3 3679 00053 2913

Assessment of Effectiveness of Geologic Isolation Systems

FINITE-ELEMENT THREE-DIMENSIONAL
GROUND-WATER (FE3DGW) FLOW MODEL - FORMULATION,
PROGRAM LISTINGS AND USER'S MANUAL

S. K. Gupta
C. R. Cole
F. W. Bond

December 1979

Prepared for the
Office of Nuclear Waste Isolation
under its Contract with the
U.S. Department of Energy

Pacific Northwest Laboratory
Richland, Washington 99352

FOREWORD

The Assessment of Effectiveness of Geologic Isolation Systems (AEGIS) Program is developing and applying the methodology for assessing the far-field, long-term post-closure safety of deep geologic nuclear waste repositories. AEGIS is being performed by Pacific Northwest Laboratory (PNL) under contract with the Office of Nuclear Waste Isolation (ONWI) for the Department of Energy (DOE). One task within AEGIS is the development of methodology for analysis of the consequences (water pathway) from loss of repository containment as defined by various release scenarios.

Analysis of the long-term, far-field consequences of release scenarios requires the application of numerical codes which simulate the hydrologic systems, model the transport of released radionuclides through the hydrologic systems to the biosphere, and, where applicable, assess the radiological dose to humans.

Essentially three modeling technologies are involved in assessing the water pathway release consequence. These models are: 1) hydrologic models that define the groundwater flow field and provide water flow paths and travel times, 2) transport models that describe the movement and concentrations of the radionuclides in the flow field, and 3) dose models that determine the resultant radiation doses to individuals and/or populations. Figure i is a schematic flow diagram for the release consequence analysis.

The various input parameters required in the analysis are compiled in data systems. The data are organized and prepared by various input subroutines for use by the hydrologic and transport codes. The hydrologic models simulate the groundwater flow systems and provide water flow directions, rates, and velocities as inputs to the transport models. Outputs from the transport models are basically graphs of radionuclide concentration in the groundwater plotted against time. After dilution in the receiving surface-water body (e.g., lake, river, bay), these data are the input source terms for the dose models, if dose assessments are required. The dose models calculate radiation dose to individuals and populations.

FIGURE i. Schematic Diagram of Consequence Analysis

Hydrologic and transport models are available at several levels of complexity or sophistication. Model selection and use are determined by the quantity and quality of input data. Model development under AEGIS and related programs provides three levels of hydrologic models, two levels of transport models, and one level of dose models (with several separate models). The models and data systems are documented as follows:

- HYDROLOGIC MODELS:

- PNL-3162 PATHS Groundwater Hydrologic Model - first level (simplest) idealized hybrid analytical/numerical model for two-dimensional, saturated groundwater flow and single component transport; homogeneous geology.
- PNL-3160 VTT (Variable Thickness Transient) Groundwater Hydrologic Model - second level (intermediate complexity) two-dimensional saturated groundwater flow, Boussinesq approximation, finite difference approach; two-dimensional (quasi three-dimensional) multiaquifer capability; heterogeneous geology.
- PNL-2939 FE3DGW (Finite Element, Three-Dimensional Groundwater) Hydrologic Model - third level (high complexity) three-dimensional, finite element approach (Galerkin formulation) for saturated groundwater flow; heterogeneous geology.

- TRANSPORT MODELS:

- PNL-2970 GETOUT Transport Model - first level one-dimensional analytical solution considering radioactive chain decay with capability for only simple release and hydrologic functions; single speciation, constant flow rate, dispersion and sorption three-member straight delay chains.
- PNL-3179 MMT (Multicomponent Mass Transport) Model - second level, one-dimensional numerical, discrete parcel random walk (DPRW) algorithm; chain decay, single speciation, equilibrium sorption, time-variant leach rate and dispersion, n-membered straight or branched decay chains.

- DOSE MODELS:

- PNL-3180 ARRRG - drinking water, external exposure to aquatic food, water and shorelines, and FOOD - terrestrial food.
- PNL-3209 PABLM - Combination of ARRRG and FOOD with additional features related to chronic releases.
- BNWL-B-264 KRONIC - chronic external dose from air pathways.
- BNWL-B-351 SUBDOSA - acute external dose from air pathways.
- BNWL-B-389 DACRIN - chronic or acute inhalation dose from air pathways.

- DATA SYSTEMS:

- PNL-3139 SIRS (Sorption Information Retrieval System) - storage and retrieval system for experimental data on sorption/desorption analyses for a wide variety of radionuclides, groundwater compositions, and rocks and minerals.
- PNL-3161 CIRMIS (Comprehensive Information Retrieval and Model Input Sequence) Data System - storage and retrieval system for model input and output data, including graphical interpretation and display.

This document consists of the description of the FE3DGW (Finite Element, Three-Dimensional Groundwater) Hydrologic Model.

Return of the form on the last page of this report is required in order to remain on the Distribution List for future revisions of the model.

ACKNOWLEDGMENT

This research was supported by the Waste Isolation Safety Assessment Program (WISAP) conducted by Pacific Northwest Laboratory. The program was sponsored by the Office of Nuclear Waste Isolation, managed by Battelle Memorial Institute for the Department of Energy under Contract EY-76-C-06-1830. On 1 October 1979, WISAP became the Assessment of Effectiveness of Geologic Isolation Systems (AEGIS) Program and the Waste/Rock Interactions Technology (WRIT) Program. This report was issued by AEGIS.

CONTENTS

FOREWORD	iii
ACKNOWLEDGMENT	vii
FIGURES	xii
TABLES	xiv
1.0 INTRODUCTION	1-1
1.1 GENERAL	1-1
1.2 SALIENT FEATURES OF FINITE ELEMENT METHODS	1-2
1.3 NEED FOR A THREE-DIMENSIONAL FINITE ELEMENT MODEL	1-2
1.4 ORGANIZATION OF THE REPORT	1-3
2.0 GALERKIN FINITE ELEMENT FORMULATION OF THREE-DIMENSIONAL FLOW	2-1
2.1 GOVERNING PARTIAL DIFFERENTIAL EQUATIONS	2-1
2.2 GALERKIN APPROXIMATION	2-4
2.3 BASIS FUNCTIONS	2-5
2.4 GALERKIN APPROXIMATION FOR GROUND-WATER FLOW EQUATION	2-7
2.5 NONHOMOGENEITY AND ANISOTROPY	2-10
2.6 CONVERGENCE REQUIREMENTS	2-10
2.7 INTEGRATION OF APPROXIMATING EQUATIONS	2-10
2.8 FORMATION OF MATRICES	2-15
3.0 COMPUTATIONAL ORGANIZATION OF THE PROGRAM FOR REDUCING CORE STORAGE REQUIREMENTS AND EXECUTION TIME	3-1
3.1 PROGRAM ORGANIZATION TO REDUCE MEMORY REQUIREMENTS	3-1
3.2 REDUCTION OF PROCESSOR TIME	3-2
3.3 DIVISION OF THE MODEL INTO INDEPENDENT SUBPROGRAMS	3-4

4.0	MAIN FEATURES PROVIDED IN THE MODEL FOR CONVENIENT MATHEMATICAL DESCRIPTION OF A GIVEN GROUND-WATER SYSTEM	4-1
5.0	SUPPORTIVE PROGRAMS DEVELOPED TO VERIFY THE INPUT DATA USED IN SIMULATION	5-1
5.1	PLOTTING OF NODE LOCATIONS, ELEMENTS, AND VERTICAL LOGS	5-1
5.2	CONTOUR AND THREE-DIMENSIONAL PLOTTING OF INITIAL HEAD, TOP ELEVATION, AND THICKNESS OF MATERIALS	5-3
5.3	PLOTTING HELD POTENTIAL AND PRESCRIBED FLUX BOUNDARY CONDITIONS	5-9
5.4	DISPLAY OF RESULTS	5-9
5.5	ROLE OF THE SUPPORTIVE PROGRAM	5-9
6.0	VERIFICATION AND SENSITIVITY ANALYSIS WITH ANALYTICAL SOLUTIONS	6-1
6.1	VERIFICATION WITH THE THEIS NONEQUILIBRIUM EQUATIONS	6-1
6.2	VERIFICATION WITH 2-DIMENSIONAL ANALYTICAL SOLUTION, PATHS	6-3
6.3	VERIFICATION WITH HANTUSH SOLUTION FOR NONSTEADY FLOW TO FLOWING WELLS IN LEAKY AQUIFERS	6-14
7.0	SAMPLE ILLUSTRATIVE PROBLEMS	7-1
7.1	SUTTER BASIN, CALIFORNIA	7-1
7.2	HYPOTHETICAL WASTE REPOSITORY IN A REPRESENTATIVE SALT FORMATION	7-6
8.0	USER'S MANUAL FOR THREE-DIMENSIONAL FINITE ELEMENT GROUND-WATER MODEL	8-1
8.1	BRIEF DESCRIPTION OF PROVISIONS AND LOGIC USED IN DESCRIBING THE PHYSICAL SYSTEM	8-1
8.2	DIMENSION LIMITS OF PRESENT MODEL	8-3
8.3	DATA PREPARATION FOR EACH SUBPROGRAM	8-3
8.4	USER'S MANUAL FOR "PLOTTEL"	8-23

8.5	USER'S MANUAL FOR PROGRAM GRIDIT	8-27
8.6	USER'S MANUAL FOR PROGRAM GRIDIN	8-29
8.7	USER'S MANUAL FOR PROGRAM GRIDH	8-30
9.0	COMPUTER PROGRAM LISTINGS	9-1
9.1	PROG1 AND ITS SUBROUTINES, PROGRAM LISTINGS	9-6
9.2	PROG2, PROGRAM LISTINGS	9-27
9.3	BAND, PROGRAM LISTINGS	9-36
9.4	PROG3I, PROGRAM LISTINGS	9-40
9.5	PROG3 AND ITS SUBROUTINES, PROGRAM LISTINGS	9-44
9.6	OTHER SUPPORTIVE PROGRAMS, PROGRAM LISTINGS	9-72
10.0	SIMPLE ILLUSTRATIVE EXAMPLES OF INPUT DATA AND EXECUTION OF FE3DGW	10-1
10.1	GENERAL	10-1
10.2	LINEAR ELEMENT WITH HELD BOUNDARY CONDITIONS AT TOP AND BOTTOM	10-1
10.3	MIXED ORDER ELEMENT WITH SURFACE FLUX AT TOP AND HELD POTENTIAL AT BASE	10-13
10.4	MIXED ORDER ELEMENT WITH HELD AND FLUX BOUNDARY CONDITIONS AT TOP AND HELD POTENTIALS AT BASE	10-15
10.5	LINEAR ELEMENT WITH POINT SOURCE AT TOP AND HELD POTENTIALS AT BASE	10-15
10.6	RADIAL ELEMENTS WITH EXPONTIALLY INCREASING HORIZONTAL COORDINATES	10-15
11.0	REFERENCES	11-1

FIGURES

i	Schematic Diagram of Consequence Analysis	iv
4.1	Illustration of Proposed Logic for Discretization of Variations Encountered in a Large, Natural, Multilayered, Ground-water Regime	4-3
4.2	All Nodes Based on Variation in Aquifer Thickness of Elevation of Different Hydrogeologic Units and Stream Locations	4-5
4.3	Illustrative Example of Describing Interfaces of Two Materials in Well Log at a Node Location	4-8
4.4	Subdivision of the Given Region Into Two-Dimensional, Mixed-Order, Finite Elements Based on Nodalization of Figure 4.2	4-9
4.5	Three-Dimensional Finite Elements Generated from Two- Dimensional Surface Elements and Well Log Details at Each Node	4-11
5.1	Illustrative Sample Plot from PLOTTEL	5-2
5.2	The Output from Program GRIDIN Provides a Means for Three-Dimensional Display of the Input Data of Results.	5-5
5.3	Illustrative Example Checking the Input Data by Use of Program GRIDIN, Which Uses the Same Input File as Used for Simulation	5-7
5.4	Illustrative Plot of Grid Values from Progrm GRIDIN Output Using Input Data of Three-Dimensional Ground- Water Model of Long Island, New York	5-8
6.1	Diagrammatic Representation of Exponentially Increasing Nodal Arrangement Used for Finite Element Solution	6-2
6.2	Analytical and Finite Element Solutions of Theis Equations Drawdown Versus Time	6-4
6.3	Example PATHS Model Region	6-5
6.4	Three-Dimensional Representation of the PATHS Steady- State Analytical Solution	6-6
6.5	Combined Square (a) 600 ft, (b) 3000 ft, and (c) 1500 ft and Constant Radial Pattern	6-8

6.6	Comparison of Analytical Solution (PATHS) and 3-D Finite Element Solution for Grid Pattern Shown in Figure 6.5(c)	6-9
6.7	Combined Square (a) 6000 ft, (c) 3000 ft, and (c) 1500 ft and Variable Radial Pattern	6-10
6.8	Combined Square and Radial Element Pattern with (a) 3 and (b) 5 Layers of Nodes Around the Cavern	6-11
6.9	Radial Element Pattern with (a) Constant and (b) Variable Node Spacing	6-12
6.10	Radial Element Pattern with Mixed-Order Elements	6-13
6.11	Schematic Representation of Simulated Leaky Aquifer	6-15
6.12	Comparison of Dimensionless Drawdown Versus Dimensionless Time of Analytical and Finite Element Solution in a Leaky Aquifer	6-17
7.1	Location of the Vertical Fault and Water-Table Contours in the Sutter Basin Study Area	7-2
7.2	Geologic Formations and Vertical Faulting Along Geologic Section A-A' in the Study Area	7-3
7.3	Three-Dimensional, Finite Element Formulation of Sutter Basin, California	7-4
7.4	Estimated Hydraulic Heads (h) at the Base of the Post-Eocene Nonmarine Deposits	7-5
7.5	Estimated Hydraulic Heads (h) at the Top of the Kione Formation, Which Contains and Conveys Connate Water	7-7
7.6	Representative Generic Stratigraphic Section for Salt	7-8
7.7	Plan View of the Hypothetical Nuclear Waste Repository Model Region	7-10
7.8	Cross Section of the Five Geologic Layers Used to Define the Hypothetical Nuclear Waste Repository Model Region	7-10
7.9	Three-Dimensional View of the Hypothetical Nuclear Waste Repository Model Region	7-11

7.10	Contour Plot of the Model-Predicted Vertical Ground-Water Potentials with Superimposed Streamlines	7-14
8.1	Flow Chart of Input Data for PROG1	8-7
8.2	Flow Chart of Input Data for PROG3I	8-13
8.3	Sample of the Remote Terminal Input Required to Run the FE3DGW	8-22
10.1	Linear Element 4 Layers with Held B.C. At. Surface and Base .	10-2
10.2	Mixed Order Element - Flux (Read in PROG3I At Top - Held Potentials at Base	10-13
10.3	Mixed Order Elem-Bottom and One Top Held Head - Rest Surface Flux B.C.	10-17
10.4	Linear Element 4 Layers with Nodal Flux at Surface and Held B.C. at Base	10-20
10.5	Radial 4 Elements Three Layered and Exponential Increasing Distance	10-24

TABLES

2.1	Parameters of β_{ξ} , β_{η} and β_{ζ} for Mixed Corner	2-6
2.2	Abcissae and Weight Coefficients of the Gaussian Quadrature Formula	2-14
6.1	Variable Time Steps Used	6-4
6.2	Summary of Results	6-14
7.1	Hydrogeologic Parameters for Generic Salt Stratigraphy . .	7-9
7.2	Finite Element Model Hydrogeologic Input Parameters Used in Hypothetical Nuclear Waste Repository . . .	7-13
8.1	Explanation of Some Symbols Used in Flow Chart of PROG1	8-15
8.2	Brief Explanation of Symbols Used in Flow Chart for Input Data of PROG3I	8-20

9.1	Overlay and Task Command Files	9-3
10.1	Copy of Terminal Output for Installation of Tasks	10-3
10.2	One Linear Element (LINEAR.PR1)	10-4
10.3	PROG3I Input File (HSTEADY.P3I) for Steady State Simulation	10-4
10.4	Interactive Execution of Steady State Using LINEAR.PR1 and HSTEADY.P3I	10-5
10.5	PROG1 Ouput of LINEAR.PR1	10-6
10.6	PROG3 Steady State Output of LINEAR.PR1	10-10
10.7	Use of TRANSIENT.P3I for Transient Simulation with Three Variable Time Steps	10-10
10.8	Interactive Execution of LINEAR.PR1 with TRANSIENT.P3I and Variable Time Steps	10-11
10.9	Interactive Transient Execution of PROG3 with Constant Time Step	10-12
10.10	MIXELEM.PR1 for Mixed Order Element	10-14
10.11	QSURFACE.P3I for Surface Flux	10-14
10.12	Interactive Execution of Mixed Order Problem with MIXELEM.PR1 and QSURFACE.P3I	10-16
10.13	ELMIXBC.PR1 with Mixed Order Element with Mixed Boundary Conditions at Surface	10-18
10.14	Interactive Execution with ELMIXBC.PR1 and QSURFACE.P3I	10-19
10.15	LINODALQ.PR1 for a Linear Element with Point Source	10-21
10.16	Interactive Execution with LINODALQ.PR1 and HSTEADY.P3I	10-22
10.17	WELL.PR1 for Three Layered and Four Element Pie Shaped Region	10-25
10.18	PTRANS.P3I for Exponentially Incremented Time Step	10-26
10.19	Interactive Execution of Radial Problem	10-26
10.20	Sample Output of P3I and PR3 for WELL.PR1 and QTRANS.P3I	10-28

1.0 INTRODUCTION

1.1 GENERAL

The objective of transport modeling for the Waste Isolation Safety Assessment Program (WISAP) is to develop an integrated modeling system to predict, in the event of a breach, the movement of radioactive contaminants from a nuclear waste repository located in geologic media through the geosphere to the biosphere and to evaluate the potential radiation dose to humans. Prediction of radioactive contaminant transport requires an estimation of water movement since water is the primary driving force for waste movement in a hydrogeologic system. Hydrologic models define water flow paths and travel times from the input data provided by repository disruptive event analysis. Three levels of the hydrologic models have been categorized to handle varying complexities and degrees of available input parameters. The first level is for the simplest one-dimensional models having analytical solutions through idealized analytic or hybrid analytic models for two-dimensional single aquifer systems with scanty input data; the second level deals with more complex single or quasi-multilayered systems; and the third level is for complex multilayered systems.

The three-dimensional, finite element, ground-water model described in this report falls under the third level of hydrologic models. This model is capable of simulating single-layered systems having variable thickness or multilayered systems, where not only thickness can be varied, but the number of layers can be changed to agree with the vertical geologic section. Also, spacing of the model nodes can be varied as required. The source or sink terms can be defined at a given point (well) or along a given line (rivers, streams, etc.) or for a given region (variable surface infiltration from natural precipitation or irrigation). Pumping stresses in each layer of the subregion can be defined as a function of time. The geologic input data for such complex multilayered systems are reduced to well-log descriptions at each surface node and subdivision of the entire region into two-dimensional

elements. To provide for interaction between appropriate scientific disciplines, supportive programs have been developed to plot grid values, contour maps, and three-dimensional charts of both the input data used in simulation and the results obtained.

1.2 SALIENT FEATURES OF FINITE ELEMENT METHODS

Finite elements provide an alternative method of approximation, with much greater power than the finite-difference methodology. The finite-difference method needs special formulas for describing boundaries in many instances; the size of the grid is not changed easily, and accounting for inhomogeneities and anisotropy involves complications. On the other hand, in finite element methodology, the whole domain can be divided into irregular subdivisions as dictated by physical geometry of the problem. The size of each subdomain (element) can be varied easily; small elements can be used in areas of rapid change, and large elements where variations are less severe. Further, inhomogeneities and anisotropy are quite easily accommodated. The advantage of irregular subdivisions and higher-order approximations is exceptionally important when the spatial domain is complex.

1.3 NEED FOR A THREE-DIMENSIONAL, FINITE ELEMENT MODEL

Geohydrologic systems and surface-water bodies (e.g., lakes, rivers) usually have irregular boundaries, and the finite element method provides a powerful tool for space and boundary definition. The spacing of the nodes near the repository, pumping wells, or surface-water bodies can be narrowed, and the hydraulic conductivities can change abruptly from element to element so that fault zones and confining layers are represented.

Some of the complex generic and site-specific geologic configurations consist of multiaquifer systems separated by semiconfining layers, and these aquifers respond conjunctively to stresses imposed on either aquifer. Such coupling of aquifers requires modeling all of the layers simultaneously. A three-dimensional model provides a realistic means of representing multilayered aquifers associated with generic or specific repository sites.

1.4 ORGANIZATION OF THE REPORT

Section 2.0 describes partial differential equations of flow through saturated porous media and the Galerkin finite element formulation used in the model.

Need for a three-dimensional ground-water model was recognized for more than two decades, but due to high core storage requirements and excessive computational costs,^(1,2) three-dimensional models for ground-water simulation were not used until recently. Section 3.0 deals with salient features used in the model to reduce core storage and cost.

Main features provided in the model for convenient mathematical description of large and complex ground-water systems are dealt with in Section 4.0. The input data structure of this model is based on hydrogeologic data usually compiled for the ground-water systems. Because the accuracy of results is a function of the quality of the input, the input data must be thoroughly verifiable by modelers, field geologists, and hydrologists, in order for simulation results to be beneficial as a predictive or management tool. Supportive programs have been developed to provide means for interaction with geologists and hydrologists by plotting the node location, finite element discretization, well log (geologic section) at each node, thickness, top and bottom elevation of each hydrogeologic unit, and boundary conditions specified in the input data file prepared for simulation. Section 5 describes how these characteristics are plotted, again using the illustrative example of the ground-water system beneath Long Island, New York.

Validation of the numerical model with the analytical solution provides a check on the algorithm developed. Section 6 describes some of the test cases. Section 7 deals with some of the applications of the model. Sections 8 and 9 are, respectively, the user's manual and computer codings of the model.

2.0 GALLERKIN FINITE ELEMENT FORMULATION OF THREE-DIMENSIONAL FLOW

2.1 GOVERNING PARTIAL DIFFERENTIAL EQUATIONS

Three-dimensional, nonsteady flow is defined by the following equation:(3)

$$\frac{\partial}{\partial x} (K_x \frac{\partial h}{\partial x}) + \frac{\partial}{\partial y} (K_y \frac{\partial h}{\partial y}) + \frac{\partial}{\partial z} (K_z \frac{\partial h}{\partial z}) - Q = S_s \frac{\partial h}{\partial t} \quad (2-1)$$

S_s = specific storage L^{-1}

$$= \rho g (\alpha_p + \theta \beta_p)$$

g = gravity field strength, $L T^{-2}$

h = head above common datum, L

$$= z + \frac{1}{g} \int_{p_0}^p \frac{dp}{\rho(p)}$$

K_x, K_y, K_z = represents hydraulic conductivity of the saturated flow in the x, y , and z directions, $L T^{-1}$

p = fluid pressure, $M L^{-1} T^{-2}$

Q = strength of sink function defined by T^{-1}

$$= \sum_{i=1}^n Q_w(x_i, y_i, z_i, t) \delta(x-x_i) \delta(y-y_i) \delta(z-z_i)$$

Q_w = the well discharge from the aquifer, $L^3 T^{-1}$

δ = Dirac delta function

t = time, T

z = elevation above given datum, L

θ = porosity of medium, dimensionless

α_p = compressibility of medium, $LT^2 M^{-1}$

β_p = compressibility of liquid, $LT^2 M^{-1}$

ρ = density of fluid, $M L^{-3}$

Equation (2-1) is for a fixed coordinated system. Compressibility of medium and liquid is considered only in coefficient S_s , called "specific storage" by Hantash⁽⁴⁾ and defined as the "volume of water a unit volume of aquifer releases from storage under a unit decline of head." DeWiest⁽⁵⁾ questioned the validity of Equation (2-1) on the fact that "on one side of the equation the net inward mass flux was calculated for a volume element without deformation, while on the other side of the equation, to compute rate of change of mass inside the volume element, the element itself was deformed." DeWiest derived the following equation:

$$\nabla^2 h - 2\rho\beta_p g \frac{\partial h}{\partial z} = \frac{S^*}{K} \frac{\partial h}{\partial t}$$

where

$$S^* = \rho g [(1 - \theta)\alpha_p + \theta\beta_p]$$

= specific storage, L^{-1}

Cooper⁽⁶⁾ distinguished the rate of flow relative to the moving grains of the medium and the rate of flow across fixed boundaries on the control volume while material is deforming; the former obeys Darcy's Law, while the latter does not. Cooper derived the following flow equation in terms of fixed coordinates, assuming that fluid conductivity K remains constant

during deformation of the material:

$$\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} - 2\rho g \beta_p \frac{\partial h}{\partial z} = \frac{\rho g}{K} (\alpha_p + \theta \beta_p) \left(\frac{\partial h}{\partial t} - \omega g \right)$$

where

$$\frac{dh}{dt} = \left(\frac{\partial h}{\partial t} + \omega g \frac{\partial h}{\partial z} \right)$$

ωg = vertical component of velocity of grains of the medium, LT^{-1}

For an equation of flow in deforming coordinates, Cooper derived the following equation:

$$\nabla^2 h - 2\rho \beta_p g \frac{\partial h}{\partial z'} = \frac{\rho g}{K} (\alpha_p + \theta \beta_p) \frac{\partial h}{\partial t}$$

where

z' = deforming spatial coordinate, L

Cooper also suggested that in almost all problems, the second term on the left is negligible, so the equation can be very closely approximated as:

$$\nabla^2 h = \frac{S_s}{K} \frac{\partial h}{\partial t}$$

Cooper concluded that the three-dimensional equation (2-1) as proposed by Jacob⁽³⁾ "appears to be exact if one considers his z coordinate to be the deforming coordinate z' ." The difference between the fixed coordinate and the deformed coordinate is negligible in most aquifer materials, except for those clay materials which compact substantially with the change of pressure of the fluid.

This work considers deep-seated aquifers, which have minor variations in head with respect to time. Equation (2-1) proposed by Jacob has been used. The program, after minor modifications, is capable of using any other alternate equation defining flow through porous media in three dimensions.

2.2 GALERKIN APPROXIMATION

Equation (2-1) can be rewritten in a more compact form as:

$$L(h) \equiv \frac{\partial}{\partial x_{\alpha}} (K_{\alpha\beta} \frac{\partial h}{\partial x_{\beta}}) - Q - S_s \frac{\partial h}{\partial t} = 0 \quad (2-2)$$

$$\alpha, \beta = 1, 2, 3$$

In the above equation and in subsequent equations, subscripts α and β are used to describe the three-dimensional flow equation in condensed form.

To solve $L(h) = 0$ by the Galerkin method, a trial solution of the following form is assumed:

$$h(x_{\alpha}, t) \approx h'(x_{\alpha}, t) = \sum_{i=1}^n H_i(t) V_i(x_{\alpha}) \quad (2-3)$$

where $V_i(x_{\alpha})$ ($i = 1, 2 \dots n$) is a system of functions (basis functions or bases) chosen beforehand and satisfying the essential boundary conditions imposed on Equation (2-1). The functions $V_i(x_{\alpha})$ ($i = 1, 2 \dots n$) are assumed to be linearly independent and to represent the first n functions of some system of functions which is complete in the given region. The functions $H_i(t)$ are undetermined coefficients that, as shown later, are the solution of Equation (2-1) at specified points (or nodes) in region Ω .

The approximating function $h'(x_{\alpha}, t)$ will be an exact solution to Equation (2-1) only if $L(h')$ is equal to zero. This can be best achieved in a variational sense using the definition of orthogonal functions. The orthogonality of expression $L(h')$ is required to all the basis functions

$V_i(x) \ (i = 1, 2 \dots n)$ or

$$\iiint_{\Omega} L(h'[x_{\alpha}, t]) V_i(x_{\alpha}) d\Omega = 0 \quad (2-4)$$

$$i = 1, 2 \dots n$$

Since only n basis functions have been selected, there are n undetermined coefficients $H_i(t)$ ($i = 1, 2 \dots n$) and therefore only n conditions of orthogonality can be satisfied. These conditions are stated as:

$$\iiint_{\Omega} L(h'[x_{\alpha}, t]) V_i(x_{\alpha}) d\Omega = \iiint_{\Omega} L\left(\sum_{j=1}^n H_j[t] V_j[x_{\alpha}]\right) V_i(x_{\alpha}) d\Omega = 0 \quad (2-4a)$$

$$i = 1, 2 \dots n$$

Assuming that the appropriate integrations can be performed, the desired solution of Equation (2-2) is obtained by substituting the values of $H_i(t)$ into Equation (2-3).

The above-mentioned formulation of the Galerkin equation was applied by Pinder and Frind⁽⁷⁾ to ground-water simulations. In this approximation technique, there is no direct dependence on variational principles, and it is a universal method applicable to equations of elliptic, parabolic, or hyperbolic type. This was applied to three-dimensional ground-water flow by Gupta et al.⁽⁸⁾

2.3 BASIS FUNCTION

The suitability of the Galerkin approximation for computer application is largely based on the choice of basis functions $V_i(x_{\alpha})$. Efficient numerical schemes can be developed when continuous piecewise polynomial functions are used, e.g., Price et al.,⁽⁹⁾ Cavendish et al.,⁽¹⁰⁾ and Culham and Varga.⁽¹¹⁾ In selecting these functions a series of nodes is chosen in the domain, and basis functions are defined such that $V_i(x, y, z)$ is unity at node i and zero at all other nodes. When functions are chosen with these constraints, the undetermined coefficients $H_i(t)$ ($i = 1, 2, \dots n$) are then the required function $h'(x_{\alpha}, t)$ at the n points.

The three-dimensional mixed basis functions⁽⁸⁾ are given below.

