
C          character*30 filein,fileout,filein1,fileout1,sav1
C
C          ncol=68
C          nrow=76
C          nlay=20
C          nrpl=nrow+1
C          ncpl=ncol+1
C          nlpl=nlay+1
C          iu=5
C
C
C          read the model grid (column and rows)
C

```

```

        open (iu,file='grids.dat')
        read (iu,*) dum
        read (iu,*) dum
        read (iu,*) (delr(i),i=1,ncol)
        read (iu,*) (delc(i),i=1,nrow)
        close (iu)
c
c      read the top elevations of each layer
c
        open (iu,file='m_elev.dat')
        read (iu,*) dum
        do 10 i=1,nlay+1
            read (iu,*) elev(i)
10 continue
        close (iu)

c
c      Read file that contains the file names for sensitivity fluxes
c
        open (iu,file='sens_nam.fil')
        j=1
        30 read (iu,300,end=50) filein(j),propor(j)
300 format (a30,f10.2)
        j=j+1
        go to 30
50 nfile=j-1
        close (iu)

c
c      Read file with calibrated fluxes; (ie base case)
c
        savl = 'sav.flx'
        open (iu,file=savl, form='unformatted')
        call celflo (qxbase,qybase,qzbase,buff,delr,delc,ncol,nrow,nlay,
1          ncpl,nrpl,nlpl,elev,iu)
        close (iu)

c
c      Loop through each of the flux files and calculate sensitivities
c
        do 1000 i=1,nfile
            filein1 = filein(i)
            write (*,*) filein1
            fileout='s'//filein(i)
crkw changed to remove disk references
c          fileout1 = 'c:'//fileout
        200 open (iu,file=filein1,form='unformatted')
            open (9, file=fileout)
            proporsn = propor(i)

c
            call celflo (qx,qy,qz,buff,delr,delc,ncol,nrow,nlay,ncpl,nrpl,
1              nlpl,elev,iu)

c
c
            call sens (qx,qy,qz,ncol,nrow,nlay,ncpl,nrpl,nlpl,sensx,
1              sensy,sensz,qxbase,qybase,qzbase,proporsn)

c
            close (iu)
            close (9)
        1000 continue
        stop
        end

c
c *****
c      Subroutine for assigning the cell by cell darcy flows to the x, y
c      and z directions and normalizing fluxes for each cells
c
c      Subroutine celflo (qx,qy,qz,buff,delr,delc,ncol,nrow,nlay,ncpl,

```

```

1          nrpl,nlpl,elev,iu)
c
common/test/ label
dimension qx(68,76,20),qy(67,77,20),qz(67,76,21),
1buff(67,76,20)
dimension delr(67),delc(76),elev(21)
c
Character*16 label

c
c Read x face flows
c
label= 'FLOW RIGHT FACE '
call rdebug (buff,ncol,nrow,nlay,iu)
do 20 k=1,nlay
do 20 i=1,nrow
qx(1,i,k) =0.0
do 20 j=1,ncol
20 qx(j+1,i,k)=buff(j,i,k)/(delc(i)*(elev(k)-elev(k+1)))
c
c Read y face flows
c
label= 'FLOW FRONT FACE '
call rdebug (buff,ncol,nrow,nlay,iu)
do 30 k=1,nlay
do 30 j=1,ncol
qy(j,1,k) =0.0
do 30 i=1,nrow
30 qy(j,i+1,k)=-buff(j,i,k)/(delr(j)*(elev(k)-elev(k+1)))
c
c Read z face flows
c
label= 'FLOW LOWER FACE '
call rdebug (buff,ncol,nrow,nlay,iu)
do 40 i=1,nrow
do 40 j=1,ncol
qz(j,i,1) =0.0
do 40 k=1,nlay
40 qz(j,i,k+1)=-buff(j,i,k)/(delc(i)*delr(j))
return
end

c
c*****
c Subroutine for reading the cell by cell darcy flows
c
Subroutine rdebug (buff,ncol,nrow,nlay,iu)
common/test/ label
dimension buff(67,76,20)
Character*16 label,text

c
ipass=0
10 ipass=ipass+1
if(ipass.gt.2) then
write (*,5000) label,iu
5000 format (a16,'Budget Terms Were Not Found ',i3)
stop
endif

c
20 read (iu,end=30) kstep,kper,text,nc,nr,nl
if (text.eq.label) then
read (iu) buff
return
else
read (iu)
endif
go to 20

```

```

30 rewind (iu)
   go to 10
   end
c
c*****
c   Subroutine sens for calculating the sensitivities and writing the
c   output to input file with a "s" concatenated to filein (ie skh1.flx)
c
c   Subroutine sens (qx,qy,qz,ncol,nrow,nlay,ncpl,nrpl,nlpl,sensx,
1      sensy,sensz,qxbase,qybase,qzbase,proporsn)
c
c
c   dimension qx(68,76,20),qy(67,77,20),qz(67,76,21),
1qxbase(68,76,20),qybase(67,77,20),qzbase(67,76,21)
c   dimension sensx(68,76,20),sensy(67,77,20),sensz(67,76,21)
c
c   Character*16 text
c
c   Calculate X Face Sensitivities
c
c   text= 'FLOW RIGHT FACE '
c   do 20 k=1,nlay
c   do 20 i=1,nrow
c   do 20 j=1,ncpl
20 sensx(j,i,k)=(qx(j,i,k)-qxbase(j,i,k))/proporsn
c
c   Write Sensitivities for X Face
c
c   write (9,*) text
c   do 21 k=1,nlay
c   do 21 i=1,nrow
21 write (9,1000) (sensx(j,i,k),j=1,ncpl)
c
c   Calculate Y Face Sensitivities
c
c   text= 'FLOW FRONT FACE '
c   do 30 k=1,nlay
c   do 30 j=1,ncol
c   do 30 i=1,nrpl
30 sensy(j,i,k)=(qy(j,i,k)-qybase(j,i,k))/proporsn
c
c   Write Sensitivities for Y Face
c
c   write (9,*) text
c   do 31 k=1,nlay
c   do 31 i=1,nrpl
31 write (9,1000) (sensy(j,i,k),j=1,ncol)
c
c   Calculate Z Face Sensitivities
c
c   text= 'FLOW LOWER FACE '
c   do 40 i=1,nrow
c   do 40 j=1,ncol
c   do 40 k=1,nlpl
40 sensz(j,i,k)=(qz(j,i,k)-qzbase(j,i,k))/proporsn
c
c   Write Sensitivities for Z Face
c
c   write (9,*) text
c   do 41 k=1,nlpl
c   do 41 i=1,nrow
41 write (9,1000) (sensz(j,i,k),j=1,ncol)
1000 format(10g12.4)
c   return
c   end
c

```

```

C***** sensvec.for *****
C
C      Written by Richard K. Waddell, GeoTrans, Inc
C      November 17, 1995
C
C      Program to assign flux sensitivities obtained from the program
C      qsens.for to element nodal locations output from the program
C      mcastnts.for
C
C      Input files:
C
C          files.ijk--List of files with element nodal numbers and the
C                      i,j,k of modflow grid cells
C          files.sen--List of files with the flux sensitivities for each of
C                      the sensitivity runs from MODFLOW; the flux
C                      sensitivity files are output from qsens.for
C
C      Output files:
C
C          Concatenate .snx, .sny, and .snz on to the end of the
C          sensitivity file names obtained from files.sen
C
C*****
C
C      parameter (nels=1001,nrow=76,ncol=68,nlay=20,nak=25)
C      dimension ii(nels),jj(nels),kk(nels),npts(nak)
C      dimension dqx(ncol+1,nrow,nlay),dqy(ncol,nrow+1,nlay),
1      dqz(ncol,nrow,nlay+1)
C      dimension dqxv(nels,nak),dqyv(nels,nak),dqzv(nels,nak)
C      character*12 fileijk,filesen
C      open (10,file='files.ijk')
C      open (11,file='files.sen')
C      nijk=0
1      read (10,*,end= 100) fileijk
C      rewind (11)
C      nijk=nijk+1
C      open (12,file=fileijk)
C      i=1
10      read (12,*,end= 15) nel,jj(nel),ii(nel),kk(nel)
C      i=i+1
C      go to 10
15      npts(nijk)=i-1
C      close(12)
C      naks=0
17      read (11,*,end=60) filesen
C      naks=naks+1
C      open (12,file=filesen)
C      READ (12,1000)
C      do 20 k=1,nlay
C      do 20 i=1,nrow
20      read (12,1000) (dqx(j,i,k),j=1,ncol+1)
C      READ (12,1000)
C      do 30 k=1,nlay
C      do 30 i=1,nrow+1
30      read (12,1000) (dqy(j,i,k),j=1,ncol)
C      READ (12,1000)
C      do 40 k=1,nlay+1
C      do 40 i=1,nrow
40      read (12,1000) (dqz(j,i,k),j=1,ncol)
C      close (12)
C      do 50 i=1,npts(nijk)
C      it=ii(i)
C      jt=jj(i)
C      kt=kk(i)
C      dqxv(i,naks)=(dqx(jt,it,kt)+dqx(jt+1,it,kt))/2.

```



```

      dqyv(i,naks)=(dqy(jt,it,kt)+dqy(jt,it+1,kt))/2.
50    dqzv(i,naks)=(dqz(jt,it,kt)+dqz(jt,it,kt+1))/2.
      goto 17
60    iper=index(fileijk, '.')
      open (12,file=fileijk(1:iper-1)//'.snx',FORM='UNFORMATTED')
      open (13,file=fileijk(1:iper-1)//'.sny',FORM='UNFORMATTED')
      open (14,file=fileijk(1:iper-1)//'.snz',FORM='UNFORMATTED')
      do 70 i=1,npts(nijk)
70    write (12) (dqxv(i,j),j=1,naks)
      do 80 i=1,npts(nijk)
80    write (13) (dqyv(i,j),j=1,naks)
      do 90 i=1,npts(nijk)
90    write (14) (dqzv(i,j),j=1,naks)
      close (12)
      close (13)
      close (14)
      goto 1
100   stop
1000  format (10g12.4)
1010  format (1p20e12.4)
      end

```

```

C
C *****
C PREPROCESSOR FOR LATIN HYPERCUBE SAMPLE PROGRAM LHS
C BASED ON INPUT FILE FOR MC_TRCOR Ver 1.0
C *****
C
C Name: PRELHS
C Version: 1.0
C
C GeoTrans, Inc.
C 4330 S. Valley View Lane, Suite 114
C Las Vegas, NV 89103
C Ken Rehfeldt 702-737-7225
C
C GeoTrans, Inc.
C 46050 Manekin Place
C Suite 100
C Sterling, VA 20166
C Barry Lester 703-444-7000
C
C November 9, 1995
C
C Funding: DOE NTS Project Number 3705-695.
C
C
C MAIN Program functions:
C
C 1) Read in the main input file for MC_COR.
C 2) Calculate statistics, as necessary, for input to the
C Latin Hypercube Sampler LHS.FOR.
C 3) Read correlation coefficients for variables to be sampled by
C LHS.FOR.
C 4) Generate input file for LHS.FOR.
C
C
C
C CHARACTER*29 TITLE2
C CHARACTER*12 FNames
C LOGICAL LEX
C
C Primary arrays and matrices:
C
C
C **PSTAT(,) = Array to hold statistic information on each parameter
C distribution:
C
C PSTAT(I,1) = Type of distribution for parameter I
C = 0 Deterministic (characterized by the mean)
C = 1 Normal distribution (mean and std)
C = 2 Lognormal distribution (mean and std)
C = 3 Uniform distribution (max and min)
C = 4 Raw data distribution [ ** untested ** ]
C
C PSTAT(I,2) = Mean value for the parameter I; if a uniform
C distribution, then this is the minimum parameter
C value.
C
C PSTAT(I,3) = Standard deviation for the parameter I; if a uniform
C distribution, then this is the max parameter value.
C
C PSTAT(I,4) = Number of data points in the raw data distribution;
C Non-zero only for PSTAT(I,1)=4.
C
C

```

```

C**PDIST(,)      = The matrix for storing the raw distribution data;
C      PDIST(I,K) = Data point K for parameter i; K=1,...,PSTAT(I,4).
C
C-----
C-----
C
C
C Open Input and output files.
C
C
C Files are as follows:
C
C      UNIT 5 = Input data file.
C
C      UNIT 6 = Output file.
C
C      UNIT 7 = MC_CORR input file.
C
C
C      UNIT 8 = Input file generated for LHS.FOR (Preprocessor Output).
C
C-----
C
C      TITLE2=' (          ,          )'
C
C      WRITE (*, 5000)
5000 FORMAT(/////,
1          ' ++++++ ',
2          '      PRELHS.for      Version 1.0      6/29/96',
3          ' ++++++ ',
4          '      Enter name of the primary input file',
C
C
C      ITRY = 0
5010 CONTINUE
C      READ (*, '(A12)') F NAMES
C
C Clean-up (trim and convert to uppercase) file name.
C
C      CALL UPCASE (FNAME)
C
C      INQUIRE (FILE = F NAMES, EXIST = LEX)
C
C      IF ( .NOT. LEX) THEN
C          WRITE (*, 5020) F NAMES
5020      FORMAT(' Input data file ',A12,' does not exist',
1          ' Re-enter filename: ',
C          ITRY = ITRY + 1
C          IF (ITRY .EQ. 5) STOP
C          GO TO 5010
C      END IF
C
C      OPEN (5, FILE = F NAMES, STATUS = 'OLD')
C      TITLE2(3:14)=F NAMES
C
C
C      WRITE (*, 6000)
6000 FORMAT(/////,
1          '      Enter name of the output file',
2          ' ',
C
C
C      READ (*, '(A12)') F NAMES
C
C Clean-up (trim and convert to uppercase) file name.

```

```

C
C      CALL UPCASE (FNAME)
C
C      OPEN (6, FILE = FNames, STATUS = 'UNKNOWN')
C
C      WRITE (*, 7000)
7000  FORMAT(//////,
1  '      Where is the distribution input read from? ',/,
3  '      | 0 = distrib. read from general input file| ',/,
4  '      | 1 = distrib. read from MC_CORR input file| ',/,
5  T20, ' ', $)
C
C      READ (*, '(I5)') IFILE
C
C      IF (IFILE.EQ.1) THEN
C
C      WRITE (*, 7010)
7010  FORMAT(//////,
1  '      Enter name of the MC_CORR input file',//,T20, ' ', $)
C
C
C      ITRY = 0
7020  CONTINUE
C      READ (*, '(A12)') FNames
C
C Clean-up (trim and convert to uppercase) file name.
C
C      CALL UPCASE (FNAME)
C
C      INQUIRE (FILE = FNames, EXIST = LEX)
C
C      IF ( .NOT. LEX) THEN
C      WRITE (*, 7030) FNames
7030  FORMAT(' MC_CORR data file ',A12,' does not exist',//,
1  '      Re-enter filename: ', $)
C      ITRY = ITRY + 1
C      IF (ITRY.EQ. 5) STOP
C      GO TO 7020
C      END IF
C
C      OPEN (7, FILE = FNames, STATUS = 'OLD')
C      TITLE2(17:28)=FNames
C      ENDIF
C
C
C      WRITE (*, 8000)
8000  FORMAT(//////,
1  '      Enter name of the file to contain input to LHS.FOR',
2  //,T20, ' ', $)
C
C
C      READ (*, '(A12)') FNames
C
C Clean-up (trim and convert to uppercase) file name.
C
C      CALL UPCASE (FNAME)
C
C      OPEN (8, FILE = FNames, STATUS = 'UNKNOWN')
C
C
C Read in the input data from UNITS 5 and 6.
C
C      ++++++
C      CALL PROCESS (TITLE2,IFILE)
C
C      STOP

```

```

END
C
C
C
C
SUBROUTINE PROCESS (TITLE2,IFILE)
C
C
PARAMETER (NVARMX=222)
CHARACTER*80 TITLE,CARD,DUMMY
CHARACTER*43 TITLE1
CHARACTER*29 TITLE2
CHARACTER*18 PVAR(NVARMX),PUSDST,PLOAD
CHARACTER*14 PRSAMP
CHARACTER*12 PRSEED,PLUNIS,PC
CHARACTER*11 PLUNI,PTRIAG
CHARACTER*10 PLNOR
CHARACTER*9 PUNIS
CHARACTER*8 PUNI
CHARACTER*7 PNOR,POUT
CHARACTER*5 PNOBS,PBETA
CHARACTER*4 PDUM(3),PDATA,PHIST,PCORR

PARAMETER(PRSAMP='RANDOM SAMPLE ',PRSEED='RANDOM SEED ',
1      PLUNIS='LOGUNIFORM* ',PLUNI='LOGUNIFORM ',
2      PTRIAG='TRIANGULAR ',PLNOR='LOGNORMAL ',
3      PUNIS='UNIFORM* ',PUNI='UNIFORM ',PNOR='NORMAL ',
4      POUT='OUTPUT ',PNOBS='NOBS ',PBETA='BETA ',
5      PC='CORRELATION ',PUSDST='USER DISTRIBUTION ',
6      PDATA='DATA',PHIST='HIST',PCORR='CORR')

DIMENSION A(NVARMX),B(NVARMX),COR(NVARMX)
DIMENSION ICV(NVARMX),IDIST(NVARMX),ICOR(NVARMX),JCOR(NVARMX)
C Routine to read in the data and generate the input file for the Latin
C Hypercube Sample program (LHS.FOR).
C
C
C
C
<----- Record 1
READ (5, '(A43)') TITLE1
TITLE(1:8) ='TITLE - '
TITLE(9:51)=TITLE1
TITLE(52:80)=TITLE2
WRITE (6, 6000) TITLE
WRITE (8, 8000) TITLE
C
C TITLE is the title for the run, to be used for output.
C Read in control variables.
C
C
C
C
<----- Record 2
READ (5, *) NVAR,NSAM,ICORR
WRITE (6, 6010) NVAR,NSAM,ICORR,IFILE
WRITE (8,8010) PNOBS,NSAM
C
C
C
C
<----- Record 3
READ (5, *) IRAN,ISEED
C
C ISEED is the seed used for the random number generator;
```

```

C      Values should be positive integers.
C
C      WRITE (6, 6020) IRAN, ISEED
C
C      IF (IRAN.EQ.1) WRITE (8,8000) PRSAMP
C      WRITE (8,8020) PRSEED, ISEED
C
C      READ (5, *) IOUT1,IOUT2,IOUT3          <----- Record 4
C      WRITE (6, 6030) IOUT1,IOUT2,IOUT3
C
C      IOUTx = Output control for general output file.
C      IOUTx = 0, Do not output
C      IOUTx = 1, Output
C
C      IPAR=0
C      IF (IOUT1.NE.0) THEN
C          IPAR=IPAR+1
C          PDUM(IPAR)=PDATA
C      ENDIF
C      IF (IOUT2.NE.0) THEN
C          IPAR=IPAR+1
C          PDUM(IPAR)=PHIST
C      ENDIF
C      IF (IOUT3.NE.0) THEN
C          IPAR=IPAR+1
C          PDUM(IPAR)=PCORR
C      ENDIF
C      WRITE(8,8030) POUT,(PDUM(ID),ID=1,IPAR)
C
C      IF (IFILE.EQ.0) THEN                    <----- Record 5
C          READ (5, *) NCARDS
C          WRITE (6,6040) NCARDS
C
C          DO 2000 IC=1,NCARDS
C              READ (5,5000) CARD
C              WRITE (6,6000) CARD
C              WRITE (8,8000) CARD
2000      CONTINUE
C
C      ELSE
C
C      WRITE (6,6050)                          <----- Record 5
C      DO 2010 IV=1,NVAR
C          READ (5,*) I,IDIST(I)
C          WRITE (6,6060) I,IDIST(I)
C          ICV(IV)=0
2010      CONTINUE
C
C      DO 2020 IDUM=1,4                          <----- MC_CORR Records
C          READ (7,7000) DUMMY
2020      CONTINUE
C          READ (7,*) NOB
C          READ (7,*) (IDUMMY,I=1,NOB)
C          READ (7,*) NCARDS
C          WRITE (6,6070) NCARDS
C          DO 2030 IC=1,NCARDS
C              READ (7,*) IDUM1,IDUM2,IDUM3,IDUM4,IVAR,AMEAN,SIGMA
C              IF (IVAR.LT.0) THEN

```

```

      IVAR=ABS(IVAR)
      IF (ICV(IVAR).EQ.0) THEN
        ICV(IVAR)=1
        ID=IDIST(IVAR)
        IF (ID.EQ.1) THEN
          A(IVAR)=AMEAN-3.09*SIGMA
          B(IVAR)=AMEAN+3.09*SIGMA
          PLOAD= '
          PLOAD(1:7)=PNOR
          PVAR(IVAR)=PLOAD
          WRITE (6,6080) IDUM1,IDUM2,IDUM3,IDUM4,IVAR,AMEAN,
1              SIGMA,A(IVAR),B(IVAR)
          ELSEIF (ID.EQ.2) THEN
            ALN=AMEAN-3.09*SIGMA
            BLN=AMEAN+3.09*SIGMA
            A(IVAR)=10.0**ALN
            B(IVAR)=10.0**BLN
            PLOAD= '
            PLOAD(1:10)= PLNOR
            PVAR(IVAR)= PLOAD
            WRITE (6,6080) IDUM1,IDUM2,IDUM3,IDUM4,IVAR,AMEAN,
1              SIGMA,A(IVAR),B(IVAR)
            ELSEIF (ID.EQ.3) THEN
              A(IVAR)=AMEAN
              B(IVAR)=SIGMA
              PLOAD= '
              PLOAD(1:8)= PUNI
              PVAR(IVAR)= PLOAD
              WRITE (6,6080) IDUM1,IDUM2,IDUM3,IDUM4,IVAR,AMEAN,
1              SIGMA,A(IVAR),B(IVAR)
            ELSE
              WRITE(*,1000) IVAR
              WRITE(6,6090) IVAR
              WRITE(8,8040) IVAR
            ENDIF
          ENDIF
        CONTINUE
2030      DO 2040 IV=1,NVAR
          ICHECK=0
          IF (ICV(IV).EQ.0) THEN
            WRITE(*,1010) IV
            WRITE(6,6100) IV
            WRITE(8,8050) IV
            ICHECK=1
          ENDIF
          IF (ICHECK.NE.0) STOP
2040      CONTINUE
          DO 2050 IV=1,NVAR
            WRITE(8,8000) PVAR(IV)
            WRITE(8,8060) A(IV),B(IV)
2050      CONTINUE
        ENDIF
C
C
C
      IF(ICORR.NE.0) THEN

        WRITE(8,8000) PC
        READ(5,*) NCOV
        WRITE(6,6110) NCOV
        WRITE(8,8070) NCOV
        DO 2060 INC=1,NCOV
          READ(5,*) ICOR(INC),JCOR(INC),COR(INC)
          WRITE(6,6120) ICOR(INC),JCOR(INC),COR(INC)

```

<----- Record 7

```

2060     CONTINUE
        WRITE (8,8080) (ICOR(INC),JCOR(INC),COR(INC),INC=1,NCOV)
        ENDIF

C
C -----FORMAT STATEMENTS-----
C
1000 FORMAT('*** ERROR*** IN VARIABLE# ',I4,/,
1       ' ONLY DISTRIBUTIONS 1 (NORMAL), 2 (LOGNORMAL),',/,
2       ' AND 3 (UNIFORM) ARE ALLOWED IN THIS OPTION',/)
1010 FORMAT('*** ERROR*** VARIABLE# ',I4,' IS NOT DEFINED',/)
5000 FORMAT(A)
6000 FORMAT(A)
6010 FORMAT(//,10X,'*** Record 2 ***',/,
1 ' Number of variables to be sampled..... (NVAR)..=',I4,/,
2 ' Number of samples to be generated..... (NSAM)..=',I4,/,
3 ' Correlation input..... (ICORR)..=',I4,/,
4 '          | 0 = correlation not considered | ',/,
5 '          | 1 = correlation matrix input   | ',/,
6 ' Distribution input..... (IFILE)..=',I4,/,
7 '          | 0 = distrib. in general input file | ',/,
8 '          | 1 = distrib. read from MC_CORR input | ',/)
6020 FORMAT(//,10X,'*** Record 3 ***',/,
1 ' Sampling type.....(IRAN)..=',I4,/,
2 '          | 0 = Latin Hypercube Sampling | ',/,
3 '          | 1 = Random Sampling          | ',/,
4 ' Seed used for the random number generator....(ISEED)..=',I12,/)

6030 FORMAT (//,10X,'*** Record 4 ***',/,,' Output desired:',/,
1 ' Generated sample output..... (IOUT1)..=',I4,/,
2 '          | 0 = no | ',/,
3 '          | 1 = yes | ',/,
4 ' Histogram output..... (IOUT2)..=',I4,/,
5 '          | 0 = no | ',/,
6 '          | 1 = yes | ',/,
7 ' Correlation output..... (IOUT3)..=',I4,/,
8 '          | 0 = no | ',/,
9 '          | 1 = yes | ',/)
6040 FORMAT(//,10X,'*** Record 5 ***',/,
1 ' Number of parameter cards to be read in.....(NCARDS)..=',I6,
2 //)
6050 FORMAT(//,10X,'VARIABLE ID DISTRIBUTION',/10X,25('-'),/)
6060 FORMAT(12X,I5,9X,I5)
6070 FORMAT(//,10X,'*** MC_CORR Input Records***',/,
1 ' Number of parameter cards to be read in.....(NCARDS)..=',I6,
2 //,2X,'PARID SP# RANGE DIST AMEAN SIGMA ',
3 ' A B',/2X,80('-'))
6080 FORMAT(2X,I5,1X,I2,1X,I5,' - ',I5,1X,I3,2X,4G13.6)
6090 FORMAT('*** ERROR*** IN VARIABLE# ',I4,/,
1 ' ONLY DISTRIBUTIONS 1 (NORMAL), 2 (LOGNORMAL),',/,
2 ' AND 3 (UNIFORM) ARE ALLOWED IN THIS OPTION',/)
6100 FORMAT('*** ERROR*** VARIABLE# ',I4,' IS NOT DEFINED',/)
6110 FORMAT(//,10X,'*** Record 7 ***',/,
1 ' Number of correlation coefficients to be read in. (NCOV)..=',I5,
2 //,' ROW COL CORRELATION',/2X,22('-'))
6120 FORMAT(2X,I4,1X,I4,1X,E11.4)
7000 FORMAT(A)
8000 FORMAT(A)
8010 FORMAT(A5,I5)
8020 FORMAT(A12,I12)
8030 FORMAT(A7,3(1X,A4))
8040 FORMAT('*** ERROR*** IN VARIABLE# ',I4,/,
1 ' ONLY DISTRIBUTIONS 1 (NORMAL), 2 (LOGNORMAL),',/,
2 ' AND 3 (UNIFORM) ARE ALLOWED IN THIS OPTION',/)
8050 FORMAT('*** ERROR*** VARIABLE# ',I4,' IS NOT DEFINED',/)

```



```
8060 FORMAT(2G20.5)
8070 FORMAT(I5)
8080 FORMAT(4(2I4,E12.4))
      RETURN
      END
```

```
C*****  
C LATIN HYPERCUBE OR RANDOM SAMPLING PROGRAM  
C<><><><><><><><><><><><><><><><><><><><><><><><><><><><><>  
C          ISSUED BY SANDIA NATIONAL LABORATORIES  
C          A PRIME CONTRACTOR TO THE  
C          UNITED STATES DEPARTMENT OF ENERGY  
C<><><><><><><><><><><><><><><><><><><><><><><><><><><><><>  
C THIS REPORT WAS PREPARED AS AN ACCOUNT OF WORK SPONSORED BY THE  
C UNITED STATES GOVERNMENT. NEITHER THE UNITED STATES NOR THE  
C UNITED STATES DEPARTMENT OF ENERGY  
C NOR ANY OF THEIR EMPLOYEES, NOR ANY OF THEIR CONTRACTORS,  
C SUBCONTRACTORS, OR THEIR EMPLOYEES, MAKES ANY WARRANTY, EXPRESS  
C OR IMPLIED, OR ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY  
C FOR THE ACCURACY, COMPLETENESS OR USEFULNESS OF ANY INFORMATION,  
C APPARATUS, PRODUCT OR PROCESS DISCLOSED, OR REPRESENTS THAT ITS  
C USE WOULD NOT INFRINGE PRIVATELY OWNED RIGHTS.  
C<><><><><><><><><><><><><><><><><><><><><><><><><><><><><>  
C AUTHORS      SANDIA NATIONAL LABORATORIES, ALBUQUERQUE, NM 87185  
C R. L. IMAN              PHONE (505)844-8834    ORG. 6415  
C M. J. SHORTENCARIER     PHONE (505)846-1662    ORG. 7223  
C<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>  
C      PROGRAM LHS  
C      PARAMETER (NMAX=1000)  
C      PARAMETER (NVAR=150)  
C      PARAMETER (LENT=125)  
C      CHARACTER*8 FNAME  
C      CHARACTER*12 FNAMES  
  
CBHL      CHARACTER*1 TITLE  
  
CBHL      LOGICAL LEX  
  
CBHL  
C      COMMON/PARAM/TITLE(LENT), ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,  
C      1             ICORR>IDIST(NVAR),IRP  
C      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,  
C      1             ICORR>IDIST(NVAR),IRP  
C      COMMON/PARAMA/TITLE(LENT)  
C      COMMON/MCLHS/XMC(NVAR)  
  
CBHL      COMMON/CMATR/CORR((NVAR*(NVAR+1))/2),LCM(NVAR),NCM  
C      COMMON/SAMP/X(NMAX*NVAR)  
C      COMMON/RANK/XV(NMAX),RXV(NMAX),IWK(NMAX)  
C      COMMON/WORKX/XX(NMAX*NVAR)  
C      COMMON/WORKC/Q((NVAR*(NVAR+1))/2),S((NVAR*(NVAR+1))/2)  
C      LOC(I,J)=(J-1)*N+I  
C      LOC1(I,J)=J+(I*I-I)/2  
  
C-----  
C  
C  
C Open UNIT 5 for input and the appropriate units for output.  
C  
C  
C Files are as follows:  
C  
C      UNIT 5 = Input data file.  
C  
C      UNIT 6 = Output of input data.  
C  
C      UNIT 7 = Output of maximum breakthrough concentration, at each  
C                receptor, for each real.  
C  
C      UNIT 8 = Output of the time of arrival of the maximum breakthrough  
C                concentration, at each receptor, for each realization.  
C  
C      UNIT 9 = Summary of means and standard deviations of the risk,
```

```

C          max. arrival time, and max arrival concentration at each
C          receptor location. [Not used at this time]
C
C      UNIT 10= Breakthrough concentrations at each receptor location
C              using the mean parameter values; this file is closed
C              after the values are printed and thus can be used
C              to examine the mean characteristics of the transport
C              model while the Monte Carlo run is executing.
C
C      UNIT 11= Parameter values used for each realization.
C
C      UNIT 12= Tabular output of maximum concentration at receptor
C              location for import into spreadsheet. This includes
C              a header record for legend or field name.
C
C      UNIT 20= Breakthrough curve summary {Binary Format}.
C
C
C
C
C
C
C      WRITE (*, 6000)
C      6000 FORMAT(/////,
C      1          ' ++++++',
C      2          '      LHS          Version X.0          9/27/95',
C      3          ' ++++++',
C      4          '      Enter name of the primary input file',
C      5          '      (do not include the .DAT suffix)',
C
C
C      ITRY = 0
C      7000 CONTINUE
C
C      READ (*, '(A8)') FNAME
C
C      Clean-up (trim and convert to uppercase) file name.
C
C      CALL UPCASE (FNAME)
C      CALL LENTRM (FNAME, K)
C
C      Concatenate file suffixes (predefined).
C
C      FNAMES = FNAME (1:K)//'.DAT'
C
C      INQUIRE (FILE = FNAMES, EXIST = LEX)
C
C      IF ( .NOT. LEX) THEN
C          WRITE (*, 6010) FNAMES
C      6010  FORMAT(' Input data file ',A12,' does not exist',
C      1          ' Re-enter filename: ',
C          ITRY = ITRY + 1
C          IF (ITRY .EQ. 5) STOP
C          GO TO 7000
C      END IF
C
C      OPEN (5, FILE = FNAMES, STATUS = 'OLD')
C
C      FNAMES = FNAME//'.OUT'
C      OPEN (6, FILE = FNAMES)
C
C      FNAMES = FNAME//'.T2'
C      OPEN (2, FILE = FNAMES, FORM = 'UNFORMATTED')
C
C      FNAMES = FNAME//'.T3'
C      OPEN (3, FILE = FNAMES, FORM = 'UNFORMATTED')
C

```

```

      FNames = FName//'.T4'
      OPEN (4, FILE = FNames, FORM = 'UNFORMATTED')
C
      FNames = FName//'.TXT'
      OPEN (7, FILE = FNames)
C
      FNames = FName//'.T8'
      OPEN (8, FILE = FNames, FORM = 'UNFORMATTED')
C
      FNames = FName//'.T9'
      OPEN (9, FILE = FNames, FORM = 'UNFORMATTED')
C
      FNames = FName//'.MCT'
      OPEN (10, FILE = FNames)
C
C
C SUBROUTINE RDPAR READS IN THE PARAMETER STATEMENTS AND DEFINES
C THE VARIABLES IN COMMON /PARAM/
C SUBROUTINES BANNER AND WRTPAR ECHO A DESCRIPTION OF THE SAMPLE
C AS INPUT BY THE USER AND A DESCRIPTION OF THE DISTRIBUTIONS
C USED WITH THE INPUT VARIABLES
C
      CALL RDPAR
      CALL BANNER(1)
      CALL WRTPAR
C
C IF THE USER HAS SPECIFIED A CORRELATION STRUCTURE (ICM=1) THEN
C THE CORRELATION MATRIX IS ECHOED AND CHECKED TO MAKE SURE THAT
C IT IS POSITIVE DEFINITE. THE CHOLESKY FACTORIZATION IS COMPUTED
C TO BE USED LATER AS PART OF THE PROCESS FOR INDUCING THE DESIRED
C CORRELATION STRUCTURE
C
      IF(ICM.NE.0)THEN
        CALL PMTRX(NCM,3)
        IF(IRP.EQ.1)THEN
          IRP=0
          WRITE(6,9003)
        ENDIF
        IF(N.LE.NV)THEN
          ICM=0
          WRITE(6,9001)
          GO TO 10
        ENDIF
        CALL POSDEF(ITER)
        IF(ITER.GT.1)THEN
          WRITE(6,9002)
          CALL PMTRX(NCM,4)
        ENDIF
        NVX=(NV*(NV+1))/2
        DO 130 I=1,NVX
130      CORR(I)=0.0
        DO 140 I=1,NV
140      CORR(LOC1(I,I))=1.0
          LB=1
          REWIND 3
          READ(3)S
          DO 170 I=1,NCM
            DO 160 K=LB,NCM
160          CORR(LOC1(LCM(K),LCM(I)))=S(LOC1(K,I))
              LB=LB+1
170          CONTINUE
            CALL CHLSKY
            REWIND 3
            WRITE(3)Q
C
C THE FOLLOWING LINES ARE INCLUDED TO HANDLE THE

```

```

C PATHOLOGICAL CASE N=3, NV=2 AND ICM=0
C
      ELSE IF(N.EQ.3.AND.NV.EQ.2)THEN
        CORR(1)=1.0
        CORR(2)=0.0001
        CORR(3)=1.0
        CALL CHLSKY
        REWIND 3
        WRITE(3)Q
      ENDIF
10  CONTINUE
C
C THIS LOOP IS EXECUTED ONCE FOR EACH REPETITION REQUESTED.
C UNITS 7, 8, AND 9 HAVE BEEN DEFINED IN SUBROUTINE RDPAR
C
      DO 1000 IREP=1,NREP
        REWIND 7
        REWIND 8
        REWIND 9
        IF(IREP.GT.1)CALL BANNER(IREP)
C
C GENERATE THE DISTRIBUTION REQUESTED FOR VARIABLE J
C
      DO 100 J=1,NV
        IDT=IDIST(J)
        IF(IDT.EQ.1)THEN
          CALL BETA(J)
          GO TO 100
        ELSE IF(IDT.EQ.2.OR.IDT.EQ.3)THEN
          CALL NORMAL(J,IDT)
          GO TO 100
        ELSE IF(IDT.GE.4.AND.IDT.LE.7)THEN
          CALL UNIFRM(J,IDT)
          GO TO 100
        ELSE IF(IDT.EQ.8)THEN
          CALL TRIANG(J)
          GO TO 100
        ELSE IF(IDT.EQ.9)THEN
          CALL USRDST(J)
          GO TO 100
        ENDIF
100  CONTINUE
      IF(N.EQ.1)GO TO 270
C
C IF A RANDOM SAMPLE HAS BEEN GENERATED IT MUST BE SORTED FROM
C FROM SMALLEST TO LARGEST ON EACH VARIABLE AS PART OF THE
C STRUCTURING TO REDUCE THE POSSIBILITY OF SPURIOUS CORRELATIONS
C
      IF (IRS .NE. 1) GO TO 260
      DO 250 I=1,NV
        DO 240 J=1,N
240      XV(J)=X(LOC(J,I))
          CALL SIFT (XV,N)
          DO 250 J=1,N
            X(LOC(J,I))=XV(J)
250      CONTINUE
260      CONTINUE
          REWIND 4
          WRITE(4)X
C
C SUBROUTINE MIX PAIRS THE SAMPLE OBSERVATIONS TO MATCH THE
C DESIRED CORRELATION STRUCTURE
C
      CALL MIX
C
C THE SAMPLE IS WRITTEN OUT TO UNIT 1

```

```

C
  270  CONTINUE
CBHL
      WRITE(10,*) (XMC(J),J=1,NV)
      DO 290 I=1,N
        WRITE(10,*) (X(LOC(I,J)),J=1,NV)
C 290  WRITE(1,*)I,NV,(X(LOC(I,J)),J=1,NV)
      290  CONTINUE
CBHL
      REWIND 4
      WRITE(4)X
C
C SUBROUTINE DATOUT OUTPUTS THE LATEST SAMPLE AND CORRESPONDING RANKS
C SUBROUTINE HSTOUT OUTPUTS HISTOGRAMS FOR THE CURRENT SAMPLE
C SUBROUTINE COROUT OUTPUTS RAW AND RANK CORRELATIONS FOR THE
C CURRENT SAMPLE
C
      IF(IDATA.NE.0)CALL DATOUT
      IF(IHIST.NE.0)CALL HSTOUT
      IF(ICORR.NE.0)CALL COROUT
1000 CONTINUE
      CALL EXIT
9001 FORMAT('0',3(9('*'),' CAUTION USER PLEASE NOTE '),10('*'),//,1X,
1      'SINCE THE SAMPLE SIZE IS LESS THAN OR EQUAL TO THE ',
2      'NUMBER OF VARIABLES',/, ' THIS IS NOT A FULL RANK CASE ',
3      'SO THE REQUESTED',/, ' CORRELATION STRUCTURE, SHOWN ',
4      'ABOVE, CANNOT BE GENERATED',/, ' THEREFORE THE INPUT ',
5      'MATRIX WILL BE GENERATED AS IF THE',/, ' INPUT ',
6      'VARIABLES WERE INDEPENDENT',//,1X,115('*'))
9002 FORMAT('0',3(9('*'),' CAUTION USER PLEASE NOTE '),10('*'),//,1X,
1      'THE INPUT RANK CORRELATION MATRIX IS NOT POSITIVE ',
2      'DEFINITE',/, ' AN ITERATIVE PROCEDURE HAS BEEN USED TO ',
3      'PRODUCE A SUBSTITUTE RANK CORRELATION MATRIX',/, ' THIS ',
4      'ADJUSTED RANK CORRELATION MATRIX APPEARS ON THE NEXT ',
5      'PAGE',/, ' THE USER SHOULD EXAMINE THIS MATRIX TO MAKE ',
6      'SURE THAT THE CORRELATION REQUIREMENTS ARE STILL ',
7      'SATISFIED',//,1X,115('*'))
9003 FORMAT('0',3(9('*'),' CAUTION USER PLEASE NOTE '),10('*'),//,1X,
1      'RANDOM PAIRING CANNOT BE USED WHEN THE USER SPECIFIES ',
2      'A CORRELATION STRUCTURE',/, ' THUS, THE RANDOM PAIRING ',
3      'PARAMETER HAS BEEN IGNORED',//,1X,115('*'))
      END
C*****
C SUBROUTINE BANNER ECHOES USER INPUT CONCERNING THE SAMPLE
C SETTING. IT IS CALLED AT THE START OF EACH REPETITION OF THE
C SAMPLE AND IS USED TO DELIMIT REPETITIONS IN THE PRINTOUT
C
      SUBROUTINE BANNER(IREP)
      PARAMETER (NVAR=150)
      PARAMETER (LENT=125)
CBHL
      CHARACTER*1 TITLE
CBHL
CBHL
C      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
C      1      ICORR,IDIST(NVAR),IRP
C      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
C      1      ICORR,IDIST(NVAR),IRP
C      COMMON/PARAM/TITLE(LENT)
CBHL
C
C IF IREP GT 1 THEN RETRIEVE THE CURRENT VALUE OF THE RANDOM
C SEED (ISEED) HERE (IF NECESSARY)
C
      WRITE(6,9001)TITLE,ISEED,NV,N
      IF(NREP.GT.1)WRITE(6,9006)IREP,NREP

```

```

        IF(IRP.EQ.1)WRITE(6,9007)
        IF(ICM.EQ.1)WRITE(6,9002)
        IF(IDATA.EQ.1)WRITE(6,9003)
        IF(IHIST.EQ.1)WRITE(6,9004)
        IF(ICORR.EQ.1)WRITE(6,9005)
        RETURN
9001 FORMAT('1',//,4X,125A1,/,4X,'RANDOM SEED = ',I11,/,4X,
1         'NUMBER OF VARIABLES = ',I3,/,4X,'NUMBER ',
2         'OF OBSERVATIONS = ',I4)
9002 FORMAT('0',3X,'AN INPUT CORRELATION MATRIX HAS BEEN SPECIFIED')
9003 FORMAT('0',3X,'THE SAMPLE INPUT VECTORS WILL BE PRINTED ',
1         'ALONG WITH THEIR CORRESPONDING RANKS')
9004 FORMAT('0',3X,'HISTOGRAMS OF THE ACTUAL SAMPLE WILL BE PLOTTED ',
1         'FOR EACH INPUT VARIABLE')
9005 FORMAT('0',3X,'THE CORRELATION MATRICES (RAW DATA AND RANK ',
1         'CORRELATIONS) WILL BE PRINTED')
9006 FORMAT('0',3X,'REPLICATION NUMBER ',I3,' OF ',I3,
1         ' REPLICATIONS')
9007 FORMAT('0',3X,'RANDOM PAIRING WILL BE USED')
        END
C*****
C SUBROUTINE BETA IS USED TO GENERATE A BETA DISTRIBUTION ON THE
C INTERVAL (A,B) AND WITH PARAMETERS P AND Q
C
        SUBROUTINE BETA(J)
        PARAMETER (NMAX=1000)
        PARAMETER (NVAR=150)
        PARAMETER (LENT=125)
CBHL
        CHARACTER*1 TITLE
CBHL
CBHL
C      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
C      1      ICORR,IDIST(NVAR),IRP
        COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
1      1      ICORR,IDIST(NVAR),IRP
        COMMON/PARAM/TITLE(LENT)
CBHL
        COMMON/SAMP/X(NVAR*NMAX)
        COMMON /PQ/P,Q,NZ
        EXTERNAL BETAFN
        LOC(I,J) = (J-1)*N+I
        CALL ERXSET(10,0)
        CON = 1.E-06
        PROBINC = 1./FLOAT(N)
        IF(IRS.NE.0)PROBINC=1.0
        READ (8)A,B,P,Q
        STRTPT = 0.
        BX = P/(P+Q)
        DEL = SQRT((P*Q)/((P+Q)**2*(P+Q+1)))
        DO 1 I = 1,N
            PROB = PROBINC*RAN(ISEED)+ STRTPT
            CALL FINVER(BETAFN,PROB,BX,DEL,CON,CON,1.)
            X(LOC(I,J)) = A + (B-A)*BX
            IF(IRS.EQ.0)STRTPT = STRTPT + PROBINC
1 CONTINUE
        RETURN
        END
C*****
C SUBROUTINE BETAFN IS USED IN GENERATING A BETA DISTRIBUTION
C
        SUBROUTINE BETAFN(X,FOFX)
        DIMENSION Y(1)
        COMMON /PQ/P,Q,NZ
        IF(X.LT.0.)X=0.
        IF(X.GT.1.)X=1.

```

```

      OMX = 1.-X
      N=1
      CALL BETAIC(X,OMX,P,Q,N,Y,NZ)
      FOFX = Y(1)
      RETURN
      END
C*****
C SUBROUTINE BETAIC IS USED IN GENERATING A BETA DISTRIBUTION
C
      SUBROUTINE BETAIC(X,OMX,A,B,N,Y,NZ)
C
      WRITTEN BY D.E. AMOS AND S.L. DANIEL, JANUARY, 1975.
C
      REFERENCE SC-DR-69 591
C
      ABSTRACT
C      BETAIC COMPUTES AN N MEMBER SEQUENCE OF BETA DISTRIBUTIONS
C       $Y(K)=I(A+K-1,B,X)$ ,  $K=1,\dots,N$ ,  $A.GT.0$ ,  $B.GT.0$ , AND
C       $0.LE.X.LE.1$ , WHERE  $I(A,B,X)$  IS THE INCOMPLETE BETA FUNCTION
C      NORMALIZED TO 1. AT  $X=1$ . THE RELATION OF THE INCOMPLETE BETA
C      FUNCTION TO THE GAUSS HYPERGEOMETRIC FUNCTION IS USED OVER
C      VARIOUS PARAMETER RANGES WITH SERIES OR ASYMPTOTIC EXPRESSIONS
C      USED FOR EVALUATION STARTING AT  $A+N-1$  AND  $BO.GT.0$ . WITH
C       $BO=B$ -INTEGER PART OF  $B$  OR 1. THEN A COMBINATION OF FORWARD
C      RECURSION ON THE PARAMETER TO RAISE  $BO$  TO  $B$  FOLLOWED BY
C      BACKWARD RECURSION ON THE FIRST PARAMETER TO DECREASE  $A+N-1$  TO
C       $A$  GETS THE REQUIRED SEQUENCE.  $I(A,B,X)$  SATISFIES A TWO-TERM
C      RELATION IN BOTH PARAMETERS WHERE ADDITIONS CAN BE USED
C      EXCLUSIVELY TO RETAIN SIGNIFICANT DIGITS. BOTH  $X$  AND  $OMX=1.-X$ 
C      ARE ENTERED IN THE CALL LIST TO AVOID LOSSES OF SIGNIFICANCE
C      IN  $OMX$  WHEN AN ANALYTICAL EXPRESSION IS AVAILABLE( SEE THE F
C      AND T DISTRIBUTIONS). BETAIC USES HYPGEO AND BETALN.
C
      DESCRIPTION OF ARGUMENTS
C
      INPUT
C      X      - ARGUMENT,  $0.LE.X.LE.1$ .
C      OMX    -  $1.-X$ 
C      A      - START VALUE OF FIRST PARAMETER,  $A.GT.0$ .
C      B      - VALUE OF SECOND PARAMETER,  $B.GT.0$ .
C      N      - NUMBER OF BETA FUNCTIONS IN THE SEQUENCE,  $N.GE.1$ 
C
      OUTPUT
C      Y      - A VECTOR WHOSE FIRST N COMPONENTS CONTAIN
C               $Y(K)=I(A+K-1,B,X)$ ,  $K=1,\dots,N$ .
C      NZ     - UNDERFLOW FLAG
C              NZ.EQ.0, A NORMAL RETURN.
C              NZ.NE.0, UNDERFLOW,  $Y(K)=0.0$ ,  $K=N-NZ+1,N$  RETURNED
C
      ERROR CONDITIONS
C      IMPROPER INPUT - A FATAL ERROR
C      UNDERFLOW - A NON-FATAL ERROR.
C
      BETAIC USES SUBROUTINES BETALN, GAMALN, HYPGEO, ERRCHK, ERRGET,
C                      ERRPRT, ERXSET, ERSTGT
C
      DIMENSION Y(1)
      DIMENSION GBZ(23),PP(35),GCOE(66),WCOE(49)
      DIMENSION GCOE1(59),GCOE2(7)
      EQUIVALENCE(GCOE(1),GCOE1(1)),(GCOE(60),GCOE2(1))
      SAVE WCOE,GCOE1,GCOE2
      DATA WCOE /-6.41025641025641E-03,-8.33333333333333E-02,
1-5.00000000000000E-01,-1.46538461538462E+01,-4.58333333333333E+00,
2-8.25000000000000E+00,-1.10000000000000E+01,-1.10000000000000E+01,
3-8.25000000000000E+00,-4.58333333333333E+00,-1.83333333333333E+00,

```



```

4-5.000000000000000E-01,-8.333333333333333E-02, 1.91752691752692E-03,
5 2.10927960927961E-02, 8.72474747474747E-01, 3.16391941391941E-01,
6 6.32783882783883E-01, 8.85897435897436E-01, 6.63847818847819E+00,
7 6.32783882783883E-01, 3.16391941391941E-01, 1.05463980463980E-01,
8 2.10927960927961E-02,-8.41750841750842E-04,-7.57575757575758E-03,
9-3.03030303030303E-02,-7.07070707070707E-02,-4.31481481481481E+00,
A-1.06060606060606E-01,-7.07070707070707E-02,-3.03030303030303E-02,
B-7.57575757575758E-03, 5.95238095238095E-04, 4.16666666666667E-03,
C 4.29166666666667E-01, 2.08333333333333E-02, 2.08333333333333E-02,
D 1.25000000000000E-02, 4.16666666666667E-03,-7.93650793650794E-04,
E-3.96825396825397E-03,-7.93650793650794E-03,-7.93650793650794E-03,
F-3.96825396825397E-03, 2.77777777777778E-03, 8.33333333333333E-03,
G 8.33333333333333E-03,-8.33333333333333E-02/
DATA GCOE1 /-8.33333333333333E-02, 1.55375180375180E-01,
1-1.08042328042328E-01, 3.81997721928277E-02,-7.78250845091123E-03,
2 9.70630021555947E-04,-7.60017468278464E-05, 3.72597920340976E-06,
3-1.10336922674162E-07, 1.79409630364491E-09,-1.22324747975790E-11,
4-9.09090909090909E-02, 1.60747354497354E-01,-1.03988095238095E-01,
5 3.35006062610229E-02,-6.06796859690378E-03, 6.52046360596708E-04,
6-4.21167695473251E-05, 1.59315751763668E-06,-3.22937334656085E-08,
7 2.69114445546737E-10,-1.00000000000000E-01, 1.66269841269841E-01,
8-9.87979497354497E-02, 2.84777336860670E-02,-4.46727109053498E-03,
9 3.97698045267490E-04,-1.98929398148148E-05, 5.16699735449735E-07,
A-5.38228891093474E-09,-1.11111111111111E-01, 1.71785714285714E-01,
B-9.21626984126984E-02, 2.31706532921811E-02,-3.02854938271605E-03,
C 2.09780092592593E-04,-7.23379629629630E-06, 9.68812003968254E-08,
D-1.25000000000000E-01, 1.76984126984127E-01,-8.36805555555556E-02,
E 1.76697530864198E-02,-1.80844907407407E-03, 8.68055555555556E-05,
F-1.55009920634921E-06,-1.42857142857143E-01, 1.81245000000000E-01,
G-7.28395061728395E-02, 1.21527777777778E-02,-8.68055555555556E-04,
H 2.17013888888889E-05,-1.66666666666667E-01, 1.83333333333333E-01,
I-5.90277777777778E-02, 6.94444444444444E-03,-2.60416666666667E-04,
J-2.00000000000000E-01, 1.80555555555556E-01,-4.16666666666667E-02/
DATA GCOE2 / 2.60416666666667E-03,-2.50000000000000E-01,
1 1.66666666666667E-01,-2.08333333333333E-02,-3.33333333333333E-01,
2 1.25000000000000E-01,-5.00000000000000E-01/

```

C

```

E1=ABS(I1MACH(12))
E2=ABS(I1MACH(13))
ALB=R1MACH(5)
ELIM=(MIN(E1,E2)*ALB-3.0)*2.303E0
UR=MAX(1.0E-14,R1MACH(4))
FLIM=1.0/(UR*1.0E3)
P9LIM=1.0-1.0E3*UR
REL2=UR*1.0E4
ALIM=UR
TOL=UR*5.0E3
MAXR=1000

```

C

C

C

C

C

```

TESTING OF VARIABLES, ABS(X+OMX-1).LT.TOL, 0.LE.X.LE.1,
A.GT.0, B.GT.0, N.GE.1

```

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

```

IF(N.LE.0) GO TO 90
IF(A.LE.0.0) GO TO 91
IF(B.LE.0.0) GO TO 92
NZ=0
AA=A+FLOAT(N)-1.
BB=B
NSAVE=N
IF (X) 93,25,75
DO 30 I=1,N
30 Y(I)=0.
RETURN
35 DO 40 I=1,N
40 Y(I)=1.

```

```

      RETURN
75    IF (X.EQ.1.) GO TO 35
      IF(OMX.LT.0.0) GO TO 93
      IF(ABS(X+OMX-1.).GT.TOL) GO TO 93
C
C    COMPUTATION OF BO
C
      IB=BB
      BO=BB-FLOAT(IB)
      IF (BO) 80,80,85
80    BO=1.0
      IB=IB-1
85    DA=AA
      DB=BB
C
C    COMPUTATION OF XLN AND OMXLN BY FORMULA (8)
C
      IF(X.GT.0.9) GO TO 95
      XLN=LOG(X)
      GO TO 105
95    ASSIGN 100 TO IDX
      DX=OMX
      GO TO 120
100   XLN=W
      105 IF(OMX.GT.0.9) GO TO 115
      OMXLN=LOG(OMX)
      GO TO 140
115   ASSIGN 135 TO IDX
      DX=X
C
C    LOGARITHM ROUTINE FOR ARGUMENTS GT 0.9 BY FORMULA (8)
C
120   SW=0.
      A1=DX
      DO 125 K=1,16
      GBZ(K)=A1
125   A1=A1*DX
      K=17
      DO 130 J=1,16
      K=K-1
      SW=SW+GBZ(K)/FLOAT(K)
130   CONTINUE
      W=-SW
      GO TO IDX,(100,135)
135   OMXLN=W
      140 IF(BB.NE.1.0) GO TO 150
C
C    BY FORMULA(1A)
C
      P=EXP(AA*XLN)
      GO TO 395
150   BP1=B+1.
      IF(BP1*X.LT.(B+.7)/BP1) GO TO 206
      IF(BO.EQ.1.) GO TO 175
      IF(X.GT.0.7) GO TO 215
      IF(X.GT.0.4) GO TO 180
      IF(AA.LE.BB) GO TO 180
C
C    BY FORMULA(4)
C
      T2=1.-DB
      T3=DA+1.
      T4=-X/OMX
CBHL
      IE=1
      COE=EXP(DA*XLN-T2*OMXLN-BETALN(DA,DB,IE))/DA

```

```

CBHL      P=COE*HYPGEO(1.,T2,T3,T4,1.)
          GO TO 395
C
C      BY FORMULA(1A)
C
175      P=EXP(AA*XLN)
          GO TO 360
180      IF(DB.GT.1.) GO TO 200
C
C      BY FORMULA(1)
C
          ASSIGN 395 TO IN1
          GO TO 210
C
200      ASSIGN 359 TO IN1
          DB=BO
          GO TO 210
206      ASSIGN 395 TO IN1
C
C      BY FORMULA(1)
C
210      T2=1.-DB
          T3=DA+1.
CBHL
          IE=1
          COE= EXP(DA*XLN-BETALN(DA,DB,IE))/DA
CBHL
          P=COE*HYPGEO(DA,T2,T3,X,1.)
          GO TO IN1,(395,359)
215      IF(AA.GT.BB) GO TO 225
C
C      BY FORMULA(3)
C
          T2=1.-AA
          T3=BB+1.
          T4=-OMX/X
CBHL
          IE=1
          COE= EXP((AA-1.)*XLN+BB*OMXLN-BETALN(AA,BB,IE))/BB
CBHL
          P=1.-COE*HYPGEO(1.,T2,T3,T4,1.)
          IF(P.GE.0.1) GO TO 395
C
225      CONTINUE
          IF(AA*OMX.GT.3.) GO TO 280
          IF(AA.LT.20.) GO TO 240
          DA=AA
          GO TO 245
240      IA=AA
          NN=20-IA
          DA=AA+FLOAT(NN+1)
          N=N+NN+1
          IF (DA*OMX.GT.3.) GO TO 280
245      CONTINUE
C
C      P FOR DA*OMX.LE.3. AND DA.GE.20. BY FORMULA (2),(2A),(2B),(2C)
C
          SBP1=BO+1.
          OMSA=1.-DA
          E=HYPGEO(BO,OMSA,SBP1,OMX,0.0)
          PHY=1.+E
CBHL
          IE=1
          C= EXP(BO*OMXLN-BETALN(DA,BO,IE))/BO
CBHL

```

```

      P=1.-C*PHY
      IF(ABS(P).GE.0.1) GO TO 275
      F=C-1.
      IF(ABS(F).GE.0.1) GO TO 270
C
C      COMPUTE W
C
      RBA=BO/DA
      SAPB=DA+BO
      RSAPB=1./SAPB
      RSAPB2=RSAPB*RSAPB
      SUMK=0.
      AK=1.
      A1=-1.
      DO 260 K=1,13
      A1=-A1*RBA
      SUMK=SUMK+A1/AK
      AK=AK+1.
260  CONTINUE
      ASUM=0.
      KK=13
      J=1
      DO 261 JJ=1,13,2
      BSUM=0.
      DO 262 K=1, KK
      BSUM=RBA*(BSUM+WCOE(J))
      J=J+1
262  CONTINUE
      ASUM=RSAPB2*ASUM+BSUM
      KK=KK-2
261  CONTINUE
      ASUM=RSAPB*ASUM
CBHL
      IE=2
C      ASUM=ASUM+(DA-.5)*SUMK+BO*LOG(SAPB*OMX)-BO-GAMALN(SBP1,0.,IE)
      ASUM=ASUM+(DA-.5)*SUMK+BO*LOG(SAPB*OMX)-BO-BETALN(SBP1,0.,IE)
CBHL
      F=0.
      A2=1.
      AK=1.
      ATEST=ALIM*ABS(ASUM)
      DO 265 K=1,25
      A2=A2*ASUM/AK
      F=F+A2
      IF(ABS(A2).LT.ATEST) GO TO 270
      AK=AK+1.
265  CONTINUE
270  P=-F-E-F*E
      275 IF(BB.LT.1.0) GO TO 395
      GO TO 360
      280 IF(AA*OMX.GT.35.) GO TO 340
      IF(AA.LT.1001.) GO TO 295
      DA=AA
      N=NSAVE
      GO TO 300
295  IN=1001.-AA
      INP1=IN+1
      N=NSAVE+INP1
      DA=AA+FLOAT(INP1)
      IF(DA*OMX.LT.35.) GO TO 300
      IN=35./OMX-AA+1.
      INP1=IN+1
      DA=AA+FLOAT(INP1)
      N=NSAVE+INP1
      GO TO 345
300  CONTINUE

```

```

      SAM1=DA-1.
      ZZ=SAM1*OMX
C
C      BIG GAMMA(BO,ZZ), 3.LT.ZZ.LT.35., ZZ=(DA-1)*OMX BY FORMULA (5)
C
      KFLAG=0
      NBAR=20
305  IF(NBAR.GT.MAXR) GO TO 94
      RNP1=0.0
      SN=NBAR+1
      DO 310 J=1,NBAR
      SN=SN-1.
      RN=ZZ+(SN-BO)*RNP1/(SN+RNP1)
310  RNP1=RN
      IF(KFLAG.NE.0) GO TO 320
      KFLAG=1
      NBAR=NBAR+NBAR
      RSAVE=RN
      GO TO 305
320  RTEST=ABS((RSAVE-RN)/RN)
      IF(RTEST.LE.REL2) GO TO 330
      RSAVE=RN
      NBAR=NBAR+NBAR
      GO TO 305
330  R1= EXP(BO*LOG(ZZ)-ZZ)
      GBZ(1)=R1/RN
C
C      RECURSION FOR BIG GAMMA(BO+K,ZZ), K=1,2,...23 BY FORMULA (6)
C
      AJ=0.
      DO 335 K=2,23
      GBZ(K)=(AJ+BO)*GBZ(K-1)+R1
      AJ=AJ+1.
      R1=R1*ZZ
335  CONTINUE
C
C      P FOR 0.965.LE.X.LT.0.997 AND DA.GE.1001 BY FORMULA (7)
C
      RA1=1./SAM1
      ASUM=0.
      K1=13
      K2=23
      J=1
      DO 336 KK=1,11
      BSUM=0.
      DO 337 K=K1,K2
      BSUM=BSUM+GCOE(J)*GBZ(K)
      J=J+1
337  CONTINUE
      K1=K1-1
      K2=K2-2
      ASUM=RA1*(ASUM+BSUM)
336  CONTINUE
      ASUM=ASUM+GBZ(1)
C
C      CBHL
      IE=1
      COE = EXP(-BETALN(DA,BO,IE) - BO*LOG(SAM1))
CBHL
      P=COE*ASUM
      GO TO 275
C
C      P FOR 0.7.LT.X.LT.0.965 AND DA*OMX.GT.35. BY FORMULA (4A)
C
340  DA=AA
345  CONTINUE

```

```

      A1=1.
      DAK=DA+1.
      DKBO=1.-BO
      RAT=-X/OMX
      DO 350 K=1,35
      A1=(RAT*DKBO/DAK)*A1
      PP(K)=A1
      DAK=DAK+1.
      DKBO=DKBO+1.
350   CONTINUE
      S=0.
      J=36
      DO 355 K=1,35
      J=J-1
355   S=S+PP(J)
      S=S+1.
CBHL
      IE=1
      COE= EXP(DA*XLN+(BO-1.)*OMXLN-BETALN(DA,BO,IE))/DA
CBHL
      P=S*COE
      GO TO 275
C
C   FORWARD RECURRENCE BY FORMULA (12A)
C
      359 DB=BB
360   IF (IB.LE.0) GO TO 395
CBHL
      IE=1
      S=DA*XLN+BO*OMXLN-BETALN(DA,BO,IE)-LOG(BO)
CBHL
C
C   SCALING FOR UNDERFLOW
C
      IF(S.GE.-ELIM .AND. P.GT.0.0) GO TO 385
366   FF=1.
      P=0.
      SUM=0.
375   IF (IB.EQ.0) GO TO 395
      IF(FF.GE.FLIM) GO TO 380
      SUM=SUM+FF
      FF=FF*((BO+DA)/(BO+1.))*OMX
      IB=IB-1
      BO=BO+1.
      GO TO 375
380   S1=S
      S=S+LOG(FF)
      IF(S.LT.-ELIM) GO TO 366
      ZZ=S1+LOG(SUM)
      IF(ZZ.LT.-ELIM) GO TO 366
      P=EXP(ZZ)
385   TERM=EXP(S)
      P=P+TERM
      IB=IB-1
      IF (IB.EQ.0) GO TO 395
      AM1=DA-1.
      BKK=BO
      DO 390 KK=1,IB
      BKK=BKK+1.
      TERM=TERM*((BKK+AM1)/BKK)*OMX
      P=P+TERM
390   CONTINUE
395   CONTINUE
      IF(N.NE.1) GO TO 405
      Y(1)=P
      IF(P.EQ.0.0) NZ=1

```

```

        GO TO 440
C
C      BACKWARD RECURRENCE BY FORMULA (9A)
C
405    DAM1=DA-1.
        DBM2=DB-2.
CBHL
        IE=1
        S=DAM1*XLN+DB*OMXLN-BETALN(DA,DB,IE)-LOG(DB+DAM1)
CBHL
C
C      SCALING FOR UNDERFLOW
C
        IF(S.GE.-ELIM .AND. P.GT.0.0) GO TO 430
        IF(P.LT.P9LIM) GO TO 411
        DO 406 I=1,NSAVE
406    Y(I)=1.
407    N=NSAVE
        RETURN
411    FF=1.
        P=0.
        SUM=0.
420    CONTINUE
        IF(N.EQ.1) GO TO 412
        IF(FF.GE.FLIM) GO TO 425
        SUM=SUM+FF
        FF=FF*(DA-1.)/((DA+DBM2)*X)
        DA=DA-1.
        N=N-1
        GO TO 420
412    DO 413 I=1,NSAVE
413    Y(I)=0.
        N =NSAVE
        NZ=N
        RETURN
425    S1=S
        S=S+LOG(FF)
        IF(S.LT.-ELIM) GO TO 411
        ZZ=S1+LOG(SUM)
        IF(ZZ.LT.-ELIM) GO TO 411
        P=EXP(ZZ)
430    TERM=EXP(S)
        IF(N-NSAVE) 431,434,432
431    IS=N+1
        DO 429 K=IS,NSAVE
429    Y(K)=0.
        NZ=NSAVE-N
        Y(N)=P
        IF(N-2) 440,408,409
408    Y(1)=P+TERM
        GO TO 440
409    KL=N-1
        GO TO 435
434    Y(NSAVE)=P
        KL=NSAVE-1
        IF(KL.GT.1) GO TO 435
441    Y(1)=P+TERM
        GO TO 440
432    KL=N-NSAVE
        DO 433 K=1,KL
        P=P+TERM
        TERM=TERM*(DA-1.)/((DA+DBM2)*X)
        DA=DA-1.
433    CONTINUE
        Y(NSAVE)=P
        KL=NSAVE-1

```