Corner Nodes

$$V_i(\xi, \eta, \zeta) = \alpha_i \beta_i$$

where

$$\alpha_i = \frac{1}{8} (1 + \xi \xi_i) (1 + \eta \eta_i) (1 + \zeta \zeta_i)$$

$$\beta_i = \beta_\xi + \beta_\eta + \beta_\zeta$$

Values of β_ξ , β_η and β_ζ are determined by the order of the element sides ($\zeta = \pm 1$, $\eta = \pm 1$), ($\zeta = \pm 1$, $\xi = \pm 1$) and ($\xi = \pm 1$, $\eta = \pm 1$), respectively, in conjunction with information from the table given below.

TABLE 2.1 Parameters of β_ξ , β_η and β_ζ for Mixed Corner

Order of Side	β_ξ	β_η	β_ζ
Linear	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
Quadratic	$\xi \xi_i - \frac{2}{3}$	$\eta \eta_i - \frac{2}{3}$	$\zeta \zeta_i - \frac{2}{3}$
Cubic	$\frac{9}{8} \xi^2 - \frac{19}{24}$	$\frac{9}{8} \eta^2 - \frac{19}{24}$	$\frac{9}{8} \zeta^2 - \frac{19}{24}$

Nodes Along the Sides of Elements

A typical midside node of a quadratic side:

$$V_i = \frac{1}{4} (1 - \xi^2) (1 + \eta\eta_i) (1 + \zeta\zeta_i) \text{ for } \xi_i = 0, \eta_i = \pm 1, \zeta_i = \pm 1$$

$$V_i = \frac{1}{4} (1 + \xi\xi_i) (1 - \eta^2) (1 + \zeta\zeta_i) \text{ for } \xi_i = \pm 1, \eta_i = 0, \zeta_i = \pm 1$$

$$V_i = \frac{1}{4} (1 + \xi\xi_i) (1 + \eta\eta_i) (1 - \zeta^2) \text{ for } \xi_i = \pm 1, \eta_i = \pm 1, \zeta_i = 0$$

A typical midside node of a cubic side:

$$V_i = \frac{9}{64} (1 - \xi^2) (1 + 9\xi\xi_i) (1 + \eta\eta_i) (1 + \zeta\zeta_i) \text{ for } \xi_i = \pm \frac{1}{3}, \eta_i = \pm 1, \zeta_i = \pm 1$$

$$V_i = \frac{9}{64} (1 + \xi\xi_i) (1 - \eta^2) (1 + 9\eta\eta_i) (1 + \zeta\zeta_i) \text{ for } \xi_i = \pm 1, \eta_i = \pm \frac{1}{3}, \zeta_i = \pm 1$$

$$V_i = \frac{9}{64} (1 + \xi\xi_i) (1 + \eta\eta_i) (1 - \zeta^2) (1 + 9\zeta\zeta_i) \text{ for } \xi_i = \pm 1, \eta_i = \pm 1, \zeta_i = \pm \frac{1}{3}$$

A convenient method of establishing the coordinate transformations from Cartesian to local ξ, η, ζ , space is to use the basis functions given above. The points with coordinates, x, y , and z , will lie at corresponding point, ξ, η, ζ , in the element as given by the general definitions of the basis functions. By these relationships, each set of local coordinates will correspond to a set of global Cartesian coordinates and in general to only one such set. Nonuniqueness arises only with violent element distortions.

2.4 GALERKIN APPROXIMATION FOR GROUND-WATER FLOW EQUATION

After generating the appropriate basis function for each node in domain Ω , it is necessary to solve Equation (2-4) for the undetermined coefficients $H_i(t)$. By substituting Equations (2-2) and (2-3) in Equation (2-4a) along

with the assumption that the principal components of the hydraulic conductivity tensor are co-linear with x, y and z, the following is obtained for three dimensions:

$$\iiint_{\Omega} \left\{ \frac{\partial}{\partial x_{\alpha}} \left(K_{\alpha\beta} \frac{\partial}{\partial x_{\beta}} \sum_{j=1}^n H_j V_j \right) - S_s \frac{\partial}{\partial t} \sum_{j=1}^n H_j V_j - Q \right\} V_i d\Omega = 0 \quad (2-4b)$$

$$i = 1, 2 \dots n$$

where the domain Ω is composed of all elements over which the i^{th} basis function is defined.

To eliminate the second derivatives in Equation (2-4b), Green's theorem in the following form can be used:

$$\iiint_{\Omega} \left[\psi \frac{\partial^2 \phi}{\partial x_{\alpha}^2} \right] d\Omega = - \iiint_{\Omega} \frac{\partial \phi}{\partial x_{\alpha}} \frac{\partial \psi}{\partial x_{\alpha}} d\Omega + \iint_{\lambda} \psi \frac{\partial \phi}{\partial x_{\alpha}} \ell_{\alpha} d\lambda \quad (2-5)$$

Assuming $K_{\alpha\beta}$ to be constant over each domain of integration and noting that H_j is a function of time only, and using Green's theorem Equation (2-5), Equation (2-4b) can be written as:

$$\begin{aligned} \iiint_{\Omega} \sum_{j=1}^n \left(K_{\alpha\beta} \frac{\partial V_i}{\partial x_{\alpha}} \frac{\partial V_j}{\partial x_{\beta}} \right) H_j d\Omega + \iiint_{\Omega} S_s V_i \sum_{j=1}^n V_j \frac{\partial H_j}{\partial t} d\Omega \\ + \iiint_{\Omega} Q V_i d\Omega - \iint_{\lambda} V_i \sum_{j=1}^n \left(K_{\alpha\beta} \frac{\partial V_j}{\partial x_{\beta}} \ell_{\alpha} \right) H_j d\lambda = 0 \end{aligned} \quad (2-6)$$

$$i = 1, 2 \dots n$$

The n equations in (2-6) can be written in the matrix form as:

$$[P] \{H\} + [R] \{dH/dt\} + \{U\} = 0 \quad (2-7)$$

where [P] and [R] are n x n matrices in which:

$$P_{ij} = \iiint_{\Omega} (K_{\alpha\beta} \frac{\partial V_i}{\partial x_{\alpha}} \frac{\partial V_j}{\partial x_{\beta}}) d\Omega$$

$$R_{ij} = \iiint_{\Omega} S_s V_i V_j d\Omega$$

and {U} is a vector in which:

$$U_i = \iiint_{\lambda} Q V_i d\Omega - \iint_{\lambda} V_i \sum_{j=1}^n (K_{\alpha\beta} \frac{\partial V_j}{\partial x_{\beta}} l_{\alpha}) H_j d\lambda \quad (2-8)$$

The last term in Equation (2-8) incorporates the Neumann boundary condition:

$$- K_{\alpha\beta} \frac{\partial h}{\partial x_{\beta}} l_{\alpha} = q_n$$

where q_n is the flux of water per unit area of boundary S, L/T. This last term is formed only when q_n is nonzero, in which case it takes the form of:

$$\iint_{\lambda} V_i q_n d\lambda$$

Under these conditions, the flux prescribed along the boundary can be integrated, and the weighted average value for each node of total outward flux used for the specific node under consideration. At nodes where a Dirichlet or constant-head boundary condition is encountered, Equation (2-6) is not generated. The matrices of Equation (2-7) are partitioned to account for these passive nodes to minimize the number of operations required for solution.

2.5 NONHOMOGENEITY AND ANISOTROPY

In the formulation of Equation (2-6) no derivatives of the hydraulic conductivity coefficients ($K_{\alpha\beta}$) occur, and therefore the formulation is valid for constant or variable values. The conductivities can change abruptly from element to element if required by the prototype system. Moreover, the general form of Equation (2-6) readily accounts for anisotropy when it exists and the appropriate parameters are known.

2.6 CONVERGENCE REQUIREMENTS

Since the formulation is defined by the first derivatives of h , continuity of only h is necessary when choosing suitable basis functions. Also, basis functions must be defined such that constant values of any of the first derivatives are available throughout the element when suitable nodal values of $\{h\}^e$ are assigned. The basis functions used in the present formulation satisfy all of the above.

2.7 INTEGRATION OF APPROXIMATING EQUATIONS

The finite element process converges if integration is sufficient to evaluate exactly the volume of the elements. The Gaussian quadrature scheme is generally used for numerical integration. Numerically integrated finite elements provide greater versatility than those employing analytical integration. For a general class of problems, the matrices are always of the same form in terms of the shape function and its derivatives. In the computer program, the shape functions and their derivatives are specified, along with the order of integration (number of Gaussian points). The use of universal shape function routines has a unique practical advantage in that, once the routine is checked decisively for errors, the computer works efficiently in dealing with any new situation or problem. This function avoids any possibility of inherent mistakes.

If the nodal values are correct and an adequate minimum number of Gaussian points is prescribed, numerical integration results in identical precision. Finite element integration is briefly described below.

Numerical approximation is carried out with respect to the nodal values, and each node is described in terms of shape factor V_i given in local coordinates; thus, we have the following relationships considering each element as a domain:

$$x = \sum_{i=1}^n V_i(\xi, \eta, \zeta) x_i \quad (2-9)$$

$$y = \sum_{i=1}^n V_i(\xi, \eta, \zeta) y_i \quad (2-10)$$

$$z = \sum_{i=1}^n V_i(\xi, \eta, \zeta) z_i \quad (2-11)$$

$$H = \sum_{i=1}^n V_i(\xi, \eta, \zeta) H_i \quad (2-12)$$

where x_i , y_i , and z_i are actual values of nodal Cartesian coordinates and n is the number of nodes in the element under consideration. H_i (as defined earlier) is the actual value of the required function $h(x, y, z, t)$.

Considering the set of local coordinates ξ, η , and ζ and a corresponding set of global coordinates x, y , and z , by usual rules of partial differentiation, we can write, for instance, the ξ derivatives as:

$$\frac{\partial V_i}{\partial \xi} = \frac{\partial V_i}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial V_i}{\partial y} \frac{\partial y}{\partial \xi} + \frac{\partial V_i}{\partial z} \frac{\partial z}{\partial \xi} \quad (2-13)$$

Performing the same differentiation with respect to the other two coordinates and writing in matrix form we have:

$$\begin{bmatrix} \frac{\partial V_i}{\partial \xi} \\ \frac{\partial V_i}{\partial \eta} \\ \frac{\partial V_i}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{Bmatrix} \frac{\partial V_i}{\partial x} \\ \frac{\partial V_i}{\partial y} \\ \frac{\partial V_i}{\partial z} \end{Bmatrix} = [J] \begin{Bmatrix} \frac{\partial V_i}{\partial x} \\ \frac{\partial V_i}{\partial y} \\ \frac{\partial V_i}{\partial z} \end{Bmatrix} \quad (2-14)$$

In the above, the left-hand side can be evaluated since the functions V_i are specified in local coordinates. Furthermore, x , y , and z are explicitly given by the relation defining the curvilinear coordinates [Equations (2-9), (2-10), and (2-11)]. The matrix $[J]$ can be formed explicitly in terms of local coordinates. This matrix is known as the "Jacobian matrix".

To find the global derivatives, we invert $[J]$ and write

$$\begin{Bmatrix} \frac{\partial V_i}{\partial x} \\ \frac{\partial V_i}{\partial y} \\ \frac{\partial V_i}{\partial z} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial V_i}{\partial \xi} \\ \frac{\partial V_i}{\partial \eta} \\ \frac{\partial V_i}{\partial \zeta} \end{Bmatrix} \quad (2-15)$$

In terms of the shape function defining the coordinate transformation $[V']$, which are identical with the shape functions $[V]$ when isoparametric formulation is used, we have:

$$[J] = \begin{bmatrix} \sum \frac{\partial V_i}{\partial \xi} x_i, & \sum \frac{\partial V_i}{\partial \xi} y_i, & \sum \frac{\partial V_i}{\partial \xi} z_i \\ \sum \frac{\partial V_i}{\partial \eta} x_i, & \sum \frac{\partial V_i}{\partial \eta} y_i, & \sum \frac{\partial V_i}{\partial \eta} z_i \\ \sum \frac{\partial V_i}{\partial \zeta} x_i, & \sum \frac{\partial V_i}{\partial \zeta} y_i, & \sum \frac{\partial V_i}{\partial \zeta} z_i \end{bmatrix}$$

The transformation of the variables and region, with respect to which integration is made, is achieved by the following:

$$dx \, dy \, dz = \det [J] \, d\xi \, d\eta \, d\zeta$$

which is valid irrespective of the number of coordinates used. Assuming that the inverse of $[J]$ can be found, by transformation we have

$$P_{ij} = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \left(K_x \frac{\partial V_i}{\partial x} \frac{\partial V_j}{\partial x} + K_y \frac{\partial V_i}{\partial y} \frac{\partial V_j}{\partial y} + K_z \frac{\partial V_i}{\partial z} \frac{\partial V_j}{\partial z} \right) \det [J] \, d\xi \, d\eta \, d\zeta \quad (2-16)$$

Similar expressions are developed for the remaining terms of Equation (2-8).

To perform an integration of the form indicated in Equation (2-16), a Gaussian quadrature scheme is used. In this technique a polynomial $f(x)$ of degree $2N-1$ may be integrated exactly as the weighted mean of its N particular values of specified points (Gaussian points). Examining in detail the form of Jacobian determinant (expressed in terms of shape function derivatives), and its order, the number of Gaussian points required for exact integration can be determined. Zienkiewicz and Cheung⁽¹²⁾ give a concise discussion of Gaussian quadrature and tables of required coefficients (Table 2.2). Zienkiewicz⁽¹³⁾ summarizes that if only a few elements are used to represent a region, more than a minimum number of Gaussian points is advised. If a large number of elements have to be used, the lowest necessary order of

TABLE 2.2. Abscissae and Weight Coefficients of the Gaussian Quadrature Formula

$$\int_{-1}^1 f(x) dx = \sum_{j=1}^n H_j f(a_j),$$

<u>+a</u>		<u>H</u>
	n = 2	
0.57735 02691 89626		1.00000 00000 00000
	n = 3	
0.77459 66692 41483		0.55555 55555 55556
0.00000 00000 00000		0.88888 88888 88889
	n = 4	
0.86113 63115 94053		0.34785 48451 37454
0.33998 10435 84856		0.65214 51548 62546
	n = 5	
0.90617 98459 38664		0.23692 68850 56189
0.53846 93101 05683		0.47862 86704 99366
0.00000 00000 00000		0.56888 88888 88889
	n = 6	
0.93246 95142 03152		0.17132 44923 79170
0.66120 93864 66265		0.36076 15730 48139
0.23861 91860 83197		0.46791 39345 72691
	n = 7	
0.94910 79123 42759		0.12948 49661 68870
0.74153 11855 99394		0.27970 53914 89277
0.40584 51513 77397		0.38183 00505 05119
0.00000 00000 00000		0.41795 91836 73469
	n = 8	
0.96028 98564 97536		0.10122 85362 90376
0.79666 64774 13627		0.22238 10344 53374
0.52553 24099 16329		0.31370 66458 77887
0.18343 64624 95650		0.36268 37833 78362
	n = 9	
0.96816 02395 07626		0.08127 43883 61574
0.83603 11073 26636		0.18064 81606 94857
0.61337 14327 00590		0.26061 06964 02935
0.32425 34234 03809		0.31234 70770 40003
0.00000 00000 00000		0.33023 93550 01260
	n = 10	
0.97390 65285 17172		0.06667 13443 08688
0.86506 33666 88985		0.14945 13491 50581
0.67940 95682 99024		0.21908 63625 15982
0.43339 53941 29247		0.26926 67193 09996
0.14887 43389 81631		0.29552 42247 14753

integration is economical. It is also reported that considerable improvement results from using a minimum integrating order. The computer program written has a choice of the use of 2, 3, or 5 Gaussian integration points.

2.8 FORMATION OF MATRICES

Once the matrices of Equation (2-7) have been developed, the task is to solve for the undetermined coefficients $H_i(t)$ ($i = 1, 2 \dots n$). To eliminate the term $\partial H / \partial t$ associated with the P matrices, a backward finite-difference approximation for the time is generally used with given initial values of $H(0)$:

$$[P] \{H\}_{t+\Delta t} + [R] (\{H\}_{t+\Delta t} - \{H\}_t) \Delta t + \{U\} = 0$$

or

$$([P] + [R]/\Delta t) \{H\}_{t+\Delta t} = [R]/\Delta t \{H\}_t - \{U\} \quad (2-17)$$

where values occurring on the right-hand side are known.

For finite difference in time, the Crank-Nicholson or central-difference scheme can be used after the first initial step. In past versions of the model, backward difference was used for the initial time step, but in the present model, rather than simulation of H , the computer program has been modified to simulate changes during the time step.

3.0 COMPUTATIONAL ORGANIZATION OF THE PROGRAM FOR REDUCING CORE STORAGE REQUIREMENTS AND EXECUTION TIME

The three-dimensional, finite-element method has the major limitations of requiring large organization and programming efforts and considerable core storage, limiting the size of problems handled on a given computer. In DAVIS-FE⁽⁸⁾ the core storage requirements and cost of computation for a three-dimensional ground-water flow were reduced by use of small independent subprograms, storing intermediate results which are needed repeatedly, and use of a partially compressed matrix for storing and solution of the sparse and unbanded system of equations. Further improvements of the DAVIS-FE and equation solver were made for application to multilayered ground-water systems beneath Long Island, New York.⁽¹⁴⁾

Under WISAP, additional modifications were carried out and a new version, FE3DGW (Finite Element 3-Dimensional Ground Water), was developed to simulate multilayered systems on a small computer (PDP-11/45 having maximum core storage of 32 K for 16-bit words). This section gives a brief description of the organization of the computer program.

3.1 PROGRAM ORGANIZATION TO REDUCE MEMORY REQUIREMENTS

Input Data

Reduction of core storage can be achieved by using a disk, provided the data are sequentially written and retrieved. Therefore, the input data for the finite-element simulation were grouped into two major categories:

Data Used at Random

Data related to material properties, nodal coordinates and boundary conditions are needed at random, as per the nodes of given elements. Therefore, this information is stored in the core memory of the given subroutine whenever possible. For simulation on a small computer, if all the nodal coordinates cannot be accommodated in core storage, they are read in as multiples of 256 integer (128 real) words, the equivalent of one record of disk block.

Data Used in Sequential Form

Element related information forms the bulk of storage requirements. Because numerical integration is carried out sequentially element by element, the element-related information is handled through disks.

System Stiffness Matrix

For a natural basin involving many layers of three-dimensional elements, the system stiffness matrix is large, sparse and unbanded. The number of unknowns in a natural multiaquifer system may be as high as 1000 or more, whereas nonzero coefficients for linear, quadratic and cubic order three-dimensional elements are respectively only 27, 81, and 135. If mixed order three-dimensional isoparametric elements are used, in general the number of unknowns in a given row is 40 to 60.

The use of two arrays, one for storing the nonzero elements and the other for column identifications, has been found to be very efficient in reducing the memory requirement for large sparse, unbanded and asymmetric systems of equations.⁽¹⁵⁾ Since provision must be made for the longest row of nonzero coefficients, storage of a limited number of zeros is required in some of the rows, resulting in partially packed arrays. Such a storage scheme is economical if the longest row after elimination process contains less than 50% nonzero elements.

3.2 REDUCTION OF PROCESSOR TIME

Storage of Intermediate Results Needed Repeatedly for Transient or Steady-State Solution

Numerical approximation is carried out with respect to the nodal values, and each node is described in terms of shape factor V_i , given in local coordinates. Details of the steps involved in numerical integration have been given in Section 2. Various components of Equation (2-17) are estimated in the following manner:

- a. For a linear element a minimum of two Gaussian points in each direction are required, which warrants estimation of shape function V_i ,

$$\frac{\partial V_i}{\partial \xi}, \frac{\partial V_i}{\partial \eta}, \frac{\partial V_i}{\partial \zeta}$$

and the matrix Jacobian $[J]$, eight times for each element. For a quadratic or cubic order element, the minimum is 27.

- b. In Equation (2-6), the only parameters required are

$$\frac{\partial V_i}{\partial \xi} \frac{\partial V_j}{\partial \xi} \det [J], \frac{\partial V_i}{\partial \eta} \frac{\partial V_j}{\partial \eta} \det [J], \frac{\partial V_i}{\partial \zeta} \frac{\partial V_j}{\partial \zeta} \det [J], V_i V_j \det [J] \text{ and}$$

$V_i \det [J]$, respectively called FIFJ, GIGJ, HIHJ, XIXJOB and XIJCOB. The values of these parameters are estimated at each Gaussian point and summed up for the given element before writing to disk. These parameters are functions of the x, y, and z coordinates of each node of the element and remain constant for steady-state and confined transient cases. The economy achieved by storing intermediate steps can be evident from the fact that Gaussian numerical integration requires 8 or 27 times estimation (depending on the order of the element) of the shape function, V_i , and its derivatives for each node of the element for every solution. For each quadratic or cubic order element this amounts to 1188 times estimation of each of the integration parameters called FIFJ, GIGJ, HIHJ, XIXJOB and XIJCOB.

- c. The element stiffness matrix is formed by rewriting Equation (2-17) in the following form:

$$P_{ij} = \text{FIFJ} \times K_{xx} = \text{GIGJ} \times K_{yy} + \text{HIHJ} \times K_{zz}$$

$$R_{ij} = \rho g (\alpha_p + \phi \beta_p) \times \text{XIXJOB}$$

$$U_i = Q \times \text{XIJCOB}$$

For each transient or steady-state problem, the stiffness matrix of the region is developed by reading the summed-up values of FIFJ, GIGJ, HIHJ, XIXJOB and XIJCOB for each node of the given element.

Solution of System Matrix with Minimal Arithmetic Operation

In finite element formulation, the subroutine for the solution of simultaneous equations constitutes another major influence on computational

cost. The computer program uses EQSOLV⁽¹⁵⁾ in which: 1) the row with the minimum number of nonzero elements is selected as the pivot row and to avoid instability, the largest absolute element in the pivotal row is used as the pivotal column; 2) the coefficients matrix is decomposed to the upper triangle, and sequential operations and coefficients for right-hand constant vector are stored such that, for constant time steps, repeated solutions are obtained by back substitution without decomposing the system matrix again; 3) the bulk of the program centers on a search of the partially packed column matrix to locate elements to be operated on, and thus actual arithmetic operations are minimal.

3.3 DIVISION OF THE MODEL INTO INDEPENDENT SUBPROGRAMS

For efficient handling of the input data, for providing checks at various stages of simulations, and for computational efficiencies, the computer model has been subdivided into the following five independent subprograms.

- a. PROG1 - This program reads the input data, assigns a new number to each node for internal estimation, generates a three-dimensional mesh, estimates the number of unknowns and generates File-Q* disk files for data processing in other subsequent programs.

By making this program independent, the modeler is provided with a means of critically examining the input data before proceeding with execution of the other programs. In other programs, reduction of memory requirement is achieved by retrieval of only relevant information.

* File-Q is a set of FORTRAN callable direct I/O subroutines written for use on PDP 11/45. The File-Q file structure for disk volumes is designed to minimize the overhead associated with direct access file operations. Once File-Q has been opened, only basic read-write limits are checked for each I/O operation and all direct access I/O transfers are started at the beginning of a physical (256 word) disk block boundary.

The computer output from large basins may run to several pages that may be needed only for the initial run and not for further additional minor modifications. Therefore, the computer program has varied options from no output to complete listing of various details.

- b. PROG2 - This program calculates the integration parameters (described in Section 3.2) needed repeatedly but which remain constant for the given x, y, z coordinates. These integration parameters are stored sequentially for each element and are retrieved for transient or steady-state solutions with different boundary conditions or hydraulic properties.
- c. BAND - Simulation of ground-water basins involves a large, sparse and unbanded system of equations. The equation solver uses partially packed arrays by storing only nonzero elements in one matrix and their column identification in another matrix. Initially as well as during decomposition of the system matrix into upper triangular form, the nonzero columns of each row are arranged in ascending order, reducing processor time by conducting the search for specific elements until the column identification of the row, being searched, is exceeded.

Through subprogram BAND, the preassigned locations of elements for initial formation of system matrix are calculated. This reduces considerably the computational efforts necessary in rearranging the columns during repeated formations of the system matrix.

- d. PROG3I - For a given problem it is usually desired to simulate potentials with different time steps, recharge rates and pumping stresses. PROG3I has been structured to read or to redefine these values for each simulation run. Thus PROG1, PROG2 and BAND are executed only once. Through PROG3I and PROG3, simulation for various combinations of time steps and boundary values are carried out.

- e. PROG3 - This is the main program which calculates the potential head using File-Q disk data generated by PROG1, PROG2, BAND and PROG3I.

Maximum efforts in computation of head are involved in formation of element matrices, the system matrix and decomposition of the system matrix into an upper triangular form. For a constant time step simulation, the coefficients of the system matrix are invariant. PROG3 has been structured to decompose the system matrix only at changes of time step magnitude. For constant time steps, the solution is achieved by recalculating the right-hand constant vector as a function of boundary conditions, prescribed flux rates and pumping stresses, followed by back substitution into the decomposed upper triangular matrix.

4.0 MAIN FEATURES PROVIDED IN THE MODEL FOR CONVENIENT MATHEMATICAL DESCRIPTION OF A GIVEN GROUND-WATER SYSTEM

In contrast to surface-water supplies, ground-water reservoirs are interconnected over large areas, and their definition and analysis require difficult and extensive geologic investigations. Geologists and geohydrologists usually define various layering patterns by a few typical well logs (defining the vertical geologic section), interpreted cross sections and contour maps showing top elevation, bottom elevation, and/or thickness of each identified hydrogeologic unit in the given region.

To simplify the input data preparation, the model is structured on the basis of two-dimensional surface elements and well logs at each node. Three-dimensional finite elements, describing geologic stratification, are generated by the model. Development of three-dimensional elements by the model ensures the continuity. Any discontinuity in adjoining three-dimensional elements results mathematically in an insulated, no flux, layer, thereby destroying the continuum.

Salient features of input data structure and steps involved in mathematical representation of the given ground-water system are illustrated below using the ground-water system beneath Long Island, New York.

1. Discontinuities, major breaks in slope or thickness, and fault zones of the hydrogeologic units are some of the controlling parameters affecting the ground-water flow regime. To define the geologic stratification for a given ground-water basin, these controlling parameters are identified by discrete nodes on each of the hydrogeologic units (Figure 4.1) and then combined at the surface (Figure 4.2).
2. Additional nodes are placed along surface-water bodies (lakes, rivers, seashore), at important recharge or pumping wells, and for further subdivision of large uniform region (for defining variable infiltration or pumping stresses).

EXTENT OF GARDINER CLAY AND NODES PLACED TO DEFINE THE EXTENT

MAGOTHY UPPER ELEVATION AND NODES TO DEFINE VARIATIONS

UPPER ELEVATION OF RARITAN CLAY AND NODES PLACED TO DEFINE THE VARIATION

FIGURE 4.1.

Illustration of Proposed Logic for Discretization of Variations Encountered in a Large, Natural, Multilayered, Ground-water Regime

FIGURE 4.2. All Nodes Based on Variation in Aquifer Thickness of Elevation of Different Hydrogeologic Units and Stream Locations

3. At locations where too many nodes are clustered together, a single node is defined.
4. At each surface node, well logs are prepared by using contour maps of hydrogeologic units. The depth increments are defined either in elevation above a given datum or in actual thickness of each material. The interface between materials is described as illustrated in Figure 4.3.
5. According to the geologic and surface-water boundaries, the nodes are joined by straight (linear) or curved (quadratic or cubic) lines to subdivide a given region into isoparametric, two-dimensional, mixed order elements (Figure 4.4). Mixed-order, isoparametric elements enable a mathematical description of straight, curved, or meandering boundaries with a minimum number of nodes.
6. Using two-dimensional surface elements and well log details at each node, three-dimensional elements (Figure 4.5) are generated by the computer program.
7. Some of the stress conditions encountered in a ground-water system are:
 - a. potential boundary conditions, i.e., given water-level elevation of lakes, ponds, seashore, etc.
 - b. vertical infiltration at the top of the ground-water surface
 - c. pumping (or recharge) at given well locations (node)
 - d. pumping (or recharge) for a given subregion (element) from a given layer.

This model has the provision of defining time invariant and variant
i) potential boundary conditions, ii) vertical infiltration rate at the surface and iii) pumping or recharge rate at a given location or element.

WELL LOG AT SURFACE NODE 35

Interface = Top Material # x 100 + Bottom Material #

Node Number = Material # of which Node is Base x 1000 + Surface Node

FIGURE 4.3. Illustrative Example of Describing Interfaces of Two Materials in Well Log (Vertical Section) at a Node Location

FIGURE 4.4. Subdivision of the Given Region Into Two-Dimensional, Mixed-Order, Finite Elements Based on Nodalization of Figure 4.2

FIGURE 4.5. Three-Dimensional Finite Elements Generated from Two-Dimensional Surface Elements and Well Log Details at Each Node

5.0 SUPPORTIVE PROGRAMS DEVELOPED TO VERIFY THE INPUT DATA USED IN SIMULATION

Digital finite element models provide a flexible means of defining a complex ground-water system. However, the probability of input errors always exists in computer simulations. When dealing with a large number of nodes, manual checking becomes tedious and time-consuming, and results are not guaranteed to be free of errors. Some means of editing the input data prior to simulation is essential.

Realizing the above needs as well as the fact that tabular output data are not adequate, the following supportive plotting programs have been developed. The input data deck initially designed for simulation is used in all plotting programs.

5.1 PLOTTING OF NODE LOCATIONS, ELEMENTS, AND VERTICAL LOGS

The subprogram (PLOTTEL) has been developed to plot the locations of each node, two-dimensional surface elements, and the vertical log at each location. Finite element solution of the governing partial differential is obtained by Gaussian integration of each element. Any error in description of an element (by defining the wrong node number or sequence of the nodes of an element) results in an erroneous integration. Through this plotting package, the user achieves a positive check on correctness of geometry of nodes and finite elements.

The logs that describe the geologic section on which the three-dimensional elements are generated by the program are another place where errors can occur. To provide a visual check for the modeler and hydrogeologist, geologic sections are drawn at each node location (Figure 5.1). A dashed line is drawn to identify the elevations of sea bed. A horizontal tick mark is placed at the interface of two hydrogeologic units on the section. The material between these tick marks (representing thickness of the hydrogeologic unit) is identified by an integer number assigned to the material. By knowing the local stratigraphy, hydrogeologists can review these

FIGURE 5.1. Illustrative Sample Plot from PLOTTEL

sections for errors. At locations where adequate data are not available, solutions can be obtained by redefining the geologic sections with alternative interpretations. Since three-dimensional elements are developed by the computer program itself, another formulation of the given ground-water system is easily obtained by changing vertical sections at disputed locations.

From the vertical sections and the element demarcations plotted by PLOTTEL, fence diagrams (Figure 4.4) can easily be developed.

5.2 CONTOUR AND THREE-DIMENSIONAL PLOTTING OF INITIAL HEAD, TOP ELEVATION, AND THICKNESS OF MATERIALS

When dealing with large multilayered reservoirs, it becomes difficult to evaluate errors from the tabulated printer output only. Contours and three-dimensional plots of initial head, top elevation, and/or thickness of each hydrogeologic unit provide a useful tool for proper verifications of input and interpretation of the results. Most of the available computer graphics packages are, however, applicable only to regular grid values, while irregular grids usually are involved in finite element methods. To estimate the regular grid values, the reference plane is divided into linear elements. Using the principles of basis functions and search techniques, program GRIDIT finds the element associated with each grid and its local coordinate values. For a given basin, this program is executed only once to generate a binary output disk file, used by GRIDIN and GRIDH. Program GRIDIN estimates grid values of the parameter of interest (top elevation or thickness of any material) from the input data file and prepares a regular grid file that can be contoured or plotted three dimensionally by the normal available graphics packages.

Figure 5.2 is a three-dimensional display of top elevations and thickness of some of the hydrogeologic units defined in the input file. Instead of three-dimensional maps, contour maps (Figure 5.3) or maps with grid value (Figure 5.4) can be generated for each of the parameters. The illustrative examples in Figures 5.2 through 5.4 are for the entire region. For detailed

FIGURE 5.2.

The Output from Program GRIDIN Provides a Means for Three-Dimensional Display of the Input Data or Results

Contour map showing the ELEVATION OF BEDROCK
as interpolated from input data of 3-D finite element
ground water model of Long Island, New York
Contour interval 25.0 meters

FIGURE 5.3. Illustrative Example Checking the Input Data by Use
of Program GRIDIN, Which Uses the Same Input File
as Used for Simulation

Top altitude of RARITAN CLAY in meters as interpolated from input data of 3-D finite element ground water model of Long Island, New York, Datum is mean sea level.

FIGURE 5.4. Illustrative Plot of Grid Values from Program GRIDIN Output Using Input Data of Three-Dimensional Ground-water Model of Long Island, New York

review, each subregion can be plotted separately. These plots provide a good check for input errors, resulting in sharp peaks, clustering of contours, or in discontinuity of particular hydrogeologic units. Because these are plotted from the same input or output files used in simulation, the plots provide a tool for proper interaction between modelers, geologists, hydrologists, and decision makers.

5.3 PLOTTING HELD POTENTIAL AND PRESCRIBED FLUX BOUNDARY CONDITIONS

Ground-water simulation is a boundary value problem, and, therefore, the results are functions of the held potential boundary conditions, as well as the flux stresses prescribed for the surface, nodes, and elements. Program PLOTBC plots the locations and values of held potential boundary conditions. PLFLUX plots the stresses prescribed for ground surface and in each element of each hydrogeologic unit. Because surface infiltration is usually described in terms of fraction of precipitation, the units of input are rate per unit area per unit time, while withdrawal and recharge from each element are in units of volume per unit time for that subregion.

5.4 DISPLAY OF RESULTS

Program GRIDIN was developed to estimate the head values at regular grid spacing for any given plane (at the surface or bottom of a hydrogeologic unit). From the data retrieved by GRIDH, contour maps, grid display, and/or three-dimensional plotting (Figures 7.4 and 7.5) can be accomplished. Program DIFH estimates the difference between a given set of potentials (measured or prescribed) to model predicted potentials.

5.5 ROLE OF THE SUPPORTIVE PROGRAM

During the last decade, major importance has been placed on displaying the capabilities of various digital models. Usually modelers and local hydrogeologists have very limited collaboration, except that modelers obtain the data from hydrogeologists and the results of the model are used by the hydrogeologist with some reservations. If the ground-water simulation has to

be used as a predictive tool, adequate interactions among various agencies are essential. Through these graphics and grid value displays (which can be enlarged for a given subarea), input and output can be examined and reviewed by the various disciplines. It is expected that with close collaboration, the probability of erroneous input can be reduced. The models capability of simulating complex multilayered systems along with these the supportive programs makes this modeling system particularly effective for field applications.

6.0 VERIFICATION AND SENSITIVITY ANALYSIS WITH ANALYTICAL SOLUTIONS

Since three-dimensional analytical solutions are not available for checking the algorithm used in the three-dimensional, finite element ground-water model, the verification of the three-dimensional model was accomplished using two-dimensional or quasi three-dimensional analytical solutions.

6.1 VERIFICATION WITH THE THEIS NONEQUILIBRIUM EQUATIONS

The Theis nonequilibrium equation [Equation (6.1)] defines unsteady radial flow to a well pumping at constant rate in an infinite, homogeneous and isotropic aquifer. It is a two-dimensional solution and the three-dimensional model can handle two-dimensional cases by prescribing identical boundary conditions for the top and bottom layers.

$$h_o - h = Q/4\pi T \int_u^{\infty} \frac{e^{-u} du}{u} \quad (6.1)$$

$$u = r^2 S/4Tt$$

where

$h_o - h$ = drawdown, L

Q = well discharge, L^3T^{-1}

T = transmissivity, L^2T^{-1}

r = distance from pumping well to observation point, L

S = storage coefficient

t = pumping time, T

Because of the symmetry of radial flow to a single pumping well, a pie-shaped wedge was modeled with finite elements. A 22.5° angle, representing 1/16 of the influenced area, was selected. By necessity, the aquifer is required to be bounded at a large distance from the well in a finite element program. This was achieved by prescribing exponentially increasing distances from the well (Figure 6.1).

FIGURE 6.1. Diagrammatic Representation of Exponentially Increasing Nodal Arrangement Used for Finite Element Solution

Numerical test cases were run for sensitivity analysis using exponentially increasing time increments (Table 6.2). Figure 6.2 compares the finite element solutions with the analytical Theis equation. All the cases gave results comparable to the analytical solution.

6.2 VERIFICATION WITH TWO-DIMENSIONAL ANALYTICAL SOLUTION, PATHS

The analytical solution PATHS,⁽¹⁵⁾ Equation (6.2), defines potential distribution in an aquifer system (Figure 6.3) having initial uniform gradient, large remote radial boundary, and caverns (cylindrical reservoirs) with a given head above the water table.

$$\phi = H_0 - U_0 x + \frac{U_0 (r_0)^2}{(x^2 + y^2)} - \frac{H_0}{\ln \left(\frac{R}{r_0} \right)} \ln \left(\frac{\sqrt{x^2 + y^2}}{r_0} \right) \quad (6.2)$$

where:

	<u>Units</u>
ϕ = ground-water potential	L
U_0 = head in the reservoir with center at x = 0, y = 0	L
H_0 = initial uniform gradient of the lateral flow	L/L
r_0 = radius of the cylindrical reservoir or cavern	L
R = large radial distance to the remote outer boundary	L

Equation (6.2) is written without any source/sink terms; however, serpen and cyclic pumping or injection wells can be simulated as shown in Figure 6.3.

Test Case

A single rectangular aquifer (30,000 ft x 24,000 ft) with a cavern (reservoir) of 2000-ft radius located 21,000 ft from the lower left-hand corner of the region was selected as a test case. Because of the symmetry

TABLE 6.1. Variable Time Steps Used

Step Number	Elapsed Time (Days)	T (Days)
1	0.1000E-04	0.1000E-04
2	0.5000E-04	0.4000E-04
3	0.2100E-03	0.1600E-03
4	0.8500E-03	0.6400E-03
5	0.3410E-02	0.2560E-02
6	0.1365E-01	0.1024E-01
7	0.5461E-01	0.4096E-01
8	0.2184	0.1638
9	0.8738	0.6554
10	3.495	2.621
11	13.98	10.48
12	55.92	41.94
13	223.7	167.8
14	894.8	671.1
15	3579.	2684.
16	0.1432E+05	0.1074E+05
17	0.5727E+05	0.4295E+05
18	0.2291E+06	0.1718E+06
19	0.9160E+06	0.6869E+06
20	0.3665E+07	0.2749E+07
21	0.1466E+08	0.1100E+08
22	0.5864E+08	0.4398E+08
23	0.2346E+09	0.1760E+09
24	0.9382E+09	0.7036E+09

FIGURE 6.2. Analytical and Finite Element Solutions of Their Equations Drawdown Versus Time

FIGURE 6.3. Example PATHS Model Region⁽¹⁶⁾

of the region, only half the area was simulated. It was assumed that the regional water table had a uniform gradient of 0.001, and the head in the cavern was 25 ft above the water table at the center of the cavern.

An example PATHS model region is shown in Figure 6.3. A three-dimensional display of the steady-state analytical (PATHS) solution over the complete region for the single aquifer system is shown in Figure 6.4. Because of the limited size of the test case region, the additional driving head in the cavern affects the flow pattern (Figure 6.4) along the boundaries (except the line of symmetry). Therefore, the three external boundaries have to be considered as held potential boundaries in the finite element solution.

For verification and sensitivity analysis, the effects of grid density, element orientation to flow pattern, and order of elements were studied. The maximum difference (actual difference divided by the imposed head in the cavern) between the analytical and numerical solution for each test case was compared.

Grid Density

Sensitivity analysis of grid density was conducted using combinations of square and radial elements. With the cavern (major source) being circular, radial elements were selected near the cavern and square elements in the remaining portion of the region. Test cases of the combined square and radial pattern consisted of 6000, 3000, and 1500-ft square elements coupled with radial linear elements with two node spacings: 1) constant (Figure 6.5), and 2) variable (progressively increasing by a factor of 2) (Figure 6.7). The intention of variable spacing was to increase nodal density in areas of steepest gradient.

FIGURE 6.4. Three-Dimensional Representation of the PATHS Steady-State Analytical Solution

The finite element results compared closely with the analytical solution (Figure 6.6) even for the coarse grids (Figure 6.5a and 6.7a). For all test cases, progressive spacing of radial nodes near the cavern performed better than the equidistant spacing. The location and magnitude of the maximum difference between analytical and finite element solutions are indicated on the figures of the test cases.

Since the maximum difference was in the region of steepest gradient (i.e., around the cavern), additional studies were conducted by increasing the number of radial nodes/elements (variable spacing) for the coarse grid pattern (i.e., 6000 ft). Test cases with 2, 3 and 5 concentric rings are shown in Figures 6.7, 6.8a, and 6.8b, respectively. The maximum difference decreased with increasing density; however, the amount of improvement rapidly decreased and became almost negligible for 5 rings, as compared to 3 rings.

Element Orientation

Element orientation studies consisted of a radial element pattern over the majority of the region and square linear elements (6000-ft node spacing) elsewhere. Two cases consisting of equidistant (Figure 6.9a) and progressively increasing (Figure 6.9b) spacings were studied. The progressive spacing case reduced the maximum difference by 1/3 over the equidistant spacing case. For an equal number of nodes, the radial pattern (Figure 6.9) had lower maximum difference than the combined square and radial grid pattern (Figure 6.7).

Element Order

Using the identical nodal pattern of Figure 6.9b, the region was simulated with mixed-order elements (Figure 6.10) involving quadratic and cubic sides. This reduced the number of elements for a given number of nodes. The mixed order element did not improve the results. A summary of the results for all test cases is shown in Table 6.2.

Conclusions

Three basic conclusions can be drawn from this verification and sensitivity analysis:

FIGURE 6.5. Combined Square (a) 6000 ft, (b) 3000 ft, and (c) 1500 ft and Constant Radial Pattern

FIGURE 6.6. Comparison of Analytical Solution (PATHS) and 3-D Finite Element Solution for Grid Pattern Shown in Figure 6.5(c)

FIGURE 6.7. Combined Square (a) 6000 ft, (c) 3000 ft, and (c) 1500 ft and Variable (2 layers) Radial Pattern

FIGURE 6.8. Combined Square (6000 ft) and Radial Element Pattern with (a) 3 and (b) 5 Layers of Nodes Around the Cavern

FIGURE 6.9. Radial Element Pattern with (a) Constant and (b) Variable Node Spacing

FIGURE 6.10. Radial Element Pattern with Mixed Order Elements

1. The three-dimensional, finite element ground-water model results compare closely with the PATHS analytical solution.
2. Increasing grid/element density appears to be the most influential factor in reducing the differences between the numerical and analytical solutions.
3. Primary differences between numerical and analytical solutions occur in areas of highest potential gradient, and therefore, increasing node/element density in these regions has the greatest impact on reducing differences.

TABLE 6.2. Summary of Results

Element Pattern (Fig.)	Number of Nodes	Number of Elements	Number of Concentric Rings of Elements Around Cavern	Maximum Difference/Change in Head at Cavern (ft/25 ft)	Comments
6.5 (a)	38	25	2	0.025	Equidistant spacing
6.5 (b)	113	88	2	0.019	Equidistant spacing
6.5 (c)	409	328	2	0.013	Equidistant spacing
6.7 (a)	38	25	2	0.023	2 ^N spacing
6.7 (b)	113	88	2	0.019	2 ^N spacing
6.7 (c)	409	328	2	0.009	2 ^N spacing
6.8 (a)	42	28	3	0.020	2 ^N spacing, increase cavern nodes
6.8 (b)	50	34	5	0.019	2 ^N spacing, increase cavern nodes
6.9 (a)	82	63	5	0.036	Equidistant spacing
6.9 (b)	82	63	5	0.013	2 ^N spacing
6.10	82	29	5	0.014	Mixed order elements

6.3 VERIFICATION WITH HANTUSH SOLUTION FOR NONSTEADY FLOW TO FLOWING WELLS IN LEAKY AQUIFERS

Hantush(4,17) developed an analytical solution to determine the drawdown distribution in the vicinity of a steady well draining an elastic artesian aquifer confined by semipervious elastic strata. Well discharge is supplied by reduction of storage in the aquifer and by leakage from the semipervious layers. The leakage is obtained from reduction of storage in the semipervious elastic beds or from other bodies of water over and/or underlying the semipervious beds or from both sources. The hydraulic conductivities in semiconfining layers have to be very small compared with those in pumped aquifers, so that flow is vertical through the semiconfining beds and horizontal in the main aquifer. In other words, the completely refracted leakages are considered to be uniformly distributed throughout the horizontal flow in the main aquifer (Figure 6.11). Using exponentially increasing nodal distances from the well location (Figure 6.1), and exponentially increasing time from the initial time (Table 6.2), a finite element simulation of the Hantush problem was performed. Figure 6.12 compares dimensionless drawdown and dimensionless time at radius 1.0 m, 2.718 m, and 7.389 m.

FIGURE 6.11. Schematic Representation of Simulated Leaky Aquifer

FIGURE 6.12. Comparison of Dimensionless Drawdown Versus Dimensionless Time of Analytical and Finite-Element Solution in a Leaky Aquifer

7.0 SAMPLE ILLUSTRATIVE PROBLEMS

In Sections 4.0 and 5.0, application of the model to the ground-water system beneath Long Island, New York, was illustrated. The Long Island ground-water basin is one of the most intensively monitored systems. Interpreted contour maps showing thickness and top elevations of major hydrogeologic units were available to follow the logic described in those sections.

For simulation of some hypothetical site, or for a given ground-water system with an adequate number of well log data penetrating all the aquifers, the input data for the system can be prepared using the well locations as nodes. Two such examples are illustrated below.

7.1 SUTTER BASIN, CALIFORNIA

Curtin,⁽¹⁸⁾ on the basis of electric logs from more than 400 wells, infers that fresh water which is recharging at Sutter Buttes rises through the Sutter Basin fault (Figure 7.1 and 7.2), creating a salt-water mound. Using 36 well logs, the finite element grid (Figure 7.3) was created⁽¹⁹⁾ that described the multilayered system. By changing the material properties and the layering in fault zone, the sensitivity of three-dimensional modeling was tested. To illustrate the effects, the bottom of the post-Eocene nonmarine deposit and top of Kione formation are taken as reference planes. The results of the following three trials are discussed briefly.

Trial 1: no layering (all the elements are assumed to be composed of post-Eocene nonmarine deposits)

Trial 2: layered system as shown in Figure 7.2

Trial 3: same as Trial 2, but vertical fault (below element 22) considered as composed of post-Eocene nonmarine deposits (i.e., no layering in fault zone)

Figure 7.4a illustrates the results of no layering (Trial 1) similar to the results obtained from two-dimensional simulation. Figure 7.4b illustrates

FIGURE 7.1. Location of the Vertical Fault and Water Table Contours (in meters) in the Sutter Basin Study Area. Geologic Section A-A' is given in Figure 7.2. (8)

FIGURE 7.2. Geologic Formations and Vertical Faulting Along Geologic Section A-A' (Figure 7.1) in the Study Area⁽⁸⁾

FIGURE 7.3. Three-Dimensional, Finite Element Formulation of Sutter Basin, California. Strata details are based on interpretation of oil and gas well logs.⁽⁸⁾

FIGURE 7.4. Estimated Hydraulic Heads (h) at the Base of the Post-Eocene Nonmarine Deposits. Figures a, b and c are computed results for Trials 1, 2 and 3 (no layering, layering including a vertical fault, and layering not including a vertical fault), respectively.⁽⁸⁾

the results of Trial 2. A comparison between Figures 7.4a and 7.4b reveals the results as expected because of layering. In Figure 7.4a, the high mounds are as expected, since the head is a function of the elevation of the bottom of the post-Eocene deposits. But, with the introduction of the semiconfining layers of Capay Shale, Sacramento Shale, and Winters Sand and Shale in Trial 2, head distribution completely changes. The effect of the fault zone is depicted by increased head values in its vicinity.

Figure 7.4c shows the head values estimated at the bottom of post-Eocene nonmarine deposits under the same conditions as those in Figure 7.4b, but the vertical fault zone is considered to be composed of alluvial fill without layering. The effect of this minor change can be seen by observing the gradual dissipation of the head around the mound on the fault zone.

The top of the Kione formation was considered as another reference plane. The Kione formation is found at each location in the basin and is the prime formation responsible for the transport of the connate water. Figure 7.5a depicts the estimated head without layering, Figure 7.5b the estimated head of Trial 2 (with layers), and Figure 7.5c the head at the top of the Kione formation without layering in the vertical fault zone. Also clearly shown is the effect of the gradual widening of the trough around the fault zone removal of the layering only in the element 22 series (which is a very small fraction of the whole basin).

7.2 HYPOTHETICAL WASTE REPOSITORY IN A REPRESENTATIVE SALT FORMATION

To examine the consequences of the accidental release of radionuclides from a hypothetical nuclear waste repository located in a generic salt formation, a three-dimensional, finite element, ground-water model was used to define hydrologic flow details. A hypothetical stratigraphy above the salt zone (Figure 7.6) was assumed. The strata overlying and surrounding the salt contain water, but are not considered prolific aquifers. At depth, in the limestone/sandstone aquifer, the water is considered to be a brine and, therefore assumed not to dissolve the salt. Hydraulic gradients of 1 m/1 km were assumed for all aquifers systems. The average permeabilities, range

FIGURE 7.5. Estimated Hydraulic Heads (h) at the Top of the Kione Formation, Which Contains and Conveys Connate Water. Figures a, b, and c are the computed results for trials 1, 2, and 3, respectively.(8)

FIGURE 7.6. Representative Generic Stratigraphic Section for Salt

TABLE 7.1. Hydrogeologic Parameters for Generic Salt Stratigraphy

<u>Rock Type</u>	<u>Hydraulic Conductivity, (cm/sec)</u>	<u>Range (cm/sec)</u>	<u>Porosity %</u>	<u>Unit Weight, (kg/m³)</u>
Unconsolidated sand, gravel, and silt	6.8×10^{-3}	$1 \times 10^{-7} - 1 \times 10^{-2}$	20	1,700
Calcareous shale partly sandy	1.2×10^{-5}	$1 \times 10^{-10} - 1 \times 10^{-3}$	13	2,300
Sandstone	8.1×10^{-5}	$1 \times 10^{-10} - 1 \times 10^{-2}$	20	2,400
Dolomite	8.1×10^{-5}	$1 \times 10^{-10} - 1 \times 10^{-2}$	20	2,750
Cherty limestone	5.8×10^{-5}	$5 \times 10^{-8} - 1 \times 10^{-3}$	20	2,600
Dolomite, some anhydrite	5.8×10^{-5}	$5 \times 10^{-8} - 1 \times 10^{-3}$	20	2,800
Interbedded shale and dolomite	6.8×10^{-6}	$1 \times 10^{-10} - 1 \times 10^{-3}$	20	2,500
Salt	2×10^{-18}	$2 \times 10^{-21} - 1 \times 10^{-8}$	0.5	2,160

of permeabilities, and average porosities for each of the materials are listed in Table 7.1. The regional discharge site for the water within the various layers is assumed to be a river 5 km from the center of the repository.

Figure 7.7 shows the plan of the model region. Since simulation of the X-Z zone (Figure 7.8) was attempted, only single layers of the three-dimensional elements were used. Figure 7.9 is the three-dimensional view of the hydrologic model; Table 7.2 lists hydrogeologic input parameters; and Figure 7.10 shows the contour plot of the model-predicted vertical ground-water potentials with superimposed streamlines.

FIGURE 7.7. Plan View of the Hypothetical Nuclear Waste Repository Model Region

FIGURE 7.8. Cross Section of the Five Geologic Layers Used to Define the Hypothetical Nuclear Waste Repository Model Region

FIGURE 7.9. Three-Dimensional View of the Hypothetical Nuclear Waste Repository Model Region

TABLE 7.2. Finite Element Model Hydrogeologic
Input Parameters Used in Hypothetical
Nuclear Waste Repository

<u>Layer</u>	<u>Rock Type</u>	<u>Thickness, m</u>	<u>Hydraulic Conductivity, m/day</u>	<u>Porosity, %</u>
1	Unconsolidated sand, gravel and silt	10	5.9	20
2	Calcareous shale, partly sandy	30	0.01	13
3	Sandstone, dolomite, cherty limestone	160	0.06	20
4	Interbedded shale and dolomite (salt caprock)	50	0.006	20
4a	Fractured interbedded shale and dolomite (salt caprock)	50	0.06	20
5	Salt	Infinite	0.00006	0.5
5a	Fractured salt	450	0.06	20

FIGURE 7.10. Contour Plot of the Hypothetical Nuclear Waste Repository Model Predicted Vertical Ground-Water Potentials with Superimposed Streamlines

8.0 USER'S MANUAL FOR THREE-DIMENSIONAL, FINITE ELEMENT, GROUND-WATER MODEL

8.1 BRIEF DESCRIPTION OF PROVISIONS AND LOGIC USED IN DESCRIBING THE PHYSICAL SYSTEM

This program has been developed for three-dimensional simulation of large multilayered ground-water basins on the PDP-11/45 computer. The main objectives and provisions of the model are to simulate multilayered ground-water systems. The prime use of the model will be to anticipate the response of the flow system to natural recharge from rain, and various expected and proposed stresses of pumping and/or recharge through wells and streams.

Attempts have been made to accommodate various conditions encountered in the field. The input data are simplified for use by engineers and geologists without specific knowledge of finite element procedures. In general, the field data required are as follows:

Properties of Geologic Formations

Various materials encountered in the basin are required to be identified numerically. The values of the permeability of each material in the x, y, and z directions can be read directly as hydraulic conductivities or in terms of intrinsic values. In the latter case, the user has two options: 1) estimate temperature as a function of depth followed by viscosity calculations as a function of temperature or 2) consider the temperature and salt concentration effects on viscosity. Since this is a water-flow model, concentration as a function of changes in the ground-water pattern is not updated.

Location of Nodes

Node location is based on the decision of the field hydrologist conversant with the ground-water system. In general, all of those major features which affect the ground-water flow of a region have to be accounted for by a node location. In brief, some of the parameters which must be defined are:

1. demarcation of the project boundary and surface-water bodies, such as seashore, lakes, ponds, streams
2. sharp changes in elevation of top, bottom, or thickness of various geologic formations
3. selected well sites penetrating all geologic formations; major pumping or recharge wells which are used as a prescribed flux boundary condition (B.C.). (If there is a cluster of these wells in some area, demarcation of each well as a node may unnecessarily increase the cost and in such situations, these wells can be considered as a source or sink within a given element.)
4. approximate dividing line of ground-water flow.
5. other physical features affecting the ground-water levels of the region.

c) Well Logs

Vertical log details at each specified node at the surface include x and y coordinates, total system thickness at this location, depth to water table, number of layers, depth or elevation of each interface between two geologic materials, and identification of each interface by two numbers (e.g., 0205 stands for material number 2 on top and 5 on bottom). The well data themselves are used for nodes located at well sites. For other nodes, such information has to be obtained or interpolated from the contour maps of thickness or elevation of each formation.

d) Subdivision of the Basin in Mixed-Order, Two-Dimensional Finite Elements

It is very difficult to subdivide large multilayered basins into three-dimensional elements. If proper continuity is not maintained between two adjacent element sides, mathematically the solution responds as if there is insulation between the two elements. To eliminate such computing errors, three-dimensional elements are generated by PROG1 based on the details of the various layers defined at each node.

The entire region is divided into two-dimensional, mixed-order isoparametric elements. Generation of a two-dimensional mesh within a simple

ground-water basin can be accomplished easily through a computer program. However, this program is basically written for handling large multilayered basins which usually have complex layering and geometry. As a result of this complexity, the decision as to how to divide the basin into two-dimensional, mixed-order elements is left with the user.

8.2 DIMENSION LIMITS OF PRESENT MODEL

At present, the program is set for following maximum dimensions.

total number of surface nodes	768
total number of nodes in the system	2560
maximum number of layers in a well log	20
maximum number assigned to the materials	99
maximum number of potential B.C.	768
maximum number of nodes located on streams	128
maximum number of nodes with prescribed flow boundary conditions	128
maximum number of non-zero nodes in an element	20
maximum number of unknown nodes	2048
maximum bandwidth of non-zero elements in decomposition to upper triangle	128
maximum number of surface elements	768
maximum total number of elements	2000
maximum number of time steps	70

The above dimensions were used to meet the need for the simulation of multilayered ground-water reservoir of Long Island, New York. If required, the above maximum limits can be increased by changing the allocation of space on the File-Q disk.

8.3 DATA PREPARATION FOR EACH SUBPROGRAM

Preparing input data for a program that has been developed by others can be very difficult. To simplify this process, the input data requirements are shown by a simple flow chart which gives the sequential manner in which

different variables are to be read. For each read statement, the prescribed format is defined and a brief description of each variable is given in the following text. Typical examples, along with the complete listing of the input data, are given. It is recommended that these simple examples be used to become familiar with the data preparation process.

PROG1 - Most of the data for a given system are input by this program. The flow chart (Figure 8.1) illustrates the preparation of the data and Table 8.1 gives a description of the variables. The print options (0, 1, 2, 3, 4, 5) are entered through a remote terminal.

PROG2 - There are no input requirements, except defining the disk unit and print options. For simulation of new problem use NPRINT = 1 and check the area of each element. If it is negative there is error in defining nodes in anticlockwise direction.

BAND - There are no input requirements, except defining the disk unit and print option.

PROG3I - The required details of the input data are shown by the flow chart in Figure 8.2. Table 8.2 gives a description of the variables.

PROG3 - The input data for this program are read in PROG3I through a remote terminal. The following input data are required:

NPRINT = 0 - generates no line printer output, but writes on the disk,
= 1 - prints the head value of each node at the end of the time step,

NBEGIN = 1 - for normal start,
= 2 - if element matrices were formed earlier and computer crashed in SYSMAT,
= 3 - to start program from MREAD. Usually UPPERT takes considerable computer time for decomposition of the system of linear equations to upper triangle form and if the computer crashes, the user will have to start with MREAD.

- = 4 - if UPPERT was stopped by switch "O" in "ON" position, program asks for one more parameter, NREDO, which should be 1 if UPPERT was stopped by switch "O",
 - = 5 - for additional solution with a different stress case after UPPERT has been fully executed,
 - = 6 - for starting PROG3 in BACKSUB. This is not usually needed except on abrupt crash in BACKSUB of PROG3.
 - = 7 - for printing the results of simulation,
- ITT = starting time step. Usually it is equal to 1. In case of a transient run, this option can be used to restart the PROG3 at other time steps.

Figure 8.3 is the sample illustrative copy of the commands operated from the terminal.

FIGURE 8.1.1. (Contd)

FIGURE 8.1. (Contd)

FIGURE 8.2. Flow Chart Showing Input Data for PROG31

TABLE 8.1. Explanation of Some Symbols Used in
Flow Chart of PROG1

- TITLE = Title of the project in 72 characters.
- NPRINT = flag for printing various levels of output.
- = 0 - generates only disks and no lineprinter output.
 - = 1 - writes only the details regarding total number of nodes, element, unknown, etc. required for checking the dimensions of the parameters.
 - = 2 - prints the coordinates of surface nodes in addition to the above.
 - = 3 - prints all the nodes generated by the model for the given system and details regarding surface elements.
 - = 4 - prints the same output as "2" but in addition, a copy of the cards is also printed to trace the errors in input data.
 - = 5 - generates output of all the nodes and all the elements.
- NZELEV = flag for identifying the depth elevation data of each layer.
- = 0 - for depth data for each layer in each vertical section.
 - = 1 - for elevation of each layer in each vertical section.
- KTYPE = flag for identifying K values.
- = 0 - if hydraulic conductivity data are read.
 - = 1 - if to be estimated from temperature-depth relationship, then K is intrinsic permeability values.
 - = 2 - if concentration data at each depth are to be read for K estimation. To use this option, one printout from NPRINT = 3 and KTYPE = 1 may be made for node identification.
- PERCON = factor for converting K values.
- XYCON = factor for converting x and y coordinates.
- ZCONV = factor for converting z data.
- HCONV = factor for converting head data.
- NSEA = flag for sea boundary conditions.
- = 0 - normal.
 - = 1 - for accounting for sea-water density.