```

      IF(KL-1) 440,441,435
435 P=P+TERM
      Y(KL)=P
      KL=KL-1
      KK=KL
      DO 436 K=1,KL
      TERM=TERM*(DA-1.)/((DA+DBM2)*X)
      DA=DA-1.
      Y(KK)=TERM+Y(KK+1)
436 KK=KK-1
440 N=NSAVE
      DO 442 I=1,N
      IF(Y(I).GT.1.) Y(I)=1.
442 CONTINUE
      RETURN
C
      90 CALL ERRCHK(32,32HIN BETAIC, IMPROPER INPUT FOR N.)
      RETURN
      91 CALL ERRCHK(32,32HIN BETAIC, IMPROPER INPUT FOR A.)
      RETURN
      92 CALL ERRCHK(32,32HIN BETAIC, IMPROPER INPUT FOR B.)
      RETURN
      93 CALL ERRCHK(39,39HIN BETAIC, IMPROPER INPUT FOR X OR OMX.)
      RETURN
      94 CALL ERRCHK(67,67HIN BETAIC, NO CONVERGENCE IN CONTINUED FRACTION
1FOR GAMMA FUNCTION.)
      RETURN
      END
C*****
C SUBROUTINE BETALN IS USED IN GENERATING A BETA DISTRIBUTION
C
      FUNCTION BETALN(A,B,IENTRY)
C
C      WRITTEN BY D.E. AMOS AND S.L. DANIEL, MARCH, 1975.
C
C      REFERENCE SC-DR-69 591 AND SAND-75-0152
C
C      ABSTRACT
C      FUNCTION BETALN COMPUTES THE NATURAL LOG OF THE COMPLETE BETA
C      FUNCTION, LN BETA(A,B), WHERE BETA(A,B) IS DEFINED IN TERMS OF
C      THE GAMMA FUNCTION BY BETA(A,B)=GAMMA(A)*GAMMA(B)/GAMMA(A+B).
C      THE NATURAL LOG OF THE GAMMA FUNCTION IS COMPUTED WITH THE
C      LOGARITHMIC TERMS COMBINED ANALYTICALLY TO MINIMIZE LOGARITHM
C      EVALUATIONS. A RATIONAL CHEBYSHEV APPROXIMATION ON (8,1000)
C      AND THE ASYMPTOTIC EXPANSION FOR X.GT.1000 COMPLETES THE
C      CALCULATION OF THE NON-LOGARITHMIC TERMS. BACKWARD RECURSION
C      ON GAMMA(X-1)=GAMMA(X)/(X-1) AND THE LOGARITHM OF THE RESULT
C      SUFFICES FOR X.LT.8.
C
C      DESCRIPTION OF ARGUMENTS
C
C      INPUT
C      A      - ARGUMENT, A.GT.0.
C      B      - ARGUMENT, B.GT.0.
C      OUTPUT
C      BETALN - VALUE FOR LN BETA(A,B)
C
C      ERROR CONDITIONS
C      INPUT PARAMETER NON-POSITIVE - A FATAL ERROR
C
C
C      BETALN USES SUBROUTINES ERRCHK, ERRGET, ERRPRT, ERXSET, ERSTGT
C      DIMENSION G(8),P(5),Q(2)
C      SAVE XLIM1,XLIM2,RTWPIL,PI,G,P
C

```



```

DATA XLIM1,XLIM2,RTWPIL,PI/ 8. , 1000. , 9.18938533204673E-01,
1 3.14159265358979E+00/
C
DATA G / 8.52516136106541E+00, 6.57925121201010E+00,
1 4.78749174278205E+00, 3.17805383034795E+00, 1.79175946922806E+00,
2 6.93147180559945E-01, 2*0.0 /
C
DATA P / 7.66345188000000E-04,-5.94095610520000E-04,
1 7.93643110484500E-04,-2.77777775657725E-03, 8.33333333333169E-02/
C
DATA Q /-2.77777777777778E-03, 8.33333333333333E-02/
C
C
IF(IENTRY.EQ.2) GO TO 999
IF(A.LE.0.) GO TO 91
IF(B.LE.0.) GO TO 91
APB=A+B
IF(APB.LE.8.) GO TO 20
IF(A.LE.8.) GO TO 15
IF(B.GT.8.) GO TO 40
GO TO 60
15 IF(B.GT.8.) GO TO 65
C
20 IFLAG=0
ASSIGN 25 TO ISET
X=APB
GO TO 110
25 T1=GAML
X=A
ASSIGN 30 TO ISET
GO TO 110
30 T2=GAML
X=B
ASSIGN 35 TO ISET
GO TO 110
35 RLBETA=T1-T2-GAML
BETALN=-RLBETA
RETURN
C
40 IFLAG=1
ASSIGN 45 TO ISET
X=APB
GO TO 110
45 T1=GAML
ASSIGN 50 TO ISET
X=A
GO TO 110
50 T2=GAML
ASSIGN 55 TO ISET
X=B
GO TO 110
55 SS=T1-T2-GAML
RLBETA=A*LOG(1.+B/A)+B*LOG(1.+A/B)+
1 .5*LOG(A*B/APB)+SS-RTWPIL
BETALN=-RLBETA
RETURN
C
60 D2=A
D3=B
GO TO 70
65 D2=B
D3=A
70 D1=APB
IFLAG=1
X=D1
ASSIGN 75 TO ISET

```

```

      GO TO 110
75  T1=GAML
      ASSIGN 80 TO ISET
      X=D2
      GO TO 110
80  T2=GAML
      SS=T1-T2
      IFLAG=0
      X=D3
      ASSIGN 85 TO ISET
      GO TO 110
85  RLBETA=(D2-.5)*LOG(1.+D3/D2)+D3*LOG(D1)-D3-GAML+SS
      BETALN=-RLBETA
      RETURN
C
C      IFLAG=0 FOR X.LE.8 AND LN GAMMA(X) IS RETURNED AT 300
C      IFLAG=1 FOR X.GT.8 AND LN GAMMA(X) LESS LOG FACTORS IS RETURNED
C      AT 300
C
CBHL
C
C      ENTRY GAMALN (A,B)
999  CONTINUE
      IFLAG=0
      ASSIGN 160 TO ISET
      X=A
      IF(X) 90,90,110
110  DX=X-XLIM1
      IF(DX) 120,150,140
120  IF(X-1.) 121,152,123
121  XZ=X+8.
      IX=INT(X)
      FXZ=X-IX
      TX=X
      FK=-.5
      NDX=7
      GO TO 126
123  DX=ABS(DX)
      NDX=DX
      DNDX=NDX
      NDX=NDX+1
      IF((DNDX-DX).EQ.0.) GO TO 151
      XZ=X+DNDX+1.
      IX=INT(X)
      FXZ=X-IX
      TX=1.
      FK=.5
126  CONTINUE
      IXZ=INT(XZ)
      XZI=FLOAT(IXZ)
      RX=1.0/XZ
      RXX=RX*RX
      DO 124 I=1,NDX
      XZI=XZI-1.0
      XK=(XZI+FXZ)/XZ
      TX=TX*XK
124  CONTINUE
      SUM=(X-FK)*LOG(XZ)-LOG(TX)-XZ
      PX=P(1)
      DO 125 I=2,5
      PX=PX*RXX+P(I)
125  CONTINUE
      GAML=PX*RX+SUM+RTWPIL
      GO TO 300
140  RX=1./X
      RXX=RX*RX

```

```

        IF((X-XLIM2).LT.0.) GO TO 141
        PX=Q(1)*RXX+Q(2)
        IF(IFLAG.EQ.1) GO TO 200
        GAML=PX*RX+(X-.5)*LOG(X)-X+RTWPIL
        GO TO 300
141  PX=P(1)
        SUM=(X-.5)*LOG(X)-X
        DO 142 I=2,5
        PX=PX*RX+P(I)
142  CONTINUE
        IF(IFLAG.EQ.1) GO TO 200
        GAML=PX*RX+SUM+RTWPIL
        GO TO 300
150  GAML=G(1)
        GO TO 300
151  GAML=G(NDX)
        GO TO 300
152  GAML=G(8)
        GO TO 300
160  BETALN=GAML
        RETURN
200  GAML=PX*RX
300  GO TO ISET,(25,30,35,45,50,55,75,80,85,160)
C
    90  CALL ERRCHK(61,61HIN BETALN, ARGUMENT FOR GAMALN IS LESS THAN OR E
        1QUAL TO ZERO.)
        RETURN
    91  CALL ERRCHK(53,53HIN BETALN, AN ARGUMENT IS LESS THAN OR EQUAL TO
        1ZERO.)
        RETURN
        END
C*****
C SUBROUTINE CHKDAT CHECKS DISTRIBUTION PARAMETERS FOR CONSISTENCY
C
    SUBROUTINE CHKDAT(PAR,A,B,P,Q)
    CHARACTER PAR*(*),PLOG*3,PBETA*4
    PARAMETER (PLOG='LOG',PBETA='BETA')
    IF(A.GE.B)THEN
        WRITE(6,9001)PAR,A,B
        STOP 'CHKDAT'
    ELSE IF(PAR(1:3).EQ.PLOG.AND.(A.LE.0.0.OR.B.LE.0.0))THEN
        WRITE(6,9002)PAR,A,B
        STOP 'CHKDAT'
    ELSE IF(PAR.EQ.PBETA.AND.(P.LT.0.5.OR.Q.LT.0.5))THEN
        WRITE(6,9003)P,Q
        STOP 'CHKDAT'
    ELSE IF(PAR.EQ.PBETA)THEN
        WRITE(8)A,B,P,Q
    ELSE
        WRITE(8)A,B
    ENDIF
    RETURN
9001  FORMAT(1H1,5X,'FOR THE ',A,'DISTRIBUTION THE LOWER LIMIT A ',
    1      G20.10,/,6X,'IS GREATER THAN OR EQUAL TO THE UPPER ',
    2      'LIMIT B ',G20.10)
9002  FORMAT(1H1,5X,'FOR THE ',A,'DISTRIBUTION THE LOWER LIMIT A ',
    1      G20.10,/,6X,'OR THE UPPER LIMIT B ',G20.10,/,6X,
    2      'SHOULD NOT BE LESS THAN OR EQUAL TO ZERO')
9003  FORMAT(1H1,5X,'FOR THE BETA DISTRIBUTION THE PARAMETER P ',
    1      G20.10,/,6X,'OR THE PARAMETER Q ',G20.10,/,6X,'SHOULD ',
    2      'NOT BE LESS THAN 0.5 OR NUMERICAL INSTABILITIES WILL ',
    3      'OCCUR AS',/,6X,'A RESULT OF TRYING TO INVERT THE BETA ',
    4      'DISTRIBUTION ON PORTIONS OF THE CURVE',/,6X,'WITH ',
    5      'EXTREME SLOPES (SEE FIGURE 5 PAGE 24 OF THE USER GUIDE)')
        END
C*****

```

```

C SUBROUTINE CHKDIM CHECKS TO SEE IF PROGRAM DIMENSIONS HAVE BEEN
C EXCEEDED
C
      SUBROUTINE CHKDIM(IOPT,NREQ,NMX,PCHR,LCHR)
      PARAMETER (NVAR=150)
      PARAMETER (LENC=80)
      PARAMETER (LENT=125)
      CHARACTER*(*) PCHR,LCHR
      CHARACTER*(LENC) CRDSTR(NVAR)
CBHL
      CHARACTER*1 TITLE
CBHL
      COMMON/CHRCRD/CRDSTR
      COMMON/OBSTR/NSTR,NOBSTR(NVAR)
CBHL
C      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
C      1          ICORR,IDIST(NVAR),IRP
      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
      1          ICORR,IDIST(NVAR),IRP
      COMMON/PARAMA/TITLE(LENT)
CBHL
      IF(IOPT.EQ.2)GO TO 100
      IF(NREQ.GT.NMX)THEN
        WRITE(6,9001)PCHR,NREQ,LCHR,NMX,LCHR,LCHR
        STOP 'CHKDIM'
      ENDIF
      RETURN
100 CONTINUE
      IF(NREQ.GT.NMX)THEN
        WRITE(6,9002)NREQ,NMX
        STOP 'CHKDIM'
      ENDIF
      IERR=0
      DO 200 I=1,NSTR
        IF(NOBSTR(I).NE.N)THEN
          WRITE(6,9003)NOBSTR(I),N,CRDSTR(I)
          IERR=1
        ENDIF
      200 CONTINUE
      IF(IERR.EQ.1)STOP 'CHKDIM'
      RETURN
9001 FORMAT(1H1,5X,'THE PARAMETER CARD ',A,'REQUESTED ',I4,1X,A,/,
1        6X,'ONLY ',I4,1X,A,' ARE CURRENTLY PERMITTED',/,6X,
2        'PLEASE CONSULT THE USER MANUAL FOR INSTRUCTIONS ON ',
3        'HOW TO ALLOW MORE ',A)
9002 FORMAT(1H1,5X,'THE NUMBER OF VARIABLES REQUESTED ',I3,/,6X,
1        ' EXCEEDS THE MAXIMUM NUMBER OF VARIABLES CURRENTLY ',
2        'PERMITTED ',I3,/,6X,'PLEASE CONSULT THE USER MANUAL FOR',
3        'INSTRUCTIONS ON HOW TO ALLOW MORE VARIABLES')
9003 FORMAT(1H1,5X,'THE FOLLOWING DISTRIBUTION CARD REQUESTED ',I4,
1        ' OBSERVATIONS',/,6X,'HOWEVER THE NOBS PARAMETER CARD ',
2        'REQUESTED ',I4,' OBSERVATIONS',/,6X,'THIS DISCREPANCY ',
3        'MUST BE RESOLVED BEFORE PROCESSING CAN CONTINUE',/,3X,
4        '****',A,'****')
      END
C*****
C SUBROUTINE CHKSTR CHECKS PARAMETERS OF THE UNIFORM* AND THE
C LOGUNIFORM* DISTRIBUTIONS FOR CONSISTENCY
C
      SUBROUTINE CHKSTR(PAR,CARD)
      PARAMETER (NVAR=150)
      PARAMETER (NINTMX=50)
      PARAMETER (LENC=80)
      CHARACTER PAR*(*),CARD*(LENC),PLOG*3
      COMMON/STAR/NSUBOB(NINTMX),SUBINT(NINTMX+1),NINT
      CHARACTER*(LENC) CRDSTR(NVAR)

```

```

COMMON/CHRCRD/CRDSTR
COMMON/OBSTR/NSTR,NOBSTR(NVAR)
PARAMETER (PLOG='LOG')
IF(NINT.EQ.0)THEN
  WRITE(6,9001)PAR
  STOP 'CHKSTR'
ELSE IF(NINT.GT.NINTMX)THEN
  WRITE(6,9002)PAR,NINT,NINTMX
  STOP 'CHKSTR'
ELSE
  WRITE(8)NINT
ENDIF
NSTR=NSTR+1
DO 100 I=1,NINT
  IF(PAR(1:3).EQ.PLOG.AND.SUBINT(I).LE.0.0)THEN
    WRITE(6,9003)PAR,I,SUBINT(I)
    STOP 'CHKSTR'
  ELSE IF(SUBINT(I).GE.SUBINT(I+1))THEN
    WRITE(6,9004)PAR,I,SUBINT(I),SUBINT(I+1)
    STOP 'CHKSTR'
  ELSE IF(NSUBOB(I).LT.0)THEN
    WRITE(6,9005)PAR,I
    STOP 'CHKSTR'
  ELSE
    NOBSTR(NSTR)=NOBSTR(NSTR)+NSUBOB(I)
    WRITE(8)NSUBOB(I),SUBINT(I),SUBINT(I+1)
  ENDIF
100 CONTINUE
CRDSTR(NSTR)=CARD
RETURN
9001 FORMAT(1H1,5X,'FOR THE ',A,'DISTRIBUTION THE NUMBER OF ',
1      'SUBINTERVALS IS ZERO')
9002 FORMAT(1H1,5X,'FOR THE ',A,'DISTRIBUTION THE NUMBER OF ',
1      'SUBINTERVALS REQUESTED ',I3,/,6X,'IS GREATER THAN THE ',
2      'MAXIMUM NUMBER OF SUBINTERVALS CURRENTLY PERMITTED ',I3,
3      '/',6X,'PLEASE CONSULT THE USER MANUAL FOR INSTRUCTIONS ',
4      'ON HOW TO ALLOW MORE SUBINTERVALS')
9003 FORMAT(1H1,5X,'FOR THE ',A,'DISTRIBUTION THE SUBINTERVAL ',
1      'LIMIT FOR SUBINTERVAL ',I3,/,6X,'IS LESS THAN OR ',
2      'EQUAL TO ZERO ',G20.10)
9004 FORMAT(1H1,5X,'ON THE ',A,'DISTRIBUTION FOR SUBINTERVAL ',
1      I3,', THE LOWER LIMIT ',G20.10,/,6X,'IS GREATER ',
2      'THAN OR EQUAL TO THE UPPER LIMIT ',G20.10)
9005 FORMAT(1H1,5X,'FOR THE ',A,'DISTRIBUTION SUBINTERVAL ',I3,
1      ' REQUESTED A NEGATIVE NUMBER OF OBSERVATIONS')
END
C*****
C SUBROUTINE CHKTRI CHECKS PARAMETERS OF THE TRIANGULAR
C DISTRIBUTION FOR CONSISTENCY
C
  SUBROUTINE CHKTRI(A,B,C)
  IF(A.GT.B.OR.B.GT.C.OR.A.EQ.C)THEN
    WRITE(6,9000)A,B,C
    STOP 'CHKTRI'
  ELSE
    WRITE(8)A,B,C
  ENDIF
  RETURN
9000 FORMAT(1H1,5X,'FOR THE TRIANGULAR DISTRIBUTION THE PARAMETERS ',
1      'A,B,C ',3G20.10,/,6X,'HAVE BEEN INPUT IN THE INCORRECT ',
2      'ORDER',/,6X,'PLEASE CONSULT THE USER MANUAL FOR ',
3      'INSTRUCTIONS ON THE ORDER OF THE PARAMETERS')
END
C*****
C SUBROUTINE CHKZRO CHECKS TO MAKE SURE THAT THE MINIMUM
C REQUIREMENTS FOR A SAMPLE HAVE BEEN MET

```

```

C
      SUBROUTINE CHKZRO(N,NV,IRSET)
      IF(N.EQ.0)THEN
        WRITE(6,9001)
        STOP 'CHKZRO'
      ELSE IF(NV.EQ.0)THEN
        WRITE(6,9002)
        STOP 'CHKZRO'
      ELSE IF(IRSET.EQ.0)THEN
        WRITE(6,9003)
        STOP 'CHKZRO'
      ENDIF
      RETURN
9001 FORMAT(1H1,5X,'THE NUMBER OF OBSERVATIONS HAS NOT BEEN ',
1        'SPECIFIED')
9002 FORMAT(1H1,5X,'NO VARIABLES HAVE BEEN SPECIFIED')
9003 FORMAT(1H1,5X,'A RANDOM SEED HAS NOT BEEN SPECIFIED')
      END
C*****
C SUBROUTINE CHLSKY COMPUTES THE CHOLESKY FACTORIZATION OF A
C CORRELATION MATRIX STORED IN LOWER TRIANGULAR SYMMETRIC FORM
C
      SUBROUTINE CHLSKY
      PARAMETER (NVAR=150)
      PARAMETER (LENT=125)
CBHL
      CHARACTER*1 TITLE
CBHL
CBHL
C      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
C      1      ICORR,IDIST(NVAR),IRP
      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
      1      ICORR,IDIST(NVAR),IRP
      COMMON/PARAM/TITLE(LENT)
CBHL
      COMMON/CMATR/CORR((NVAR*(NVAR+1))/2),LCM(NVAR),NCM
      COMMON/WORKC/Q((NVAR*(NVAR+1))/2),S((NVAR*(NVAR+1))/2)
      LOC1(I,J)=J+(I*I-I)/2
      NVX=(NV*(NV+1))/2
      DO 70 I1=1,NVX
100  Q(I1)=0.0
      I1=1
      DO 80 I2=1,NV
      80  Q(LOC1(I2,I1))=CORR(LOC1(I2,I1))
      90  I1=I1+1
      I1MIN=I1-1
      DO 100 I2=1,I1MIN
100  Q(LOC1(I1,I1))=Q(LOC1(I1,I1))+Q(LOC1(I1,I2))**2
      Q(LOC1(I1,I1))=SQRT(1.0-Q(LOC1(I1,I1)))
      IF (I1.GE.NV) GO TO 130
      IPLUS=I1+1
      DO 120 I2=IPLUS,NV
      DO 110 K=1,I1MIN
      Q(LOC1(I2,I1))=Q(LOC1(I2,I1))+Q(LOC1(I2,K))*Q(LOC1(I1,K))
110  CONTINUE
120  Q(LOC1(I2,I1))=(CORR(LOC1(I2,I1))-Q(LOC1(I2,I1)))/Q(LOC1(I1,I1))
      GO TO 90
130  CONTINUE
      RETURN
      END
C*****
C SUBROUTINE CMCRD PROCESSES THE CORRELATION MATRIX PARAMETER CARD
C
      SUBROUTINE CMCRD
      PARAMETER (NVAR=150)
      PARAMETER (NCVAR=150)

```

```

      PARAMETER (LENT=125)
CBHL
      CHARACTER*1 TITLE
CBHL
CBHL
C      COMMON/PARAM/TITLE(LENT), ISEED,N,NV,IRS,ICM,NREP, IDATA, IHIST,
C      1      ICORR, IDIST(NVAR), IRP
      COMMON/PARAM/ ISEED,N,NV,IRS,ICM,NREP, IDATA, IHIST,
      1      ICORR, IDIST(NVAR), IRP
      COMMON/PARAMA/TITLE(LENT)
CBHL
      COMMON/CMATR/CORR((NVAR*(NVAR+1))/2), LCM(NVAR), NCM
      COMMON/UICORR/ICVAR(NCVAR), JCVAR(NCVAR), CVAR(NCVAR), NCV
      DIMENSION RIJ(NCVAR*2), IJCVAR(2*NCVAR)
      LOC1(I,J)=J+(I*I-I)/2
      NCV2=NCV*2
      DO 100 ICV=1,NCV
        I=ICVAR(ICV)
        J=JCVAR(ICV)
        IF(ABS(CVAR(ICV)).GE.1.0) THEN
          WRITE(6,9001) I,J,CVAR(ICV)
          STOP 'CMCRD'
        ELSE IF(I.EQ.J.AND.CVAR(ICV).NE.1.0) THEN
          WRITE(6,9002) I,J,CVAR(ICV)
          STOP 'CMCRD'
        ELSE IF(I.GT.NV.OR.J.GT.NV) THEN
          WRITE(6,9003) I,J,CVAR(ICV),NV
          STOP 'CMCRD'
        ELSE
          IJCVAR(ICV)=I
          IJCVAR(NCV+ICV)=J
        ENDIF
      100 CONTINUE
      DO 110 I=1,NCV2
      110 RIJ(I)=IJCVAR(I)
      CALL SIFT(RIJ,NCV2)
      DO 120 I=1,NCV2
      120 IJCVAR(I)=RIJ(I)
      NCM=1
      LCM(NCM)=IJCVAR(1)
      DO 200 I=2,NCV2
        IF(IJCVAR(I).NE.LCM(NCM)) THEN
          NCM=NCM+1
          LCM(NCM)=IJCVAR(I)
        ENDIF
      200 CONTINUE
      NSIZE=(NCM*(NCM+1))/2
      DO 300 I=1,NSIZE
      300 CORR(I)=0.0
      DO 400 I=1,NCM
      400 CORR(LOC1(I,I))=1.0
      DO 600 ICV=1,NCV
        I=ICVAR(ICV)
        J=JCVAR(ICV)
        DO 500 KCM=1,NCM
          IF(I.EQ.LCM(KCM)) IM=KCM
          IF(J.EQ.LCM(KCM)) JM=KCM
        500 CONTINUE
        IF(IM.GT.JM) THEN
          CORR(LOC1(IM,JM))=CVAR(ICV)
        ELSE
          CORR(LOC1(JM,IM))=CVAR(ICV)
        ENDIF
      600 CONTINUE
      RETURN
      9001 FORMAT('1',3X,'THE CORRELATION BETWEEN VARIABLE ',I3,' AND ',

```

```

1      'VARIABLE ',I3,' IS GREATER THAN ONE IN ABSOLUTE ',
2      'VALUE ',F5.2)
9002 FORMAT('1',3X,'THE CORRELATION BETWEEN VARIABLE ',I3,' AND ',
1      'VARIABLE ',I3,' IS NOT EQUAL TO ONE ',F5.2)
9003 FORMAT('1',3X,'THE CORRELATION BETWEEN VARIABLE ',I3,' AND ',
2      'VARIABLE ',I3,' IS ',F5.2,/,4X,'HOWEVER ONLY ',I3,
3      ' VARIABLES HAVE BEEN DEFINED')
      END
C*****
C SUBROUTINE CORCAL COMPUTES A CORRELATION MATRIX FROM THE SAMPLE
C
      SUBROUTINE CORCAL
      PARAMETER (NMAX=1000)
      PARAMETER (NVAR=150)
      PARAMETER (LENT=125)
CBHL
      CHARACTER*1 TITLE
CBHL
      COMMON/SAMP/X(NMAX*NVAR)
CBHL
      COMMON/PARAM/TITLE(LENT), ISEED,N,NV, IRS, ICM,NREP, IDATA, IHIST,
C      1      ICORR, IDIST(NVAR), IRP
      COMMON/PARAM/ISEED,N,NV, IRS, ICM,NREP, IDATA, IHIST,
1      1      ICORR, IDIST(NVAR), IRP
      COMMON/PARAM/TITLE(LENT)
CBHL
      COMMON/CMATR/CORR((NVAR*(NVAR+1))/2), LCM(NVAR), NCM
      DIMENSION XM(NVAR), SSQ(NVAR)
      LOC(I,J)=(J-1)*N+I
      LOC1(I,J)=J+(I*I-I)/2
      IF(NV.EQ.1) THEN
        CORR(1)=1.0
        RETURN
      ENDIF
      DO 10 I=1,NV
        XM(I)=0.0
        SSQ(I)=0.0
10    CONTINUE
      K=(NV*(NV+1))/2
      DO 20 I=1,K
20    CORR(I)=0.0
      FN=N
C
C      COMPUTE THE MEAN
C
      DO 30 J=1,NV
      DO 30 K=1,N
30    XM(J)=XM(J)+X(LOC(K,J))
      DO 40 J=1,NV
40    XM(J)=XM(J)/FN
C
C      SUBTRACT THE MEAN FROM ALL OF THE ELEMENTS OF THE MATRIX
C      AND COMPUTE THE SUM OF SQUARES
C
      DO 50 J=1,NV
      DO 50 K=1,N
        X(LOC(K,J))=X(LOC(K,J))-XM(J)
        SSQ(J)=SSQ(J)+X(LOC(K,J))*X(LOC(K,J))
50    CONTINUE
C
C      COMPUTE THE CORRELATION
C
      DO 70 I=2,NV
      IL1=I-1
      DO 70 J=1,IL1
        DO 60 K=1,N

```



```

60   CORR(LOC1(I,J))=CORR(LOC1(I,J))+X(LOC(K,I))*X(LOC(K,J))
70   CONTINUE
    DO 80 I=2,NV
      IL1=I-1
      DO 80 J=1,IL1
60   CORR(LOC1(I,J))=CORR(LOC1(I,J))/SQRT(SSQ(I)*SSQ(J))
      DO 90 I=1,NV
90   CORR(LOC1(I,I))=1.0
      RETURN
    END
C*****
C SUBROUTINE COROUT WILL OUTPUT THE RAW AND RANK CORRELATION
C MATRICES OF THE SAMPLE IF REQUESTED
C
    SUBROUTINE COROUT
      PARAMETER (NMAX=1000)
      PARAMETER (NVAR=150)
      PARAMETER (LENT=125)
CBHL
    CHARACTER*1 TITLE
CBHL
CBHL
C    COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
C    1      ICORR,IDIST(NVAR),IRP
    COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
    1      ICORR,IDIST(NVAR),IRP
    COMMON/PARAM/TITLE(LENT)
CBHL
    COMMON/CMATR/CORR((NVAR*(NVAR+1))/2),LCM(NVAR),NCM
    COMMON/SAMP/X(NMAX*NVAR)
    COMMON/RANK/XV(NMAX),RXV(NMAX),IWK(NMAX)
    LOC(I,J)=(J-1)*N+I
    REWIND 4
    READ(4)X
    DO 600 I=1,NV
600   LCM(I)=I
      CALL CORCAL
      CALL PMTRX(NV,1)
      IF(N.GT.NV.AND.ICM.EQ.0)CALL VIF
      DO 620 J=1,NV
        DO 610 I=1,N
610   XV(I)=X(LOC(I,J))
          CALL RANKER
          DO 620 I=1,N
            X(LOC(I,J))=RXV(I)
620   CONTINUE
          CALL CORCAL
          CALL PMTRX(NV,2)
          IF(N.GT.NV.AND.ICM.EQ.0)CALL VIF
          RETURN
        END
C*****
C SUBROUTINE DATOUT WILL OUTPUT THE SAMPLE AND ITS RANKS IF
C REQUESTED
C
    SUBROUTINE DATOUT
      PARAMETER (NMAX=1000)
      PARAMETER (NVAR=150)
      PARAMETER (LENT=125)
CBHL
    CHARACTER*1 TITLE
CBHL
CBHL
C    COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
C    1      ICORR,IDIST(NVAR),IRP
    COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,

```

```

1          ICORR, IDIST(NVAR), IRP
COMMON/PARAMA/TITLE(LENT)
CBHL
COMMON/RANK/XV(NMAX),RXV(NMAX),IWK(NMAX)
COMMON/SAMP/X(NMAX*NVAR)
LOC(I,J)=(J-1)*N+I
REWIND 4
READ(4)X
CALL OUTDAT(0)
DO 620 J=1,NV
  DO 610 I=1,N
610    XV(I)=X(LOC(I,J))
      CALL RANKER
      DO 620 I=1,N
        X(LOC(I,J))=RXV(I)
620  CONTINUE
      CALL OUTDAT(1)
      RETURN
      END
C*****
C SUBROUTINE DATSQZ PROCESSES PARAMETER CARDS WHICH REQUIRE
C CONVERTING CHARACTER DATA TO INTEGER DATA
C
      SUBROUTINE DATSQZ(CARD,CRDTYP,TMPCRD)
      PARAMETER (LENT=11)
      CHARACTER CARD*(*),CRDTYP*(*),TMPCRD*(LENT),BLANK,MINUS
      PARAMETER (BLANK=' ',MINUS='-')
      IZERO=ICHAR('0')
      ININE=ICHAR('9')
      LENC=LEN(CARD)
      IC=0
C SEARCH FOR BEGINNING OF NON-BLANK CHARACTER STRING
10  CONTINUE
      IC=IC+1
      IF(IC.GT.LENC)THEN
        WRITE(6,9001)CRDTYP
        STOP 'DATSQZ'
      ENDIF
      IF(CARD(IC:IC).EQ.BLANK)GO TO 10
      IBEG=IC
      IF(CARD(IC:IC).EQ.MINUS)GO TO 20
      ITEST=ICHAR(CARD(IC:IC))
      IF(ITEST.LT.IZERO.OR.ITEST.GT.ININE)THEN
        WRITE(6,9002)CRDTYP,CARD(IC:IC)
        STOP 'DATSQZ'
      ENDIF
C SEARCH FOR ENDING OF NON-BLANK CHARACTER STRING
20  CONTINUE
      IC=IC+1
      IF(IC.GT.LENC)GO TO 30
      IF(CARD(IC:IC).EQ.BLANK)GO TO 30
      ITEST=ICHAR(CARD(IC:IC))
      IF(ITEST.LT.IZERO.OR.ITEST.GT.ININE)THEN
        WRITE(6,9002)CRDTYP,CARD(IC:IC)
        STOP 'DATSQZ'
      ENDIF
      GO TO 20
C MOVE NON-BLANK CHARACTER STRING INTO TMPCRD RIGHT-JUSTIFIED
30  CONTINUE
      IEND=IC-1
      ILEN=IEND-IBEG+1
      IF(ILEN.GT.LENT)THEN
        WRITE(6,9003)CRDTYP,ILEN,LENT
        STOP 'DATSQZ'
      ENDIF
      TMPCRD=BLANK

```

```

        IT=LENT-ILEN
        DO 40 I=IBEG,IEND
            IT=IT+1
            TMPCRD(IT:IT)=CARD(I:I)
40    CONTINUE
        RETURN
9001  FORMAT(1H1,5X,'THE PARAMETER CARD ',A,'CONTAINS NO DATA')
9002  FORMAT(1H1,5X,'THE PARAMETER CARD ',A,'CONTAINS THE ',
1      'NON-NUMERIC CHARACTER ',A)
9003  FORMAT(1H1,5X,'THE DATA ON PARAMETER CARD ',A,'CONTAINS ',I2,
1      ' DIGITS',/,6X,'THE MAXIMUM NUMBER OF DIGITS ALLOWED ',
2      ' IS ',I2)
        END
C*****
C SUBROUTINE DMFSD IS USED IN INVERTING A CORRELATION MATRIX
C
        SUBROUTINE DMFSD (N,IPARM)
            PARAMETER (NVAR=150)
            COMMON/CMATR/CORR((NVAR*(NVAR+1))/2),LCM(NVAR),NCM
            KPIV=0
            DO 60 K=1,N
                KPIV=KPIV+K
                IND=KPIV
                LEND=K-1
                TOL=ABS(.01*(CORR(KPIV)))
                DO 60 I=K,N
                    DSUM=0.0
                    IF (LEND.EQ.0) GO TO 20
                    DO 10 L=1,LEND
                        LANF=KPIV-L
                        LIND=IND-L
10                     DSUM=DSUM+CORR(LANF)*CORR(LIND)
20                     DSUM=CORR(IND)-DSUM
                        IF (I.NE.K) GO TO 50
                        IF (DSUM-TOL) 30,30,40
30                     IF (DSUM.LE.0.0) GO TO 70
                        KT=K-1
                        WRITE(6,80)KT
40                     DPIV=SQRT(DSUM)
                        CORR(KPIV)=DPIV
                        DPIV=1.0/DPIV
                        GO TO 60
50                     CORR(IND)=DSUM*DPIV
60                     IND=IND+I
                        RETURN
70                     WRITE(6,90)K
                        IPARM=-K
                        RETURN
C
C
80                     FORMAT(20X,'ROUNDING ERROR IN ROW ',I2)
90                     FORMAT(20X,'MATRIX IS SINGULAR AT ROW ',I2)
                    END
C*****
C SUBROUTINE DSINV IS USED IN INVERTING A CORRELATION MATRIX
C
        SUBROUTINE DSINV (N)
            PARAMETER (NVAR=150)
            COMMON/CMATR/CORR((NVAR*(NVAR+1))/2),LCM(NVAR),NCM
            IPARM=0
            CALL DMFSD (N,IPARM)
            IF (IPARM.LT.0) RETURN
            IPIV=N*(N+1)/2
            IND=IPIV
            DO 40 I=1,N
                DIN=1.0/CORR(IPIV)

```

```

CORR(IPIV)=DIN
MIN=N
KEND=I-1
LANF=N-KEND
IF (KEND.LE.0) GO TO 30
J=IND
DO 20 K=1,KEND
WORK=0.0
MIN=MIN-1
LHOR=IPIV
LVER=J
DO 10 L=LANF,MIN
LVER=LVER+1
LHOR=LHOR+L
10 WORK=WORK+CORR(LVER)*CORR(LHOR)
CORR(J)=-WORK*DIN
20 J=J-MIN
30 IPIV=IPIV-MIN
40 IND=IND-1
DO 60 I=1,N
IPIV=IPIV+I
J=IPIV
DO 60 K=I,N
WORK=0.0
LHOR=J
DO 50 L=K,N
LVER=LHOR+K-I
WORK=WORK+CORR(LHOR)*CORR(LVER)
50 LHOR=LHOR+L
CORR(J)=WORK
60 J=J+K
RETURN
END
C*****
C SUBROUTINE ERRCHK IS AN ERROR CHECKING ROUTINE USED IN
C GENERATING A BETA DISTRIBUTION
C
C SUBROUTINE ERRCHK(NCHARS,NARRAY)
C
C SANDIA MATHEMATICAL PROGRAM LIBRARY
C APPLIED MATHEMATICS DIVISION 2642
C SANDIA LABORATORIES
C ALBUQUERQUE, NEW MEXICO 87115
C
C SIMPLIFIED VERSION FOR STAND-ALONE USE. APRIL 1977
C
C ABSTRACT
C THE ROUTINES ERRCHK, ERXSET, AND ERRGET TOGETHER PROVIDE
C A UNIFORM METHOD WITH SEVERAL OPTIONS FOR THE PROCESSING
C OF DIAGNOSTICS AND WARNING MESSAGES WHICH ORIGINATE
C IN THE MATHEMATICAL PROGRAM LIBRARY ROUTINES.
C ERRCHK IS THE CENTRAL ROUTINE, WHICH ACTUALLY PROCESSES
C MESSAGES.
C
C DESCRIPTION OF ARGUMENTS
C NCHARS - NUMBER OF CHARACTERS IN HOLLERITH MESSAGE.
C IF NCHARS IS NEGATED, ERRCHK WILL UNCONDITIONALLY
C PRINT THE MESSAGE AND STOP EXECUTION. OTHERWISE,
C THE BEHAVIOR OF ERRCHK MAY BE CONTROLLED BY
C AN APPROPRIATE CALL TO ERXSET.
C NARRAY - NAME OF ARRAY OR VARIABLE CONTAINING THE MESSAGE,
C OR ELSE A LITERAL HOLLERITH CONSTANT CONTAINING
C THE MESSAGE. BY CONVENTION, ALL MESSAGES SHOULD
C BEGIN WITH *IN SUBNAM, ..., WHERE SUBNAM IS THE
C NAME OF THE ROUTINE CALLING ERRCHK.
C

```

```

C      EXAMPLES
C      1. TO ALLOW CONTROL BY CALLING ERXSET, USE
C          CALL ERRCHK(30,30HIN QUAD, INVALID VALUE OF ERR.)
C      2. TO UNCONDITIONALLY PRINT A MESSAGE AND STOP EXECUTION, USE
C          CALL ERRCHK(-30,30HIN QUAD, INVALID VALUE OF ERR.)
C
C
C      ERRCHK USES SUBROUTINES ERRGET, ERRPRT, ERXSET, ERSTGT
C
C      DIMENSION NARRAY(16)
C
C      CALL ERRGET(NF,NT)
C      IF ERRCHK WAS CALLED WITH NEGATIVE CHARACTER COUNT, SET FATAL FLAG
C      IF (NCHARS.LT.0) NF = -1
C      IF MESSAGES ARE TO BE SUPPRESSED, RETURN
C      IF (NF.EQ.0) RETURN
C      IF CHARACTER COUNT IS INVALID, STOP
C      IF (NCHARS.EQ.0) WRITE(6,5)
C      5 FORMAT(/31H ERRCHK WAS CALLED INCORRECTLY.)
C      IF (NCHARS.EQ.0) STOP 'ERRCHK'
C      PRINT MESSAGE
C      CALL ERRPRT(IABS(NCHARS),NARRAY)
C      IF LAST MESSAGE, SAY SO
C      IF (NF.EQ.1) WRITE(6,10)
C      10 FORMAT (30H ERRCHK MESSAGE LIMIT REACHED.)
C      PRINT TRACE-BACK IF ASKED TO
C      IF ((NT.GT.0).OR.(NF.LT.0)) CALL SYSTEM ROUTINE FOR TRACEBACK
C      DECREMENT MESSAGE COUNT
C      IF (NF.GT.0) NF = NF-1
C      CALL ERXSET(NF,NT)
C      IF ALL IS WELL, RETURN
C      IF (NF.GE.0) RETURN
C      IF THIS MESSAGE IS SUPPRESSABLE BY AN ERXSET CALL,
C      THEN EXPLAIN ERXSET USAGE.
C      IF (NCHARS.GT.0) WRITE(6,15)
C      15 FORMAT (/13H *** NOTE ***
C      1/53H TO MAKE THE ERROR MESSAGE PRINTED ABOVE BE NONFATAL,,
C      2/39H OR TO SUPPRESS THE MESSAGE COMPLETELY,,
C      3/37H INSERT AN APPROPRIATE CALL TO ERXSET,
C      4 30H AT THE START OF YOUR PROGRAM.,
C      5/62H FOR EXAMPLE, TO PRINT UP TO 10 NONFATAL WARNING MESSAGES, USE,
C      6/27H      CALL ERXSET(10,0)      )
C      WRITE(6,20)
C      20 FORMAT (/28H PROGRAM ABORT DUE TO ERROR.)
C      STOP 'ERRCHK'
C      END
C*****
C SUBROUTINE ERRGET IS AN ERROR CHECKING ROUTINE USED IN
C GENERATING A BETA DISTRIBUTION
C
C      SUBROUTINE ERRGET(NFATAL,NTRACE)
C
C      ABSTRACT
C      ERRGET IS A COMPANION ROUTINE TO SUBROUTINE ERRCHK.
C      ERRGET ASSIGNS TO NFATAL AND NTRACE RESPECTIVELY THE VALUES
C      OF NF AND NT IN COMMON BLOCK MBLK0 THEREBY ASCERTAINING THE
C      STATE OF THE OPTIONS WHICH CONTROL THE EXECUTION OF ERRCHK.
C
C      DESCRIPTION OF ARGUMENTS
C      BOTH ARGUMENTS ARE OUTPUT ARGUMENTS OF DATA TYPE INTEGER.
C      NFATAL - CURRENT VALUE OF NF (SEE DESCRIPTION OF ERXSET.)
C      NTRACE - CURRENT VALUE OF NT (SEE DESCRIPTION OF ERXSET.)
C
C      CALL ERSTGT(1,NFATAL,NTRACE)
C      RETURN

```

```

      END
C*****
C SUBROUTINE ERRPRT IS AN ERROR CHECKING ROUTINE USED IN
C GENERATING A BETA DISTRIBUTION
C
      SUBROUTINE ERRPRT(NCHARS,NARRAY)
C
C     UTILITY ROUTINE TO SIMPLY PRINT THE HOLLERITH MESSAGE IN NARRAY,
C     WHOSE LENGTH IS NCHARS CHARACTERS.
C
      DIMENSION NARRAY(16)
C
      NOTE - NCH MUST BE THE NUMBER OF HOLLERITH CHARACTERS STORED
      PER WORD. IF NCH IS CHANGED, FORMAT 1 MUST ALSO BE
      CHANGED CORRESPONDINGLY.
C
      NCH = 8
C     FOR LINE PRINTERS, USE
      1 FORMAT (1X,16A8)
C     FOR DATA TERMINALS, USE
      1 FORMAT (1X,9A8)
      NWORDS = (NCHARS+NCH-1)/NCH
      WRITE(6,1)(NARRAY(I),I=1,NWORDS)
      RETURN
      END
C*****
C SUBROUTINE ERSTGT IS AN ERROR CHECKING ROUTINE USED IN
C GENERATING A BETA DISTRIBUTION
C
      SUBROUTINE ERSTGT(K,NFATAL,NTRACE)
C
C     THIS ROUTINE IS A SLAVE TO ERRGET AND ERRSET WHICH KEEPS
C     THE FLAGS AS LOCAL VARIABLES.
C
C     *** IF LOCAL VARIABLES ARE NOT NORMALLY RETAINED BETWEEN
C     CALLS ON THIS SYSTEM, THE VARIABLES LNF AND LNT CAN BE
C     PLACED IN A COMMON BLOCK AND PRESET TO THE FOLLOWING
C     VALUES IN THE MAIN PROGRAM.
C
      SAVE LNF,LNT
      DATA LNF/-1/,LNT/0/
      IF (K.LE.0) LNF = NFATAL
      IF (K.LE.0) LNT = NTRACE
      IF (K.GT.0) NFATAL = LNF
      IF (K.GT.0) NTRACE = LNT
      RETURN
      END
C*****
C SUBROUTINE ERXSET IS AN ERROR CHECKING ROUTINE USED IN
C GENERATING A BETA DISTRIBUTION
C
      SUBROUTINE ERXSET(NFATAL,NTRACE)
C
C     ABSTRACT
C
C     ERXSET IS A COMPANION ROUTINE TO SUBROUTINE ERRCHK.
C     ERXSET ASSIGNS THE VALUES OF NFATAL AND NTRACE RESPECTIVELY
C     TO NF AND NT IN COMMON BLOCK MBLK0 THEREBY SPECIFYING THE
C     STATE OF THE OPTIONS WHICH CONTROL THE EXECUTION OF ERRCHK.
C
C     DESCRIPTION OF ARGUMENTS
C
C     BOTH ARGUMENTS ARE INPUT ARGUMENTS OF DATA TYPE INTEGER.
C     NFATAL - IS A FATAL-ERROR / MESSAGE-LIMIT FLAG. A NEGATIVE
C     VALUE DENOTES THAT DETECTED DIFFICULTIES ARE TO BE
C     TREATED AS FATAL ERRORS. NONNEGATIVE MEANS NONFATAL.
C     A NONNEGATIVE VALUE IS THE MAXIMUM NUMBER OF NONFATAL
C     WARNING MESSAGES WHICH WILL BE PRINTED BY ERRCHK,

```

```

C          AFTER WHICH NONFATAL MESSAGES WILL NOT BE PRINTED.
C          (DEFAULT VALUE IS -1.)
C      NTRACE - .GE.1 WILL CAUSE A TRACE-BACK TO BE GIVEN,
C              IF THIS FEATURE IS IMPLEMENTED ON THIS SYSTEM.
C              .LE.0 WILL SUPPRESS ANY TRACE-BACK, EXCEPT FOR
C              CASES WHEN EXECUTION IS TERMINATED.
C              (DEFAULT VALUE IS 0.)
C
C      *NOTE* -- SOME CALLS TO ERRCHK WILL CAUSE UNCONDITIONAL
C      TERMINATION OF EXECUTION.  ERXSET HAS NO EFFECT ON SUCH CALLS.
C
C      EXAMPLES
C      1. TO PRINT UP TO 100 MESSAGES AS NONFATAL WARNINGS USE
C          CALL ERXSET(100,0)
C      2. TO SUPPRESS ALL MATHLIB WARNING MESSAGES USE
C          CALL ERXSET(0,0)
C
C      ERXSET USES SUBROUTINES ERSTGT
C
C          CALL ERSTGT(0,NFATAL,NTRACE)
C          RETURN
C          END
C*****
C SUBROUTINE FDUMP IS AN ERROR CHECKING ROUTINE USED IN THE
C POSTIVE DEFINITE CHECK OF THE CORRELATION MATRIX
C
C      SUBROUTINE FDUMP
C      ***BEGIN PROLOGUE  FDUMP
C      ***DATE WRITTEN    790801   (YYMMDD)
C      ***REVISION DATE   820801   (YYMMDD)
C      ***CATEGORY NO.    Z
C      ***KEYWORDS        ERROR,XERROR PACKAGE
C      ***AUTHOR          JONES, R. E., (SNLA)
C      ***PURPOSE          SYMBOLIC DUMP (SHOULD BE LOCALLY WRITTEN)
C      ***DESCRIPTION
C          ***NOTE*** MACHINE DEPENDENT ROUTINE
C          FDUMP IS INTENDED TO BE REPLACED BY A LOCALLY WRITTEN
C          VERSION WHICH PRODUCES A SYMBOLIC DUMP.  FAILING THIS,
C          IT SHOULD BE REPLACED BY A VERSION WHICH PRINTS THE
C          SUBPROGRAM NESTING LIST.  NOTE THAT THIS DUMP MUST BE
C          PRINTED ON EACH OF UP TO FIVE FILES, AS INDICATED BY THE
C          XGETUA ROUTINE.  SEE XSETUA AND XGETUA FOR DETAILS.
C
C          WRITTEN BY RON JONES, WITH SLATEC COMMON MATH LIBRARY SUBCOMMITTEE
C          LATEST REVISION --- 23 MAY 1979
C      ***REFERENCES      JONES R.E., *SLATEC COMMON MATHEMATICAL LIBRARY ERROR
C                          HANDLING PACKAGE*, SAND78-1189, SANDIA LABORATORIES,
C                          1978.
C      ***ROUTINES CALLED (NONE)
C      ***END PROLOGUE  FDUMP
C
C      ***FIRST EXECUTABLE STATEMENT  FDUMP
C          RETURN
C          END
C*****
C SUBROUTINE FINDIT IS USED IN THE POSITIVE DEFINITE CHECK
C OF THE CORRELATION MATRIX
C
C      SUBROUTINE FINDIT(NP,ITEST,M,EIG,ICONV)
C      PARAMETER (NMAX=1000)
C      PARAMETER (NVAR=150)
C      PARAMETER (LENT=125)
CBHL
C      CHARACTER*1 TITLE

```

```

CBHL
CBHL
C      COMMON/PARAM/TITLE(LENT), ISEED,N,NV, IRS, ICM,NREP, IDATA, IHIST,
C      1      ICORR, IDIST(NVAR), IRP
C      COMMON/PARAM/ISEED,N,NV, IRS, ICM,NREP, IDATA, IHIST,
C      1      ICORR, IDIST(NVAR), IRP
C      COMMON/PARAM/TITLE(LENT)
CBHL
      COMMON/CMATR/CORR((NVAR*(NVAR+1))/2), LCM(NVAR), NCM
      COMMON/PDMAT/Z(NVAR,NVAR), D(NVAR)
      COMMON/SAMP/X(NMAX*NVAR)
      LOC(I,J)=(J-1)*N+I
      NEV = 0
      DO 9 I = 1,NP
9      IF(D(I).LT.0.0)NEV=NEV+1
      IF(NEV.EQ.0)THEN
          ICONV=1
          RETURN
      ELSE
          DO 11 I = 1,NEV
11         D(I)=EIG
          L1=NEV+1
          L2=NEV+NEV
          DO 12 I = L1,L2
          IF(D(I).LT.EIG)D(I)=EIG
12        CONTINUE
          DO 4 I = 1,M
          DO 4 J = 1,M
          4      X(LOC(I,J)) = 0.0
          DO 5 I = 1,NP
          DO 5 J = 1,NP
          DO 5 K = 1,NP
          5      X(LOC(I,J)) = X(LOC(I,J)) + Z(I,K)*D(K)*Z(J,K)
          DO 30 I = 1,NP
          30     X(LOC(I,I)) = 1.0
          KI = 0
          DO 10 I = 1,NP
          DO 10 J = 1,I
          KI = KI + 1
          CORR(KI) = X(LOC(I,J))
          10    CONTINUE
          ENDIF
          RETURN
          END
C*****
C SUBROUTINE FINVER IS USED IN GENERATING THE BETA DISTRIBUTION
C
C      SUBROUTINE FINVER(FUN,Y,X,DELX,ABSERR,RELERR,FMON)
C
C      WRITTEN BY D.E.AMOS AND S.L.DANIEL, SEPTEMBER,1969.
C
C      REFERENCES
C      SC-DR-72-0917
C
C      ELEMENTS OF NUMERICAL ANALYSIS BY P. HENRICI, WILEY, 1964,
C      PP. 211-213
C
C      ABSTRACT
C      FINVER INVERTS F(X)=Y OR FINDS A ROOT OF F(X)-Y=0.
C      AN INITIAL ESTIMATE FOR X MUST BE SUPPLIED AS WELL AS AN
C      INCREMENT FOR SEARCHING OUT A CHANGE IN SIGN OF F(X)-Y.
C      MONOTONICITY OF F IN THE AREA OF SEARCH MUST ALSO BE
C      SPECIFIED. ONCE BOUNDS ARE ESTABLISHED ON A ROOT,
C      THE ESTIMATE IS REFINED BY LINEAR INTERPOLATION,
C      WHICH GIVES A THIRD POINT. SUCCESSIVE UPPER AND LOWER
C      BOUNDS ARE REFINED REPEATEDLY BY QUADRATIC INTERPOLATION.

```



```

C
C DESCRIPTION OF ARGUMENTS
C
C INPUT
C   FUN      - AN EXTERNAL SUBROUTINE FOR F(X) TO BE SUPPLIED
C              BY THE USER WITH THE CALL LIST
C
C              SUBROUTINE FUN(X,FOFX)
C
C              AN EXTERNAL STATEMENT
C
C              EXTERNAL FUN
C
C              MUST APPEAR IN THE CALLING PROGRAM.
C   Y        - GIVEN FUNCTION VALUE
C   X        - INITIAL ESTIMATE FOR THE SOLUTION OF F(X)-Y=0
C   DELX     - SEARCH INCREMENT FOR BOUNDING THE ROOT, DELX.GT.0.0
C   ABSERR   - ABSOLUTE ERROR TOLERANCE IN MIXED ERROR TEST
C              ABS(F(X)-Y).LE.RELERR*ABS(Y)+ABSERR.
C              RELERR=0.0 FOR A PURE ABSOLUTE ERROR.
C              0.LE.ABSERR.LE.1.
C   RELERR   - RELATIVE ERROR TOLERANCE IN MIXED ERROR TEST
C              ABS(F(X)-Y).LE.RELERR*ABS(Y)+ABSERR.
C              ABSERR=0.0 FOR A PURE RELATIVE ERROR.
C              0.LE.RELERR.LE.1.
C   FMON     - A MONOTONICITY INDICATOR FOR F(X)
C              FMON=-1. MEANS F IS MONOTONE DECREASING,
C              FMON=+1. MEANS F IS MONOTONE INCREASING
C              IN THE VICINITY OF X.
C
C OUTPUT
C   X        - RESULT FOR THE SOLUTION OF F(X)=Y
C
C ERROR CONDITIONS
C   IMPROPER VALUES FOR DELX OR FMON ARE FATAL ERRORS.
C
C
C FINVER USES SUBROUTINES ERRCHK, ERRGET, ERRPRT, ERXSET, ERSTGT
C
C EXTERNAL FUN
C IF(FMON.NE.-1.0 .AND. FMON.NE.1.0) GO TO 901
C IF(DELX.EQ.0.0) GO TO 902
C IF(ABSERR.LT.0.0 .OR. ABSERR.GT.1.) GO TO 903
C IF(RELERR.LT.0.0 .OR. RELERR.GT.1.) GO TO 904
C SIGN=FMON/ ABS(FMON)
C FLAG=0.
C ALFA=SIGN*Y
C D=DELX
C IFLAG=0
C PTOL=RELERR*ABS(ALFA)+ABSERR
C
C
C COMPUTATION OF FOFX
100 CONTINUE
C
C CALL FUN(X,FOFX)
C
C
C FOFX=SIGN*FOFX
C F2=FOFX-ALFA
C FAB= ABS(F2)
C IF(FAB-PTOL) 200,200,86
86 CONTINUE
C IF(IFLAG.EQ.1) GO TO 36
C

```

```

C      BOUNDING THE ZERO OF F(X)-Y
C
      IF(FLAG) 20,21,20
21  IF(F2) 25,200,85
20  IF(F2) 30,200,70
30  IF(F1) 40,75,80
25  FLAG=1.
40  XP=X
      X=X+D
      F1=F2
      GO TO 100
70  IF(F1) 80,75,90
85  FLAG=1.
90  XP=X
      X=X-D
      F1=F2
      GO TO 100
75  F2=F1
      X=XP
      GO TO 100
      END PROGRAM FOR BOUNDING THE ZERO
C
C      MULLERS METHOD FOR INVERSE INTERPOLATION
C      HENRICI, ELEMENTS OF NUMERICAL ANALYSIS, P.212
80  CONTINUE
      F0=F1
      F1=F2
      X00=XP
      X10=X
      XP1 = X
      X11=(F1*X00-F0*X10)/(F1-F0)
      X20=X11
      IF(X20.GE.XP.AND.X20.LE.XP1) GO TO 88
      IF(X20.GE.XP1.AND.X20.LE.XP) GO TO 88
      X=.5*(XP+X)
      GO TO 100
88  X=X20
      IFLAG = 1
      GO TO 100
36  IFLAG=0
      DEN1=F2-F1
      IF(DEN1.EQ.0.0) GO TO 205
      DEN2 = F2-F0
      IF(DEN2.EQ.0.0) GO TO 205
      X21 = (F2*X10-F1*X20)/DEN1
      X22 = (F2*X11-F0*X21)/DEN2
      IF(X22.GE.XP.AND.X22.LE.XP1) GO TO 99
      IF(X22.GE.XP1.AND.X22.LE.XP) GO TO 99
      X=.5*(X00+X10)
      D=.5*D
      GO TO 100
99  F0=F1
      F1=F2
      X00=X10
      X10=X20
      X11=X21
      X20=X22
      GO TO 88
200  CONTINUE
C
C      FAB=ABS(FOFX-Y)
C
      RETURN
205  IFLAG = 0
      FLAG = 0.
      D=D*.5

```

```

      GO TO 100
901 CALL ERRCHK(54,54HIN FINVER, MONOTONIC INDICATOR, FMON, IS NOT 1.
      1OR -1.)
      RETURN
902 CALL ERRCHK(23,23HIN FINVER, DELX IS ZERO)
      RETURN
903 CALL ERRCHK(37,37HIN FINVER, IMPROPER INPUT FOR ABSERR.)
      RETURN
904 CALL ERRCHK(37,37HIN FINVER, IMPROPER INPUT FOR RELERR.)
      RETURN
      END
C*****
C SUBROUTINE FINVNOR IS USED IN GENERATING THE NORMAL AND
C LOGNORMAL DISTRIBUTIONS
C
      FUNCTION FINVNOR (X)
      IF (X-0.5) 20,10,30
10 FINVNOR=0.0
      RETURN
20 F=-1.0
      Y=X
      GO TO 40
30 F=1.0
      Y=1.0-X
40 FINVNOR=SQRT(2.0)*F*RIERFCL(2.*Y)
      RETURN
      END
C*****
C SUBROUTINE HISTO GENERATES HISTOGRAMS OF THE VARIABLES
C
      SUBROUTINE HISTO
      PARAMETER (NMAX=1000)
      PARAMETER (NVAR=150)
      PARAMETER (LENT=125)
      PARAMETER (IFLG=20)
CBHL
      CHARACTER*1 TITLE,KX
CBHL
CBHL
C      COMMON/PARAM/TITLE(LENT), ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
C      1 ICORR,IDIST(NVAR),IRP
      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
      1 ICORR,IDIST(NVAR),IRP
      COMMON/PARAMA/TITLE(LENT)
CBHL
      COMMON/RANK/XV(NMAX),RXV(NMAX),IWK(NMAX)
      SAVE KX
C      THIS IS A MODIFICATION OF DISTAT -- IT MAKES THE HISTOGRAM
C      AND PRINTS THE MEAN AND THE MEDIAN
C      N = NO. OF OBSERVATIONS
C      XV = ARRAY NAME
      DATA KX/1HX/
      IF (N-1) 10,20,30
10 WRITE(6,260)
      RETURN
20 WRITE(6,270)XV(1)
      RETURN
30 CONTINUE
      CALL SIFT (XV,N)
      SUM=0.0
      SUMSQ=0.0
      DO 40 I=1,N
      SUM=SUM+XV(I)
      SUMSQ=SUMSQ+XV(I)*XV(I)
40 CONTINUE
      XMEAN=SUM/FLOAT(N)

```

```

XVAR=(SUMSQ-FLOAT(N)*XMEAN*XMEAN)/FLOAT(N)
IF(IRS.NE.0)XVAR=XVAR*FLOAT(N)/FLOAT(N-1)
N2=N/2
LOC=(N+1)/2-N2
FMED=(XV(N2+LOC)+XV((N+1-LOC)/2+1))/2.0
R=XV(N)-XV(1)
IF(R.NE.0.0)GO TO 50
WRITE(6,280)
GO TO 250
50 CONTINUE
M=1
SIZE=R/IFLG
POWER=LOG10(SIZE)
IF(POWER.GE.0.)GO TO 60
POWER=AIN(T(POWER))
GO TO 70
60 POWER=AIN(T(POWER))+1.
70 SIZE=SIZE/10.**POWER
CON=.01
80 IF(SIZE.LE.(CON+.005))GO TO 90
CON=CON+.01
GO TO 80
90 SIZE=CON*10.**POWER
TEMP=XV(1)/SIZE
IF(TEMP)100,100,110
100 TEMP=AIN(T(TEMP-1.0))
GO TO 120
110 TEMP=AIN(T(TEMP))
120 IF(XV(1)-TEMP*SIZE.GT.0.0)GO TO 130
TEMP=TEMP-0.5
130 CELL=TEMP*SIZE+SIZE
POINT=TEMP*SIZE+SIZE*0.5
I=0
KSUM=0
WRITE(6,290)
K=-1
M=1
140 I=I+1
K=K+1
150 IF(I-N)160,160,190
160 IF(XV(I)-CELL)140,140,170
170 IF(K)180,180,200
180 WRITE(6,310)POINT,K
GO TO 240
190 M=2
200 IF(K-90)210,210,220
210 KK=K
GO TO 230
220 KK=90
230 WRITE(6,310)POINT,K,(KX,J=1,KK)
KSUM=KSUM+K
IF(M.GT.1)GO TO 250
K=0
240 CELL=CELL+SIZE
POINT=POINT+SIZE
GO TO 150
250 WRITE(6,320)KSUM
WRITE(6,300)XV(1),XV(N),R,XMEAN,FMED,XVAR
RETURN
C
260 FORMAT(' N IS ZERO',//)
270 FORMAT(' ONE OBS. ',E17.8,//)
280 FORMAT(' NO HISTOGRAM - RANGE =0',//)
290 FORMAT(/,5X,'MIDPOINT',10X,'FREQ.',//)
300 FORMAT(/,6X,'MIN',12X,'MAX',11X,'RANGE',11X,'MEAN',10X,
1 'MEDIAN',8X,'VARIANCE',//,1X,6G15.7,/)

```

```

310 FORMAT(1X,G15.7,5X,I5,4X,90A1)
320 FORMAT('0',20X,I5)
END
C*****
C SUBROUTINE HPSRT IS USED IN THE RANKING OF THE SAMPLED DATA
C
      SUBROUTINE HPSRT
      PARAMETER (NMAX=1000)
      PARAMETER (NVAR=150)
      PARAMETER (LENT=125)
CBHL
      CHARACTER*1 TITLE
CBHL
CBHL
C      COMMON/PARAM/TITLE(LENT), ISEED,N,NV,IRS,ICM,NREP, IDATA, IHIST,
C      1      ICORR, IDIST(NVAR), IRP
      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP, IDATA, IHIST,
      1      ICORR, IDIST(NVAR), IRP
      COMMON/PARAM/TITLE(LENT)
CBHL
      COMMON/RANK/XV(NMAX),RXV(NMAX),IWK(NMAX)
      INTEGER R
      L=N/2+1
      R=N
10    IF (L.LE.1) GO TO 70
      L=L-1
      XHOLD=XV(L)
      YHOLD=RXV(L)
20    J=L
30    I=J
      J=2*J
      IF (J-R) 40,50,60
40    IF (XV(J).LT.XV(J+1)) J=J+1
50    IF (XHOLD.GE.XV(J)) GO TO 60
      XV(I)=XV(J)
      RXV(I)=RXV(J)
      GO TO 30
60    XV(I)=XHOLD
      RXV(I)=YHOLD
      GO TO 10
70    XHOLD=XV(R)
      YHOLD=RXV(R)
      XV(R)=XV(1)
      RXV(R)=RXV(1)
      R=R-1
      IF (R.GT.1) GO TO 20
      XV(1)=XHOLD
      RXV(1)=YHOLD
      RETURN
      END
C*****
C SUBROUTINE HSTOUT IS USED TO GENERATE HISTOGRAMS OF THE
C VARIABLES
C
      SUBROUTINE HSTOUT
      PARAMETER (NMAX=1000)
      PARAMETER (NVAR=150)
      PARAMETER (LEND=9)
      PARAMETER (LENT=125)
CBHL
      CHARACTER*1 TITLE
CBHL
CBHL
C      COMMON/PARAM/TITLE(LENT), ISEED,N,NV,IRS,ICM,NREP, IDATA, IHIST,
C      1      ICORR, IDIST(NVAR), IRP
      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP, IDATA, IHIST,

```