NOTE: Sometime same variable is used for different parameters. Therefore, the list of variables is given in the same order as they are listed in Figure 8.1.

TABLE 8.1. (contd)

NSEC = flag for termination of one type of data.

I = material number assigned.

AMATER = 72 character description of material.

D1,D2,D3 = are K-X, K-Y, K-Z, OF MAT #I.

THETA = porosity of Mat #I.

BETAP = compressibility of liquid.

PZERO = reference pressure at which THETA is measured of Mat #I.

ALPHA = compressibility of the medium of Mat #I.

SPEC = specific yield of Mat #I.

TSURF = surface temperature

TEACHZ = increase in temperature each unit depth.

RRHO = aquifer fluid density with 0 concentration.

TRHO = reference temperature of RRHO

TCOEF = coefficient of expansion per unit change of temperature

TVISC = viscosity of aquifer fluid at TRHO

CONS = constant to estimate hydraulic conductivity in given unit
= gravity term (inappropriate unit)/FACT

FACT = factor for converting centipoise to other units

= 3.6 to kg/mhr

= 6.72×10^{-4} to lb/ft sec

= 2.419 to lb/ft hr

NVISC = number of points describing concentration versus viscosity
relationship

BR = regression coefficient of viscosity and temperature relationship

TTEMP = temperature at which concentration viscosity relationship is given

CON = concentration (ppm)

VISC = viscosity (cp)

K = node number (less than 1000) of surface.

XX = x coordinate.

YY = y coordinate.

WTT = water table elevation at surface. Water table elevations of nodes
below the surface node are set equal to this. This can be modified
by reading the actual measured values as described later.

TABLE 8.1. (contd)

BED = depth to bottom of ground-water regime.
NBB = 4, if corner node of the element.
5, if midnode of the element.
This parameter will be used for plotting purposes in the future.
For now it can be left blank.
LMAX = maximum number of layer interfaces (layer + 1) at this given node.
LL = identification of interface of various layers - two integers for each material, e.g., "0205" means upper material at the depth is #2, while #5 is the lower material.
ZZ = elevation of each interface if NZELEV = 1.
= depth increment, if NZELEV = 0.
NSEC = 1, if piezometric head is to be read.
HEAD = initial condition.
CONCEN* = salt concentration in PPM of all nodes.
NODB = node # having prescribed head boundary condition.
BIV = value of B.C. at this node.
NFF = number of points in the stream.
STREAM = four character name of stream.
NF = node numbers on the given stream.
ELEV = elevation at given NF.
NODBF = node number having flux B.C.
BIVF = prescribed flux $M^3 T^{-1}$.
K = element number at surface 1000.
N1 = element order, if sides are of same order (1, 2, and 3, respectively for linear, quadratic or cubic).
= 0 - if the element is mixed order.

* Note: For preparing these data, first get output with NPRINT = 3, KTYPE = 1, and NSEC = 1 after coordinate data cards. This will print all the nodes generated. As these data may need too many cards, the option for format is left to the user.

TABLE 8.1. (contd)

NN = total number of nodes including zero following the code given below.

NOD = node number.

MIXNOD = order of each corner, if N1 = 0 and defined by three integers: first (hundredth position) denotes ζ side of element, second (tenth position) is for η side, and third (unit position) for ζ vertical (at present this should be equal to 1) as shown in the following example.

MIXNOD for corner 1, 2, 3, and 4 are 221, 211, 311, and 321, respectively.

NN = 12.

(NOD(I), I = 1, MN) are 1, 2, 3, 4, 5, 0, 7, 0, 0, 11, 12.

Note: The above illustration is for location, while data for NOD actual node numbers read may be as given below.

TABLE 8.1. (contd)

35, 37, 28, 25, 36, 0, 30, 0, 0, 0, 26, 27.

TABLE 8.2. Brief Explanation of Symbols Used in Flow Chart
for Input Data of PROG3I

DETAIL = 80 character description of the trial run.

NPRINT = Flag for controlling various levels of output.
= 0 omits most of the print statement.
= 1 prints the minimum required information; normally this value should be used.

NSTEDY = 0 for steady state solution.
= 1 for transient solution.

ITOTAL = Number of times to which a transient solution is required to be carried out.
= 0 for steady state.

NQWCP = 1 for no change in sources or sinks of each element, then read in PROG1.
= 2 for reading constant source or sink.
= 3 for reading new source or sink for each time step.

NQ = 1 if no surface flux is prescribed.
= 2 constant surface flux for all time step or steady state.
= 3 variable flux for every time step.

MNTRY = parameter provided for simulation of whole basin as homogeneous material.
= 0 for normal simulation using hydraulic properties of the assigned material to each given element.
= 0 for simulation of whole basin as homogeneous material. The program overrides previously assigned material numbers of each element by the value of MNTRY. This is a useful numerical study if normal simulation with highly confining layer causes unreasonable solution.

NHBOND = 0 for constant potential boundary conditions.
= 1 for time variant potential boundary condition.

NTEXP = 0 for reading total elapsed time for each transient.
= 1 for geometricly increasing elapsed times. Only initial time and FACT are read.

TABLE 8.2. (contd)

FILENAME = Name of Binary File.

Record 1 = Number of points (NBTT) and conversion factor (conv)

Record 2 = Potential Boundary Condition for time Step 1.

Record TTOTAL +1 = Potential Boundary Conditions for time step
TTOTAL.

FACT = Geometric time increment, i.e., $\text{DELT}(I) = \text{DELT}(I-1) * \text{FACT}$

TT = Total Elapsed Time from Start.

CONVER = Convergence factor.

SUMQ = Total volume of source or sink for each element.

Q = Prescribed flux per unit area for each surface element.

```

MCR>RUN DP2:PROG1%
  ENTER FILE-Q DISK UNIT
2
  ENTER INPUT FILE NAME
DP2:THEIS.OUT
  NPRINT(0-5),NFINCH(0=FINCHED ELEMENTS,1=NO FINCHES,
1,0
  IS RUN NEW PROBLEM(=0)OR INPUT MODIFICATION(=1)
0
PROG1  --  STOP

MCR>RUN DP2:PROG2%
  ENTER FILE-Q DISK UNIT
2
  ENTER THE PRINT OPTION (0,1,2,3,4,5)
0
PROG2  --  STOP

MCR>RUN DP2:RAND%
  ENTER FILE-Q DISK UNIT
2
  ENTER THE PRINT OPTION (0,1,2)
0
RAND  ---  STOP

MCR>RUN DP2:PROG3I%
  ENTER FILE-Q DISK UNIT
2
  ENTER INPUT FILE NAME
DP2:THEIS.P3I
  ENTER THE PRINT OPTION (0,1,2)
0
  IF YOU WISH TO CHANGE THE TITLE TYPE "1" OTHERWISE 0
0
  NSTEDY      1 ITOTAL      40 NNGCP      1 NG      1 N-TRY      0 N-COND
    0 NTEXP      0
PROG3I  --  STOP

MCR>RUN DP2:PROG3%
  ENTER FILE-Q DISK UNIT
2
  NPRINT(0,1,2),NBEGIN(1,2,3,4,5,6,7),IT,NBID,NIDNB
0,1,1,4,0

  NBEGIN      1 NPRINT      0 IT      1

PROG3  --  STOP

```

FIGURE 8.3. Sample of the Remote Terminal Input
Required to Run the FE3DGW

8.4 USER'S MANUAL FOR "PLOTTEL"

This program plots node location, two-dimensional surface finite elements, and the vertical logs using the input file used for simulation. Prior to making simulation runs, it is desirable to check the node location, element description, and vertical logs.

This program has been structured such that it serves as a tool to test the input data at various stages. For simulation of a new ground-water reservoir, the user may like to check only the node location first. By assigning 0 to switches NELEMP and LOGS, only the node location is plotted. This plotting may be useful to check the coordinates of each node and to define a given region with two-dimensional, mixed-order surface elements.

At the second stage of checking, the user may set switch NELEMP = 1 to plot the elements. Invariably it may take two or three runs with this option to remove the errors in element delineation. For the convenience of the user, the node numbers and the element numbers are plotted on the map. To distinguish between these two numbers, element numbers are slightly larger and are assigned a different color.

Once the elements are defined correctly the switch LOGS is set equal to 1 to plot the logs. For surface below mean sea level, the vertical log is a broken line. Each interface of the materials is distinguished by a horizontal tick mark along with material identification numbers for each hydrogeologic unit.

Input data for PLOTTEL are as follows:

NHEAD = flag for plotting head values of a given material
(0 = No, 1 = Yes)

NPRINT = Print option (= 0 normal, = 1 for debugging)

If NHEAD = 1 the following additional information (FILE, NMAT, NTOP) is required. Because the model is three-dimensional, user has to choose a given material number whose associated HEAD values (top or bottom) are plotted at appropriate location.

FILE = Name of file created using program "HCOPY" for the given time step.

NMAT = Material number (if user desires to plot top or bottom head values on the given basin, this is specified as "0")

NTOP = flag to specify associated head values with top (= 0) or bottom (= 1) for the given layer.

"PLOT CONTROL FILE" is another file which user specifies on the terminal. A sample of this control file is given below. This control file is structured to read some dummy lines (for user's I.D. variables followed by the list of variables.

	<u>Line Number</u>
INPUT FILE USED FOR SIMULATION OF FLOW	1
HAIN3.OUT	2
NDEX,NELEMP,NDEL,SZNOD,SZELE,KELEM	3
0,1,0,.1,.1,0	4
FACTOR,XXMIN,YYMIN	5
.001,15000.,0.	6
LOGS,MSL,ISO,FACTZ,ZMINUS,ANGLE(RADIAN),YINCREASE,LAYER#	7
1,0,1,1,.0004,0.,1.04,1.0,0	8
NPEN1,NPEN2,NPEN3	9
1,1,1	10
LSUBD,NSEGM,NTOTAL,NOLOGS	11
1,2,10,1	12
4,6,8,21,25,28,39,40,41,42,	13
200,289,1	14
400,403,1	15
NSEGMENTS,NO LOG NODES	16
1,0	17
230,279,1	18

Line #2 is the name of file prepared for simulation and explanation of variables for other lines are as follows:

Line #4

NDEX = flag for plotting node numbers (0 = no, 1 = yes)

NELEMP = flag for plotting element (0 = no, 1 = yes)

NDEL = flag for plotting element numbers (0 = no, 1 = yes)

SZNOD = letter size for node numbers

SZELE = letter size for element number

KELEM = 0 (normal) or 1 (if material properties associated with element. Option used with unconfined version)

Line #6

FACTOR = factor to convert X- and Y-coordinates to the plotting scale.
This factor is an additional scale factor beyond XYCONV specified in the simulation file

XXMIN = Minimum X-coordinates of the region to be plotted

YYMIN = Minimum Y-coordinates of the region to be plotted

Line #8

LOGS = flag for plotting logs (0 = no, 1 = yes)

All the remaining data of line #8 needed only if LOGS = 1

MSL = plotting with reference to mean sea level (0 = yes, 1 = no)

If 1 is used the logs are plotted considering "0" elevation at each node

ISO = flag for isometric plot (0 = no, 1 = yes). Useful mainly for rectangular or square grid

ISQU = 0 (normal) and 1 (if elements are square). In square elements the Y coordinates are distorted further by factor 'YINCREASE'

FACTZ = conversion factor for vertical coordinates in addition to "ZCONV"

ZMINUS = values to be subtracted from Z elevation for plotting logs

ANGLE = needed only for isometric view. The angle is specified in radians

YINCREASE = factor by which Y-coordinates are further distorted to avoid overlapping of the vertical logs

LAYER# = 1 (for plotting material numbers at logs) or 0 (for omission)

Line #10

NPEN1,NPEN2,NPEN3 are pen numbers for plotting. Values between 1-3 can be specified for each.

Line #12

Variables in this line and below provide option for plotting whole region, or part of the region

LSUBD = 0 for plotting whole region

= 1 for plotting the nodes specified

= -1 for excluding the given list of nodes

NSEGM = number of sets (segments) of data read in to define the nodes to be included or excluded. If LSUBD = 0 or all nodes are defined in NTOTAL this is specified as "0"

NTOTAL = number of nodes to be read to define the subregion to be plotted or excluded. This is in addition to nodes generated in series. In above example 10 nodes were desired to be read and all the nodes between 200-289 and 400-403 are generated by two NSEGM

NOLOGS = flag for omission of logs at specified nodes. For stratification mapping in a given subregion the elimination of plotting of logs is achieved by making this switch = 1

Line #13

LIST = NTOTAL nodes. If NTOTAL = 0 this line is omitted.

Line 14 and 15

NSTART,NSTOP,INC = starting and stopping node and the increment

Line 17

This is needed if NOLOGS > 0. First variable defines number of segments (NSEGM) and (NTOTAL) at which no vertical logs are to be plotted.

Line #18

Because NTOTAL for NOLOGS is zero, no node list was read and this line defines that all nodes between 230 and 279 will not have plot of logs.

8.5 USER'S MANUAL FOR PROGRAM GRIDIT

Natural hydrogeologic units encountered in ground-water basins are invariably of irregular shape. Finite-element methods provide a powerful tool for mathematical definition. However, most of the available computer graphic packages are applicable to regular grid values only.

Program GRIDIT is used to associate an element with a given grid point and to find the local coordinates of the grid point with reference to the four nodes of the element. At present, only linear elements are considered when finding the local coordinates. Therefore, the user may plot only node location by using program PLOTTEL with switch NELEMP = 0 and LOGS = 0 and redefine the given basin in linear elements for plotting purposes only. This program is executed only once for a given basin. The input data for this program are defined in a control file with extension of ".GRD". A sample of control file names as HAIN3.GRD is shown below:

	<u>Line Number</u>
INPUT FILE OF PROG1	1
311,7 DOUBLE.OUT	2
1,1,16	3
2,9,15,22,28,29,34,35,41,42,43,44,45,46,47,48	4
200,450,1	5
NFILE,KELEM,FACTOR	6
0.0,.005	7

First Line of the control file is a dummy read. Second line for name of the simulation file used in PROG1.

Line #3

LSUBD = 0 for whole region

= 1 for including only part of the region

= -1 for excluding part of the region

NSEGM = number of segments defined by node series (e.g. line #5). If LSUBD = 0, this is not required.

NTOTAL = number of nodes to define subregion for inclusion (LSUBD = 1) or exclusion (LSUBD = -1)

Line #4

LIST = NTOTAL nodes

Line #5

NSTART, NSTOP, INC. are first and last node of the series and the increment respectively.

In above example the region defined by 16 nodes of line 4 and the nodes between 200-450 with increment of 1 will be included.

Line #6 is dummy read for variable identification

Line #7

NFILE = 0 if linear elements are used

= 1 if mixed order elements are used

KELEM = 0 for confined model

FACTOR = conversion of XY coordinates to scale which is suitable for the given device. Normally maximum of 80 in X-direction and 30 to 40 Y-direction is recommended.

If NFILE = 1, Line #8 should be the name of the file containing only the linear elements.

The program creates two binary files with extension of ".GRD" and ".NOD". In above example user will specify on terminal "HAIN3" and the program generates HAIN3.GRD and HAIN3.NOD. These two files contain the following:

1. *.GRD contains integer X (NX) and Y (NY) of a grid point, 4 nodes (NOD) of the associated linear element, and the local coordinates ξ (SC) and η (ET). This file is used by program GRIDH and GRIDIN for estimation of head and input data at each grid point, respectively.
2. *.NOD contains binary data of node number (K), water table (WTT), bottom elevation of logs (BED), number of layers (LMAX), and details of each interface of two materials (LL) and elevation of each interface (ZZ). This file is used by program GRIDIN to estimate grid values of top elevation or thickness of each hydrogeologic unit.

8.6 USER'S MANUAL FOR PROGRAM GRIDIN

This program has been primarily designed to estimate grid values of the given hydrogeologic unit from input data. Usually the conceptual model of the stratification of a given ground-water reservoir is designed using contour maps of top elevation or thickness of each major hydrogeologic unit. Through use of this program, grid values are estimated from discrete data, such that a contour map of each parameter of interest can be retrieved and checked with original maps used for discretization. Once the grid values of these parameters are estimated, it is easy to develop three-dimensional plots which can be used as illustrations.

This program uses binary files *.GRD and *.NOD generated by program GRIDIT. The following are the additional parameters entered through the terminal:

1. The program prompts "NPRINT, MATERIAL#, NTOP", and the user can define these as per details given below.

NPRINT = 0 minimum print output

= 1 for printing minimum and maximum value of the parameter of interest in a given basin.

= 2 in addition to above, each grid point, X and Y coordinate of grid points, associated nodes of the element with the grid point and the local coordinates $\xi(SC)$ and $\eta(ET)$ are printed. These are not normally required.

MATERIAL# = Identifies the material for which an estimate of the top elevation or thickness is desired.

NTOP = flag for top elevation or thickness

= 0 for top elevation

= 1 for thickness

VALUE is the binary output containing MYMAX records and in each record NXMAX words are written.

8.7 USER'S MANUAL FOR PROGRAM GRIDH

This program has been designed for generating grid values of the initial or predicted head at various hydrogeologic units or at the surface. Based on these grid values, contour maps or three-dimensional plots of head can be made for interpretation and illustration of results.

This program uses the binary file *.GRD generated by program GRIDIT and the name without extension is entered by user.

The program calls for the following additional parameters to be entered through the terminal:

1. 'NAME OF BINARY FILE HAVING HEAD DATA'. This file is created by using program HCOPI, which reads FILE-Q disk for the head data at given time step or steady-state solution on initial data and writes on this file along with a 64-character alphanumeric description of these results. This alphanumeric description is printed on the terminal for identification purposes only.
2. NPRINT, MAT# BOTTOM OF WHICH IS TO BE PLOTTED (0 = TOP).
NPRINT = 0 for minimum output
 = 1 for printing the head and grid values.
 = 2 in addition to above, the head values of the nodes of interest (given plane) are printed.
 = 3 in addition to above, the grid sequential number, X (NX), Y (NY) coordinate, 4 nodes of associated element, local coordinates ξ (SC) and η (ET) are printed. Normally this is not needed by the user.
MAT# = 0 for top surface
 = given material identification. The program searches for these head values which are at the base of the hydrogeologic unit selected by the user through this option. If the given hydrogeologic unit is missing from all the four nodes or even from two nodes of the associated linear element of the grid point, no value is assigned to these grid points.

The output file VALUE contains binary NYMAX records each containing NXMAX grid values.

9.0 COMPUTER PROGRAM LISTINGS

Computer program listings of the FE3DGW, for PDP 11/45 computer (with FILE-Q direct data access) contain the following:

1. PROG1: (to read input data and generate FILE-Q data base)

PROG1.FTN	Main Program
KXYZ.FTN	Subroutine
NODES.FTN	Subroutine
BONDRY.FTN	Subroutine
ELEM.FTN	Subroutine

2. PROG2:FTN: (to estimate integration parameter for each element)

3. BAND.FTN: (to find locations of column - identification matrix of non-zeros initial (prior to decomposition) of system matrix)

4. PROG31.FTN: (to read the surface and element based flux rates, steady state or transient simulation options, for transient simulation the time steps and time variant held boundary conditions (if required) are prescribed through this program)

5. PROG3: (to estimate steady state or transient hydraulic head as function of input data read in PROG1 and PROG3I)

LPROG3.FTN	Main
START3.FTN	Subroutine
ELEMAT.FTN	Subroutine
SYSMAT.FTN	Subroutine
MREAD.FTN	Subroutine
UPPERT.FTN	Subroutine
MREADB.FTN	Subroutine
UPERTB.FTN	Subroutine
BACKSU.FIN	Subroutine
BVALUE.FTN	Subroutine
HWRITE.FTN	Subroutine

6. PLOTEL: (to plot node location, elements and log details)

PLOTEL.FTN	Main
PLMAIN.FTN	Subroutine
HREADPL.FTN	Subroutine

7. GRIDIT.FTN: (to find local coordinates of regular grid system)

8. GRIDIN.FTN: (to find elevation and thickness of each hydrogeologic unit of the regular grid generated in GRIDIT.)

9. GRIDH: (to estimate regular grid values of computed or initial hydraulic head)

GRIDH.FTN	Main
HREAD.FTN	Subroutine
GRID.FTN	Subroutine
GRDFLQ.FTN	Subroutine

10. PLOTBC.FTN: (to plot the held potential)

11. PLFLUX.FTN: (to plot the flows prescribed for simulation)

PROG1, PROG3, PLOTEL, and GRIDH are structured for overlay execution. Copies of the task building command and overlay files are also included (Table 9.1).

TABLE 9.1. Overlay and Task Command Files

FILE NAME= PROG1.CMD

```
OP2:PROG1,=OP0:PROG1,DDL/MP
ASG=TI:2,DB0:1,OP0:3,LP:6,OP0:5
/
```

FILE NAME= PROG1.DDL

```
A:      .FCTR  OP2:PROG1
B:      .FCTR  [1,1]BNWLIB,OLB/LB
C:      .FCTR  OP2:KXYZ
D:      .FCTR  OP2:NODES
E:      .FCTR  OP2:BNDRY
F:      .FCTR  OP2:ELEM
H:      .FCTR  [4,4]PTIME
        .ROOT  A-B-H-w(C,D,E,F)
        .END
```

FILE NAME= PROG2.CMD

```
OP2:PROG2,=OP2:PROG2,OP0:[311,1]PTIME,[1,1]BNWLIB,OLB/LB
/
ASG=TI:2,LP:6,DB0:1
/
```

FILE NAME= BANO.CMD

```
OP2:BANO=OP2:BANO,OP0:[4,4]PTIME,[1,1]BNWLIB,OLB/LB
/
ASG=TI:2,DB0:1,LP:6
/
```

FILE NAME= PROG3I.CMD

```
OP2:PROG3I,=OP2:PROG3I,OP0:[4,4]PTIME,[1,1]BNWLIB,OLB/LB
/
ASG=TI:2,DB0:1,OP0:5,LP:6
/
```

FILE NAME= PROG3.CMD

```
OP2:[311,3]PROG3,LP1/SH=OP0:[311,3]PROG3.DDL/MP
ASG=TI:2,DB0:1,LP:6
/
```

TABLE 9.1. (contd)

FILE NAME= PROG3.OOL

```

A01      .FCTR   DP21(311,3)PROG3
A001     .FCTR   (4,4)PTIME
A1       .FCTR   (1,1)BNWL18,OLB/LB
B1       .FCTR   DP21(311,3)ELEMAT
C1       .FCTR   DP21(311,3)SYSHAT
D1       .FCTR   DP21(311,3)HREAD
DD1      .FCTR   DP21(311,3)HREADB
E1       .FCTR   DP21(311,3)UPPERT
EE1      .FCTR   DP21(311,3)UPPERTB
F1       .FCTR   DP21(311,3)BVALUE
G1       .FCTR   DP21(311,3)BACKSU
H1       .FCTR   DP21(311,3)HWRITE
I1       .FCTR   DP21(311,3)START3
        .ROOT    A2=A-*(A00,I,B,C,D,DD,E,EE,F,G,H)
        .END

```

FILE NAME= PLOTCL.CMD

```

DP2:PLOTCL=PLOTCL.OOL/MP
ASG=TI12,DP011:4:3:5,LP16
ACTFIL=6
/

```

FILE NAME= PLOTCL.OOL

```

A1       .FCTR   DP2:PLOTCL
B1       .FCTR   (1,1)BNWL18,OLB/LB
C1       .FCTR   DP2:HREADPL
D1       .FCTR   DP2:PLMAIN
E1       .FCTR   (4,4)ICHR2
        .ROOT    A=B-E-*(C,D)
        .END

```

FILE NAME= GRIDH.CMD

```

DP1:GRIDH=GRIDH.OOL/MP
ACTFIL=7
UNITS=7
ASG=TI12,DP011:4:3:5:7,LP16
/

```

FILE NAME= GRIDH.OOL

```

A1       .FCTR   GRIDH
B1       .FCTR   (1,1)BNWL18,OLB/LB
C1       .FCTR   HREAD
D1       .FCTR   GRID
I1       .FCTR   GROFLQ
E1       .FCTR   (4,4)ICHR2
H1       .FCTR   (311,1)PTIME
        .ROOT    A=B-E-H-*(C,D,I)
        .END

```

TABLE 9.1. (contd)

FILE NAME= PLOT8C,CMD

DP2:PLOT8C,LP1/SH=DP2:PLOT8C,DP0:[4,4]ICHR2,[1,1]BNWL18/LB
/
ASG=TI:2,DP0:5,LP:6,TG:3,DB0:1
/

FILE NAME= PLFLUX,CMD

DP2:PLFLUX,LP1/SH=DP2:PLFLUX,DP0:[4,4]ICHR2,[1,1]BNWL18/LB
/
ASG=TI:2,DP0:5,LP:6,TG:3,DB0:1
/

FILE NAME= GRIDIN,CMD

DP1:GRIDIN,LP1/SH=GRIDIN,[1,1]BNWL18/LB
[4,4]ICHR2
/
ASG=TI:2,DP0:5:1:4:3,LP:6
ACTFIL=5
/

FILE NAME= GRIDIT,CMD

DP1:GRIDIT,*GRIDIT,DP0:[1,1]BNWL18/LB
/
ACTFIL=6
ASG=TI:2,DP0:5:1:3:4,LP:6
/

9.1 PROG1 AND ITS SUBROUTINES, PROGRAM LISTINGS

```

FORTRAN IV=PLUS V02=510      16101124      13-SEP-79      PAGE 1
PROG1,FTN      /TRI8BLOCKS/WR

C      ***** PROG1,FTN *****
C-----
C      THIS PROGRAM READS THE INPUT DATA.
C      MAKES THREE DIMENSIONAL MESH AND GENERATES DISKS FOR
C      OTHER PROGRAMS.
C-----
0001      BYTE MCR(80)
0002      COMMON NUMBER(2560),IDENT(760),LAYERS(760)
0003      COMMON/BLK1/NODE(2040),NODR(760)
0004      INCLUDE 'HDR,FTN'
0005      * COMMON/HDR/LNPT,NTT,NBPTC,NBFPTC,NBTREH,NPT,NSELEM,NELEM,NODMAX,
*      #NDNZER,NQD,MATN,KTYPE,NMAX,NSTIF,NTIME,NSTORE,NQUM,TITLE(20),
*      #DP1,DP2,DP3,DP4,DP5,DP6,DP7,DP8,DP9,DP10,DP11,DP12,DP13,DP14,DP15,
*      #DP16,DP17,DP18,NTSURF,NESURF
0006      COMMON/TRACE/NPROG1,NPRDG2,NBAND,NPROG3,NPROG4,NPROG5,NPROG6
0007      COMMON/VIS/NVISC,OR,TTEMP,CON(20),VISC(20),TSURF,TEACH2,RRHO,TRHO,
+TCDEF,TVISC,CONS
0008      DIMENSION FILE(7)
0009      CALL GETMCR (MCR,ISW)
0010      IF (ISW.GT.79) ISW = -40
0011      IF (ISW.LT.0) GO TO 110
0012      DO 100 I=ISW+1,80
0013      100 MCR(I) = ' '
0014      110 CONTINUE
0015      IF (ISW.GT.1) ISW = 1
0016      IF (ISW.LT.0) ISW = -1
0017      IF (ISW.NE.1) GO TO 130
0018      IF (MCR(5).EQ.' 'OR,MCR(5).EQ.0) ISW = 2
0019      IF (ISW.NE.1) GO TO 130
0020      DECODE (43,120,MCR(5))IUN,NPRINT,NEWRUN,NPINCH,FILE
0021      120 FORMAT (4I5,7A4)
0022      130 CONTINUE
0023      IF (ISW.EQ.1) GO TO 140
0024      WRITE (2,*) 'ENTER FILE-Q DISK UNIT'
0025      READ (2,*) IUN
0026      140 IF (IUN.NE.3) CALL ASNLUN (1,'DP',IUN,ISTAT)
0027      IF (ISW.EQ.1) GO TO 160
0028      WRITE (2,*) 'ENTER INPUT FILE NAME'
0029      READ (2,150) FILE
0030      150 FORMAT (7A4)
0031      160 N = ICHR(FILE)
0032      OPEN (UNIT=5,NAME=FILE,TYPE='OLD',READONLY)
0033      IF (ISW.EQ.1) GO TO 170
0034      WRITE(2,*)'NPRINT(0,5),NPINCH(0=MISSING MATERIAL BY WEDGED ELEM,1=
+OTHERWISE)'
0035      READ (2,*) NPRINT,NPINCH
0036      WRITE (2,*) 'NEW PROBLEM(=0) OR INPUT MODIFICATION(=1)'
0037      READ (2,*) NEWRUN
0038      170 IF (NPRINT.NE.0) CALL PTIME (6)
0039      CALL OPFILE (1,'DESCRIP',25,,IPTR1)
0040      180 FORMAT (20A4)
0041      CALL OPFILE (1,'TRACE',1,,IPTR18)
0042      IF (NEWRUN.EQ.1) CALL DPR (IPTR18,0,,NPROG1,6,)
0043      IF (NEWRUN.EQ.1) CALL DPR (IPTR1,0,,LNPT,94,)
0044      READ (5,180) TITLE
0045      DPI = 25.

```

0046 DP2 = 4.
 0047 DP3 = 60.
 0048 DP4 = 2571.
 0049 DP5 = 2241.
 0050 DP7 = 28.
 0051 DP89 = 60.
 0052 DP14 = 1420.
 0053 DP15 = 2572.
 0054 DP16 = 25.
 0055 DP18 = 1.

C-----
 C IMPORTANT= IF THE VALUES OF DIMENSIONS HAVE BEEN CHANGED KEEP
 C FOLLOWING VALUES UPDATED.
 C-----

0056 LNPT = 768
 0057 NTT = 2560
 0058 NMAX = 256
 0059 NPT = 2048
 0060 NSTOP = 0
 0061 CALL KXYZ (NPRINT,NZELEV,XYCONV,ZCONV,HCONV,NSTOP,NSEA)
 0062 CALL NODES (NPRINT,NZELEV,XYCONV,ZCONV,HCONV,NSTOP,NSEC)
 0063 CALL BONDRY (NSTOP,NSEC,NPRINT,NSEA,HCONV)
 0064 CALL ELEM (NSTOP,NSEC,NPRINT,NPINCH)
 0065 DP6 = NELEM*7,+1
 0066 DP10 = NPT*2,+30.
 0067 DP11 = NELEM*13,+1.
 0068 DP12 = NPT*3,+1
 0069 DP17 = DP12
 0070 DP13 = NPT*3,+1.
 0071 CALL DPW (IPTR1,0,,LNPT,94.)
 0072 CALL DPW (IPTR1,1,,NUMBER,4896.)
 0073 CALL DPW (IPTR1,11,,IDENT,768.)
 0074 CALL DPW (IPTR1,14,,LAYERS,768.)
 0075 CALL DPW (IPTR1,17,,NODE,2048.)
 0076 IF (NPRINT.EQ.0) GO TO 190
 0077 PRINT 200, NTSURF, LNPT, NTT, NBPTC, NBFPTC, NSTREM, NPT
 0078 PRINT 210, NESURF, NSELEM, NELEM, NDOHMAX, NONZER, MATN
 0079 190 NPROG1 = 1
 0080 CALL DPW (IPTR10,0,,NPROG1,6.)
 0081 CALL DPFILE (1,'DTANDQ3',DP5,IPTR5)
 0082 CALL DPFILE (1,'BVALUE',DP15,IPTR15)
 0083 CALL DPFILE (1,'RESTART',DP16,IPTR16)
 0084 CALL DPFILE (1,'NCOL',DP10,IPTR10)
 0085 CALL DPFILE (1,'ELEMENT1',DP11,IPTR11)
 0086 C CALL DPFILE (1,'UPTRIANG',DP12,IPTR12)
 0087 CALL DPFILE (1,'BACKSUB',DP13,IPTR13)
 0088 C CALL DPFILE (1,'ELEMATRIX',DP6,IPTR6)
 0089 C CALL DPFILE (1,'UPBACKUP',DP17,IPTR17)
 0088 IF (NPRINT.NE.0) CALL PTIME (6)
 0089 200 FORMAT (/120('#')/,75X,'DIMENSIONS IN OTHER PROGRAMS HAVE TO BE SE
 +T FOR FOLLOWING MAXIMUM VALUES',/20X,'TOTAL NUMBER OF SURFACE NODE
 +S(NTSURF)=',T80,I10/,20X,'MAXIMUM NUMBER ASSIGNED TO SURFACE NODES
 +(LNPT)=',T80,I10/,20X,'TOTAL NODES IN THE SYSTEM (NTT)=',T80,I10//
 +,12X,'THE FOLLOWING PARAMETERS ARE RELATED TO B.C.',/20X,'TOTAL NUM
 +BER OF POTENTIAL B.C. PRESCRIBED (NBPTC)=',T80,I10/20X,'TOTAL NUMB
 +ER OF FLUX B.C. PRESCRIBED (NBFPTC)=',T80,I10/,20X,'TOTAL NUMBER 0

```

      +F STREAMS B.C. PRESCRIBED (NSTREM)=' ,T80,I10//,12X,'THE FOLLOWING
      + TWO PARAMETERS ARE RELATED TO STIFFNESS MATRIX'//,20X,'TOTAL NUMBE
      +R OF UNKNOWN(NPT)=' ,T80,I10//,120(' ')//)
0090      210      FORMAT (12X,'THE FOLLOWING PARAMETERS ARE ASSOCIATED WITH ELEMENTS
      +'/20X,'TOTAL NUMBER OF SURFACE ELEMENTS(NESURF) =' ,T80,I10//,20X,'
      +MAXIMUM NUMBER ASSIGNED TO SURFACE ELEMENT(NSELEM)=' ,T80,I10/20X,'
      +TOTAL NUMBER OF ELEMENTS(NELEM) =' ,T80,I10//,20X,'MAXIMUM NUMBER OF
      + NODES IN ANY ELEMENT(NODMAX)=' ,T80,I10//,20X,'MAXIMUM NON-ZERO NO
      +DES IN ANY ELEMENTS(NONZER) =' ,T80,I10//,12X,'THE FOLLOWING IS REL
      +ATED TO MATERIAL NUMBER ASSIGNED'//,20X,'MAXIMUM NUMBER ASSIGNED TO
      + MATERIAL(MATN) =' ,T80,I10//,120(' ')//)
0091      STOP
0092      END
  
```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODE1	002024 522	RW,I,CON,LCL
SPDATA	000452 149	RW,D,CON,LCL
SIDATA	002412 645	RW,D,CON,LCL
SVARS	000242 81	RW,D,CON,LCL
,SSSS.	020000 4096	RW,D,OVR,G8L
BLK1	013000 2816	RW,D,OVR,G8L
HDR	000274 94	RW,D,OVR,G8L
TRACE	000014 6	RW,D,OVR,G8L
VIS	000306 99	RW,D,OVR,G8L

TOTAL SPACE ALLOCATED = 041170 8508

,LP1=PROG1

```

C ***** KXYZ.FTN *****
C-----
C SUBROUTINE TO READ CONVERSION FACTOR AND HYDRAULIC PROPERTIES
C-----
0001 SUBROUTINE KXYZ (NPRINT,NZELEV,XYCONV,ZCONV,HCONV,NSTOP,NSEA)
0002 COMMON /KXYZ/ XK(50),YK(50),ZK(50),SPEC(50),THETA(50),ALPHA(50),
    +BETAP(50),PZERO(50),AMATER(18)
0003 INCLUDE 'HDR.FTN'
0004 * COMMON/HDR/LNPT,NTT,NBPTC,NBFTC,NSTREM,NPT,NSELEM,NELEM,NDOHAX,
    * #NONZER,NDO,MATN,KTYPE,NMAX,NSTIF,NTIME,NSTORE,NDOH,TITLE(20),
    * #DP1,DP2,DP3,DP4,DP5,DP6,DP7,DP89,DP10,DP11,DP12,DP13,DP14,DP15,
    * #DP16,DP17,DP18,NTSURF,NESURF
0005 * COMMON/VIS/NVISC,BR,TTEMP,CON(20),VISC(20),TSURF,TEACHZ,RRHO,TRHO,
    +TCOEF,TVISC,CONS
0006 DIMENSION UNITS(2)
0007 DATA UNITS/' L/T','L**2'/
0008 MA = 50
0009 CALL DPFILE (1,'PROPERTY',DP2,IPTR2)
C-----
C VALUE OF MA SHOULD BE CHANGED AS PER THE DIMENSION OF XK,YK,...
C-----
0010 IF (NPRINT.NE.0) PRINT 100, TITLE
0011 READ (5,150) NZELEV,KTYPE,PERCON,XYCONV,ZCONV,HCONV,NSEA
0012 PRINT 190
0013 IF (NPRINT.GE.2) PRINT 210, PERCON,XYCONV,ZCONV,HCONV,KTYPE,NPRINT
0014 IM = 1
0015 IF (KTYPE.EQ.1.OR,KTYPE.EQ.2) IM = 2
C-----
C MATERIAL PROPERTIES
C-----
0016 MATN = 0
0017 IF (NPRINT.GE.2) PRINT 240
0018 100 READ (5,140) NSEC,I,(AMATER(J),J=1,18)
0019 IF (NSEC.NE.0) GO TO 120
0020 IF (I.GT.MATN) MATN = I
0021 IF (I.LT.MA) GO TO 110
0022 NSTOP = 1
0023 PRINT 230, I,MA
0024 GO TO 100
0025 110 READ (5,160) O1,O2,O3,THETA(I),BETAP(I),PZERO(I),ALPHA(I),SPEC(I)
0026 XK(I) = PERCON*O1
0027 YK(I) = PERCON*O2
0028 ZK(I) = PERCON*O3
0029 IF (XK(I).EQ.0.0.OR,YK(I).EQ.0.0.OR,ZK(I).EQ.0.0) WRITE (2,*) 'MAT
    +#',I,'XK,YK,OR ZK IS 0.0 -CHECK**VALUES ARE',XK(I),YK(I),ZK(I)
0030 IF (NPRINT.LE.1) GO TO 100
0031 PRINT 220, I,(AMATER(J),J=1,18),XK(I),UNITS(IM),YK(I),UNITS(IM),
    +ZK(I),UNITS(IM),THETA(I),ALPHA(I),BETAP(I),PZERO(I),SPEC(I)
    GO TO 100
0032 120 CONTINUE
0033 CALL DPW (IPTR2,0.,XK,000.)
0034 CALL DPCL (IPTR2)
0035 IF (MATN.GT.MA) PRINT 200, MATN,MA
0036 IF (KTYPE.EQ.0) GO TO 130
C-----
C LINEAR RELATIONSHIP BETWEEN TEMPERATURE AND DEPTH ASSUMED

```

```

C      TSURF = SURFACE TEMPERATURE
C      TEACHZ = INCREASE IN TEMPERATURE EACH UNIT DEPTH
C      RRHO = AQUIFER FLUID DENSITY WITH 0 CONCENTRATION
C      TRHO = REFERENCE TEMPERATURE OF THE RRHO
C      TCOEF = COEFFICIENT OF EXPANSION PER UNIT CHANGE OF TEMPERATURE
C      TVISC = VISCOSITY OF AQUIFER FLUID AT TRHO
C      CONS = CONSTANT TO ESTIMATE HYDRAULIC CONDUCTIVITY IN GIVEN
C             UNITS USED FOR VISCOSITY, GRAVITY, DENSITY AND HEAD.
C
C      FOLLOWING IS BRIEF LISTFOR SOME CONVERSION FACTORS
C
C      CENTIPOISE*3.6 = KG/(M)(HR)
C      CENTIPOISE*6.72*10,E-4 = LB/(FT)(SEC)
C      CENTIPOISE*2.419 = LB/(FT)(HR)
C      CONS = GRAVITY TERM (IN APPROPRIATE UNITS)/(ABOVE FACTOR)
C      K = PERMEABILITY*DENSITY*G/VISCOSITY
C      = CONS * PERMEABILITY(L**2)*DENSITY(M/L**3)/VISCOSITY(CP)
C-----
0030      READ (5,160) TSURF,TEACHZ,RRHO,TRHO,TCOEF,TVISC,CONS
0039      IF (NPRINT.GE.2) PRINT 260,TSURF,TEACHZ,RRHO,TRHO,TCOEF,TVISC,CONS
C-----
C      VISCOSITY-TEMP LEAST SQUARE RELATIONSHIP AND VISCOSITY-CONCENTRATI
C      THESE PARAMETERS ARE READ IF KTYPE = 2
C-----
0040      IF (KTYPE.NE.2) GO TO 130
0041      READ (5,170) NVISC,OR,TTEMP,(CON(I),VISC(I),I=1,NVISC)
0042      IF (NPRINT.GE.2)PRINT 250,NVISC,OR,TTEMP,(CON(I),VISC(I),I=1,NVISC)
0043      130 RETURN
0044      140 FORMAT (I1,I4,10A4)
0045      150 FORMAT (2I5,4F10.0,I5)
0046      160 FORMAT (8E10.3)
0047      170 FORMAT (I10,2E10.4/20(8E10.4/))
0048      180 FORMAT (//120(' ')//,5X,20A4,//120(' ')//)
0049      190 FORMAT (//)
0050      200 FORMAT (' MAT#',I5,' MATN',I5,' PROGRAM SHALL BE STOPPED')
0051      210 FORMAT (//120(' ')//,5X,' VALUES OF CONVERSION FACTORS SPECIFIED'/
+//120(' ')//5X,' CONVERSION FACTOR FOR PERMEABILITY DATA',T70,F10.4
+//5X,' CONVERSION FACTOR FOR XY-COORDINATES',T70,F10.4//,5X,' CONVE
+RSION FACTOR FOR Z-COORDINATES',T70,F10.4//,5X,' CONVERSION FACTOR
+FOR HEAD(BOTH INITIAL AND B.C.)',T70,F10.2//,5X,' KTYPE =',T70,I10/
+//,5X,' FLAG FOR CONTROLLING THE LEVEL OF PRINT OUTPUT=',T70,I10//,
+120(' ')//)
0052      220 FORMAT (1H0//5X,23HPROPERTIES FOR MATERIAL,I4,4X,8H#####//,5X,I
+8A4//,10X,'K-X='T50,G15.6,A4//,10X,'K-Y='T50,G15.6,A4//,10X,'K-Z='
+T50,G15.6,A4//,10X,'THETA='T50,G15.6//,10X,'MEDIUM COMPRESSIBILITY'
+T50,G15.6,' 1/L'/10X,'FLUID COMPRESSIBILITY',T50,G15.6,' 1/L'/10X
+,'REFERENCE PRESSURE HEAD OF THETA',T50,G15.2,' L'/10X,'SPECIFI
+C STORAGE COEFF='T50,G15.6,' 1/L'//)
0053      230 FORMAT (//5X,'MAXIMUM MAT# READ IS',I5,'WHILE DIMENSIONED FOR',I5/
+5X,'CHANGE THE DIMENSION OF MATERIAL PROPERTIES'//,5X,' NOTE =- P
+R O G R A M S H A L L B E S S T O P P E D'//)
0054      240 FORMAT (1H1//)
0055      250 FORMAT (1H1//,5X,'NUMBER OF POINTS DEFINING VISCOSITY-CONCENTRATIO
+N RELATIONSHIP',T50,I10//5X,'LEAST SQUARE FIT FOR VISCOSITY-TEMPER
+ATURE ',T50,G10.4//5X,'REFERENCE TEMPERATURE FOR VISCOSITY-CONCENT
+RATION DATA',T50,G10.4//50(10X,'CONCENTRATION(PPM)',G15.4,10X,'VIS

```



```

0056      260      +COSITY(CP)',G15.4,/)
          FORMAT (//5X,'SURFACE TEMPERATURE',T50,G10.4,' DEGREES'//,5X,'IN
          +CRF ASE IN TEMPERATURE EACH DEPTH INCREMENT(TEACHZ)',T50,G10.4,'
          +DEGREES/UNIT DEPTH'//,5X,'DENSITY OF AQUIFER FLUID AT 0, CONCENTRA
          +TION(WRHO)',T50,G10.4,' M/L**3'//,5X,'REFERENCE TEMPERATURE FOR
          +RRHO" (TRHO)',T50,G10.4//,5X,'COEFFICIENT OF FLUID EXPANSION DUE
          +TO TEMPERATURE(TCOEF)',T50,G10.4,' M/(L**3,DEGREE)'//,5X,'VISCOS
          +ITY AT REFERENCE TEMPERATURE(TRHO) NEEDED ONLY FOR KTYPE=1',T50,G1
          +0.4//,5X,'CONSTANT (CONVERSION FACTOR) TO OBTAIN K FROM INTRINSIC
          +VALUES(CONS)',T50,G10.4//,15X,'THIS SHOULD BE EQUAL TO FOLLOWING DE
          +PENDING UPON THE UNITS USED'//,15X,' GRAVITY UNITS/APPROPRIATE CONVE
          +RSION CONSTANT FOR VISCOSITY'//,15X,'GIVEN BELOW ARE SOME CONVERSION
          +',20X,'CENTIPOISE*3.6',20X,'KG/(M)(HR)'/20X,'CENTIPOISE*6.7
          +2*10E-4',20X,'LB/(FT)(SEC)'/20X,'CENTIPOISE*2.419',20X,'LB/(FT)(H
          +R)'//)
0057      END

```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODE1	002402 641	RW,I,CON,LCL
SPDATA	000102 33	RW,O,CON,LCL
SIDATA	003704 994	RW,O,CON,LCL
SVARS	000046 19	RW,O,CON,LCL
KXYZ	003210 836	RW,O,OVR,GBL
WOR	000274 94	RW,O,OVR,GBL
VIS	000306 99	RW,O,OVR,GBL

TOTAL SPACE ALLOCATED = 012470 2716

,LP1=KXYZ

```

C ***** NODES,FTN *****
C-----
C READ NODE COORDINATES AND LAYER DETAILS
C-----
0001 SUBROUTINE NODES (NPRINT,NZELEV,XYCONV,ZCONV,HCONV,NSTOP,NSEC)
0002 INCLUDE 'H0R,FTN'
0003 * COMMON/H0R/LNPT,NTT,NBPTC,NBFPTC,NSTREX,NPT,NSELEM,NELEM,NODMAX,
* #NONZER,NQD,MATN,KTYPE,NMAX,NSTIP,NTIME,NSTORE,NDOU,TITLE(20),
* #OP1,OP2,OP3,OP4,OP5,OP6,OP7,OP89,OP10,OP11,OP12,OP13,OP14,OP15,
* #OP16,OP17,OP18,NTSURF,NEISURF
0004 COMMON /NODES/ X(256),Y(256),Z(256),FACTOR(256),HEAD(256),RHO(256)
0005 COMMON/VIS/NVIS8,8N,TTEMP,CON(20),VISC(20),TSURF,TEACHZ,RRHO,TRHO,
+TCOEFF,TVISC,CONS
0006 DIMENSION CONCEN(20),LL(20),ZZ(20),Z5(20),WM(20)
0007 COMMON NUMBER(2560),IDENT(760),LAYERS(760)
0008 CALL OPFILE (1,'XYZCOR',OP3,IPTR3)
0009 CALL OPFILE (1,'FACTOR',OP89,IPTR89)
0010 CALL OPFILE (1,'HEAD',OP14,IPTR14)
0011 LN = LNPT
0012 N3 = NTT
0013 NNEW = 0
0014 NMAX = 256
0015 DO 100 I=1,20
0016 100 CONCEN(I) = 0.0
0017 LNPT = 0
0018 NSTOP = 0
0019 NT = 0
0020 NONZER = 0
0021 NSAVE = 1000
0022 KSAVE = 1000
0023 PAG = 0.
0024 IF (NPRINT.GE.2) PRINT 400
0025 DO 300 I=1,1000
0026 READ (5,360) NSEC,K,XX,YY,WT,RED,NBR,LMAX,(LL(J),ZZ(J),J=1,LMAX)
0027 IF (NPRINT.EQ.4) PRINT 400, NSEC,K,XX,YY,WT,RED,NBR,LMAX,LL(1),
+ (ZZ(J),LL(J+1),J=1,LMAX)
0028 IF (NSEC.NE.0) GO TO 290
0029 IF (KTYPE.EQ.2) READ (5,380) (CONCEN(J),J=1,LMAX)
0030 LAYERS(K) = LL(LMAX)/100
0031 IF (NZELEV.EQ.1) GO TO 110
0032 IF (RED.GT.0.0) GO TO 110
0033 PRINT 390, RED,K
0034 NSTOP = 1
0035 GO TO 300
0036 110 DO 120 J=1,LMAX
0037 120 ZZ(J) = ZZ(J)*ZCONV
0038 IF (K.EQ.NSAVE) PRINT 470, K,NSAVE
0039 XX = XX*XYCONV
0040 YY = YY*XYCONV
0041 NSAVE = K
0042 KK = K/100
0043 IF (NZELEV.EQ.1) GO TO 150
0044 SUM = 0.0
0045 DO 130 J=1,LMAX
0046 130 SUM = SUM+ZZ(J)
0047 DIF = (RED-SUM)/SUM

```

```

0048      SUM = 0.0
0049      Z5(1) = WTT
0050      WM(1) = WTT
0051      DO 140 J=1,LMAX
0052      ZZ(J) = ZZ(J)*(1+DIF)
0053      SUM = SUM+ZZ(J)
0054      Z5(J+1) = WTT-SUM
0055      WM(J+1) = WTT
0056      140  CONTINUE
0057      GO TO 170
0058      150  Z5(1) = ZZ(1)
0059      WM(1) = WTT
0060      DO 160 J=2,LMAX
0061      Z5(J) = ZZ(J)
0062      160  WM(J) = WTT
0063      170  JJ = 0
0064      LAST = 0
0065      ISTOP = 0
0066      DO 300 J=1,LMAX
0067      IF (LL(J).EQ.0) GO TO 220
0068      JJ = JJ+1
0069      LLL = LL(J)
0070      IF (JJ.EQ.1) GO TO 180
0071      KK = LLL/100
0072      LAST = LLL-KK*100
0073      IF (KK.LT.KSAVE) PRINT 420, KK,KSAVE,K,J
0074      GO TO 190
0075      180  KK = 0
0076      190  IF (KK.EQ.KSAVE) PRINT 440, KK,KSAVE,K,J
0077      KSAVE = KK
0078      NT = NT+1
0079      IF (NT.GT.N3) GO TO 210
0080      NNN = K+KK*1000
0081      IF (K.GT.LNPT) LNPT = K
0082      IF (K.GT.LN) GO TO 200
0083      NUMBER(NT) = NNN
0084      IF (JJ.EQ.1) IDENT(K) = NT
0085      NNEW = NNEW+1
0086      X(NNEW) = XX
0087      Y(NNEW) = YY
0088      Z(NNEW) = Z5(J)*ZCONV
0089      HEAD(NNEW) = WM(J)*HCONV
0090      GO TO 220
0091      200  NSTOP = 1
0092      PRINT 460, K,LN
0093      GO TO 220
0094      210  NSTOP = 1
0095      220  CONTINUE
0096      IF (KTYPE.EQ.0.AND.NPRINT.LE.1) GO TO 200
0097      IF (KTYPE.EQ.0) GO TO 240

```

```

C-----
C   LINEAR RELATIONSHIP OF TEMPERATURE WITH DEPTH HAS BEEN ASSUMED
C-----

```

```

0098      ZP = Z5(1)-Z5(J)
0099      TEMP = TSURF+TEACHZ*ZP
0100      RHO(NNEW) = RRHO*(1+TCOEF*(TEMP-TRHO))

```

```

0101      IF (KTYPE.EQ.1) XMEW = TVISC*EXP(BR*(1/TEMP-1/TTEMP))
0102      IF (KTYPE.NE.2) GO TO 230
0103      CONC = CONCEN(J)
0104      CALL MEW (XMEW,NVISC,CON,VISC,CONC)
0105      RHO(NNEW) = RHO(NNEW)*(1+CONC*1,E-6)
0106      XMEW = XMEW*EXP(BR*(1/TEMP-1/TTEMP))
0107      230 FACTOR(NNEW) = RHO(NNEW)*CONS/XMEW
0108      240 IF (NPRINT.LE.1) GO TO 280
0109      N = NUMBER(NT)
0110      IF (N.GT.999) GO TO 250
0111      IF (NPRINT.NE.2) PRINT 450
0112      GO TO 260
0113      250 IF (NPRINT.EQ.3.OR.NPRINT.EQ.5) GO TO 260
0114      IF (NPRINT.EQ.3) GO TO 260
0115      GO TO 280
0116      260 IF (KTYPE.EQ.0) GO TO 270
0117      PRINT 430, NT,N,XX,YY,Z(NNEW),HEAD(NNEW),TEMP,RHO(NNEW),XMEW,
+CONCEN(J),FACTOR(NNEW)
0118      GO TO 280
0119      270 PRINT 430, NT,N,XX,YY,Z(NNEW),HEAD(NNEW)
0120      280 IF (NNEW.LT.NMAX) GO TO 300
0121      290 IF (NNEW.EQ.0) GO TO 310
0122      CALL DPW (IPTR3,PAG,Z,512,)
0123      CALL DPW (IPTR3,PAG+20,,X,512,)
0124      CALL DPW (IPTR3,PAG+40,,Y,512,)
0125      NNEW = 0
0126      CALL DPW (IPTR14,PAG,HEAD,512,)
0127      IF (KTYPE.NE.0) CALL DPW (IPTR89,PAG,FACTOR,512,)
0128      IF (KTYPE.NE.0) CALL DPW (IPTR89,PAG+40,,RHO,512,)
0129      IF (NSEC.NE.0) GO TO 310
0130      PAG = PAG+2.
0131      300 CONTINUE
0132      310 NTSURF = I-1
0133      IF (NPRINT.NE.0) PRINT 510, NT,NTSURF,LNPT
0134      IF (LNPT.GT.LN) PRINT 500, LNPT,LN
0135      NTT = NT
0136      IF (NTT.GT.N3) PRINT 490, N3,NTT
0137      IF (NSTOP.EQ.1) STOP
0138      LOCATE = 1
0139      IF (NPRINT.EQ.4) PRINT 410, NSEC,LOCATE
0140      IF (NSEC.NE.1) GO TO 350
0141      NTIMES = NTT/NMAX+1
0142      NMM = NMAX
0143      DO 340 I=1,NTIMES
0144      IF (I.EQ.NTIMES) NMM = NTT-NMAX*(I-1)
0145      IF (CONVER.EQ.0,0,OR.CONVER.EQ.1,0) GO TO 330
0146      READ (5,380) (HEAD(J),J=1,NMM)
0147      DO 320 J=1,NMM
0148      320 HEAD(J) = HEAD(J)*HCONV
0149      330 PRINT 530, (NUMBER((I-1)*NMAX+J),HEAD(J),J=1,NMM)
0150      RNMM = NMM*2.
0151      340 CALL DPW (IPTR14,I*2,-1,HEAD,RNMM)
0152      PRINT 520
0153      READ (5,370) NSEC
0154      350 CONTINUE
C-----

```

```

C      FORMATS
C-----
0155  360  FORMAT (I1,I4,4F10,2,3I5,/2(6(F8,1,I5)/))
0156  370  FORMAT (I1,I4,F10,0)
0157  380  FORMAT (16F5,0)
0158  390  FORMAT (/ ' WARNING ## CHECK TOTAL DEPTH',F8,2,' FOR NODE ',I5/)
0159  400  FORMAT (2X,'COPY OF COORDINATE DATA CARD',I3,I4,4F10,2,3I5,/32X,2(
+6(F8,1,I5)/))
0160  410  FORMAT (' NSEC=',I3,' AT LOCATION=',I3)
0161  420  FORMAT (' NEW NODE = PREVIOUS=KK',I5,' KSAVE',I5,' NODE=',I5,
+ ' INTERFACE ',I5)
0162  430  FORMAT (2I9,5F10,1,F10,3,3E10,3)
0163  440  FORMAT (' HOW COME THE NEW KK',I5,' IS SAME AS PREVIOUS',I5,' NODE'
+ ',I5,' INTERFACE',I5)
0164  450  FORMAT (//)
0165  460  FORMAT (' NODE',I5,' GREATER THAN LNPT',I5,' PROGRAM SHALL STOP')
0166  470  FORMAT (' PRESENT NODE IS',I5,' # PREVIOUS NODE WAS ',I5/)
0167  480  FORMAT (1H1//,120(' ')/,10X,' COORDINATES AND OTHER DETAILS OF EAC
+H NODE',I5,120(' ')/,4X,' INTERNAL USER  X-COORD.  Y-COORD.  Z-
+COORD.  INITIAL  ',TEMP C DEG  RHO  XMEW CONCENTRATION FACTOR'/4X
+ ', NODE#  NODE#',33X,'HEAD',32X,' PPM')
0168  490  FORMAT (/5X,'PRESCRIBED NIT=',I5,' WHILE NEEDED IS=',I5,/,5X,'N O
+T E = - P R O G R A M S H A L L B E S T O P P E D'/)
0169  500  FORMAT (/5X,'PRESCRIBED MAXIMUM NODE NUMBER OF SURFACE NODE READ
+ AS',I5,' WHILE DIMENSIONED FOR(LNPT)',I5,/,10X,' P R O G R A M
+ S T O P P E D ##### CHECK THE INPUT DATA OR CHANGE THE DIMENSION
+//)
0170  510  FORMAT (/10X,'TOTAL NOUES IN SYSTEM=',T80,I7/,10X,'NUMBER OF SURFA
+CE NODES',T80,I7/,10X,'MAXIMUM NUMBER ASSIGNED TO THE SURFACE NODE
+ ',T80,I7/)
0171  520  FORMAT (/120(' ')/)
0172  530  FORMAT (1H1//,120(' ')/,5X,'PRESCRIBED INITIAL CONDITIONS'/120(' '
+ ')/,200(4(' NODE#',I6,' HEAD=',F8,0)/))
0173  RETURN
0174  END

```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODE1	005370 1404	RW,I,CON,LCL
SPDATA	000042 17	RW,D,CON,LCL
SIDATA	002370 636	RW,U,CON,LCL
SVARS	000714 230	RW,D,CON,LCL
STEMPS	000010 4	RW,D,CON,LCL
HDR	000274 94	RW,U,OVR,GBL
NODES	014000 3072	RW,D,OVR,GBL
VIS	000306 99	RW,D,OVR,GBL
,\$\$\$\$	020000 4096	RW,U,OVR,GBL

TOTAL SPACE ALLOCATED = 045550 9652

```

0001      SUBROUTINE MEW (XMEW,NVISC,CON,VISC,CONC)
0002      DIMENSION VISC(NVISC), CON(NVISC)
0003      DO 130 I=1,NVISC
0004      IF (CONC.GT,CON(I)) GO TO 130
0005      TT = CON(I)
0006      IF (CONC.NE,TT) GO TO 100
0007      YY = VISC(I)
0008      GO TO 140
0009  100    IF (I.NE,1) GO TO 110
0010      YY = VISC(1)
0011      GO TO 140
0012  110    IF (I.NE,NVISC) GO TO 120
0013      YY = VISC(NVISC)
0014      GO TO 140
0015  120    DIF = VISC(I)-VISC(I+1)
0016      COIF = CON(I+1)-TT
0017      YY = VISC(I)-(DIF*(CONC-TT)/COIF)
0018      GO TO 140
0019  130    CONTINUE
0020      YY = VISC(NVISC)
0021  140    XMEW = YY
0022      RETURN
0023      END
  