```

1          ICORR, IDIST(NVAR), IRP
COMMON/PARAMA/TITLE(LENT)
CBHL
COMMON/SAMP/X(NMAX*NVAR)
COMMON/RANK/XV(NMAX),RXV(NMAX),IWK(NMAX)
CBHL  CHARACTER*13 DIST(LENT)
C      SAVE DIST
C      DATA DIST(1)/'BETA'//,DIST(2)/'NORMAL'//,DIST(3)/'LOGNORMAL'//,
C      1      DIST(4)/'UNIFORM'//,DIST(5)/'LOGUNIFORM'//,
C      2      DIST(6)/'UNIFORM*'//,DIST(7)/'LOGUNIFORM*'//,
CBHL  3      DIST(8)/'TRIANGULAR'//,DIST(9)/'USER SUPPLIED'/
CHARACTER*13 DDIST(LENT)
SAVE DDIST
DATA DDIST(1)/'BETA'//,DDIST(2)/'NORMAL'//,DDIST(3)/'LOGNORMAL'//,
1      DDIST(4)/'UNIFORM'//,DDIST(5)/'LOGUNIFORM'//,
2      DDIST(6)/'UNIFORM*'//,DDIST(7)/'LOGUNIFORM*'//,
3      DDIST(8)/'TRIANGULAR'//,DDIST(9)/'USER SUPPLIED'/
CBHL
LOC(I,J)=(J-1)*N+I
REWIND 4
READ(4)X
DO 590 I=1,NV
    IDT=IDIST(I)
    WRITE(6,9001)TITLE
    WRITE(6,9002)I,DDIST(IDT)
    DO 530 J=1,N
530      XV(J)=X(LOC(J,I))
        CALL HISTO
590 CONTINUE
9001 FORMAT('1',3X,125A1)
9002 FORMAT('0  HISTOGRAM FOR VARIABLE NO.',I3,5X,A,' DISTRIBUTION')
RETURN
END
C*****
C FUNCTION HYPGEO IS USED IN GENERATING A BETA DISTRIBUTION
C
    FUNCTION HYPGEO(A,B,C,X,SUM)
C
C      FUNCTION HYPGEO SUMS THE SERIES FOR F(A,B,C,X) WHEN SUM=1. OR
C      F(A,B,C,X)-1. WHEN SUM=0. WHERE F(A,B,C,X) IS THE GAUSS
C      HYPERGEOMETRIC FUNCTION.
C      UPDATED, JANUARY, 1975
C
    KMAX=1000
    TOL=MAX(1.0E-14,R1MACH(4))
    M=2.-SUM
    FSUM=X*A*(B/C)
    TK=FSUM
    TLIM=TOL
    IF(M.EQ.2) TLIM=TLIM*ABS(FSUM)
    AK=1.
    DO 10 K=2,KMAX
    TK=TK*((A+AK)/(AK+1.))*((B+AK)/(C+AK))*X
    FSUM=FSUM+TK
    IF(ABS(TK).LE.TLIM) GO TO 20
    AK=AK+1.
10 CONTINUE
    CALL ERRCHK(35,35HIN HYPGEO, NO CONVERGENCE OF SERIES)
    RETURN
20 HYPGEO=FSUM+SUM
    RETURN
    END
C*****
C FUNCTION ILMACH SETS INTEGER MACHINE DEPENDENT CONSTANTS
C
    INTEGER FUNCTION ILMACH(I)

```

```

C***BEGIN PROLOGUE  ILMACH
C***DATE WRITTEN   750101   (YYMMDD)
C***REVISION DATE  801001   (YYMMDD)
C***CATEGORY NO.   R1
C***KEYWORDS       MACHINE CONSTANTS
C***AUTHOR         FOX, P. A., (BELL LABS)
C                   HALL, A. D., (BELL LABS)
C                   SCHRYER, N. L., (BELL LABS)
C***PURPOSE        Returns integer machine dependent constants
C***DESCRIPTION
C * * * * *
C   These machine constant routines must be activated for
C   a particular environment.
C * * * * *
C
C   ILMACH can be used to obtain machine-dependent parameters
C   for the local machine environment. It is a function
C   subroutine with one (input) argument, and can be called
C   as follows, for example
C
C       K = ILMACH(I)
C
C   where I=1,...,16. The (output) value of K above is
C   determined by the (input) value of I. The results for
C   various values of I are discussed below.
C
C I/O unit numbers.
C   ILMACH( 1) = the standard input unit.
C   ILMACH( 2) = the standard output unit.
C   ILMACH( 3) = the standard punch unit.
C   ILMACH( 4) = the standard error message unit.
C
C Words.
C   ILMACH( 5) = the number of bits per integer storage unit.
C   ILMACH( 6) = the number of characters per integer storage unit.
C
C Integers.
C   assume integers are represented in the S-digit, base-A form
C
C       sign ( X(S-1)*A**(S-1) + ... + X(1)*A + X(0) )
C
C       where 0 .LE. X(I) .LT. A for I=0,...,S-1.
C   ILMACH( 7) = A, the base.
C   ILMACH( 8) = S, the number of base-A digits.
C   ILMACH( 9) = A**S - 1, the largest magnitude.
C
C Floating-Point Numbers.
C   Assume floating-point numbers are represented in the T-digit,
C   base-B form
C
C       sign (B**E)*( (X(1)/B) + ... + (X(T)/B**T) )
C
C       where 0 .LE. X(I) .LT. B for I=1,...,T,
C       0 .LT. X(1), and EMIN .LE. E .LE. EMAX.
C   ILMACH(10) = B, the base.
C
C Single-Precision
C   ILMACH(11) = T, the number of base-B digits.
C   ILMACH(12) = EMIN, the smallest exponent E.
C   ILMACH(13) = EMAX, the largest exponent E.
C
C Double-Precision
C   ILMACH(14) = T, the number of base-B digits.
C   ILMACH(15) = EMIN, the smallest exponent E.
C   ILMACH(16) = EMAX, the largest exponent E.
C
C To alter this function for a particular environment,

```

```

C the desired set of DATA statements should be activated by
C removing the C from column 1. Also, the values of
C ILMACH(1) - ILMACH(4) should be checked for consistency
C with the local operating system.
C***REFERENCES FOX P.A., HALL A.D., SCHRYER N.L.,*FRAMEWORK FOR A
C PORTABLE LIBRARY*, ACM TRANSACTIONS ON MATHEMATICAL
C SOFTWARE, VOL. 4, NO. 2, JUNE 1978, PP. 177-188.
C***ROUTINES CALLED (NONE)
C***END PROLOGUE ILMACH
C
C INTEGER IMACH(16),OUTPUT
C EQUIVALENCE (IMACH(4),OUTPUT)
C SAVE IMACH
C
C MACHINE CONSTANTS FOR THE BURROUGHS 1700 SYSTEM.
C
C DATA IMACH( 1) / 7 /
C DATA IMACH( 2) / 2 /
C DATA IMACH( 3) / 2 /
C DATA IMACH( 4) / 2 /
C DATA IMACH( 5) / 36 /
C DATA IMACH( 6) / 4 /
C DATA IMACH( 7) / 2 /
C DATA IMACH( 8) / 33 /
C DATA IMACH( 9) / Z1FFFFFFFF /
C DATA IMACH(10) / 2 /
C DATA IMACH(11) / 24 /
C DATA IMACH(12) / -256 /
C DATA IMACH(13) / 255 /
C DATA IMACH(14) / 60 /
C DATA IMACH(15) / -256 /
C DATA IMACH(16) / 255 /
C
C MACHINE CONSTANTS FOR THE BURROUGHS 5700 SYSTEM.
C
C DATA IMACH( 1) / 5 /
C DATA IMACH( 2) / 6 /
C DATA IMACH( 3) / 7 /
C DATA IMACH( 4) / 6 /
C DATA IMACH( 5) / 48 /
C DATA IMACH( 6) / 6 /
C DATA IMACH( 7) / 2 /
C DATA IMACH( 8) / 39 /
C DATA IMACH( 9) / 0000777777777777 /
C DATA IMACH(10) / 8 /
C DATA IMACH(11) / 13 /
C DATA IMACH(12) / -50 /
C DATA IMACH(13) / 76 /
C DATA IMACH(14) / 26 /
C DATA IMACH(15) / -50 /
C DATA IMACH(16) / 76 /
C
C MACHINE CONSTANTS FOR THE BURROUGHS 6700/7700 SYSTEMS.
C
C DATA IMACH( 1) / 5 /
C DATA IMACH( 2) / 6 /
C DATA IMACH( 3) / 7 /
C DATA IMACH( 4) / 6 /
C DATA IMACH( 5) / 48 /
C DATA IMACH( 6) / 6 /
C DATA IMACH( 7) / 2 /
C DATA IMACH( 8) / 39 /
C DATA IMACH( 9) / 0000777777777777 /
C DATA IMACH(10) / 8 /
C DATA IMACH(11) / 13 /
C DATA IMACH(12) / -50 /

```



```

C      DATA IMACH(13) / 76 /
C      DATA IMACH(14) / 26 /
C      DATA IMACH(15) / -32754 /
C      DATA IMACH(16) / 32780 /
C
C      MACHINE CONSTANTS FOR THE CDC 6000/7000 SERIES.
C
C      DATA IMACH( 1) / 5 /
C      DATA IMACH( 2) / 6 /
C      DATA IMACH( 3) / 7 /
C      DATA IMACH( 4) / 6LOUTPUT /
C      DATA IMACH( 5) / 60 /
C      DATA IMACH( 6) / 10 /
C      DATA IMACH( 7) / 2 /
C      DATA IMACH( 8) / 48 /
C      DATA IMACH( 9) / 0000777777777777777B /
C      DATA IMACH(10) / 2 /
C      DATA IMACH(11) / 47 /
C      DATA IMACH(12) / -929 /
C      DATA IMACH(13) / 1070 /
C      DATA IMACH(14) / 94 /
C      DATA IMACH(15) / -929 /
C      DATA IMACH(16) / 1069 /
C
C      MACHINE CONSTANTS FOR THE CRAY 1
C
C      DATA IMACH( 1) / 5 /
C      DATA IMACH( 2) / 6 /
C      DATA IMACH( 3) / 102 /
C      DATA IMACH( 4) / 6 /
C      DATA IMACH( 5) / 64 /
C      DATA IMACH( 6) / 8 /
C      DATA IMACH( 7) / 2 /
C      DATA IMACH( 8) / 63 /
C      DATA IMACH( 9) / 777777777777777777777B /
C      DATA IMACH(10) / 2 /
C      DATA IMACH(11) / 47 /
C      DATA IMACH(12) / -8189 /
C      DATA IMACH(13) / 8190 /
C      DATA IMACH(14) / 94 /
C      DATA IMACH(15) / -8099 /
C      DATA IMACH(16) / 8190 /
C
C      MACHINE CONSTANTS FOR THE DATA GENERAL ECLIPSE S/200
C
C      DATA IMACH( 1) / 11 /
C      DATA IMACH( 2) / 12 /
C      DATA IMACH( 3) / 8 /
C      DATA IMACH( 4) / 10 /
C      DATA IMACH( 5) / 16 /
C      DATA IMACH( 6) / 2 /
C      DATA IMACH( 7) / 2 /
C      DATA IMACH( 8) / 15 /
C      DATA IMACH( 9) / 32767 /
C      DATA IMACH(10) / 16 /
C      DATA IMACH(11) / 6 /
C      DATA IMACH(12) / -64 /
C      DATA IMACH(13) / 63 /
C      DATA IMACH(14) / 14 /
C      DATA IMACH(15) / -64 /
C      DATA IMACH(16) / 63 /
C
C      MACHINE CONSTANTS FOR THE HARRIS 220
C
C      DATA IMACH( 1) / 5 /
C      DATA IMACH( 2) / 6 /

```

```

C      DATA IMACH( 3) /          0 /
C      DATA IMACH( 4) /          6 /
C      DATA IMACH( 5) /         24 /
C      DATA IMACH( 6) /          3 /
C      DATA IMACH( 7) /          2 /
C      DATA IMACH( 8) /         23 /
C      DATA IMACH( 9) / 8388607 /
C      DATA IMACH(10) /          2 /
C      DATA IMACH(11) /         23 /
C      DATA IMACH(12) /        -127 /
C      DATA IMACH(13) /         127 /
C      DATA IMACH(14) /          38 /
C      DATA IMACH(15) /        -127 /
C      DATA IMACH(16) /         127 /

```

```

C
C      MACHINE CONSTANTS FOR THE HONEYWELL 600/6000 SERIES.
C

```

```

C      DATA IMACH( 1) /          5 /
C      DATA IMACH( 2) /          6 /
C      DATA IMACH( 3) /         43 /
C      DATA IMACH( 4) /          6 /
C      DATA IMACH( 5) /         36 /
C      DATA IMACH( 6) /          6 /
C      DATA IMACH( 7) /          2 /
C      DATA IMACH( 8) /         35 /
C      DATA IMACH( 9) / 03777777777777 /
C      DATA IMACH(10) /          2 /
C      DATA IMACH(11) /         27 /
C      DATA IMACH(12) /        -127 /
C      DATA IMACH(13) /         127 /
C      DATA IMACH(14) /          63 /
C      DATA IMACH(15) /        -127 /
C      DATA IMACH(16) /         127 /

```

```

C
C      MACHINE CONSTANTS FOR THE HP 2100
C      3 WORD DOUBLE PRECISION OPTION WITH FTN4
C

```

```

C      DATA IMACH(1) /          5 /
C      DATA IMACH(2) /          6 /
C      DATA IMACH(3) /          4 /
C      DATA IMACH(4) /          1 /
C      DATA IMACH(5) /         16 /
C      DATA IMACH(6) /          2 /
C      DATA IMACH(7) /          2 /
C      DATA IMACH(8) /         15 /
C      DATA IMACH(9) /    32767 /
C      DATA IMACH(10)/          2 /
C      DATA IMACH(11)/         23 /
C      DATA IMACH(12)/        -128 /
C      DATA IMACH(13)/         127 /
C      DATA IMACH(14)/          39 /
C      DATA IMACH(15)/        -128 /
C      DATA IMACH(16)/         127 /

```

```

C
C      MACHINE CONSTANTS FOR THE HP 2100
C      4 WORD DOUBLE PRECISION OPTION WITH FTN4
C

```

```

C      DATA IMACH(1) /          5 /
C      DATA IMACH(2) /          6 /
C      DATA IMACH(3) /          4 /
C      DATA IMACH(4) /          1 /
C      DATA IMACH(5) /         16 /
C      DATA IMACH(6) /          2 /
C      DATA IMACH(7) /          2 /
C      DATA IMACH(8) /         15 /
C      DATA IMACH(9) /    32767 /

```

```

C      DATA IMACH(10)/      2 /
C      DATA IMACH(11)/     23 /
C      DATA IMACH(12)/    -128 /
C      DATA IMACH(13)/     127 /
C      DATA IMACH(14)/     55 /
C      DATA IMACH(15)/    -128 /
C      DATA IMACH(16)/     127 /
C
C      MACHINE CONSTANTS FOR THE IBM 360/370 SERIES,
C      THE XEROX SIGMA 5/7/9, THE SEL SYSTEMS 85/86, AND
C      THE PERKIN ELMER (INTERDATA) 7/32.
C
C      DATA IMACH( 1) /      5 /
C      DATA IMACH( 2) /      6 /
C      DATA IMACH( 3) /      7 /
C      DATA IMACH( 4) /      6 /
C      DATA IMACH( 5) /     32 /
C      DATA IMACH( 6) /      4 /
C      DATA IMACH( 7) /     16 /
C      DATA IMACH( 8) /     31 /
C      DATA IMACH( 9) / Z7FFFFFFF /
C      DATA IMACH(10) /     16 /
C      DATA IMACH(11) /      6 /
C      DATA IMACH(12) /    -64 /
C      DATA IMACH(13) /     63 /
C      DATA IMACH(14) /     14 /
C      DATA IMACH(15) /    -64 /
C      DATA IMACH(16) /     63 /
C
C      MACHINE CONSTANTS FOR THE PDP-10 (KA PROCESSOR).
C
C      DATA IMACH( 1) /      5 /
C      DATA IMACH( 2) /      6 /
C      DATA IMACH( 3) /      5 /
C      DATA IMACH( 4) /      6 /
C      DATA IMACH( 5) /     36 /
C      DATA IMACH( 6) /      5 /
C      DATA IMACH( 7) /      2 /
C      DATA IMACH( 8) /     35 /
C      DATA IMACH( 9) / "377777777777 /
C      DATA IMACH(10) /      2 /
C      DATA IMACH(11) /     27 /
C      DATA IMACH(12) /   -128 /
C      DATA IMACH(13) /     127 /
C      DATA IMACH(14) /     54 /
C      DATA IMACH(15) /   -101 /
C      DATA IMACH(16) /     127 /
C
C      MACHINE CONSTANTS FOR THE PDP-10 (KI PROCESSOR).
C
C      DATA IMACH( 1) /      5 /
C      DATA IMACH( 2) /      6 /
C      DATA IMACH( 3) /      5 /
C      DATA IMACH( 4) /      6 /
C      DATA IMACH( 5) /     36 /
C      DATA IMACH( 6) /      5 /
C      DATA IMACH( 7) /      2 /
C      DATA IMACH( 8) /     35 /
C      DATA IMACH( 9) / "377777777777 /
C      DATA IMACH(10) /      2 /
C      DATA IMACH(11) /     27 /
C      DATA IMACH(12) /   -128 /
C      DATA IMACH(13) /     127 /
C      DATA IMACH(14) /     62 /
C      DATA IMACH(15) /   -128 /
C      DATA IMACH(16) /     127 /

```

```

C
C MACHINE CONSTANTS FOR PDP-11 FORTRAN SUPPORTING
C 32-BIT INTEGER ARITHMETIC.
C
C DATA IMACH( 1) /      5 /
C DATA IMACH( 2) /      6 /
C DATA IMACH( 3) /      5 /
C DATA IMACH( 4) /      6 /
C DATA IMACH( 5) /     32 /
C DATA IMACH( 6) /      4 /
C DATA IMACH( 7) /      2 /
C DATA IMACH( 8) /     31 /
C DATA IMACH( 9) / 2147483647 /
C DATA IMACH(10) /      2 /
C DATA IMACH(11) /     24 /
C DATA IMACH(12) / -127 /
C DATA IMACH(13) /    127 /
C DATA IMACH(14) /     56 /
C DATA IMACH(15) / -127 /
C DATA IMACH(16) /    127 /
C
C MACHINE CONSTANTS FOR PDP-11 FORTRAN SUPPORTING
C 16-BIT INTEGER ARITHMETIC.
C
C DATA IMACH( 1) /      5 /
C DATA IMACH( 2) /      6 /
C DATA IMACH( 3) /      5 /
C DATA IMACH( 4) /      6 /
C DATA IMACH( 5) /     16 /
C DATA IMACH( 6) /      2 /
C DATA IMACH( 7) /      2 /
C DATA IMACH( 8) /     15 /
C DATA IMACH( 9) / 32767 /
C DATA IMACH(10) /      2 /
C DATA IMACH(11) /     24 /
C DATA IMACH(12) / -127 /
C DATA IMACH(13) /    127 /
C DATA IMACH(14) /     56 /
C DATA IMACH(15) / -127 /
C DATA IMACH(16) /    127 /
C
C MACHINE CONSTANTS FOR THE UNIVAC 1100 SERIES. FTN COMPILER
C
C DATA IMACH( 1) /      5 /
C DATA IMACH( 2) /      6 /
C DATA IMACH( 3) /      1 /
C DATA IMACH( 4) /      6 /
C DATA IMACH( 5) /     36 /
C DATA IMACH( 6) /      4 /
C DATA IMACH( 7) /      2 /
C DATA IMACH( 8) /     35 /
C DATA IMACH( 9) / 03777777777777 /
C DATA IMACH(10) /      2 /
C DATA IMACH(11) /     27 /
C DATA IMACH(12) / -128 /
C DATA IMACH(13) /    127 /
C DATA IMACH(14) /     60 /
C DATA IMACH(15) / -1024 /
C DATA IMACH(16) /   1023 /
C
C MACHINE CONSTANTS FOR THE UNIVAC 1100 SERIES. FOR COMPILER
C
C DATA IMACH( 1) /      5 /
C DATA IMACH( 2) /      6 /
C DATA IMACH( 3) /      7 /

```

```

C      DATA IMACH( 4) /      6 /
C      DATA IMACH( 5) /     36 /
C      DATA IMACH( 6) /      6 /
C      DATA IMACH( 7) /      2 /
C      DATA IMACH( 8) /     35 /
C      DATA IMACH( 9) / 037777777777 /
C      DATA IMACH(10) /      2 /
C      DATA IMACH(11) /     27 /
C      DATA IMACH(12) /    -128 /
C      DATA IMACH(13) /     127 /
C      DATA IMACH(14) /     60 /
C      DATA IMACH(15) / -1024 /
C      DATA IMACH(16) /    1023 /
C
C
C      MACHINE CONSTANTS FOR THE VAX 11/780
C
C      DATA IMACH(1) /      5 /
C      DATA IMACH(2) /      6 /
C      DATA IMACH(3) /      5 /
C      DATA IMACH(4) /      6 /
C      DATA IMACH(5) /     32 /
C      DATA IMACH(6) /      4 /
C      DATA IMACH(7) /      2 /
C      DATA IMACH(8) /     31 /
C      DATA IMACH(9) /2147483647 /
C      DATA IMACH(10)/      2 /
C      DATA IMACH(11)/     24 /
C      DATA IMACH(12)/    -127 /
C      DATA IMACH(13)/     127 /
C      DATA IMACH(14)/     56 /
C      DATA IMACH(15)/    -127 /
C      DATA IMACH(16)/     127 /
C
C      MACHINE CONSTANTS FOR THE VAX 11/780. G-FLOATING OPTION
C
C      DATA IMACH(1) /      5 /
C      DATA IMACH(2) /      6 /
C      DATA IMACH(3) /      5 /
C      DATA IMACH(4) /      6 /
C      DATA IMACH(5) /     32 /
C      DATA IMACH(6) /      4 /
C      DATA IMACH(7) /      2 /
C      DATA IMACH(8) /     31 /
C      DATA IMACH(9) /2147483647 /
C      DATA IMACH(10)/      2 /
C      DATA IMACH(11)/     24 /
C      DATA IMACH(12)/    -127 /
C      DATA IMACH(13)/     127 /
C      DATA IMACH(14)/     54 /
C      DATA IMACH(15)/    -511 /
C      DATA IMACH(16)/     511 /
C
C***FIRST EXECUTABLE STATEMENT  IIMACH
C
C      IF ( I .LT. 1 .OR.  I .GT. 16) GO TO 10
C
C      IIMACH=IMACH(I)
C      RETURN
C
C      10 CONTINUE
C      WRITE(OUTPUT,9000)
9000  FORMAT(39H1ERROR      1 IN IIMACH - I OUT OF BOUNDS )
C
C      CALL FDUMP
C

```

```

C
      STOP
      END
C*****
C SUBROUTINE IMTQL2 IS USED IN THE POSTIVE DEFINITE CHECK OF THE
C CORRELATION MATRIX
C
      SUBROUTINE IMTQL2(NM,N,D,E,Z,IERR)
C***BEGIN PROLOGUE  IMTQL2
C***DATE WRITTEN   760101   (YYMMDD)
C***REVISION DATE  830518   (YYMMDD)
C***CATEGORY NO.   D4A5,D4C2A
C***KEYWORDS       EISPACK,EIGENVALUES,EIGENVECTORS
C***AUTHOR         SMITH ET AL
C***PURPOSE        COMPUTES EIGENVALUES AND EIGENVECTORS OF SYMMETRIC
C                   TRIDIAGONAL MATRIX USING IMPLICIT QL METHOD.
C***DESCRIPTION
C   THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE IMTQL2,
C   NUM. MATH. 12, 377-383(1968) BY MARTIN AND WILKINSON,
C   AS MODIFIED IN NUM. MATH. 15, 450(1970) BY DUBRULLE.
C   HANDBOOK FOR AUTO. COMP., VOL.II-LINEAR ALGEBRA, 241-248(1971).
C
C   THIS SUBROUTINE FINDS THE EIGENVALUES AND EIGENVECTORS
C   OF A SYMMETRIC TRIDIAGONAL MATRIX BY THE IMPLICIT QL METHOD.
C   THE EIGENVECTORS OF A FULL SYMMETRIC MATRIX CAN ALSO
C   BE FOUND IF TRED2 HAS BEEN USED TO REDUCE THIS
C   FULL MATRIX TO TRIDIAGONAL FORM.
C
C   ON INPUT
C
C       NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL
C       ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM
C       DIMENSION STATEMENT.
C
C       N IS THE ORDER OF THE MATRIX.
C
C       D CONTAINS THE DIAGONAL ELEMENTS OF THE INPUT MATRIX.
C
C       E CONTAINS THE SUBDIAGONAL ELEMENTS OF THE INPUT MATRIX
C       IN ITS LAST N-1 POSITIONS.  E(1) IS ARBITRARY.
C
C       Z CONTAINS THE TRANSFORMATION MATRIX PRODUCED IN THE
C       REDUCTION BY TRED2, IF PERFORMED.  IF THE EIGENVECTORS
C       OF THE TRIDIAGONAL MATRIX ARE DESIRED, Z MUST CONTAIN
C       THE IDENTITY MATRIX.
C
C   ON OUTPUT
C
C       D CONTAINS THE EIGENVALUES IN ASCENDING ORDER.  IF AN
C       ERROR EXIT IS MADE, THE EIGENVALUES ARE CORRECT BUT
C       UNORDERED FOR INDICES 1,2,...,IERR-1.
C
C       E HAS BEEN DESTROYED.
C
C       Z CONTAINS ORTHONORMAL EIGENVECTORS OF THE SYMMETRIC
C       TRIDIAGONAL (OR FULL) MATRIX.  IF AN ERROR EXIT IS MADE,
C       Z CONTAINS THE EIGENVECTORS ASSOCIATED WITH THE STORED
C       EIGENVALUES.
C
C       IERR IS SET TO
C           ZERO      FOR NORMAL RETURN,
C           J         IF THE J-TH EIGENVALUE HAS NOT BEEN
C                   DETERMINED AFTER 30 ITERATIONS.
C
C   CALLS PYTHAG(A,B) FOR  $\sqrt{A^2 + B^2}$ .
C

```

```

C      QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,
C      APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY
C      -----
C***REFERENCES *MATRIX EIGENSYSTEM ROUTINES-EISPACKGUIDE*,
C              B.T.SMITH,J.M.BOYLE,J.J.DONGARRA,B.S.GARBOW,
C              Y.I.KEBE,V.C.KLEMA,C.B.MOLER,SPRINGER-VERLAG,
C              1976.
C***ROUTINES CALLED PYTHAG
C***END PROLOGUE IMTQL2
C
C      INTEGER I,J,K,L,M,N,II,NM,MML,IERR
C      REAL D(N),E(N),Z(NM,N)
C      REAL B,C,F,G,P,R,S,S1,S2
C      REAL PYTHAG
C
C***FIRST EXECUTABLE STATEMENT IMTQL2
C      IERR = 0
C      IF (N .EQ. 1) GO TO 1001
C
C      DO 100 I = 2, N
100  E(I-1) = E(I)
C
C      E(N) = 0.0E0
C
C      DO 240 L = 1, N
C      J = 0
C      ..... LOOK FOR SMALL SUB-DIAGONAL ELEMENT .....
105  DO 110 M = L, N
C      IF (M .EQ. N) GO TO 120
C      S1 = ABS(D(M)) + ABS(D(M+1))
C      S2 = S1 + ABS(E(M))
C      IF (S2 .EQ. S1) GO TO 120
110  CONTINUE
C
C      120  P = D(L)
C      IF (M .EQ. L) GO TO 240
C      IF (J .EQ. 30) GO TO 1000
C      J = J + 1
C      ..... FORM SHIFT .....
C      G = (D(L+1) - P) / (2.0E0 * E(L))
C      R = PYTHAG(G,1.0E0)
C      G = D(M) - P + E(L) / (G + SIGN(R,G))
C      S = 1.0E0
C      C = 1.0E0
C      P = 0.0E0
C      MML = M - L
C      ..... FOR I=M-1 STEP -1 UNTIL L DO -- .....
C      DO 200 II = 1, MML
C      I = M - II
C      F = S * E(I)
C      B = C * E(I)
C      IF (ABS(F) .LT. ABS(G)) GO TO 150
C      C = G / F
C      R = SQRT(C*C+1.0E0)
C      E(I+1) = F * R
C      S = 1.0E0 / R
C      C = C * S
C      GO TO 160
150  S = F / G
C      R = SQRT(S*S+1.0E0)
C      E(I+1) = G * R
C      C = 1.0E0 / R
C      S = S * C
160  G = D(I+1) - P
C      R = (D(I) - G) * S + 2.0E0 * C * B
C      P = S * R

```

```

        D(I+1) = G + P
        G = C * R - B
C      ..... FORM VECTOR .....
        DO 180 K = 1, N
            F = Z(K,I+1)
            Z(K,I+1) = S * Z(K,I) + C * F
            Z(K,I) = C * Z(K,I) - S * F
180    CONTINUE
C
200    CONTINUE
C
        D(L) = D(L) - P
        E(L) = G
        E(M) = 0.0E0
        GO TO 105
240 CONTINUE
C      ..... ORDER EIGENVALUES AND EIGENVECTORS .....
        DO 300 II = 2, N
            I = II - 1
            K = I
            P = D(I)
C
            DO 260 J = II, N
                IF (D(J) .GE. P) GO TO 260
                K = J
                P = D(J)
260    CONTINUE
C
            IF (K .EQ. I) GO TO 300
            D(K) = D(I)
            D(I) = P
C
            DO 280 J = 1, N
                P = Z(J,I)
                Z(J,I) = Z(J,K)
                Z(J,K) = P
280    CONTINUE
C
300 CONTINUE
C
        GO TO 1001
C      ..... SET ERROR -- NO CONVERGENCE TO AN
C      EIGENVALUE AFTER 30 ITERATIONS .....
1000 IERR = L
1001 RETURN
        END
C*****
C FUNCTION J4SAVE IS USED IN THE POSTIVE DEFINITE CHECK OF THE
C CORRELATION MATRIX
C
        FUNCTION J4SAVE(IWHICH,IVALUE,ISET)
C***BEGIN PROLOGUE J4SAVE
C***DATE WRITTEN 790801 (YYMMDD)
C***REVISION DATE 820801 (YYMMDD)
C***CATEGORY NO. Z
C***KEYWORDS ERROR,XERROR PACKAGE
C***AUTHOR JONES, R. E., (SNLA)
C***PURPOSE SAVES AND RECALLS SOME GLOBAL VARIABLES NEEDED BY LIBRARY
C      ERROR HANDLING ROUTINES
C***DESCRIPTION
C      ABSTRACT
C      J4SAVE SAVES AND RECALLS SEVERAL GLOBAL VARIABLES NEEDED
C      BY THE LIBRARY ERROR HANDLING ROUTINES.
C
C      DESCRIPTION OF PARAMETERS
C      --INPUT--

```



```

C      IWHICH - INDEX OF ITEM DESIRED.
C      = 1 REFERS TO CURRENT ERROR NUMBER.
C      = 2 REFERS TO CURRENT ERROR CONTROL FLAG.
C      = 3 REFERS TO CURRENT UNIT NUMBER TO WHICH ERROR
C      MESSAGES ARE TO BE SENT. (0 MEANS USE STANDARD.)
C      = 4 REFERS TO THE MAXIMUM NUMBER OF TIMES ANY
C      MESSAGE IS TO BE PRINTED (AS SET BY XERMAX).
C      = 5 REFERS TO THE TOTAL NUMBER OF UNITS TO WHICH
C      EACH ERROR MESSAGE IS TO BE WRITTEN.
C      = 6 REFERS TO THE 2ND UNIT FOR ERROR MESSAGES
C      = 7 REFERS TO THE 3RD UNIT FOR ERROR MESSAGES
C      = 8 REFERS TO THE 4TH UNIT FOR ERROR MESSAGES
C      = 9 REFERS TO THE 5TH UNIT FOR ERROR MESSAGES
C      IVALUE - THE VALUE TO BE SET FOR THE IWHICH-TH PARAMETER,
C      IF ISET IS .TRUE. .
C      ISET   - IF ISET=.TRUE., THE IWHICH-TH PARAMETER WILL BE
C      GIVEN THE VALUE, IVALUE. IF ISET=.FALSE., THE
C      IWHICH-TH PARAMETER WILL BE UNCHANGED, AND IVALUE
C      IS A DUMMY PARAMETER.
C      --OUTPUT--
C      THE (OLD) VALUE OF THE IWHICH-TH PARAMETER WILL BE RETURNED
C      IN THE FUNCTION VALUE, J4SAVE.
C
C      WRITTEN BY RON JONES, WITH SLATEC COMMON MATH LIBRARY SUBCOMMITTEE
C      ADAPTED FROM BELL LABORATORIES PORT LIBRARY ERROR HANDLER
C      LATEST REVISION --- 23 MAY 1979
C***REFERENCES JONES R.E., *SLATEC COMMON MATHEMATICAL LIBRARY ERROR
C      HANDLING PACKAGE*, SAND78-1189, SANDIA LABORATORIES,
C      1978.
C***ROUTINES CALLED (NONE)
C***END PROLOGUE J4SAVE
C
C      LOGICAL ISET
C      INTEGER IPARAM(9)
C      SAVE IPARAM
C      DATA IPARAM(1),IPARAM(2),IPARAM(3),IPARAM(4)/0,1,0,10/
C      DATA IPARAM(5)/1/
C      DATA IPARAM(6),IPARAM(7),IPARAM(8),IPARAM(9)/0,0,0,0/
C***FIRST EXECUTABLE STATEMENT J4SAVE
C      J4SAVE = IPARAM(IWHICH)
C      IF (ISET) IPARAM(IWHICH) = IVALUE
C      RETURN
C      END
C*****
C SUBROUTINE MATINV IS USED TO INVERT A LOWER TRIANGULAR MATRIX
C
C      SUBROUTINE MATINV
C      PARAMETER (NVAR=150)
C      PARAMETER (LENT=125)
C
CBHL      CHARACTER*1 TITLE
CBHL
CBHL      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
C      1 ICORR,IDIST(NVAR),IRP
C      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
C      1 ICORR,IDIST(NVAR),IRP
CBHL      COMMON/PARAM/TITLE(LENT)
C
CBHL      COMMON/WORKC/Q((NVAR*(NVAR+1))/2),S((NVAR*(NVAR+1))/2)
C      LOC1(I,J)=J+(I*I-I)/2
C      DO 10 I=1,NV
C      10 Q(LOC1(I,I))=1.0/Q(LOC1(I,I))
C      K=NV
C      20 J=K-1
C      30 TEMP=0.0

```

```

        JPLUS=J+1
        DO 40 I=JPLUS,K
40    TEMP=TEMP+Q(LOC1(K,I))*Q(LOC1(I,J))
        Q(LOC1(K,J))=-TEMP*Q(LOC1(J,J))
        J=J-1
        IF (J.GE.1) GO TO 30
        K=K-1
        IF (K.GT.1) GO TO 20
        RETURN
    END
C*****
C SUBROUTINE MIX IMPLEMENTS THE SCHEME DESCRIBED IN IMAN AND
C CONOVER (1982) -SEE USERS GUIDE- FOR INDUCING A DESIRED
C CORRELATION STRUCTURE
C
    SUBROUTINE MIX
    PARAMETER (NMAX=1000)
    PARAMETER (NVAR=150)
    PARAMETER (LENT=125)
CBHL
    CHARACTER*1 TITLE
CBHL
CBHL
C    COMMON/PARAM/TITLE(LENT), ISEED,N,NV,IRS,ICM,NREP, IDATA, IHIST,
C    1    ICORR, IDIST(NVAR), IRP
    COMMON/PARAM/ ISEED,N,NV,IRS,ICM,NREP, IDATA, IHIST,
    1    ICORR, IDIST(NVAR), IRP
    COMMON/PARAMA/TITLE(LENT)
CBHL
    COMMON/CMATR/CORR((NVAR*(NVAR+1))/2), LCM(NVAR), NCM
    COMMON/SAMP/X(NMAX*NVAR)
    COMMON/RANK/XV(NMAX), RXV(NMAX), IWK(NMAX)
    COMMON/WORKX/XX(NMAX*NVAR)
    COMMON/WORKC/Q((NVAR*(NVAR+1))/2), S((NVAR*(NVAR+1))/2)
    LOC1(I,J)=J+(I*I-I)/2
    LOC(I,J)=(J-1)*N+I
C
C    IF ICM EQUALS 0 THEN A MIX OF THE GENERATED DATA IS
C    OBTAINED. OTHERWISE, A MIX OF RANDOM INTEGERS IS
C    GENERATED AND MULTIPLIED BY THE CORRELATION MATRIX.
C    THE RESULT IS RANKED AND THE RANKS ARE USED TO GENERATE
C    THE RANDOM SAMPLE.
C
C    FOR THE FULL RANK CASE(N.GT.NV)THE FOLLOWING PROCEDURE IS USED,
C    GENERATE A SAMPLE SUCH THAT NONE OF THE OFF-DIAGONAL ELEMENTS OF
C    THE CORRESPONDING CORRELATION MARTIX ARE EQUAL TO ONE IN ABSOLUTE
C    VALUE.CHOLESKY FACTORIZATION IS DONE NEXT.
C    FOR THE LESS THAN FULL RANK CASE(N.LE.NV) 25 SAMPLES ARE GENERATED
C    AND THAT SAMPLE WHOSE CORRESPONDING CORRELATION MATRIX HAS THE
C    SMALLEST MAXIMUM(IN ABSOLUTE VALUE)OFF-DIAGONAL ELEMENT IS THE
C    SAMPLE THAT IS CHOSEN. CHOLESKY FACTORIZATION IS SKIPPED IN
C    THIS CASE.
C
    NVN=NV*N
    CMXOLD=9999.0
    DN=N+1
    DO 1 I = 1,N
        XV(I) = FINVNOR(FLOAT(I)/DN)
    1 CONTINUE
    DO 50 IJK=1,25
10 CONTINUE
        DO 2 I=1,N
            2 IWK(I)=I
        DO 30 I=1,NV
            DO 20 J=2,N
                IR=N+2-J

```

```

        JJ=IR*RAN( ISEED)+1
        KK=IWK( IR)
        IWK( IR)=IWK( JJ)
20    IWK( JJ)=KK
        DO 30 J=1,N
        NDX=IWK( J)
        X( LOC( J,I) )=XV( NDX)
30    CONTINUE
        IF( NV.EQ.1.OR.IRP.EQ.1)GO TO 235
C
C      COMPUTE THE CORRELATION MATRIX ON THE VDW SCORES
C
        CALL CORCAL
        IF( N.LE.NV)GO TO 37
        DO 33 I=2,NV
        IMIN=I-1
        DO 33 J=1,IMIN
33    IF( ABS( CORR( LOC1( I,J) ) ) .GE.0.9999)GO TO 10
        GO TO 60
37    CONTINUE
        CMX=0.0
        DO 40 I=2,NV
        IMIN=I-1
        DO 40 J=1,IMIN
        ELEM=ABS( CORR( LOC1( I,J) ) )
        IF( ELEM.GT.CMX)CMX=ELEM
40    CONTINUE
        IF( CMX.GE.CMXOLD)GO TO 50
        CMXOLD=CMX
        REWIND 2
        WRITE( 2) ( X( I) , I=1,NVN)
50    CONTINUE
        REWIND 2
        READ( 2) ( X( I) , I=1,NVN)
C
C      X IS AN N X NV MATRIX OF VDW SCORES. IF THIS IS A FULL RANK CASE
C      ( N.GT.NV) THE RANK CORRELATION MATRIX OF X MUST BE FOUND AND
C      THE CHOLESKY FACTORIZATION PERFORMED. IF THIS IS NOT A FULL RANK
C      CASE ( N.LE.NV) THIS MATRIX OF VDW SCORES WILL BE USED TO MIX THE
C      INTERVALS OF THE RAW DATA MATRIX XX
C
        GO TO 235
60    CONTINUE
        NVX=( NV*( NV+1) )/2
        CALL CHLSKY
C
C      INVERT THE LOWER TRIANGULAR MATRIX Q RESULTING FROM THE
C      CHOLESKY FACTORIZATION
C
CBHL
C      CALL MATINV( NV)
        CALL MATINV
        IF( ICM.EQ.0.AND.( N.NE.3.OR.NV.NE.2) )GO TO 160
C
C      IF THE USER HAS SPECIFIED HIS OWN RANK CORRELATION STRUCTURE
C      THIS MATRIX MULTIPLICATION LOOP IS NEEDED
C
        REWIND 3
        READ( 3) CORR
        DO 150 K=1,NV
        KX=K
        DO 140 IT=1,KX
        XV( IT)=0.0
        ITX=IT
        DO 140 IW=ITX,KX
140    XV( IT)=CORR( LOC1( K,IW) ) *Q( LOC1( IW,IT) )+XV( IT)

```

```

      DO 150 IW=1,NV
150  S(LOC1(K,IW))=XV(IW)
      GO TO 180
C
C      THIS ASSUMES THE DESIRED CORRELATION STRUCTURE IS THE IDENTITY
C      MATRIX, I.E. CLOSE TO ORTHOGONAL
C
160  DO 170 I=1,NVX
170  S(I)=Q(I)
180  CONTINUE
      DO 210 K=1,N
      DO 190 IT=1,NV
      XV(IT)=0.0
      ITX=IT
      DO 190 IW=1,ITX
190  XV(IT)=X(LOC(K,IW))*S(LOC1(IT,IW))+XV(IT)
      DO 200 IW=1,NV
200  X(LOC(K,IW))=XV(IW)
210  CONTINUE
C
C      SINCE X NO LONGER CONTAINS INTEGERS WE MUST RANK EACH COLUMN
C      IN ORDER TO GET A NEW MATRIX TO USE FOR MIXING THE INTERVALS
C      OF THE RAW DATA MATRIX XX
C
235  CONTINUE
      REWIND 4
      READ(4)XX
      DO 230 J=1,NV
      DO 220 I=1,N
220  XV(I)=X(LOC(I,J))
      CALL RANKER
      DO 230 I=1,N
230  X(LOC(I,J))=RXV(I)
      DO 240 J=1,NV
      DO 240 I=1,N
      K=INT(X(LOC(I,J))+0.01)
240  X(LOC(I,J))=XX(LOC(K,J))
      RETURN
      END
C*****
C SUBROUTINE NORMAL GENERATES NORMAL AND LOGNORMAL DISTRIBUTIONS
C
      SUBROUTINE NORMAL(J,IDT)
      PARAMETER (NMAX=1000)
      PARAMETER (NVAR=150)
      PARAMETER (LENT=125)
CBHL
      CHARACTER*1 TITLE
CBHL
CBHL
C      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
C      1      ICORR,IDIST(NVAR),IRP
C      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
C      1      ICORR,IDIST(NVAR),IRP
C      COMMON/PARAM/TITLE(LENT)
CBHL
      COMMON/SAMP/X(NVAR*NMAX)
      LOC(I,J)=(J-1)*N+I
      PROBINC=1./FLOAT(N)
      IF(IRS.EQ.1)PROBINC=1.0
C
C      A IS ASSUMED TO BE THE LOWER .001 QUANTILE OF THE NORMAL
C      DISTRIBUTION.
C      B IS ASSUMED TO BE THE UPPER .001 QUANTILE OF THE NORMAL
C      DISTRIBUTION.
C

```

```

      READ(8)A,B
      IF (IDT.EQ.3) THEN
        A=LOG(A)
        B=LOG(B)
      ENDIF
      PMU=(A+B)/2.
      SIG=(B-PMU)/3.0902323062
      STRTPT=0.
      DO 10 I=1,N
        R=PROBINC*RAN( ISEED)+STRTPT
        IF(R.GE.0.999)R=0.999
        IF(R.LE.0.001)R=0.001
        X(LOC(I,J))=FINVNOR(R)*SIG+PMU
        IF (IDT.EQ.3)X(LOC(I,J))=EXP(X(LOC(I,J)))
        IF (IRS.EQ.0)STRTPT=STRTPT+PROBINC
10    CONTINUE
      RETURN
      END
C*****
C SUBROUTINE ONECHK IS AN ERROR CHECKING ROUTINE USED IN
C GENERATING A BETA DISTRIBUTION
C
      SUBROUTINE ONECHK(NCHARS,NARRAY)
C
C   ABSTRACT
C     ONECHK IS A COMPANION ROUTINE OF ERRCHK. IT IS CALLED
C     JUST LIKE ERRCHK, AND MESSAGES FROM IT MAY BE SUPPRESSED
C     BY AN APPROPRIATE CALL TO ERXSET. IT DIFFERS FROM ERRCHK
C     IN THAT EACH CALL TO ONECHK WILL PRODUCE NO MORE THAN ONE
C     PRINTED MESSAGE, REGARDLESS OF HOW MANY TIMES THAT CALL IS
C     EXECUTED, AND ONECHK NEVER TERMINATES EXECUTION.
C     ITS PURPOSE IS TO PROVIDE ONE-TIME-ONLY INFORMATIVE
C     DIAGNOSTICS.
C
C   DESCRIPTION OF ARGUMENTS
C     NCHARS - NUMBER OF CHARACTERS IN THE MESSAGE.
C              IF NEGATED, THE MESSAGE WILL BE PRINTED (ONCE) EVEN
C              IF NFATAL HAS BEEN SET TO 0 (SEE ERXSET).
C     NARRAY - SAME AS IN ERRCHK
C
C
C     ONECHK USES SUBROUTINES ERRGET, ERRPRT, ERXSET, ERSTGT
C
      DIMENSION NARRAY(16)
      SAVE NFLAG
      DATA NFLAG/4H.$,*/
      IF (NARRAY(1).EQ.NFLAG) RETURN
      CALL ERRGET(NF,NT)
      IF ((NF.EQ.0).AND.(NCHARS.GT.0)) RETURN
      CALL ERRPRT (59,59)THE FOLLOWING INFORMATIVE DIAGNOSTIC WILL APPEA
1R ONLY ONCE.)
      CALL ERRPRT(IABS(NCHARS),NARRAY)
      IF (NF.GT.0) NF = NF-1
      CALL ERXSET(NF,NT)
      NARRAY(1) = NFLAG
      END
C*****
C SUBROUTINE OUTCRD PROCESSES THE OUTPUT PARAMETER OPTIONS
C
      SUBROUTINE OUTCRD(CARD)
      PARAMETER (NVAR=150)
      PARAMETER (LENC=80)
      PARAMETER (LENT=125)
CBHL  CHARACTER*1 TITLE

```

```

CBHL
CBHL
C      COMMON/PARAM/TITLE(LENT), ISEED,N,NV, IRS, ICM,NREP, IDATA, IHIST,
C      1      ICORR, IDIST(NVAR), IRP
C      COMMON/PARAM/ISEED,N,NV, IRS, ICM,NREP, IDATA, IHIST,
C      1      ICORR, IDIST(NVAR), IRP
C      COMMON/PARAM/TITLE(LENT)
CBHL
      CHARACTER CARD*(LENC), PDATA*4, PHIST*4, PCORR*4, BLANK
      PARAMETER (PDATA='DATA', PHIST='HIST', PCORR='CORR', BLANK=' ')
      IC=7
10  CONTINUE
      IC=IC+1
      IF(IC.GT.LENC)GO TO 20
      IF(CARD(IC:IC).EQ.BLANK)GO TO 10
      IE=IC+3
      IF(CARD(IC:IE).EQ.PDATA)THEN
          IDATA=1
          IC=IE+1
          GO TO 10
      ELSE IF(CARD(IC:IE).EQ.PHIST)THEN
          IHIST=1
          IC=IE+1
          GO TO 10
      ELSE IF(CARD(IC:IE).EQ.PCORR)THEN
          ICORR=1
          IC=IE+1
          GO TO 10
      ELSE
          WRITE(6,9001)CARD
          STOP 'OUTCRD'
      ENDIF
20  CONTINUE
      RETURN
9001 FORMAT(1H1,5X,'THE FOLLOWING OUTPUT OPTION CARD REQUESTED ',
1          'AN UNDEFINED OUTPUT OPTION',/,6X,'PLEASE CHECK THE ',
2          'USER MANUAL FOR THE CORRECT OUTPUT OPTION CARD ',
3          'SYNTAX',/,3X,'***',A,'***')
      END
C*****
C SUBROUTINE OUTDAT OUTPUTS THE SAMPLE AND ITS RANKS
C
      SUBROUTINE OUTDAT(IRK)
      PARAMETER (NMAX=1000)
      PARAMETER (NVAR=150)
      PARAMETER (LENT=125)
CBHL
      CHARACTER*1 TITLE
CBHL
      COMMON/SAMP/X(NVAR*NMAX)
CBHL
C      COMMON/PARAM/TITLE(LENT), ISEED,N,NV, IRS, ICM,NREP, IDATA, IHIST,
C      1      ICORR, IDIST(NVAR), IRP
C      COMMON/PARAM/ISEED,N,NV, IRS, ICM,NREP, IDATA, IHIST,
C      1      ICORR, IDIST(NVAR), IRP
C      COMMON/PARAM/TITLE(LENT)
CBHL
      CHARACTER LAB1*2, LAB2
      PARAMETER (LAB1='X(', LAB2=')')
      LOC(I,J)=(J-1)*N+I
      K=1
      I1=1
      I2=12
      ITEMP=N/12
      IF (ITEMP*12.NE.NV) ITEMP=ITEMP+1
300 IF (K.EQ.ITEMP) GO TO 350

```

```

IF (K.GT.1) GO TO 320
WRITE(6,820)TITLE
IF(IRS.EQ.0)THEN
  IF(IRK.EQ.0)THEN
    WRITE(6,810)(LAB1,I,LAB2,I=I1,I2)
  ELSE
    WRITE(6,850)(LAB1,I,LAB2,I=I1,I2)
  ENDIF
ELSE
  IF(IRK.EQ.0)THEN
    WRITE(6,815)(LAB1,I,LAB2,I=I1,I2)
  ELSE
    WRITE(6,855)(LAB1,I,LAB2,I=I1,I2)
  ENDIF
ENDIF
DO 310 J=1,N
  IF(IRK.EQ.0)THEN
    WRITE(6,840)J,(X(LOC(J,I)),I=I1,I2)
  ELSE
    WRITE(6,860)J,(X(LOC(J,I)),I=I1,I2)
  ENDIF
310 CONTINUE
GO TO 340
320 WRITE(6,820)TITLE
IF(IRS.EQ.0)THEN
  IF(IRK.EQ.0)THEN
    WRITE(6,830)(LAB1,I,LAB2,I=I1,I2)
  ELSE
    WRITE(6,870)(LAB1,I,LAB2,I=I1,I2)
  ENDIF
ELSE
  IF(IRK.EQ.0)THEN
    WRITE(6,835)(LAB1,I,LAB2,I=I1,I2)
  ELSE
    WRITE(6,875)(LAB1,I,LAB2,I=I1,I2)
  ENDIF
ENDIF
DO 330 J=1,N
  IF(IRK.EQ.0)THEN
    WRITE(6,840)J,(X(LOC(J,I)),I=I1,I2)
  ELSE
    WRITE(6,860)J,(X(LOC(J,I)),I=I1,I2)
  ENDIF
330 CONTINUE
340 K=K+1
I1=I1+12
I2=I2+12
GO TO 300
350 IF (K.GT.1) GO TO 370
WRITE(6,820)TITLE
IF(IRS.EQ.0)THEN
  IF(IRK.EQ.0)THEN
    WRITE(6,810)(LAB1,I,LAB2,I=I1,NV)
  ELSE
    WRITE(6,850)(LAB1,I,LAB2,I=I1,NV)
  ENDIF
ELSE
  IF(IRK.EQ.0)THEN
    WRITE(6,815)(LAB1,I,LAB2,I=I1,NV)
  ELSE
    WRITE(6,855)(LAB1,I,LAB2,I=I1,NV)
  ENDIF
ENDIF
DO 360 J=1,N
  IF(IRK.EQ.0)THEN
    WRITE(6,840)J,(X(LOC(J,I)),I=I1,NV)

```

```

        ELSE
            WRITE(6,860)J,(X(LOC(J,I)),I=I1,NV)
        ENDIF
360 CONTINUE
GO TO 390
370 WRITE(6,820)TITLE
    IF(IRS.EQ.0)THEN
        IF(IRK.EQ.0)THEN
            WRITE(6,830)(LAB1,I,LAB2,I=I1,NV)
        ELSE
            WRITE(6,870)(LAB1,I,LAB2,I=I1,NV)
        ENDIF
    ELSE
        IF(IRK.EQ.0)THEN
            WRITE(6,835)(LAB1,I,LAB2,I=I1,NV)
        ELSE
            WRITE(6,875)(LAB1,I,LAB2,I=I1,NV)
        ENDIF
    ENDIF
DO 380 J=1,N
    IF(IRK.EQ.0)THEN
        WRITE(6,840)J,(X(LOC(J,I)),I=I1,NV)
    ELSE
        WRITE(6,860)J,(X(LOC(J,I)),I=I1,NV)
    ENDIF
380 CONTINUE
390 CONTINUE
RETURN
810 FORMAT('0LATIN HYPERCUBE SAMPLE INPUT VECTORS',//,' RUN NO.',
1      1X,A,I1,A,8(6X,A,I1,A),3(5X,A,I2,A))
815 FORMAT('0RANDOM SAMPLE INPUT VECTORS',//,' RUN NO.',
1      1X,A,I1,A,8(6X,A,I1,A),3(5X,A,I2,A))
820 FORMAT(1H1,125A1)
830 FORMAT('0LATIN HYPERCUBE SAMPLE INPUT VECTORS',//,' RUN NO.',
1      2X,A,I2,A,11(5X,A,I2,A))
835 FORMAT('0RANDOM SAMPLE INPUT VECTORS',//,' RUN NO.',
1      2X,A,I2,A,11(5X,A,I2,A))
840 FORMAT(1H0,I5,12(1PG10.3))
850 FORMAT('0RANKS OF LATIN HYPERCUBE SAMPLE INPUT VECTORS',//,
1      ' RUN NO.',4X,A,I1,A,8(6X,A,I1,A),3(5X,A,I2,A))
855 FORMAT('0RANKS OF RANDOM SAMPLE INPUT VECTORS',//,' RUN NO.',
1      4X,A,I1,A,8(6X,A,I1,A),3(5X,A,I2,A))
860 FORMAT(1H0,I5,12F10.0)
870 FORMAT('0RANKS OF LATIN HYPERCUBE SAMPLE INPUT VECTORS',//,
1      ' RUN NO.',4X,A,I2,A,11(5X,A,I2,A))
875 FORMAT('0RANKS OF RANDOM SAMPLE INPUT VECTORS',//,' RUN NO.',
1      4X,A,I2,A,11(5X,A,I2,A))
END
C*****
C SUBROUTINE PMTRX PRINTS OUT A CORRELATION MATRIX
C
    SUBROUTINE PMTRX(NC,IFLAG)
    PARAMETER (NVAR=150)
    PARAMETER (NROW=28)
    PARAMETER (NCOL=15)
    PARAMETER (LENT=125)
CBHL
    CHARACTER*1 TITLE
CBHL
CBHL
C    COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
C    1      ICORR,IDIST(NVAR),IRP
    COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
    1      ICORR,IDIST(NVAR),IRP
    COMMON/PARAM/TITLE(LENT)
CBHL

```



```

COMMON/CMATR/CORR((NVAR*(NVAR+1))/2),LCM(NVAR),NCM
LOC1(I,J)=J+(I*I-I)/2
I=1
I1=1
I2=NROW
NPROW=NC/NROW
IF (NPROW*NROW.NE.NC) NPROW=NPROW+1
NPCOL=NC/NCOL
IF (NPCOL*NCOL.NE.NC) NPCOL=NPCOL+1
DO 100 NR=1,NPROW
J1=1
J2=NCOL
DO 90 IC=1,NPCOL
IF (J1.GT.I2) GO TO 90
IF (IFLAG.EQ.2) GO TO 10
IF (IFLAG.EQ.3) GO TO 20
IF (IFLAG.EQ.4) GO TO 20
WRITE(6,180)TITLE
IF (IRS.EQ.0)WRITE(6,110)I
IF (IRS.NE.0)WRITE(6,115)I
GO TO 30
10 WRITE(6,180)TITLE
IF (IRS.EQ.0)WRITE(6,120)I
IF (IRS.NE.0)WRITE(6,125)I
GO TO 30
20 WRITE(6,180)TITLE
IF (IFLAG.EQ.3)WRITE(6,130)I
IF (IFLAG.EQ.4)WRITE(6,190)I
30 IF (NC.LT.I2) I2=NC
IF (NC.LT.J2) J2=NC
DO 60 K=I1,I2
K1=K
IF (K.GE.J1) GO TO 40
WRITE(6,140)
GO TO 60
40 IF (K.GT.J2) GO TO 50
WRITE(6,150)LCM(K),(CORR(LOC1(K,J)),J=J1,K)
GO TO 60
50 WRITE(6,150)LCM(K),(CORR(LOC1(K,J)),J=J1,J2)
60 CONTINUE
IF (K1.LT.J2) GO TO 70
WRITE(6,160)(LCM(II),II=J1,J2)
WRITE(6,170)
GO TO 80
70 WRITE(6,160)(LCM(II),II=J1,K1)
WRITE(6,170)
80 CONTINUE
J1=J1+NCOL
J2=J2+NCOL
I=I+1
90 CONTINUE
I1=I1+NROW
I2=I2+NROW
100 CONTINUE
RETURN
110 FORMAT('0CORRELATIONS AMONG INPUT VARIABLES CREATED BY THE LATIN',
1 ' HYPERCUBE SAMPLE FOR RAW DATA',30X,'PAGE ',I3)
115 FORMAT('0CORRELATIONS AMONG INPUT VARIABLES CREATED BY THE ',
1 ' RANDOM SAMPLE FOR RAW DATA',40X,'PAGE ',I3)
120 FORMAT('0CORRELATIONS AMONG INPUT VARIABLES CREATED BY THE LATIN',
1 ' HYPERCUBE SAMPLE FOR RANK DATA',30X,'PAGE ',I3)
125 FORMAT('0CORRELATIONS AMONG INPUT VARIABLES CREATED BY THE ',
1 ' RANDOM SAMPLE FOR RANK DATA',40X,'PAGE ',I3)
130 FORMAT('0INPUT RANK CORRELATION MATRIX',95X,'PAGE ',I3)
140 FORMAT('0')
150 FORMAT('0',I5,15(F8.4))

```

```

160 FORMAT('0',5X,15I8)
170 FORMAT('0VARIABLES')
180 FORMAT('1',125A1)
190 FORMAT('0ADJUSTED RANK CORRELATION MATRIX',92X,'PAGE ',I3)
END
C*****
C SUBROUTINE POSDEF IS USED IN THE POSITIVE DEFINITE CHECK
C OF THE CORRELATION MATRIX
C
      SUBROUTINE POSDEF(ITEST)
      PARAMETER (NVAR=150)
      COMMON/CMATR/CORR((NVAR*(NVAR+1))/2),LCM(NVAR),NCM
      COMMON/PDMAT/Z(NVAR,NVAR),D(NVAR)
      COMMON/WORK/WK((NVAR*(NVAR+1))/2)
      SAVE NN,EIG
C
      NN = THE DIMENSIONS OF Z, I.E. Z(NN,NN).
      DATA NN/NVAR/
C
C
C      EIG = THE VALUE THAT THE NEGATIVE EIGENVALUES ARE SET EQUAL TO.
      DATA EIG/0.001/
C
C
C      M = MAXIMUM NUMBER OF ITERATIONS ALLOWED.
      M=20
C
      NP=NCM
      NKX=(NP*(NP+1))/2
      ITEST=0
      ICONV=0
100 CONTINUE
      ITEST=ITEST+1
      IF(ITEST.GT.M)THEN
        WRITE(6,1000)
        STOP 'POSDEF'
      ENDIF
      REWIND 3
      WRITE(3)CORR
      CALL SSPEV(CORR,NP,D,Z,NN,WK,1,INFO)
      CALL FINDIT(NP,ITEST,NN,EIG,ICONV)
      IF(ICONV.EQ.0)GO TO 100
      REWIND 3
      READ(3)CORR
      RETURN
1000 FORMAT(1H1,'THE INPUT RANK CORRELATION MATRIX IS NOT POSITIVE ',
1      'DEFINITE.',/, ' AN ITERATIVE PROCEDURE HAS FAILED TO ',
2      'PRODUCE A POSITIVE DEFINITE MATRIX AFTER 20 ITERATIONS.',
3      '/', ' THEREFORE, THE PROGRAM HAS BEEN TERMINATED.',/, ' THE',
4      ' USER NEEDS TO REEVALUATE THE RELATIONSHIP OF THE ',
5      'CORRELATED VARIABLES IN THE MATRIX.')
END
C*****
C FUNCTION PYTHAG IS USED IN THE POSTIVE DEFINITE CHECK OF THE
C CORRELATION MATRIX
C
      REAL FUNCTION PYTHAG(A,B)
C***BEGIN PROLOGUE PYTHAG
C***REFER TO EISDOC
C      FINDS SQRT(A**2+B**2) WITHOUT OVERFLOW OR DESTRUCTIVE UNDERFLOW
C***ROUTINES CALLED (NONE)
C***END PROLOGUE PYTHAG
      REAL A,B
C
      REAL P,Q,R,S,T
C***FIRST EXECUTABLE STATEMENT PYTHAG

```

```

        P = MAX(ABS(A),ABS(B))
        Q = MIN(ABS(A),ABS(B))
        IF (Q .EQ. 0.0E0) GO TO 20
10    CONTINUE
        R = (Q/P)**2
        T = 4.0E0 + R
        IF (T .EQ. 4.0E0) GO TO 20
        S = R/T
        P = P + 2.0E0*P*S
        Q = Q*S
        GO TO 10
20    PYTHAG = P
        RETURN
        END
C*****
C FUNCTION R1MACH IS USED TO SET FLOATING POINT MACHINE
C DEPENDENT CONSTANTS
C
        REAL FUNCTION R1MACH(I)
C***BEGIN PROLOGUE  R1MACH
C***DATE WRITTEN   790101   (YYMMDD)
C***REVISION DATE  801001   (YYMMDD)
C***CATEGORY NO.  R1
C***KEYWORDS  MACHINE CONSTANTS
C***AUTHOR  FOX, P. A., (BELL LABS)
C           HALL, A. D., (BELL LABS)
C           SCHRYER, N. L., (BELL LABS)
C***PURPOSE  Returns single precision machine dependent constants
C***DESCRIPTION
C           R1MACH can be used to obtain machine-dependent parameters
C           for the local machine environment.  It is a function
C           subroutine with one (input) argument, and can be called
C           as follows, for example
C
C           A = R1MACH(I)
C
C           where I=1,...,5.  The (output) value of A above is
C           determined by the (input) value of I.  The results for
C           various values of I are discussed below.
C
C   Single-Precision Machine Constants
C   R1MACH(1) = B**(EMIN-1), the smallest positive magnitude.
C   R1MACH(2) = B**EMAX*(1 - B**(-T)), the largest magnitude.
C   R1MACH(3) = B**(-T), the smallest relative spacing.
C   R1MACH(4) = B**(1-T), the largest relative spacing.
C   R1MACH(5) = LOG10(B)
C***REFERENCES  FOX, P.A., HALL, A.D., SCHRYER, N.L., *FRAMEWORK FOR
C               A PORTABLE LIBRARY*, ACM TRANSACTIONS ON MATHE-
C               MATICAL SOFTWARE, VOL. 4, NO. 2, JUNE 1978,
C               PP. 177-188.
C***ROUTINES CALLED  XERROR
C***END PROLOGUE  R1MACH
C
        INTEGER SMALL(2)
        INTEGER LARGE(2)
        INTEGER RIGHT(2)
        INTEGER DIVER(2)
        INTEGER LOG10(2)
C
        REAL RMACH(5)
        SAVE RMACH,SMALL,LARGE,RIGHT,DIVER,LOG10
C
C   EQUIVALENCE (RMACH(1),SMALL(1))
C   EQUIVALENCE (RMACH(2),LARGE(1))
C   EQUIVALENCE (RMACH(3),RIGHT(1))
C   EQUIVALENCE (RMACH(4),DIVER(1))

```

```

C      EQUIVALENCE (RMACH(5),LOG10(1))
C
C
C      MACHINE CONSTANTS FOR THE BURROUGHS 1700 SYSTEM.
C
C      DATA RMACH(1) / Z400800000 /
C      DATA RMACH(2) / Z5FFFFFFFF /
C      DATA RMACH(3) / Z4E9800000 /
C      DATA RMACH(4) / Z4EA800000 /
C      DATA RMACH(5) / Z500E730E8 /
C
C      MACHINE CONSTANTS FOR THE BURROUGHS 5700/6700/7700 SYSTEMS.
C
C      DATA RMACH(1) / 017710000000000000 /
C      DATA RMACH(2) / 007777777777777777 /
C      DATA RMACH(3) / 013110000000000000 /
C      DATA RMACH(4) / 013010000000000000 /
C      DATA RMACH(5) / 01157163034761675 /
C
C      MACHINE CONSTANTS FOR THE CDC 6000/7000 SERIES.
C
C      DATA RMACH(1) / 00564000000000000000B /
C      DATA RMACH(2) / 3776777777777777777B /
C      DATA RMACH(3) / 16414000000000000000B /
C      DATA RMACH(4) / 16424000000000000000B /
C      DATA RMACH(5) / 17164642023241175720B /
C
C      MACHINE CONSTANTS FOR THE CRAY 1
C
C      DATA RMACH(1) / 200034000000000000000B /
C      DATA RMACH(2) / 5777677777777777777B /
C      DATA RMACH(3) / 377224000000000000000B /
C      DATA RMACH(4) / 377234000000000000000B /
C      DATA RMACH(5) / 377774642023241175720B /
C
C      MACHINE CONSTANTS FOR THE DATA GENERAL ECLIPSE S/200
C
C      NOTE - IT MAY BE APPROPRIATE TO INCLUDE THE FOLLOWING CARD -
C      STATIC RMACH(5)
C
C      DATA SMALL/20K,0/,LARGE/77777K,177777K/
C      DATA RIGHT/35420K,0/,DIVER/36020K,0/
C      DATA LOG10/40423K,42023K/
C
C      MACHINE CONSTANTS FOR THE HARRIS 220
C
C      DATA SMALL(1),SMALL(2) / '20000000, '00000201 /
C      DATA LARGE(1),LARGE(2) / '37777777, '00000177 /
C      DATA RIGHT(1),RIGHT(2) / '20000000, '00000352 /
C      DATA DIVER(1),DIVER(2) / '20000000, '00000353 /
C      DATA LOG10(1),LOG10(2) / '23210115, '00000377 /
C
C      MACHINE CONSTANTS FOR THE HONEYWELL 600/6000 SERIES.
C
C      DATA RMACH(1) / 0402400000000 /
C      DATA RMACH(2) / 0376777777777 /
C      DATA RMACH(3) / 0714400000000 /
C      DATA RMACH(4) / 0716400000000 /
C      DATA RMACH(5) / 0776464202324 /
C
C      MACHINE CONSTANTS FOR THE HP 2100
C
C      3 WORD DOUBLE PRECISION WITH FTN4
C
C      DATA SMALL(1), SMALL(2) / 40000B,          1 /
C      DATA LARGE(1), LARGE(2) / 77777B, 177776B /

```

```

C      DATA RIGHT(1), RIGHT(2) / 40000B,    325B /
C      DATA DIVER(1), DIVER(2) / 40000B,    327B /
C      DATA LOG10(1), LOG10(2) / 46420B,  46777B /
C
C      MACHINE CONSTANTS FOR THE HP 2100
C      4 WORD DOUBLE PRECISION WITH FTN4
C
C      DATA SMALL(1), SMALL(2) / 40000B,      1 /
C      DATA LARGE(1), LARGE(2) / 77777B, 177776B /
C      DATA RIGHT(1), RIGHT(2) / 40000B,    325B /
C      DATA DIVER(1), DIVER(2) / 40000B,    327B /
C      DATA LOG10(1), LOG10(2) / 46420B,  46777B /
C
C      MACHINE CONSTANTS FOR THE IBM 360/370 SERIES,
C      THE XEROX SIGMA 5/7/9, THE SEL SYSTEMS 85/86 AND
C      THE PERKIN ELMER (INTERDATA) 7/32.
C
C      DATA RMACH(1) / Z00100000 /
C      DATA RMACH(2) / Z7FFFFFFF /
C      DATA RMACH(3) / Z3B100000 /
C      DATA RMACH(4) / Z3C100000 /
C      DATA RMACH(5) / Z41134413 /
C
C      MACHINE CONSTANTS FOR THE PDP-10 (KA OR KI PROCESSOR).
C
C      DATA RMACH(1) / "000400000000 /
C      DATA RMACH(2) / "377777777777 /
C      DATA RMACH(3) / "146400000000 /
C      DATA RMACH(4) / "147400000000 /
C      DATA RMACH(5) / "177464202324 /
C
C      MACHINE CONSTANTS FOR PDP-11 FORTRAN SUPPORTING
C      32-BIT INTEGERS (EXPRESSED IN INTEGER AND OCTAL).
C
C      DATA SMALL(1) /      8388608 /
C      DATA LARGE(1) / 2147483647 /
C      DATA RIGHT(1) /  880803840 /
C      DATA DIVER(1) /  889192448 /
C      DATA LOG10(1) / 1067065499 /
C
C      DATA RMACH(1) / 000040000000 /
C      DATA RMACH(2) / 017777777777 /
C      DATA RMACH(3) / 006440000000 /
C      DATA RMACH(4) / 006500000000 /
C      DATA RMACH(5) / 007746420233 /
C
C      MACHINE CONSTANTS FOR PDP-11 FORTRAN SUPPORTING
C      16-BIT INTEGERS (EXPRESSED IN INTEGER AND OCTAL).
C
C      DATA SMALL(1),SMALL(2) /   128,      0 /
C      DATA LARGE(1),LARGE(2) / 32767,    -1 /
C      DATA RIGHT(1),RIGHT(2) / 13440,      0 /
C      DATA DIVER(1),DIVER(2) / 13568,      0 /
C      DATA LOG10(1),LOG10(2) / 16282,  8347 /
C
C      DATA SMALL(1),SMALL(2) / 0000200, 0000000 /
C      DATA LARGE(1),LARGE(2) / 0077777, 0177777 /
C      DATA RIGHT(1),RIGHT(2) / 0032200, 0000000 /
C      DATA DIVER(1),DIVER(2) / 0032400, 0000000 /
C      DATA LOG10(1),LOG10(2) / 0037632, 0020233 /
C
C      MACHINE CONSTANTS FOR THE UNIVAC 1100 SERIES.
C
C      DATA RMACH(1) / 0000400000000 /
C      DATA RMACH(2) / 0377777777777 /
C      DATA RMACH(3) / 0146400000000 /

```

```

C      DATA RMACH(4) / 0147400000000 /
C      DATA RMACH(5) / 0177464202324 /
C
C      MACHINE CONSTANTS FOR THE VAX 11/780
C      (EXPRESSED IN INTEGER AND HEXADECIMAL)
C      ***THE HEX FORMAT BELOW MAY NOT BE SUITABLE FOR UNIX SYSTEMS***
C      *** THE INTEGER FORMAT SHOULD BE OK FOR UNIX SYSTEMS***
C
C      DATA SMALL(1) /      128 /
C      DATA LARGE(1) /    -32769 /
C      DATA RIGHT(1) /     13440 /
C      DATA DIVER(1) /     13568 /
C      DATA LOG10(1) / 547045274 /
C
C      DATA SMALL(1) / Z00000080 /
C      DATA LARGE(1) / ZFFFF7FFF /
C      DATA RIGHT(1) / Z00003480 /
C      DATA DIVER(1) / Z00003500 /
C      DATA LOG10(1) / Z209B3F9A /
C
C***FIRST EXECUTABLE STATEMENT  R1MACH
C
C      IF (I .LT. 1 .OR. I .GT. 5)
C      1 CALL XERROR (25HR1MACH -- I OUT OF BOUNDS,25,1,2)
C
C      RMACH(1)=SMALL(1)
C      RMACH(2)=LARGE(1)
C      RMACH(3)=RIGHT(1)
C      RMACH(4)=DIVER(1)
C      RMACH(5)=LOG10(1)
C      R1MACH = RMACH(I)
C      RETURN
C
C      END
C*****
C SUBROUTINE RANKER IS USED TO FIND THE RANKS OF A VECTOR OF DATA
C
C      SUBROUTINE RANKER
C      PARAMETER (NMAX=1000)
C      PARAMETER (NVAR=150)
C      PARAMETER (LENT=125)
CBHL
C      CHARACTER*1 TITLE
CBHL
CBHL
C      COMMON/PARAM/TITLE(LENT), ISEED,N,NV,IRS,ICM,NREP, IDATA, IHIST,
C      1 ICORR, IDIST(NVAR), IRP
C      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP, IDATA, IHIST,
C      1 ICORR, IDIST(NVAR), IRP
C      COMMON/PARAMA/TITLE(LENT)
CBHL
C      COMMON/RANK/XV(NMAX),RXV(NMAX),IWK(NMAX)
C      DO 10 I=1,N
C      10 RXV(I)=FLOAT(I)
C      CALL HPSRT
C      DO 20 I=1,N
C      20 IWK(I)=IFIX(RXV(I))
C      20 RXV(I)=FLOAT(I)
C
C      FIND TIES
C      I=0
C      30 I=I+1
C      IF (I.GE.N) GO TO 80
C      IF (XV(I).NE.XV(I+1)) GO TO 30
C
C      COUNT TIES

```

```

        NTIES=2
40    II=I+NTIES
        IF (II.GT.N) GO TO 50
        IF (XV(I).NE.XV(II)) GO TO 50
        NTIES=NTIES+1
        GO TO 40
C
C    AVERAGE TIED RANKS
50    AVG=0.0
        DO 60 J=1,NTIES
60    AVG=AVG+RXV(I+J-1)
        AVG=AVG/FLOAT(NTIES)
        I=I-1
        DO 70 J=1,NTIES
        I=I+1
70    RXV(I)=AVG
        GO TO 30
80    CONTINUE
C
C    REORDER
        I=1
90    K=IWK(I)
        IF (K.NE.I) GO TO 100
        I=I+1
        IF (I.LT.N) GO TO 90
        RETURN
100   XHOLD=XV(I)
        RHOLD=RXV(I)
        XV(I)=XV(K)
        RXV(I)=RXV(K)
        XV(K)=XHOLD
        RXV(K)=RHOLD
        IWK(I)=IWK(K)
        IWK(K)=K
        GO TO 90
        END
C*****
C SUBROUTINE RDPAR PROCESSES THE PARAMETER STATEMENTS, DEFINES
C VARIABLES IN COMMON /PARAM/ AND STORES DISTRIBUTION INFORMATION
C ON UNITS 7, 8 AND 9
C
        SUBROUTINE RDPAR
        PARAMETER (NMAX=1000)
        PARAMETER (NVAR=150)
        PARAMETER (NCVAR=150)
        PARAMETER (NINTMX=50)
        PARAMETER (LENC=80)
        PARAMETER (LENT=125)
        PARAMETER (LENTC=11)
        CHARACTER CARD*(LENC),TMPCRD*(LENTC),CDUM1,CDUM2
        CHARACTER PTITLE*6,PRSAMP*14,PRSEED*12,PNOBS*5,LOBS*12,PNREP*6,
1          PC,PCMINP*18,LCMINP*29,POUT*7,PBETA*5,PNOR*7,PLNOR*10,
2          PUNI*8,PLUNI*11,PUNIS*9,PLUNIS*12,PUSDST*18,BLANK,
3          PERIOD,MINUS,PTRIAG*11,PIRP*15
CBHL
        CHARACTER*1 TITLE
CBHL
        COMMON/STAR/NSUBOB(NINTMX),SUBINT(NINTMX+1),NINT
        COMMON/UICORR/ICVAR(NCVAR),JCVAR(NCVAR),CVAR(NCVAR),NCV
CBHL
C    COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
C    1          ICORR,IDIST(NVAR),IRP
        COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
1          ICORR,IDIST(NVAR),IRP
        COMMON/PARAMA/TITLE(LENT)
        COMMON/MCLHS/XMC(NVAR)

```

```

CBHL
  PARAMETER (PTITLE='TITLE ',PRSAMP='RANDOM SAMPLE ',PC='C',
1      PRSEED='RANDOM SEED ',PNOBS='NOBS ',
2      PNREP='NREPS ',PCMINP='CORRELATION MATRIX',
3      LCMINP='PAIRS OF CORRELATED VARIABLES',
4      PBETA='BETA ',PNOR='NORMAL ',PLNOR='LOGNORMAL ',
5      PUNI='UNIFORM ',PLUNI='LOGUNIFORM ',
6      PLUNIS='LOGUNIFORM* ',PUSDST='USER DISTRIBUTION ',
7      BLANK=' ',PERIOD='.',MINUS='-',PTRIAG='TRIANGULAR ',
8      LOBS='OBSERVATIONS',DUM1=9999.0,DUM2=9999.0,
9      POUT='OUTPUT ',PUNIS='UNIFORM* ',
A      PIRP='RANDOM PAIRING ')
  CHARACTER*19 IRSAMP
  SAVE IRSAMP
  DATA IRSAMP/'***RANDOM SAMPLE***'/
  IZERO=ICHAR('0')
  ININE=ICHAR('9')
  IRSET=0
  ICOUNT=0
C SET DEFAULT VALUES FOR PARAMETERS
  CALL SETDEF
C READ PARAMETER CARD
  10 CONTINUE
  READ(5,9001,END=8000)CARD
C TITLE CARD
  20 CONTINUE
  IF(CARD(1:6).EQ.PTITLE)THEN
    READ(CARD,9002)(TITLE(I),I=1,80)
    GO TO 10
C RANDOM SAMPLE CARD
  ELSE IF(CARD(1:14).EQ.PRSAMP)THEN
    READ(IRSAMP,9002)(TITLE(I),I=98,116)
    IRS=1
    GO TO 10
C RANDOM SEED CARD
  ELSE IF(CARD(1:12).EQ.PRSEED)THEN
    CALL DATSQZ(CARD(13:80),PRSEED,TMPCRD)
    READ(TMPCRD,9003)ISEED
C SET RANDOM NUMBER GENERATOR HERE IF NECESSARY
    IRSET=1
    GO TO 10
C RANDOM PAIRING CARD
  ELSE IF(CARD(1:15).EQ.PIRP)THEN
    IRP=1
    GO TO 10
C NUMBER OF OBSERVATIONS CARD
  ELSE IF(CARD(1:5).EQ.PNOBS)THEN
    CALL DATSQZ(CARD(6:80),PNOBS,TMPCRD)
    READ(TMPCRD,9003)N
    IF(N.LT.1)THEN
      WRITE(6,9006)N
      STOP 'RDPAR'
    ENDIF
    CALL CHKDIM(1,N,NMAX,PNOBS,LOBS)
    GO TO 10
C NUMBER OF REPETITIONS CARD
  ELSE IF(CARD(1:6).EQ.PNREP)THEN
    CALL DATSQZ(CARD(7:80),PNREP,TMPCRD)
    READ(TMPCRD,9003)NREP
    IF(NREP.LT.1)THEN
      WRITE(6,9007)NREP
      STOP 'RDPAR'
    ENDIF
    GO TO 10
C CORRELATION MATRIX CARD
  ELSE IF(CARD(1:1).EQ.PC)THEN

```



```

      READ(5,*,ERR=9000)NCV,
1      (ICVAR(I),JCVAR(I),CVAR(I),I=1,NCV)
      CALL CHKDIM(1,NCV,NCVAR,PCMINP,LCMINP)
      ICM=1
      GO TO 10
C OUTPUT OPTIONS CARD
      ELSE IF(CARD(1:7).EQ.POUT)THEN
        CALL OUTCRD(CARD)
        GO TO 10
C BETA DISTRIBUTION CARD
      ELSE IF(CARD(1:5).EQ.PBETA)THEN
        READ(5,*,ERR=9000)A,B,P,Q
        CALL CHKDAT(PBETA,A,B,P,Q)
        CALL WRTCRD(1,CARD(6:80))
        GO TO 10
C NORMAL DISTRIBUTION CARD
      ELSE IF(CARD(1:7).EQ.PNOR)THEN
        READ(5,*,ERR=9000)A,B
CBHL
        ICOUNT=ICOUNT+1
        XMC(ICOUNT)=0.5*(A+B)
CBHL
        CALL CHKDAT(PNOR,A,B,DUM1,DUM2)
        CALL WRTCRD(2,CARD(8:80))
        GO TO 10
C LOGNORMAL DISTRIBUTION CARD
      ELSE IF(CARD(1:10).EQ.PLNOR)THEN
        READ(5,*,ERR=9000)A,B
CBHL
        ICOUNT=ICOUNT+1
        ALN=LOG10(A)*2.30259
        BLN=LOG10(B)*2.30259
        DMEAN=0.5*(ALN+BLN)
        DSIGMA=(BLN-ALN)/6.18
        XMC(ICOUNT)=EXP(DMEAN+0.5*DSIGMA*DSIGMA)
CBHL
        CALL CHKDAT(PLNOR,A,B,A,B)
        CALL WRTCRD(3,CARD(11:80))
        GO TO 10
C UNIFORM DISTRIBUTION CARD
      ELSE IF(CARD(1:8).EQ.PUNI)THEN
        READ(5,*,ERR=9000)A,B
CBHL
        ICOUNT=ICOUNT+1
        XMC(ICOUNT)=0.5*(A+B)
CBHL
        CALL CHKDAT(PUNI,A,B,DUM1,DUM2)
        CALL WRTCRD(4,CARD(9:80))
        GO TO 10
C LOGUNIFORM DISTRIBUTION CARD
      ELSE IF(CARD(1:11).EQ.PLUNI)THEN
        READ(5,*,ERR=9000)A,B
        CALL CHKDAT(PLUNI,A,B,A,B)
        CALL WRTCRD(5,CARD(12:80))
        GO TO 10
C UNIFORM* DISTRIBUTION CARD
      ELSE IF(CARD(1:9).EQ.PUNIS)THEN
        READ(5,*,ERR=9000)NINT,(NSUBOB(I),I=1,NINT),
1      (SUBINT(I),I=1,NINT+1)
        CALL CHKSTR(PUNIS,CARD)
        CALL WRTCRD(6,CARD(10:80))
        GO TO 10
C LOGUNIFORM* DISTRIBUTION CARD
      ELSE IF(CARD(1:12).EQ.PLUNIS)THEN
        READ(5,*,ERR=9000)NINT,(NSUBOB(I),I=1,NINT),
1      (SUBINT(I),I=1,NINT+1)

```

```

        CALL CHKSTR(PLUNIS,CARD)
        CALL WRTCRD(7,CARD(13:80))
        GO TO 10
C TRIANGULAR DISTRIBUTION CARD
        ELSE IF(CARD(1:11).EQ.PTRIAG)THEN
            READ(5,*,ERR=9000)A,B,C
            CALL CHKTRI(A,B,C)
            CALL WRTCRD(8,CARD(12:80))
            GO TO 10
C USER DISTRIBUTION/DATA CARD(S)
        ELSE IF(CARD(1:18).EQ.PUSDST)THEN
            CALL WRTCRD(9,CARD(19:80))
30      CONTINUE
            READ(5,9001,END=8000)CARD
            ITEST=ICHAR(CARD(1:1))
            IF(CARD(1:1).EQ.BLANK.OR.CARD(1:1).EQ.PERIOD.OR.CARD(1:1).EQ.
1          MINUS.OR.(ITEST.GE.IZERO.AND.ITEST.LE.ININE))THEN
                WRITE(7,9001)CARD
                GO TO 30
            ELSE
                GO TO 20
            ENDIF
C UNDEFINED PARAMETER/DATA CARD
        ELSE
            WRITE(6,9004)CARD
            STOP 'RDPAR'
        ENDIF
C CHECK FOR INPUT ERRORS
8000  CONTINUE
        CALL CHKZRO(N,NV,IRSET)
        CALL CHKDIM(2,NV,NVAR,CDUM1,CDUM2)
        IF(ICM.EQ.1)CALL CMCRD
        RETURN
9000  CONTINUE
        WRITE(6,9005)CARD
        STOP 'RDPAR'
9001  FORMAT(A)
9002  FORMAT(80A1)
9003  FORMAT(I11)
9004  FORMAT(1H1,5X,'THE FOLLOWING CARD (POSSIBLY BLANK) IS NOT A ',
1      'VALID PARAMETER/DATA CARD',/,6X,'PLEASE CONSULT THE ',
2      'USER MANUAL FOR THE CORRECT PARAMETER/DATA CARD SYNTAX',
3      '//,3X,'***',A,'***')
9005  FORMAT(1H1,5X,'THE FOLLOWING PARAMETER CARD DID NOT HAVE THE ',
1      'CORRECT DATA CARD ASSOCIATED WITH IT',/,6X,'PLEASE ',
2      'CONSULT THE USER MANUAL FOR THE CORRECT DATA ',
3      'CARD SYNTAX',/,3X,'***',A,'***')
9006  FORMAT(1H1,5X,'THE NUMBER OF OBSERVATIONS REQUESTED IS LESS ',
1      'THAN ONE',I5)
9007  FORMAT(1H1,5X,'THE NUMBER OF REPETITIONS REQUESTED IS LESS ',
1      'THAN ONE',I5)
        END
C*****
C FUNCTION RIERFC1 IS USED IN GENERATING THE NORMAL AND
C LOGNORMAL DISTRIBUTIONS
C
        FUNCTION RIERFC1 (Y)
C
C
C      THIS IS THE SAME AS THE RIERFC ROUTINE IN THE DEAMOS LIBRARY
C      THE NAME MODIFICATION WAS TO PREVENT OUR LIBRARY FROM CALLING
C      ANOTHER LIBRARY.                      ELF    OCTOBER 1980
C
C      WRITTEN BY D.E. AMOS AND S.L. DANIEL, SEPTEMBER, 1972.
C
C      REFERENCES

```

```

C      HASTINGS, C.JR., APPROXIMATIONS FOR DIGITAL COMPUTERS,
C      PRINCETON UNIVERSITY PRESS, PRINCETON, N.J., 1955
C
C      COMMUNICATION FROM L.F. SHAMPINE FOR CHEBYSHEV COEFFICIENTS.
C
C      ABSTRACT
C      RIERFC EVALUATES THE INVERSE COERROR FUNCTION DEFINED BY
C
C          Y= ERFC(X)      0.LE.X.LT.INFINITY
C
C      WHERE 0.LT.Y.LE.1. CHEBYSHEV APPROXIMATIONS ON
C
C          EXP(-81).LE.Y.LT.0.1, 0.1.LE.Y.LT.0.5, 0.5.LE.Y.LE.1.
C
C      ARE USED WITH A CHANGE OF VARIABLES
C
C          YY=C1*W+C2, W=SQRT(-LN(Y)), YY=5.*Y-1.5, YY=2.*(1.-Y)
C
C      RESPECTIVELY. THE INVERSE OF THE NORMAL DISTRIBUTION IS GIVEN
C      BY
C
C          SQRT(2)*RIERFC(2.*(1.-RN))  0.5.LE.RN.LT.1.0,
C      X=
C          -SQRT(2)*RIERFC(2.*RN)      0.0.LT.RN.LT.0.5.
C
C      THE RELATIVE ERROR IN RIERFC DECREASES FROM 1.E-10 TO 3.E-13
C      AS Y INCREASES FROM EXP(-81) TO 1.0.
C
C      DESCRIPTION OF ARGUMENTS
C
C      INPUT
C
C          Y      - Y, EXP(-81).LE.Y.LE.1.
C
C      OUTPUT
C
C          RIERFC - VALUE FOR THE INVERSE COERROR FUNCTION
C
C      ERROR CONDITIONS
C          Y.LT.EXP(-81) OR Y.GT.1 ARE FATAL ERRORS
C
C      RIERFC USES SUBROUTINES ERRCHK, ERRGET, ERRPRT, ERXSET, ERSTGT
C      COMPILE DECKS RIERFC, ERRCHK
C
C      DIMENSION A(22,3)
C      DIMENSION A1(22), A2(22), A3(22)
C      SAVE A1,A2,A3,C1,C2
C      EQUIVALENCE (A(1,1),A1(1))
C      EQUIVALENCE (A(1,2),A2(1))
C      EQUIVALENCE (A(1,3),A3(1))
C
C      DATA(A1(I),I=1,22)/9.18725611735013E-01,0.,1.68792878000327E-02
C      1,0.,6.60337139058300E-04,0.,3.20203849839380E-05,0.,1.720
C      260607522481E-06,0.,9.81965971588191E-08,0.,5.83049613537653E
C      3-09,0.,3.56019351836136E-10,0.,2.21968915783128E-11,0.,1.
C      440639693109741E-12,0.,9.02597345404862E-14,0./
C
C      DATA(A2(I),I=1,22)/1.54701109458613E+00,-3.31460331083896E-01,4.33
C      1001124090060E-02,-1.06564004165532E-02,2.90613542304156E-03,-8.618
C      272838022491E-04,2.67933751795053E-04,-8.60838893942933E-05,2.83232
C      3058814598E-05,-9.48870819734494E-06,3.22422655069385E-06,-1.108157
C      478472076E-06,3.84464770797987E-07,-1.34439275565208E-07,4.73255976
C      5052393E-08,-1.67556011100019E-08,5.96199003969093E-09,-2.130705032
C      691886E-09,7.64427040920545E-10,-2.75198005584737E-10,9.93792246090
C      7789E-11,-3.59877382902119E-11/
C

```

```

DATA(A3(I),I=1,22)/1.10642888011036E+01,4.34299147561447E+00,-2.33
1781774969295E-02,4.23345215362947E-03,8.68757084192089E-06,-5.9826
21113270881E-04,4.50490139240298E-04,-2.54858131942102E-04,1.278241
389261340E-04,-5.97873878043957E-05,2.66474012012582E-05,-1.1438183
46209267E-05,4.75393030377615E-06,-1.91759589929610E-06,7.508064655
594834E-07,-2.84791180387123E-07,1.04187791696225E-07,-3.6456724368
69145E-08,1.20129296139030E-08,-3.61030126779729E-09,9.123561400817
759E-10,-1.36851363400914E-10/

C
DATA C1,C2/2.35777520630369E-01,1.35777520630369E+00/

C
IF (Y.LT.6.63967719958073E-36.OR.Y.GT.1.0) GO TO 50
IF (Y.GE.0.5) GO TO 10
IF (Y.GE.0.1) GO TO 20
J=3
W=SQRT(-LOG(Y))
D=C1*W-C2
GO TO 30
10 J=1
D=1.-Y
D=D+D
GO TO 30
20 J=2
D=5.*Y-1.5
30 TD=D+D
VNP1=0.
VN=0.
DO 40 L=1,21
K=22-L+1
TEMP=VN
VN=TD*VN-VNP1+A(K,J)
40 VNP1=TEMP
RIERFC1=D*VN-VNP1+.5*A(1,J)
IF (J.EQ.1) RIERFC1=D*RIERFC1
RETURN
50 WRITE(6,60)
RETURN

C
60 FORMAT(' Y LESS THAN EXP(-81.) OR Y GREATER THAN 1.0')
END
C*****
C SUBROUTINE S88FMT IS AN ERROR HANDLING ROUTINE USED IN THE
C POSITIVE DEFINITE CHECK OF THE CORRELATION MATRIX
C
SUBROUTINE S88FMT(N,IVALUE,IFMT)
C***BEGIN PROLOGUE S88FMT
C***DATE WRITTEN 790801 (YYMMDD)
C***REVISION DATE 820801 (YYMMDD)
C***CATEGORY NO. Z
C***KEYWORDS ERROR,XERROR PACKAGE
C***AUTHOR JONES, R. E., (SNLA)
C***PURPOSE INTEGER TO CHARACTER CONVERSION
C***DESCRIPTION
C ABSTRACT
C S88FMT REPLACES IFMT(1), ... ,IFMT(N) WITH THE
C CHARACTERS CORRESPONDING TO THE N LEAST SIGNIFICANT
C DIGITS OF IVALUE.
C
C TAKEN FROM THE BELL LABORATORIES PORT LIBRARY ERROR HANDLER
C LATEST REVISION --- 7 JUNE 1978
C***REFERENCES JONES R.E., *SLATEC COMMON MATHEMATICAL LIBRARY ERROR
C HANDLING PACKAGE*, SAND78-1189, SANDIA LABORATORIES,
C 1978.
C***ROUTINES CALLED (NONE)
C***END PROLOGUE S88FMT
C

```

```

        DIMENSION IFMT(N),IDIGIT(10)
        SAVE IDIGIT
        DATA IDIGIT(1),IDIGIT(2),IDIGIT(3),IDIGIT(4),IDIGIT(5),
1         IDIGIT(6),IDIGIT(7),IDIGIT(8),IDIGIT(9),IDIGIT(10)
2         /1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
C***FIRST EXECUTABLE STATEMENT  S88FMT
        NT = N
        IT = IVALUE
10     IF (NT .EQ. 0) RETURN
        INDEX = MOD(IT,10)
        IFMT(NT) = IDIGIT(INDEX+1)
        IT = IT/10
        NT = NT - 1
        GO TO 10
        END
C*****
C SUBROUTINE SETDEF SETS THE DEFAULT VALUES OF THE PARAMETERS
C
        SUBROUTINE SETDEF
        PARAMETER (NVAR=150)
        PARAMETER (LENT=125)
CBHL
        CHARACTER*1 TITLE
CBHL
CBHL
C     COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
C     1     ICORR,IDIST(NVAR),IRP
        COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
1     ICORR,IDIST(NVAR),IRP
        COMMON/PARAM/TITLE(LENT)
CBHL
        COMMON/OBSTR/NSTR,NOBSTR(NVAR)
        DO 10 I=1,LENT
CBHL    10 TITLE(I)=1H
10     TITLE(I)=' '
        N=0
        NV=0
        IRS=0
        ICM=0
        NREP=1
        IRP=0
        IDATA=0
        IHIST=0
        ICORR=0
        NSTR=0
        DO 20 I=1,NVAR
            IDIST(I)=0
            NOBSTR(I)=0
        20 CONTINUE
        RETURN
        END
C*****
C SUBROUTINE SIFT WILL SORT A VECTOR OF DATA IN INCREASING ORDER
C
        SUBROUTINE SIFT (XV,N)
        DIMENSION XV(N)
        M=N
10     M=M/2
        IF (M) 30,20,30
20     RETURN
30     K=N-M
        J=1
40     I=J
50     L=I+M
        IF (XV(I)-XV(L)) 70,70,60
60     A=XV(I)

```

```

      XV(I)=XV(L)
      XV(L)=A
      I=I-M
      IF (I) 70,70,50
70 J=J+1
      IF (J-K) 40,40,10
      END
C*****
C SUBROUTINE SSPEV IS USED IN THE POSITIVE DEFINITE CHECK OF THE
C CORRELATION MATRIX
C
      SUBROUTINE SSPEV(A,N,E,V,LDV,WORK,JOB,INFO)
C***BEGIN PROLOGUE  SSPEV
C***DATE WRITTEN   800808   (YYMMDD)
C***REVISION DATE  820801   (YYMMDD)
C***CATEGORY NO.  D4A1
C***KEYWORDS  EISPACK,EIGENVALUES,EIGENVECTORS,SYMMETRIC,REAL,PACKED
C***AUTHOR  KAHANER, K. K., (NBS)
C           MOLER, C. B., (U. OF NEW MEXICO)
C           STEWART, G. W., (U. OF MARYLAND)
C***PURPOSE  TO COMPUTE THE EIGENVALUES AND, OPTIONALLY, THE EIGEN-
C            VECTORS OF A REAL SYMMETRIC MATRIX STORED IN PACKED FORM.
C***DESCRIPTION
C   LICEPACK. THIS VERSION DATED 08/08/80.
C   DAVID KAHANER, CLEVE MOLER, PETE STEWART
C   N.B.S.      U.N.M.      N.B.S./U.MD.
C
C   ABSTRACT
C   SSPEV COMPUTES THE EIGENVALUES AND, OPTIONALLY, THE EIGENVECTORS
C   OF A REAL SYMMETRIC MATRIX STORED IN PACKED FORM.
C
C   CALL SEQUENCE PARAMETERS-
C   (THE VALUES OF PARAMETERS MARKED WITH * (STAR) WILL BE CHANGED
C   BY SSPEV.)
C
C   A*      REAL(N*(N+1)/2)
C           REAL SYMMETRIC PACKED INPUT MATRIX. CONTAINS UPPER
C           TRIANGLE AND DIAGONAL OF A, BY COLUMN (ELEMENTS
C           11, 12, 22, 13, 23, 33, ...).
C
C   N       INTEGER
C           SET BY THE USER TO
C           THE ORDER OF THE MATRIX A.
C
C   E*      REAL(N)
C           ON RETURN FROM SSPEV, E CONTAINS THE EIGENVALUES OF A.
C           SEE ALSO INFO BELOW.
C
C   V*      REAL(LDV,N)
C           ON RETURN FROM SSPEV, IF THE USER HAS SET JOB
C           = 0      V IS NOT REFERENCED.
C           = NONZERO THE N EIGENVECTORS OF A ARE STORED IN THE
C           FIRST N COLUMNS OF V. SEE ALSO INFO BELOW.
C
C   LDV     INTEGER
C           SET BY THE USER TO
C           THE LEADING DIMENSION OF THE ARRAY V IF JOB IS ALSO
C           SET NONZERO. IN THAT CASE, N MUST BE .LE. LDV.
C           IF JOB IS SET TO ZERO, LDV IS NOT REFERENCED.
C
C   WORK*   REAL(2N)
C           TEMPORARY STORAGE VECTOR. CONTENTS CHANGED BY SSPEV.
C
C   JOB     INTEGER
C           SET BY THE USER TO
C           = 0      EIGENVALUES ONLY TO BE CALCULATED BY SSPEV.

```

```

C          NEITHER V NOR LDV ARE REFERENCED.
C          = NONZERO EIGENVALUES AND VECTORS TO BE CALCULATED.
C          IN THIS CASE, A & V MUST BE DISTINCT ARRAYS.
C          ALSO, IF LDA .GT. LDV, SSPEV CHANGES ALL THE
C          ELEMENTS OF A THRU COLUMN N. IF LDA < LDV,
C          SSPEV CHANGES ALL THE ELEMENTS OF V THROUGH
C          COLUMN N. IF LDA=LDV, ONLY A(I,J) AND V(I,
C          J) FOR I,J = 1,...,N ARE CHANGED BY SSPEV.
C
C          INFO*   INTEGER
C                  ON RETURN FROM SSPEV, THE VALUE OF INFO IS
C                  = 0 FOR NORMAL RETURN.
C                  = K IF THE EIGENVALUE ITERATION FAILS TO CONVERGE.
C                  EIGENVALUES AND VECTORS 1 THROUGH K-1 ARE CORRECT.
C
C          ERROR MESSAGES-
C          NO. 1   RECOVERABLE  N IS GREATER THAN LDV AND JOB IS NONZERO
C          NO. 2   RECOVERABLE  N IS LESS THAN ONE
C
C          SUBROUTINES USED
C
C          EISPACK- IMTQL2, TQLRAT, TRBAK3, TRED3
C          SLATEC- XERROR
C***REFERENCES (NONE)
C***ROUTINES CALLED IMTQL2,TQLRAT,TRBAK3,TRED3,XERROR
C***END PROLOGUE SSPEV
      INTEGER I,INFO,J,K,LDV,M,N
      REAL A((N*(N+1))/2),E(N),V(LDV,N),WORK(1)
C***FIRST EXECUTABLE STATEMENT SSPEV
      NV=(N*(N+1))/2
      IF(N .GT. LDV) CALL XERROR(17HSSPEV-N .GT. LDV.,17,1,1)
      IF(N .GT. LDV) RETURN
      IF(N .LT. 1) CALL XERROR(14HSSPEV-N .LT. 1,14,2,1)
      IF(N .LT. 1) RETURN
C
C          CHECK N=1 CASE
C
C          E(1) = A(1)
C          INFO = 0
C          IF(N .EQ. 1) RETURN
C
C          IF(JOB.NE.0) GO TO 20
C
C          EIGENVALUES ONLY
C
C          CALL TRED3(N,NV,A,E,WORK(1),WORK(N+1))
C          CALL TQLRAT(N,E,WORK(N+1),INFO)
C          RETURN
C
C          EIGENVALUES AND EIGENVECTORS
C
C          20 CALL TRED3(N,NV,A,E,WORK(1),WORK(1))
C             DO 30 I = 1, N
C                 DO 25 J = 1, N
C                     25 V(I,J) = 0.
C             30 V(I,I) = 1.
C             CALL IMTQL2(LDV,N,E,WORK,V,INFO)
C             M = N
C             IF(INFO .NE. 0) M = INFO - 1
C             CALL TRBAK3(LDV,N,NV,A,M,V)
C             RETURN
C             END
C*****
C SUBROUTINE TQLRAT IS USED IN THE POSITIVE DEFINITE CHECK OF THE
C CORRELATION MATRIX

```

```

C
      SUBROUTINE TQLRAT(N,D,E2,IERR)
C***BEGIN PROLOGUE  TQLRAT
C***DATE WRITTEN   760101   (YYMMDD)
C***REVISION DATE  830518   (YYMMDD)
C***CATEGORY NO.   D4A5,D4C2A
C***KEYWORDS       EISPACK,EIGENVALUES,EIGENVECTORS
C***AUTHOR         SMITH ET AL
C***PURPOSE        COMPUTES EIGENVALUES OF SYMMETRIC TRIDIAGONAL MATRIX
C                   A RATIONAL VARIANT OF THE QL METHOD.
C***DESCRIPTION
C      THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE TQLRAT,
C      ALGORITHM 464, COMM. ACM 16, 689(1973) BY REINSCH.
C
C      THIS SUBROUTINE FINDS THE EIGENVALUES OF A SYMMETRIC
C      TRIDIAGONAL MATRIX BY THE RATIONAL QL METHOD.
C
C      ON INPUT
C
C          N IS THE ORDER OF THE MATRIX.
C
C          D CONTAINS THE DIAGONAL ELEMENTS OF THE INPUT MATRIX.
C
C          E2 CONTAINS THE SQUARES OF THE SUBDIAGONAL ELEMENTS OF THE
C          INPUT MATRIX IN ITS LAST N-1 POSITIONS.  E2(1) IS ARBITRARY.
C
C      ON OUTPUT
C
C          D CONTAINS THE EIGENVALUES IN ASCENDING ORDER.  IF AN
C          ERROR EXIT IS MADE, THE EIGENVALUES ARE CORRECT AND
C          ORDERED FOR INDICES 1,2,...IERR-1, BUT MAY NOT BE
C          THE SMALLEST EIGENVALUES.
C
C          E2 HAS BEEN DESTROYED.
C
C          IERR IS SET TO
C              ZERO      FOR NORMAL RETURN,
C              J          IF THE J-TH EIGENVALUE HAS NOT BEEN
C                      DETERMINED AFTER 30 ITERATIONS.
C
C      CALLS PYTHAG(A,B) FOR  $\text{SQRT}(A^2 + B^2)$ .
C
C      QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,
C      APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY
C      -----
C***REFERENCES  *MATRIX EIGENSYSTEM ROUTINES-EISPACKGUIDE*,
C                B.T.SMITH,J.M.BOYLE,J.J.DONGARRA,B.S.GARBOW,
C                Y.I.KEBE,V.C.KLEMA,C.B.MOLER,SPRINGER-VERLAG,
C                1976.
C***ROUTINES CALLED  PYTHAG
C***END PROLOGUE  TQLRAT
C
      INTEGER I,J,L,M,N,II,L1,MML,IERR
      REAL D(N),E2(N)
      REAL B,C,F,G,H,P,R,S,MACHEP
      REAL PYTHAG
      SAVE MACHEP
C
      DATA MACHEP/1.0E0/
C***FIRST EXECUTABLE STATEMENT  TQLRAT
      IF (MACHEP .NE. 1.0E0) GO TO 10
      05 MACHEP = 0.5E0*MACHEP
      IF (1.0E0 + MACHEP .GT. 1.0E0) GO TO 05
      MACHEP = 2.0E0*MACHEP
C
      10 IERR = 0

```



```

      IF (N .EQ. 1) GO TO 1001
C
      DO 100 I = 2, N
100  E2(I-1) = E2(I)
C
      F = 0.0E0
      B = 0.0E0
      E2(N) = 0.0E0
C
      DO 290 L = 1, N
        J = 0
        H = MACHEP * (ABS(D(L)) + SQRT(E2(L)))
        IF (B .GT. H) GO TO 105
        B = H
        C = B * B
C
      ..... LOOK FOR SMALL SQUARED SUB-DIAGONAL ELEMENT .....
105  DO 110 M = L, N
        IF (E2(M) .LE. C) GO TO 120
C
      ..... E2(N) IS ALWAYS ZERO, SO THERE IS NO EXIT
C
      ..... THROUGH THE BOTTOM OF THE LOOP .....
110  CONTINUE
C
120  IF (M .EQ. L) GO TO 210
130  IF (J .EQ. 30) GO TO 1000
      J = J + 1
C
      ..... FORM SHIFT .....
      L1 = L + 1
      S = SQRT(E2(L))
      G = D(L)
      P = (D(L1) - G) / (2.0E0 * S)
      R = PYTHAG(P, 1.0E0)
      D(L) = S / (P + SIGN(R, P))
      H = G - D(L)
C
      DO 140 I = L1, N
140  D(I) = D(I) - H
C
      F = F + H
C
      ..... RATIONAL QL TRANSFORMATION .....
      G = D(M)
      IF (G .EQ. 0.0E0) G = B
      H = G
      S = 0.0E0
      MML = M - L
C
      ..... FOR I=M-1 STEP -1 UNTIL L DO -- .....
      DO 200 II = 1, MML
        I = M - II
        P = G * H
        R = P + E2(I)
        E2(I+1) = S * R
        S = E2(I) / R
        D(I+1) = H + S * (H + D(I))
        G = D(I) - E2(I) / G
        IF (G .EQ. 0.0E0) G = B
        H = G * P / R
200  CONTINUE
C
      E2(L) = S * G
      D(L) = H
C
      ..... GUARD AGAINST UNDERFLOW IN CONVERGENCE TEST .....
      IF (H .EQ. 0.0E0) GO TO 210
      IF (ABS(E2(L)) .LE. ABS(C/H)) GO TO 210
      E2(L) = H * E2(L)
      IF (E2(L) .NE. 0.0E0) GO TO 130
210  P = D(L) + F
C
      ..... ORDER EIGENVALUES .....

```

```

      IF (L .EQ. 1) GO TO 250
C      ..... FOR I=L STEP -1 UNTIL 2 DO -- .....
      DO 230 II = 2, L
        I = L + 2 - II
        IF (P .GE. D(I-1)) GO TO 270
        D(I) = D(I-1)
230    CONTINUE
C
250    I = 1
270    D(I) = P
290    CONTINUE
C
      GO TO 1001
C      ..... SET ERROR -- NO CONVERGENCE TO AN
C      EIGENVALUE AFTER 30 ITERATIONS .....
1000  IERR = L
1001  RETURN
      END
C*****
C SUBROUTINE TRBAK3 IS USED IN THE POSITIVE DEFINITE CHECK OF THE
C CORRELATION MATRIX
C
      SUBROUTINE TRBAK3(NM,N,NV,A,M,Z)
C***BEGIN PROLOGUE  TRBAK3
C***DATE WRITTEN   760101   (YYMMDD)
C***REVISION DATE  830518   (YYMMDD)
C***CATEGORY NO.   D4C4
C***KEYWORDS      EISPACK,EIGENVALUES,EIGENVECTORS
C***AUTHOR        SMITH ET AL
C***PURPOSE       FORMS EIGENVECTORS OF REAL SYMMETRIC MATRIX FROM THE
C                  EIGENVECTORS OF SYMMETRIC TRIDIAGONAL MATRIX FORMED
C                  TRED3.
C***DESCRIPTION
C      THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE TRBAK3,
C      NUM. MATH. 11, 181-195(1968) BY MARTIN, REINSCH, AND WILKINSON.
C      HANDBOOK FOR AUTO. COMP., VOL.II-LINEAR ALGEBRA, 212-226(1971).
C
C      THIS SUBROUTINE FORMS THE EIGENVECTORS OF A REAL SYMMETRIC
C      MATRIX BY BACK TRANSFORMING THOSE OF THE CORRESPONDING
C      SYMMETRIC TRIDIAGONAL MATRIX DETERMINED BY TRED3.
C
C      ON INPUT
C
C      NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL
C      ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM
C      DIMENSION STATEMENT.
C
C      N IS THE ORDER OF THE MATRIX.
C
C      NV MUST BE SET TO THE DIMENSION OF THE ARRAY PARAMETER A
C      AS DECLARED IN THE CALLING PROGRAM DIMENSION STATEMENT.
C
C      A CONTAINS INFORMATION ABOUT THE ORTHOGONAL TRANSFORMATIONS
C      USED IN THE REDUCTION BY TRED3 IN ITS FIRST
C      N*(N+1)/2 POSITIONS.
C
C      M IS THE NUMBER OF EIGENVECTORS TO BE BACK TRANSFORMED.
C
C      Z CONTAINS THE EIGENVECTORS TO BE BACK TRANSFORMED
C      IN ITS FIRST M COLUMNS.
C
C      ON OUTPUT
C
C      Z CONTAINS THE TRANSFORMED EIGENVECTORS
C      IN ITS FIRST M COLUMNS.
C

```

```

C      NOTE THAT TRBAK3 PRESERVES VECTOR EUCLIDEAN NORMS.
C
C      QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,
C      APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY
C      -----
C***REFERENCES  *MATRIX EIGENSYSTEM ROUTINES-EISPACKGUIDE*,
C                B.T.SMITH,J.M.BOYLE,J.J.DONGARRA,B.S.GARBOW,
C                Y.I.KEBE,V.C.KLEMA,C.B.MOLER,SPRINGER-VERLAG,
C                1976.
C***ROUTINES CALLED  (NONE)
C***END PROLOGUE  TRBAK3
C
C      INTEGER I,J,K,L,M,N,IK,IZ,NM,NV
C      REAL A(NV),Z(NM,M)
C      REAL H,S
C
C***FIRST EXECUTABLE STATEMENT  TRBAK3
C      IF (M .EQ. 0) GO TO 200
C      IF (N .EQ. 1) GO TO 200
C
C      DO 140 I = 2, N
C        L = I - 1
C        IZ = (I * L) / 2
C        IK = IZ + I
C        H = A(IK)
C        IF (H .EQ. 0.0E0) GO TO 140
C
C        DO 130 J = 1, M
C          S = 0.0E0
C          IK = IZ
C
C          DO 110 K = 1, L
C            IK = IK + 1
C            S = S + A(IK) * Z(K,J)
C          110 CONTINUE
C          ..... DOUBLE DIVISION AVOIDS POSSIBLE UNDERFLOW .....
C          S = (S / H) / H
C          IK = IZ
C
C          DO 120 K = 1, L
C            IK = IK + 1
C            Z(K,J) = Z(K,J) - S * A(IK)
C          120 CONTINUE
C
C        130 CONTINUE
C
C      140 CONTINUE
C
C      200 RETURN
C      END
C*****
C SUBROUTINE TRED3 IS USED IN THE POSITIVE DEFINITE CHECK OF THE
C CORRELATION MATRIX
C
C      SUBROUTINE TRED3(N,NV,A,D,E,E2)
C***BEGIN PROLOGUE  TRED3
C***DATE WRITTEN   760101   (YYMMDD)
C***REVISION DATE  830518   (YYMMDD)
C***CATEGORY NO.   D4C1B1
C***KEYWORDS       EISPACK,EIGENVALUES,EIGENVECTORS
C***AUTHOR         SMITH ET AL
C***PURPOSE        REDUCE REAL SYMMETRIC MATRIX STORED IN PACKED FORM TO
C                  SYMMETRIC TRIDIAGONAL MATRIX USING ORTHOGONAL
C                  TRANSFORMATIONS.
C***DESCRIPTION
C      THIS SUBROUTINE IS A TRANSLATION OF THE ALGOL PROCEDURE TRED3,

```

```

C      NUM. MATH. 11, 181-195(1968) BY MARTIN, REINSCH, AND WILKINSON.
C      HANDBOOK FOR AUTO. COMP., VOL.II-LINEAR ALGEBRA, 212-226(1971).
C
C      THIS SUBROUTINE REDUCES A REAL SYMMETRIC MATRIX, STORED AS
C      A ONE-DIMENSIONAL ARRAY, TO A SYMMETRIC TRIDIAGONAL MATRIX
C      USING ORTHOGONAL SIMILARITY TRANSFORMATIONS.
C
C      ON INPUT
C
C          N IS THE ORDER OF THE MATRIX.
C
C          NV MUST BE SET TO THE DIMENSION OF THE ARRAY PARAMETER A
C          AS DECLARED IN THE CALLING PROGRAM DIMENSION STATEMENT.
C
C          A CONTAINS THE LOWER TRIANGLE OF THE REAL SYMMETRIC
C          INPUT MATRIX, STORED ROW-WISE AS A ONE-DIMENSIONAL
C          ARRAY, IN ITS FIRST N*(N+1)/2 POSITIONS.
C
C      ON OUTPUT
C
C          A CONTAINS INFORMATION ABOUT THE ORTHOGONAL
C          TRANSFORMATIONS USED IN THE REDUCTION.
C
C          D CONTAINS THE DIAGONAL ELEMENTS OF THE TRIDIAGONAL MATRIX.
C
C          E CONTAINS THE SUBDIAGONAL ELEMENTS OF THE TRIDIAGONAL
C          MATRIX IN ITS LAST N-1 POSITIONS.  E(1) IS SET TO ZERO.
C
C          E2 CONTAINS THE SQUARES OF THE CORRESPONDING ELEMENTS OF E.
C          E2 MAY COINCIDE WITH E IF THE SQUARES ARE NOT NEEDED.
C
C      QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,
C      APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY
C      -----
C***REFERENCES  *MATRIX EIGENSYSTEM ROUTINES-EISPACKGUIDE*,
C                B.T.SMITH,J.M.BOYLE,J.J.DONGARRA,B.S.GARBOW,
C                Y.I.KEBE,V.C.KLEMA,C.B.MOLER,SPRINGER-VERLAG,
C                1976.
C***ROUTINES CALLED  (NONE)
C***END PROLOGUE  TRED3
C
C      INTEGER I,J,K,L,N,II,IZ,JK,NV
C      REAL A(NV),D(N),E(N),E2(N)
C      REAL F,G,H,HH,SCALE
C
C***FIRST EXECUTABLE STATEMENT  TRED3
C      ..... FOR I=N STEP -1 UNTIL 1 DO -- .....
C      DO 300 II = 1, N
C          I = N + 1 - II
C          L = I - 1
C          IZ = (I * L) / 2
C          H = 0.0E0
C          SCALE = 0.0E0
C          IF (L .LT. 1) GO TO 130
C      ..... SCALE ROW (ALGOL TOL THEN NOT NEEDED) .....
C      DO 120 K = 1, L
C          IZ = IZ + 1
C          D(K) = A(IZ)
C          SCALE = SCALE + ABS(D(K))
120      CONTINUE
C
C      IF (SCALE .NE. 0.0E0) GO TO 140
130      E(I) = 0.0E0
C          E2(I) = 0.0E0
C          GO TO 290
C

```

```

140      DO 150 K = 1, L
          D(K) = D(K) / SCALE
          H = H + D(K) * D(K)
150      CONTINUE
C
          E2(I) = SCALE * SCALE * H
          F = D(L)
          G = -SIGN(SQRT(H),F)
          E(I) = SCALE * G
          H = H - F * G
          D(L) = F - G
          A(IZ) = SCALE * D(L)
          IF (L .EQ. 1) GO TO 290
          F = 0.0E0
C
          DO 240 J = 1, L
              G = 0.0E0
              JK = (J * (J-1)) / 2
C          ..... FORM ELEMENT OF A*U .....
              DO 180 K = 1, L
                  JK = JK + 1
                  IF (K .GT. J) JK = JK + K - 2
                  G = G + A(JK) * D(K)
180          CONTINUE
C          ..... FORM ELEMENT OF P .....
              E(J) = G / H
              F = F + E(J) * D(J)
240          CONTINUE
C
              HH = F / (H + H)
              JK = 0
C          ..... FORM REDUCED A .....
              DO 260 J = 1, L
                  F = D(J)
                  G = E(J) - HH * F
                  E(J) = G
C
                  DO 260 K = 1, J
                      JK = JK + 1
                      A(JK) = A(JK) - F * E(K) - G * D(K)
260          CONTINUE
C
290          D(I) = A(IZ+1)
          A(IZ+1) = SCALE * SQRT(H)
300      CONTINUE
C
          RETURN
          END
C*****
C SUBROUTINE TRIANG GENERATES THE TRIANGULAR DISTRIBUTION
C
      SUBROUTINE TRIANG(J)
      PARAMETER (NMAX=1000)
      PARAMETER (NVAR=150)
      PARAMETER (LENT=125)
CBHL
      CHARACTER*1 TITLE
CBHL
CBHL
C      COMMON/PARAM/TITLE(LENT), ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
C      1      ICORR,IDIST(NVAR),IRP
      COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
      1      ICORR,IDIST(NVAR),IRP
      COMMON/PARAM/TITLE(LENT)
CBHL
      COMMON/SAMP/X(NVAR*NMAX)

```

```

      LOC(I,J)=(J-1)*N+I
      PROBINC=1./FLOAT(N)
      IF(IRS.EQ.1)PROBINC=1.0
      READ(8)A,B,C
      C1=C-A
      C2=(B-A)/C1
      STRTPT=0.
      DO 10 I=1,N
        R=PROBINC*RAN( ISEED)+STRTPT
        IF(R.LE.C2)THEN
          X(LOC(I,J))=A+SQRT(R*C1*(B-A))
        ELSE
          X(LOC(I,J))=C-SQRT((1.-R)*C1*(C-B))
        ENDIF
        IF(IRS.EQ.0)STRTPT=STRTPT+PROBINC
10    CONTINUE
      RETURN
      END
C*****
C SUBROUTINE UNIFRM GENERATES THE UNIFORM, LOGUNIFORM, UNIFORM*,
C OR THE LOGUNIFORM* DISTRIBUTIONS
C
      SUBROUTINE UNIFRM(J, IDT)
      PARAMETER (NMAX=1000)
      PARAMETER (NVAR=150)
      PARAMETER (LENT=125)
CBHL
      CHARACTER*1 TITLE
CBHL
CBHL
C      COMMON/PARAM/TITLE(LENT), ISEED,N,NV, IRS, ICM,NREP, IDATA, IHIST,
C      1 ICORR, IDIST(NVAR), IRP
      COMMON/PARAM/ ISEED,N,NV, IRS, ICM,NREP, IDATA, IHIST,
      1 ICORR, IDIST(NVAR), IRP
CBHL
      COMMON/PARAMA/TITLE(LENT)
CBHL
      COMMON/SAMP/X(NVAR*NMAX)
      LOC(I,J)=(J-1)*N+I
      I=0
      IF( IDT.EQ.6.OR.IDT.EQ.7)THEN
        READ(8)NINT
      ELSE
        NINT=1
        NPT=N
      ENDIF
      DO 40 K=1,NINT
        IF( IDT.EQ.4.OR.IDT.EQ.5)THEN
          READ(8)A,B
        ELSE
          READ(8)NPT,A,B
          IF(NPT.EQ.0)GO TO 40
        ENDIF
        IF( IDT.EQ.5.OR.IDT.EQ.7)THEN
          A=LOG10(A)
          B=LOG10(B)
        ENDIF
        PROBINC=(B-A)/FLOAT(NPT)
        IF(IRS.NE.0)PROBINC=B-A
        DO 30 ID=1,NPT
          I=I+1
          IF(IRS.EQ.0)THEN
            STRTPT=(ID-1)*PROBINC
          ELSE
            STRTPT=0.0
          ENDIF
          X(LOC(I,J))=A+STRTPT+PROBINC*RAN( ISEED)

```

```

      IF (IDT.EQ.5.OR.IDT.EQ.7)X(LOC(I,J))=10.** (X(LOC(I,J)))
30  CONTINUE
40  CONTINUE
    RETURN
    END
C*****
C SUBROUTINE VIF COMPUTES THE VARIANCE INFLATION FACTOR OF A
C CORRELATION MATRIX
C
    SUBROUTINE VIF
    PARAMETER (NVAR=150)
    PARAMETER (LENT=125)
CBHL
    CHARACTER*1 TITLE
CBHL
CBHL
C    COMMON/PARAM/TITLE(LENT), ISEED,N,NV,IRS,ICM,NREP, IDATA, IHIST,
C    1    ICORR, IDIST(NVAR), IRP
    COMMON/PARAM/ ISEED,N,NV,IRS,ICM,NREP, IDATA, IHIST,
    1    ICORR, IDIST(NVAR), IRP
    COMMON/PARAM/ TITLE(LENT)
CBHL
    COMMON/CMATR/CORR( (NVAR*(NVAR+1))/2),LCM(NVAR),NCM
    LOC1(I,J)=J+(I*I-I)/2
    CALL DSINV(NV)
    CRKMX=0.0
    DO 630 I=1,NV
        CRK=CORR(LOC1(I,I))
        IF (CRK.GT.CRKMX)CRKMX=CRK
    630 CONTINUE
    WRITE(6,9001)CRKMX
    RETURN
9001 FORMAT('0THE VARIANCE INFLATION FACTOR FOR THIS MATRIX IS',
    1    F6.2)
    END
C*****
C SUBROUTINE WRTCRD PROCESSES THE DISTRIBUTION PARAMETER
C STATEMENTS
C
    SUBROUTINE WRTCRD( ITYPE,LABEL)
    PARAMETER (NVAR=150)
    PARAMETER (LENC=80)
    PARAMETER (LENT=125)
CBHL
    CHARACTER*1 TITLE
CBHL
CBHL
C    COMMON/PARAM/TITLE(LENT), ISEED,N,NV,IRS,ICM,NREP, IDATA, IHIST,
C    1    ICORR, IDIST(NVAR), IRP
    COMMON/PARAM/ ISEED,N,NV,IRS,ICM,NREP, IDATA, IHIST,
    1    ICORR, IDIST(NVAR), IRP
    COMMON/PARAM/ TITLE(LENT)
CBHL
    CHARACTER LABEL*(*) ,LABVAR*(LENC),BLANK
    PARAMETER (BLANK=' ')
    NV=NV+1
    IDIST(NV)=ITYPE
    NC=LEN(LABEL)
    IC=0
    LABVAR=BLANK
    100 CONTINUE
        IC=IC+1
        IF (IC.GT.NC)GO TO 200
        IF (LABEL(IC:IC).EQ.BLANK)GO TO 100
        LABVAR=LABEL(IC:NC)
    200 CONTINUE

```

```

        WRITE(9)LABVAR
        RETURN
        END
C*****
C SUBROUTINE WRTPAR PRINTS OUT THE DISTRIBUTION PARAMETERS
C
        SUBROUTINE WRTPAR
        PARAMETER (NVAR=150)
        PARAMETER (LENC=80)
        PARAMETER (LEND=8)
        PARAMETER (LENT=125)
CBHL
        CHARACTER*1 TITLE
CBHL
CBHL
C      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
C      1      ICORR,IDIST(NVAR),IRP
        COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
        1      ICORR,IDIST(NVAR),IRP
        COMMON/PARAM/TITLE(LENT)
CBHL
        CHARACTER LABEL*(LENC)
        CHARACTER*11 DIST(LEND)
        SAVE DIST
        DATA DIST(1)/'BETA',DIST(2)/'NORMAL',DIST(3)/'LOGNORMAL',
        1      DIST(4)/'UNIFORM',DIST(5)/'LOGUNIFORM',
        2      DIST(6)/'UNIFORM*',DIST(7)/'LOGUNIFORM*',
        3      DIST(8)/'TRIANGULAR'
        REWIND 8
        REWIND 9
        WRITE(6,9001)TITLE
        DO 100 I=1,NV
            ID=IDIST(I)
            READ(9)LABEL
            IF(ID.EQ.1)THEN
                READ(8)A,B,P,Q
                XMN=(A*Q+B*P)/(P+Q)
                VAR=P*Q*(B-A)**2/((P+Q+1)*(P+Q)**2)
                WRITE(6,9002)I,DIST(ID),A,B,LABEL
                WRITE(6,9006)P,Q,XMN,VAR
                GO TO 100
            ELSE IF(ID.EQ.8)THEN
                READ(8)A,B,C
                WRITE(6,9007)I,DIST(ID),LABEL,A,B,C
                GO TO 100
            ELSE IF(ID.EQ.9)THEN
                WRITE(6,9003)I,LABEL
                GO TO 100
            ELSE IF(ID.EQ.6.OR.ID.EQ.7)THEN
                READ(8)NINT
                WRITE(6,9004)I,DIST(ID),NINT,LABEL
                DO 50 J=1,NINT
                    READ(8)INT,A,B
                    WRITE(6,9005)INT,A,B
50                CONTINUE
                GO TO 100
            ELSE
                READ(8)A,B
                WRITE(6,9002)I,DIST(ID),A,B,LABEL
            ENDIF
        100 CONTINUE
        RETURN
9001 FORMAT('1',/,4X,125A1,/,4X,'VARIABLE  DISTRIBUTION',10X,
        1      'RANGE',12X,'LABEL')
9002 FORMAT('0',5X,I3,5X,A,2X,1PG10.3,' TO ',1PG10.3,2X,A)
9003 FORMAT('0',5X,I3,5X,'USER SUPPLIED DISTRIBUTION ',12X,A)

```



```

9004 FORMAT('0',5X,I3,5X,A,2X,'WITH ',I2,' SUBINTERVALS',6X,A)
9005 FORMAT(' ',13X,I4,' OBS',5X,1PG10.3,' TO ',1PG10.3)
9006 FORMAT(' ',13X,'WITH PARAMETERS P = ',F5.2,2X,'Q = ',F5.2,,
1      14X,'THIS CHOICE OF PARAMETERS GIVES A ',/,14X,
2      'POPULATION MEAN OF ',1PG10.3,' AND A',/,14X,
3      'POPULATION VARIANCE OF ',1PG10.3)
9007 FORMAT('0',5X,I3,5X,A,2X,'WITH PARAMETERS BELOW',5X,A,/,27X,
1      'A= ',1PG10.3,/,27X,'B= ',1PG10.3,/,27X,'C= ',1PG10.3)
      END
C*****
C SUBROUTINE XERABT IS AN ERROR HANDLING ROUTINE USED IN THE
C POSITIVE DEFINITE CHECK OF THE CORRELATION MATRIX
C
      SUBROUTINE XERABT(MESSG,NMESSG)
C***BEGIN PROLOGUE XERABT
C***DATE WRITTEN 790801 (YYMMDD)
C***REVISION DATE 820801 (YYMMDD)
C***CATEGORY NO. R3C
C***KEYWORDS ERROR,XERROR PACKAGE
C***AUTHOR JONES, R. E., (SNLA)
C***PURPOSE ABORTS PROGRAM EXECUTION AND PRINTS ERROR MESSAGE
C***DESCRIPTION
C      ABSTRACT
C      ***NOTE*** MACHINE DEPENDENT ROUTINE
C      XERABT ABORTS THE EXECUTION OF THE PROGRAM.
C      THE ERROR MESSAGE CAUSING THE ABORT IS GIVEN IN THE CALLING
C      SEQUENCE, IN CASE ONE NEEDS IT FOR PRINTING ON A DAYFILE,
C      FOR EXAMPLE.
C
C      DESCRIPTION OF PARAMETERS
C      MESSG AND NMESSG ARE AS IN XERROR, EXCEPT THAT NMESSG MAY
C      BE ZERO, IN WHICH CASE NO MESSAGE IS BEING SUPPLIED.
C
C      WRITTEN BY RON JONES, WITH SLATEC COMMON MATH LIBRARY SUBCOMMITTEE
C      LATEST REVISION --- 7 JUNE 1978
C***REFERENCES JONES R.E., *SLATEC COMMON MATHEMATICAL LIBRARY ERROR
C      HANDLING PACKAGE*, SAND78-1189, SANDIA LABORATORIES,
C      1978.
C***ROUTINES CALLED (NONE)
C***END PROLOGUE XERABT
C
      DIMENSION MESSG(NMESSG)
C***FIRST EXECUTABLE STATEMENT XERABT
      STOP
C      RETURN
      END
C*****
C SUBROUTINE XERCTL IS AN ERROR HANDLING ROUTINE USED IN THE
C POSITIVE DEFINITE CHECK OF THE CORRELATION MATRIX
C
      SUBROUTINE XERCTL(MESSG1,NMESSG,NERR,LEVEL,KONTRL)
C***BEGIN PROLOGUE XERCTL
C***DATE WRITTEN 790801 (YYMMDD)
C***REVISION DATE 820801 (YYMMDD)
C***CATEGORY NO. R3C
C***KEYWORDS ERROR,XERROR PACKAGE
C***AUTHOR JONES, R. E., (SNLA)
C***PURPOSE ALLOWS USER CONTROL OVER HANDLING OF INDIVIDUAL ERRORS
C***DESCRIPTION
C      ABSTRACT
C      ALLOWS USER CONTROL OVER HANDLING OF INDIVIDUAL ERRORS.
C      JUST AFTER EACH MESSAGE IS RECORDED, BUT BEFORE IT IS
C      PROCESSED ANY FURTHER (I.E., BEFORE IT IS PRINTED OR
C      A DECISION TO ABORT IS MADE), A CALL IS MADE TO XERCTL.
C      IF THE USER HAS PROVIDED HIS OWN VERSION OF XERCTL, HE
C      CAN THEN OVERRIDE THE VALUE OF KONTRL USED IN PROCESSING

```

```

C      THIS MESSAGE BY REDEFINING ITS VALUE.
C      KONTRL MAY BE SET TO ANY VALUE FROM -2 TO 2.
C      THE MEANINGS FOR KONTRL ARE THE SAME AS IN XSETF, EXCEPT
C      THAT THE VALUE OF KONTRL CHANGES ONLY FOR THIS MESSAGE.
C      IF KONTRL IS SET TO A VALUE OUTSIDE THE RANGE FROM -2 TO 2,
C      IT WILL BE MOVED BACK INTO THAT RANGE.
C
C      DESCRIPTION OF PARAMETERS
C
C      --INPUT--
C      MESSG1 - THE FIRST WORD (ONLY) OF THE ERROR MESSAGE.
C      NMESSG - SAME AS IN THE CALL TO XERROR OR XERRWV.
C      NERR - SAME AS IN THE CALL TO XERROR OR XERRWV.
C      LEVEL - SAME AS IN THE CALL TO XERROR OR XERRWV.
C      KONTRL - THE CURRENT VALUE OF THE CONTROL FLAG AS SET
C               BY A CALL TO XSETF.
C
C      --OUTPUT--
C      KONTRL - THE NEW VALUE OF KONTRL. IF KONTRL IS NOT
C               DEFINED, IT WILL REMAIN AT ITS ORIGINAL VALUE.
C               THIS CHANGED VALUE OF CONTROL AFFECTS ONLY
C               THE CURRENT OCCURRENCE OF THE CURRENT MESSAGE.
C***REFERENCES JONES R.E., *SLATEC COMMON MATHEMATICAL LIBRARY ERROR
C               HANDLING PACKAGE*, SAND78-1189, SANDIA LABORATORIES,
C               1978.
C***ROUTINES CALLED (NONE)
C***END PROLOGUE XERCTL
C
C***FIRST EXECUTABLE STATEMENT XERCTL
C      RETURN
C      END
C*****
C SUBROUTINE XERPRT IS AN ERROR HANDLING ROUTINE USED IN THE
C POSITIVE DEFINITE CHECK OF THE CORRELATION MATRIX
C
C      SUBROUTINE XERPRT(MESSG,NMESSG)
C***BEGIN PROLOGUE XERPRT
C***DATE WRITTEN 790801 (YYMMDD)
C***REVISION DATE 820801 (YYMMDD)
C***CATEGORY NO. Z
C***KEYWORDS ERROR,XERROR PACKAGE
C***AUTHOR JONES, R. E., (SNLA)
C***PURPOSE PRINTS ERROR MESSAGES
C***DESCRIPTION
C      ABSTRACT
C      PRINT THE HOLLERITH MESSAGE IN MESSG, OF LENGTH NMESSG,
C      ON EACH FILE INDICATED BY XGETUA.
C***REFERENCES JONES R.E., *SLATEC COMMON MATHEMATICAL LIBRARY ERROR
C               HANDLING PACKAGE*, SAND78-1189, SANDIA LABORATORIES,
C               1978.
C***ROUTINES CALLED ILMACH,S88FMT,XGETUA
C***END PROLOGUE XERPRT
C
C      INTEGER F(10),G(14),LUN(5)
C      DIMENSION MESSG(NMESSG)
C      SAVE F,G,LA,LCOM,LBLANK
C      DATA F(1),F(2),F(3),F(4),F(5),F(6),F(7),F(8),F(9),F(10)
C 1 / 1H( ,1H1 ,1HX ,1H , ,1H ,1H ,1HA ,1H ,1H ,1H) /
C      DATA G(1),G(2),G(3),G(4),G(5),G(6),G(7),G(8),G(9),G(10)
C 1 / 1H( ,1H1 ,1HX ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H /
C      DATA G(11),G(12),G(13),G(14)
C 1 / 1H ,1H ,1H ,1H) /
C      DATA LA/1HA/,LCOM/1H/,/,LBLANK/1H /
C      PREPARE FORMAT FOR WHOLE LINES
C***FIRST EXECUTABLE STATEMENT XERPRT
C      NCHAR = ILMACH(6)

```