```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODE1	000462 153	RW,I,CON,LCL
SIDATA	000040 16	RW,D,CON,LCL
SVAR5	000022 9	RW,D,CON,LCL
STEMPS	000002 1	RW,D,CON,LCL

TOTAL SPACE ALLOCATED = 000546 179

,LP:=NODES

```

C ***** BONDRY,FTN *****
C-----
C      SUBROUTINE FOR READING THE BOUNDARY CONDITIONS
C-----
0001      SUBROUTINE BONDRY (NSTOP,NSEC,NPRINT,NSEA,HCONV)
0002      INCLUDE 'HDR,FTN'
0003      * COMMON/HDR/LNPT,NTT,NBPTC,NBFTC,NSTREM,NPT,NSELEM,NELEM,NODMAX,
      * #NONZER,NQ0,HATN,KTYPE,NMAX,NSTIF,NTIME,NSTORE,NOUM,TITLE(20),
      * #DP1,DP2,DP3,DP4,DP5,DP6,DP7,DP89,DP10,DP11,DP12,DP13,DP14,DP15,
      * #DP16,DP17,DP18,NTSURF,NESURF
0004      COMMON NUMBER(2560),IDENT(768),LAYERS(768)
0005      COMMON/BLK1/NODE(2048),NODB(768)
0006      COMMON /BNDRY/RIV(768),BIVF(128),STREAM(128),ELEVA(6),NF(6,128),
      +NPOINT(128),NFLUX(768),LENGTH(768),ELEV(768),NODRF(128)
0007      DIMENSION HEAD(256),X(256),Y(256),Z(256)
0008      CALL DPFIL (1,'BOUNDCONO',DP7,IPTR7)
0009      CALL DPFIL (1,'HEAD',DP14,IPTR14)
0010      CALL DPFIL (1,'XYZCOR',DP3,IPTR3)
0011      LP = 512
0012      I = 1
0013      LOCATE = 2
0014      IF (NPRINT.EQ.4) PRINT 350, NSEC,LOCATE
0015      IF (NSEC.NE.2) GO TO 160
0016      NMM = 0
0017      NMAX3 = 768
0018      PAG = 0.
0019      CALL DPR (IPTR14,PAG,HEAD,512.)
0020      CALL DPR (IPTR3,PAG,Z,512.)
0021      IFLAG = 1
0022      IF (HCONV.EQ.0.0.OR.HCONV.EQ.1.0) IFLAG = 0
0023      DO 140 I=1,1000
0024      READ (5,320) NSEC,MP,81
0025      IF (NPRINT.EQ.4) WRITE (6,360) NSEC,MP,81
0026      IF (NSEC.NE.0) GO TO 160
0027      IF (I.GT.LP) GO TO 150
0028      IF (IFLAG.EQ.1) 81 = 81*HCONV
0029      NT = MOD(MP,1000)
0030      NT = IDENT(NT)
0031      IF (NT.LE.0.OR.NT.GT.NTT) GO TO 130
0032      DO 120 J=NT,NTT
0033      IF (MP.NE.NUMBER(J)) GO TO 120
0034      NODR(I) = J
0035      NM = (J-1)/NMAX
0036      IF (NM.EQ.NMM) GO TO 100
0037      CALL DPH (IPTR14,PAG,HEAD,512.)
0038      NMM = NM
0039      PAG = NM*2.
0040      CALL DPR (IPTR3,PAG,Z,512.)
0041      CALL DPR (IPTR14,PAG,HEAD,512.)
0042      M = J-NM*NMAX
0043      IF (81.NE.0.0) GO TO 110
0044      IF (Z(M).GE.0.0) GO TO 110
0045      IF (NSFA.EQ.0) GO TO 110
C-----
C      THIS IS DONE FOR ACCOUNTING SEA WATER DENSITY
C-----

```

```

0046      RI = -Z(M)*0.025
0047      110  HEAD(M) = RI
0048          RIV(I) = RI
0049          GO TO 140
0050      120  CONTINUE
0051      130  PRINT 370, I,NOOR(I),NT
0052          NOOR(I) = 0
0053      140  CONTINUE
0054          GO TO 160
0055      150  NSTOP = 1
0056          PRINT 380, LP, I
0057          STOP
0058      160  NBPTC = I-1
0059          CALL DPR (IPTR14,PAG,HEAD,512.)
0060          IF (NBPTC,LT,0) NBPTC = 0
0061          DO 170 I=1,LNPT
0062              LENGTH(I) = 0
0063              ELFV(I) = 0.0
0064      170  NFLUX(I) = 0
0065              I = 0
0066              NSTREN = 0
0067              LOCATE = 3
0068              IF (NPRINT.EQ,4) PRINT 350, NSEC,LOCATE
0069              IF (NSEC.NE,3) GO TO 220
0070              NMH = 0
0071              PAG = NMH*2
0072              CALL DPR (IPTR3,PAG+20.,X,512.)
0073              CALL DPR (IPTR3,PAG+40.,Y,512.)
0074      180  I = I+1
0075          DO 190 J=1,4
0076      190  NF(J,I) = 0
0077              READ (5,330) NSEC,NFF,STREAM(I),(NF(J,I),ELEVA(J),J=1,NFF)
0078              NPOINT(I) = NFF
0079              IF (NPRINT.EQ,4) PRINT 340, STREAM(I),NFF,(NF(J,I),J=1,NFF)
0080              IF (NSEC.NE,0) GO TO 240
0081              DO 210 J=1,NFF
0082                  NN = NF(J,I)
0083                  N = IDENT(NN)
0084                  NFLUX(N) = 1
0085                  ELEVA(N) = ELEVA(J)
0086                  NM = (N-1)/NMAX
0087                  IF (NM.EQ,NMH) GO TO 200
0088                  NMH = NM
0089                  PAG = NM*2.
0090                  CALL DPR (IPTR3,PAG+20.,X,512.)
0091                  CALL DPR (IPTR3,PAG+40.,Y,512.)
0092      200  M = N-NM*NMAX
0093          C  CHECK THE LENGTH OF STREAM
0094              IF (J.EQ,1) GO TO 210
0095              LT = 0.5*SQRT((X(M)-XPRE)**2+(Y(N)-YPRE)**2)
0096              LENGTH(N) = LENGTH(N)+LT
0097              LENGTH(NPRE) = LENGTH(NPRE)+LT
0098      210  NPRE = N
0099              XPRE = X(M)
0100              YPRE = Y(N)
0101              GO TO 180

```



```

0101      220      IF (I.EQ.0) I = 1
0102          NSTREM = I-1
0103          IF (NSTREM.LT.0) GO TO 240
0104          IF (NPRINT.LE.1) GO TO 240
0105          DO 230 I=1,NPT
0106          IF (NPLUX(I).EQ.0) GO TO 230
0107          PRINT 390, I,NUMBER(I),LENGTH(I),ELEV(I)
0108      230      CONTINUE
0109      240      LOCATE = 4
0110          IF (NPRINT.EQ.4) PRINT 350, NSEC,LOCATE
0111          CALL DPCL (IPTR3)
0112          NBPTC = 0
0113          NODBF(1) = 0
0114          BIVF(1) = 0.0
0115          IF (NSEC.NE.4) GO TO 290
0116          DO 270 I=1,NTT
0117          READ (5,320) NSEC,NODBF(I),BIVF(I)
0118          IF (NPRINT.EQ.4) PRINT 400, NSEC,NODBF(I),BIVF(I)
0119          IF (NSEC.NE.0) GO TO 280
0120          NP = NODBF(I)
0121          NT = MOD(NP,1000)
0122          NT = IDENT(NT)
0123          IF (NT.LE.0.OR.NT.GT.NTT) GO TO 260
0124          DO 250 J=NT,NTT
0125          IF (NP.NE.NUMBER(J)) GO TO 250
0126          NODBF(I) = J
0127          GO TO 270
0128      250      CONTINUE
0129      260      PRINT 410, I,NODBF(I),NT
0130          NODBF(I) = 0
0131      270      CONTINUE
0132      280      NBPTC = I-1
0133      290      LOCATE = 5
0134          IF (NBPTC.EQ.0) PRINT 450
0135          CALL DPW (IPTR7,0.,NODB,768.)
0136          CALL DPW (IPTR7,3.,BIV,1536.)
0137          CALL DPW (IPTR7,9.,NPLUX,768.)
0138          CALL DPW (IPTR7,12.,LENGTH,768.)
0139          CALL DPW (IPTR7,15.,ELEV,1536.)
0140          CALL DPW (IPTR7,21.,NPOINT,120.)
0141          CALL DPW (IPTR7,22.,STREAM,256.)
0142          CALL DPW (IPTR7,23.,NP,768.)
0143          CALL DPW (IPTR7,26.,NODBF,128.)
0144          CALL DPW (IPTR7,27.,BIVF,256.)
0145          CALL DPCL (IPTR7)
0146          IF (NPRINT.EQ.4) PRINT 350, NSEC,LOCATE
0147          IF (NPRINT.LE.1) GO TO 310
0148          IF (NBPTC.EQ.0) GO TO 300
0149          PRINT 420, NBPTC,(NUMBER(NODB(I)),BIV(I),I=1,NBPTC)
0150          PRINT 430
0151      300      IF (NBPTC.EQ.0) GO TO 310
0152          PRINT 440, NBPTC,(NUMBER(NODBF(I)),BIVF(I),I=1,NBPTC)
0153          PRINT 430
0154      310      RETURN
0155      320      FORMAT (I1,I4,F10.0)
0156      330      FORMAT (I1,I4,1X,A4,7(I5,F5.1))

```

```

0157 340 FORMAT (' STREAM NAME=',A4,' TOTAL NODES=',I5,' THE NODES ARE=',
+10I5)
0158 350 FORMAT (' NSEC=',I3,' AT LOCATION=',I3)
0159 360 FORMAT (' COPY OF HEAD B.C. CARD ',I1,I5,F10.2)
0160 370 FORMAT (' FOR I = ',I5,' THE BOUNDARY NODE PRESCRIBED AS',I5,' NO
+ COORDINATES HAVE BEEN READ = NT=',I10)
0161 380 FORMAT (///,5X,'PRESCRIBED POTENTIAL B.C. ARE',I5,'WHILE ALREADY RE
+AD ARE',I5//,5X,'N O T E = - P R O G R A M S T O P P E D',//,20X,
+'CHANGE NBPTC VALUE AND ASSOCIATED PARAMETERS OR CHECK THE ERRO
+R IN INPUT DECK'//)
0162 390 FORMAT (' INTERNAL NODE#',I5,' GIVEN NODE#',I5,' TOTAL LENGTH REP
+RESENTED',I5,' ELEVATION=',F12.1)
0163 400 FORMAT (' COPY OF FLUX B.C. CARD ',I1,I5,F10.2)
0164 410 FORMAT (' ERROR ## ERROR IN FLUX NODE READ FOR I=',I4,' NODE GIVEN
+',I5,' FOR WHICH NO COORDINATES GIVEN-- NT=',I10)
0165 420 FORMAT (1H1//,120(' ')/,5X,'TOTAL NUMBER OF NODES WITH PRESCRIBED
+ HEAD B.C.ARE=',I10//,120(' ')/,200(4(' NODE#',I6,' HEAD=',F8.2)/)
+)
0166 430 FORMAT (//120(' ')/)
0167 440 FORMAT (1H1//,120(' ')/,5X,'TOTAL NUMBER OF NODES WITH FLUX B.C.='
+',I10//,120(' ')/,200(4(' NODE#',I6,' FLUX=',G10.3)/))
0168 450 FORMAT (//120(' ')/,10X,'SINCE NOT A SINGLE POTENTIAL B.C. GIVEN'
+THE SOLUTION OF PROBLEM SHALL NOT BE UNIQUE,'//,10X,'CHECK THIS AS
+PECT BEFORE RUNNING OTHER PROGRAMS ### IMPORTANT'//,120(' ')/)
0169      END

```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODE1	004726 1259	RW,I,CON,LCL
SPDATA	000134 46	RW,O,CON,LCL
SIDATA	002310 612	RW,O,CON,LCL
SVARS	010070 2076	RW,O,CON,LCL
STEMPS	000012 5	RW,O,CON,LCL
HDR	000274 94	RW,O,QVR,GBL
,3335.	020000 4096	RW,O,QVR,GBL
BLK1	013000 2816	RW,O,QVR,GBL
BNDY	030030 6156	RW,O,QVR,GBL

TOTAL SPACE ALLOCATED = 103020 17160

,LP: BONDY

```

C ***** ELEM,FTN *****
C-----
C      SUBROUTINE FOR FORMATION OF ELEMENTS
C-----
0001      SUBROUTINE ELEM (NSTOP,NSEC,NPRINT,NPINCH)
0002      INCLUDE 'HDR,FTN'
0003      * COMMON/HDR/LNPT,NTT,NBPTC,NBFTC,NSTREM,NPT,NSELEM,NELEM,NODMAX,
      * * *
      * * *
0004      COMMON NUMBER(2560),IDENT(768),LAYERS(768)
0005      COMMON/BLK1/NODE(2048),NOD8(768)
0006      COMMON /ELEM/N1,NNT,NODNEW(44),MM1,NDUMM,NOELEM,NHEAD,MIXNOD(8)
0007      COMMON/EL/NYES(2400),NM(44),NOD(44),NNODE(44),ANJ(8),NENAME(2560)
0008      DATA ANJ/3H0NE,3HTWO,4HTHRE,4HMXED /
0009      CALL DPFILE (1,'ELEMENT0',DP4,IPTR4)
0010      KL = NPT
0011      NO = 44
0012      DO 100 I=1,2400
0013      100      NYES(I) = 0
0014      NODMAX = 0
0015      NELEM = 0
0016      NSELEM = 0
0017      IF (NSEC,NE,5) GO TO 300
0018      DO 370 IE=1,2600
0019      INNM = 0
0020      DO 110 I=1,NO
0021      110      NOD(I) = 0
0022      READ (5,520) NSEC,K,N1,NN,(NOD(J),J=1,NN)
0023      IF (NPRINT,EQ,4) PRINT 660, NSEC,K,N1,NN,(NOD(J),J=1,NN)
0024      IF (NSEC,NE,0) GO TO 300
0025      IF (NSELEM,LT,K) NSELEM = K
0026      IF (N1,EQ,0) READ (5,530) (MIXNOD(I),I=1,4)
0027      IF (N1,EQ,0,AND,NPRINT,EQ,4) PRINT 670, (MIXNOD(I),I=1,4)
0028      IF (N1,NE,0) GO TO 170
0029      DO 160 I=1,4
0030      NKT = MIXNOD(I)
0031      MD1 = NKT/100
0032      MD2 = MOD(NKT,100)/10
0033      MD3 = MOD(NKT,10)
0034      IF (MD3,NE,1) PRINT 620, I,K,MD3
0035      IF (I,GT,2) GO TO 150
0036      IF (MD1,EQ,2,AND,NOD(5),EQ,0) PRINT 540, I,MIXNOD(I),K
0037      IF (MD1,EQ,3,AND,NOD(9),EQ,0) PRINT 540, I,MIXNOD(I),K
0038      IF (MD1,EQ,3,AND,NOD(10),EQ,0) PRINT 540, I,MIXNOD(I),K
0039      IF (I,EQ,2) GO TO 130
0040      120      IF (MD2,EQ,3,AND,NOD(13),EQ,0) PRINT 540, I,MIXNOD(I),K
0041      IF (MD2,EQ,3,AND,NOD(14),EQ,0) PRINT 540, I,MIXNOD(I),K
0042      IF (MD2,EQ,2,AND,NOD(7),EQ,0) PRINT 540, I,MIXNOD(I),K
0043      IF (I,GT,2) GO TO 140
0044      GO TO 160
0045      130      IF (MD2,EQ,3,AND,NOD(15),EQ,0) PRINT 540, I,MIXNOD(I),K
0046      IF (MD2,EQ,3,AND,NOD(16),EQ,0) PRINT 540, I,MIXNOD(I),K
0047      IF (MD2,EQ,2,AND,NOD(8),EQ,0) PRINT 540, I,MIXNOD(I),K
0048      IF (I,GT,2) GO TO 140
0049      GO TO 160

```

```

0050      140      IF (MD1, EQ, 2, AND, NOD(6), EQ, 0) PRINT 540, I, MIXNOD(I), K
0051              IF (MD1, EQ, 3, AND, NOD(11), EQ, 0) PRINT 540, I, MIXNOD(I), K
0052              IF (MD1, EQ, 3, AND, NOD(12), EQ, 0) PRINT 540, I, MIXNOD(I), K
0053              GO TO 160
0054      150      IF (I, EQ, 3) GO TO 130
0055              IF (I, EQ, 4) GO TO 120
0056      160      CONTINUE
0057      170      CONTINUE
0058              MAX = 0
0059              MIN = 100
0060              DO 180 I=1, NN
0061                  ND = NOD(I)
0062                  IF (ND, EQ, 0) GO TO 180
0063                  LLL = LAYERS(ND)
0064                  IF (MAX, LT, LLL) MAX = LLL
0065                  NT = IDENT(ND)
0066                  LTOP = NUMBER(NT+1)/1000
0067                  NM(I) = NT
0068                  IF (LTOP, LT, MIN) MIN = LTOP
0069      180      CONTINUE
0070              DO 360 IT=1, MAX
0071      C              IF (IT, NE, 1) NN = 4
0072                  NO1 = 0
0073                  DO 190 I=1, 36
0074                      NODNEW(I) = 0
0075      190      NNODE(I) = 0
0076                  DO 200 I=1, NN
0077                      IF (NOD(I), EQ, 0) GO TO 200
0078                      NNODE(I) = NOD(I)
0079      200      CONTINUE
0080                  NN2 = 16+NN
0081                  DO 230 I=17, NN2
0082                      NP = I-16
0083                      IF (NOD(NP), EQ, 0) GO TO 230
0084                      NT = NM(NP)+1
0085                      LLL = NUMBER(NT)+.001
0086                      IF (LLL, EQ, MIN) GO TO 210
0087                      IF (NPINCH, EQ, 0) GO TO 220
0088                      MIN = MIN+1
0089                      GO TO 360
0090      210      NO1 = 1
0091                      NNODE(I) = NUMBER(NT)
0092                      NOD(NP) = NNODE(I)
0093                      NM(NP) = NT
0094                      GO TO 230
0095      220      NNODE(I) = NOD(NP)
0096      230      CONTINUE
0097                      MM1 = MIN
0098                      MIN = MIN+1
0099                      IF (NO1, EQ, 0) GO TO 360
0100                      DO 240 I=1, 4
0101                          NODNEW(I+4) = NNODE(I+16)
0102      240      NODNEW(I) = NNODE(I)
0103                      IF (N1, EQ, 1) GO TO 250
0104                      NODNEW(9) = NNODE(5)
0105                      NODNEW(10) = NNODE(6)

```

```

0105      NODNEW(11) = NNODE(21)
0106      NODNEW(12) = NNODE(22)
0107      NODNEW(13) = NNODE(7)
0108      NODNEW(14) = NNODE(8)
0109      NODNEW(15) = NNODE(23)
0110      NODNEW(16) = NNODE(24)
0111      IF (N1.EQ.2) GO TO 250
0112      NODNEW(21) = NNODE(9)
0113      NODNEW(22) = NNODE(10)
0114      NODNEW(23) = NNODE(11)
0115      NODNEW(24) = NNODE(12)
0116      NODNEW(25) = NNODE(25)
0117      NODNEW(26) = NNODE(26)
0118      NODNEW(27) = NNODE(27)
0119      NODNEW(28) = NNODE(28)
0120      NODNEW(29) = NNODE(13)
0121      NODNEW(30) = NNODE(14)
0122      NODNEW(31) = NNODE(15)
0123      NODNEW(32) = NNODE(16)
0124      NODNEW(33) = NNODE(29)
0125      NODNEW(34) = NNODE(30)
0126      NODNEW(35) = NNODE(31)
0127      NODNEW(36) = NNODE(32)
0128      250 NMT = 36
0129      NCOUNT = 8
0130      NNT = 8
0131      DO 260 I=9,NMT
0132      IF (NODNEW(I).EQ.0) GO TO 260
0133      NNT = I
0134      NCOUNT = NCOUNT+1
0135      260 CONTINUE
0136      IF (NCOUNT.GT.NONZER) NONZER = NCOUNT
0137      DO 270 I=1,4
0138      270 MIXNOD(I+4) = MIXNOD(I)
0139      NOELEM = K+(IT-1)*1000
0140      NHEAD = 0
0141      IF (NPRINT.LE.1) GO TO 300
0142      IF (IT.EQ.1) GO TO 280
0143      IF (NPRINT.EQ.5) GO TO 290
0144      GO TO 300
0145      280 IF (NPRINT.EQ.5) PRINT 590, NOELEM
0146      290 M5 = N1
0147      IF (N1.EQ.0) M5 = 4
0148      NE = NOELEM
0149      PRINT 550, NE, ANJ(M5), MM1, NNT, (NODNEW(I), I=1,8)
0150      IF (N1.EQ.2) PRINT 650, (NODNEW(I), I=9,20)
0151      IF (N1.EQ.3) PRINT 640, (NODNEW(I), I=21,44)
0152      IF (N1.NE.0) GO TO 300
0153      PRINT 630, (NODNEW(I), I=9,NNT)
0154      PRINT 570
0155      PRINT 600, (MIXNOD(I), I=1,8)
0156      300 DO 350 I=1,NNT
0157      NP = NODNEW(I)
0158      IF (NP.EQ.0) GO TO 350
0159      ID = MOD(NP,1000)
0160      ID = IDENT(ID)

```

```

0161      IF (ID.NE.0.AND.ID.LT.NTY) GO TO 310
0162      ID = 1
0163      310  DO 320 J=ID,NTY
0164          IF (NP.NE.NUMBER(J)) GO TO 320
0165          NODNEW(I) = J
0166          NYES(J) = 1
0167          GO TO 330
0168      320  CONTINUE
0169          PRINT 500, NODNEW(I),NOELEM
0170          INNN = 1
0171          GO TO 350
0172      330  IF (NHEAD.EQ.1) GO TO 350
0173          DO 340 KKK=1,NBPTC
0174          IF (J.NE.NODB(KKK)) GO TO 340
0175          NHEAD = 1
0176          GO TO 350
0177      340  CONTINUE
0178      350  CONTINUE
0179          IF (NODMAX.LT.NNT) NODMAX = NNT
0180          NELEM = NELEM+1
0181          ADR = NELEM
0182          NENAME(NELEM) = NOELEM
0183          CALL DPW (IPTR4,ADR,N1,50.)
0184      360  CONTINUE
0185      370  CONTINUE
0186      380  NESURF = IE-1
0187          IF (NESURF.LT.0) NESURF = 0
0188          IF (NSEC.NE.7) GO TO 470
-----
C-----
C      NEW ELEMENT WITH COMPLETE NODES DEFINED
C-----
0189      DO 450 IE=1,2560
0190      READ (5,520) NSEC,K,N1,NN,(NOD(I),I=1,NN),MM1
0191      IF (NSEC.NE.0) GO TO 460
0192      IF (NPRINT.GE.2) PRINT 550, K,ANJ(M5),NN,MM1,(NOD(I),I=1,NN)
0193      IF (N1.EQ.1) GO TO 390
0194      WRITE (2,*) 'ELEM#',K,'HAS ORDER HIGHER THAN IT IS NOT USED'
0195      GO TO 450
0196      390  NNT = NN
0197          NOELEM = K
0198          DO 440 I=1,NNT
0199          NP = NOD(I)
0200          IF (NP.EQ.0) GO TO 440
0201          ID = MOD(NP,1000)
0202          ID = IDENT(ID)
0203          IF (ID.NE.0.AND.ID.LT.NTY) GO TO 400
0204          ID = 1
0205      400  DO 410 J=ID,NTY
0206          IF (NP.NE.NUMBER(J)) GO TO 410
0207          NODNEW(I) = J
0208          NYES(J) = 1
0209          GO TO 420
0210      410  CONTINUE
0211          PRINT 580, NODNEW(I),K
0212          INNN = 1
0213          GO TO 440

```

ELEM,FTN

/TRIBLOCKS/WR

```

0214      420      IF (NHEAD.EQ.1) GO TO 440
0215              DO 430 KKK=1,NBPTC
0216              IF (J.NE.NOD8(KKK)) GO TO 430
0217              NHEAD = 1
0218              GO TO 440
0219      430      CONTINUE
0220      440      CONTINUE
0221              IF (NODMAX.LT.NNT) NODMAX = NNT
0222              NELEM = NELEM+1
0223              NENAME(NELEM) = K
0224              ADR = NELEM
0225              CALL UPW (IPTR4,ADR,N1,50,)
0226      450      CONTINUE
0227      460      IF (NPRINT.GE.2) PRINT *, 'ADDED 3=0 ELEMENTS=',IE-1
0228      470      IF (NPRINT.GE.2) PRINT 610
0229              J = 0
0230              DO 510 I=1,NNT
0231              IF (NYES(I).NE.1) GO TO 510
0232              IF (NBPTC.EQ.0) GO TO 490
0233              DO 480 K=1,NBPTC
0234              IF (NOD8(K).EQ.I) GO TO 510
0235      480      CONTINUE
0236      490      J = J+1
0237              IF (J.GT.KL) GO TO 500
0238              NODE(J) = I
0239              GO TO 510
0240      500      NSTOP = 1
0241      510      CONTINUE
0242              NPT = J
0243              IF (NPT.GT.KL) PRINT 690, NPT,KL
0244              IF (NPRINT.GT.1) PRINT 560, NPT,(NUMBER(NODE(I)),I=1,NPT)
0245              IF (NBPTC.EQ.0) PRINT 680
0246              CALL DPH (IPTR4,2561.,NENAME,2560,)
0247              RETURN
0248      520      FORMAT (I1,I4,15I5/16I5)
0249      530      FORMAT (16I5)
0250      540      FORMAT (/ 'FORCORNER',I3,' MIXNOD',I5,' IN ELEM',I5,' SEEMS TO B
+ E WRONG'/)
0251      550      FORMAT (' ELEM',I4,' ORDER ',A4,' MAT',I3,' TOTAL NODES',I3,
+ ' CORNER NODES',8I6)
0252      560      FORMAT (1H1//,120('=')//,5X,'TOTAL NUMBER OF UNKNOWN ARE=',I10,'
+THESE NODES ARE AS GIVEN BELOW',//,120('=')//200(15I8//))
0253      570      FORMAT (/)
0254      580      FORMAT (' NOTE ##### FOR NODE',I5,' IN ELEM',I5,' NO COORDINATE
+S ARE GIVEN'/)
0255      590      FORMAT (/,1X,120('=')// ' NEW SET BELOW ELEMENT #',I5,120('=')//)
0256      600      FORMAT (' MIXED ELEMENT-CORNER ORDER',8I10)
0257      610      FORMAT (1H1//)
0258      620      FORMAT (/,' AM AHA ## FOR NODE',I5,' ELEM',I5,' VERTICAL
+ ORDER IS',I3,' WHILE IT SHOULD BE 1'/)
0259      630      FORMAT (/,' MID QUADRATIC NODES',12I8//,' MID CURIC NODES ',12I
+8,120X,12I8)
0260      640      FORMAT (/ ' CUBIC NODES',8X,12I8/20X,12I8/)
0261      650      FORMAT (/ ' MID NODES',10X,12I8/)
0262      660      FORMAT (' COPY OF ELEM. DETAILS # NODE CARD',I3,I4,15I5/35X,16I5)
0263      670      FORMAT (' COPY OF CARD OF MIX-CORNER ORDER DETAILS ',16I5)

```

FORTRAN IV=PLUS V02-51D
ELEM,FTN /TRIRBLOCKS/HR

16103109 13-SEP-79

PAGE 6

```
0264      680      FORMAT (/120('#')//,10X,'SINCE NOT A SINGLE POTENTIAL B,C, GIVEN'  
+THE SOLUTION OF PROBLEM SHALL NOT BE UNIQUE,'//,10X,'CHECK THIS AS  
+PECT BEFORE RUNNING OTHER PROGRAMS ### IMPORTANT'//,120('#')//)  
0265      690      FORMAT (/5X,'N O T E = "NUMBER OF UNKNOWN ESTIMATED ARE=',15,16  
+X,'WHILE THE PRESCRIBED ARE=',15,' P R O G R A M S H A L L B E S  
+T O P P E D'//,16X,'CHANGE AS PER THE NEED PRINTED AT THE END'//)  
0266      END
```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODE1	010104 2002	RW,I,CON,LCL
SPDATA	000130 44	RW,D,CON,LCL
SIDATA	002100 544	RW,D,CON,LCL
SVARS	000076 31	RW,D,CON,LCL
STEMPS	000030 12	RW,D,CON,LCL
HDR	000274 44	RW,D,OVR,G8L
,SSSS,	020000 4096	RW,D,OVR,G8L
BLK1	013000 2016	RW,D,OVR,G8L
ELEM	000164 50	RW,D,OVR,G8L
EL	023730 5100	RW,D,OVR,G8L