```

        NFIELD = 72/NCHAR
        CALL S88FMT(2,NFIELD,F(5))
        CALL S88FMT(2,NCHAR,F(8))
C      PREPARE FORMAT FOR LAST, PARTIAL LINE, IF NEEDED
        NCHARL = NFIELD*NCHAR
        NLines = NMESG/NCHARL
        NWORD = NLines*NFIELD
        NCHREM = NMESG - NLines*NCHARL
        IF (NCHREM.LE.0) GO TO 40
        DO 10 I=4,13
10       G(I) = LBLANK
        NFIELD = NCHREM/NCHAR
        IF (NFIELD.LE.0) GO TO 20
C      PREPARE WHOLE WORD FIELDS
        G(4) = LCOM
        CALL S88FMT(2,NFIELD,G(5))
        G(7) = LA
        CALL S88FMT(2,NCHAR,G(8))
20       CONTINUE
        NCHLST = MOD(NCHREM,NCHAR)
        IF (NCHLST.LE.0) GO TO 30
C      PREPARE PARTIAL WORD FIELD
        G(10) = LCOM
        G(11) = LA
        CALL S88FMT(2,NCHLST,G(12))
30       CONTINUE
40       CONTINUE
C      PRINT THE MESSAGE
        NWORD1 = NWORD+1
        NWORD2 = (NMESG+NCHAR-1)/NCHAR
        CALL XGETUA(LUN,NUNIT)
        DO 50 KUNIT = 1,NUNIT
            IUNIT = LUN(KUNIT)
            IF (IUNIT.EQ.0) IUNIT = ILMACH(4)
C          IF (NWORD.GT.0) WRITE (IUNIT,F) (MESSG(I),I=1,NWORD)
C          IF (NCHREM.GT.0) WRITE (IUNIT,G) (MESSG(I),I=NWORD1,NWORD2)
            IF (NWORD.GT.0) WRITE (IUNIT,45) (MESSG(I),I=1,NWORD)
            IF (NCHREM.GT.0) WRITE (IUNIT,45) (MESSG(I),I=NWORD1,NWORD2)
45          FORMAT(20I4)
50       CONTINUE
CBHL
        RETURN
        END
C*****
C SUBROUTINE XERROR IS AN ERROR HANDLING ROUTINE USED IN THE
C POSITIVE DEFINITE CHECK OF THE CORRELATION MATRIX
C
        SUBROUTINE XERROR(MESSG,NMESG,NERR,LEVEL)
C***BEGIN PROLOGUE  XERROR
C***DATE WRITTEN   790801   (YYMMDD)
C***REVISION DATE  820801   (YYMMDD)
C***CATEGORY NO.  R3C
C***KEYWORDS  ERROR,XERROR PACKAGE
C***AUTHOR  JONES, R. E., (SNLA)
C***PURPOSE  PROCESSES AN ERROR (DIAGNOSTIC) MESSAGE
C***DESCRIPTION
C      ABSTRACT
C          XERROR PROCESSES A DIAGNOSTIC MESSAGE, IN A MANNER
C          DETERMINED BY THE VALUE OF LEVEL AND THE CURRENT VALUE
C          OF THE LIBRARY ERROR CONTROL FLAG, KONTRL.
C          (SEE SUBROUTINE XSETF FOR DETAILS.)
C
C      DESCRIPTION OF PARAMETERS
C      --INPUT--
C          MESSG - THE HOLLERITH MESSAGE TO BE PROCESSED, CONTAINING
C                  NO MORE THAN 72 CHARACTERS.

```

```

C      NMESSG- THE ACTUAL NUMBER OF CHARACTERS IN MESSG.
C      NERR   - THE ERROR NUMBER ASSOCIATED WITH THIS MESSAGE.
C              NERR MUST NOT BE ZERO.
C      LEVEL  - ERROR CATEGORY.
C              =2 MEANS THIS IS AN UNCONDITIONALLY FATAL ERROR.
C              =1 MEANS THIS IS A RECOVERABLE ERROR.  (I.E., IT IS
C                NON-FATAL IF XSETF HAS BEEN APPROPRIATELY CALLED.)
C              =0 MEANS THIS IS A WARNING MESSAGE ONLY.
C              =-1 MEANS THIS IS A WARNING MESSAGE WHICH IS TO BE
C                PRINTED AT MOST ONCE, REGARDLESS OF HOW MANY
C                TIMES THIS CALL IS EXECUTED.
C
C      EXAMPLES
C      CALL XERROR(23HSMOOTH -- NUM WAS ZERO.,23,1,2)
C      CALL XERROR(43HINTEG -- LESS THAN FULL ACCURACY ACHIEVED.,
C                43,2,1)
C      CALL XERROR(65HROOTER -- ACTUAL ZERO OF F FOUND BEFORE INTERVAL
C 1 FULLY COLLAPSED.,65,3,0)
C      CALL XERROR(39HEXP   -- UNDERFLOWS BEING SET TO ZERO.,39,1,-1)
C
C      WRITTEN BY RON JONES, WITH SLATEC COMMON MATH LIBRARY SUBCOMMITTEE
C      LATEST REVISION --- 7 FEB 1979
C***REFERENCES  JONES R.E., *SLATEC COMMON MATHEMATICAL LIBRARY ERROR
C                HANDLING PACKAGE*, SAND78-1189, SANDIA LABORATORIES,
C                1978.
C***ROUTINES CALLED  XERRWV
C***END PROLOGUE  XERROR
C
C      DIMENSION MESSG(NMESSG)
C***FIRST EXECUTABLE STATEMENT  XERROR
C      CALL XERRWV(MESSG,NMESSG,NERR,LEVEL,0,0,0,0,0.,0.)
C      RETURN
C      END
C*****
C      SUBROUTINE XERRWV IS AN ERROR HANDLING ROUTINE USED IN THE
C      POSITIVE DEFINITE CHECK OF THE CORRELATION MATRIX
C
C      SUBROUTINE XERRWV(MESSG,NMESSG,NERR,LEVEL,NI,I1,I2,NR,R1,R2)
C***BEGIN PROLOGUE  XERRWV
C***DATE WRITEN 800319 (YYMMDD)
C***REVISION DATE 820801 (YYMMDD)
C***CATEGORY NO. Q
C***KEYWORDS (NONE)
C***AUTHOR JONES, R. E., (SLA)
C***PURPOSE PROCESSES ERROR MESSAGE ALLOWING 2 INTEGER AND TWO REAL
C            VALUES TO BE INCLUDED IN THE MESSAGE.
C***DESCRIPTION
C      ABSTRACT
C      XERRWV PROCESSES A DIAGNOSTIC MESSAGE, IN A MANNER
C      DETERMINED BY THE VALUE OF LEVEL AND THE CURRENT VALUE
C      OF THE LIBRARY ERROR CONTROL FLAG, KONTRL.
C      (SEE SUBROUTINE XSETF FOR DETAILS.)
C      IN ADDITION, UP TO TWO INTEGER VALUES AND TWO REAL
C      VALUES MAY BE PRINTED ALONG WITH THE MESSAGE.
C
C      DESCRIPTION OF PARAMETERS
C      --INPUT--
C      MESSG - THE HOLLERITH MESSAGE TO BE PROCESSED.
C      NMESSG- THE ACTUAL NUMBER OF CHARACTERS IN MESSG.
C      NERR - THE ERROR NUMBER ASSOCIATED WITH THIS MESSAGE.
C            NERR MUST NOT BE ZERO.
C      LEVEL - ERROR CATEGORY.
C            =2 MEANS THIS IS AN UNCONDITIONALLY FATAL ERROR.
C            =1 MEANS THIS IS A RECOVERABLE ERROR.  (I.E., IT IS
C              NON-FATAL IF XSETF HAS BEEN APPROPRIATELY CALLED.)
C            =0 MEANS THIS IS A WARNING MESSAGE ONLY.

```

```

C          =-1 MEANS THIS IS A WARNING MESSAGE WHICH IS TO BE
C          PRINTED AT MOST ONCE, REGARDLESS OF HOW MANY
C          TIMES THIS CALL IS EXECUTED.
C      NI    - NUMBER OF INTEGER VALUES TO BE PRINTED. (0 TO 2)
C      I1    - FIRST INTEGER VALUE.
C      I2    - SECOND INTEGER VALUE.
C      NR    - NUMBER OF REAL VALUES TO BE PRINTED. (0 TO 2)
C      R1    - FIRST REAL VALUE.
C      R2    - SECOND REAL VALUE.
C
C      EXAMPLES
C      CALL XERROR(29HSMOOTH -- NUM (=I1) WAS ZERO.,29,1,2,
C 1  1,NUM,0,0,0.,0.)
C      CALL XERRWV(54HQADXY -- REQUESTED ERROR (R1) LESS THAN MINIMUM
C 1  (R2).,54,77,1,0,0,0,2,ERRREQ,ERRMIN)
C
C      WRITTEN BY RON JONES, WITH SLATEC COMMON MATH LIBRARY SUBCOMMITTEE
C      LATEST REVISION --- 19 MAR 1980
C***REFERENCES JONES R.E., *SLATEC COMMON MATHEMATICAL LIBRARY ERROR
C              HANDLING PACKAGE*, SAND78-1189, SANDIA LABORATORIES,
C              1978.
C***ROUTINES CALLED FDUMP,I1MACH,J4SAVE,XERABT,XERCTL,XERPRT,XERSAV,
C              XGETUA
C***END PROLOGUE XERRWV
C
C      DIMENSION MESSG(NMESSG),LUN(5)
C***FIRST EXECUTABLE STATEMENT XERRWV
C      LKNTRL = J4SAVE(2,0,.FALSE.)
C      MAXMES = J4SAVE(4,0,.FALSE.)
C      CHECK FOR VALID INPUT
C      IF ((NMESSG.GT.0).AND.(NERR.NE.0).AND.
C 1      (LEVEL.GE.(-1)).AND.(LEVEL.LE.2)) GO TO 10
C      IF (LKNTRL.GT.0) CALL XERPRT(17HFATAL ERROR IN...,17)
C      CALL XERPRT(23HXERROR -- INVALID INPUT,23)
C      IF (LKNTRL.GT.0) CALL FDUMP
C      IF (LKNTRL.GT.0) CALL XERPRT(29HJOB ABORT DUE TO FATAL ERROR.,
C 1  29)
C      IF (LKNTRL.GT.0) CALL XERSAV(1H ,0,0,0,KDUMMY)
C      CALL XERABT(23HXERROR -- INVALID INPUT,23)
C      RETURN
C 10 CONTINUE
C      RECORD MESSAGE
C      JUNK = J4SAVE(1,NERR,.TRUE.)
C      CALL XERSAV(MESSG,NMESSG,NERR,LEVEL,KOUNT)
C      LET USER OVERRIDE
C      LFIRST = MESSG(1)
C      LMESSG = NMESSG
C      LERR = NERR
C      LLEVEL = LEVEL
C      CALL XERCTL(LFIRST,LMESSG,LERR,LLEVEL,LKNTRL)
C      RESET TO ORIGINAL VALUES
C      LMESSG = NMESSG
C      LERR = NERR
C      LLEVEL = LEVEL
C      LKNTRL = MAX0(-2,MIN0(2,LKNTRL))
C      MKNTRL = IABS(LKNTRL)
C      DECIDE WHETHER TO PRINT MESSAGE
C      IF ((LLEVEL.LT.2).AND.(LKNTRL.EQ.0)) GO TO 100
C      IF (((LLEVEL.EQ.(-1)).AND.(KOUNT.GT.MIN0(1,MAXMES)))
C 1.OR.((LLEVEL.EQ.0) .AND.(KOUNT.GT.MAXMES))
C 2.OR.((LLEVEL.EQ.1) .AND.(KOUNT.GT.MAXMES).AND.(MKNTRL.EQ.1))
C 3.OR.((LLEVEL.EQ.2) .AND.(KOUNT.GT.MAX0(1,MAXMES)))) GO TO 100
C      IF (LKNTRL.LE.0) GO TO 20
C      CALL XERPRT(1H ,1)
C      INTRODUCTION
C      IF (LLEVEL.EQ.(-1)) CALL XERPRT

```

```

1(57HWARNING MESSAGE...THIS MESSAGE WILL ONLY BE PRINTED ONCE.,57)
  IF (LLEVEL.EQ.0) CALL XERPRT(13HWARNING IN...,13)
  IF (LLEVEL.EQ.1) CALL XERPRT
1    (23HRECOVERABLE ERROR IN...,23)
  IF (LLEVEL.EQ.2) CALL XERPRT(17HFATAL ERROR IN...,17)
20  CONTINUE
C    MESSAGE
  CALL XERPRT(MESSG,LMESSG)
  CALL XGETUA(LUN,NUNIT)
  DO 50 KUNIT=1,NUNIT
    IUNIT = LUN(KUNIT)
    IF (IUNIT.EQ.0) IUNIT = ILMACH(4)
    IF (NI.GE.1) WRITE (IUNIT,22) I1
    IF (NI.GE.2) WRITE (IUNIT,23) I2
    IF (NR.GE.1) WRITE (IUNIT,24) R1
    IF (NR.GE.2) WRITE (IUNIT,25) R2
22    FORMAT (11X,21HIN ABOVE MESSAGE, I1=,I10)
23    FORMAT (11X,21HIN ABOVE MESSAGE, I2=,I10)
24    FORMAT (11X,21HIN ABOVE MESSAGE, R1=,E20.10)
25    FORMAT (11X,21HIN ABOVE MESSAGE, R2=,E20.10)
    IF (LKNTL.LE.0) GO TO 40
C      ERROR NUMBER
      WRITE (IUNIT,30) LERR
30    FORMAT (15H ERROR NUMBER =,I10)
40    CONTINUE
50    CONTINUE
C    TRACE-BACK
    CALL FDUMP
100  CONTINUE
    IFATAL = 0
    IF ((LLEVEL.EQ.2).OR.((LLEVEL.EQ.1).AND.(MKNTL.EQ.2)))
1    IFATAL = 1
C    QUIT HERE IF MESSAGE IS NOT FATAL
    IF (IFATAL.LE.0) RETURN
    IF ((LKNTL.LE.0).OR.(KOUNT.GT.MAX0(1,MAXMES))) GO TO 120
C    PRINT REASON FOR ABORT
    IF (LLEVEL.EQ.1) CALL XERPRT
1    (35HJOB ABORT DUE TO UNRECOVERED ERROR.,35)
    IF (LLEVEL.EQ.2) CALL XERPRT
1    (29HJOB ABORT DUE TO FATAL ERROR.,29)
C    PRINT ERROR SUMMARY
    CALL XERSAV(1H ,-1,0,0,KDUMMY)
120  CONTINUE
C    ABORT
    IF ((LLEVEL.EQ.2).AND.(KOUNT.GT.MAX0(1,MAXMES))) LMESSG = 0
    CALL XERABT(MESSG,LMESSG)
    RETURN
    END
C*****
C SUBROUTINE XERSAV IS AN ERROR HANDLING ROUTINE USED IN THE
C POSITIVE DEFINITE CHECK OF THE CORRELATION MATRIX
C
  SUBROUTINE XERSAV(MESSG,NMESSG,NERR,LEVEL,ICOUNT)
C***BEGIN PROLOGUE XERSAV
C***DATE WRITTEN 800319 (YYMMDD)
C***REVISION DATE 820801 (YYMMDD)
C***CATEGORY NO. Q
C***KEYWORDS (NONE)
C***AUTHOR JONES, R. E., (SLA)
C***PURPOSE Records that an error occurred.
C***DESCRIPTION
C    Abstract
C    Record that this error occurred.
C
C    Description of Parameters
C    --Input--

```

```

C      MESSG, NMESSG, NERR, LEVEL are as in XERROR,
C      except that when NMESSG=0 the tables will be
C      dumped and cleared, and when NMESSG is less than zero the
C      tables will be dumped and not cleared.
C      --Output--
C      ICOUNT will be the number of times this message has
C      been seen, or zero if the table has overflowed and
C      does not contain this message specifically.
C      When NMESSG=0, ICOUNT will not be altered.
C
C      Written by Ron Jones, with SLATEC Common Math Library Subcommittee
C      Latest revision --- 19 Mar 1980
C***REFERENCES JONES R.E., *SLATEC COMMON MATHEMATICAL LIBRARY ERROR
C      HANDLING PACKAGE*, SAND78-1189, SANDIA LABORATORIES,
C      1978.
C***ROUTINES CALLED ILMACH,S88FMT,XGETUA
C***END PROLOGUE XERSAV
C
C      INTEGER FF(17),LUN(5)
C      DIMENSION MESSG(1)
C      DIMENSION MESTAB(10),NERTAB(10),LEVTAB(10),KOUNT(10)
C      SAVE MESTAB,NERTAB,LEVTAB,KOUNT,KOUNTX,FF
C      DATA MESTAB(1),MESTAB(2),MESTAB(3),MESTAB(4),MESTAB(5),
1      MESTAB(6),MESTAB(7),MESTAB(8),MESTAB(9),MESTAB(10)
2      /0,0,0,0,0,0,0,0,0,0/
C      DATA NERTAB(1),NERTAB(2),NERTAB(3),NERTAB(4),NERTAB(5),
1      NERTAB(6),NERTAB(7),NERTAB(8),NERTAB(9),NERTAB(10)
2      /0,0,0,0,0,0,0,0,0,0/
C      DATA LEVTAB(1),LEVTab(2),LEVTab(3),LEVTab(4),LEVTab(5),
1      LEVTAB(6),LEVTab(7),LEVTab(8),LEVTab(9),LEVTab(10)
2      /0,0,0,0,0,0,0,0,0,0/
C      DATA KOUNT(1),KOUNT(2),KOUNT(3),KOUNT(4),KOUNT(5),
1      KOUNT(6),KOUNT(7),KOUNT(8),KOUNT(9),KOUNT(10)
2      /0,0,0,0,0,0,0,0,0,0/
C      DATA KOUNTX/0/
C      DATA FF(1),FF(2),FF(3),FF(4),FF(5),FF(6),FF(7),FF(8),FF(9),FF(10),
1      FF(11),FF(12),FF(13),FF(14),FF(15),FF(16),FF(17)
2      /1H( ,1H1 ,1HX ,1H ,1HA ,1H ,1H ,1H ,1HI ,1H ,
3      1H ,1H ,1H2 ,1HI ,1H1 ,1H0 ,1H) /
C***FIRST EXECUTABLE STATEMENT XERSAV
C      IF (NMESSG.GT.0) GO TO 80
C      DUMP THE TABLE
C      IF (KOUNT(1).EQ.0) RETURN
C      PREPARE FORMAT
C      NCHAR = ILMACH(6)
C      CALL S88FMT(2,NCHAR,FF(6))
C      NCOL = 20 - NCHAR
C      CALL S88FMT(2,NCOL,FF(10))
C      PRINT TO EACH UNIT
C      CALL XGETUA(LUN,NUNIT)
C      DO 60 KUNIT=1,NUNIT
C      IUNIT = LUN(KUNIT)
C      IF (IUNIT.EQ.0) IUNIT = ILMACH(4)
C      PRINT TABLE HEADER
C      WRITE (IUNIT,10)
10      FORMAT (32H0 ERROR MESSAGE SUMMARY/
1      41H FIRST WORD NERR LEVEL COUNT)
C      PRINT BODY OF TABLE
C      DO 20 I=1,10
C      IF (KOUNT(I).EQ.0) GO TO 30
CBHL      WRITE (IUNIT,FF) MESTAB(I),NERTAB(I),LEVTab(I),KOUNT(I)
15      WRITE (IUNIT,15) MESTAB(I),NERTAB(I),LEVTab(I),KOUNT(I)
CBHL      FORMAT(4I10)
20      CONTINUE
30      CONTINUE

```

```

C          PRINT NUMBER OF OTHER ERRORS
          IF (KOUNTX.NE.0) WRITE (IUNIT,40) KOUNTX
40         FORMAT (41H00OTHER ERRORS NOT INDIVIDUALLY TABULATED=,I10)
          WRITE (IUNIT,50)
50         FORMAT (1X)
60         CONTINUE
          IF (NMESSG.LT.0) RETURN
C          CLEAR THE ERROR TABLES
          DO 70 I=1,10
70          KOUNT(I) = 0
          KOUNTX = 0
          RETURN
80 CONTINUE
C          PROCESS A MESSAGE...
C          SEARCH FOR THIS MESSG, OR ELSE AN EMPTY SLOT FOR THIS MESSG,
C          OR ELSE DETERMINE THAT THE ERROR TABLE IS FULL.
          DO 90 I=1,10
            II = I
            IF (KOUNT(I).EQ.0) GO TO 110
            IF (MESSG(1).NE.MESTAB(I)) GO TO 90
            IF (NERR.NE.NERTAB(I)) GO TO 90
            IF (LEVEL.NE.LEVTAB(I)) GO TO 90
            GO TO 100
90 CONTINUE
C          THREE POSSIBLE CASES...
C          TABLE IS FULL
          KOUNTX = KOUNTX+1
          ICOUNT = 1
          RETURN
C          MESSAGE FOUND IN TABLE
100         KOUNT(II) = KOUNT(II) + 1
          ICOUNT = KOUNT(II)
          RETURN
C          EMPTY SLOT FOUND FOR NEW MESSAGE
110         MESTAB(II) = MESSG(1)
          NERTAB(II) = NERR
          LEVTAB(II) = LEVEL
          KOUNT(II) = 1
          ICOUNT = 1
          RETURN
END
C*****
C SUBROUTINE XGETUA IS AN ERROR HANDLING ROUTINE USED IN THE
C POSITIVE DEFINITE CHECK OF THE CORRELATION MATRIX
C
          SUBROUTINE XGETUA(IUNIT,N)
C***BEGIN PROLOGUE  XGETUA
C***DATE WRITTEN   790801   (YYMMDD)
C***REVISION DATE  820801   (YYMMDD)
C***CATEGORY NO.   Q
C***KEYWORDS       (NONE)
C***AUTHOR         JONES, R. E., (SLA)
C***PURPOSE        RETURNS UNIT NUMBER(S) TO WHICH ERROR MESSAGES ARE BEING
C                   SENT
C***DESCRIPTION
C          ABSTRACT
C          XGETUA MAY BE CALLED TO DETERMINE THE UNIT NUMBER OR NUMBERS
C          TO WHICH ERROR MESSAGES ARE BEING SENT.
C          THESE UNIT NUMBERS MAY HAVE BEEN SET BY A CALL TO XSETUN,
C          OR A CALL TO XSETUA, OR MAY BE A DEFAULT VALUE.
C
C          DESCRIPTION OF PARAMETERS
C          --OUTPUT--
C          IUNIT - AN ARRAY OF ONE TO FIVE UNIT NUMBERS, DEPENDING
C                  ON THE VALUE OF N. A VALUE OF ZERO REFERS TO THE
C                  DEFAULT UNIT, AS DEFINED BY THE ILMACH MACHINE

```



```

C          CONSTANT ROUTINE.  ONLY IUNIT(1),...,IUNIT(N) ARE
C          DEFINED BY XGETUA.  THE VALUES OF IUNIT(N+1),...,
C          IUNIT(5) ARE NOT DEFINED (FOR N .LT. 5) OR ALTERED
C          IN ANY WAY BY XGETUA.
C          N      - THE NUMBER OF UNITS TO WHICH COPIES OF THE
C          ERROR MESSAGES ARE BEING SENT.  N WILL BE IN THE
C          RANGE FROM 1 TO 5.
C
C          WRITTEN BY RON JONES, WITH SLATEC COMMON MATH LIBRARY SUBCOMMITTEE
C***REFERENCES  JONES R.E., *SLATEC COMMON MATHEMATICAL LIBRARY ERROR
C              HANDLING PACKAGE*, SAND78-1189, SANDIA LABORATORIES,
C              1978.
C***ROUTINES CALLED  J4SAVE
C***END PROLOGUE  XGETUA
C
C          DIMENSION IUNIT(5)
C***FIRST EXECUTABLE STATEMENT  XGETUA
C          N = J4SAVE(5,0,.FALSE.)
C          DO 30 I=1,N
C              INDEX = I+4
C              IF (I.EQ.1) INDEX = 3
C              IUNIT(I) = J4SAVE(INDEX,0,.FALSE.)
C          30 CONTINUE
C          RETURN
C          END
C*****
C          SUBROUTINE USRDST(J)
C
C          C THE FOLLOWING SIX LINES OF CODE ARE REQUIRED BY USRDST
C
C          PARAMETER (NMAX=1000)
C          PARAMETER (NVAR=150)
C          PARAMETER (LENT=125)
C
C          CBHL
C          CHARACTER*1 TITLE
C
C          CBHL
C          COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
C          1      ICORR,IDIST(NVAR),IRP
C          COMMON/PARAM/ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
C          1      ICORR,IDIST(NVAR),IRP
C          COMMON/PARAM/TITLE(LENT)
C
C          CBHL
C          COMMON/SAMP/X(NMAX*NVAR)
C
C          C THE FOLLOWING LINE OF CODE IS SUPPLIED BY THE USER
C          C XVAL AND FREQ MUST BE DIMENSIONED TO THE NUMBER OF UNIQUE VALUES
C          C THAT THE DISCRETE RANDOM VARIABLE TAKES ON AND CDF MUST BE
C          C DIMENSIONED TO THE NUMBER OF UNIQUE VALUES PLUS 1
C
C          DIMENSION XVAL(4),FREQ(4),CDF(5)
C
C          C THE FOLLOWING LINE OF CODE IS SUPPLIED BY THE USER
C          C SVAL MUST BE DIMENSIONED TO THE NUMBER OF SAMPLE VALUES AND
C          C EDF MUST BE DIMENSIONED TO THE NUMBER OF SAMPLE VALUES PLUS 1
C
C          DIMENSION SVAL(8),EDF(9)
C
C          C THE FOLLOWING FUNCTION DEFINITION IS REQUIRED BY USRDST
C
C          LOC(I,J)=(J-1)*N+I
C
C          C BRANCH TO STATEMENT 5 IF THE EMPIRICAL DISTRIBUTION IS REQUESTED
C
C          IF(J.EQ.7)GO TO 5
C

```

```

C READ IN THE VALUES FOR THE DISCRETE PROBABILITY FUNCTION
C NP IS THE NUMBER OF UNIQUE VALUES OF THE RANDOM VARIABLE
C XVAL(K) IS THE KTH UNIQUE VALUE OF THE RANDOM VARIABLE
C FREQ(K) IS THE PROBABILITY ASSOCIATED WITH THE KTH UNIQUE VALUE
C NOTE THAT THE READ STATEMENT MUST BE OF THE FORM READ(7,*)....
C
      READ(7,*)NP
      DO 1 K=1,NP
1     READ(7,*)XVAL(K),FREQ(K)
C
C CONSTRUCT THE CUMULATIVE DISTRIBUTION FUNCTION
C
      CDF(1)=0.0
      DO 2 K=1,NP
2     CDF(K+1)=CDF(K)+FREQ(K)
C
C SET THE STARTING POINT (STRPT) EQUAL TO ZERO AND THE PROBABILITY
C INCREMENT (PROBINC) EQUAL TO 1/N FOR A LHS WHERE N IS THE SAMPLE SIZE
C
      STRPT=0.0
      PROBINC=1.0/FLOAT(N)
C
C IF A RANDOM SAMPLE HAS BEEN SPECIFIED IN THE PARAMETER LIST THEN THE
C ARGUMENT IRS HAS BEEN SET EQUAL TO 1 IN THE MAIN PROGRAM, HENCE THE
C PROBABILITY INCREMENT IS SET EQUAL TO 1 SO THAT ALL OBSERVATIONS ARE
C SELECTED BY USING THE INTERVAL (0,1)
C
      IF(IRS.EQ.1)PROBINC=1.0
C
C THIS LOOP WILL OBTAIN THE N SAMPLE VALUES
C
      DO 4 I=1,N
C
C R IS A RANDOMLY SELECTED POINT IN THE CURRENT SUBINTERVAL OBTAINED
C BY USING THE RANDOM NUMBER GENERATOR RAN
C
      R=STRPT+PROBINC*RAN(ISEED)
C
C THIS LOOP WILL SELECT THE SPECIFIC VALUE OF THE RANDOM VARIABLE
C CORRESPONDING TO R THROUGH THE INVERSE CUMULATIVE DISTRIBUTION
C FUNCTION THESE VALUES ARE STORED IN THE VECTOR X THROUGH THE
C USE OF THE LOC FUNCTION
C
      DO 3 K=1,NP
          IF(R.GE.CDF(K).AND.R.LT.CDF(K+1))X(LOC(I,J))=XVAL(K)
3     CONTINUE
C
C RESET THE STARTING POINT TO THE BEGINNING OF THE NEXT SUBINTERVAL
C UNLESS A RANDOM SAMPLE HAS BEEN SPECIFIED
C
      IF(IRS.NE.1)STRPT=STRPT+PROBINC
4     CONTINUE
      RETURN
5     CONTINUE
C
C READ IN THE SAMPLE SIZE NP AND THE SAMPLE VALUES
C NOTE THAT THE READ STATEMENT MUST BE OF THE FORM READ(7,*)....
C
      READ(7,*)NP,(SVAL(K),K=1,NP)
C
C CONSTRUCT THE EMPIRICAL DISTRIBUTION FUNCTION
C
      STEP=1.0/FLOAT(NP)
      EDF(1)=0.0
      DO 6 K=1,NP
6     EDF(K+1)=STEP*FLOAT(K)

```

```

C
C SET THE STARTING POINT (STRPT) EQUAL TO ZERO AND THE PROBABILITY
C INCREMENT (PROBINC) EQUAL TO 1/N FOR A LHS WHERE N IS THE SAMPLE SIZE
C
      STRPT=0.0
      PROBINC=1.0/FLOAT(N)
C
C IF A RANDOM SAMPLE HAS BEEN SPECIFIED IN THE PARAMETER LIST THEN THE
C ARGUMENT IRS HAS BEEN SET EQUAL TO 1 IN THE MAIN PROGRAM, HENCE THE
C PROBABILITY INCREMENT IS SET EQUAL TO 1 SO THAT ALL OBSERVATIONS ARE
C SELECTED BY USING THE INTERVAL (0,1)
C
      IF(IRS.EQ.1)PROBINC=1.0
C
C THIS LOOP WILL OBTAIN THE N SAMPLE VALUES
C
      DO 8 I=1,N
C
C R IS A RANDOMLY SELECTED POINT IN THE CURRENT SUBINTERVAL OBTAINED
C BY USING THE RANDOM NUMBER GENERATOR RAN
C
      R=STRPT+PROBINC*RAN(ISEED)
C
C THIS LOOP WILL SELECT THE SPECIFIC SAMPLE VALUE CORRESPONDING
C TO R THROUGH THE INVERSE EMPIRICAL DISTRIBUTION FUNCTION
C THESE VALUES ARE STORED IN THE VECTOR X THROUGH THE USE OF THE
C LOC FUNCTION
C
      DO 7 K=1,NP
          IF(R.GE.EDF(K).AND.R.LT.EDF(K+1))X(LOC(I,J))=SVAL(K)
7      CONTINUE
C
C RESET THE STARTING POINT TO THE BEGINNING OF THE NEXT SUBINTERVAL
C UNLESS A RANDOM SAMPLE HAS BEEN SPECIFIED
C
      IF(IRS.NE.1)STRPT=STRPT+PROBINC
8      CONTINUE
      RETURN
      END
C
      SUBROUTINE LENTRM (STR, K)
C
C Trim a string by removing all trailing blanks.
C
      CHARACTER*( *) STR
      INTEGER*4 K
C
      IL = LEN (STR)
      K = 1
      DO 1000 I = 1, IL
          IC = ICHAR (STR (I:I))
          IF (IC .EQ. 32) GO TO 1010
          K = K + 1
1000 CONTINUE
1010 CONTINUE
      K = K - 1
      RETURN
      END
C
C
C
      SUBROUTINE UPCASE (STR)
C
C Convert a mixed string to all uppercase.
C
      CHARACTER*( *) STR
C

```

```

      IL = LEN (STR)
      DO 1000 I = 1, IL
        IC = ICHAR (STR (I:I))
        IF (IC .GE. 97 .AND. IC .LE. 122) STR (I:I) = CHAR (IC - 32)
1000 CONTINUE
      RETURN
      END

C
C
C-----
C      FUNCTION RAN (SEED)
C
C      IMPLICIT DOUBLE PRECISION (A - H, O - Z)
C      IMPLICIT INTEGER*4 (I - N)
C
C      DIMENSION VV (N)
C      DIMENSION VV (1)
C      COMMON /MRANDC/ ITAB (55), N1, N2, NSEED
C      INTEGER*4 RN1, SEED
C
C      MARSAGLIAS random number generator.  This random number generator
C      generates random numbers in ]0,1[.  Note that if the seed value is
C      zero on the first call, a default value of 5731 will be used in a
C      linear congruential generator to generate 55 odd integers for the
C      array 'ITAB()'.  These values are preserved by the common statement,
C      so that they may be used in subsequent calls by setting the seed to
C      zero.  If the value of 'SEED' is greater than zero in a call to the
C      subroutine, then the array 'ITAB' will be initialized and a new seed
C      value returned by the subroutine.  Best results are obtained by
C      making the initial call with a seed of your choice and then setting
C      the seed to '0' for all subsequent calls.
C-----
C
C      Test to see if 55 odd integers must be generated.
C
      IF ((SEED .LE. 0) .AND. (NSEED .GE. 1)) GO TO 1000
      NSEED = SEED
      IF (SEED .LE. 0) NSEED = 5731
C
C      Generate 55 odd integers in array ITAB().
C
      DO 1010 I = 1, 55
        RN1 = MOD (NSEED*9069, 32768)
        IF (MOD (RN1, 2) .EQ. 0) RN1 = RN1 - 1
        ITAB (I) = RN1
        NSEED = RN1
1010 CONTINUE
C
C      Generate "N" random components for the vector "VV".
C
      N1 = 0
      N2 = 24
1000 CONTINUE
      N = 1
      DO 1020 I = 1, N
        ITAB (55 - N1) = MOD (ITAB (55 - N2)*ITAB (55 - N1), 32768)
        NSEED = ITAB (55 - N1)/2
        VV (I) = FLOAT (NSEED)/FLOAT (15384)
        N1 = MOD (N1 + 1, 55)
        N2 = MOD (N2 + 1, 55)
1020 CONTINUE
C
      IF (VV (1) .LE. 0.0 .OR. VV (1) .GE. 1.0) GO TO 1000
C

```

```
RAN = VV(1)
IF (SEED .GT. 0) SEED = NSEED
RETURN
END
```

C

```

        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
        IMPLICIT INTEGER*4 (I-N)
        CHARACTER*8 FNAME,DUMMY
        CHARACTER*12 FNAMES,FNAMES1,FNAMES2,FNAMES3
        REAL*4 SBTC,BTC1,BTC2,BTC3,BTC4,CPCT
        PARAMETER (NSTPS=501,NMC=200,NNOBS=30,NMC1=1,
2          NNOBS1=1,NSTPS1=1,NNPER=9)
        COMMON/SAVE/ VEL(NNOBS),FWIDTH(NNOBS),SBTC(NNOBS,NSTPS),
1          BTC1(NMC,NNOBS,NSTPS),BTC2(NMC1,NNOBS1,NSTPS1),
2          BTC3(NMC1,NNOBS1,NSTPS1),BTC4(NMC1,NNOBS1,NSTPS1),
3          CPCT(NNOBS,NSTPS,NNPER)
        DIMENSION FACTM(7)
        DIMENSION IOBS(NNOBS),IPER(NNOBS)

C
C
C      FACTM(1)=0.1932000
C      FACTM(2)=2.4759000
C      FACTM(3)=0.0350000
C      FACTM(4)=0.0014630
C      FACTM(5)=0.0000448
C      FACTM(6)=0.1785000
C      FACTM(7)=0.0994000
C      FACTM(1)=1.0
C      FACTM(2)=1.0
C      FACTM(3)=1.0
C      FACTM(4)=1.0
C      FACTM(5)=1.0
C      FACTM(6)=1.0
C      FACTM(7)=1.0
        WRITE (*,1000)
1000  FORMAT (' ENTER THE NAME OF SIMULATION FILE (eg. mc_tr001)',/,
        &      ' (DO NOT INCLUDE SUFFIX): ',$,
        READ (*,'(A8)') FNAME

C
C CLEAN UP AND TRIM THE FILE NAME
C
        CALL UPCASE (FNAME)
        CALL LENTRM (FNAME,K)

C
C CONCATONATE FILE SUFFIXES (PREDEFINED)
C
        FNAMES = FNAME (1:K)//'.DAT'
        FNAMES1 = FNAME (1:K)//'.BIN'
C      FNAMES2 = FNAME (1:K)//'.OUT'
C      OPEN (11,FILE = FNAMES)
C      OPEN (12,FILE = FNAMES1)
C      OPEN (13,FILE = FNAMES2)
C
        write (*,200)
        read (*,*) iconvrt
        WRITE (*,100)
        READ (*,*) NPR
        WRITE (*,101)
        READ (*,*) (IPER(IPR),IPR=1,NPR)
        WRITE (*,102)
        READ (*,*) MREM
        IF (MREM.GT.0) THEN
            WRITE (*,103)
            READ (*,*) IMREM
            FACTMR=FACTM(IMREM)
        ELSE
            FACTMR=1.0
        ENDIF
100  FORMAT (1X,' HOW MANY PERCENTILES ARE TO BE READ IN: ',\ )
101  FORMAT (1X,' WHAT PECIALITIES ARE TO BE PLOTTED (INTEGER): ',\ )
102  FORMAT (1X,' CONVERSION TO MREMS (TYPE: 0 FOR NO, 1 FOR YES) ',\ )

```

```

103 FORMAT (1X,' WHAT ISOTOPE ARE YOU SIMULATING?:      ',/,
1      1X,'          TYPE IN 1 FOR I-129,      ',/,
2      1X,'          TYPE IN 2 FOR Pu-239/240,    ',/,
3      1X,'          TYPE IN 3 FOR Cs-137,      ',/,
4      1X,'          TYPE IN 4 FOR C-14,          ',/,
5      1X,'          TYPE IN 5 FOR H-3,          ',/,
6      1X,'          TYPE IN 6 FOR U-238,        ',/,
7      1X,'          TYPE IN 7 FOR Sr-90:        ',\))
200 format (1x,' convert distance from meters to kilometers?  yes = 1
1 :')
      OPEN(UNIT=6,FILE=FNAMES1,FORM='UNFORMATTED',STATUS='OLD')
C      OPEN(UNIT=6,FILE=FNAMES2,ACCESS='SEQUENTIAL')
      OPEN(UNIT=7,FILE=FNAMES,ACCESS='SEQUENTIAL')
C
C      <----- MC_plxx Records
C
      READ (7,7000) DUMMY
      WRITE (*,7000) DUMMY
7000 FORMAT(A8)
      READ (7,*) MC
      WRITE (*,*) MC
      READ (7,*) IDUM1,DX,IDUM2,DT
      WRITE (*,*) IDUM1,DX,IDUM2,DT
      READ (7,7000) DUMMY
      WRITE (*,7000) DUMMY
      READ (7,*) NOB
      WRITE (*,*) NOB
      READ (7,*) (IOBS(I),I=1,NOB)
      WRITE (*,*) (IOBS(I),I=1,NOB)
      READ (7,*) IDUM,NSPEC
      WRITE (*,*) IDUM,NSPEC
      AMC=REAL(MC)
C
      MCP1=MC+1
      DO 20 IMCC= 1,MCP1
      DO 20 IS= 1,NSPEC
C
C      Need to write out the BTC for this species to an output file
C      for later processing.
C
      READ (6) DELT, DELX, NUMTT, NUMOBS, ISPEC, IMC, IOU20
      WRITE (*,*) DELT, DELX, NUMTT, NUMOBS, ISPEC, IMC, IOU20
      NADD=(NUMTT-1)/IOU20
      NUMT=1+NADD
      IF (NUMOBS.NE.NOB) THEN
        WRITE(*,6)
6        FORMAT(1X,'THE DATA AND BINARY FILES ARE INCOMPATIBLE')
        STOP
      ENDIF
      READ (6) (VEL (IO),FWIDTH (IO),IO = 1, NUMOBS)

      READ (6) ((SBTC (IO,IT), IT = 1,
1      NUMT), IO = 1, NUMOBS)
      IF(IMCC.GT.1) THEN
        IF(ISPEC.EQ.1) THEN
          DO 21 IT=1,NUMT
          DO 21 IO=1,NUMOBS
            BTC1(IMC,IO,IT)=SBTC(IO,IT)*FACTMR
21      CONTINUE
        ELSE IF(ISPEC.EQ.2) THEN
          DO 22 IT=1,NUMT
          DO 22 IO=1,NUMOBS
            BTC2(IMC,IO,IT)=SBTC(IO,IT)*FACTMR

```

```

22      CONTINUE
      ELSE IF( ISPEC.EQ.3) THEN

          DO 23 IT=1,NUMT
              DO 23 IO=1,NUMOBS
                  BTC3( IMC,IO,IT)=SBTC( IO,IT)*FACTMR

23      CONTINUE

          ELSE

              DO 24 IT=1,NUMT
                  DO 24 IO=1,NUMOBS
                      BTC4( IMC,IO,IT)=SBTC( IO,IT)*FACTMR
24      CONTINUE
          ENDIF
      ENDIF
20 CONTINUE
C
      DO 40 ISPEC=1,NSPEC
          IF( ISPEC.EQ.1) THEN
              DO 41 IT=1,NUMT
                  DO 41 IO=1,NUMOBS
                      CALL HPSORT(MC,BTC1(1,IO,IT))
                      DO 42 IPR=1,NPR
                          IF( IPER( IPR).NE.100) THEN
                              PERCNT=REAL( IPER( IPR))/100.0
                              POSIT=PERCNT*AMC
                              I1=INT(POSIT)
                              I2=I1+1
                              FRAC=POSIT-REAL( I1)
                              CPCT( IO,IT,IPR)=BTC1( I1,IO,IT)+FRAC*(BTC1( I2,IO,IT)-
1                      BTC1( I1,IO,IT))

                              ELSE
                                  CPCT( IO,IT,IPR)=BTC1( MC,IO,IT)
                              ENDIF
42      CONTINUE
41      CONTINUE
                      DO 51 IPR=1,NPR
                          FNAMES3=FNAME(1:K) //' .T'
                          INT1=48+IPR
                          INT2=48+ISPEC
                          FNAMES3(K+3:K+3)=CHAR( INT1)
                          FNAMES3(K+4:K+4)=CHAR( INT2)
                          OPEN(UNIT=8,FILE=FNAMES3,ACCESS='SEQUENTIAL' )
                          WRITE(8,16)
                          IT=0
                          DO 52 ITT=1,NUMTT,IOU20
                              IT=ITT+1
                              DO 52 IO=1,NUMOBS
                                  XOBS=REAL( IOBS( IO)-1)*DX
                                  if (iconvrt.gt.0) xobs = xobs/1000.
                                  TOBS=REAL( ITT-1)*DT
                                  WRITE(8,26) XOBS,TOBS,CPCT( IO,IT,IPR)
52      CONTINUE
                                  CLOSE(8)
51      CONTINUE
          ELSE IF( ISPEC.EQ.2) THEN
              DO 43 IT=1,NUMT
                  DO 43 IO=1,NUMOBS
                      CALL HPSORT(MC,BTC2(1,IO,IT))
                      DO 44 IPR=1,NPR
                          PERCNT=REAL( IPER( IPR))/100.0
                          POSIT=PERCNT*AMC
                          I1=INT(POSIT)

```



```

1      I2=I1+1
      FRAC=POSIT-REAL(I1)
      CPCT(IO,IT,IPR)=BTC2(I1,IO,IT)+FRAC*(BTC2(I2,IO,IT)-
44      BTC2(I1,IO,IT))
43      CONTINUE
      CONTINUE
      DO 53 IPR=1,NPR
          FNAMES3=FNAME(1:K) //' .T'
          INT1=48+IPR
          INT2=48+ISPEC
          FNAMES3(K+3:K+3)=CHAR(INT1)
          FNAMES3(K+4:K+4)=CHAR(INT2)
          OPEN(UNIT=8,FILE=FNAMES3,ACCESS='SEQUENTIAL')
          WRITE(8,16)
          IT=0
          DO 54 ITT=1,NUMTT,IOU20
              IT=IT+1
              DO 54 IO=1,NUMOBS
                  XOBS=REAL(IOBS(IO)-1)*DX
                  if (iconvrt.gt.0) xobs = xobs/1000.
                  TOBS=REAL(ITT-1)*DT
                  WRITE(8,26) XOBS,TOBS,CPCT(IO,IT,IPR)
54      CONTINUE
          CLOSE(8)
53      CONTINUE
      ELSE IF(ISPEC.EQ.3) THEN
          DO 45 IT=1,NUMT
          DO 45 IO=1,NUMOBS
              CALL HPSORT(MC,BTC3(1,IO,IT))
              DO 46 IPR=1,NPR
                  PERCNT=REAL(IPER(IPR))/100.0
                  POSIT=PERCNT*AMC
                  I1=INT(POSIT)
                  I2=I1+1
                  FRAC=POSIT-REAL(I1)
                  CPCT(IO,IT,IPR)=BTC3(I1,IO,IT)+FRAC*(BTC3(I2,IO,IT)-
1      BTC3(I1,IO,IT))
46      CONTINUE
45      CONTINUE
          DO 55 IPR=1,NPR
              FNAMES3=FNAME(1:K) //' .T'
              INT1=48+IPR
              INT2=48+ISPEC
              FNAMES3(K+3:K+3)=CHAR(INT1)
              FNAMES3(K+4:K+4)=CHAR(INT2)
              OPEN(UNIT=8,FILE=FNAMES3,ACCESS='SEQUENTIAL')
              WRITE(8,16)
              IT=0
              DO 56 ITT=1,NUMTT,IOU20
                  IT=IT+1
                  DO 56 IO=1,NUMOBS
                      XOBS=REAL(IOBS(IO)-1)*DX
                      if (iconvrt.gt.0) xobs = xobs/1000.
                      TOBS=REAL(ITT-1)*DT
                      WRITE(8,26) XOBS,TOBS,CPCT(IO,IT,IPR)
56      CONTINUE
              CLOSE(8)
55      CONTINUE
      ELSE
          DO 47 IT=1,NUMT
          DO 47 IO=1,NUMOBS
              CALL HPSORT(MC,BTC4(1,IO,IT))
              DO 48 IPR=1,NPR
                  PERCNT=REAL(IPER(IPR))/100.0
                  POSIT=PERCNT*AMC
                  I1=INT(POSIT)

```

```

        I2=I1+1
        FRAC=POSIT-REAL(I1)
        CPCT(IO,IT,IPR)=BTC4(I1,IO,IT)+FRAC*(BTC4(I2,IO,IT)-
1          BTC4(I1,IO,IT))
48      CONTINUE
47      CONTINUE
        DO 57 IPR=1,NPR
            FNAMES3=FNAM(1:K)//'.T'
            INT1=48+IPR
            INT2=48+ISPEC
            FNAMES3(K+3:K+3)=CHAR(INT1)
            FNAMES3(K+4:K+4)=CHAR(INT2)
            OPEN(UNIT=8,FILE=FNAMES3,ACCESS='SEQUENTIAL')
            WRITE(8,16)
            IT=0
            DO 58 ITT=1,NUMTT,IOU20
                IT=ITT+1
                DO 58 IO=1,NUMOBS
                    XOBS=REAL(IOBS(IO)-1)*DX
                    if (iconvrt.gt.0) xobs = xobs/1000.
                    TOBS=REAL(ITT-1)*DT
                    WRITE(8,26) XOBS,TOBS,CPCT(IO,IT,IPR)
58          CONTINUE
            CLOSE(8)
57      CONTINUE
        ENDIF
40 CONTINUE
16 FORMAT('DISTANCE,TIME,CONCENTRATION')
26 FORMAT(F10.1,',',F10.2,',',1P,E15.8,0P)
        STOP
        END
C
        SUBROUTINE HPSORT(N,RA)
        DIMENSION RA(N)
        IF (N.LT.2) RETURN
        L=N/2+1
        IR=N
10 CONTINUE
        IF(L.GT.1) THEN
            L=L-1
            RRA=RA(L)
        ELSE
            RRA=RA(IR)
            RA(IR)=RA(1)
            IR=IR-1
            IF (IR.EQ.1) THEN
                RA(1)=RRA
                RETURN
            ENDIF
        ENDIF
        I=L
        J=L+L
20 IF (J.LE.IR) THEN
            IF (J.LT.IR) THEN
                IF (RA(J).LT.RA(J+1)) J=J+1
            ENDIF
            IF (RRA.LT.RA(J)) THEN
                RA(I)=RA(J)
                I=J
                J=J+J
            ELSE
                J=IR+1
            ENDIF
            GO TO 20
        ENDIF
        RA(I)=RRA

```

```

        GO TO 10
        END
C
C
        SUBROUTINE LENTRM (STR, K)
C
C Trim a string by removing all trailing blanks.
C
        CHARACTER*( *) STR
        INTEGER*4 K
C
        IL = LEN (STR)
        K = 1
        DO 1000 I = 1, IL
            IC = ICHAR (STR (I:I))
            IF (IC .EQ. 32) GO TO 1010
            K = K + 1
1000 CONTINUE
1010 CONTINUE
        K = K - 1
        RETURN
        END
C
C
        SUBROUTINE UPCASE (STR)
C
C Convert a mixed string to all uppercase.
C
        CHARACTER*( *) STR
C
        IL = LEN (STR)
        DO 1000 I = 1, IL
            IC = ICHAR (STR (I:I))
            IF (IC .GE. 97 .AND. IC .LE. 122) STR (I:I) = CHAR (IC - 32)
1000 CONTINUE
        RETURN
        END
C

```

```

C*****
C
C                      BTCHIST.FOR
C
C      WRITTEN BY KEN R. REHFELDT, GEOTRANS, INC
C
C      APRIL 17, 1996
C
C      VERSION 1.0
C
C      PROGRAM TO READ A BINARY BREAKTHROUGH CURVE FILE FROM MC_TRNSF
C      AND STATISTICALLY SUMMARIZE THE DATA INTO HISTOGRAM CATEGORIES.
C      ALL THE DATA WITHIN THE 30 YEARS SURROUNDING THE PEAK TRITIUM
C      ARE USED.
C
C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C      IMPLICIT INTEGER*4 (I-N)
C      REAL*4 C,ch,ra
C      PARAMETER (NUMOBS=30,NUMTIME=251,NUMSPEC=1,IMC =201)
C      PARAMETER (NUMYRS =251)
C      CHARACTER*8 FNAME
C      CHARACTER*12 FNAMES,FNAMES1,FNAMES2
C      COMMON /SAVE/ C(NUMOBS,NUMTIME,NUMSPEC),
C      &          SUMC(NUMOBS,NUMTIME,NUMSPEC),
C      &          VEL(NUMOBS),FWIDTH(NUMOBS),CH(NUMOBS,NUMYRS,IMC),
C      &          CMEAN(NUMOBS,NUMTIME,NUMSPEC),
C      &          ra(imc)
C      &          CH1(NUMOBS,NUMYRS,IMC)
C      DIMENSION mcbad(1000),JEND(30),H(51),HC(50),JBEG(30),JMAX(30)
C      dimension ra(imc)
C      WRITE (*,1000)
C 1000  FORMAT (' ENTER THE NAME OF THE BINARY BTC INPUT FILE',
C      &          '(eg. mc_pl001)',/,
C      &          '(DO NOT INCLUDE SUFFIX): ',$,)
C      READ (*,'(A8)') FNAME
C      WRITE (*,1002)
C 1002  FORMAT (' LOWER BOUND CUTOFF VALUE? : ',$,)
C      READ (*,*) CUTOFF
C      write (*,1003)
C 1003  format (' mean (0) or median (1) ? : ',$,)
C      read (*,*) nmean
C      OPEN (31,FILE='HISTOGRM.DAT')
C      READ (31,*) NUMBINS
C      DO 8 K=1,NUMBINS+1
C      READ (31,*) H(K)
C      8  CONTINUE
C
C      CLEAN UP AND TRIM THE FILE NAME
C
C      CALL UPCASE (FNAME)
C      CALL LENTRM (FNAME,K)
C
C      CONCATONATE FILE SUFFIXES (PREDEFINED)
C
C      FNAMES = FNAME (1:K)//'.BIN'
C      FNAMES1 = FNAME (1:K)//'.DAT'
C      FNAMES2 = FNAME(1:K)//'.HIS'
C      OPEN (14,FILE = FNAMES2)
C      WRITE (14,200)
C      WRITE (14,204) (H(K),K=1,NUMBINS)
C      WRITE (14,204) (H(K),K=2,NUMBINS+1)
C      OPEN (11,FILE = FNAMES, FORM = 'UNFORMATTED')
C      OPEN (12,FILE = FNAMES1)
C      OPEN (9,FILE='HIST.OUT')
C

```

```

C READ THREE DUMMY LINES
C
      READ (12,'(A8)') D
      READ (12,*) MC
      READ (12,*) D1,d2,numt,d4
      READ (12,'(A8)') D
      READ (12,*) D
      READ (12,*) D
      READ (12,*) D,NSPEC

C
C add input to exclude bad monte carlo runs
C
      nmb = 0
      open (26,file='btcstats.bad')
      do 16 j=1,mc
        read (26,*,end=15) mcbad(j)
        nmb = nmb + 1
16      continue

C
C CALL THE SUBROUTINE TO READ THE INFORMATION
C
      MGOOD = 0
15      DO 10 M=1,MC+1
        READ (11) DELT,DELX,NUMTT,NOB,ISPEC,IMC,nskip
        NADD = (NUMTT-1)/NSKIP
        NUMT = 1 + NADD
        WRITE (*,'(I5,2F8.2,4I6)') M,DELT,DELX,NUMT,NOB,ISPEC,IMC,nskip
C ZERO THE ARRAY FOR THE NEXT SET
        DO 11 N1 = 1,NOB
          DO 12 N2 = 1,NUMT
            C(N1,N2,ISPEC) = 0.0
12          CONTINUE
11          CONTINUE
C      CALL INPUT (C,NUMT,NOB,NSPEC,ISPEC,nskip)
        READ(11) (VEL(n),FWIDTH(n),n=1,NOB)
        READ(11) ((C(IO,IT,ISPEC),IT=1,NUMT),IO=1,NOB)
        if (m.eq.1) go to 10
C WRITE THE DATA INTO A SMALLER DATA SET FOR PROCESSING
        do 18 mb = 1,nmb
          if (m.eq.mcbad(mb)) go to 10
18        continue
        MGOOD = MGOOD + 1
        N5 = 0
        DO 19 N3 = 1,NUMT
          N5 = N5 + 1
        DO 17 N4 = 1,NOB
          CH(N4,N5,MGOOD) = C(N4,N3,ISPEC)
C          IF (N4.EQ.11.AND.N5.EQ.20) WRITE (9,*) CH(N4,N5,MGOOD)
C          IF(CH(N4,N5,MGOOD).LT.CUTOFF) CH(N4,N5,MGOOD)= CUTOFF
C          ch1(N4,N5,MGOOD) = ch(N4,N5,MGOOD)
17        CONTINUE
19        CONTINUE
        DO 20 I=1,NOB
          DO 30 J=1,numt

C
C      IF (C(I,J,ISPEC).LT.CUTOFF) C(I,J,ISPEC)=CUTOFF
C      SUMC(I,J,ISPEC) = SUMC(I,J,ISPEC) + (C(I,J,ISPEC))
30        CONTINUE
300      FORMAT (10E12.2)
20        CONTINUE
10        CONTINUE

C
C CALCULATE THE STATISTICS
C
      nval = mc - nmb
      if (nmean) 86,86,88

```

```

86      DO 40 I=1,NOB
          JCOUNT = 0
          CMAX = 0.0
          DO 50 J=1,numt
              JCOUNT = JCOUNT + 1
              CMEAN(I,J,ISPEC) = SUMC(I,J,ISPEC)/nval
              IF (CMEAN(I,J,ISPEC).GT.CMAX) THEN
                  CMAX=CMEAN(I,J,ISPEC)
                  JM = JCOUNT
              END IF
50      CONTINUE
          JMAX(I) = JM
          write (*,*) ' jmax(i) and cmax ',jmax(i), cmax
40      CONTINUE
          go to 87
88      DO 42 IO=1,NOB
          cmax = 0.0
          jcount = 0
          DO 41 IT=1,NUMT
              jcount = jcount + 1
              do 92, jk = 1,nval
                  ra(jk) = ch(io,it,jk)
92      continue
                  CALL HPSORT(nval,ra)
                  IF (io.EQ.11.AND.it.EQ.20) WRITE (9,*) (ra(mg),
&                  mg=1,nval)
c          DO 42 IPR=1,NPR
c          IF(IPER(IPR).NE.100) THEN
              PERCNT=50./100.0
              POSIT=PERCNT*real(nval)
              I1=INT(POSIT)
              I2=I1+1
              FRAC=POSIT-REAL(I1)
              CPCT=ra(i1)+FRAC*(ra(i2)-
1              ra(i1))
              if (cpct.gt.cmax) then
                  cmax = cpct
                  jm = JCOUNT
              end if
c          ELSE
c          CPCT(IO,IT,IPR)=BTC1(MC,IO,IT)
c          ENDIF
c          42      CONTINUE
c          41      CONTINUE
              jmax(io) = jm
              write (*,*) ' jmax and cmax ',jmax(io), cmax
          42      continue
C
C CALCULATE THE HISTORGRAMS FOR THE 30 YEARS SURROUNDING THE PEAK
C
87      DO 541 I=NOB,1,-1
          DO 560 L = 1, NUMBINS
              HC(L) = 0.0
560      CONTINUE
C
C DETERMINE THE STARTING AND ENDING TIME FOR THE 30 YEAR SPAN
C
          IF (JMAX(I).LE.6) THEN
              JBEG(I) = 1
              JEND(I) = 31
              GO TO 119
          END IF
          IF (JMAX(I).GE.NUMT-25) THEN
              JBEG(I) = NUMT-30
              JEND(I) = NUMT
              GO TO 119

```

```

      END IF
      JBEG(I) = JMAX(I) - 5
      JEND(I) = JMAX(I) + 25
119  NTOTAL = 0
C
C PUT VALUES INTO THE CORRECT HISTOGRAM BINS
C
      DO 542 J = JBEG(I),JEND(I)
      DO 543 K = 1,NVAL
      DO 544 L = 1,NUMBINS
      IF (CH(I,J,K).GE.H(L).AND.CH(I,J,K).LT.H(L+1).or.
&      ch(i,j,k).lt.h(l).and.ch(i,j,k).ge.h(l+1)) THEN
      HC(L) = HC(L) + 1
      NTOTAL = NTOTAL + 1
      GO TO 543
      END IF
544  CONTINUE
543  CONTINUE
542  CONTINUE
C
C DETERMINE THE HISTOGRAM PERCENTAGES
C
      SUMHC = 0
      DO 545 L = 1,NUMBINS
      HC(L) = HC(L)/NTOTAL
      SUMHC = SUMHC + HC(L)
545  CONTINUE
C
C WRITE THE HISTOGRAMS FOR EACH RECEPTOR IN ONE FILE
C
      TMIN = (JBEG(I) - 1)*NSKIP*DELT
      TMAX = (JEND(I) - 1)*NSKIP*DELT
      WRITE (14,205) I,SUMHC,TMIN,TMAX,(HC(L),L=1,NUMBINS)
541  CONTINUE
204  FORMAT (35X,50(1X,G10.2))
205  FORMAT (I5,2X,F8.4,2X,F8.1,2X,F8.1,50(1X,F10.4))
200  FORMAT ( ' HISTOGRAMS FOR EACH OBSERVATION LOCATION NUMBER  ' )
      STOP
      END
C
C
C SUBROUTINE LENTRM (STR, K)
C Trim a string by removing all trailing blanks.
C
      CHARACTER*( *) STR
      INTEGER*4 K
C
      IL = LEN (STR)
      K = 1
      DO 1000 I = 1, IL
      IC = ICHAR (STR (I:I))
      IF (IC .EQ. 32) GO TO 1010
      K = K + 1
1000 CONTINUE
1010 CONTINUE
      K = K - 1
      RETURN
      END
C
C
C SUBROUTINE UPCASE (STR)
C Convert a mixed string to all uppercase.
C

```

	CHARACTER*(*) STR	UPCA0070
C		UPCA0080
	IL = LEN (STR)	UPCA0090
	DO 1000 I = 1, IL	UPCA0100
	IC = ICHAR (STR (I:I))	UPCA0110
	IF (IC .GE. 97 .AND. IC .LE. 122) STR (I:I) = CHAR (IC - 32)	UPCA0120
1000	CONTINUE	UPCA0130
	RETURN	UPCA0140
	END	UPCA0150
C		
C	-----	
C		
c	SUBROUTINE INPUT (C,NUMT,NUMOBS,NUMSPEC,ISPEC,nskip)	
c	IMPLICIT DOUBLE PRECISION (A-H,O-Z)	
c	IMPLICIT INTEGER*4 (I - N)	
c	REAL*4 C	
c	DIMENSION VEL(NUMOBS), FWIDTH(NUMOBS),C(NUMOBS,NUMT,NUMSPEC)	
c	READ(11) (VEL(M),FWIDTH(M),M=1,NUMOBS)	
c	READ(11) ((C(IO,IT,ISPEC),IT=1,NUMT),IO=1,NUMOBS)	
C	DO 10 I=1,NUMOBS	
C	WRITE (22,300) (C(I,J,ISPEC),J=1,NUMT)	
C 300	FORMAT (10E12.2)	
C 10	CONTINUE	
c	RETURN	
c	END	
	SUBROUTINE HPSORT(N,RA)	
	DIMENSION RA(N)	
C	write (*,*) ' n = ',n	
	IF (N.LT.2) RETURN	
	L=N/2+1	
	IR=N	
10	CONTINUE	
	IF(L.GT.1) THEN	
	L=L-1	
	RRA=RA(L)	
	ELSE	
	RRA=RA(IR)	
	RA(IR)=RA(1)	
	IR=IR-1	
	IF (IR.EQ.1) THEN	
	RA(1)=RRA	
	RETURN	
	ENDIF	
	ENDIF	
	I=L	
	J=L+L	
20	IF (J.LE.IR) THEN	
	IF (J.LT.IR) THEN	
	IF (RA(J).LT.RA(J+1)) J=J+1	
	ENDIF	
	IF (RRA.LT.RA(J)) THEN	
	RA(I)=RA(J)	
	I=J	
	J=J+J	
	ELSE	
	J=IR+1	
	ENDIF	
	GO TO 20	
	ENDIF	
	RA(I)=RRA	
	GO TO 10	
	END	

Appendix D

Summary of Travel Times and Travel Distances for all Starting Locations

Appendix D, presented two parts, contains the summary of the pathlines from each of the starting locations identified below the water table for a fixed distance, and again for a fixed time. In the case of fixed distance, the travel time is based on advective velocity and is presented for selected distances downgradient from the shots. The fixed time summaries present the travel distances for selected points in time also based on advective velocities only. The order of locations is the same as presented in the Nuclear Test Database in Appendix B with the exception that only shots with a location below the water table are in this appendix.

There are seven columns associated with each location for the fixed distance case:

Shotname-	the detonation name as provided by USDOE (1994)
ID-	a numerical counter assigned for internal tracking
SLOC-	designates if the location is the bottom (B), working point (WP), or top (T) of a shot cavity
5000-	the travel time in years from the starting location to 5000 meters
10000-	the travel time in years from the starting location to 10000 meters
20000-	the travel time in years from the starting location to 20000 meters
40000-	the travel time in years from the starting location to 40000 meters
70000-	the travel time in years from the starting location to 70000 meters

There are five columns associated with each location for the fixed time case:

Shotname-	the detonation name as provided by USDOE (1994)
ID-	a numerical counter assigned for internal tracking
SLOC-	designates if the location is the bottom (B), working point (WP), or top (T) of a shot cavity
50-	the travel distance (meters) from the starting location after 50 years travel time
100-	the travel distance (meters) from the starting location after 100 years travel time
150-	the travel distance (meters) from the starting location after 150 years travel time
200-	the travel distance (meters) from the starting location after 200 years travel time

TRAVEL TIME(yrs) AT FIXED DISTANCES(m)

SHOTNAME	ID	SLOC	5000.	10000.	20000.	40000.	70000.
COMMODORE	1	B	418.	560.	1033.	1307.	1418.
COMMODORE	2	WP	1246.	1530.	2113.	2407.	2536.
COMMODORE	3	T	2372.	2494.	2598.	2683.	2745.
CLARKSMOBILE	4	B	769.	854.	917.	998.	1058.
KNOX	5	B	732.	844.	1887.	2181.	2262.
KNOX	6	WP	413.	525.	592.	687.	746.
KNOX	7	T	335.	372.	404.	468.	528.
ILDRIM	8	B	443.	528.	584.	662.	722.
CALABASH	9	B	207.	511.	727.	849.	916.
CALABASH	10	WP	770.	920.	1054.	1151.	1213.
STANYAN	11	B	361.	659.	826.	961.	1038.
STANYAN	12	WP	152.	272.	367.	446.	508.
PORTMANTEAU	13	B	308.	755.	1054.	1899.	2093.
PORTMANTEAU	14	WP	215.	368.	693.	907.	1007.
YANNIGAN	15	B	526.	614.	688.	770.	832.
YANNIGAN	16	B	830.	920.	997.	1075.	1137.
FLASK	17	B	766.	885.	956.	1062.	1121.
FLASK	18	WP	408.	495.	563.	644.	704.
HAREBELL	19	B	91.	156.	274.	371.	434.
STARWORT	20	B	791.	893.	997.	1098.	1160.
STARWORT	21	WP	629.	710.	769.	842.	903.
MINIATA	22	B	326.	472.	513.	619.	676.
MINIATA	23	WP	235.	326.	362.	433.	490.
PORTULACA	24	B	228.	443.	564.	705.	770.
STODDARD	25	B	1093.	1701.	2014.	2231.	2301.
GORBEA	26	B	197.	295.	499.	622.	688.
CREW	27	B	123.	250.	483.	595.	664.
CREW	28	WP	123.	250.	483.	595.	664.
CREW	29	T	123.	250.	483.	595.	664.
COFFER	30	B	141.	321.	611.	748.	825.
HUTCH	31	B	1842.	2124.	2316.	2455.	2520.
CARPETBAG	32	B	1374.	1605.	2024.	2679.	2804.
CARPETBAG	33	WP	1558.	1803.	2220.	2351.	2417.
FLAX	34	B	1354.	1653.	1966.	2102.	2166.
FLAX	35	WP	1354.	1653.	1966.	2102.	2166.
FLAX	36	T	1354.	1653.	1966.	2102.	2166.
CABRILLO	37	B	2357.	2603.	3011.	3436.	3557.
CABRILLO	38	WP	139.	296.	535.	655.	728.
FALLON	39	B	841.	1005.	1189.	1294.	1356.
EDAM	40	B	72.	274.	506.	598.	662.
BANON	41	B	1227.	1520.	1804.	1954.	2032.
COULOMMIERS	42	B	73.	119.	217.	300.	362.
CHIBERTA	43	B	217.	275.	685.	901.	982.
CHIBERTA	44	WP	3028.	3235.	3973.	4313.	4452.
CHIBERTA	45	T	978.	1062.	1124.	1196.	1257.
MARSILLY	46	B	939.	1071.	2209.	2516.	2600.
MARSILLY	47	WP	8032.	8191.	9141.	9505.	9591.
MARSILLY	48	T	854.	939.	1008.	1085.	1146.
REBLOCHON	49	B	4343.	4471.	5302.	5656.	5741.
REBLOCHON	50	WP	2424.	2851.	3002.	3134.	3199.
REBLOCHON	51	T	1043.	1132.	1203.	1276.	1337.
KLOSTER	52	B	118.	216.	360.	472.	536.
NESSSEL	53	B	1275.	1364.	1439.	1517.	1579.
AKAVI	54	B	83.	175.	285.	371.	433.
ROMANO	55	B	255.	373.	527.	631.	698.
FARALLONES	56	B	2328.	2556.	2973.	3520.	3648.
FARALLONES	57	WP	1832.	2099.	2378.	2516.	2581.

QUARGEL	58	B	1344.	1623.	1995.	2125.	2192.
QUARGEL	59	WP	42.	121.	287.	378.	441.
FAJY	60	B	969.	1212.	1375.	1516.	1582.
PANAMINT	61	B	1134.	1221.	1290.	1362.	1424.
BORATE	62	B	536.	828.	1082.	1215.	1280.
SHELLBOURNE	63	B	2476.	2616.	2742.	2836.	2898.
INGOT	64	B	90.	200.	321.	414.	475.
METROPOLIS	65	B	495.	629.	701.	835.	894.
CREPE	66	B	279.	378.	415.	491.	549.
DUMONT	67	B	540.	635.	1798.	2048.	2130.
DUMONT	68	WP	842.	1039.	1118.	1253.	1312.
DUMONT	69	T	343.	452.	510.	595.	654.
AGILE	70	B	932.	1239.	2219.	2975.	3137.
AGILE	71	WP	2756.	2996.	3706.	4297.	4418.
AGILE	72	T	1356.	1439.	1510.	1585.	1647.
LANPHER	73	B	496.	604.	1132.	1334.	1416.
LANPHER	74	WP	1745.	1799.	2691.	3036.	3169.
LANPHER	75	T	582.	690.	771.	869.	928.
AARDVARK	76	B	1431.	1614.	1677.	1781.	1840.
WAGTAIL	77	B	669.	1091.	1987.	2191.	2318.
WAGTAIL	78	WP	875.	1110.	1170.	1289.	1416.
WAGTAIL	79	T	1161.	1392.	1449.	1555.	1617.
HAYMAKER	80	B	629.	973.	1196.	1272.	2493.
MERRIMAC	81	B	537.	699.	779.	855.	1047.
BELEN	82	B	464.	715.	790.	884.	1254.
PUCE	83	B	334.	502.	561.	659.	746.
BILBY	84	B	114.	254.	862.	989.	2613.
BILBY	85	WP	359.	456.	547.	615.	890.
BILBY	86	T	1320.	1479.	1560.	1641.	1807.
CANVASBACK	87	B	262.	281.	330.	402.	488.
KESTREL	88	B	1054.	1120.	1389.	1455.	3062.
BUFF	89	B	648.	842.	897.	997.	1057.
MORRONES	90	B	1152.	1326.	1395.	1496.	1561.
WISE	91	B	392.	522.	615.	678.	1077.
PICCALILLI	92	B	48.	71.	176.	249.	813.
LAGUNA	93	B	1883.	2443.	2708.	2807.	4599.
TORTUGAS	94	B	1505.	2945.	4487.	4655.	4792.
TORTUGAS	95	WP	1864.	2128.	2184.	2294.	2363.
TORTUGAS	96	T	1763.	2049.	2105.	2212.	2276.
BONARDA	97	B	679.	903.	961.	1061.	1153.
ABEYAS	98	B	416.	442.	694.	753.	1780.
CUMARIN	99	B	20.	40.	100.	148.	577.
SCUPPER	100	B	1042.	1226.	1549.	1639.	3390.
ANGUS	101	B	1191.	1336.	1733.	1819.	3605.
PUYE	102	B	1173.	1401.	1742.	1828.	3539.
ALGODONES	103	B	1969.	2220.	2280.	2395.	2611.
ALGODONES	104	WP	873.	1102.	1165.	1262.	1387.
MONERO	105	B	54.	70.	113.	166.	266.
MONERO	106	WP	27.	43.	84.	185.	242.
MARSH	107	B	805.	893.	1167.	1231.	2685.
OFFSHORE	108	B	154.	187.	314.	368.	1261.
ALEMAN	109	B	2387.	2674.	2730.	2840.	2903.
ALEMAN	110	WP	1320.	1636.	1707.	1828.	1887.
BOUSCHET	111	B	1691.	1944.	2001.	2115.	2188.
BOUSCHET	112	WP	2043.	2330.	2386.	2493.	2555.
VAUGHN	113	B	805.	1091.	1150.	1261.	1394.
DUORO	114	B	2666.	3161.	3568.	3674.	5458.
KINIBITO	115	B	10482.	10934.	12834.	13124.	14237.
KINIBITO	116	WP	1312.	1586.	1661.	1766.	2259.
TAHOKA	117	B	6694.	7128.	8968.	9265.	10275.
TAHOKA	118	WP	2568.	2830.	2888.	3004.	3081.

TAHOKA	119	T	903.	1144.	1202.	1305.	1365.
LUBBOCK	120	B	1980.	2242.	2299.	2408.	2476.
STRAIT	121	B	61.	88.	241.	357.	471.
STRAIT	122	WP	240.	319.	1423.	1665.	1748.
STRAIT	123	T	442.	585.	625.	730.	786.
BURZET	124	B	1723.	1996.	2111.	2227.	2286.
MONTEREY	125	B	966.	1158.	1247.	1397.	1456.
TILCI	126	B	2033.	2362.	2497.	2612.	2671.
MANTECA	127	B	1827.	2111.	2233.	2356.	2415.
COSO	128	B	981.	1161.	1242.	1367.	1427.
COSO	129	B	705.	908.	1015.	1159.	1220.
BRETON	130	B	1521.	1793.	1914.	2029.	2088.
ROQUEFORT	131	B	201.	993.	1320.	1449.	1515.
BULLFROG	132	B	1508.	1785.	1894.	2008.	2067.
BRISTOL	133	B	798.	972.	1047.	1212.	1270.
ZAZA	134	B	205.	292.	1427.	1600.	1683.
ZAZA	135	WP	626.	884.	952.	1068.	1126.
ZAZA	136	T	377.	513.	583.	745.	802.
LATIR	137	B	237.	312.	1475.	1704.	1787.
LATIR	138	WP	394.	598.	646.	772.	830.
LATIR	139	T	266.	387.	426.	507.	564.
TOPGALLANT	140	B	398.	478.	1128.	1276.	1360.
TOPGALLANT	141	WP	1115.	1202.	2200.	2509.	2640.
TOPGALLANT	142	T	592.	751.	792.	908.	965.
TRANSOM	143	B	752.	998.	1048.	1173.	1231.
TRANSOM	144	WP	752.	998.	1048.	1173.	1231.
TRANSOM	145	T	752.	998.	1048.	1173.	1231.
ICEBERG	146	B	173.	287.	916.	1165.	1287.
ICEBERG	147	WP	353.	537.	583.	716.	773.
ICEBERG	148	T	257.	388.	437.	533.	590.
SCANTLING	149	B	290.	377.	824.	1044.	1124.
SCANTLING	150	WP	950.	1034.	1164.	1290.	1353.
SCANTLING	151	T	501.	648.	715.	855.	913.
GLENCOE	152	B	1430.	1694.	1798.	1915.	1976.
GLENCOE	153	WP	1504.	1796.	1897.	2010.	2069.
GLENCOE	154	T	1513.	1762.	1853.	1999.	2057.
JORNADA	155	B	303.	482.	1214.	1458.	1593.
JORNADA	156	WP	433.	650.	697.	830.	887.
JORNADA	157	T	439.	589.	664.	881.	941.
QUINELLA	158	B	760.	853.	958.	1083.	1147.
QUINELLA	159	WP	535.	698.	740.	872.	928.
HEARTS	160	B	426.	575.	630.	747.	810.
HEARTS	161	WP	759.	1006.	1056.	1181.	1239.
HEARTS	162	T	595.	743.	841.	1054.	1114.
TECHADO	163	B	290.	440.	483.	615.	671.
TECHADO	164	WP	234.	351.	391.	475.	532.
ROUSANNE	165	B	192.	311.	350.	444.	500.
ROUSANNE	166	WP	222.	309.	344.	413.	470.
CAPROCK	167	B	155.	232.	282.	404.	463.
CAPROCK	168	WP	302.	429.	469.	572.	627.
GASCON	169	B	250.	332.	381.	499.	557.
GASCON	170	WP	351.	490.	531.	664.	720.
DALHART	171	B	273.	413.	466.	585.	648.
DALHART	172	WP	607.	824.	871.	1006.	1064.
DALHART	173	T	499.	656.	736.	956.	1020.
WISHBONE	174	B	492851.	500755.	500981.	501008.	-999.
DILUTED WATER	175	B	18452.	24004.	24079.	24105.	-999.
CAMBRIC	176	B	11503.	11846.	11891.	11917.	-999.
CAMBRIC	177	WP	10656.	11001.	11047.	11073.	-999.
CAMBRIC	178	T	5638.	6003.	6046.	6073.	-999.
DERRINGER	179	B	2863.	4749.	5005.	5032.	6403.

MILKSHAKE	180	B	15452.	22477.	23017.	23044.	24305.
ESCABOSA	181	B	529.	726.	785.	900.	975.
ESCABOSA	182	WP	376.	520.	589.	694.	752.
MIERA	183	B	32.	46.	85.	141.	197.
STRAKE	184	B	18.	32.	70.	161.	222.
POTRILLO	185	B	100.	130.	175.	251.	307.
POTRILLO	186	WP	94.	156.	200.	269.	321.
OBAR	187	B	21.	35.	72.	130.	186.
MIZZEN	188	B	28.	236.	353.	477.	561.
MIZZEN	189	WP	179.	275.	319.	432.	489.
KEELSON	190	B	246.	2207.	3275.	3419.	3680.
KEELSON	191	WP	678.	876.	932.	1033.	1094.
RUDDER	192	B	80.	103.	151.	264.	320.
RUDDER	193	WP	124.	213.	255.	345.	401.
ESROM	194	B	17.	59.	110.	235.	296.
ESROM	195	WP	80.	161.	199.	281.	337.
DRAUGHTS	196	B	322.	508.	612.	732.	792.
BULKHEAD	197	B	85.	113.	157.	222.	279.
BULKHEAD	198	WP	85.	133.	174.	239.	291.
BILLET	199	B	540.	753.	810.	927.	1002.
BILLET	200	WP	998.	1246.	1300.	1405.	1464.
BILLET	201	T	445.	644.	715.	922.	987.
CREWLINE	202	B	280.	430.	476.	588.	646.
CREWLINE	203	WP	363.	527.	579.	774.	832.
SANDREEF	204	B	9479.	9727.	10832.	11179.	11891.
SANDREEF	205	WP	801.	1119.	1177.	1291.	1355.
SANDREEF	206	T	2375.	2673.	2730.	2839.	2903.
RUMMY	207	B	431.	610.	667.	783.	852.
RUMMY	208	WP	688.	959.	1011.	1122.	1181.
RUMMY	209	T	1322.	1520.	1568.	1727.	1785.
LOWBALL	210	B	448.	665.	718.	823.	882.
LOWBALL	211	WP	1079.	1232.	1282.	1431.	1489.
BASEBALL	212	B	419.	608.	658.	768.	826.
BASEBALL	213	WP	938.	1115.	1163.	1334.	1393.
PALIZA	214	B	40.	54.	95.	172.	225.
PYRAMID	215	B	71.	99.	141.	199.	255.
PYRAMID	216	WP	77.	106.	143.	209.	261.
TAJO	217	B	13.	26.	60.	152.	238.
MUNDO	218	B	14.	27.	61.	137.	202.
ATRISCO	219	B	24.	41.	88.	137.	194.
ATRISCO	220	WP	21.	35.	75.	158.	214.
BORREGO	221	B	309.	486.	534.	645.	703.
BORREGO	222	WP	455.	624.	690.	896.	955.
HERMOSA	223	B	54.	77.	125.	223.	278.
HERMOSA	224	WP	110.	194.	238.	327.	381.
TURQUOISE	225	B	1441.	1715.	1770.	1880.	1943.
TURQUOISE	226	WP	921.	1108.	1189.	1333.	1392.
MIDLAND	227	B	23.	36.	70.	170.	255.
TEXARKANA	228	B	17.	31.	71.	159.	207.
FLOYDADA	229	B	34.	44.	81.	304.	406.
PIRANHA	230	B	229.	320.	373.	494.	550.
PIRANHA	231	WP	304.	452.	533.	680.	753.
BRONZE	232	B	107.	148.	189.	258.	313.
CHARCOAL	233	B	132.	195.	255.	342.	396.
LAMPBLACK	234	B	51.	65.	108.	160.	214.
LAMPBLACK	235	WP	67.	79.	116.	230.	295.
LIME	236	B	17.	27.	61.	165.	250.
TAN	237	B	908.	1129.	1186.	1295.	1364.
TAN	238	WP	934.	1093.	1157.	1272.	1330.
MICKEY	239	B	12.	27.	67.	187.	252.
BOURBON	240	B	10.	25.	59.	180.	241.

DAIQUIRI	241	B	112.	199.	238.	316.	371.
DAIQUIRI	242	WP	110.	178.	216.	283.	337.
BLENTON	243	B	70.	85.	130.	183.	237.
BLENTON	244	WP	135.	149.	191.	275.	330.
SHAPER	245	B	120.	142.	190.	273.	326.
SHAPER	246	WP	159.	184.	229.	287.	338.
GRAPE A	247	B	13.	27.	70.	127.	184.
GRAPE A	248	WP	34.	45.	85.	296.	386.
THISTLE	249	B	11.	25.	66.	122.	187.
THISTLE	250	WP	49.	60.	96.	311.	414.
COBBLER	251	B	206.	296.	344.	452.	509.
COBBLER	252	WP	206.	289.	336.	453.	508.
COBBLER	253	T	240.	384.	428.	614.	671.
GRAPE B	254	B	65.	81.	121.	176.	231.
TORRIDO	255	B	12.	27.	68.	171.	223.
ARTESIA	256	B	13.	22.	54.	219.	310.
TIJERAS	257	B	51.	68.	112.	169.	222.
TIJERAS	258	WP	53.	69.	113.	187.	235.
OSCURO	259	B	310.	455.	507.	612.	671.
OSCURO	260	WP	397.	580.	641.	857.	932.
COTTAGE	261	B	56.	183.	254.	513.	594.
FRISCO	262	B	101.	217.	434.	673.	755.
MISSISSIPPI	263	B	602.	724.	840.	932.	996.
FORE	264	B	339.	462.	581.	677.	744.
TERRINE	265	B	87.	111.	143.	212.	275.
NOGGIN	266	B	67.	180.	246.	342.	406.
NOGGIN	267	WP	308.	351.	383.	450.	510.
CUP	268	B	9.	25.	58.	126.	188.
YARD	269	B	22.	42.	77.	146.	207.
STACCATO	270	B	45.	60.	87.	178.	278.
CORNICE	271	B	9.	17.	46.	600.	1413.
TURF	272	B	41.	87.	144.	212.	273.
KLICKITAT	273	B	32.	41.	69.	141.	243.
CORDUROY	274	B	11.	24.	63.	169.	254.
CORDUROY	275	WP	10.	21.	53.	130.	193.
KANKAKEE	276	B	12.	24.	50.	143.	268.
STANLEY	277	B	33.	53.	82.	148.	207.
SHUFFLE	278	B	14.	24.	53.	205.	290.
PINSTRIFE	279	B	3131.	5724.	6431.	6464.	7659.
NEWPOINT	280	B	5951.	8263.	8682.	8711.	9968.
DIANA MOON	281	B	2884.	5844.	6467.	6495.	7710.
MINUTE STEAK	282	B	10394.	15437.	16179.	16207.	17413.
DIAGONAL LINE	283	B	7425.	11533.	12126.	12155.	13377.
MINI JADE	284	B	5557.	11201.	22063.	23060.	25004.
HUNTERS TROPH	285	B	4556.	4759.	6176.	6868.	8342.
SCOTCH	286	B	10568.	22854.	26400.	27665.	155870.
SCOTCH	287	WP	356.	1822.	4147.	5134.	96576.
SCOTCH	288	T	100.	534.	778.	894.	7608.
SHEEPSHEAD	289	B	33.	43.	48.	66.	3436.
TOWANDA	290	B	121.	272.	286.	526.	2019.
TIERRA	291	B	28.	153.	185.	213.	2504.
CHANCELLOR	292	B	62.	109.	124.	164.	4945.
NEBBIOLO	293	B	28.	36.	42.	72.	3434.
SERPA	294	B	761.	906.	1270.	1310.	-999.
HARZER	295	B	1106.	1167.	1243.	1415.	11319.
HOSTA	296	B	402.	1795.	1898.	1926.	-999.
LABQUARK	297	B	116.	188.	254.	300.	4558.
LOCKNEY	298	B	22.	27.	31.	72.	4194.
CYBAR	299	B	58.	116.	134.	184.	5776.
ALAMO	300	B	44.	60.	72.	105.	4764.
KEARSARGE	301	B	37.	111.	165.	209.	5693.

AMARILLO	302	B	66.	125.	145.	195.	4903.
HOUSTON	303	B	25.	30.	34.	65.	3828.
HALFBEAK	304	B	23.	137.	3212.	4885.	41767.
HALFBEAK	305	WP	15.	115.	806.	928.	5765.
HALFBEAK	306	T	474.	813.	907.	1048.	6739.
BEXAR	307	B	87.	156.	204.	249.	9790.
JUNCTION	308	B	56.	98.	113.	153.	3945.
RICKEY	309	B	56.	375.	403.	692.	15126.
CHARTREUSE	310	B	551.	1825.	1924.	1949.	-999.
CHARTREUSE	311	WP	134.	320.	379.	395.	-999.
MUENSTER	312	B	1020.	1723.	12118.	41593.	433303.
MUENSTER	313	WP	7316.	35205.	37630.	463712.	840710.
MUENSTER	314	T	6782.	14457.	19957.	21792.	85596.
INLET	315	B	516.	846.	887.	1102.	6603.
INLET	316	WP	32.	38.	47.	118.	1887.
ESTUARY	317	B	104.	431.	1230.	1313.	12382.
ESTUARY	318	WP	65.	142.	309.	345.	-999.
ESTUARY	319	T	262.	307.	382.	401.	-999.
SLED	320	B	286.	1387.	1412.	1788.	5465.
SLED	321	WP	49.	74.	82.	232.	1640.
STINGER	322	B	13.	112.	352.	580.	3749.
STINGER	323	WP	46.	249.	388.	408.	-999.
POOL	324	B	4582.	15307.	15496.	38099.	77138.
POOL	325	WP	397.	5782.	5850.	16079.	45701.
POOL	326	T	54.	230.	239.	488.	2191.
CAMEMBERT	327	B	357490.	357955.	360904.	362535.	1124553.
CAMEMBERT	328	WP	216882.	217226.	224006.	226841.	2137692.
CAMEMBERT	329	T	12442.	24152.	27538.	29081.	922337.
MAST	330	B	88.	322.	1248.	1751.	10359.
MAST	331	WP	28.	76.	303.	494.	3108.
MAST	332	T	111.	394.	476.	498.	-999.
ALMENDRO	333	B	9547.	61153.	61374.	1182536.	3507155.
ALMENDRO	334	WP	3905.	17212.	17304.	17648.	49572.
ALMENDRO	335	T	393.	576.	597.	1255.	7498.
BACKBEACH	336	B	28.	36.	45.	78.	2977.
PANIR	337	B	89.	203.	311.	394.	7275.
PANIR	338	WP	47.	71.	87.	117.	4519.
FONDUTTA	339	B	726.	792.	860.	1041.	14562.
BUTEO	340	B	16.	24.	27.	-999.	-999.
BUTEO	341	WP	16.	20.	22.	-999.	-999.
COLBY	342	B	3370.	5698.	12061.	19878.	39460.
COLBY	343	WP	826.	1925.	3191.	3800.	18654.
COLBY	344	T	323.	544.	804.	1560.	-999.
FARM	345	B	67.	104.	114.	167.	-999.
FARM	346	WP	27.	42.	46.	51.	-999.
COLWICK	347	B	34.	49.	53.	-999.	-999.
COLWICK	348	WP	18.	24.	26.	-999.	-999.
PEPATO	349	B	24.	59.	87.	-999.	-999.
PEPATO	350	WP	1.	3.	5.	-999.	-999.
TAFI	351	B	32.	48.	59.	-999.	-999.
TAFI	352	WP	1.	3.	5.	-999.	-999.
KASH	353	B	13.	40.	45.	-999.	-999.
KASH	354	WP	1.	2.	4.	-999.	-999.
MOLBO	355	B	2.	3.	6.	-999.	-999.
MOLBO	356	WP	1.	1.	4.	-999.	-999.
GIBNE	357	B	19.	27.	31.	-999.	-999.
JEFFERSON	358	B	12.	26.	29.	-999.	-999.
CABRA	359	B	2.	3.	6.	-999.	-999.
SALUT	360	B	2.	3.	6.	-999.	-999.
EGMONT	361	B	13.	24.	34.	-999.	-999.
KAPPELI	362	B	36.	68.	73.	95.	-999.

SERENA	363	B	21.	23.	31.	36.	-999.
GOLDSTONE	364	B	1.	2.	4.	-999.	-999.
BODIE	365	B	24.	51.	56.	-999.	-999.
DARWIN	366	B	26.	28.	35.	-999.	-999.
KERVILLE	367	B	9.	10.	13.	-999.	-999.
BELMONT	368	B	4.	6.	8.	-999.	-999.
DELAMAR	369	B	34.	37.	42.	-999.	-999.
HARDIN	370	B	21.	24.	28.	-999.	-999.
CONTACT	371	B	30.	45.	50.	55.	-999.
COMSTOCK	372	B	10.	22.	25.	-999.	-999.
BARNWELL	373	B	22.	23.	26.	-999.	-999.
PIPKIN	374	B	42.	62.	68.	-999.	-999.
TENABO	375	B	7.	8.	12.	-999.	-999.
HORNITOS	376	B	3.	5.	11.	-999.	-999.
BULLION	377	B	50.	77.	84.	307.	-999.
BULLION	378	WP	20.	31.	34.	-999.	-999.
HOYA	379	B	31.	34.	38.	-999.	-999.
MONTELLO	380	B	14.	27.	30.	-999.	-999.
BENHAM	381	B	5232.	9416.	17870.	30129.	80292.
BENHAM	382	WP	1178.	2835.	3060.	5961.	32933.
BENHAM	383	T	450.	925.	980.	4271.	11421.
KNICKERBOCKER	384	B	2.	3.	5.	-999.	-999.
JORUM	385	B	1985.	3532.	7312.	9564.	24997.
JORUM	386	WP	401.	828.	1746.	1837.	12783.
JORUM	387	T	146.	351.	708.	730.	-999.
FONTINA	388	B	2790.	6400.	16252.	36256.	57480.
FONTINA	389	WP	769.	2035.	3085.	14573.	22570.
FONTINA	390	T	263.	515.	732.	-999.	-999.
GREELEY	391	B	14793.	16614.	22224.	22840.	73734.
GREELEY	392	WP	5341.	5666.	6253.	6341.	18629.
GREELEY	393	T	1239.	1593.	2171.	2211.	9950.
REX	394	B	21.	40.	43.	-999.	-999.
REX	395	WP	21.	29.	31.	-999.	-999.
BOXCAR	396	B	1608.	3420.	6763.	8866.	21339.
BOXCAR	397	WP	324.	659.	1468.	1938.	11156.
BOXCAR	398	T	117.	233.	548.	652.	-999.
HANDLEY	399	B	2134238.	2154913.	2229105.	2271345.	2492654.
HANDLEY	400	WP	778423.	785429.	808331.	826313.	849607.
HANDLEY	401	T	175664.	177462.	181999.	204856.	-999.
CHESHIRE	402	B	1673.	3448.	4056.	5731.	376723.
CHESHIRE	403	WP	394.	1161.	1449.	5791.	32802.
CHESHIRE	404	T	226.	574.	690.	1623.	9879.
STILTON	405	B	26939.	28638.	31972.	32642.	-999.
STILTON	406	WP	2421.	3542.	4619.	4999.	-999.
STILTON	407	T	6100.	6484.	7459.	7617.	-999.
CHATEAUGAY	408	B	4.	5.	8.	-999.	-999.
PURSE	409	B	125.	252.	341.	-999.	-999.
PURSE	410	WP	3.	32.	36.	-999.	-999.
TYBO	411	B	87.	239.	255.	-999.	-999.
TYBO	412	WP	2.	3.	5.	-999.	-999.
KASSERI	413	B	304113.	309117.	318691.	341037.	467261.
KASSERI	414	WP	118326.	122704.	131843.	142255.	208970.
KASSERI	415	T	29798.	32909.	37644.	38510.	65455.

TRAVEL DISTANCE(m) AT FIXED TIMES(yrs)

SHOTNAME	ID	SLOC	50.	100.	150.	200.
COMMODORE	1	B	1131.	1799.	2114.	2414.
COMMODORE	2	WP	9.	18.	26.	35.
COMMODORE	3	T	58.	117.	177.	239.
CLARKSMOBILE	4	B	71.	144.	218.	295.
KNOX	5	B	20.	39.	59.	78.
KNOX	6	WP	113.	295.	656.	1049.
KNOX	7	T	363.	766.	990.	1202.
ILDRIM	8	B	288.	451.	770.	1131.
CALABASH	9	B	622.	1242.	2994.	4848.
CALABASH	10	WP	25.	50.	76.	101.
STANYAN	11	B	216.	465.	896.	1669.
STANYAN	12	WP	3047.	4533.	4983.	6358.
PORTMANTEAU	13	B	1032.	1880.	2550.	3867.
PORTMANTEAU	14	WP	1224.	2364.	3040.	3994.
YANNIGAN	15	B	368.	683.	884.	971.
YANNIGAN	16	B	360.	677.	877.	951.
FLASK	17	B	88.	179.	273.	370.
FLASK	18	WP	461.	802.	1165.	1553.
HAREBELL	19	B	2041.	6920.	9752.	11585.
STARWORT	20	B	71.	144.	219.	296.
STARWORT	21	WP	346.	552.	635.	720.
MINIATA	22	B	361.	762.	1161.	1850.
MINIATA	23	WP	407.	795.	1361.	2715.
PORTULACA	24	B	1457.	2518.	3127.	3835.
STODDARD	25	B	110.	231.	363.	510.
GORBEA	26	B	959.	1429.	2234.	5457.
CREW	27	B	3684.	4552.	6641.	8516.
CREW	28	WP	3684.	4552.	6641.	8516.
CREW	29	T	3684.	4552.	6641.	8516.
COFFER	30	B	3533.	4314.	5284.	7912.
HUTCH	31	B	57.	115.	172.	231.
CARPETBAG	32	B	49.	97.	147.	198.
CARPETBAG	33	WP	100.	202.	305.	410.
FLAX	34	B	58.	116.	175.	233.
FLAX	35	WP	58.	116.	175.	233.
FLAX	36	T	58.	116.	175.	233.
CABRILLO	37	B	10.	21.	31.	42.
CABRILLO	38	WP	3288.	4102.	5763.	7916.
FALLON	39	B	1226.	1327.	1428.	1529.
EDAM	40	B	4114.	6448.	7902.	8982.
BANON	41	B	93.	187.	282.	379.
COULOMMIERS	42	B	2468.	8000.	11392.	15473.
CHIBERTA	43	B	1146.	2598.	3833.	4720.
CHIBERTA	44	WP	8.	16.	24.	32.
CHIBERTA	45	T	181.	248.	317.	388.
MARSILLY	46	B	5.	10.	15.	19.
MARSILLY	47	WP	5.	10.	16.	21.
MARSILLY	48	T	323.	392.	463.	537.
REBLOCHON	49	B	6.	11.	17.	23.
REBLOCHON	50	WP	58.	117.	178.	240.
REBLOCHON	51	T	373.	760.	1115.	1195.
KLOSTER	52	B	1793.	3987.	7294.	8977.
NESSSEL	53	B	128.	182.	238.	296.
AKAVI	54	B	2846.	5740.	8357.	11362.
ROMANO	55	B	460.	563.	883.	2368.
FARALLONES	56	B	7.	14.	21.	27.
FARALLONES	57	WP	55.	110.	166.	222.

QUARGEL	58	B	96.	193.	803.	991.
QUARGEL	59	WP	5463.	8318.	10965.	13849.
FAJY	60	B	246.	307.	368.	430.
PANAMINT	61	B	377.	777.	967.	1000.
BORATE	62	B	102.	204.	307.	409.
SCHELLBOURNE	63	B	1048.	1101.	1130.	1158.
INGOT	64	B	2458.	5439.	7745.	10014.
METROPOLIS	65	B	110.	223.	450.	809.
CREPE	66	B	371.	790.	1283.	1908.
DUMONT	67	B	71.	348.	611.	859.
DUMONT	68	WP	102.	207.	316.	405.
DUMONT	69	T	335.	692.	1102.	1572.
AGILE	70	B	857.	1267.	1567.	1848.
AGILE	71	WP	10.	21.	31.	41.
AGILE	72	T	52.	105.	159.	214.
LANPHER	73	B	310.	602.	880.	1144.
LANPHER	74	WP	17.	34.	51.	68.
LANPHER	75	T	101.	206.	314.	426.
AARDVARK	76	B	210.	251.	293.	336.
WAGTAIL	77	B	60.	125.	197.	276.
WAGTAIL	78	WP	65.	134.	206.	281.
WAGTAIL	79	T	42.	84.	127.	170.
HAYMAKER	80	B	1357.	1432.	1506.	1580.
MERRIMAC	81	B	1046.	1116.	1186.	1257.
BELEN	82	B	1992.	2155.	2327.	2511.
PUCE	83	B	1555.	1696.	2206.	2645.
BILBY	84	B	1923.	3681.	7873.	8492.
BILBY	85	WP	537.	960.	1362.	1841.
BILBY	86	T	70.	141.	211.	281.
CANVASBACK	87	B	242.	494.	757.	2913.
KESTREL	88	B	71.	142.	214.	285.
BUFF	89	B	72.	146.	242.	347.
MORRONES	90	B	25.	49.	74.	99.
WISE	91	B	890.	1192.	1556.	1849.
PICCALILLI	92	B	5361.	11286.	16248.	26614.
LAGUNA	93	B	71.	143.	214.	286.
TORTUGAS	94	B	70.	140.	210.	257.
TORTUGAS	95	WP	37.	84.	131.	178.
TORTUGAS	96	T	328.	641.	680.	721.
BONARDA	97	B	2470.	2540.	2611.	2682.
ABEYAS	98	B	383.	1281.	2017.	2638.
CUMARIN	99	B	11611.	20112.	40822.	65774.
SCUPPER	100	B	923.	987.	1052.	1118.
ANGUS	101	B	587.	643.	699.	756.
PUYE	102	B	82.	164.	247.	329.
ALGODONES	103	B	28.	55.	83.	110.
ALGODONES	104	WP	1345.	1415.	1485.	1556.
MONERO	105	B	3649.	16443.	32782.	59786.
MONERO	106	WP	11729.	24175.	33027.	48567.
MARSH	107	B	82.	165.	248.	331.
OFFSHORE	108	B	707.	1344.	4036.	11123.
ALEMAN	109	B	41.	82.	122.	162.
ALEMAN	110	WP	320.	640.	969.	1048.
BOUSCHET	111	B	137.	188.	239.	291.
BOUSCHET	112	WP	37.	74.	111.	148.
VAUGHN	113	B	1719.	1753.	1788.	1823.
DUORO	114	B	571.	611.	650.	691.
KINIBITO	115	B	16.	32.	49.	65.
KINIBITO	116	WP	1424.	1451.	1478.	1506.
TAHOKA	117	B	17.	34.	51.	62.
TAHOKA	118	WP	17.	34.	51.	68.

TAHOKA	119	T	2101.	2172.	2242.	2312.
LUBBOCK	120	B	36.	72.	107.	142.
STRAIT	121	B	4070.	11809.	18331.	19211.
STRAIT	122	WP	795.	1585.	2335.	3656.
STRAIT	123	T	344.	690.	1017.	1351.
BURZET	124	B	286.	359.	429.	498.
MONTEREY	125	B	216.	423.	668.	896.
TILCI	126	B	164.	231.	295.	359.
MANTECA	127	B	666.	841.	901.	959.
COSO	128	B	218.	537.	915.	1039.
COSO	129	B	218.	537.	915.	1271.
BRETON	130	B	377.	452.	524.	595.
ROQUEFORT	131	B	1022.	1533.	2461.	4923.
BULLFROG	132	B	172.	246.	317.	386.
BRISTOL	133	B	496.	788.	1007.	1145.
ZAZA	134	B	897.	1588.	2770.	4703.
ZAZA	135	WP	414.	698.	925.	1149.
ZAZA	136	T	478.	831.	1188.	1623.
LATIR	137	B	645.	1424.	2137.	3556.
LATIR	138	WP	324.	623.	1023.	1544.
LATIR	139	T	410.	802.	1176.	2049.
TOPGALLANT	140	B	252.	486.	714.	1112.
TOPGALLANT	141	WP	19.	37.	56.	75.
TOPGALLANT	142	T	109.	221.	387.	727.
TRANSOM	143	B	56.	112.	169.	226.
TRANSOM	144	WP	56.	112.	169.	226.
TRANSOM	145	T	56.	112.	169.	226.
ICEBERG	146	B	567.	1108.	3226.	8129.
ICEBERG	147	WP	304.	606.	1149.	1865.
ICEBERG	148	T	479.	956.	1406.	2441.
SCANTLING	149	B	282.	706.	1791.	2560.
SCANTLING	150	WP	20.	39.	59.	78.
SCANTLING	151	T	113.	291.	641.	1019.
GLENCOE	152	B	54.	108.	164.	221.
GLENCOE	153	WP	70.	139.	205.	270.
GLENCOE	154	T	280.	552.	817.	1051.
JORNADA	155	B	233.	400.	580.	869.
JORNADA	156	WP	224.	530.	931.	1179.
JORNADA	157	T	507.	914.	1346.	1906.
QUINELLA	158	B	30.	60.	91.	121.
QUINELLA	159	WP	149.	438.	711.	990.
HEARTS	160	B	123.	254.	397.	553.
HEARTS	161	WP	57.	113.	171.	228.
HEARTS	162	T	318.	676.	1017.	1412.
TECHADO	163	B	397.	771.	1524.	2686.
TECHADO	164	WP	397.	845.	1682.	2868.
ROUSANNE	165	B	421.	1651.	2673.	5314.
ROUSANNE	166	WP	396.	774.	1364.	2844.
CAPROCK	167	B	680.	1274.	4265.	8581.
CAPROCK	168	WP	365.	831.	1241.	2119.
GASCON	169	B	589.	1158.	1990.	2719.
GASCON	170	WP	328.	676.	1069.	1830.
DALHART	171	B	162.	337.	613.	2007.
DALHART	172	WP	58.	116.	174.	351.
DALHART	173	T	339.	809.	1282.	1605.
WISHBONE	174	B	0.	0.	0.	0.
DILUTED WATER	175	B	20.	40.	61.	81.
CAMBRIC	176	B	23.	45.	68.	90.
CAMBRIC	177	WP	23.	45.	68.	90.
CAMBRIC	178	T	23.	45.	68.	90.
DERRINGER	179	B	365.	756.	1182.	1656.

MILKSHAKE	180	B	34.	68.	102.	121.
ESCABOSA	181	B	83.	170.	262.	358.
ESCABOSA	182	WP	222.	552.	1223.	2256.
MIERA	183	B	11310.	23217.	44835.	73110.
STRAKE	184	B	15520.	25355.	39105.	55920.
POTRILLO	185	B	2973.	5011.	15070.	23991.
POTRILLO	186	WP	2769.	5224.	7990.	19953.
OBAR	187	B	14993.	27008.	49869.	75983.
MIZZEN	188	B	6958.	8017.	8850.	9545.
MIZZEN	189	WP	981.	2425.	4003.	5811.
KEELSON	190	B	171.	351.	545.	1401.
KEELSON	191	WP	58.	142.	241.	345.
RUDDER	192	B	1811.	9129.	19884.	24506.
RUDDER	193	WP	1333.	4108.	6103.	7765.
ESROM	194	B	8840.	18136.	24525.	27115.
ESROM	195	WP	2817.	6027.	7894.	20113.
DRAUGHTS	196	B	25.	51.	76.	458.
BULKHEAD	197	B	3349.	6542.	18761.	30644.
BULKHEAD	198	WP	3233.	5586.	14904.	25199.
BILLET	199	B	61.	121.	355.	651.
BILLET	200	WP	38.	77.	115.	153.
BILLET	201	T	309.	651.	813.	996.
CREWLINE	202	B	564.	1367.	2078.	3235.
CREWLINE	203	WP	338.	769.	1464.	1814.
SANDREEF	204	B	5.	9.	14.	18.
SANDREEF	205	WP	334.	576.	635.	695.
SANDREEF	206	T	42.	84.	126.	167.
RUMMY	207	B	204.	449.	693.	914.
RUMMY	208	WP	48.	96.	144.	276.
RUMMY	209	T	269.	515.	573.	621.
LOWBALL	210	B	220.	477.	856.	1435.
LOWBALL	211	WP	47.	93.	140.	187.
BASEBALL	212	B	206.	435.	773.	1283.
BASEBALL	213	WP	55.	111.	167.	223.
PALIZA	214	B	8630.	21397.	34274.	53775.
PYRAMID	215	B	3945.	10150.	21601.	40762.
PYRAMID	216	WP	3355.	7666.	21339.	37099.
TAJO	217	B	17470.	26276.	39174.	48787.
MUNDO	218	B	17428.	27053.	43571.	68658.
ATRISCO	219	B	12176.	23681.	46535.	73299.
ATRISCO	220	WP	14370.	26497.	38916.	61177.
BORREGO	221	B	531.	1198.	1820.	2623.
BORREGO	222	WP	334.	770.	1161.	1431.
HERMOSA	223	B	3927.	15395.	23627.	30895.
HERMOSA	224	WP	2342.	4545.	6430.	11802.
TURQUOISE	225	B	56.	112.	169.	226.
TURQUOISE	226	WP	310.	429.	464.	499.
MIDLAND	227	B	14495.	26432.	36254.	42801.
TEXARKANA	228	B	15330.	28987.	38931.	64171.
FLOYDADA	229	B	12132.	23831.	31560.	33357.
PIRANHA	230	B	749.	2029.	3192.	4083.
PIRANHA	231	WP	276.	610.	964.	1382.
BRONZE	232	B	2113.	4698.	10832.	22091.
CHARCOAL	233	B	2133.	4030.	5479.	11277.
LAMPBLACK	234	B	4801.	18398.	36084.	61613.
LAMPBLACK	235	WP	3481.	16149.	28039.	34767.
LIME	236	B	17211.	27241.	36596.	43155.
TAN	237	B	70.	143.	220.	302.
TAN	238	WP	56.	113.	171.	231.
MICKEY	239	B	16550.	23789.	36968.	46894.
BOURBON	240	B	17405.	28763.	33945.	43145.

DAIQUIRI	241	B	1295.	4194.	6657.	10439.
DAIQUIRI	242	WP	1135.	4373.	6764.	16525.
BLENTON	243	B	2587.	13806.	25508.	47475.
BLENTON	244	WP	67.	2183.	10303.	22744.
SHAPER	245	B	2347.	4215.	11919.	22254.
SHAPER	246	WP	168.	1854.	4612.	14350.
GRAPE A	247	B	16025.	32637.	53051.	73161.
GRAPE A	248	WP	11575.	23419.	31938.	32959.
THISTLE	249	B	16898.	32629.	52217.	74179.
THISTLE	250	WP	5327.	20677.	30469.	33655.
COBBLER	251	B	997.	2155.	3575.	4763.
COBBLER	252	WP	939.	2088.	3552.	4786.
COBBLER	253	T	755.	1146.	1582.	3052.
GRAPE B	254	B	3977.	15376.	28998.	51763.
TORRIDO	255	B	16295.	23698.	37865.	54998.
ARTESIA	256	B	18938.	26673.	28767.	36425.
TIJERAS	257	B	4547.	16954.	32109.	55863.
TIJERAS	258	WP	3821.	16914.	31211.	47577.
OSCURO	259	B	520.	1243.	2482.	3351.
OSCURO	260	WP	143.	302.	497.	745.
COTTAGE	261	B	4255.	7257.	9328.	11631.
FRISCO	262	B	3645.	4984.	7031.	9073.
MISSISSIPPI	263	B	30.	60.	89.	119.
FORE	264	B	29.	59.	335.	2735.
TERRINE	265	B	1569.	6615.	21827.	37730.
NOGGIN	266	B	4411.	6906.	8746.	10743.
NOGGIN	267	WP	866.	1310.	1724.	2111.
CUP	268	B	18036.	29098.	49647.	75899.
YARD	269	B	12224.	25203.	42309.	64454.
STACCATO	270	B	5457.	24010.	33693.	45723.
CORNICE	271	B	21112.	31555.	32784.	33928.
TURF	272	B	6665.	10701.	21402.	34459.
KLICKITAT	273	B	13797.	27591.	43725.	49508.
CORDUROY	274	B	17439.	25572.	34278.	51735.
CORDUROY	275	WP	19159.	29843.	48055.	73520.
KANKAKEE	276	B	20022.	32655.	40793.	50028.
STANLEY	277	B	9340.	24212.	41358.	64596.
SHUFFLE	278	B	19164.	31286.	38027.	39492.
PINSTRIFE	279	B	246.	499.	758.	1023.
NEWPOINT	280	B	408.	831.	1278.	1754.
DIANA MOON	281	B	33.	76.	119.	163.
MINUTE STEAK	282	B	1.	2.	3.	4.
DIAGONAL LINE	283	B	10.	20.	31.	41.
MINI JADE	284	B	55.	110.	162.	215.
HUNTERS TROPH	285	B	64.	127.	189.	250.
SCOTCH	286	B	17.	34.	51.	69.
SCOTCH	287	WP	2158.	3495.	3560.	3628.
SCOTCH	288	T	2158.	5009.	6442.	7997.
SHEEPSHEAD	289	B	23343.	40983.	41683.	42120.
TOWANDA	290	B	4067.	4732.	5361.	6116.
TIERRA	291	B	6754.	8206.	9616.	29805.
CHANCELLOR	292	B	4253.	7343.	29878.	50037.
NEBBIOLO	293	B	28807.	41869.	43272.	43728.
SERPA	294	B	2902.	3022.	3142.	3262.
HARZER	295	B	98.	197.	299.	453.
HOSTA	296	B	4003.	4162.	4319.	4476.
LABQUARK	297	B	3271.	4168.	7781.	10710.
LOCKNEY	298	B	38511.	41827.	42947.	44274.
CYBAR	299	B	4557.	7484.	24670.	48523.
ALAMO	300	B	6609.	35429.	47342.	47579.
KEARSARGE	301	B	5868.	9443.	13081.	34183.

AMARILLO	302	B	4042.	6995.	21576.	41714.
HOUSTON	303	B	38979.	42344.	43517.	44573.
HALFBEAK	304	B	6743.	8256.	10674.	11433.
HALFBEAK	305	WP	7073.	9257.	11929.	13943.
HALFBEAK	306	T	1644.	2173.	2748.	3379.
BEXAR	307	B	3066.	6132.	9650.	16156.
JUNCTION	308	B	4635.	10636.	37557.	50428.
RICKEY	309	B	4428.	6650.	7417.	8022.
CHARTREUSE	310	B	3779.	3932.	4084.	4236.
CHARTREUSE	311	WP	3921.	4495.	5260.	6167.
MUENSTER	312	B	187.	380.	596.	819.
MUENSTER	313	WP	604.	1156.	1589.	2072.
MUENSTER	314	T	1504.	2440.	2465.	2491.
INLET	315	B	779.	1956.	2095.	2210.
INLET	316	WP	25903.	39740.	40457.	41136.
ESTUARY	317	B	2904.	4774.	7728.	7952.
ESTUARY	318	WP	3981.	7069.	10606.	12711.
ESTUARY	319	T	799.	1014.	1253.	1524.
SLED	320	B	1307.	2391.	2896.	3695.
SLED	321	WP	5182.	36995.	38461.	39417.
STINGER	322	B	7761.	9496.	12042.	15315.
STINGER	323	WP	5065.	6023.	6594.	6921.
POOL	324	B	25.	50.	75.	100.
POOL	325	WP	2202.	2699.	3258.	3857.
POOL	326	T	4702.	6145.	6774.	7343.
CAMEMBERT	327	B	2.	3.	5.	7.
CAMEMBERT	328	WP	2.	3.	5.	6.
CAMEMBERT	329	T	20.	40.	60.	80.
MAST	330	B	2931.	5553.	8827.	8875.
MAST	331	WP	9023.	11021.	13776.	16450.
MAST	332	T	4256.	4834.	5622.	6537.
ALMENDRO	333	B	8.	16.	24.	32.
ALMENDRO	334	WP	26.	111.	296.	521.
ALMENDRO	335	T	1761.	3110.	4047.	4395.
BACKBEACH	336	B	25823.	40626.	42006.	43196.
PANIR	337	B	2796.	5667.	8025.	9285.
PANIR	338	WP	5504.	28875.	50376.	50611.
FONDUTTA	339	B	170.	831.	1004.	1149.
BUTEO	340	B	-999.	-999.	-999.	-999.
BUTEO	341	WP	-999.	-999.	-999.	-999.
COLBY	342	B	57.	115.	173.	231.
COLBY	343	WP	236.	477.	725.	979.
COLBY	344	T	614.	1271.	1981.	2751.
FARM	345	B	3350.	9198.	39831.	40320.
FARM	346	WP	31805.	-999.	-999.	-999.
COLWICK	347	B	11308.	-999.	-999.	-999.
COLWICK	348	WP	-999.	-999.	-999.	-999.
PEPATO	349	B	8483.	39344.	39632.	-999.
PEPATO	350	WP	-999.	-999.	-999.	-999.
TAFI	351	B	10784.	34189.	34446.	34683.
TAFI	352	WP	-999.	-999.	-999.	-999.
KASH	353	B	31497.	-999.	-999.	-999.
KASH	354	WP	-999.	-999.	-999.	-999.
MOLBO	355	B	-999.	-999.	-999.	-999.
MOLBO	356	WP	-999.	-999.	-999.	-999.
GIBNE	357	B	-999.	-999.	-999.	-999.
JEFFERSON	358	B	-999.	-999.	-999.	-999.
CABRA	359	B	-999.	-999.	-999.	-999.
SALUT	360	B	-999.	-999.	-999.	-999.
EGMONT	361	B	33336.	-999.	-999.	-999.
KAPPELI	362	B	6794.	40182.	-999.	-999.

SERENA	363	B	-999.	-999.	-999.	-999.
GOLDSTONE	364	B	-999.	-999.	-999.	-999.
BODIE	365	B	8407.	-999.	-999.	-999.
DARWIN	366	B	-999.	-999.	-999.	-999.
KERVILLE	367	B	-999.	-999.	-999.	-999.
BELMONT	368	B	-999.	-999.	-999.	-999.
DELAMAR	369	B	-999.	-999.	-999.	-999.
HARDIN	370	B	-999.	-999.	-999.	-999.
CONTACT	371	B	18620.	-999.	-999.	-999.
COMSTOCK	372	B	-999.	-999.	-999.	-999.
BARNWELL	373	B	36523.	-999.	-999.	-999.
PIPKIN	374	B	6151.	-999.	-999.	-999.
TENABO	375	B	-999.	-999.	-999.	-999.
HORNITOS	376	B	-999.	-999.	-999.	-999.
BULLION	377	B	4993.	38068.	38562.	39051.
BULLION	378	WP	-999.	-999.	-999.	-999.
HOYA	379	B	35143.	-999.	-999.	-999.
MONTELLO	380	B	-999.	-999.	-999.	-999.
BENHAM	381	B	44.	89.	134.	179.
BENHAM	382	WP	201.	402.	601.	800.
BENHAM	383	T	998.	1992.	2397.	2803.
KNICKERBOCKER	384	B	-999.	-999.	-999.	-999.
JORUM	385	B	104.	208.	313.	418.
JORUM	386	WP	520.	1058.	1616.	2199.
JORUM	387	T	1480.	3149.	5201.	6895.
FONTINA	388	B	88.	178.	268.	359.
FONTINA	389	WP	368.	753.	1156.	1578.
FONTINA	390	T	978.	2078.	3222.	3943.
GREELEY	391	B	30.	59.	89.	119.
GREELEY	392	WP	148.	283.	405.	446.
GREELEY	393	T	342.	407.	470.	529.
REX	394	B	-999.	-999.	-999.	-999.
REX	395	WP	-999.	-999.	-999.	-999.
BOXCAR	396	B	120.	242.	365.	490.
BOXCAR	397	WP	608.	1253.	1950.	2716.
BOXCAR	398	T	1772.	4046.	6867.	9325.
HANDLEY	399	B	0.	0.	0.	0.
HANDLEY	400	WP	0.	0.	1.	1.
HANDLEY	401	T	1.	2.	3.	4.
CHESHIRE	402	B	116.	239.	375.	517.
CHESHIRE	403	WP	330.	716.	1146.	1626.
CHESHIRE	404	T	1193.	2637.	3585.	4526.
STILTON	405	B	86.	172.	259.	346.
STILTON	406	WP	254.	517.	1432.	1527.
STILTON	407	T	5.	10.	15.	20.
CHATEAUGAY	408	B	-999.	-999.	-999.	-999.
PURSE	409	B	1800.	3507.	5814.	8468.
PURSE	410	WP	-999.	-999.	-999.	-999.
TYBO	411	B	2806.	5332.	6645.	8020.
TYBO	412	WP	-999.	-999.	-999.	-999.
KASSERI	413	B	0.	1.	1.	1.
KASSERI	414	WP	1.	3.	4.	5.
KASSERI	415	T	27.	54.	81.	107.

Appendix E

Input Data Files to MC_Trans

A listing of the input files for each of the three pathlines is presented in this appendix. The file structure is described in [Appendix A](#). In the files, the end of each line represents a data label that is not read by the model, but is provided to aid the modeler in editing the files at a later time. In many of the data files, the integer immediately preceeding the data label is the numeric designation of the hydrostratigraphic unit. (See [Table 3-1](#) of the main text)

The files are:

KANKAKEE /BOURBON /MICKEY-TORRIDO

HOUSTON

PEPATO /KASH /TYBO

5	1	45	45	0	2.7807	0.00000	/VEL
5	1	46	46	0	2.7474	0.00000	/VEL
5	1	47	47	0	2.7149	0.00000	/VEL
5	1	48	48	0	2.6897	0.00000	/VEL
5	1	49	49	0	2.6638	0.00000	/VEL
5	1	50	50	0	2.6638	0.00000	/VEL
5	1	51	51	0	2.6394	0.00000	/VEL
5	1	52	52	0	2.6160	0.00000	/VEL
5	1	53	53	0	2.5934	0.00000	/VEL
5	1	54	54	0	2.0629	0.00000	/VEL
5	1	55	55	0	2.0326	0.00000	/VEL
5	1	56	56	0	1.9996	0.00000	/VEL
5	1	57	57	0	1.9756	0.00000	/VEL
5	1	58	58	0	1.9486	0.00000	/VEL
5	1	59	59	0	1.9486	0.00000	/VEL
5	1	60	60	0	1.9275	0.00000	/VEL
5	1	61	61	0	1.9034	0.00000	/VEL
5	1	62	62	0	1.8886	0.00000	/VEL
5	1	63	63	0	2.9092	0.00000	/VEL
5	1	64	64	0	2.9116	0.00000	/VEL
5	1	65	65	0	2.8927	0.00000	/VEL
5	1	66	66	0	2.8947	0.00000	/VEL
5	1	67	67	0	2.8785	0.00000	/VEL
5	1	68	68	0	2.8676	0.00000	/VEL
5	1	69	69	0	2.8552	0.00000	/VEL
5	1	70	70	0	3.0364	0.00000	/VEL
5	1	71	71	0	3.0364	0.00000	/VEL
5	1	72	72	0	3.0221	0.00000	/VEL
5	1	73	73	0	3.0201	0.00000	/VEL
5	1	74	74	0	3.0077	0.00000	/VEL
5	1	75	75	0	3.0027	0.00000	/VEL
5	1	76	76	0	3.0023	0.00000	/VEL
5	1	77	77	0	3.0023	0.00000	/VEL
5	1	78	78	0	2.9932	0.00000	/VEL
5	1	79	79	0	2.9955	0.00000	/VEL
5	1	80	80	0	2.9886	0.00000	/VEL
5	1	81	81	0	2.9920	0.00000	/VEL
5	1	82	82	0	2.9920	0.00000	/VEL
5	1	83	83	0	2.9977	0.00000	/VEL
5	1	84	84	0	2.9938	0.00000	/VEL
5	1	85	85	0	3.0008	0.00000	/VEL
5	1	86	86	0	3.0094	0.00000	/VEL
5	1	87	87	0	3.0136	0.00000	/VEL
5	1	88	88	0	3.0136	0.00000	/VEL
5	1	89	89	0	2.9357	0.00000	/VEL
5	1	90	90	0	2.9272	0.00000	/VEL
5	1	91	91	0	2.7605	0.00000	/VEL
5	1	92	92	0	2.7605	0.00000	/VEL
5	1	93	93	0	2.7564	0.00000	/VEL
5	1	94	94	0	2.7476	0.00000	/VEL
5	1	95	95	0	2.7406	0.00000	/VEL
5	1	96	96	0	2.7406	0.00000	/VEL
5	1	97	97	0	2.7361	0.00000	/VEL
5	1	98	98	0	2.7293	0.00000	/VEL
5	1	99	99	0	2.7209	0.00000	/VEL
5	1	100	100	0	2.7194	0.00000	/VEL
5	1	101	101	0	2.7194	0.00000	/VEL
5	1	102	102	0	2.7113	0.00000	/VEL
5	1	103	103	0	2.7100	0.00000	/VEL
5	1	104	104	0	2.7030	0.00000	/VEL
5	1	105	105	0	2.6969	0.00000	/VEL
5	1	106	106	0	2.6969	0.00000	/VEL
5	1	107	107	0	2.6952	0.00000	/VEL