TOTAL SPACE ALLOCATED = 072072 14877

,LP1=ELEM

9.2 PROG2, PROGRAM LISTINGS

FORTRAN IV-PLUS V02-510 16103158 13-SEP-79 PAGE 1
 PROG2.FTN /TRIBLOCKS/WR

```

C      **** PROG2.FTN ****
C-----
C      SUB-PROGRAM TWO FOR ESTIMATION OF THE GAUSSIAN INTEGRATION PARAMET
C-----
0001      BYTE MCR(80)
0002      COMMON /XYZ/X(256),Y(256),Z(256)
0003      INCLUDE '[311,3]HQR.FTN'
0004      * COMMON/HQR/LNPT,NTT,NBPTC,NBPTC,NSTREN,NPT,NSELEM,NELEM,NODMAX,
      * #NONZER,NOD,MATN,KTYPE,NMAX,NSTIF,NTIME,NSTORE,NDUM,TITLE(20),
      * #DP1,DP2,DP3,DP4,DP5,DP6,DP7,DP8,DP9,DP10,DP11,DP12,DP13,DP14,DP15,
      * #DP16,DP17,DP18,NTSURF,NESURF
0005      * COMMON /ELEM0/N1,NN,NOD(44),MM1,NQW,NE,NHEAD,MIXNOD(8),          Q
      * + W(10),XQW(10),YQW(10),ZQW(10)
0006      * COMMON /ELEM1/NNN,NODNEW(20),MM1N,NQWN,SUMQW,NEN,NHEADN,        X
      * + IJCOB(20),XIDLEM(20),FIFJ(400),GIGJ(400),
      * + HIHJ(400),XIXJ(400),IDUM,AREA
0007      * COMMON/TRACE/NPROG1,NPROG2,NBAND,NPROG3,NPROG3,NSUB3
0008      * DIMENSION
      * + XQ(44),YQ(44),ZQ(44),F(44),G(44),H(44),XN(44),AA1(44),AA2(44),
      * + AA3(44),SCA(44),ETA(44),EZA(44),NSURFA(44),XN2(44),XIDLE(44),
      * + SS(3),MCC(3),AA(3),DD(3),BETAD(3),GQWT(5),GQPO(5),NUMBER(2560)
C-----
0009      DATA SCA /
      * + -1., 1., 1., -1., -1., 1., 1., -1., 0., 0., 0., 0.,
      * + -1., 1., -1., 1., -1., 1., 1., -1.,
      * + -.333333333333, .333333333333, -.333333333333, .333333333333,
      * + -.333333333333, .333333333333, -.333333333333, .333333333333,
      * + -1., -1., 1., 1., 1., -1., 1., 1., -1., -1., 1., 1.,
      * + -1., -1., 1., 1., 1., -1., -1., -1., -1./
C-----
0010      DATA ETA/
      * + -1., -1., 1., 1., -1., -1., 1., 1., -1., 1., -1., 1.,
      * + 0., 0., 0., 0., -1., -1., 1., 1., -1., -1., 1., 1.,
      * + -1., -1., 1., 1.,
      * + -.333333333333, .333333333333, -.333333333333, .333333333333,
      * + -.333333333333, .333333333333, -.333333333333, .333333333333,
      * + -1., -1., -1., -1., 1., 1., 1., 1./
C-----
0011      DATA EZA/
      * + 1., 1., 1., 1., -1., -1., -1., -1., 1., 1., -1., -1.,
      * + 1., 1., -1., -1., 0., 0., 0., 0., 1., 1., 1., 1.,
      * + -1., -1., -1., -1., 1., 1., 1., 1., -1., -1., -1., -1.,
      * + .333333333333, -.333333333333, .333333333333, -.333333333333,
      * + .333333333333, -.333333333333, .333333333333, -.333333333333 /
C-----
0012      CALL GETMCR (MCR,ISW)
0013      IF (ISW.GT.79) ISW = -40
0014      IF (ISW.LT.0) GO TO 110
0015      DO 100 I=ISW+1,80
0016      100 MCR(I) = 0
0017      110 IF (ISW.GT.1) ISW = 1
0018      IF (ISW.LT.0) ISW = -1
0019      IF (ISW.NE.1) GO TO 130
0020      IF (MCR(5).EQ.' 'OR.MCR(5).EQ.0) ISW = 2
0021      IF (ISW.NE.1) GO TO 130
0022      DECODE (38,120,MCR(5))IUN,NPRINT

```

```

0023 120 FORMAT (215,7A4)
0024 130 IF (ISW.EQ.1) GO TO 140
0025 WRITE (2,*) 'ENTER FILE-Q DISK UNIT'
0026 READ (2,*) IUN
0027 140 IF (IUN.NE.3) CALL ASNLUN (1,'DP',IUN,ISTAT)
0028 IF (ISW.EQ.1) GO TO 150
0029 WRITE (2,*) 'ENTER PRINT OPTION (0,1,2)'
0030 READ (2,*) NPRINT
0031 150 CALL OPFILE (1,'TRACE',1.,IPTR10)
0032 CALL DPR (IPTR10,0.,NPROG1,6.)
0033 IF (NPROG1.EQ.1) GO TO 160
0034 WRITE (2,*) 'PROG1 NOT COMPLETED THEREFORE PROG2 STOPPED'
0035 STOP
0036 160 CALL OPFILE (1,'DESCRIP',25.,IPTR1)
0037 CALL DPR (IPTR1,0.,LNPT,94.)
0038 NPROG2 = 0
0039 CALL DPR (IPTR1,1.,NUMBER,2560.)
0040 CALL DPH (IPTR10,0.,NPROG1,6.)
0041 CALL OPFILE (1,'XYZCOR',UP3,IPTR3)
0042 CALL OPFILE (1,'ELEMENT0',OP4,IPTR4)
0043 CALL OPFILE (1,'ELEMENT1',OP11,IPTR11)
0044 IF (NPRINT.NE.0) CALL PTIME (6)
0045 NMAX1 = 256
0046 NODM1 = 44
0047 NONZ1 = 20
0048 NGW1 = 10
0049 NSTOP = 0
0050 IF (NMAX.LE.NMAX1) GO TO 170
0051 PRINT 230
0052 NSTOP = 1
0053 170 NMAX = NMAX1
0054 IF (NODMAX.LE.NODM1) GO TO 180
0055 PRINT 240
0056 NSTOP = 1
0057 180 NODMAX = NODM1
0058 IF (NONZER.LE.NONZ1) GO TO 190
0059 PRINT 250
0060 NSTOP = 1
0061 190 NONZER = NONZ1
0062 IF (NGW.LE.NGW1) GO TO 200
0063 PRINT 260
0064 NSTOP = 1
0065 200 NGW = NGW1
0066 IF (NSTOP.EQ.0) GO TO 210
0067 PRINT 270
0068 STOP
0069 210 CONTINUE
0070 230 FORMAT (//10X,'INCREASE THE DIMENSION',//15X,'OF PARAMETERS RELATED
+//,20X,'TO TOTAL NODES'//)
0071 240 FORMAT (//10X,'INCREASE THE DIMENSION',//15X,'OF PARAMETERS RELATED
+//,20X,'ELEMENT NODES'//)
0072 250 FORMAT (//5X,'INCREASE DIMENSIONS',//10X,'OF PARAMETERS ASSOCIATED'
+//,15X,'WITH NONZERO VALUES OF ELEMENT'//)
0073 260 FORMAT (//5X,'INCREASE THE DIMENSION',//15X,'OF PARAMETERTS',//25X,'
+RELATED TO NUMNER OF',//30X,'SOURCE OR SINKS IN ELEM.'//)
0074 270 FORMAT (//10X,'P R O G R A M S T O P P E D BECAUSE OF ABOVE')

```

```

0075      NRR = 0
0076      THIR0 = 1./3
0077      THIR02 = 2./3
0078      R1924 = 19./24
0079      R964 = 9./64.
0080      DO 730 IX=1,NELEM
0081      AREA = 0.0
0082      VOL = 0.0
0083      ADR = IX
0084      CALL DPR (IPTR4,ADR,N1,138.)
0085      IF (NPRINT,NE,0) PRINT 280, NE,NN,(NUMBER(NOD(I)),I+1,NN)
0086      280  FORMAT (' NE',I3,' NN',I3,' NOD',20I5/10X,20I5/10X,20I5)
0087      MM1N = MM1
0088      NGWN = NGW
0089      NEN = NE
0090      NHEADN = NHEAD
0091      IF (NN,GT,NODMAX) WRITE (2,*) 'NE',NE,' NN',NN,'WHICH EXCEEDS NODM
+AX'
0092      NO = 2
0093      IF (NN,GT,0) NO = 3
0094      GQPO(1) = 0.0
0095      GQPO(2) = 0.774596669241483
0096      GQPO(3) = -GQPO(2)
0097      GQWT(1) = 0.000000000000009
0098      GQWT(2) = 0.555555555555556
0099      GQWT(3) = GQWT(2)
0100      IF (NO=3) 290,310,300
0101      290  GQWT(1) = 1.0
0102      GQWT(2) = 1.0
0103      GQPO(1) = 0.577350269189626
0104      GQPO(2) = -0.577350269189626
0105      GO TO 310
0106      300  GQPO(2) = 0.906179845938664
0107      GQPO(3) = 0.538469310105683
0108      GQPO(4) = -GQPO(3)
0109      GQPO(5) = -GQPO(2)
0110      GQWT(1) = 0.368888888888889
0111      GQWT(2) = 0.236926885056189
0112      GQWT(3) = 0.478628670499366
0113      GQWT(4) = GQWT(3)
0114      GQWT(5) = GQWT(2)
0115      310  CONTINUE
0116      JJ = 0
0117      DO 350 J=1,NN
0118      K = NOD(J)
0119      IF (K,EQ,0) GO TO 350
0120      ADD = 0.0
0121      NR = (K-1)/NMAX+1
0122      IF (NR,EQ,NRR) GO TO 320
0123      ADR = (NR-1)*2.
0124      CALL DPR (IPTR3,ADR,Z,512.)
0125      CALL DPR (IPTR3,ADR+20.,X,512.)
0126      CALL DPR (IPTR3,ADR+40.,Y,512.)
0127      320  KK = K-(NR-1)*NMAX
0128      NRR = NR
0129      XX = X(KK)

```

```

0130      YY = Y(KK)
0131      ZZ = Z(KK)
0132      IF (J,EQ,1) GO TO 340
0133      J1 = J-1
0134      DO 330 I=1,J1
0135      KK = NOD(I)
0136      IF (KK,EQ,0) GO TO 330
0137      IF (KK,NE,K) GO TO 330
0138      ADD = -0.25
0139      GO TO 340
0140 330    CONTINUE
0141 340    JJ = JJ+1
0142      XQ(J) = XX
0143      YQ(J) = YY
0144      ZQ(J) = ZZ+ADD
0145      NODNEW(JJ) = K
0146 350    CONTINUE
C-----
C      CHECK FOR ANTICLOCK WISE NODE DIRECTION
C-----
0147      A1 = XQ(1)*(YQ(2)-YQ(4))+XQ(2)*(YQ(4)-YQ(1))+XQ(4)*(YQ(1)-YQ(2))
0148      A2 = XQ(2)*(YQ(3)-YQ(4))+XQ(3)*(YQ(4)-YQ(2))+XQ(4)*(YQ(2)-YQ(3))
0149      IF (A1+A2,GT,0) GO TO 360
0150      WRITE (2,*) 'ERROR****ERRORINELEM',NE,'NODESEQUENCE'
0151      WRITE (6,*) 'ERROR****ERRORINELEM',NE,'NODESEQUENCE'
0152      NSTOP = 1
0153 360    AREA1 = 0.0
0154      DO 370 I=1,4
0155      II = I+1
0156      IF (II,EQ,5) II = 1
0157 370    AREA1 = AREA1+YQ(I)*(XQ(II)-XQ(I))-XQ(I)*(YQ(II)-YQ(I))
0158      IF (AREA1,LE,0.0) GO TO 380
0159      WRITE (2,*) 'ERRORERROR*****NODESEQUENCEWRONGINELEM',NE
0160      WRITE (6,*) 'ERRORERROR*****NODESEQUENCEWRONGINELEM',NE
0161      NSTOP = 1
0162 380    NNN = JJ
0163      DO 390 I=1,NNN
0164      XIDLE(I) = 0.0
0165      XIDLEM(I) = 0.0
0166      NSURFA(I) = 0
0167 390    XIJCOB(I) = 0.0
0168      NXX = NNN*NNN
0169      DO 400 I=1,NXX
0170      FIFJ(I) = 0.0
0171      GIGJ(I) = 0.0
0172      HIMJ(I) = 0.0
0173      XIXJOB(I) = 0.0
0174 400    CONTINUE
0175      DO 410 I=1,3
0176 410    AA(I) = 1.0
0177      DO 600 L=1,NO
0178      EZ = GQPO(L)
0179      WW = GQWT(L)
0180      DO 420 I=1,NN
0181 420    NSURFA(I) = 0
0182      DO 600 N=1,NO

```

```

0183      SC = GQPO(N)
0184      WSC = GQWT(N)*WW
0185      DO 600 M=1,NO
0186      ET = GQPO(M)
0187      W = WSC*GQWT(M)
0188      D11 = 0.0
0189      D12 = 0.0
0190      D13 = 0.0
0191      D21 = 0.0
0192      D22 = 0.0
0193      D23 = 0.0
0194      D31 = 0.0
0195      D32 = 0.0
0196      D33 = 0.0
0197      DXSC = 0.0
0198      DYSC = 0.0
0199      DZSC = 0.0
0200      OXET = 0.0
0201      DYET = 0.0
0202      DZET = 0.0
0203      SS(1) = SC
0204      SS(2) = ET
0205      SS(3) = EZ
0206      DO 600 I=1,NN
0207      IF (NOD(I).EQ.0) GO TO 600
0208      NSURF = 0
0209      IF (L.NE.1.OR.NE.GT.999) GO TO 440
0210      IF (M.NE.1.OR.N.NE.1) GO TO 430
0211      IF (I.LE.4) NSURF = 1
0212      IF (I.EQ.9.OR.I.EQ.10.OR.I.EQ.13.OR.I.EQ.14) NSURF = 1
0213      IF (I.GE.21.AND.I.LE.24) NSURF = 1
0214      IF (I.GE.29.AND.I.LE.32) NSURF = 1
0215      NSURFA(I) = NSURF
0216      430 NSURF = NSURFA(I)
0217      440 DO(1) = SCA(I)
0218      DO(2) = ETA(I)
0219      DO(3) = EZA(I)
0220      DO 450 LT=1,3
0221      450 AA(LT) = 1+DO(LT)*SS(LT)
0222      IF (I.GT.8) GO TO 520
0223      ALP = .125
0224      BETA = 0.0
0225      SBETA = 0.0
0226      ALPHA = .125*AA(1)*AA(2)*AA(3)
0227      IF (NSURF.EQ.1) ALPHA2 = 0.25*(1+DO(1)*SS(1))*(1+DO(2)*SS(2))
0228      MMM = N1
0229      IF (N1.NE.0) GO TO 460
0230      MCODE = MIXNOD(I)
0231      MCC(1) = MCODE/100
0232      MCC(2) = MCODE/10-MCC(1)*10
0233      MCC(3) = MCODE-MCC(1)*100-MCC(2)*10
0234      460 DO 510 LT=1,3
0235      D1 = DO(LT)
0236      S1 = SS(LT)
0237      IF (N1.NE.0) GO TO 470
0238      MMM = MCC(LT)

```

PROG2.FTN

/TRIRLOCKS/WR

```

0239      470      GO TO (480,490,500), MMH
0240      480      BETA = BETA+THIRD
0241              BETAD(LT) = 0.0
0242              IF (LT.EQ.3) GO TO 510
0243              IF (NSURF.EQ.0) GO TO 510
0244              SBETA = SBETA+0.5
0245              GO TO 510
0246      490      BETA = BETA+01*31*THIRD2
0247              BETAD(LT) = 01
0248              IF (LT.EQ.3) GO TO 510
0249              IF (NSURF.EQ.0) GO TO 510
0250              SBETA = SBETA+01*81*0.5
0251              GO TO 510
0252      500      BETA = BETA+1.125*31*31-R1924
0253              BETAD(LT) = 2.25*81
0254              IF (LT.EQ.3) GO TO 510
0255              IF (NSURF.EQ.0) GO TO 510
0256              SBETA = SBETA+1.125*31*81-5./8.
0257      510      CONTINUE
0258              XN(I) = ALPHA*BETA
0259              DVSC = ALP*DD(1)*AA(2)*AA(3)*BETA+ALPHA*BETAD(1)
0260              DVET = ALP*DD(2)*AA(1)*AA(3)*BETA+ALPHA*BETAD(2)
0261              DVEZ = ALP*DD(3)*AA(1)*AA(2)*BETA+ALPHA*BETAD(3)
0262              IF (NSURF.EQ.0) GO TO 590
0263              XN2(I) = ALPHA2*SBETA
0264              DDVSC = 0.25*AA(2)*DD(1)*SBETA+0.25*AA(2)*AA(1)*BETAD(1)
0265              DDVET = 0.25*AA(1)*(DD(2)*SBETA+AA(2)*BETAD(2))
0266              GO TO 590
0267      520      IF (I.GT.20) GO TO 550
0268              IF (I.GT.12) GO TO 530
0269              AAA = (1-8C*3C)*0.25
0270              XN(I) = AAA*AA(2)*AA(3)
0271              DVSC = -.5*3C*AA(2)*AA(3)
0272              DVET = DD(2)*AAA*AA(3)
0273              DVEZ = DD(3)*AAA*AA(2)
0274              IF (NSURF.EQ.0) GO TO 590
0275              GO TO 580
0276      530      IF (I.GT.16) GO TO 540
0277              AAA = (1-ET*ET)*0.25
0278              XN(I) = AA(1)*AAA*AA(3)
0279              DVSC = DD(1)*AAA*AA(3)
0280              DVET = AA(1)*(-.5*ET)*AA(3)
0281              DVEZ = AA(1)*AAA*DD(3)
0282              IF (NSURF.EQ.0) GO TO 590
0283              GO TO 580
0284      540      AAA = (1-EZ*EZ)*0.25
0285              XN(I) = AA(1)*AA(2)*AAA
0286              DVSC = DD(1)*AA(2)*AAA
0287              DVET = AA(1)*DD(2)*AAA
0288              DVEZ = AA(1)*AA(2)*(-.5*EZ)
0289              GO TO 590
0290      550      IF (I.GT.36) GO TO 570
0291              IF (I.GT.28) GO TO 560
0292              AAA = R964*(1-3C*3C)*(1+9*DD(1)*3C)
0293              XN(I) = AAA*AA(2)*AA(3)
0294              DVSC = R964*(-2*3C+9*DD(1)*(1-3*3C*3C))*AA(2)*AA(3)

```

```

0295      DVET = AAA*DD(2)*AA(3)
0296      DVEZ = AAA*AA(2)*DD(3)
0297      IF (NSURF,EQ,0) GO TO 590
0298      GO TO 580
0299      560    AAA = R964*(1-ET*ET)*(1+9*DD(2)*ET)
0300      XN(I) = AA(1)*AAA*AA(3)
0301      DVSC = DD(1)*AAA*AA(3)
0302      DVEY = AA(1)*R964*(-2*ET+9*DD(2)*(1-3*ET*ET))*AA(3)
0303      DVEZ = AA(1)*AAA*DD(3)
0304      IF (NSURF,EQ,0) GO TO 590
0305      GO TO 580
0306      570    AAA = R964*(1-EZ*EZ)*(1+9*DD(3)*EZ)
0307      XN(I) = AA(1)*AA(2)*AAA
0308      DVSC = DD(1)*AA(2)*AAA
0309      DVEY = AA(1)*DD(2)*AAA
0310      DVEZ = AA(1)*AA(2)*R964*(-2*EZ+9*DD(3)*(1-3*EZ*EZ))
0311      GO TO 590
0312      580    DDVSC = DVSC*2./AA(3)
0313      DDVET = DVET*2./AA(3)
0314      XN2(I) = XN(I)*2./AA(3)
0315      590    AA1(I) = DVSC
0316      AA2(I) = DVEY
0317      AA3(I) = DVEZ
0318      XQQ = XQ(I)
0319      YQQ = YQ(I)
0320      ZQQ = ZQ(I)
0321      D11 = DVSC*XQQ+D11
0322      D21 = DVEY*XQQ+D21
0323      D31 = DVEZ*XQQ+D31
0324      D12 = DVSC*YQQ+D12
0325      D22 = DVEY*YQQ+D22
0326      D32 = DVEZ*YQQ+D32
0327      D13 = DVSC*ZQQ+D13
0328      D23 = DVEY*ZQQ+D23
0329      D33 = DVEZ*ZQQ+D33
0330      IF (NE,GT,1000) GO TO 600
0331      IF (NSURF,EQ,0) GO TO 600
0332      DXSC = DXSC+DDVSC*XQQ
0333      DYSC = DYSC+DDVSC*YQQ
0334      DZSC = DZSC+DDVSC*ZQQ
0335      DXET = DXET+DDVET*XQQ
0336      DYEY = DYEY+DDVET*YQQ
0337      DZET = DZET+DDVET*ZQQ
0338      600    CONTINUE
0339      DU = D11*D22*D33+D12*D23*D31+D13*D21*D32-D13*D22*D31-D21*D12*D33-D
        +11*D32*D23
0340      IF (NE,GT,999.OR,L,NE,1) GO TO 610
0341      XYJACO = DXSC*DVEY-DXET*DYSC
0342      XZJACO = DXSC*DZET-DXET*DZSC
0343      YZJACO = DYSC*DZET-DYEY*DZSC
0344      DLEMMA = SURT(XYJACO*XYJACO+XZJACO*XZJACO+YZJACO*YZJACO)
0345      DLEMMA = DLEMMA*GQWT(N)*GQWT(M)
0346      AREA = AREA+DLEMMA
0347      610    IF (DU,EQ,0.0) GO TO 620
0348      DU = 1.0/DU
0349      GO TO 630

```

```

0350 620 PRINT 750, NE, SC, ET, EZ, (MOD(I), XQ(I), YQ(I), ZQ(I), I=1, NN)
0351 GO TO 680
0352 630 DO 640 I=1, NN
0353 IF (MOD(I), EQ, 0) GO TO 640
0354 F(I) = DU*(AA1(I)*D22+D33+AA3(I)*D12+D23+AA2(I)*D32+D13-AA3(I)*D22
+*D13-AA2(I)*D33+D12-AA1(I)*D32+D23)
0355 G(I) = DU*(AA2(I)*D11+D33+AA1(I)*D23+D31+AA3(I)*D21+D13-AA2(I)*D31
+*D13-AA1(I)*D21+D33+D11-AA3(I)*D23)
0356 H(I) = DU*(AA3(I)*D11+D22+D12+AA2(I)*D31+D21+D32+AA1(I)*AA1(I)*D22
+*D31+D12+D21+AA3(I)*D11+D32+AA2(I))
0357 640 CONTINUE
0358 XJCOB = 1.0/DU
0359 DU = XJCOB*H
0360 II = 0
0361 DO 670 I=1, NN
0362 IF (MOD(I), EQ, 0) GO TO 670
0363 II = II+1
0364 FI = F(I)*DU
0365 GI = G(I)*DU
0366 HI = H(I)*DU
0367 XNI = XN(I)
0368 XNIDU = XNI*DU
0369 JJ = 0
0370 NSURF = NSURFA(I)
0371 IF (NSURF, EQ, 0) GO TO 650
0372 XIULEM(II) = XIULEM(II)+XN2(I)*OLEMDA
0373 XIDN = XN2(I)*OLEMDA
0374 650 DO 660 J=1, NN
0375 IF (MOD(J), EQ, 0) GO TO 660
0376 JJ = JJ+1
0377 FJ = F(J)
0378 GJ = G(J)
0379 HJ = H(J)
0380 XNJ = XN(J)
0381 XIXJ = XNJ*XNIDU
0382 IIJJ = (II-1)*NNN+JJ
0383 XIXJOB(IIJJ) = XIXJOB(IIJJ)+XIXJ
0384 FIFJ(IIJJ) = FIFJ(IIJJ)+FI*FJ
0385 GIGJ(IIJJ) = GIGJ(IIJJ)+GI*GJ
0386 MIHJ(IIJJ) = MIHJ(IIJJ)+HI*HJ
0387 IF (NSURF, EQ, 0, OR, NSURFA(J), EQ, 0) GO TO 660
0388 XIDLE(II) = XIDLE(II)+XIDN*XN2(J)
0389 660 CONTINUE
0390 XIJCOB(II) = XIJCOB(II)+XNIDU
0391 670 CONTINUE
0392 IF (DU, LT, 0.) WRITE (2, *) 'ERROR IN ELEM', NE, 'NODE SEQUENCE OR COO
+*, 'RDINATES, JACOBIAN LESS THAN 0.'
0393 VOL = VOL+DU
0394 680 CONTINUE
0395 DO 690 I=1, NNN
0396 690 XIJCOB(I) = XIJCOB(I)/VOL
0397 IF (AREA, GT, 0.0, OR, NE, GT, 999) GO TO 700
0398 WRITE (2, *) 'ELEM', NE, 'AREA<0.0CHECKNODES-SHOULDBEANTICLOCK'
0399 WRITE (6, *) 'ELEM', NE, 'AREA<0.0CHECKNODES-SHOULDBEANTICLOCK'
0400 700 IF (NE, LT, 999, AND, NPRINT, GE, 1) PRINT 760, NE, AREA, VOL
0401 IF (NE, EQ, 1, AND, NPRINT, NE, 0) PRINT 710, NE, (I, XIJCOB(I), I=1, NNN)

```



```

0402      IF (NPRINT.EQ.2) PRINT *, VOL,(XIJCOB(I),I=1,NNN)
0403      710  FORMAT (' XIJCOB FOR ELEMENT',I4/,9(I3,G10.2)/3(9(I3,G10.2)/))
0404      ADR1 = (IX-1)*13
0405      CALL DPH (IPTR11,ADR1,NNN,3310.)
0406      IF (IX.NE.1.OR,NPRINT.NE.3) GO TO 730
0407      PRINT 770, (XIJC0B(I),I=1,NNN)
0408      IF (NE.LT.1000) PRINT 000, (XIDLEM(I),I=1,NNN)
0409      IJ = 1
0410      IJJ = 0
0411      DO 720 I=1,NNN
0412      IJJ = IJJ+NNN
0413      PRINT 780, I,(FIFJ(IJ),GIGJ(IJ),HIMJ(IJ),II=IJ,IJJ)
0414      PRINT 790, I,(XIXJOB(IJ),II=IJ,IJJ)
0415      720  IJ = IJJ+NNN
0416      730  CONTINUE
0417      IF (NSTOP.EQ.0) NPROG2 = 1
0418      CALL DPH (IPTR10,0.,NPROG1,6.)
0419      IF (NSTOP.EQ.0) GO TO 740
0420      WRITE (2,*) 'PROG2 NOT COMPLETE DUE TO NODE SEQUENTIAL ERROR IN',
      * ' SOME ELEMENTS'
0421      WRITE (2,*) 'REMOVE ERROR * RERUN PROG1 BEFORE PROCEEDING AHEAD'
0422      740  IF (NPRINT.NE.0) CALL PTIME (0)
0423      STOP
0424      750  FORMAT (/ ' DU=0.0 FOR ELM # ',I5, ' SC= ',F10.5, ' ET= ',F10.5, ' E
      +Z= ',F10.5,/32(' NODE',I5, ' X = ',E13.7, ' Y = ',E13.7, ' Z = ',
      +E13.7//))
0425      760  FORMAT (' ESTIMATION OF GAUSSIAN INTEGRATION HAS BEEN DONE FOR ELE
      +M#',I5,10X,F15.0, ' *AREA AND VOL = ',G10.3)
0426      770  FORMAT (' XIJCOB OF ELEMENT#1',/50(12E10.3/))
0427      780  FORMAT (' FIFJ,GIGJ,HIMJ OF ELEM#1 NODE#',I3/,/50(12E10.3/))
0428      790  FORMAT (' XIXJOB OF ELEM.#1 NODE#',I3/,/50(12E10.3/))
0429      800  FORMAT (' XIDLEMDA FOR ELEM#1 ARE',/50(12E10.3/))
0430      END

```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCONE1	014646 3253	RW,I,CON,LCL
SPDATA	001074 206	RW,D,CON,LCL
SIDATA	001750 500	RW,D,CON,LCL
SVARS	020260 4104	RW,D,CON,LCL
STEMP3	000044 10	RW,D,CON,LCL
XYZ	006000 1536	RW,D,OVR,GBL
HDR	000274 94	RW,D,OVR,GBL
ELEM0	000424 130	RW,D,OVR,GBL
ELEM1	014734 3310	RW,D,OVR,GBL
TRACE	000014 6	RW,D,OVR,GBL