5	1	108	108	0	2.6937	0.00000	/VEL
5	1	109	109	0	2.4606	0.00000	/VEL
5	1	110	110	0	2.4478	0.00000	/VEL
5	1	111	111	0	2.4478	0.00000	/VEL
5	1	112	112	0	2.4300	0.00000	/VEL
5	1	113	113	0	2.4211	0.00000	/VEL
5	1	114	114	0	2.3539	0.00000	/VEL
5	1	115	115	0	2.3570	0.00000	/VEL
5	1	116	116	0	2.3701	0.00000	/VEL
5	1	117	117	0	2.3688	0.00000	/VEL
5	1	118	118	0	2.3822	0.00000	/VEL
5	1	119	119	0	2.3822	0.00000	/VEL

5	1	120	120	0	2.3864	0.00000	/VEL
5	1	121	121	0	2.3948	0.00000	/VEL
5	1	122	122	0	2.3992	0.00000	/VEL
5	1	123	123	0	2.4132	0.00000	/VEL
5	1	124	124	0	2.4178	0.00000	/VEL
5	1	125	125	0	2.4266	0.00000	/VEL
5	1	126	126	0	2.4320	0.00000	/VEL
5	1	127	127	0	2.2498	0.00000	/VEL
5	1	128	128	0	2.2464	0.00000	/VEL
5	1	129	129	0	2.2464	0.00000	/VEL
5	1	130	130	0	2.2426	0.00000	/VEL
5	1	131	131	0	2.2397	0.00000	/VEL
5	1	132	132	0	2.2408	0.00000	/VEL
5	1	133	133	0	2.2428	0.00000	/VEL
5	1	134	134	0	2.2390	0.00000	/VEL
5	1	135	135	0	2.2361	0.00000	/VEL
5	1	136	136	0	2.2373	0.00000	/VEL
5	1	137	137	0	2.2345	0.00000	/VEL
5	1	138	138	0	2.2361	0.00000	/VEL
5	1	139	139	0	2.2324	0.00000	/VEL
5	1	140	140	0	2.2296	0.00000	/VEL
5	1	141	141	0	2.2317	0.00000	/VEL
5	1	142	142	0	2.2317	0.00000	/VEL
5	1	143	143	0	2.2281	0.00000	/VEL
5	1	144	144	0	2.1651	0.00000	/VEL
5	1	145	145	0	2.1562	0.00000	/VEL
5	1	146	146	0	2.1633	0.00000	/VEL
5	1	147	147	0	1.7266	0.00000	/VEL
5	1	148	148	0	1.7276	0.00000	/VEL
5	1	149	149	0	1.7243	0.00000	/VEL
5	1	150	150	0	1.7254	0.00000	/VEL
5	1	151	151	0	1.7254	0.00000	/VEL
5	1	152	152	0	1.7225	0.00000	/VEL
5	1	153	153	0	1.7232	0.00000	/VEL
5	1	154	154	0	1.7203	0.00000	/VEL
5	1	155	155	0	1.7210	0.00000	/VEL
5	1	156	156	0	1.7185	0.00000	/VEL
5	1	157	157	0	1.7192	0.00000	/VEL
5	1	158	158	0	1.7163	0.00000	/VEL
5	1	159	159	0	1.7171	0.00000	/VEL
5	1	160	160	0	1.6675	0.00000	/VEL
5	1	161	161	0	1.6679	0.00000	/VEL
5	1	162	162	0	1.6652	0.00000	/VEL
5	1	163	163	0	1.6652	0.00000	/VEL
5	1	164	164	0	1.6688	0.00000	/VEL
5	1	165	165	0	1.6658	0.00000	/VEL
5	1	166	166	0	1.6662	0.00000	/VEL
5	1	167	167	0	1.6632	0.00000	/VEL
5	1	168	168	0	1.6638	0.00000	/VEL
5	1	169	169	0	1.6677	0.00000	/VEL
5	1	170	170	0	1.6609	0.00000	/VEL
5	1	171	171	0	1.6651	0.00000	/VEL
5	1	172	172	0	1.6619	0.00000	/VEL
5	1	173	173	0	1.6625	0.00000	/VEL
5	1	174	174	0	1.6625	0.00000	/VEL
5	1	175	175	0	1.6628	0.00000	/VEL
5	1	176	176	0	1.6635	0.00000	/VEL
5	1	177	177	0	1.6399	0.00000	/VEL
5	1	178	178	0	1.6443	0.00000	/VEL
5	1	179	179	0	1.6443	0.00000	/VEL
5	1	180	180	0	1.6447	0.00000	/VEL
5	1	181	181	0	1.6488	0.00000	/VEL
5	1	182	182	0	1.5541	0.00000	/VEL
5	1	183	183	0	1.5541	0.00000	/VEL
5	1	184	184	0	1.5535	0.00000	/VEL
5	1	185	185	0	1.5535	0.00000	/VEL
5	1	186	186	0	1.5569	0.00000	/VEL
5	1	187	187	0	1.5529	0.00000	/VEL
5	1	188	188	0	1.5559	0.00000	/VEL
5	1	189	189	0	1.5593	0.00000	/VEL
5	1	190	190	0	1.5553	0.00000	/VEL
5	1	191	191	0	1.5587	0.00000	/VEL
5	1	192	192	0	1.5584	0.00000	/VEL
5	1	193	193	0	1.5463	0.00000	/VEL
5	1	194	194	0	1.5457	0.00000	/VEL

5	1	195	195	0	1.5486	0.00000	/VEL
5	1	196	196	0	1.5514	0.00000	/VEL
5	1	197	197	0	1.5514	0.00000	/VEL
5	1	198	198	0	1.5511	0.00000	/VEL
5	1	199	199	0	1.5540	0.00000	/VEL
5	1	200	200	0	1.5537	0.00000	/VEL
5	1	201	201	0	1.5566	0.00000	/VEL
5	1	202	202	0	1.5595	0.00000	/VEL
5	1	203	203	0	1.5589	0.00000	/VEL
5	1	204	204	0	1.5621	0.00000	/VEL
5	1	205	205	0	1.5650	0.00000	/VEL
5	1	206	206	0	1.5644	0.00000	/VEL
5	1	207	207	0	1.5676	0.00000	/VEL
5	1	208	208	0	1.5705	0.00000	/VEL
5	1	209	209	0	1.5539	0.00000	/VEL
5	1	210	210	0	1.5577	0.00000	/VEL
5	1	211	211	0	1.5574	0.00000	/VEL
5	1	212	212	0	1.5569	0.00000	/VEL
5	1	213	213	0	1.5564	0.00000	/VEL
5	1	214	214	0	1.5602	0.00000	/VEL
5	1	215	215	0	1.5559	0.00000	/VEL
5	1	216	216	0	1.5594	0.00000	/VEL
5	1	217	217	0	1.5591	0.00000	/VEL
5	1	218	218	0	1.5626	0.00000	/VEL
5	1	219	219	0	1.5583	0.00000	/VEL
5	1	220	220	0	1.5621	0.00000	/VEL
5	1	221	221	0	1.4560	0.00000	/VEL
5	1	222	222	0	1.4594	0.00000	/VEL
5	1	223	223	0	1.4623	0.00000	/VEL
5	1	224	224	0	1.4623	0.00000	/VEL
5	1	225	225	0	1.4618	0.00000	/VEL
5	1	226	226	0	1.4457	0.00000	/VEL
5	1	227	227	0	1.4455	0.00000	/VEL
5	1	228	228	0	1.4455	0.00000	/VEL
5	1	229	229	0	1.4457	0.00000	/VEL
5	1	230	230	0	1.4422	0.00000	/VEL
5	1	231	231	0	1.4422	0.00000	/VEL
5	1	232	232	0	1.4424	0.00000	/VEL
5	1	233	233	0	1.4389	0.00000	/VEL
5	1	234	234	0	1.4392	0.00000	/VEL
5	1	235	235	0	1.4357	0.00000	/VEL
5	1	236	236	0	1.4389	0.00000	/VEL
5	1	237	237	0	1.4359	0.00000	/VEL
5	1	238	238	0	1.4324	0.00000	/VEL
5	1	239	239	0	1.4326	0.00000	/VEL
5	1	240	240	0	1.4324	0.00000	/VEL
5	1	241	241	0	1.4324	0.00000	/VEL
5	1	242	242	0	1.4150	0.00000	/VEL
5	1	243	243	0	1.4059	0.00000	/VEL
5	1	244	244	0	1.4031	0.00000	/VEL
5	1	245	245	0	1.4003	0.00000	/VEL
5	1	246	246	0	1.3910	0.00000	/VEL
5	1	247	247	0	1.3884	0.00000	/VEL
5	1	248	248	0	1.3823	0.00000	/VEL
5	1	249	249	0	1.3794	0.00000	/VEL
5	1	250	250	0	1.3737	0.00000	/VEL
5	1	251	251	0	1.3676	0.00000	/VEL
5	1	252	252	0	1.3650	0.00000	/VEL
5	1	253	253	0	1.3589	0.00000	/VEL
5	1	254	254	0	1.3531	0.00000	/VEL
5	1	255	255	0	1.3501	0.00000	/VEL
5	1	256	256	0	1.3445	0.00000	/VEL
5	1	257	257	0	1.3205	0.00000	/VEL
5	1	258	258	0	1.3143	0.00000	/VEL
5	1	259	259	0	1.3143	0.00000	/VEL
5	1	260	260	0	1.3113	0.00000	/VEL
5	1	261	261	0	1.3052	0.00000	/VEL
5	1	262	262	0	1.3022	0.00000	/VEL
5	1	263	263	0	1.2990	0.00000	/VEL
5	1	264	264	0	1.2962	0.00000	/VEL
5	1	265	265	0	1.2931	0.00000	/VEL
5	1	266	266	0	1.5692	0.00000	/VEL
5	1	267	267	0	1.5690	0.00000	/VEL
5	1	268	268	0	1.5688	0.00000	/VEL
5	1	269	269	0	1.5726	0.00000	/VEL

5	1	270	270	0	1.5685	0.00000	/VEL
5	1	271	271	0	1.5724	0.00000	/VEL
5	1	272	272	0	1.5721	0.00000	/VEL
5	1	273	273	0	1.5531	0.00000	/VEL
5	1	274	274	0	1.5493	0.00000	/VEL
5	1	275	275	0	1.5455	0.00000	/VEL
5	1	276	276	0	1.5418	0.00000	/VEL
5	1	277	277	0	1.5381	0.00000	/VEL
5	1	278	278	0	1.5344	0.00000	/VEL
5	1	279	279	0	1.5305	0.00000	/VEL
5	1	280	280	0	1.5268	0.00000	/VEL
5	1	281	281	0	1.5268	0.00000	/VEL
5	1	282	282	0	1.5231	0.00000	/VEL
5	1	283	283	0	1.5156	0.00000	/VEL
5	1	284	284	0	1.5155	0.00000	/VEL
5	1	285	285	0	1.5119	0.00000	/VEL
5	1	286	286	0	1.5118	0.00000	/VEL
5	1	287	287	0	1.5044	0.00000	/VEL
5	1	288	288	0	1.4805	0.00000	/VEL
5	1	289	289	0	1.4732	0.00000	/VEL
5	1	290	290	0	1.4660	0.00000	/VEL
5	1	291	291	0	1.4550	0.00000	/VEL
5	1	292	292	0	1.4514	0.00000	/VEL
5	1	293	293	0	1.4405	0.00000	/VEL
5	1	294	294	0	1.4332	0.00000	/VEL
5	1	295	295	0	1.4295	0.00000	/VEL
5	1	296	296	0	1.4187	0.00000	/VEL
5	1	297	297	0	1.4113	0.00000	/VEL
5	1	298	298	0	1.4041	0.00000	/VEL
5	1	299	299	0	1.3968	0.00000	/VEL
5	1	300	300	0	1.1106	0.00000	/VEL
5	1	301	301	0	1.1072	0.00000	/VEL
5	1	302	302	0	1.1006	0.00000	/VEL
5	1	303	303	0	1.0707	0.00000	/VEL
5	1	304	304	0	1.0707	0.00000	/VEL
5	1	305	305	0	1.0327	0.00000	/VEL
5	1	306	306	0	0.99681	0.00000	/VEL
5	1	307	307	0	0.96298	0.00000	/VEL
5	1	308	308	0	0.92927	0.00000	/VEL
5	1	309	309	0	0.89760	0.00000	/VEL
5	1	310	310	0	0.86601	0.00000	/VEL
5	1	311	311	0	0.83646	0.00000	/VEL
5	1	312	312	0	0.80704	0.00000	/VEL
5	1	313	313	0	0.77960	0.00000	/VEL
5	1	314	314	0	0.75230	0.00000	/VEL
5	1	315	315	0	0.72705	0.00000	/VEL
5	1	316	316	0	0.69978	0.00000	/VEL
5	1	317	317	0	0.67873	0.00000	/VEL
5	1	318	318	0	0.62594	0.00000	/VEL
5	1	319	319	0	0.59381	0.00000	/VEL
5	1	320	320	0	0.56334	0.00000	/VEL
5	1	321	321	0	0.53630	0.00000	/VEL
5	1	322	322	0	0.94809	0.00000	/VEL
5	1	323	323	0	0.93650	0.00000	/VEL
5	1	324	324	0	0.92539	0.00000	/VEL
5	1	325	325	0	0.91426	0.00000	/VEL
5	1	326	326	0	0.90751	0.00000	/VEL
5	1	327	327	0	0.89711	0.00000	/VEL
5	1	328	328	0	0.88903	0.00000	/VEL
5	1	329	329	0	0.88903	0.00000	/VEL
5	1	330	330	0	0.88149	0.00000	/VEL
5	1	331	331	0	0.87249	0.00000	/VEL
5	1	332	332	0	0.86754	0.00000	/VEL
5	1	333	333	0	0.86316	0.00000	/VEL
5	1	334	334	0	0.85542	0.00000	/VEL
5	1	335	335	0	1.4138	0.00000	/VEL
5	1	336	336	0	1.3189	0.00000	/VEL
5	1	337	337	0	1.3189	0.00000	/VEL
5	1	338	338	0	1.2342	0.00000	/VEL
5	1	339	339	0	1.1600	0.00000	/VEL
5	1	340	340	0	1.0983	0.00000	/VEL
5	1	341	341	0	1.0454	0.00000	/VEL
5	1	342	342	0	1.0004	0.00000	/VEL
5	1	343	343	0	0.96571	0.00000	/VEL
5	1	344	344	0	0.93907	0.00000	/VEL

5	1	345	345	0	0.78801	0.00000	/VEL
5	1	346	346	0	0.78220	0.00000	/VEL
5	1	347	347	0	0.77724	0.00000	/VEL
5	1	348	348	0	0.77725	0.00000	/VEL
5	1	349	349	0	0.77902	0.00000	/VEL
5	1	350	350	0	0.78408	0.00000	/VEL
5	1	351	351	0	0.79048	0.00000	/VEL
5	1	352	352	0	0.81208	0.00000	/VEL
5	1	353	353	0	0.82461	0.00000	/VEL
5	1	354	354	0	0.84038	0.00000	/VEL
5	1	355	355	0	0.85680	0.00000	/VEL
5	1	356	356	0	2.3241	0.00000	/VEL
5	1	357	357	0	2.2974	0.00000	/VEL
5	1	358	358	0	2.2974	0.00000	/VEL
5	1	359	359	0	2.2164	0.00000	/VEL
5	1	360	360	0	2.1406	0.00000	/VEL
5	1	361	361	0	2.1406	0.00000	/VEL
5	1	362	362	0	2.0703	0.00000	/VEL
5	1	363	363	0	2.0019	0.00000	/VEL
5	1	364	364	0	1.9427	0.00000	/VEL
5	1	365	365	0	1.8861	0.00000	/VEL
5	1	366	366	0	1.8353	0.00000	/VEL
5	1	367	367	0	1.7858	0.00000	/VEL
5	1	368	368	0	1.7858	0.00000	/VEL
5	1	369	369	0	1.7506	0.00000	/VEL
5	1	370	370	0	1.7102	0.00000	/VEL
5	1	371	371	0	1.6803	0.00000	/VEL
5	1	372	372	0	1.6519	0.00000	/VEL
5	1	373	373	0	1.6308	0.00000	/VEL
5	1	374	374	0	1.6155	0.00000	/VEL
5	1	375	375	0	1.2979	0.00000	/VEL
5	1	376	376	0	1.2627	0.00000	/VEL
5	1	377	377	0	1.2320	0.00000	/VEL
5	1	378	378	0	1.2073	0.00000	/VEL
5	1	379	379	0	1.2073	0.00000	/VEL
5	1	380	380	0	1.1887	0.00000	/VEL
5	1	381	381	0	1.1724	0.00000	/VEL
5	1	382	382	0	1.1652	0.00000	/VEL
5	1	383	383	0	1.1594	0.00000	/VEL
5	1	384	384	0	1.1591	0.00000	/VEL
5	1	385	385	0	1.8555	0.00000	/VEL
5	1	386	386	0	1.8639	0.00000	/VEL
5	1	387	387	0	1.8639	0.00000	/VEL
5	1	388	388	0	1.8694	0.00000	/VEL
5	1	389	389	0	1.8772	0.00000	/VEL
5	1	390	390	0	1.8882	0.00000	/VEL
5	1	391	391	0	1.8977	0.00000	/VEL
5	1	392	392	0	1.9084	0.00000	/VEL
5	1	393	393	0	1.6509	0.00000	/VEL
5	1	394	394	0	1.6509	0.00000	/VEL
5	1	395	395	0	1.5858	0.00000	/VEL
5	1	396	396	0	1.5296	0.00000	/VEL
5	1	397	397	0	1.5296	0.00000	/VEL
5	1	398	398	0	1.4847	0.00000	/VEL
5	1	399	399	0	1.4450	0.00000	/VEL
5	1	400	400	0	1.4173	0.00000	/VEL
5	1	401	401	0	1.4173	0.00000	/VEL
5	1	402	402	0	1.3935	0.00000	/VEL
5	1	403	403	0	1.3818	0.00000	/VEL
5	1	404	404	0	1.3787	0.00000	/VEL
5	1	405	405	0	1.3844	0.00000	/VEL
5	1	406	406	0	1.3844	0.00000	/VEL
5	1	407	407	0	1.3965	0.00000	/VEL
5	1	408	408	0	1.4187	0.00000	/VEL
5	1	409	409	0	1.4496	0.00000	/VEL
5	1	410	410	0	2.9115	0.00000	/VEL
5	1	411	411	0	2.9257	0.00000	/VEL
5	1	412	412	0	2.9376	0.00000	/VEL
5	1	413	413	0	2.6742	0.00000	/VEL
5	1	414	414	0	2.6742	0.00000	/VEL
5	1	415	415	0	2.6682	0.00000	/VEL
5	1	416	416	0	2.6617	0.00000	/VEL
5	1	417	417	0	2.6552	0.00000	/VEL
5	1	418	418	0	2.6558	0.00000	/VEL
5	1	419	419	0	2.6558	0.00000	/VEL

5	1	420	420	0	2.6428	0.00000	/VEL
5	1	421	421	0	2.6433	0.00000	/VEL
5	1	422	422	0	2.6368	0.00000	/VEL
5	1	423	423	0	2.6371	0.00000	/VEL
5	1	424	424	0	2.6309	0.00000	/VEL
5	1	425	425	0	2.6309	0.00000	/VEL
5	1	426	426	0	2.6182	0.00000	/VEL
5	1	427	427	0	2.6185	0.00000	/VEL
5	1	428	428	0	2.6185	0.00000	/VEL
5	1	429	429	0	2.6187	0.00000	/VEL
5	1	430	430	0	2.6192	0.00000	/VEL
5	1	431	431	0	2.6323	0.00000	/VEL
5	1	432	432	0	2.6327	0.00000	/VEL
5	1	433	433	0	2.6395	0.00000	/VEL
5	1	434	434	0	2.6465	0.00000	/VEL
5	1	435	435	0	2.6465	0.00000	/VEL
5	1	436	436	0	2.6601	0.00000	/VEL
5	1	437	437	0	2.6603	0.00000	/VEL
5	1	438	438	0	2.6673	0.00000	/VEL
5	1	439	439	0	2.6673	0.00000	/VEL
5	1	440	440	0	2.6809	0.00000	/VEL
5	1	441	441	0	2.6814	0.00000	/VEL
5	1	442	442	0	2.6881	0.00000	/VEL
5	1	443	443	0	2.6735	0.00000	/VEL
5	1	444	444	0	2.6213	0.00000	/VEL
5	1	445	445	0	2.5578	0.00000	/VEL
5	1	446	446	0	2.5060	0.00000	/VEL
5	1	447	447	0	2.4540	0.00000	/VEL
5	1	448	448	0	2.4023	0.00000	/VEL
5	1	449	449	0	2.3508	0.00000	/VEL
5	1	450	450	0	2.3050	0.00000	/VEL
5	1	451	451	0	2.2536	0.00000	/VEL
5	1	452	452	0	2.2026	0.00000	/VEL
5	1	453	453	0	2.1632	0.00000	/VEL
5	1	454	454	0	2.1121	0.00000	/VEL
5	1	455	455	0	2.0729	0.00000	/VEL
5	1	456	456	0	2.0281	0.00000	/VEL
5	1	457	457	0	0.81173	0.00000	/VEL
5	1	458	458	0	1.0679	0.00000	/VEL
5	1	459	459	0	1.1102	0.00000	/VEL
5	1	460	460	0	1.1543	0.00000	/VEL
5	1	461	461	0	1.2039	0.00000	/VEL
5	1	462	462	0	1.2549	0.00000	/VEL
5	1	463	463	0	1.3125	0.00000	/VEL
5	1	464	464	0	1.3125	0.00000	/VEL
5	1	465	465	0	1.3717	0.00000	/VEL
5	1	466	466	0	1.4362	0.00000	/VEL
5	1	467	467	0	1.5062	0.00000	/VEL
5	1	468	468	0	1.5062	0.00000	/VEL
5	1	469	469	0	1.5797	0.00000	/VEL
5	1	470	470	0	1.6883	0.00000	/VEL
5	1	471	471	0	1.6795	0.00000	/VEL
5	1	472	472	0	1.6726	0.00000	/VEL
5	1	473	473	0	1.6642	0.00000	/VEL
5	1	474	474	0	0.84212	0.00000	/VEL
5	1	475	475	0	0.87227	0.00000	/VEL
5	1	476	476	0	0.88788	0.00000	/VEL
5	1	477	477	0	0.91889	0.00000	/VEL
5	1	478	478	0	0.95471	0.00000	/VEL
5	1	479	479	0	0.97058	0.00000	/VEL
5	1	480	480	0	1.0674	0.00000	/VEL
5	1	481	481	0	1.0931	0.00000	/VEL
5	1	482	482	0	1.1220	0.00000	/VEL
5	1	483	483	0	1.1486	0.00000	/VEL
5	1	484	484	0	1.1786	0.00000	/VEL
5	1	485	485	0	1.2083	0.00000	/VEL
5	1	486	486	0	1.0871	0.00000	/VEL
5	1	487	487	0	1.1028	0.00000	/VEL
5	1	488	488	0	1.1206	0.00000	/VEL
5	1	489	489	0	1.1392	0.00000	/VEL
5	1	490	490	0	1.1560	0.00000	/VEL
5	1	491	491	0	1.1560	0.00000	/VEL
5	1	492	492	0	1.1755	0.00000	/VEL
5	1	493	493	0	1.1933	0.00000	/VEL
5	1	494	494	0	1.2160	0.00000	/VEL

5	1	495	495	0	1.2348	0.00000	/VEL
5	1	496	496	0	1.2588	0.00000	/VEL
5	1	497	497	0	1.2666	0.00000	/VEL
5	1	498	498	0	1.2710	0.00000	/VEL
5	1	499	499	0	1.2768	0.00000	/VEL
5	1	500	500	0	1.2875	0.00000	/VEL
5	1	501	501	0	1.2875	0.00000	/VEL
5	1	502	502	0	1.2948	0.00000	/VEL
5	1	503	503	0	1.8140	0.00000	/VEL
5	1	504	504	0	1.8234	0.00000	/VEL
5	1	505	505	0	1.8316	0.00000	/VEL
5	1	506	506	0	1.8447	0.00000	/VEL
5	1	507	507	0	1.8535	0.00000	/VEL
5	1	508	508	0	1.8623	0.00000	/VEL
5	1	509	509	0	1.8711	0.00000	/VEL
5	1	510	510	0	1.8843	0.00000	/VEL
5	1	511	511	0	1.8892	0.00000	/VEL
5	1	512	512	0	1.9019	0.00000	/VEL
5	1	513	513	0	1.9977	0.00000	/VEL
5	1	514	514	0	2.0082	0.00000	/VEL
5	1	515	515	0	2.0241	0.00000	/VEL
5	1	516	516	0	2.0241	0.00000	/VEL
5	1	517	517	0	2.0403	0.00000	/VEL
5	1	518	518	0	2.0577	0.00000	/VEL
5	1	519	519	0	2.0768	0.00000	/VEL
5	1	520	520	0	2.0972	0.00000	/VEL
5	1	521	521	0	2.1147	0.00000	/VEL
5	1	522	522	0	2.1367	0.00000	/VEL
5	1	523	523	0	2.1367	0.00000	/VEL
5	1	524	524	0	2.1615	0.00000	/VEL
5	1	525	525	0	2.1836	0.00000	/VEL
5	1	526	526	0	2.2126	0.00000	/VEL
5	1	527	527	0	2.2359	0.00000	/VEL
5	1	528	528	0	2.2359	0.00000	/VEL
5	1	529	529	0	2.2682	0.00000	/VEL
5	1	530	530	0	2.2967	0.00000	/VEL
5	1	531	531	0	2.4951	0.00000	/VEL
5	1	532	532	0	2.5046	0.00000	/VEL
5	1	533	533	0	2.5046	0.00000	/VEL
5	1	534	534	0	2.5121	0.00000	/VEL
5	1	535	535	0	2.5216	0.00000	/VEL
5	1	536	536	0	2.5282	0.00000	/VEL
5	1	537	537	0	2.5377	0.00000	/VEL
5	1	538	538	0	3.4145	0.00000	/VEL
5	1	539	539	0	3.4101	0.00000	/VEL
5	1	540	540	0	3.4052	0.00000	/VEL
5	1	541	541	0	3.4052	0.00000	/VEL
5	1	542	542	0	3.3986	0.00000	/VEL
5	1	543	543	0	3.3976	0.00000	/VEL
5	1	544	544	0	3.3894	0.00000	/VEL
5	1	545	545	0	3.3862	0.00000	/VEL
5	1	546	546	0	3.3819	0.00000	/VEL
5	1	547	547	0	3.3770	0.00000	/VEL
5	1	548	548	0	3.3721	0.00000	/VEL
5	1	549	549	0	3.3695	0.00000	/VEL
5	1	550	550	0	3.3695	0.00000	/VEL
5	1	551	551	0	3.3726	0.00000	/VEL
5	1	552	552	0	3.3837	0.00000	/VEL
5	1	553	553	0	3.3996	0.00000	/VEL
5	1	554	554	0	3.3996	0.00000	/VEL
5	1	555	555	0	3.4139	0.00000	/VEL
5	1	556	556	0	3.4286	0.00000	/VEL
5	1	557	557	0	3.4414	0.00000	/VEL
5	1	558	558	0	3.4544	0.00000	/VEL
5	1	559	559	0	3.4707	0.00000	/VEL
5	1	560	560	0	3.4903	0.00000	/VEL
5	1	561	561	0	3.5002	0.00000	/VEL
5	1	562	562	0	3.5002	0.00000	/VEL
5	1	563	563	0	3.5151	0.00000	/VEL
5	1	564	564	0	3.5351	0.00000	/VEL
5	1	565	565	0	3.5484	0.00000	/VEL
5	1	566	566	0	3.5669	0.00000	/VEL
5	1	567	567	0	3.5804	0.00000	/VEL
5	1	568	568	0	3.5804	0.00000	/VEL
5	1	569	569	0	3.4236	0.00000	/VEL

5	1	570	570	0	3.4237	0.00000	/VEL
5	1	571	571	0	3.4239	0.00000	/VEL
5	1	572	572	0	3.4304	0.00000	/VEL
5	1	573	573	0	3.4344	0.00000	/VEL
5	1	574	574	0	3.4367	0.00000	/VEL
5	1	575	575	0	3.4457	0.00000	/VEL
5	1	576	576	0	3.4457	0.00000	/VEL
5	1	577	577	0	3.4470	0.00000	/VEL
5	1	578	578	0	3.4535	0.00000	/VEL
5	1	579	579	0	3.4635	0.00000	/VEL
5	1	580	580	0	3.4654	0.00000	/VEL
5	1	581	581	0	3.4761	0.00000	/VEL
5	1	582	582	0	3.4761	0.00000	/VEL
5	1	583	583	0	3.4855	0.00000	/VEL
5	1	584	584	0	3.4897	0.00000	/VEL
5	1	585	585	0	3.5050	0.00000	/VEL
5	1	586	586	0	3.4126	0.00000	/VEL
5	1	587	587	0	3.4094	0.00000	/VEL
5	1	588	588	0	3.4000	0.00000	/VEL
5	1	589	589	0	3.4000	0.00000	/VEL
5	1	590	590	0	3.3997	0.00000	/VEL
5	1	591	591	0	3.3949	0.00000	/VEL
5	1	592	592	0	3.3960	0.00000	/VEL
5	1	593	593	0	3.3886	0.00000	/VEL
5	1	594	594	0	3.5468	0.00000	/VEL
5	1	595	595	0	3.5468	0.00000	/VEL
5	1	596	596	0	3.5441	0.00000	/VEL
5	1	597	597	0	3.5449	0.00000	/VEL
5	1	598	598	0	3.5440	0.00000	/VEL
5	1	599	599	0	3.5448	0.00000	/VEL
5	1	600	600	0	3.5481	0.00000	/VEL
5	1	601	601	0	3.5473	0.00000	/VEL
5	1	602	602	0	3.5482	0.00000	/VEL
5	1	603	603	0	3.5482	0.00000	/VEL
5	1	604	604	0	3.5474	0.00000	/VEL
5	1	605	605	0	3.5039	0.00000	/VEL
5	1	606	606	0	3.5046	0.00000	/VEL
5	1	607	607	0	3.5053	0.00000	/VEL
5	1	608	608	0	3.5075	0.00000	/VEL
5	1	609	609	0	3.5075	0.00000	/VEL
5	1	610	610	0	3.5135	0.00000	/VEL
5	1	611	611	0	3.5143	0.00000	/VEL
5	1	612	612	0	3.5151	0.00000	/VEL
5	1	613	613	0	3.5174	0.00000	/VEL
5	1	614	614	0	3.5174	0.00000	/VEL
5	1	615	615	0	3.5220	0.00000	/VEL
5	1	616	616	0	3.5243	0.00000	/VEL
5	1	617	617	0	3.5252	0.00000	/VEL
5	1	618	618	0	3.5252	0.00000	/VEL
5	1	619	619	0	3.5261	0.00000	/VEL
5	1	620	620	0	3.5323	0.00000	/VEL
5	1	621	621	0	3.5347	0.00000	/VEL
5	1	622	622	0	3.5356	0.00000	/VEL
5	1	623	623	0	3.5356	0.00000	/VEL
5	1	624	624	0	3.5478	0.00000	/VEL
5	1	625	625	0	3.5468	0.00000	/VEL
5	1	626	626	0	3.5550	0.00000	/VEL
5	1	627	627	0	3.5579	0.00000	/VEL
5	1	628	628	0	3.5579	0.00000	/VEL
5	1	629	629	0	3.5623	0.00000	/VEL
5	1	630	630	0	3.5691	0.00000	/VEL
5	1	631	631	0	3.5710	0.00000	/VEL
5	1	632	632	0	3.5710	0.00000	/VEL
5	1	633	633	0	3.5763	0.00000	/VEL
5	1	634	634	0	3.5821	0.00000	/VEL
5	1	635	635	0	3.5850	0.00000	/VEL
5	1	636	636	0	3.5918	0.00000	/VEL
5	1	637	637	0	3.5918	0.00000	/VEL
5	1	638	638	0	3.5923	0.00000	/VEL
5	1	639	639	0	3.6005	0.00000	/VEL
5	1	640	640	0	3.6034	0.00000	/VEL
5	1	641	641	0	3.6034	0.00000	/VEL
5	1	642	642	0	3.6078	0.00000	/VEL
5	1	643	643	0	3.2687	0.00000	/VEL
5	1	644	644	0	3.2687	0.00000	/VEL

5	1	645	645	0	3.2760	0.00000	/VEL
5	1	646	646	0	3.2811	0.00000	/VEL
5	1	647	647	0	3.2804	0.00000	/VEL
5	1	648	648	0	3.2804	0.00000	/VEL
5	1	649	649	0	3.2897	0.00000	/VEL
5	1	650	650	0	3.2943	0.00000	/VEL
5	1	651	651	0	3.2943	0.00000	/VEL
5	1	652	652	0	3.2964	0.00000	/VEL
5	1	653	653	0	3.3064	0.00000	/VEL
5	1	654	654	0	3.3156	0.00000	/VEL
5	1	655	655	0	3.3156	0.00000	/VEL
5	1	656	656	0	3.3183	0.00000	/VEL
5	1	657	657	0	3.3279	0.00000	/VEL
5	1	658	658	0	3.3360	0.00000	/VEL
5	1	659	659	0	3.3360	0.00000	/VEL
5	1	660	660	0	3.3421	0.00000	/VEL
5	1	661	661	0	3.3547	0.00000	/VEL
5	1	662	662	0	3.3547	0.00000	/VEL
5	1	663	663	0	3.3623	0.00000	/VEL
5	1	664	664	0	3.1177	0.00000	/VEL
5	1	665	665	0	3.1147	0.00000	/VEL
5	1	666	666	0	3.1147	0.00000	/VEL
5	1	667	667	0	3.1152	0.00000	/VEL
5	1	668	668	0	3.1175	0.00000	/VEL
5	1	669	669	0	3.1191	0.00000	/VEL
5	1	670	670	0	3.1235	0.00000	/VEL
5	1	671	671	0	3.1218	0.00000	/VEL
5	1	672	672	0	3.1218	0.00000	/VEL
5	1	673	673	0	3.1317	0.00000	/VEL
5	1	674	674	0	3.1377	0.00000	/VEL
5	1	675	675	0	3.1442	0.00000	/VEL
5	1	676	676	0	3.1559	0.00000	/VEL
5	1	677	677	0	3.1559	0.00000	/VEL
5	1	678	678	0	3.1587	0.00000	/VEL
5	1	679	679	0	3.1736	0.00000	/VEL
5	1	680	680	0	3.1869	0.00000	/VEL
5	1	681	681	0	3.1930	0.00000	/VEL
5	1	682	682	0	3.1930	0.00000	/VEL
5	1	683	683	0	3.2133	0.00000	/VEL
5	1	684	684	0	2.7465	0.00000	/VEL
5	1	685	685	0	2.8661	0.00000	/VEL
5	1	686	686	0	2.9141	0.00000	/VEL
5	1	687	687	0	2.9870	0.00000	/VEL
5	1	688	688	0	3.0435	0.00000	/VEL
5	1	689	689	0	3.1001	0.00000	/VEL
5	1	690	690	0	3.1821	0.00000	/VEL
5	1	691	691	0	3.2390	0.00000	/VEL
5	1	692	692	0	3.3134	0.00000	/VEL
5	1	693	693	0	3.3788	0.00000	/VEL
5	1	694	694	0	3.4534	0.00000	/VEL
5	1	695	695	0	3.5276	0.00000	/VEL
5	1	696	696	0	3.6108	0.00000	/VEL
5	1	697	697	0	3.6856	0.00000	/VEL
5	1	698	698	0	3.7605	0.00000	/VEL
5	1	699	699	0	9.1178	0.00000	/VEL
5	1	700	700	0	8.7989	0.00000	/VEL
5	1	701	701	0	8.8047	0.00000	/VEL
5	1	702	702	0	8.8214	0.00000	/VEL
5	1	703	703	0	8.8214	0.00000	/VEL
5	1	704	704	0	8.8457	0.00000	/VEL
5	1	705	705	0	8.8703	0.00000	/VEL
5	1	706	706	0	8.9099	0.00000	/VEL
5	1	707	707	0	8.9099	0.00000	/VEL
5	1	708	708	0	8.9686	0.00000	/VEL
5	1	709	709	0	9.0082	0.00000	/VEL
5	1	710	710	0	9.0908	0.00000	/VEL
5	1	711	711	0	9.1502	0.00000	/VEL
5	1	712	712	0	9.1502	0.00000	/VEL
5	1	713	713	0	9.2353	0.00000	/VEL
5	1	714	714	0	9.3311	0.00000	/VEL
5	1	715	715	0	9.4343	0.00000	/VEL
5	1	716	716	0	9.4343	0.00000	/VEL
5	1	717	717	0	9.5474	0.00000	/VEL
5	1	718	718	0	9.6615	0.00000	/VEL
5	1	719	719	0	10.719	0.00000	/VEL

5	1	720	720	0	10.550	0.00000	/VEL
5	1	721	721	0	10.550	0.00000	/VEL
5	1	722	722	0	10.344	0.00000	/VEL
5	1	723	723	0	8.4425	0.00000	/VEL
5	1	724	724	0	8.3181	0.00000	/VEL
5	1	725	725	0	4.0844	0.00000	/VEL
5	1	726	726	0	3.9927	0.00000	/VEL
5	1	727	727	0	3.9927	0.00000	/VEL
5	1	728	728	0	3.9010	0.00000	/VEL
5	1	729	729	0	3.8117	0.00000	/VEL
5	1	730	730	0	3.7269	0.00000	/VEL
5	1	731	731	0	3.6422	0.00000	/VEL
5	1	732	732	0	3.5600	0.00000	/VEL
5	1	733	733	0	3.4801	0.00000	/VEL
5	1	734	734	0	3.4008	0.00000	/VEL
5	1	735	735	0	3.3279	0.00000	/VEL
5	1	736	736	0	3.2524	0.00000	/VEL
5	1	737	737	0	3.1104	0.00000	/VEL
5	1	738	738	0	3.0379	0.00000	/VEL
5	1	739	739	0	3.0379	0.00000	/VEL
5	1	740	740	0	2.9537	0.00000	/VEL
5	1	741	741	0	2.8745	0.00000	/VEL
5	1	742	742	0	2.7975	0.00000	/VEL
5	1	743	743	0	2.7216	0.00000	/VEL
5	1	744	744	0	2.6529	0.00000	/VEL
5	1	745	745	0	2.5868	0.00000	/VEL
5	1	746	746	0	2.5187	0.00000	/VEL
5	1	747	747	0	2.4613	0.00000	/VEL
5	1	748	748	0	2.4037	0.00000	/VEL
5	1	749	749	0	2.3469	0.00000	/VEL
5	1	750	750	0	2.2972	0.00000	/VEL
5	1	751	751	0	2.2501	0.00000	/VEL
5	1	752	752	0	2.2058	0.00000	/VEL
5	1	753	753	0	2.1642	0.00000	/VEL
5	1	754	754	0	2.3804	0.00000	/VEL
5	1	755	755	0	2.3866	0.00000	/VEL
5	1	756	756	0	2.3866	0.00000	/VEL
5	1	757	757	0	2.3920	0.00000	/VEL
5	1	758	758	0	2.4034	0.00000	/VEL
5	1	759	759	0	2.4166	0.00000	/VEL
5	1	760	760	0	2.4166	0.00000	/VEL
5	1	761	761	0	2.4370	0.00000	/VEL
5	1	762	762	0	2.4553	0.00000	/VEL
5	1	763	763	0	2.4833	0.00000	/VEL
5	1	764	764	0	2.4833	0.00000	/VEL
5	1	765	765	0	2.5126	0.00000	/VEL
5	1	766	766	0	2.5460	0.00000	/VEL
5	1	767	767	0	2.5460	0.00000	/VEL
5	1	768	768	0	4.7113	0.00000	/VEL
5	1	769	769	0	4.7001	0.00000	/VEL
5	1	770	770	0	4.7043	0.00000	/VEL
5	1	771	771	0	4.6902	0.00000	/VEL
5	1	772	772	0	4.6803	0.00000	/VEL
5	1	773	773	0	4.6880	0.00000	/VEL
5	1	774	774	0	4.6880	0.00000	/VEL
5	1	775	775	0	4.6509	0.00000	/VEL
5	1	776	776	0	4.6068	0.00000	/VEL
5	1	777	777	0	4.5639	0.00000	/VEL
5	1	778	778	0	4.5234	0.00000	/VEL
5	1	779	779	0	4.4851	0.00000	/VEL
5	1	780	780	0	4.4446	0.00000	/VEL
5	1	781	781	0	4.4041	0.00000	/VEL
5	1	782	782	0	4.4041	0.00000	/VEL
5	1	783	783	0	4.3658	0.00000	/VEL
5	1	784	784	0	4.3241	0.00000	/VEL
5	1	785	785	0	4.2894	0.00000	/VEL
5	1	786	786	0	4.2489	0.00000	/VEL
5	1	787	787	0	4.2130	0.00000	/VEL
5	1	788	788	0	4.1771	0.00000	/VEL
5	1	789	789	0	4.1366	0.00000	/VEL
5	1	790	790	0	4.4608	0.00000	/VEL
5	1	791	791	0	4.4085	0.00000	/VEL
5	1	792	792	0	4.4085	0.00000	/VEL
5	1	793	793	0	4.3524	0.00000	/VEL
5	1	794	794	0	4.3020	0.00000	/VEL

5	1	795	795	0	4.2479	0.00000	/VEL
5	1	796	796	0	4.1976	0.00000	/VEL
5	1	797	797	0	4.1976	0.00000	/VEL
5	1	798	798	0	4.1454	0.00000	/VEL
5	1	799	799	0	4.0951	0.00000	/VEL
5	1	800	800	0	4.0512	0.00000	/VEL
5	1	801	801	0	3.9965	0.00000	/VEL
5	1	802	802	0	3.9527	0.00000	/VEL
5	1	803	803	0	3.8999	0.00000	/VEL
5	1	804	804	0	3.8562	0.00000	/VEL
5	1	805	805	0	3.8562	0.00000	/VEL
5	1	806	806	0	3.8098	0.00000	/VEL
5	1	807	807	0	3.7635	0.00000	/VEL
5	1	808	808	0	3.8886	0.00000	/VEL
5	1	809	809	0	4.4222	0.00000	/VEL
5	1	810	810	0	4.3785	0.00000	/VEL
5	1	811	811	0	4.3785	0.00000	/VEL
5	1	812	812	0	4.3379	0.00000	/VEL
5	1	813	813	0	4.2957	0.00000	/VEL
5	1	814	814	0	4.2521	0.00000	/VEL
5	1	815	815	0	4.2521	0.00000	/VEL
5	1	816	816	0	4.2176	0.00000	/VEL
5	1	817	817	0	4.1754	0.00000	/VEL
5	1	818	818	0	4.1754	0.00000	/VEL
5	1	819	819	0	4.1301	0.00000	/VEL
5	1	820	820	0	4.0941	0.00000	/VEL
5	1	821	821	0	4.0535	0.00000	/VEL
5	1	822	822	0	4.0535	0.00000	/VEL
5	1	823	823	0	4.0190	0.00000	/VEL
5	1	824	824	0	3.9737	0.00000	/VEL
5	1	825	825	0	3.9407	0.00000	/VEL
5	1	826	826	0	3.9407	0.00000	/VEL
5	1	827	827	0	3.9048	0.00000	/VEL
5	1	828	828	0	3.8610	0.00000	/VEL
5	1	829	829	0	4.6198	0.00000	/VEL
5	1	830	830	0	4.6259	0.00000	/VEL
5	1	831	831	0	4.6211	0.00000	/VEL
5	1	832	832	0	4.6183	0.00000	/VEL
5	1	833	833	0	4.6183	0.00000	/VEL
5	1	834	834	0	4.6254	0.00000	/VEL
5	1	835	835	0	4.6295	0.00000	/VEL
5	1	836	836	0	4.4685	0.00000	/VEL
5	1	837	837	0	4.4867	0.00000	/VEL
5	1	838	838	0	4.4951	0.00000	/VEL
5	1	839	839	0	4.5134	0.00000	/VEL
5	1	840	840	0	4.5203	0.00000	/VEL
5	1	841	841	0	4.5317	0.00000	/VEL
5	1	842	842	0	4.5386	0.00000	/VEL
5	1	843	843	0	4.5569	0.00000	/VEL
5	1	844	844	0	4.5569	0.00000	/VEL
5	1	845	845	0	4.5753	0.00000	/VEL
5	1	846	846	0	4.5753	0.00000	/VEL
5	1	847	847	0	4.5922	0.00000	/VEL
5	1	848	848	0	4.6091	0.00000	/VEL
5	1	849	849	0	4.6121	0.00000	/VEL
5	1	850	850	0	4.6361	0.00000	/VEL
5	1	851	851	0	4.6375	0.00000	/VEL
5	1	852	852	0	4.6545	0.00000	/VEL
5	1	853	853	0	4.6645	0.00000	/VEL
5	1	854	854	0	4.6645	0.00000	/VEL
5	1	855	855	0	4.6816	0.00000	/VEL
5	1	856	856	0	4.6901	0.00000	/VEL
5	1	857	857	0	4.7086	0.00000	/VEL
5	1	858	858	0	4.9806	0.00000	/VEL
5	1	859	859	0	4.6812	0.00000	/VEL
5	1	860	860	0	4.5231	0.00000	/VEL
5	1	861	861	0	4.5231	0.00000	/VEL
5	1	862	862	0	4.3877	0.00000	/VEL
5	1	863	863	0	4.2454	0.00000	/VEL
5	1	864	864	0	4.2454	0.00000	/VEL
5	1	865	865	0	4.1118	0.00000	/VEL
5	1	866	866	0	3.9870	0.00000	/VEL
5	1	867	867	0	3.8613	0.00000	/VEL
5	1	868	868	0	3.7596	0.00000	/VEL
5	1	869	869	0	3.7596	0.00000	/VEL

5	1	870	870	0	3.6370	0.00000	/VEL
5	1	871	871	0	3.5428	0.00000	/VEL
5	1	872	872	0	3.4378	0.00000	/VEL
5	1	873	873	0	3.3519	0.00000	/VEL
5	1	874	874	0	3.2557	0.00000	/VEL
5	1	875	875	0	3.1731	0.00000	/VEL
5	1	876	876	0	3.0905	0.00000	/VEL
5	1	877	877	0	3.0187	0.00000	/VEL
5	1	878	878	0	2.9443	0.00000	/VEL
5	1	879	879	0	2.8795	0.00000	/VEL
5	1	880	880	0	2.8158	0.00000	/VEL
5	1	881	881	0	2.7520	0.00000	/VEL
5	1	882	882	0	2.2306	0.00000	/VEL
5	1	883	883	0	2.2096	0.00000	/VEL
5	1	884	884	0	2.1843	0.00000	/VEL
5	1	885	885	0	2.1587	0.00000	/VEL
5	1	886	886	0	3.2147	0.00000	/VEL
5	1	887	887	0	3.2602	0.00000	/VEL
5	1	888	888	0	3.3141	0.00000	/VEL
5	1	889	889	0	3.5705	0.00000	/VEL
5	1	890	890	0	3.6183	0.00000	/VEL
5	1	891	891	0	3.6538	0.00000	/VEL
5	1	892	892	0	3.7032	0.00000	/VEL
5	1	893	893	0	3.7032	0.00000	/VEL
5	1	894	894	0	3.7512	0.00000	/VEL
5	1	895	895	0	3.7913	0.00000	/VEL
5	1	896	896	0	3.8409	0.00000	/VEL
5	1	897	897	0	3.8867	0.00000	/VEL
5	1	898	898	0	3.9325	0.00000	/VEL
5	1	899	899	0	3.9325	0.00000	/VEL
5	1	900	900	0	3.9866	0.00000	/VEL
5	1	901	901	0	4.0327	0.00000	/VEL
5	1	902	902	0	4.0833	0.00000	/VEL
5	1	903	903	0	4.0833	0.00000	/VEL
5	1	904	904	0	4.1377	0.00000	/VEL
5	1	905	905	0	4.1885	0.00000	/VEL
5	1	906	906	0	4.2394	0.00000	/VEL
5	1	907	907	0	4.2394	0.00000	/VEL
5	1	908	908	0	4.0937	0.00000	/VEL
5	1	909	909	0	4.1022	0.00000	/VEL
5	1	910	910	0	4.1041	0.00000	/VEL
5	1	911	911	0	3.9950	0.00000	/VEL
5	1	912	912	0	3.9950	0.00000	/VEL
5	1	913	913	0	4.0008	0.00000	/VEL
5	1	914	914	0	4.0053	0.00000	/VEL
5	1	915	915	0	4.0053	0.00000	/VEL
5	1	916	916	0	4.0082	0.00000	/VEL
5	1	917	917	0	4.0127	0.00000	/VEL
5	1	918	918	0	4.0141	0.00000	/VEL
5	1	919	919	0	4.0141	0.00000	/VEL
5	1	920	920	0	4.0201	0.00000	/VEL
5	1	921	921	0	4.0246	0.00000	/VEL
5	1	922	922	0	4.0246	0.00000	/VEL
5	1	923	923	0	4.0274	0.00000	/VEL
5	1	924	924	0	4.0274	0.00000	/VEL
5	1	925	925	0	4.0348	0.00000	/VEL
5	1	926	926	0	4.0348	0.00000	/VEL
5	1	927	927	0	4.0394	0.00000	/VEL
5	1	928	928	0	4.0394	0.00000	/VEL
5	1	929	929	0	4.0468	0.00000	/VEL
5	1	930	930	0	4.0468	0.00000	/VEL
5	1	931	931	0	4.0482	0.00000	/VEL
5	1	932	932	0	5.9517	0.00000	/VEL
5	1	933	933	0	5.9512	0.00000	/VEL
5	1	934	934	0	5.9493	0.00000	/VEL
5	1	935	935	0	5.9493	0.00000	/VEL
5	1	936	936	0	6.2349	0.00000	/VEL
5	1	937	937	0	6.2492	0.00000	/VEL
5	1	938	938	0	6.2433	0.00000	/VEL
5	1	939	939	0	6.2687	0.00000	/VEL
5	1	940	940	0	6.2629	0.00000	/VEL
5	1	941	941	0	6.2768	0.00000	/VEL
5	1	942	942	0	6.2946	0.00000	/VEL
5	1	943	943	0	6.2889	0.00000	/VEL
5	1	944	944	0	6.3150	0.00000	/VEL

5	1	945	945	0	6.3150	0.00000	/VEL
5	1	946	946	0	6.3093	0.00000	/VEL
5	1	947	947	0	6.3375	0.00000	/VEL
5	1	948	948	0	6.3319	0.00000	/VEL
5	1	949	949	0	6.3484	0.00000	/VEL
5	1	950	950	0	6.3650	0.00000	/VEL
5	1	951	951	0	6.3715	0.00000	/VEL
5	1	952	952	0	6.3882	0.00000	/VEL
5	1	953	953	0	6.3930	0.00000	/VEL
5	1	954	954	0	6.3930	0.00000	/VEL
5	1	955	955	0	6.3997	0.00000	/VEL
5	1	956	956	0	6.4185	0.00000	/VEL
5	1	957	957	0	6.4235	0.00000	/VEL
5	1	958	958	0	6.5852	0.00000	/VEL
5	1	959	959	0	6.5816	0.00000	/VEL
5	1	960	960	0	6.5784	0.00000	/VEL
5	1	961	961	0	6.5904	0.00000	/VEL
5	1	962	962	0	6.5871	0.00000	/VEL
5	1	963	963	0	6.5855	0.00000	/VEL
5	1	964	964	0	6.5976	0.00000	/VEL
5	1	965	965	0	6.6112	0.00000	/VEL
5	1	966	966	0	6.6080	0.00000	/VEL
5	1	967	967	0	6.6049	0.00000	/VEL
5	1	968	968	0	6.0842	0.00000	/VEL
5	1	969	969	0	6.0677	0.00000	/VEL
5	1	970	970	0	6.0811	0.00000	/VEL
5	1	971	971	0	6.0647	0.00000	/VEL
5	1	972	972	0	6.0781	0.00000	/VEL
5	1	973	973	0	6.0632	0.00000	/VEL
5	1	974	974	0	6.0602	0.00000	/VEL
5	1	975	975	0	6.0602	0.00000	/VEL
5	1	976	976	0	6.0721	0.00000	/VEL
5	1	977	977	0	6.0572	0.00000	/VEL
5	1	978	978	0	6.0542	0.00000	/VEL
5	1	979	979	0	6.0542	0.00000	/VEL
5	1	980	980	0	6.0527	0.00000	/VEL
5	1	981	981	0	6.0512	0.00000	/VEL
5	1	982	982	0	6.0363	0.00000	/VEL
5	1	983	983	0	6.0363	0.00000	/VEL
5	1	984	984	0	6.0483	0.00000	/VEL
5	1	985	985	0	6.0468	0.00000	/VEL
5	1	986	986	0	6.0453	0.00000	/VEL
5	1	987	987	0	6.0304	0.00000	/VEL
5	1	988	988	0	6.0424	0.00000	/VEL
5	1	989	989	0	6.0275	0.00000	/VEL
5	1	990	990	0	6.1963	0.00000	/VEL
6	1	1	695	-1	100.00	1000.0	7 /DISP
6	1	696	768	-2	100.00	1000.0	5 /DISP
6	1	769	784	-1	100.00	1000.0	7 /DISP
6	1	785	872	-2	100.00	1000.0	5 /DISP
6	1	873	991	-3	50.000	500.00	20 /DISP
7	1	1	695	0	0.47300E-01	0.00000	7 /DM
7	1	696	768	0	0.47300E-01	0.00000	5 /DM
7	1	769	784	0	0.47300E-01	0.00000	7 /DM
7	1	785	872	0	0.47300E-01	0.00000	5 /DM
7	1	873	991	0	0.47300E-01	0.00000	20 /DM
8	1	1	695	0	0.10000	0.00000	7 /TAU
8	1	696	768	0	0.10000	0.00000	5 /TAU
8	1	769	784	0	0.10000	0.00000	7 /TAU
8	1	785	872	0	0.10000	0.00000	5 /TAU
8	1	873	991	0	0.10000	0.00000	20 /TAU
9	1	1	695	-4	-2.4600	0.25000	7 /THETA1
9	1	696	768	-5	-2.4600	0.25000	5 /THETA1
9	1	769	784	-4	-2.4600	0.25000	7 /THETA1
9	1	785	872	-5	-2.4600	0.25000	5 /THETA1
9	1	873	991	-6	0.25000	0.35000	20 /THETA1
10	1	1	695	-7	0.10000E-01	0.15000	7 /THETA2
10	1	696	768	-8	0.10000E-01	0.15000	5 /THETA2
10	1	769	784	-7	0.10000E-01	0.15000	7 /THETA2
10	1	785	872	-8	0.10000E-01	0.15000	5 /THETA2
10	1	873	991	0	0.00000	0.00000	20 /THETA2
11	1	1	695	-9	-0.67000	0.26000	7 /BWIDTH
11	1	696	768	-10	-0.67000	0.26000	5 /BWIDTH
11	1	769	784	-9	-0.67000	0.26000	7 /BWIDTH
11	1	785	872	-10	-0.67000	0.26000	5 /BWIDTH

11	1	873	991	0	1.0000	0.00000	20	/BWIDTH
12	1	1	695	0	2.0000	0.00000	7	/FWIDTH
12	1	696	768	0	2.0000	0.00000	5	/FWIDTH
12	1	769	784	0	2.0000	0.00000	7	/FWIDTH
12	1	785	872	0	2.0000	0.00000	5	/FWIDTH
12	1	873	991	0	2.0000	0.00000	20	/FWIDTH
13	1	1	695	0	0.00000	0.00000	7	/RHOWB
13	1	696	768	0	0.00000	0.00000	5	/RHOWB
13	1	769	784	0	0.00000	0.00000	7	/RHOWB
13	1	785	872	0	0.00000	0.00000	5	/RHOWB
13	1	873	991	0	0.00000	0.00000	20	/RHOWB
14	1	1	695	0	0.00000	0.00000	7	/RKD1
14	1	696	768	0	0.00000	0.00000	5	/RKD1
14	1	769	784	0	0.00000	0.00000	7	/RKD1
14	1	785	872	0	0.00000	0.00000	5	/RKD1
14	1	873	991	0	0.00000	0.00000	20	/RKD1
15	1	1	695	0	0.00000	0.00000	7	/RKD2
15	1	696	768	0	0.00000	0.00000	5	/RKD2
15	1	769	784	0	0.00000	0.00000	7	/RKD2
15	1	785	872	0	0.00000	0.00000	5	/RKD2
15	1	873	991	0	0.00000	0.00000	20	/RKD2
16	1	1	695	-11	-2.8420	0.24000	7	/DIFF
16	1	696	768	-12	-2.8420	0.24000	5	/DIFF
16	1	769	784	-11	-2.8420	0.24000	7	/DIFF
16	1	785	872	-12	-2.8420	0.24000	5	/DIFF
16	1	873	991	0	0.00000	0.00000	20	/DIFF
17	1	1	695	0	0.05640	0.00000	7	/LAMBDA1W
17	1	696	768	0	0.05640	0.00000	5	/LAMBDA1W
17	1	769	784	0	0.05640	0.00000	7	/LAMBDA1W
17	1	785	872	0	0.05640	0.00000	5	/LAMBDA1W
17	1	873	991	0	0.05640	0.00000	20	/LAMBDA1W
18	1	1	1	-13	-3.0700	0.25000		/KH_01_1
19	1	1	1	-14	-1.1700	0.25000		/KH_06_1
20	1	1	1	-14	-0.77000	0.25000		/KH_06_2
21	1	1	1	-14	-1.0700	0.25000		/KH_06_3
22	1	1	1	-15	1.9700	0.25000		/KH_07_1
23	1	1	1	-15	1.9700	0.25000		/KH_07_2
24	1	1	1	-15	-2.0700	0.25000		/KH_07_3
25	1	1	1	-15	0.83000	0.25000		/KH_07_4
26	1	1	1	-15	-0.17000	0.25000		/KH_07_5
27	1	1	1	-15	-0.47000	0.25000		/KH_07_6
28	1	1	1	-15	0.23000	0.25000		/KH_07_7
29	1	1	1	-15	-0.17000	0.25000		/KH_07_8
30	1	1	1	-15	0.63000	0.25000		/KH_07_9
31	1	1	1	-16	-2.7700	0.25000		/KH_08_1
32	1	1	1	-17	-1.3700	0.25000		/KH_09_1
33	1	1	1	-18	-0.37000	0.25000		/KH_10_1
34	1	1	1	-18	-0.77000	0.25000		/KH_10_2
35	1	1	1	-19	0.10000	0.25000		/KH_11_1
36	1	1	1	-19	-0.77000	0.25000		/KH_11_2
37	1	1	1	-19	-0.47000	0.25000		/KH_11_3
38	1	1	1	-20	-0.99000	0.25000		/KH_12_1
39	1	1	1	-20	-0.99000	0.25000		/KH_12_2
40	1	1	1	-21	0.23000	0.25000		/KH_13_1
41	1	1	1	-21	-0.70000E-01	0.25000		/KH_13_2
42	1	1	1	-21	0.83000	0.25000		/KH_13_3
43	1	1	1	-22	1.4700	0.25000		/KH_14_1
44	1	1	1	-22	0.23000	0.25000		/KH_14_2
45	1	1	1	-22	0.23000	0.25000		/KH_14_3
46	1	1	1	-22	0.23000	0.25000		/KH_14_4
47	1	1	1	-22	-0.77000	0.25000		/KH_14_5
48	1	1	1	-23	-0.59000	0.25000		/KH_15_1
49	1	1	1	-23	-0.59000	0.25000		/KH_15_2
50	1	1	1	-23	-0.59000	0.25000		/KH_15_3
51	1	1	1	-23	-0.59000	0.25000		/KH_15_4
52	1	1	1	-23	-0.77000	0.25000		/KH_15_5
53	1	1	1	-24	1.4700	0.25000		/KH_16_1
54	1	1	1	-24	0.23000	0.25000		/KH_16_2
55	1	1	1	-24	0.23000	0.25000		/KH_16_3
56	1	1	1	-24	-0.77000	0.25000		/KH_16_4
57	1	1	1	-25	-0.59000	0.25000		/KH_17_1
58	1	1	1	-25	-1.5900	0.25000		/KH_17_2
59	1	1	1	-25	-1.7700	0.25000		/KH_17_3
60	1	1	1	-26	1.4700	0.25000		/KH_18_1
61	1	1	1	-26	1.1300	0.25000		/KH_18_2

62	1	1	1	-26	1.2300	0.25000	/KH_18_3	
63	1	1	1	-26	0.53000	0.25000	/KH_18_4	
64	1	1	1	-26	-0.77000	0.25000	/KH_18_5	
65	1	1	1	-27	1.4700	0.25000	/KH_19_1	
66	1	1	1	-27	1.0000	0.25000	/KH_19_2	
67	1	1	1	-27	1.1300	0.25000	/KH_19_3	
68	1	1	1	-27	1.2300	0.25000	/KH_19_4	
69	1	1	1	-27	-0.77000	0.25000	/KH_19_5	
70	1	1	1	-28	0.97000	0.25000	/KH_20_1	
71	1	1	1	-28	0.57000	0.25000	/KH_20_2	
72	1	1	1	-28	-2.0700	0.25000	/KH_20_3	
73	1	1	1	-28	0.97000	0.25000	/KH_20_4	
74	1	1	1	-28	0.97000	0.25000	/KH_20_5	
75	1	1	1	-28	0.97000	0.25000	/KH_20_6	
76	1	1	1	-29	0.00000	0.30000E-02	/LD_01_1	
77	1	1	1	-30	0.00000	0.24000E-02	/LD_06_1	
78	1	1	1	-30	0.00000	0.24000E-02	/LD_06_2	
79	1	1	1	-30	0.00000	0.24000E-02	/LD_06_3	
80	1	1	1	-31	0.00000	0.20000E-02	/LD_07_1	
81	1	1	1	-31	0.00000	0.20000E-02	/LD_07_2	
82	1	1	1	-31	0.00000	0.20000E-02	/LD_07_3	
83	1	1	1	-31	0.00000	0.20000E-02	/LD_07_4	
84	1	1	1	-31	0.00000	0.20000E-02	/LD_07_5	
85	1	1	1	-31	0.00000	0.20000E-02	/LD_07_6	
86	1	1	1	-31	0.00000	0.52000E-02	/LD_07_7	
87	1	1	1	-31	0.00000	0.20000E-02	/LD_07_8	
88	1	1	1	-31	0.00000	0.20000E-02	/LD_07_9	
89	1	1	1	-32	0.00000	0.30000E-02	/LD_08_1	
90	1	1	1	-33	0.00000	0.20000E-02	/LD_09_1	
91	1	1	1	-34	0.00000	0.80000E-02	/LD_10_1	
92	1	1	1	-34	0.00000	0.80000E-02	/LD_10_2	
93	1	1	1	-35	0.70000E-03	0.45000E-02	/LD_11_1	
94	1	1	1	-35	0.70000E-03	0.45000E-02	/LD_11_2	
95	1	1	1	-35	0.70000E-03	0.45000E-02	/LD_11_3	
96	1	1	1	-36	0.70000E-03	0.45000E-02	/LD_12_1	
97	1	1	1	-36	0.70000E-03	0.45000E-02	/LD_12_2	
98	1	1	1	-37	0.70000E-03	0.45000E-02	/LD_13_1	
99	1	1	1	-37	0.70000E-03	0.45000E-02	/LD_13_2	
100	1	1	1	-37	0.70000E-03	0.45000E-02	/LD_13_3	
101	1	1	1	-38	0.70000E-03	0.45000E-02	/LD_14_1	
102	1	1	1	-38	0.70000E-03	0.45000E-02	/LD_14_2	
103	1	1	1	-38	0.70000E-03	0.45000E-02	/LD_14_3	
104	1	1	1	-38	0.70000E-03	0.45000E-02	/LD_14_4	
105	1	1	1	-38	0.70000E-03	0.45000E-02	/LD_14_5	
106	1	1	1	-39	0.70000E-03	0.45000E-02	/LD_15_1	
107	1	1	1	-39	0.70000E-03	0.45000E-02	/LD_15_2	
108	1	1	1	-39	0.70000E-03	0.45000E-02	/LD_15_3	
109	1	1	1	-39	0.70000E-03	0.45000E-02	/LD_15_4	
110	1	1	1	-39	0.70000E-03	0.45000E-02	/LD_15_5	
111	1	1	1	-40	0.70000E-03	0.45000E-02	/LD_16_1	
112	1	1	1	-40	0.70000E-03	0.45000E-02	/LD_16_2	
113	1	1	1	-40	0.70000E-03	0.45000E-02	/LD_16_3	
114	1	1	1	-40	0.70000E-03	0.45000E-02	/LD_16_4	
115	1	1	1	-41	0.70000E-03	0.45000E-02	/LD_17_1	
116	1	1	1	-41	0.70000E-03	0.45000E-02	/LD_17_2	
117	1	1	1	-41	0.70000E-03	0.45000E-02	/LD_17_3	
118	1	1	1	-42	0.70000E-03	0.45000E-02	/LD_18_1	
119	1	1	1	-42	0.70000E-03	0.45000E-02	/LD_18_2	
120	1	1	1	-42	0.70000E-03	0.45000E-02	/LD_18_3	
121	1	1	1	-42	0.70000E-03	0.45000E-02	/LD_18_4	
122	1	1	1	-42	0.70000E-03	0.45000E-02	/LD_18_5	
123	1	1	1	-43	0.70000E-03	0.45000E-02	/LD_19_1	
124	1	1	1	-43	0.70000E-03	0.45000E-02	/LD_19_2	
125	1	1	1	-43	0.70000E-03	0.45000E-02	/LD_19_3	
126	1	1	1	-43	0.70000E-03	0.45000E-02	/LD_19_4	
127	1	1	1	-43	0.70000E-03	0.45000E-02	/LD_19_5	
128	1	1	1	-44	0.00000	0.74000E-02	/LD_20_1	
129	1	1	1	-44	0.00000	0.74000E-02	/LD_20_2	
130	1	1	1	-44	0.00000	0.74000E-02	/LD_20_3	
131	1	1	1	-44	0.00000	0.74000E-02	/LD_20_4	
132	1	1	1	-44	0.00000	0.74000E-02	/LD_20_5	
133	1	1	1	-44	0.00000	0.74000E-02	/LD_20_6	
134	1	1	1	-45	0.00000	0.25000	/RC_01_1	
1	1	1	1	0	29	0	1	29

200	2243					
501		100.000	4800	0.04167	0.	-1.0

/NPARD, NSPEC, NFSP

2	11	21	31	41	51	61	71	81	91
101	151	201	251	301	351	401	451	501	

[illegible]

5	1	56	56	0	3.0898	0.00000	/VEL
5	1	57	57	0	3.0373	0.00000	/VEL
5	1	58	58	0	2.9922	0.00000	/VEL
5	1	59	59	0	2.9545	0.00000	/VEL
5	1	60	60	0	2.9022	0.00000	/VEL
5	1	61	61	0	2.8568	0.00000	/VEL
5	1	62	62	0	2.8192	0.00000	/VEL
5	1	63	63	0	2.7669	0.00000	/VEL
5	1	64	64	0	2.7292	0.00000	/VEL
5	1	65	65	0	2.8221	0.00000	/VEL
5	1	66	66	0	2.8221	0.00000	/VEL
5	1	67	67	0	2.7327	0.00000	/VEL
5	1	68	68	0	2.6470	0.00000	/VEL
5	1	69	69	0	2.5669	0.00000	/VEL
5	1	70	70	0	2.4869	0.00000	/VEL
5	1	71	71	0	2.4126	0.00000	/VEL
5	1	72	72	0	2.3384	0.00000	/VEL
5	1	73	73	0	2.2641	0.00000	/VEL
5	1	74	74	0	2.1956	0.00000	/VEL
5	1	75	75	0	2.1271	0.00000	/VEL
5	1	76	76	0	2.0643	0.00000	/VEL
5	1	77	77	0	2.0015	0.00000	/VEL
5	1	78	78	0	1.9388	0.00000	/VEL
5	1	79	79	0	1.8760	0.00000	/VEL
5	1	80	80	0	1.9156	0.00000	/VEL
5	1	81	81	0	1.9156	0.00000	/VEL
5	1	82	82	0	1.8789	0.00000	/VEL
5	1	83	83	0	1.8389	0.00000	/VEL
5	1	84	84	0	1.7989	0.00000	/VEL
5	1	85	85	0	1.7640	0.00000	/VEL
5	1	86	86	0	1.7292	0.00000	/VEL
5	1	87	87	0	1.6946	0.00000	/VEL
5	1	88	88	0	1.6646	0.00000	/VEL
5	1	89	89	0	1.6311	0.00000	/VEL
5	1	90	90	0	1.3842	0.00000	/VEL
5	1	91	91	0	1.3791	0.00000	/VEL
5	1	92	92	0	1.3646	0.00000	/VEL
5	1	93	93	0	1.3563	0.00000	/VEL
5	1	94	94	0	1.3483	0.00000	/VEL
5	1	95	95	0	1.3340	0.00000	/VEL
5	1	96	96	0	1.3288	0.00000	/VEL
5	1	97	97	0	1.2585	0.00000	/VEL
5	1	98	98	0	1.2478	0.00000	/VEL
5	1	99	99	0	1.2404	0.00000	/VEL
5	1	100	100	0	1.2354	0.00000	/VEL
5	1	101	101	0	1.2253	0.00000	/VEL
5	1	102	102	0	1.2177	0.00000	/VEL
5	1	103	103	0	1.2073	0.00000	/VEL
5	1	104	104	0	1.2027	0.00000	/VEL
5	1	105	105	0	1.1952	0.00000	/VEL
5	1	106	106	0	1.1850	0.00000	/VEL
5	1	107	107	0	1.1850	0.00000	/VEL
5	1	108	108	0	1.1803	0.00000	/VEL
5	1	109	109	0	1.1729	0.00000	/VEL
5	1	110	110	0	1.1655	0.00000	/VEL
5	1	111	111	0	1.1581	0.00000	/VEL
5	1	112	112	0	1.1510	0.00000	/VEL
5	1	113	113	0	1.1423	0.00000	/VEL
5	1	114	114	0	1.1394	0.00000	/VEL
5	1	115	115	0	1.1421	0.00000	/VEL
5	1	116	116	0	1.1394	0.00000	/VEL
5	1	117	117	0	1.1394	0.00000	/VEL
5	1	118	118	0	1.1366	0.00000	/VEL
5	1	119	119	0	1.1366	0.00000	/VEL
5	1	120	120	0	1.1364	0.00000	/VEL
5	1	121	121	0	1.1338	0.00000	/VEL
5	1	122	122	0	1.1338	0.00000	/VEL
5	1	123	123	0	1.1338	0.00000	/VEL
5	1	124	124	0	1.1336	0.00000	/VEL
5	1	125	125	0	1.1308	0.00000	/VEL
5	1	126	126	0	1.1283	0.00000	/VEL
5	1	127	127	0	1.1308	0.00000	/VEL
5	1	128	128	0	1.1815	0.00000	/VEL
5	1	129	129	0	1.1867	0.00000	/VEL
5	1	130	130	0	1.1920	0.00000	/VEL

5	1	131	131	0	1.2003	0.00000	/VEL
5	1	132	132	0	1.2027	0.00000	/VEL
5	1	133	133	0	1.2079	0.00000	/VEL
5	1	134	134	0	1.2910	0.00000	/VEL
5	1	135	135	0	1.2964	0.00000	/VEL
5	1	136	136	0	1.3058	0.00000	/VEL
5	1	137	137	0	1.3150	0.00000	/VEL
5	1	138	138	0	1.3275	0.00000	/VEL
5	1	139	139	0	1.3334	0.00000	/VEL
5	1	140	140	0	1.3460	0.00000	/VEL
5	1	141	141	0	1.3553	0.00000	/VEL
5	1	142	142	0	1.3646	0.00000	/VEL
5	1	143	143	0	1.3772	0.00000	/VEL
5	1	144	144	0	1.4292	0.00000	/VEL
5	1	145	145	0	1.4387	0.00000	/VEL
5	1	146	146	0	1.4419	0.00000	/VEL
5	1	147	147	0	1.4515	0.00000	/VEL
5	1	148	148	0	1.4579	0.00000	/VEL
5	1	149	149	0	1.4679	0.00000	/VEL
5	1	150	150	0	1.4707	0.00000	/VEL
5	1	151	151	0	1.4806	0.00000	/VEL
5	1	152	152	0	1.4870	0.00000	/VEL
5	1	153	153	0	1.4971	0.00000	/VEL
5	1	154	154	0	1.5036	0.00000	/VEL
5	1	155	155	0	1.5100	0.00000	/VEL
5	1	156	156	0	1.5166	0.00000	/VEL
5	1	157	157	0	1.5266	0.00000	/VEL
5	1	158	158	0	1.5266	0.00000	/VEL
5	1	159	159	0	1.5331	0.00000	/VEL
5	1	160	160	0	1.5608	0.00000	/VEL
5	1	161	161	0	1.5438	0.00000	/VEL
5	1	162	162	0	1.5306	0.00000	/VEL
5	1	163	163	0	1.5137	0.00000	/VEL
5	1	164	164	0	1.5004	0.00000	/VEL
5	1	165	165	0	1.4872	0.00000	/VEL
5	1	166	166	0	1.4706	0.00000	/VEL
5	1	167	167	0	1.4573	0.00000	/VEL
5	1	168	168	0	1.4441	0.00000	/VEL
5	1	169	169	0	1.4278	0.00000	/VEL
5	1	170	170	0	1.4181	0.00000	/VEL
5	1	171	171	0	1.4016	0.00000	/VEL
5	1	172	172	0	1.3888	0.00000	/VEL
5	1	173	173	0	1.3791	0.00000	/VEL
5	1	174	174	0	1.3630	0.00000	/VEL
5	1	175	175	0	1.3317	0.00000	/VEL
5	1	176	176	0	1.3109	0.00000	/VEL
5	1	177	177	0	1.2902	0.00000	/VEL
5	1	178	178	0	1.2695	0.00000	/VEL
5	1	179	179	0	1.2488	0.00000	/VEL
5	1	180	180	0	1.2488	0.00000	/VEL
5	1	181	181	0	1.2311	0.00000	/VEL
5	1	182	182	0	1.2104	0.00000	/VEL
5	1	183	183	0	1.1927	0.00000	/VEL
5	1	184	184	0	1.1721	0.00000	/VEL
5	1	185	185	0	1.1545	0.00000	/VEL
5	1	186	186	0	1.1369	0.00000	/VEL
5	1	187	187	0	1.1016	0.00000	/VEL
5	1	188	188	0	1.0840	0.00000	/VEL
5	1	189	189	0	1.0665	0.00000	/VEL
5	1	190	190	0	1.0982	0.00000	/VEL
5	1	191	191	0	1.1007	0.00000	/VEL
5	1	192	192	0	1.1007	0.00000	/VEL
5	1	193	193	0	1.1009	0.00000	/VEL
5	1	194	194	0	1.1009	0.00000	/VEL
5	1	195	195	0	1.1007	0.00000	/VEL
5	1	196	196	0	1.1010	0.00000	/VEL
5	1	197	197	0	1.1036	0.00000	/VEL
5	1	198	198	0	1.1036	0.00000	/VEL
5	1	199	199	0	1.1037	0.00000	/VEL
5	1	200	200	0	1.1037	0.00000	/VEL
5	1	201	201	0	1.1064	0.00000	/VEL
5	1	202	202	0	1.1037	0.00000	/VEL
5	1	203	203	0	1.1065	0.00000	/VEL
5	1	204	204	0	1.1065	0.00000	/VEL
5	1	205	205	0	1.2268	0.00000	/VEL

5	1	206	206	0	1.2428	0.00000	/VEL
5	1	207	207	0	1.2560	0.00000	/VEL
5	1	208	208	0	1.2693	0.00000	/VEL
5	1	209	209	0	1.2830	0.00000	/VEL
5	1	210	210	0	1.2992	0.00000	/VEL
5	1	211	211	0	1.3133	0.00000	/VEL
5	1	212	212	0	1.3326	0.00000	/VEL
5	1	213	213	0	1.3438	0.00000	/VEL
5	1	214	214	0	1.3636	0.00000	/VEL
5	1	215	215	0	1.3807	0.00000	/VEL
5	1	216	216	0	1.3951	0.00000	/VEL
5	1	217	217	0	1.3951	0.00000	/VEL
5	1	218	218	0	1.4151	0.00000	/VEL
5	1	219	219	0	1.4296	0.00000	/VEL
5	1	220	220	0	1.4500	0.00000	/VEL
5	1	221	221	0	1.8487	0.00000	/VEL
5	1	222	222	0	1.8688	0.00000	/VEL
5	1	223	223	0	1.8932	0.00000	/VEL
5	1	224	224	0	1.9234	0.00000	/VEL
5	1	225	225	0	1.9527	0.00000	/VEL
5	1	226	226	0	1.9871	0.00000	/VEL
5	1	227	227	0	2.0298	0.00000	/VEL
5	1	228	228	0	2.0734	0.00000	/VEL
5	1	229	229	0	2.1195	0.00000	/VEL
5	1	230	230	0	2.1668	0.00000	/VEL
5	1	231	231	0	2.2253	0.00000	/VEL
5	1	232	232	0	2.2808	0.00000	/VEL
5	1	233	233	0	2.3419	0.00000	/VEL
5	1	234	234	0	2.4094	0.00000	/VEL
5	1	235	235	0	2.4094	0.00000	/VEL
5	1	236	236	0	2.4726	0.00000	/VEL
5	1	237	237	0	1.6251	0.00000	/VEL
5	1	238	238	0	1.6251	0.00000	/VEL
5	1	239	239	0	1.6235	0.00000	/VEL
5	1	240	240	0	1.6208	0.00000	/VEL
5	1	241	241	0	1.6208	0.00000	/VEL
5	1	242	242	0	1.6155	0.00000	/VEL
5	1	243	243	0	1.6140	0.00000	/VEL
5	1	244	244	0	1.6095	0.00000	/VEL
5	1	245	245	0	1.6095	0.00000	/VEL
5	1	246	246	0	1.6080	0.00000	/VEL
5	1	247	247	0	1.6042	0.00000	/VEL
5	1	248	248	0	1.6004	0.00000	/VEL
5	1	249	249	0	1.6004	0.00000	/VEL
5	1	250	250	0	1.5990	0.00000	/VEL
5	1	251	251	0	1.5953	0.00000	/VEL
5	1	252	252	0	1.5916	0.00000	/VEL
5	1	253	253	0	1.5916	0.00000	/VEL
5	1	254	254	0	1.5903	0.00000	/VEL
5	1	255	255	0	1.5867	0.00000	/VEL
5	1	256	256	0	1.5849	0.00000	/VEL
5	1	257	257	0	1.5849	0.00000	/VEL
5	1	258	258	0	1.5638	0.00000	/VEL
5	1	259	259	0	1.4580	0.00000	/VEL
5	1	260	260	0	1.4602	0.00000	/VEL
5	1	261	261	0	1.4634	0.00000	/VEL
5	1	262	262	0	1.4634	0.00000	/VEL
5	1	263	263	0	1.4643	0.00000	/VEL
5	1	264	264	0	1.4664	0.00000	/VEL
5	1	265	265	0	1.4724	0.00000	/VEL
5	1	266	266	0	1.4759	0.00000	/VEL
5	1	267	267	0	0.61018	0.00000	/VEL
5	1	268	268	0	0.61426	0.00000	/VEL
5	1	269	269	0	0.61917	0.00000	/VEL
5	1	270	270	0	0.61917	0.00000	/VEL
5	1	271	271	0	0.62386	0.00000	/VEL
5	1	272	272	0	0.63026	0.00000	/VEL
5	1	273	273	0	0.63555	0.00000	/VEL
5	1	274	274	0	0.64131	0.00000	/VEL
5	1	275	275	0	0.64837	0.00000	/VEL
5	1	276	276	0	0.64837	0.00000	/VEL
5	1	277	277	0	0.73973	0.00000	/VEL
5	1	278	278	0	0.74258	0.00000	/VEL
5	1	279	279	0	0.74491	0.00000	/VEL
5	1	280	280	0	0.74811	0.00000	/VEL

5	1	281	281	0	0.74811	0.00000	/VEL
5	1	282	282	0	0.75050	0.00000	/VEL
5	1	283	283	0	0.75294	0.00000	/VEL
5	1	284	284	0	0.81122	0.00000	/VEL
5	1	285	285	0	0.81558	0.00000	/VEL
5	1	286	286	0	0.81951	0.00000	/VEL
5	1	287	287	0	0.82491	0.00000	/VEL
5	1	288	288	0	0.83029	0.00000	/VEL
5	1	289	289	0	0.83444	0.00000	/VEL
5	1	290	290	0	0.84039	0.00000	/VEL
5	1	291	291	0	0.84039	0.00000	/VEL
5	1	292	292	0	0.84552	0.00000	/VEL
5	1	293	293	0	0.85074	0.00000	/VEL
5	1	294	294	0	0.85725	0.00000	/VEL
5	1	295	295	0	0.84613	0.00000	/VEL
5	1	296	296	0	0.85227	0.00000	/VEL
5	1	297	297	0	0.85878	0.00000	/VEL
5	1	298	298	0	0.86631	0.00000	/VEL
5	1	299	299	0	0.87342	0.00000	/VEL
5	1	300	300	0	0.88058	0.00000	/VEL
5	1	301	301	0	0.88759	0.00000	/VEL
5	1	302	302	0	0.89557	0.00000	/VEL
5	1	303	303	0	0.90320	0.00000	/VEL
5	1	304	304	0	0.90320	0.00000	/VEL
5	1	305	305	0	0.91157	0.00000	/VEL
5	1	306	306	0	0.91951	0.00000	/VEL
5	1	307	307	0	0.92750	0.00000	/VEL
5	1	308	308	0	0.93652	0.00000	/VEL
5	1	309	309	0	0.94559	0.00000	/VEL
5	1	310	310	0	0.95365	0.00000	/VEL
5	1	311	311	0	0.93087	0.00000	/VEL
5	1	312	312	0	0.93617	0.00000	/VEL
5	1	313	313	0	0.94296	0.00000	/VEL
5	1	314	314	0	0.94913	0.00000	/VEL
5	1	315	315	0	0.95510	0.00000	/VEL
5	1	316	316	0	0.96190	0.00000	/VEL
5	1	317	317	0	0.96768	0.00000	/VEL
5	1	318	318	0	0.97511	0.00000	/VEL
5	1	319	319	0	0.98053	0.00000	/VEL
5	1	320	320	0	0.98796	0.00000	/VEL
5	1	321	321	0	0.96241	0.00000	/VEL
5	1	322	322	0	0.96569	0.00000	/VEL
5	1	323	323	0	0.96995	0.00000	/VEL
5	1	324	324	0	0.97361	0.00000	/VEL
5	1	325	325	0	0.97812	0.00000	/VEL
5	1	326	326	0	0.98179	0.00000	/VEL
5	1	327	327	0	0.97130	0.00000	/VEL
5	1	328	328	0	0.97232	0.00000	/VEL
5	1	329	329	0	0.97433	0.00000	/VEL
5	1	330	330	0	0.97655	0.00000	/VEL
5	1	331	331	0	0.97776	0.00000	/VEL
5	1	332	332	0	0.97976	0.00000	/VEL
5	1	333	333	0	0.98139	0.00000	/VEL
5	1	334	334	0	0.98320	0.00000	/VEL
5	1	335	335	0	0.98502	0.00000	/VEL
5	1	336	336	0	0.98683	0.00000	/VEL
5	1	337	337	0	0.98864	0.00000	/VEL
5	1	338	338	0	0.98985	0.00000	/VEL
5	1	339	339	0	0.99227	0.00000	/VEL
5	1	340	340	0	0.99330	0.00000	/VEL
5	1	341	341	0	0.99590	0.00000	/VEL
5	1	342	342	0	0.99114	0.00000	/VEL
5	1	343	343	0	0.99247	0.00000	/VEL
5	1	344	344	0	0.99300	0.00000	/VEL
5	1	345	345	0	0.99361	0.00000	/VEL
5	1	346	346	0	0.99432	0.00000	/VEL
5	1	347	347	0	0.99547	0.00000	/VEL
5	1	348	348	0	0.99556	0.00000	/VEL
5	1	349	349	0	0.99732	0.00000	/VEL
5	1	350	350	0	0.99741	0.00000	/VEL
5	1	351	351	0	0.99856	0.00000	/VEL
5	1	352	352	0	0.99926	0.00000	/VEL
5	1	353	353	0	0.99988	0.00000	/VEL
5	1	354	354	0	1.0010	0.00000	/VEL
5	1	355	355	0	1.0017	0.00000	/VEL

5	1	356	356	0	1.0024	0.00000	/VEL
5	1	357	357	0	0.99592	0.00000	/VEL
5	1	358	358	0	0.98940	0.00000	/VEL
5	1	359	359	0	0.98347	0.00000	/VEL
5	1	360	360	0	0.97754	0.00000	/VEL
5	1	361	361	0	0.97161	0.00000	/VEL
5	1	362	362	0	0.96510	0.00000	/VEL
5	1	363	363	0	0.95919	0.00000	/VEL
5	1	364	364	0	0.95326	0.00000	/VEL
5	1	365	365	0	0.94736	0.00000	/VEL
5	1	366	366	0	0.94142	0.00000	/VEL
5	1	367	367	0	0.93549	0.00000	/VEL
5	1	368	368	0	0.92964	0.00000	/VEL
5	1	369	369	0	0.92370	0.00000	/VEL
5	1	370	370	0	0.91837	0.00000	/VEL
5	1	371	371	0	0.91248	0.00000	/VEL
5	1	372	372	0	0.89848	0.00000	/VEL
5	1	373	373	0	0.88577	0.00000	/VEL
5	1	374	374	0	0.87390	0.00000	/VEL
5	1	375	375	0	0.87390	0.00000	/VEL
5	1	376	376	0	0.86169	0.00000	/VEL
5	1	377	377	0	0.84982	0.00000	/VEL
5	1	378	378	0	0.83794	0.00000	/VEL
5	1	379	379	0	0.82675	0.00000	/VEL
5	1	380	380	0	0.81557	0.00000	/VEL
5	1	381	381	0	0.80420	0.00000	/VEL
5	1	382	382	0	0.79301	0.00000	/VEL
5	1	383	383	0	0.78215	0.00000	/VEL
5	1	384	384	0	0.76080	0.00000	/VEL
5	1	385	385	0	0.75026	0.00000	/VEL
5	1	386	386	0	0.74028	0.00000	/VEL
5	1	387	387	0	0.73343	0.00000	/VEL
5	1	388	388	0	0.72809	0.00000	/VEL
5	1	389	389	0	0.72328	0.00000	/VEL
5	1	390	390	0	0.71848	0.00000	/VEL
5	1	391	391	0	0.71368	0.00000	/VEL
5	1	392	392	0	0.70921	0.00000	/VEL
5	1	393	393	0	0.70431	0.00000	/VEL
5	1	394	394	0	0.70004	0.00000	/VEL
5	1	395	395	0	0.69514	0.00000	/VEL
5	1	396	396	0	0.69077	0.00000	/VEL
5	1	397	397	0	0.68587	0.00000	/VEL
5	1	398	398	0	0.68151	0.00000	/VEL
5	1	399	399	0	0.67714	0.00000	/VEL
5	1	400	400	0	0.67277	0.00000	/VEL
5	1	401	401	0	0.66788	0.00000	/VEL
5	1	402	402	0	0.67121	0.00000	/VEL
5	1	403	403	0	0.67007	0.00000	/VEL
5	1	404	404	0	0.66884	0.00000	/VEL
5	1	405	405	0	0.66761	0.00000	/VEL
5	1	406	406	0	0.66597	0.00000	/VEL
5	1	407	407	0	0.66524	0.00000	/VEL
5	1	408	408	0	0.66401	0.00000	/VEL
5	1	409	409	0	0.66278	0.00000	/VEL
5	1	410	410	0	0.66156	0.00000	/VEL
5	1	411	411	0	0.66042	0.00000	/VEL
5	1	412	412	0	0.65919	0.00000	/VEL
5	1	413	413	0	0.65755	0.00000	/VEL
5	1	414	414	0	0.65683	0.00000	/VEL
5	1	415	415	0	0.65560	0.00000	/VEL
5	1	416	416	0	0.65438	0.00000	/VEL
5	1	417	417	0	0.65968	0.00000	/VEL
5	1	418	418	0	0.66026	0.00000	/VEL
5	1	419	419	0	0.65968	0.00000	/VEL
5	1	420	420	0	0.66026	0.00000	/VEL
5	1	421	421	0	0.66026	0.00000	/VEL
5	1	422	422	0	0.66026	0.00000	/VEL
5	1	423	423	0	0.66066	0.00000	/VEL
5	1	424	424	0	0.66066	0.00000	/VEL
5	1	425	425	0	0.66085	0.00000	/VEL
5	1	426	426	0	0.66125	0.00000	/VEL
5	1	427	427	0	0.66125	0.00000	/VEL
5	1	428	428	0	0.66165	0.00000	/VEL
5	1	429	429	0	0.66165	0.00000	/VEL
5	1	430	430	0	0.66184	0.00000	/VEL

5	1	431	431	0	0.66224	0.00000	/VEL
5	1	432	432	0	0.66224	0.00000	/VEL
5	1	433	433	0	0.66639	0.00000	/VEL
5	1	434	434	0	0.67689	0.00000	/VEL
5	1	435	435	0	0.69820	0.00000	/VEL
5	1	436	436	0	0.70891	0.00000	/VEL
5	1	437	437	0	0.71976	0.00000	/VEL
5	1	438	438	0	0.73079	0.00000	/VEL
5	1	439	439	0	0.74262	0.00000	/VEL
5	1	440	440	0	0.75428	0.00000	/VEL
5	1	441	441	0	0.76567	0.00000	/VEL
5	1	442	442	0	0.77815	0.00000	/VEL
5	1	443	443	0	0.79017	0.00000	/VEL
5	1	444	444	0	0.80266	0.00000	/VEL
5	1	445	445	0	0.12985	0.00000	/VEL
5	1	446	446	0	0.13486	0.00000	/VEL
5	1	447	447	0	0.14103	0.00000	/VEL
5	1	448	448	0	0.14699	0.00000	/VEL
5	1	449	449	0	0.15381	0.00000	/VEL
5	1	450	450	0	0.16111	0.00000	/VEL
5	1	451	451	0	0.16885	0.00000	/VEL
5	1	452	452	0	0.17711	0.00000	/VEL
5	1	453	453	0	0.26581	0.00000	/VEL
5	1	454	454	0	0.26928	0.00000	/VEL
5	1	455	455	0	0.27279	0.00000	/VEL
5	1	456	456	0	0.27631	0.00000	/VEL
5	1	457	457	0	0.27979	0.00000	/VEL
5	1	458	458	0	0.28332	0.00000	/VEL
5	1	459	459	0	0.28755	0.00000	/VEL
5	1	460	460	0	0.29108	0.00000	/VEL
5	1	461	461	0	0.38505	0.00000	/VEL
5	1	462	462	0	0.38155	0.00000	/VEL
5	1	463	463	0	0.38023	0.00000	/VEL
5	1	464	464	0	0.16658	0.00000	/VEL
5	1	465	465	0	0.17034	0.00000	/VEL
5	1	466	466	0	0.17416	0.00000	/VEL
5	1	467	467	0	0.17795	0.00000	/VEL
5	1	468	468	0	0.18223	0.00000	/VEL
5	1	469	469	0	0.18603	0.00000	/VEL
5	1	470	470	0	0.19031	0.00000	/VEL
5	1	471	471	0	0.19412	0.00000	/VEL
5	1	472	472	0	0.19890	0.00000	/VEL
5	1	473	473	0	0.20322	0.00000	/VEL
5	1	474	474	0	0.20799	0.00000	/VEL
5	1	475	475	0	0.21278	0.00000	/VEL
5	1	476	476	0	0.22347	0.00000	/VEL
5	1	477	477	0	0.23366	0.00000	/VEL
5	1	478	478	0	0.25431	0.00000	/VEL
5	1	479	479	0	0.26535	0.00000	/VEL
5	1	480	480	0	0.27650	0.00000	/VEL
5	1	481	481	0	0.28837	0.00000	/VEL
5	1	482	482	0	0.30178	0.00000	/VEL
5	1	483	483	0	0.31452	0.00000	/VEL
5	1	484	484	0	0.32872	0.00000	/VEL
5	1	485	485	0	0.34302	0.00000	/VEL
5	1	486	486	0	0.35874	0.00000	/VEL
5	1	487	487	0	0.37382	0.00000	/VEL
5	1	488	488	0	0.37382	0.00000	/VEL
5	1	489	489	0	0.39188	0.00000	/VEL
5	1	490	490	0	0.40847	0.00000	/VEL
5	1	491	491	0	0.47593	0.00000	/VEL
5	1	492	492	0	0.47236	0.00000	/VEL
5	1	493	493	0	0.47053	0.00000	/VEL
5	1	494	494	0	0.46716	0.00000	/VEL
5	1	495	495	0	0.22276	0.00000	/VEL
5	1	496	496	0	0.23159	0.00000	/VEL
5	1	497	497	0	0.24181	0.00000	/VEL
5	1	498	498	0	0.25271	0.00000	/VEL
5	1	499	499	0	0.26364	0.00000	/VEL
5	1	500	500	0	0.27524	0.00000	/VEL
6	1	1	63	-1	100.00	1000.0	18 /DISP
6	1	64	384	-2	100.00	1000.0	19 /DISP
6	1	385	425	-1	100.00	1000.0	18 /DISP
6	1	426	501	-3	100.00	500.00	12 /DISP
7	1	1	63	0	0.47300E-01	0.00000	18 /DM

7	1	64	384	0	0.47300E-01	0.00000	19	/DM
7	1	385	425	0	0.47300E-01	0.00000	18	/DM
7	1	426	501	0	0.47300E-01	0.00000	12	/DM
8	1	1	63	0	0.10000	0.00000	18	/TAU
8	1	64	384	0	0.10000	0.00000	19	/TAU
8	1	385	425	0	0.10000	0.00000	18	/TAU
8	1	426	501	0	0.10000	0.00000	12	/TAU
9	1	1	63	-4	-2.8700	0.21000	18	/THETA1
9	1	64	384	-5	-2.8700	0.21000	19	/THETA1
9	1	385	425	-4	-2.8700	0.21000	18	/THETA1
9	1	426	501	-6	0.70000E-01	0.47000	12	/THETA1
10	1	1	63	-7	0.12000	0.45000	18	/THETA2
10	1	64	384	-8	0.80000E-01	0.50000	19	/THETA2
10	1	385	425	-7	0.12000	0.45000	18	/THETA2
10	1	426	501	0	0.00000	0.00000	12	/THETA2
11	1	1	63	-9	0.30000	1.3000	18	/BWIDTH
11	1	64	384	-10	0.70000	2.5000	19	/BWIDTH
11	1	385	425	-9	0.30000	1.3000	18	/BWIDTH
11	1	426	501	0	1.0000	0.00000	12	/BWIDTH
12	1	1	63	0	2.0000	0.00000	18	/FWIDTH
12	1	64	384	0	2.0000	0.00000	19	/FWIDTH
12	1	385	425	0	2.0000	0.00000	18	/FWIDTH
12	1	426	501	0	2.0000	0.00000	12	/FWIDTH
13	1	1	63	0	0.00000	0.00000	18	/RHOWB
13	1	64	384	0	0.00000	0.00000	19	/RHOWB
13	1	385	425	0	0.00000	0.00000	18	/RHOWB
13	1	426	501	0	0.00000	0.00000	12	/RHOWB
14	1	1	63	0	0.00000	0.00000	18	/RKD1
14	1	64	384	0	0.00000	0.00000	19	/RKD1
14	1	385	425	0	0.00000	0.00000	18	/RKD1
14	1	426	501	0	0.00000	0.00000	12	/RKD1
15	1	1	63	0	0.00000	0.00000	18	/RKD2
15	1	64	384	0	0.00000	0.00000	19	/RKD2
15	1	385	425	0	0.00000	0.00000	18	/RKD2
15	1	426	501	0	0.00000	0.00000	12	/RKD2
16	1	1	63	-11	-2.8420	0.24000	18	/DIFF
16	1	64	384	-12	-2.8420	0.24000	19	/DIFF
16	1	385	425	-11	-2.8420	0.24000	18	/DIFF
16	1	426	501	0	0.00000	0.00000	12	/DIFF
17	1	1	63	0	0.56400E-01	0.00000	18	/LAMBDA1W
17	1	64	384	0	0.56400E-01	0.00000	19	/LAMBDA1W
17	1	385	425	0	0.56400E-01	0.00000	18	/LAMBDA1W
17	1	426	501	0	0.56400E-01	0.00000	12	/LAMBDA1W
18	1	1	1	-13	-3.0700	0.25000		/KH_01_1
19	1	1	1	-14	-1.1700	0.25000		/KH_06_1
20	1	1	1	-14	-0.77000	0.25000		/KH_06_2
21	1	1	1	-14	-1.0700	0.25000		/KH_06_3
22	1	1	1	-15	1.9700	0.25000		/KH_07_1
23	1	1	1	-15	1.9700	0.25000		/KH_07_2
24	1	1	1	-15	-2.0700	0.25000		/KH_07_3
25	1	1	1	-15	0.83000	0.25000		/KH_07_4
26	1	1	1	-15	-0.17000	0.25000		/KH_07_5
27	1	1	1	-15	-0.47000	0.25000		/KH_07_6
28	1	1	1	-15	0.23000	0.25000		/KH_07_7
29	1	1	1	-15	-0.17000	0.25000		/KH_07_8
30	1	1	1	-15	0.63000	0.25000		/KH_07_9
31	1	1	1	-16	-2.7700	0.25000		/KH_08_1
32	1	1	1	-17	-1.3700	0.25000		/KH_09_1
33	1	1	1	-18	-0.37000	0.25000		/KH_10_1
34	1	1	1	-18	-0.77000	0.25000		/KH_10_2
35	1	1	1	-19	0.10000	0.25000		/KH_11_1
36	1	1	1	-19	-0.77000	0.25000		/KH_11_2
37	1	1	1	-19	-0.47000	0.25000		/KH_11_3
38	1	1	1	-20	-0.99000	0.25000		/KH_12_1
39	1	1	1	-20	-0.99000	0.25000		/KH_12_2
40	1	1	1	-21	0.23000	0.25000		/KH_13_1
41	1	1	1	-21	-0.70000E-01	0.25000		/KH_13_2
42	1	1	1	-21	0.83000	0.25000		/KH_13_3
43	1	1	1	-22	1.4700	0.25000		/KH_14_1
44	1	1	1	-22	0.23000	0.25000		/KH_14_2
45	1	1	1	-22	0.23000	0.25000		/KH_14_3
46	1	1	1	-22	0.23000	0.25000		/KH_14_4
47	1	1	1	-22	-0.77000	0.25000		/KH_14_5
48	1	1	1	-23	-0.59000	0.25000		/KH_15_1
49	1	1	1	-23	-0.59000	0.25000		/KH_15_2

50	1	1	1	-23	-0.59000	0.25000	/KH_15_3
51	1	1	1	-23	-0.59000	0.25000	/KH_15_4
52	1	1	1	-23	-0.77000	0.25000	/KH_15_5
53	1	1	1	-24	1.4700	0.25000	/KH_16_1
54	1	1	1	-24	0.23000	0.25000	/KH_16_2
55	1	1	1	-24	0.23000	0.25000	/KH_16_3
56	1	1	1	-24	-0.77000	0.25000	/KH_16_4
57	1	1	1	-25	-0.59000	0.25000	/KH_17_1
58	1	1	1	-25	-1.5900	0.25000	/KH_17_2
59	1	1	1	-25	-1.7700	0.25000	/KH_17_3
60	1	1	1	-26	1.4700	0.25000	/KH_18_1
61	1	1	1	-26	1.1300	0.25000	/KH_18_2
62	1	1	1	-26	1.2300	0.25000	/KH_18_3
63	1	1	1	-26	0.53000	0.25000	/KH_18_4
64	1	1	1	-26	-0.77000	0.25000	/KH_18_5
65	1	1	1	-27	1.4700	0.25000	/KH_19_1
66	1	1	1	-27	1.0000	0.25000	/KH_19_2
67	1	1	1	-27	1.1300	0.25000	/KH_19_3
68	1	1	1	-27	1.2300	0.25000	/KH_19_4
69	1	1	1	-27	-0.77000	0.25000	/KH_19_5
70	1	1	1	-28	0.97000	0.25000	/KH_20_1
71	1	1	1	-28	0.57000	0.25000	/KH_20_2
72	1	1	1	-28	-2.0700	0.25000	/KH_20_3
73	1	1	1	-28	0.97000	0.25000	/KH_20_4
74	1	1	1	-28	0.97000	0.25000	/KH_20_5
75	1	1	1	-28	0.97000	0.25000	/KH_20_6
76	1	1	1	-29	0.00000	0.30000E-02	/LD_01_1
77	1	1	1	-30	0.00000	0.24000E-02	/LD_06_1
78	1	1	1	-30	0.00000	0.24000E-02	/LD_06_2
79	1	1	1	-30	0.00000	0.24000E-02	/LD_06_3
80	1	1	1	-31	0.00000	0.20000E-02	/LD_07_1
81	1	1	1	-31	0.00000	0.20000E-02	/LD_07_2
82	1	1	1	-31	0.00000	0.20000E-02	/LD_07_3
83	1	1	1	-31	0.00000	0.20000E-02	/LD_07_4
84	1	1	1	-31	0.00000	0.20000E-02	/LD_07_5
85	1	1	1	-31	0.00000	0.20000E-02	/LD_07_6
86	1	1	1	-31	0.00000	0.52000E-02	/LD_07_7
87	1	1	1	-31	0.00000	0.20000E-02	/LD_07_8
88	1	1	1	-31	0.00000	0.20000E-02	/LD_07_9
89	1	1	1	-32	0.00000	0.30000E-02	/LD_08_1
90	1	1	1	-33	0.00000	0.20000E-02	/LD_09_1
91	1	1	1	-34	0.00000	0.80000E-02	/LD_10_1
92	1	1	1	-34	0.00000	0.80000E-02	/LD_10_2
93	1	1	1	-35	0.70000E-03	0.45000E-02	/LD_11_1
94	1	1	1	-35	0.70000E-03	0.45000E-02	/LD_11_2
95	1	1	1	-35	0.70000E-03	0.45000E-02	/LD_11_3
96	1	1	1	-36	0.70000E-03	0.45000E-02	/LD_12_1
97	1	1	1	-36	0.70000E-03	0.45000E-02	/LD_12_2
98	1	1	1	-37	0.70000E-03	0.45000E-02	/LD_13_1
99	1	1	1	-37	0.70000E-03	0.45000E-02	/LD_13_2
100	1	1	1	-37	0.70000E-03	0.45000E-02	/LD_13_3
101	1	1	1	-38	0.70000E-03	0.45000E-02	/LD_14_1
102	1	1	1	-38	0.70000E-03	0.45000E-02	/LD_14_2
103	1	1	1	-38	0.70000E-03	0.45000E-02	/LD_14_3
104	1	1	1	-38	0.70000E-03	0.45000E-02	/LD_14_4
105	1	1	1	-38	0.70000E-03	0.45000E-02	/LD_14_5
106	1	1	1	-39	0.70000E-03	0.45000E-02	/LD_15_1
107	1	1	1	-39	0.70000E-03	0.45000E-02	/LD_15_2
108	1	1	1	-39	0.70000E-03	0.45000E-02	/LD_15_3
109	1	1	1	-39	0.70000E-03	0.45000E-02	/LD_15_4
110	1	1	1	-39	0.70000E-03	0.45000E-02	/LD_15_5
111	1	1	1	-40	0.70000E-03	0.45000E-02	/LD_16_1
112	1	1	1	-40	0.70000E-03	0.45000E-02	/LD_16_2
113	1	1	1	-40	0.70000E-03	0.45000E-02	/LD_16_3
114	1	1	1	-40	0.70000E-03	0.45000E-02	/LD_16_4
115	1	1	1	-41	0.70000E-03	0.45000E-02	/LD_17_1
116	1	1	1	-41	0.70000E-03	0.45000E-02	/LD_17_2
117	1	1	1	-41	0.70000E-03	0.45000E-02	/LD_17_3
118	1	1	1	-42	0.70000E-03	0.45000E-02	/LD_18_1
119	1	1	1	-42	0.70000E-03	0.45000E-02	/LD_18_2
120	1	1	1	-42	0.70000E-03	0.45000E-02	/LD_18_3
121	1	1	1	-42	0.70000E-03	0.45000E-02	/LD_18_4
122	1	1	1	-42	0.70000E-03	0.45000E-02	/LD_18_5
123	1	1	1	-43	0.70000E-03	0.45000E-02	/LD_19_1
124	1	1	1	-43	0.70000E-03	0.45000E-02	/LD_19_2

125	1		1	-43	0.70000E-03	0.45000E-02	/LD_19_3	
126	1		1	-43	0.70000E-03	0.45000E-02	/LD_19_4	
127	1		1	-43	0.70000E-03	0.45000E-02	/LD_19_5	
128	1		1	-44	0.00000	0.74000E-02	/LD_20_1	
129	1		1	-44	0.00000	0.74000E-02	/LD_20_2	
130	1		1	-44	0.00000	0.74000E-02	/LD_20_3	
131	1		1	-44	0.00000	0.74000E-02	/LD_20_4	
132	1		1	-44	0.00000	0.74000E-02	/LD_20_5	
133	1		1	-44	0.00000	0.74000E-02	/LD_20_6	
134	1		1	-45	0.00000	0.25000	/RC_01_1	
	1	1	1	0	24	0	1	24

Transport of Tritium, with LHS, Event Pepato

200 2243

381 97.938 11600 0.01724 0. -1.0

2

17

2 11 21 31 41 51 76 101 126 151

176 201 251 301 326 351 380

524 1 117 0 41 1

1 1 1 381 0 0.00000 0.00000

2 1 1 381 0 0.0564 0.00000

3 1 1 4 -39 7.65050 0.44000

3 1 5 10 0 0.00000 0.00000

3 1 11 15 -40 8.40300 0.39000

3 1 16 79 0 0.00000 0.00000

3 1 80 84 -41 8.40300 0.39000

3 1 85 381 0 0.00000 0.00000

4 1 1 4 -39 7.65000 0.44000

4 1 5 10 0 0.00000 0.00000

4 1 11 15 -40 8.40300 0.39000

4 1 16 79 0 0.00000 0.00000

4 1 80 84 -41 8.40300 0.39000

4 1 85 381 0 0.00000 0.00000

5 1 1 1 0 2.0862 0.00000

5 1 2 2 0 2.0979 0.00000

5 1 3 3 0 2.1214 0.00000

5 1 4 4 0 2.1330 0.00000

5 1 5 5 0 2.1507 0.00000

5 1 6 6 0 2.1686 0.00000

5 1 7 7 0 2.2221 0.00000

5 1 8 8 0 2.2459 0.00000

5 1 9 9 0 2.2759 0.00000

5 1 10 10 0 2.2995 0.00000

5 1 11 11 0 2.3295 0.00000

5 1 12 12 0 2.3592 0.00000

5 1 13 13 0 2.3890 0.00000

5 1 14 14 0 2.4189 0.00000

5 1 15 15 0 2.4487 0.00000

5 1 16 16 0 2.4785 0.00000

5 1 17 17 0 2.5144 0.00000

5 1 18 18 0 2.5144 0.00000

5 1 19 19 0 2.5382 0.00000

5 1 20 20 0 2.5741 0.00000

5 1 21 21 0 2.6039 0.00000

5 1 22 22 0 2.6416 0.00000

5 1 23 23 0 2.6939 0.00000

5 1 24 24 0 2.7309 0.00000

5 1 25 25 0 2.7832 0.00000

5 1 26 26 0 2.8276 0.00000

5 1 27 27 0 2.8723 0.00000

5 1 28 28 0 2.9243 0.00000

5 1 29 29 0 2.9692 0.00000

5 1 30 30 0 3.0210 0.00000

5 1 31 31 0 3.0733 0.00000

5 1 32 32 0 3.1251 0.00000

5 1 33 33 0 3.1774 0.00000

5 1 34 34 0 3.2371 0.00000

5 1 35 35 0 3.2815 0.00000

5 1 36 36 0 3.3412 0.00000

5 1 37 37 0 3.3806 0.00000

5 1 38 38 0 3.3725 0.00000

5 1 39 39 0 3.3644 0.00000

5 1 40 40 0 3.3563 0.00000

5 1 41 41 0 3.3563 0.00000

5 1 42 42 0 3.3405 0.00000

5 1 43 43 0 3.3320 0.00000

5 1 44 44 0 3.3320 0.00000

5 1 45 45 0 3.3320 0.00000

5 1 46 46 0 3.3158 0.00000

5 1 47 47 0 3.3158 0.00000

5 1 48 48 0 3.3077 0.00000

5 1 49 49 0 3.2920 0.00000

5 1 50 50 0 3.2916 0.00000

5 1 51 51 0 3.2835 0.00000

5 1 52 52 0 3.2754 0.00000

5 1 53 53 0 3.7067 0.00000

/MC, ISEED

/NX, DX, NT, DT, SCL_PRM

/INBOUN

/NUMOBS

/NPARD, NSPEC, NFSP

19 /C_SRCE

19 /GAMMA

19 /CINIT

19 /CINIT

19 /CINIT

19 /CINIT

19 /CINIT

19 /CINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

19 /SINIT

5	1	54	54	0	3.6612	0.00000	/VEL
5	1	55	55	0	3.6256	0.00000	/VEL
5	1	56	56	0	3.5984	0.00000	/VEL
5	1	57	57	0	3.5537	0.00000	/VEL
5	1	58	58	0	3.5175	0.00000	/VEL
5	1	59	59	0	3.4820	0.00000	/VEL
5	1	60	60	0	1.3843	0.00000	/VEL
5	1	61	61	0	1.3744	0.00000	/VEL
5	1	62	62	0	1.3746	0.00000	/VEL
5	1	63	63	0	1.3649	0.00000	/VEL
5	1	64	64	0	1.3619	0.00000	/VEL
5	1	65	65	0	1.3552	0.00000	/VEL
5	1	66	66	0	1.3488	0.00000	/VEL
5	1	67	67	0	1.3426	0.00000	/VEL
5	1	68	68	0	1.3704	0.00000	/VEL
5	1	69	69	0	1.3663	0.00000	/VEL
5	1	70	70	0	1.3584	0.00000	/VEL
5	1	71	71	0	1.3541	0.00000	/VEL
5	1	72	72	0	1.3541	0.00000	/VEL
5	1	73	73	0	1.3497	0.00000	/VEL
5	1	74	74	0	1.3422	0.00000	/VEL
5	1	75	75	0	1.3377	0.00000	/VEL
5	1	76	76	0	1.3337	0.00000	/VEL
5	1	77	77	0	1.3263	0.00000	/VEL
5	1	78	78	0	1.3222	0.00000	/VEL
5	1	79	79	0	1.3180	0.00000	/VEL
5	1	80	80	0	1.3139	0.00000	/VEL
5	1	81	81	0	1.3069	0.00000	/VEL
5	1	82	82	0	1.3026	0.00000	/VEL
5	1	83	83	0	1.2989	0.00000	/VEL
5	1	84	84	0	1.3827	0.00000	/VEL
5	1	85	85	0	1.3804	0.00000	/VEL
5	1	86	86	0	1.3759	0.00000	/VEL
5	1	87	87	0	1.3710	0.00000	/VEL
5	1	88	88	0	1.3710	0.00000	/VEL
5	1	89	89	0	1.3722	0.00000	/VEL
5	1	90	90	0	1.3675	0.00000	/VEL
5	1	91	91	0	1.3638	0.00000	/VEL
5	1	92	92	0	1.3552	0.00000	/VEL
5	1	93	93	0	1.3536	0.00000	/VEL
5	1	94	94	0	1.3535	0.00000	/VEL
5	1	95	95	0	1.3521	0.00000	/VEL
5	1	96	96	0	1.3525	0.00000	/VEL
5	1	97	97	0	1.3525	0.00000	/VEL
5	1	98	98	0	1.3513	0.00000	/VEL
5	1	99	99	0	1.3497	0.00000	/VEL
5	1	100	100	0	1.3513	0.00000	/VEL
5	1	101	101	0	1.3521	0.00000	/VEL
5	1	102	102	0	1.3521	0.00000	/VEL
5	1	103	103	0	1.4457	0.00000	/VEL
5	1	104	104	0	1.4478	0.00000	/VEL
5	1	105	105	0	1.4514	0.00000	/VEL
5	1	106	106	0	1.4514	0.00000	/VEL
5	1	107	107	0	1.4536	0.00000	/VEL
5	1	108	108	0	1.4560	0.00000	/VEL
5	1	109	109	0	1.4602	0.00000	/VEL
5	1	110	110	0	1.4602	0.00000	/VEL
5	1	111	111	0	1.4626	0.00000	/VEL
5	1	112	112	0	1.4651	0.00000	/VEL
5	1	113	113	0	1.4676	0.00000	/VEL
5	1	114	114	0	1.4676	0.00000	/VEL
5	1	115	115	0	1.4725	0.00000	/VEL
5	1	116	116	0	1.5511	0.00000	/VEL
5	1	117	117	0	1.5579	0.00000	/VEL
5	1	118	118	0	1.5579	0.00000	/VEL
5	1	119	119	0	1.5576	0.00000	/VEL
5	1	120	120	0	1.5652	0.00000	/VEL
5	1	121	121	0	1.5680	0.00000	/VEL
5	1	122	122	0	1.5737	0.00000	/VEL
5	1	123	123	0	1.6988	0.00000	/VEL
5	1	124	124	0	1.6988	0.00000	/VEL
5	1	125	125	0	1.7067	0.00000	/VEL
5	1	126	126	0	1.7140	0.00000	/VEL
5	1	127	127	0	1.7244	0.00000	/VEL
5	1	128	128	0	1.7328	0.00000	/VEL

5	1	129	129	0	1.7415	0.00000	/VEL
5	1	130	130	0	1.7512	0.00000	/VEL
5	1	131	131	0	1.7512	0.00000	/VEL
5	1	132	132	0	1.7630	0.00000	/VEL
5	1	133	133	0	1.7739	0.00000	/VEL
5	1	134	134	0	1.7842	0.00000	/VEL
5	1	135	135	0	1.7965	0.00000	/VEL
5	1	136	136	0	1.7965	0.00000	/VEL
5	1	137	137	0	1.7667	0.00000	/VEL
5	1	138	138	0	1.7759	0.00000	/VEL
5	1	139	139	0	1.7880	0.00000	/VEL
5	1	140	140	0	1.7984	0.00000	/VEL
5	1	141	141	0	1.8107	0.00000	/VEL
5	1	142	142	0	1.8212	0.00000	/VEL
5	1	143	143	0	1.8329	0.00000	/VEL
5	1	144	144	0	1.8464	0.00000	/VEL
5	1	145	145	0	1.8599	0.00000	/VEL
5	1	146	146	0	1.8701	0.00000	/VEL
5	1	147	147	0	1.8701	0.00000	/VEL
5	1	148	148	0	1.8838	0.00000	/VEL
5	1	149	149	0	1.8986	0.00000	/VEL
5	1	150	150	0	1.9108	0.00000	/VEL
5	1	151	151	0	1.8801	0.00000	/VEL
5	1	152	152	0	1.8920	0.00000	/VEL
5	1	153	153	0	1.9022	0.00000	/VEL
5	1	154	154	0	1.9299	0.00000	/VEL
5	1	155	155	0	1.9376	0.00000	/VEL
5	1	156	156	0	1.9408	0.00000	/VEL
5	1	157	157	0	1.9408	0.00000	/VEL
5	1	158	158	0	1.9468	0.00000	/VEL
5	1	159	159	0	1.9536	0.00000	/VEL
5	1	160	160	0	1.9595	0.00000	/VEL
5	1	161	161	0	1.9629	0.00000	/VEL
5	1	162	162	0	1.9706	0.00000	/VEL
5	1	163	163	0	1.9756	0.00000	/VEL
5	1	164	164	0	1.9816	0.00000	/VEL
5	1	165	165	0	1.9816	0.00000	/VEL
5	1	166	166	0	1.9893	0.00000	/VEL
5	1	167	167	0	1.9926	0.00000	/VEL
5	1	168	168	0	1.9986	0.00000	/VEL
5	1	169	169	0	2.0063	0.00000	/VEL
5	1	170	170	0	2.0113	0.00000	/VEL
5	1	171	171	0	1.9770	0.00000	/VEL
5	1	172	172	0	1.9770	0.00000	/VEL
5	1	173	173	0	1.9792	0.00000	/VEL
5	1	174	174	0	1.9824	0.00000	/VEL
5	1	175	175	0	1.9829	0.00000	/VEL
5	1	176	176	0	1.9856	0.00000	/VEL
5	1	177	177	0	1.9872	0.00000	/VEL
5	1	178	178	0	1.9894	0.00000	/VEL
5	1	179	179	0	1.9921	0.00000	/VEL
5	1	180	180	0	1.9932	0.00000	/VEL
5	1	181	181	0	1.9970	0.00000	/VEL
5	1	182	182	0	1.9970	0.00000	/VEL
5	1	183	183	0	1.9970	0.00000	/VEL
5	1	184	184	0	1.9997	0.00000	/VEL
5	1	185	185	0	1.9986	0.00000	/VEL
5	1	186	186	0	2.0004	0.00000	/VEL
5	1	187	187	0	2.0055	0.00000	/VEL
5	1	188	188	0	2.0182	0.00000	/VEL
5	1	189	189	0	2.0182	0.00000	/VEL
5	1	190	190	0	2.0177	0.00000	/VEL
5	1	191	191	0	2.0173	0.00000	/VEL
5	1	192	192	0	2.0179	0.00000	/VEL
5	1	193	193	0	2.0175	0.00000	/VEL
5	1	194	194	0	2.0171	0.00000	/VEL
5	1	195	195	0	2.0160	0.00000	/VEL
5	1	196	196	0	2.0160	0.00000	/VEL
5	1	197	197	0	2.0163	0.00000	/VEL
5	1	198	198	0	2.0170	0.00000	/VEL
5	1	199	199	0	2.0159	0.00000	/VEL
5	1	200	200	0	2.0173	0.00000	/VEL
5	1	201	201	0	2.0153	0.00000	/VEL
5	1	202	202	0	2.0167	0.00000	/VEL
5	1	203	203	0	2.0156	0.00000	/VEL

5	1	204	204	0	2.0156	0.00000	/VEL
5	1	205	205	0	2.0171	0.00000	/VEL
5	1	206	206	0	2.0447	0.00000	/VEL
5	1	207	207	0	2.0444	0.00000	/VEL
5	1	208	208	0	2.0442	0.00000	/VEL
5	1	209	209	0	2.0410	0.00000	/VEL
5	1	210	210	0	2.0399	0.00000	/VEL
5	1	211	211	0	2.0367	0.00000	/VEL
5	1	212	212	0	2.0367	0.00000	/VEL
5	1	213	213	0	2.0376	0.00000	/VEL
5	1	214	214	0	2.0366	0.00000	/VEL
5	1	215	215	0	2.0324	0.00000	/VEL
5	1	216	216	0	2.0324	0.00000	/VEL
5	1	217	217	0	1.7789	0.00000	/VEL
5	1	218	218	0	1.7769	0.00000	/VEL
5	1	219	219	0	1.7794	0.00000	/VEL
5	1	220	220	0	1.7794	0.00000	/VEL
5	1	221	221	0	1.7803	0.00000	/VEL
5	1	222	222	0	1.7791	0.00000	/VEL
5	1	223	223	0	1.7817	0.00000	/VEL
5	1	224	224	0	1.9544	0.00000	/VEL
5	1	225	225	0	1.9557	0.00000	/VEL
5	1	226	226	0	1.9557	0.00000	/VEL
5	1	227	227	0	1.9587	0.00000	/VEL
5	1	228	228	0	1.9624	0.00000	/VEL
5	1	229	229	0	1.9660	0.00000	/VEL
5	1	230	230	0	1.9660	0.00000	/VEL
5	1	231	231	0	1.9666	0.00000	/VEL
5	1	232	232	0	1.9703	0.00000	/VEL
5	1	233	233	0	1.9740	0.00000	/VEL
5	1	234	234	0	1.9740	0.00000	/VEL
5	1	235	235	0	1.9778	0.00000	/VEL
5	1	236	236	0	1.9807	0.00000	/VEL
5	1	237	237	0	1.9844	0.00000	/VEL
5	1	238	238	0	1.9844	0.00000	/VEL
5	1	239	239	0	1.9858	0.00000	/VEL
5	1	240	240	0	1.7978	0.00000	/VEL
5	1	241	241	0	1.8047	0.00000	/VEL
5	1	242	242	0	1.8027	0.00000	/VEL
5	1	243	243	0	1.8027	0.00000	/VEL
5	1	244	244	0	1.8089	0.00000	/VEL
5	1	245	245	0	1.8096	0.00000	/VEL
5	1	246	246	0	2.8375	0.00000	/VEL
5	1	247	247	0	2.8392	0.00000	/VEL
5	1	248	248	0	2.8525	0.00000	/VEL
5	1	249	249	0	2.8549	0.00000	/VEL
5	1	250	250	0	2.8624	0.00000	/VEL
5	1	251	251	0	2.8699	0.00000	/VEL
5	1	252	252	0	2.8774	0.00000	/VEL
5	1	253	253	0	2.8857	0.00000	/VEL
5	1	254	254	0	2.8857	0.00000	/VEL
5	1	255	255	0	2.8865	0.00000	/VEL
5	1	256	256	0	2.8948	0.00000	/VEL
5	1	257	257	0	2.9089	0.00000	/VEL
5	1	258	258	0	2.8787	0.00000	/VEL
5	1	259	259	0	2.8762	0.00000	/VEL
5	1	260	260	0	2.8619	0.00000	/VEL
5	1	261	261	0	2.8587	0.00000	/VEL
5	1	262	262	0	2.8453	0.00000	/VEL
5	1	263	263	0	2.8453	0.00000	/VEL
5	1	264	264	0	2.8429	0.00000	/VEL
5	1	265	265	0	2.8343	0.00000	/VEL
5	1	266	266	0	2.8211	0.00000	/VEL
5	1	267	267	0	2.8181	0.00000	/VEL
5	1	268	268	0	2.8104	0.00000	/VEL
5	1	269	269	0	2.8028	0.00000	/VEL
5	1	270	270	0	2.7944	0.00000	/VEL
5	1	271	271	0	2.7869	0.00000	/VEL
5	1	272	272	0	2.7869	0.00000	/VEL
5	1	273	273	0	2.7794	0.00000	/VEL
5	1	274	274	0	2.7774	0.00000	/VEL
5	1	275	275	0	2.6827	0.00000	/VEL
5	1	276	276	0	2.6746	0.00000	/VEL
5	1	277	277	0	2.6620	0.00000	/VEL
5	1	278	278	0	2.6584	0.00000	/VEL

5	1	279	279	0	2.6401	0.00000	/VEL
5	1	280	280	0	2.6401	0.00000	/VEL
5	1	281	281	0	2.6372	0.00000	/VEL
5	1	282	282	0	2.6248	0.00000	/VEL
5	1	283	283	0	2.6162	0.00000	/VEL
5	1	284	284	0	2.6032	0.00000	/VEL
5	1	285	285	0	2.5961	0.00000	/VEL
5	1	286	286	0	2.5876	0.00000	/VEL
5	1	287	287	0	2.5755	0.00000	/VEL
5	1	288	288	0	2.5755	0.00000	/VEL
5	1	289	289	0	2.5678	0.00000	/VEL
5	1	290	290	0	2.5602	0.00000	/VEL
5	1	291	291	0	2.5475	0.00000	/VEL
5	1	292	292	0	2.4785	0.00000	/VEL
5	1	293	293	0	2.4803	0.00000	/VEL
5	1	294	294	0	4.4639	0.00000	/VEL
5	1	295	295	0	4.5432	0.00000	/VEL
5	1	296	296	0	4.6003	0.00000	/VEL
5	1	297	297	0	4.6469	0.00000	/VEL
5	1	298	298	0	4.6916	0.00000	/VEL
5	1	299	299	0	4.7499	0.00000	/VEL
5	1	300	300	0	4.7840	0.00000	/VEL
5	1	301	301	0	4.8533	0.00000	/VEL
5	1	302	302	0	4.9000	0.00000	/VEL
5	1	303	303	0	4.9452	0.00000	/VEL
5	1	304	304	0	5.0038	0.00000	/VEL
5	1	305	305	0	5.0507	0.00000	/VEL
5	1	306	306	0	5.0507	0.00000	/VEL
5	1	307	307	0	5.1086	0.00000	/VEL
5	1	308	308	0	5.1656	0.00000	/VEL
5	1	309	309	0	5.1924	0.00000	/VEL
5	1	310	310	0	5.2436	0.00000	/VEL
5	1	311	311	0	5.2826	0.00000	/VEL
5	1	312	312	0	5.3347	0.00000	/VEL
5	1	313	313	0	5.3729	0.00000	/VEL
5	1	314	314	0	5.4120	0.00000	/VEL
5	1	315	315	0	5.4511	0.00000	/VEL
5	1	316	316	0	5.5033	0.00000	/VEL
5	1	317	317	0	5.5417	0.00000	/VEL
5	1	318	318	0	5.5947	0.00000	/VEL
5	1	319	319	0	5.6331	0.00000	/VEL
5	1	320	320	0	5.6854	0.00000	/VEL
5	1	321	321	0	5.7247	0.00000	/VEL
5	1	322	322	0	5.7770	0.00000	/VEL
5	1	323	323	0	5.8503	0.00000	/VEL
5	1	324	324	0	5.8493	0.00000	/VEL
5	1	325	325	0	5.8611	0.00000	/VEL
5	1	326	326	0	5.8453	0.00000	/VEL
5	1	327	327	0	5.8581	0.00000	/VEL
5	1	328	328	0	5.8710	0.00000	/VEL
5	1	329	329	0	5.8551	0.00000	/VEL
5	1	330	330	0	5.8670	0.00000	/VEL
5	1	331	331	0	5.8660	0.00000	/VEL
5	1	332	332	0	5.8651	0.00000	/VEL
5	1	333	333	0	5.8651	0.00000	/VEL
5	1	334	334	0	5.8770	0.00000	/VEL
5	1	335	335	0	5.8750	0.00000	/VEL
5	1	336	336	0	5.8750	0.00000	/VEL
5	1	337	337	0	5.8722	0.00000	/VEL
5	1	338	338	0	5.8851	0.00000	/VEL
5	1	339	339	0	5.8922	0.00000	/VEL
5	1	340	340	0	5.7912	0.00000	/VEL
5	1	341	341	0	5.7024	0.00000	/VEL
5	1	342	342	0	5.6015	0.00000	/VEL
5	1	343	343	0	5.5138	0.00000	/VEL
5	1	344	344	0	5.4373	0.00000	/VEL
5	1	345	345	0	5.3377	0.00000	/VEL
5	1	346	346	0	5.3377	0.00000	/VEL
5	1	347	347	0	5.2623	0.00000	/VEL
5	1	348	348	0	5.1618	0.00000	/VEL
5	1	349	349	0	5.0974	0.00000	/VEL
5	1	350	350	0	5.0113	0.00000	/VEL
5	1	351	351	0	4.9230	0.00000	/VEL
5	1	352	352	0	4.8468	0.00000	/VEL
5	1	353	353	0	4.7729	0.00000	/VEL

5	1	354	354	0	4.6969	0.00000	/VEL
5	1	355	355	0	4.7140	0.00000	/VEL
5	1	356	356	0	4.7245	0.00000	/VEL
5	1	357	357	0	4.7361	0.00000	/VEL
5	1	358	358	0	4.7466	0.00000	/VEL
5	1	359	359	0	4.7488	0.00000	/VEL
5	1	360	360	0	4.7699	0.00000	/VEL
5	1	361	361	0	4.7710	0.00000	/VEL
5	1	362	362	0	4.7826	0.00000	/VEL
5	1	363	363	0	4.8036	0.00000	/VEL
5	1	364	364	0	4.8058	0.00000	/VEL
5	1	365	365	0	4.8164	0.00000	/VEL
5	1	366	366	0	4.8280	0.00000	/VEL
5	1	367	367	0	4.8385	0.00000	/VEL
5	1	368	368	0	4.8512	0.00000	/VEL
5	1	369	369	0	4.8512	0.00000	/VEL
5	1	370	370	0	4.8618	0.00000	/VEL
5	1	371	371	0	5.0100	0.00000	/VEL
5	1	372	372	0	5.0310	0.00000	/VEL
5	1	373	373	0	5.0219	0.00000	/VEL
5	1	374	374	0	5.0455	0.00000	/VEL
5	1	375	375	0	5.0455	0.00000	/VEL
5	1	376	376	0	5.0573	0.00000	/VEL
5	1	377	377	0	5.0678	0.00000	/VEL
5	1	378	378	0	5.0705	0.00000	/VEL
5	1	379	379	0	5.0809	0.00000	/VEL
5	1	380	380	0	5.0941	0.00000	/VEL
5	1	381	381	0	5.0941	0.00000	/VEL
6	1	1	381	-1	100.00	1000.0	19 /DISP
7	1	1	381	0	0.47300E-01	0.00000	19 /DM
8	1	1	381	0	0.10000	0.00000	19 /TAU
9	1	1	381	-2	-2.8700	0.21000	19 /THETA1
10	1	1	381	-3	0.80000E-01	0.50000	19 /THETA2
11	1	1	381	-4	0.70000	2.5000	19 /BWIDTH
12	1	1	381	0	2.0000	0.00000	19 /FWIDTH
13	1	1	381	0	0.00000	0.00000	19 /RHOWB
14	1	1	381	0	0.00000	0.00000	19 /RKD1
15	1	1	381	0	0.00000	0.00000	19 /RKD2
16	1	1	381	-5	-2.8420	0.24000	19 /DIFF
17	1	1	381	0	0.0564	0.00000	19 /LAMBDA1W
18	1	1	1	-6	-3.0700	0.25000	/KH_01_1
19	1	1	1	-7	-1.1700	0.25000	/KH_06_1
20	1	1	1	-7	-0.77000	0.25000	/KH_06_2
21	1	1	1	-7	-1.0700	0.25000	/KH_06_3
22	1	1	1	-8	1.9700	0.25000	/KH_07_1
23	1	1	1	-8	1.9700	0.25000	/KH_07_2
24	1	1	1	-8	-2.0700	0.25000	/KH_07_3
25	1	1	1	-8	0.83000	0.25000	/KH_07_4
26	1	1	1	-8	-0.17000	0.25000	/KH_07_5
27	1	1	1	-8	-0.47000	0.25000	/KH_07_6
28	1	1	1	-8	0.23000	0.25000	/KH_07_7
29	1	1	1	-8	-0.17000	0.25000	/KH_07_8
30	1	1	1	-8	0.63000	0.25000	/KH_07_9
31	1	1	1	-9	-2.7700	0.25000	/KH_08_1
32	1	1	1	-10	-1.3700	0.25000	/KH_09_1
33	1	1	1	-11	-0.37000	0.25000	/KH_10_1
34	1	1	1	-11	-0.77000	0.25000	/KH_10_2
35	1	1	1	-12	0.10000	0.25000	/KH_11_1
36	1	1	1	-12	-0.77000	0.25000	/KH_11_2
37	1	1	1	-12	-0.47000	0.25000	/KH_11_3
38	1	1	1	-13	-0.99000	0.25000	/KH_12_1
39	1	1	1	-13	-0.99000	0.25000	/KH_12_2
40	1	1	1	-14	0.23000	0.25000	/KH_13_1
41	1	1	1	-14	-0.70000E-01	0.25000	/KH_13_2
42	1	1	1	-14	0.83000	0.25000	/KH_13_3
43	1	1	1	-15	1.4700	0.25000	/KH_14_1
44	1	1	1	-15	0.23000	0.25000	/KH_14_2
45	1	1	1	-15	0.23000	0.25000	/KH_14_3
46	1	1	1	-15	0.23000	0.25000	/KH_14_4
47	1	1	1	-15	-0.77000	0.25000	/KH_14_5
48	1	1	1	-16	-0.59000	0.25000	/KH_15_1
49	1	1	1	-16	-0.59000	0.25000	/KH_15_2
50	1	1	1	-16	-0.59000	0.25000	/KH_15_3
51	1	1	1	-16	-0.59000	0.25000	/KH_15_4
52	1	1	1	-16	-0.77000	0.25000	/KH_15_5

53	1	1	1	-17	1.4700	0.25000	/KH_16_1
54	1	1	1	-17	0.23000	0.25000	/KH_16_2
55	1	1	1	-17	0.23000	0.25000	/KH_16_3
56	1	1	1	-17	-0.77000	0.25000	/KH_16_4
57	1	1	1	-18	-0.59000	0.25000	/KH_17_1
58	1	1	1	-18	-1.5900	0.25000	/KH_17_2
59	1	1	1	-18	-1.7700	0.25000	/KH_17_3
60	1	1	1	-19	1.4700	0.25000	/KH_18_1
61	1	1	1	-19	1.1300	0.25000	/KH_18_2
62	1	1	1	-19	1.2300	0.25000	/KH_18_3
63	1	1	1	-19	0.53000	0.25000	/KH_18_4
64	1	1	1	-19	-0.77000	0.25000	/KH_18_5
65	1	1	1	-20	1.4700	0.25000	/KH_19_1
66	1	1	1	-20	1.0000	0.25000	/KH_19_2
67	1	1	1	-20	1.1300	0.25000	/KH_19_3
68	1	1	1	-20	1.2300	0.25000	/KH_19_4
69	1	1	1	-20	-0.77000	0.25000	/KH_19_5
70	1	1	1	-21	0.97000	0.25000	/KH_20_1
71	1	1	1	-21	0.57000	0.25000	/KH_20_2
72	1	1	1	-21	-2.0700	0.25000	/KH_20_3
73	1	1	1	-21	0.97000	0.25000	/KH_20_4
74	1	1	1	-21	0.97000	0.25000	/KH_20_5
75	1	1	1	-21	0.97000	0.25000	/KH_20_6
76	1	1	1	-22	0.00000	0.30000E-02	/LD_01_1
77	1	1	1	-23	0.00000	0.24000E-02	/LD_06_1
78	1	1	1	-23	0.00000	0.24000E-02	/LD_06_2
79	1	1	1	-23	0.00000	0.24000E-02	/LD_06_3
80	1	1	1	-24	0.00000	0.20000E-02	/LD_07_1
81	1	1	1	-24	0.00000	0.20000E-02	/LD_07_2
82	1	1	1	-24	0.00000	0.20000E-02	/LD_07_3
83	1	1	1	-24	0.00000	0.20000E-02	/LD_07_4
84	1	1	1	-24	0.00000	0.20000E-02	/LD_07_5
85	1	1	1	-24	0.00000	0.20000E-02	/LD_07_6
86	1	1	1	-24	0.00000	0.52000E-02	/LD_07_7
87	1	1	1	-24	0.00000	0.20000E-02	/LD_07_8
88	1	1	1	-24	0.00000	0.20000E-02	/LD_07_9
89	1	1	1	-25	0.00000	0.30000E-02	/LD_08_1
90	1	1	1	-26	0.00000	0.20000E-02	/LD_09_1
91	1	1	1	-27	0.00000	0.80000E-02	/LD_10_1
92	1	1	1	-27	0.00000	0.80000E-02	/LD_10_2
93	1	1	1	-28	0.70000E-03	0.45000E-02	/LD_11_1
94	1	1	1	-28	0.70000E-03	0.45000E-02	/LD_11_2
95	1	1	1	-28	0.70000E-03	0.45000E-02	/LD_11_3
96	1	1	1	-29	0.70000E-03	0.45000E-02	/LD_12_1
97	1	1	1	-29	0.70000E-03	0.45000E-02	/LD_12_2
98	1	1	1	-30	0.70000E-03	0.45000E-02	/LD_13_1
99	1	1	1	-30	0.70000E-03	0.45000E-02	/LD_13_2
100	1	1	1	-30	0.70000E-03	0.45000E-02	/LD_13_3
101	1	1	1	-31	0.70000E-03	0.45000E-02	/LD_14_1
102	1	1	1	-31	0.70000E-03	0.45000E-02	/LD_14_2
103	1	1	1	-31	0.70000E-03	0.45000E-02	/LD_14_3
104	1	1	1	-31	0.70000E-03	0.45000E-02	/LD_14_4
105	1	1	1	-31	0.70000E-03	0.45000E-02	/LD_14_5
106	1	1	1	-32	0.70000E-03	0.45000E-02	/LD_15_1
107	1	1	1	-32	0.70000E-03	0.45000E-02	/LD_15_2
108	1	1	1	-32	0.70000E-03	0.45000E-02	/LD_15_3
109	1	1	1	-32	0.70000E-03	0.45000E-02	/LD_15_4
110	1	1	1	-32	0.70000E-03	0.45000E-02	/LD_15_5
111	1	1	1	-33	0.70000E-03	0.45000E-02	/LD_16_1
112	1	1	1	-33	0.70000E-03	0.45000E-02	/LD_16_2
113	1	1	1	-33	0.70000E-03	0.45000E-02	/LD_16_3
114	1	1	1	-33	0.70000E-03	0.45000E-02	/LD_16_4
115	1	1	1	-34	0.70000E-03	0.45000E-02	/LD_17_1
116	1	1	1	-34	0.70000E-03	0.45000E-02	/LD_17_2
117	1	1	1	-34	0.70000E-03	0.45000E-02	/LD_17_3
118	1	1	1	-35	0.70000E-03	0.45000E-02	/LD_18_1
119	1	1	1	-35	0.70000E-03	0.45000E-02	/LD_18_2
120	1	1	1	-35	0.70000E-03	0.45000E-02	/LD_18_3
121	1	1	1	-35	0.70000E-03	0.45000E-02	/LD_18_4
122	1	1	1	-35	0.70000E-03	0.45000E-02	/LD_18_5
123	1	1	1	-36	0.70000E-03	0.45000E-02	/LD_19_1
124	1	1	1	-36	0.70000E-03	0.45000E-02	/LD_19_2
125	1	1	1	-36	0.70000E-03	0.45000E-02	/LD_19_3
126	1	1	1	-36	0.70000E-03	0.45000E-02	/LD_19_4
127	1	1	1	-36	0.70000E-03	0.45000E-02	/LD_19_5

128	1		1	-37	0.00000	0.74000E-02	/LD_20_1	
129	1		1	-37	0.00000	0.74000E-02	/LD_20_2	
130	1		1	-37	0.00000	0.74000E-02	/LD_20_3	
131	1		1	-37	0.00000	0.74000E-02	/LD_20_4	
132	1		1	-37	0.00000	0.74000E-02	/LD_20_5	
133	1		1	-37	0.00000	0.74000E-02	/LD_20_6	
134	1		1	-38	0.00000	0.25000	/RC_01_1	
	1	1	1	0	58	0	1	58