TOTAL SPACE ALLOCATED = 064126 13355

,LP1=PROG2

9.3 BAND, PROGRAM LISTINGS

FORTRAN IV-PLUS V02-310
BAND.FTN /TRIBLOCKS/WR

13-SEP-79

PAGE 1

```

C      *****RAND.FTN *****
C-----
C      PROGRAM TO FORM THE LOCATION OF ELEMENTS IN INITIAL SYSTEM MATRIX
C-----
0001      BYTE MCR(80)
0002      INCLUDE 'HOR.FTN'
0003      *      COMMON/HOR/LNPT,NTT,NBPTC,NBFPTC,NSTREH,NPT,NSELEH,NELEH,NOOMAX,
*      *      #NONZER,NQD,MATN,KTYPE,NMAX,NSTIF,NTIME,NSTORE,NQUM,TITLE(20),
*      *      #OP1,DP2,DP3,DP4,DP5,DP6,DP7,DP8,DP9,DP10,DP11,DP12,DP13,DP14,DP15,
*      *      #DP16,DP17,DP18,NTSURF,NESURF
0004      COMMON/ELEM/N1,NN,NQD(40),MM1,NQW,NE,NH
0005      COMMON/TRACE/NPROG1,NPROG2,NBAND,NPROG3,NSUB3
0006      DIMENSION NOOE(2048),NCOL(128)
*      + ,IRANDW(2048),NSTART(2048),NSTOP(2048)
0007      CALL GETMCR (MCR,ISW)
0008      IF (ISW.GT.79) ISW = -40
0009      IF (ISW.LT.0) GO TO 110
0010      DO 100 I=ISW+1,80
0011      100   MCR(I) = ' '
0012      110   IF (ISW.GT.1) ISW = 1
0013      IF (ISW.LT.0) ISW = -1
0014      IF (ISW.NE.1) GO TO 130
0015      IF (MCR(5).EQ.' ' .OR. MCR(5).EQ.0) ISW = 2
0016      IF (ISW.NE.1) GO TO 130
0017      DECODE (30,120,MCR(5)) IUN,NPRINT
0018      120   FORMAT (2I5,7A4)
0019      130   IF (ISW.EQ.1) GO TO 140
0020      WRITE (2,*) 'ENTER FILE=0 DISK UNIT'
0021      READ (2,*) IUN
0022      140   CALL ASNLUN (1,'DP',IUN,ISTAT)
0023      IF (ISW.EQ.1) GO TO 150
0024      WRITE (2,*) 'ENTER PRINT OPTION(0,1,2)'
0025      READ (2,*) NPRINT
0026      150   CALL DPFILE (1,'TRACE',1.,IPTR10)
0027      CALL DPR (IPTR10,0.,NPROG1,6.)
0028      IF (NPROG1.EQ.1) GO TO 160
0029      WRITE (2,*) 'PROG1 NOT COMPLETED THEREFORE BAND STOPPED'
0030      STOP
0031      160   CALL DPFILE (1,'DESCRIP',25.,IPTR1)
0032      CALL DPR (IPTR1,0.,LNPT,94.)
0033      NBAND = 0
0034      CALL DPW (IPTR10,0.,NPROG1,6.)
0035      IF (NPRINT.EQ.0) GO TO 165
0036      CALL PTIME (6)
0037      PRINT 400, NTSURF, LNPT, NTT, NBPTC, NBFPTC, NSTREH, NPT, NSTIF
0038      PRINT 410, NESURF, NSELEH, NELEH, NOOMAX, NONZER, MATN
0039      165   CALL DPFILE (1,'NCOL',DP10,IPTR10)
0040      CALL DPR (IPTR1,17.,NOOE,2048.)
0041      CALL DPFILE (1,'ELEMENT0',DP4,IPTR4)
0042      KS = 128
0043      NSTIF = 0
0044      DO 170 I=1,NPT
0045      ADR = I+2-1
0046      CALL DPW (IPTR10,ADR,NCOL,128.)
0047      170   NSTART(I) = 10000
0048      DO 300 IE=1,NELEH

```

```

0049      ADR = IE
0050      CALL DPR (IPTR4,ADR,N1,50.)
0051      IF (NE.LT,1000.AND,NPRINT,NE,0) PRINT 300, NE
0052      DO 290 K=1,NN
0053      KM = NOD(K)
0054      IF (KM.EQ,0) GO TO 290
0055      DO 180 I=1,NPT
0056      IF (KM.EQ,NODE(I)) GO TO 190
0057      180  CONTINUE
0058      GO TO 290
0059      190  KK = I
0060      ADR = KK*2-1
0061      CALL DPR (IPTR10,ADR,NCOL,128.)
0062      DO 280 L=1,NN
0063      JL = NOD(L)
0064      IF (JL.EQ,0) GO TO 280
0065      DO 200 I=1,NPT
0066      IF (JL.EQ,NODE(I)) GO TO 210
0067      200  CONTINUE
0068      GO TO 280
0069      210  JJ = I
0070      IF (NSTART(JJ).GT,KK) NSTART(JJ) = KK
0071      IF (NSTOP(JJ).LT,KK) NSTOP(JJ) = KK
0072      IF (NCOL(1).NE,0) GO TO 220
0073      NCOL(1) = JJ
0074      GO TO 280
0075      220  DO 240 LL=1,KS+20
0076      IF (LL.LE,128) GO TO 230
0077      WRITE (2,*) 'IN DO LOOP 240 LL>128',LL
0078      PRINT *, 'IN DO LOOP 240 LL > 128',LL
0079      230  IF (NCOL(LL).EQ,0) GO TO 270
0080      IF (NCOL(LL).LT,JJ) GO TO 240
0081      IF (NCOL(LL).NE,JJ) GO TO 250
0082      GO TO 280
0083      240  CONTINUE
0084      250  LAST = NCOL(LL)
0085      NFM = KS-1
0086      DO 260 LLL=LL,NFM
0087      LI = LLL+1
0088      NEXT = NCOL(LI)
0089      NCOL(LI) = LAST
0090      IF (NEXT.EQ,0) GO TO 270
0091      260  LAST = NEXT
0092      270  NCOL(LL) = JJ
0093      280  CONTINUE
0094      CALL DPR (IPTR10,ADR,NCOL,128.)
0095      290  CONTINUE
0096      300  CONTINUE
0097      DO 340 I=1,NPT
0098      ADR = I*2-1
0099      CALL DPR (IPTR10,ADR,NCOL,128.)
0100      DO 310 J=1,KS
0101      IF (NCOL(J).EQ,0) GO TO 320
0102      310  CONTINUE
0103      320  JJ = J-1
0104      I*HANDW(1) = JJ

```

```

0105      IF (NPRINT.GE.2) PRINT 330, I,JJ,(NCOL(J),J=1,JJ)
0106      330  FORMAT (' ROW',I5,' BAND',I5,' NCOL',20I5/6(20X,20I5/))
0107      IF (NSTIF.LT.JJ) NSTIF = JJ
0108      340  CONTINUE
0109      IF (NPRINT.EQ.0) GO TO 350
0110      PRINT 400, NTSURF, LNPT, NTT, NBPTC, NBFPTC, NSTREM, NPT, NSTIF
0111      PRINT 410, NESURF, NSELEM, NELEM, NODMAX, NONZER, MATN
0112      350  CALL DPW (IPTR1,0,,LNPT,92.)
0113      ADR = NPT*2.+1.
0114      CALL DPW (IPTR10,ADR,IBANDW,2040,)
0115      ADR = ADR+8.
0116      CALL DPW (IPTR10,ADR,NSTART,2040,)
0117      ADR = ADR+8.
0118      CALL DPW (IPTR10,ADR,NSTOP,2040,)
0119      IF (NPRINT.EQ.0) GO TO 370
0120      DO 360 I=1,NPT
0121      360  PRINT 390, I,IBANDW(I),NSTART(I),NSTOP(I)
0122      370  NBAND = 1
0123      CALL DPW (IPTR10,0,,NPRUG1,6.)
0124      IF (NPRINT.NE.0) CALL PTIME (6)
0125      380  FORMAT (5X,'ELEMENT # ',I5)
0126      390  FORMAT (4I10)
0127      400  FORMAT(/120(' ')/,
+ 20X,'TOTAL NUMBER OF SURFACE NODES(NTSURF)=',T80,I10/,
+ 20X,'MAXIMUM NUMBER ASSIGNED TO SURFACE NODES(LNPT)=',T80,I10/,
+ 20X,'TOTAL NODES IN THE SYSTEM (NTT)=',T80,I10//,
+ 12X,'THE FOLLOWING PARAMETERS ARE RELATED TO B.C'//,
+ 20X,'TOTAL NUMBER OF POTENTIAL B.C. PRESCRIBED (NBPTC)=',T80,I10/,
+ 20X,'TOTAL NUMBER OF FLUX B.C. PRESCRIBED (NBFPTC)=',T80,I10/,
+ 20X,'TOTAL STREAMS B.C. PRESCRIBED (NSTREM)=',T80,I10/,
+/,12X,'THE FOLLOWING TWO PARAMETERS ARE RELATED TO STIFFNESS MATRI
+X'//,20X,'TOTAL NUMBER OF UNKNOWN(NPT)=',T80,I10/,
+ 20X,'MAXIMUM NUMBER OF NON-ZERO COEFFICIENTS(NSTIF)=',T80,I10/,
+ 20X,'IN PROGMS NCOL AND 9 ARRAY SHOULD HAVE COLUMN DIMENSIONS '
+ /21X,'MORE THAN NSTIF.'//)
0128      410  FORMAT(12X,'THE FOLLOWING PARAMETERS ARE ASSOCIATED WITH ELEMENTS
+//20X,'TOTAL NUMBER OF SURFACE ELEMENTS(NESURF) =',T80,I10/,
+ /20X,'MAXIMUM NUMBER ASSIGNED TO SURFACE ELEMENT(NSELEM)=',T80,I10
+ /20X,'TOTAL NUMBER OF ELEMENTS IN THE SYSTEM (NELEM)=',T80,I10/
+ 20X,'MAXIMUM NUMBER OF NODES IN ANY ELEMENT(NODMAX)=',T80,I10/,
+ 20X,'MAXIMUM NON-ZERO NODES IN ANY ELEMENTS(NONZER) =',
+ T80,I10//,
+ 12X,'THE FOLLOWING IS RELATED TO MATERIAL NUMBER ASSIGNED'//,
+ 20X,'MAXIMUM NUMBER ASSIGNED TO MATERIAL(MATN) =',T80,I10/,
+ /120(' ')//)
0129      STOP
0130      END

```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODE1	003524 938	RW,I,CON,LCL
SPDATA	000336 111	RW,D,CON,LCL
SIDATA	002414 646	RW,D,CON,LCL

FORTRAN IV-PLUS V02-51D
BAND.FIN /TRIRLOCKS/WR

16105146 13-SEP-79

PAGE 4

SVARS	040604	8366	RW,0,CON,LCL
STEPS	000016	7	RW,0,CON,LCL
HDR	000274	94	RW,0,OVR,G8L
ELEM	000144	50	RW,0,OVR,G8L
TRACE	000014	6	RW,0,OVR,G8L

TOTAL SPACE ALLOCATED = 047774 10238

,LP1=BAND

9.4 PROG3I, PROGRAM LISTINGS

FORTRAN IV-PLUS V02-510
PROG3I.FTN /TRILLOCKS/WR

16106112

13-SEP-79

PAGE 1

```

C      *** PROG3I.FTN *****
C-----
C      PROGRAM TO READ TIME STEPS AND ALSO ELEMENT RELATED STRESS CONDIITI
C      IT HAS OPTION TO READ TIME VARIANT HELD POTENTIAL BOUNDARY CONDIITI
C-----
0001      BYTE MCR(80)
0002      INCLUDE 'HOR.FTN'
0003      * COMMON/HOR/LNPT,NTT,NBPTC,NBPTC,NSTREM,NPT,NSELEM,NELEM,NODMAX,
*      * NONZER,NQD,MATN,KTYPE,NMAX,NSTIF,NTIME,NSTORE,NDUM,TITLE(20),
*      * #OP1,OP2,OP3,OP4,OP5,OP6,OP7,OP8,OP9,OP10,OP11,OP12,OP13,OP14,OP15,
*      * #OP16,OP17,OP18,NTSURF,NE$URF
0004      INCLUDE 'P3I.FTN'
0005      * COMMON/P3I/NSTEODY,ITOTAL,NQWCP,NQ,MNTRY,ITT,NMBOND,NPDUM,TT(70),
*      * #DETAIL(20)
0006      COMMON/TRACE/NPROG1,NPROG2,NBAND,NPROG3,NPROG3,NSUB3
0007      DIMENSION SUMQ(2560),Q(768),FILE(7),FILE1(7),BIV(768),NODB(768),D
+      + ELT(70)
0008      CALL GETMCR (MCR,ISW)
0009      IF (ISW.GT.79) ISW = -40
0010      IF (ISW.LT.0) GO TO 110
0011      DO 100 I=ISW+1,80
0012      100  MCR(I) = ' '
0013      110  IF (ISW.GT.1) ISW = 1
0014      IF (ISW.LT.0) ISW = -1
0015      IF (ISW.NE.1) GO TO 130
0016      IF (MCR(5).EQ.' ',OR,MCR(5).EQ.0) ISW = 2
0017      IF (ISW.NE.1) GO TO 130
0018      DECODE (30,120,MCR(5))IUN,NPRINT,FILE
0019      120  FORMAT (2I5,7A4)
0020      130  IF (ISW.EQ.1) GO TO 140
0021      WRITE (2,*) 'ENTER FILE=Q DISK UNIT'
0022      READ (2,*) IUN
0023      140  IF (IUN.NE.3) CALL ABNLUN (1,'OP',IUN,ISTAT)
0024      IF (ISW.EQ.1) GO TO 160
0025      WRITE (2,*) 'ENTER INPUT FILE NAME'
0026      READ (2,150) FILE
0027      150  FORMAT (20A4)
0028      160  N = ICHR(FILE)
0029      OPEN (UNIT=5,NAME=FILE,TYPE='OLD',READONLY)
0030      IF (ISW.EQ.1) GO TO 170
0031      WRITE (2,*) 'ENTER PRINT OPTION(0,1,2)'
0032      READ (2,*) NPRINT
0033      170  CALL OPFILE (1,'TRACE',1,,IPTR10)
0034      CALL DPR (IPTR10,0,,NPROG1,6.)
0035      IF (NPROG1.EQ.1) GO TO 180
0036      WRITE (2,*) 'PROG1 NOT COMPLETED THEREFORE PROG3I STOPPED'
0037      STOP
0038      180  CALL OPFILE (1,'DESCRIP',25,,IPTR1)
0039      CALL DPR (IPTR1,0,,LNPT,94,)
0040      NPROG3I = 0
0041      CALL OPFILE (1,'DTANDQS',DP5,IPTR5)
0042      ITT = 1
0043      IF (ISW.EQ.1) GO TO 190
0044      WRITE (2,*) 'NEW TITLE(10Y,0=N)'
0045      READ (2,*) NEW
0046      IF (NEW.EQ.0) GO TO 190

```

```

0047      WRITE (2,*) 'WRITE NEW TITLE'
0048      READ (2,150) TITLE
0049      CALL DPH (IPTR1,0,,LNPT,94,.)
0050      190 READ (5,200) DETAIL
0051      200 FORMAT (20A4)
0052      IF (NPRINT.NE.0) PRINT 210, DETAIL
0053      210 FORMAT (1H1/120(' '),/1X,20A4/120(' ')/)
0054      READ (5,220) NSTEDY,ITOTAL,NQWCP,NQ,MNTRY,NHBOND,NTEXP
0055      IF (NPRINT.NE.0) PRINT *, 'NSTEDY',NSTEDY,'ITOTAL',ITOTAL,'NQWCP',
+      NQWCP,'NQ',NQ,'MNTRY',MNTRY,'NHBOND',NHBOND,'NTEXP',NTEXP
0056      220 FORMAT (16I5)
0057      WRITE (2,*) 'NSTEDY',NSTEDY,'ITOTAL',ITOTAL,'NQWCP',NQWCP,'NQ',NQ,
+      'MNTRY',MNTRY,'NHBOND',NHBOND,'NTEXP',NTEXP
0058      IF (NSTEDY.EQ.0) GO TO 360
0059      IF (NPRINT.NE.0) PRINT 230
0060      230 FORMAT (10X,'TRANSIENT RUN OF MODEL PRESCRIBED'//)
0061      IF (NPRINT.NE.0) PRINT 240, ITOTAL
0062      240 FORMAT (10X'TOTAL NUMBER OF TIME STEPS(ITOTAL)',T50,I10//)
0063      IF (ITOTAL.LE.70) GO TO 250
0064      WRITE (2,*) 'ITOTAL',ITOTAL,'>THAN 70 PROGRAM STOPPED'
0065      PRINT *, 'ERROR ITOTAL',ITOTAL,'WHICH IS > THAN 70 PROGRAM STOPPED'
+
0066      STOP
0067      250 IF (NTEXP.EQ.1) GO TO 280
0068      READ (5,260) (TT(I),I=1,ITOTAL)
0069      260 FORMAT (16F5,0)
0070      DELT(1) = TT(1)
0071      DO 270 I=2,ITOTAL
0072      270 DELT(I) = TT(I)-TT(I-1)
0073      GO TO 300
0074      280 READ (5,260) TT(1),FACT
0075      DELT(1) = TT(1)
0076      DO 290 I=2,ITOTAL
0077      290 DELT(I) = DELT(I-1)*FACT
0078      TT(I) = TT(I-1)+DELT(I)
0079      300 DO 310 I=1,ITOTAL
0080      IF (DELT(I).LE.0.0) WRITE (2,*) 'ERROR ** DELT = 0.0 FOR TIME
+      STEP',I
0081      310 IF (NPRINT.GE.1) PRINT 320, I,DELT(I),TT(I)
0082      320 FORMAT (10X,I5,' TIME STEP',015.4,5X,'ELAPSED TIME',015.4)
0083      IF (NHBOND.EQ.0) GO TO 380
+-----+
C      TIME VARIANT HELD HEAD B,C, READ FROM BINAR FILE
+-----+
0084      CALL DPHFILE (1,'BOUNDCON',DP7,IPTR7)
0085      CALL DPR (IPTR7,0,,NODB,768,.)
0086      READ (5,150) FILE1
0087      N = ICHR(FILE1)
0088      OPEN (UNIT=3,NAME=FILE1,FORM='UNFORMATTED',TYPE='OLD',READONLY)
0089      READ (3) NBTT,CONV
0090      IF (NBTT.NE.NBPTC) WRITE (2,*) 'ERROR IN NBTT',NBTT,' NBPTC',NBPTC
0091      DO 330 I=1,ITOTAL
0092      330 READ (3) (BIV(J),J=1,NBPTC)
0093      IF (CONV.EQ.1,0,OR,CONV.EQ.0.0) GO TO 340
0094      DO 330 J=1,NBPTC
0095      330 BIV(J) = BIV(J)*CONV

```

```

0096      340      ADR = (I-1)*6+1821,
0097              CALL DPW (IPTR5,ADR,8IV,1936,)
0098              IF (NPRINT.EQ,0) GO TO 350
0099              PRINT 660, I
0100              PRINT 670, (NOOB(J),8IV(J),J=1,NBPTC)
0101      350      CONTINUE
0102              GO TO 380
0103      360      IF (NPRINT.NE,0) PRINT 370
0104      370      FORMAT (//10X,'STEADY STATE SOLUTION IS PRESCRIBED'//)
0105      380      CALL DPW (IPTR5,0,,NSTEDY,188,)
0106              GO TO (390,410,460), NWCPC
C-----
C      ELEMENT BASE SOURCE AND SINK TERMS
C-----
0107      390      IF (NPRINT.NE,0) PRINT 400
0108      400      FORMAT (//10X,'NO NEW SOURCE OR SINK FOR EACH ELEMENT' ' IS READ IN
+      SOLUTION BASED ON SOURCE/SINK READ' ' IN PROGRAM 1'//)
0109              GO TO 510
0110      410      READ (5,420) CONVER
0111              READ (5,420) (SUMQ(J),J=1,NELEM)
0112      420      FORMAT (8E10,3)
0113              IF (CONVER.EQ,0,,OR,CONVER.EQ,1) GO TO 440
0114              DO 430 J=1,NELEM
0115      430      SUMQ(J) = SUMQ(J)*CONVER
0116      440      CALL DPW (IPTR5,1,,SUMQ,5120,)
0117              IF (NPRINT.NE,0) PRINT 450, (J,SUMQ(J),J=1,NELEM)
0118      450      FORMAT (1H1//,120(' '),10X,'NEW CONSTANT TOTAL SOURCE/SINK FOR ' 'E
+      ACH ELEMENT ' 'IS READ AS BELOW'//120(' ')//,200(1X,8(' ',I4,F10,3)
+      +/))
0119              GO TO 510
0120      460      CONTINUE
0121              READ (5,420) CONVER
0122              DO 500 I=1,ITOTAL
0123              READ (5,420) (SUMQ(J),J=1,NELEM)
0124              ADR = (I-1)*20,+1
0125              IF (CONVER.EQ,0,,OR,CONVER.EQ,1) GO TO 480
0126              DO 470 J=1,NELEM
0127      470      SUMQ(J) = SUMQ(J)*CONVER
0128      480      CONTINUE
0129              CALL DPW (IPTR5,ADR,SUMQ,5120,)
0130              IF (NPRINT.NE,0) PRINT 490, I, (SUMQ(J),J=1,NELEM)
0131      490      FORMAT (1H1//,120(' '),10X,'EACH ELEMENT SOURCE/SINK FOR' ' TIME ST
+      EP',I5//120(' ')//,200(1X,8(' ',I4,F10,3)/))
0132      500      CONTINUE
0133      510      CONTINUE
0134              GO TO (520,540,590), NO
C-----
C      INFILTRATION AT TOP OF EACH SURFACE ELEMENT READ
C-----
0135      520      IF (NPRINT.NE,0) PRINT 530
0136      530      FORMAT (//10X,'NO SURFACE FLUX IS PRESCRIBED')
0137              GO TO 640
0138      540      READ (5,420) CONVER
0139              READ (5,550) (Q(I),I=1,NSALEM)
0140      550      FORMAT (8E10,5)
0141              IF (NPRINT.NE,0) PRINT *, 'TOTAL SURFACE ELEMENTS',NSALEM

```



```

0142      IF (CONVER.EQ.0..OR.CONVER.EQ.1.) GO TO 570
0143      DO 560 J=1,NSELEM
0144      Q(J) = Q(J)*CONVER
0145      560  CONTINUE
0146      570  CONTINUE
0147      IF (NPRINT.NE.0) PRINT 580, (J,Q(J),J=1,NSELEM)
0148      580  FORMAT (1H1//,120(' '),10X,'CONSTANT SURFACE FLUX PRESCRIBED'/120
+(' ')/,200(1X,8(' ',I4,F10.3)/))
0149      CALL DPW (IPTR5,1401.,Q,1536.)
0150      GO TO 640
0151      590  CONTINUE
0152      READ (5,420) CONVER
0153      DO 630 I=1,ITOTAL
0154      READ (5,550) (Q(J),J=1,NSELEM)
0155      IF (CONVER.EQ.0..OR.CONVER.EQ.1.) GO TO 610
0156      DO 600 J=1,NSELEM
0157      Q(J) = Q(J)*CONVER
0158      600  CONTINUE
0159      610  CONTINUE
0160      IF (NPRINT.NE.0) PRINT 620, I,(J,Q(J),J=1,NSELEM)
0161      620  FORMAT (1H1//,120(' '),10X,'SURFACE FLUX PRESCRIBED FOR TIME''STEP
+ ',I5//120(' ')/,200(1X,8(' ',I4,F10.3)/))
0162      ADR = (I-1)*6.+1401.
0163      CALL DPW (IPTR5,ADR,Q,1536.)
0164      630  CONTINUE
0165      640  CALL DPCL (IPTR5)
0166      NPROG1 = 1
0167      CALL DPW (IPTR10,0.,NPROG1,6.)
0168      CALL DPCL (IPTR1)
0169      IF (MNTY.NE.0) PRINT 650, MNTY
0170      650  FORMAT (/120(' ')/,10X,'IMPORTANT:--MNTY=',I5,'SIMULATION SHALL
+ BE DONE USING THIS AS THE '' MATERIAL # FOR ALL ELEMENTS'',//10X,'
+ THIS MEANS THE ASSIGNED MATERIAL # OF EVERY ELEMENT'' SHALL BE OVE
+ RRIDDEN BY THIS VALUE'//,120(' ')/)
0171      660  FORMAT (1H1//,' POTENTIAL HELD BONDARY VALUES',T60,I5//)
0172      670  FORMAT (4(1X,I5,' NODE',F12.3,' DIV'))
0173      STOP
0174      END

```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
BCODE1	006232 1613	RW,I,CON,LCL
SPDATA	000710 220	RW,D,CON,LCL
SIDATA	002324 610	RW,D,CON,LCL
SVAR5	043714 9190	RW,D,CON,LCL
STEMP5	000006 3	RW,D,CON,LCL
HDR	000274 94	RW,D,OVR,GBL
P31	000570 188	RW,D,OVR,GBL
TRACE	000014 6	RW,D,OVR,GBL

TOTAL SPACE ALLOCATED = 056510 11940

9.5 PROG3 AND ITS SUBROUTINES, PROGRAM LISTINGS

FORTRAN IV=PLUS V02=510
PROG3.FTN /TRILOCK/WR

16107102 13-SEP-79

PAGE 1

```

C      * PROG3.FTN
C-----
C      PROGRAM 3 FOR DETERMINATION OF HEAD
C-----
0001      COMMON /POINT/IPTR1,IPTR2,IPTR3,IPTR4,IPTR5,IPTR6,IPTR7,IPTR8,
          + IPTR9,IPTR10,IPTR11,IPTR12,IPTR13,IPTR14,IPTR15,IPTR16,IPTR17,IPTR18
0002      INCLUDE 'HDR.FTN'
0003      * COMMON/HDR/LNPT,NTT,NBPTC,NBPTC,NBTREM,NPT,NSELEM,NELEM,NODMAX,
          * #NONZER,NQD,MATN,KTYPE,NMAX,NSTIF,NTIME,NSTORE,NDUM,TITLE(20),
          * #DP1,DP2,DP3,DP4,DP5,DP6,DP7,DP8,DP9,DP10,DP11,DP12,DP13,DP14,DP15,
          * #DP16,DP17,DP18,NTSURF,NEBURF
0004      INCLUDE 'P3I.FTN'
0005      * COMMON/P3I/NSTEADY,ITOTAL,NQWCP,NQ,MNTRY,ITT,NHBOND,NFDUM,TT(70),
          * #DETAIL(20)
0006      COMMON/TRACE/NPROG1,NPROG2,NBAND,NPROG3,NSUB3
0007      CALL START3 (NPRINT,NBEGIN,NREDO,NDIG,NCONS,CDELT,TTOTAL,IBBAND)
0008      IF (NPRINT.NE.0) CALL PTIME (6)
0009      CALL DPFIL (1,'PROPERTY',DP2,IPTR2)
0010      NPPP = NPRINT
0011      DELTAT = CDELT
0012      IT = ITT
0013      IF (ITT.GT.1) TTIME = TT(IT-1)
0014      IF (NCONS.EQ.1) GO TO 100
0015      DO 220 ITM=ITT,TTOTAL
0016      ITX = ITM
0017      GO TO 110
0018      100 IF (ITT.EQ.1) TBEG = 0.0
0019      IF (ITT.GT.1) TBEG = TT(IT-1)
0020      DO 220 TX=TBEG,TT(ITOTAL),CDELT
0021      ITX = ITX+1
0022      110 NWRITE = 0
0023      ISTEP = ISTEP+1
0024      IF (ISTEP.GT.1.AND.NPRINT.GT.1) NPRINT = 1
0025      IF (NCONS.EQ.0) GO TO 120
0026      IF (ITX.NE.1) NBEGIN = 5
0027      TTIME = TTIME+DELTAT
0028      IF (TTIME.LT.TT(IT)) GO TO 130
0029      NWRITE = 1
0030      GO TO 130
0031      120 IT = ITX
0032      IF (IT.EQ.1) DELTAT = TT(IT)
0033      IF (IT.GT.1) DELTAT = TT(IT)-TT(IT-1)
0034      NWRITE = 1
0035      TTIME = DELTAT+TTIME
0036      130 IF (NPPP.NE.0) GO TO 140
0037      NPRINT = NPPP
0038      CALL SWITCH (1,ISW)
0039      IF (ISW.EQ.1) NPRINT = 1
0040      140 ITER = 1
0041      GO TO (150,160,170,180,190,200,210), NBEGIN
0042      150 CALL ELEMAT (NSTEADY,MNTRY,DELTAT,IT,NBPTC,NELEM,KTYPE,NMAX,NPRINT)
0043      160 CALL SYSMAT (NELEM,NSTIF,NPT,NPRINT)
0044      170 IF (THRAND.EQ.0) CALL MREAD (NPT,NPRINT)
0045      IF (THRAND.EQ.1) CALL MREADR (NPT,NPRINT)
0046      180 IF (TRRAND.EQ.0) CALL UPERT (NPRINT,NREDO)
0047      IF (TRRAND.EQ.1) CALL UPERTR (NPRINT,NREDO)

```

```

0048 190 CALL BVALUE(IT,NQWCP,NQ,NBFTC,NSTREM,NMAX,NPT,NPRINT,NELEM,NSELEM
      +,NSTEDY,LNPT,NBPTC,NHBOND)
0049 200 CALL BACKSU (NPRINT)
0050 210 CALL HWRITE (NPRINT,IT,NOIG,DELTAT,NCONS,TTIME,NWRITE,NSI1)
0051      CLOSE (UNIT=6)
0052      NBEGIN = 5
0053      NREND = 0
0054      IF (NSTEDY.EQ.0) GO TO 220
0055      IF (NCONS.EQ.0.AND.ABS(IT(IT+1)-TT(IT)-DELTAT)/DELTAT.GT.,.1E-4) NB
      +EGIN = 1
0056 220 CONTINUE
0057      NPROG3 = 1
0058      NSUB3 = 0
0059      CALL DPW (IPTR18,0.,NPROG1,6.)
0060      IF (NPRINT.NE.0) CALL PTIME (6)
0061      STOP
0062      END
  
```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
\$CODE1	001424 394	RW,I,CON,LCL
\$PDATA	000056 23	RW,D,CON,LCL
\$IDATA	000224 74	RW,D,CON,LCL
\$VARS	000066 27	RW,D,CON,LCL
\$TEMPS	000012 3	RW,D,CON,LCL
POINT	000042 17	RW,D,OVR,GBL
HDR	000274 94	RW,D,OVR,GBL
P31	000570 188	RW,D,OVR,GBL
TRACE	000014 6	RW,D,OVR,GBL

TOTAL SPACE ALLOCATED = 003170 828

,LF:PROG3/CO:00

```

C      START3.FTN
0001      SUBROUTINE START3 (NPRINT,NBEGIN,NREDO,NDIG,NCONS,COELT,TTOTAL,
+IBAND)
0002      BYTE MCR(80)
0003      COMMON /POINT/IPTR1,IPTR2,IPTR3,IPTR4,IPTR5,IPTR6,IPTR7,IPTR8 ,
+ IPTR10,IPTR11,IPTR12,IPTR13,IPTR14,IPTR15,IPTR16,IPTR17,IPTR18
0004      INCLUDE 'HDR.FTN'
0005 *      COMMON/HDR/LNPT,NTT,NBPTC,NBPPTC,NSTREM,NPT,NSELEM,NELEM,NODMAX,
*      #NONZER,NQD,MATN,KTYPE,NMAX,NSTIF,NTIME,NSTORE,NDOUM,TITLE(20),
*      #DP1,DP2,DP3,DP4,DP5,DP6,DP7,DP89,DP10,DP11,DP12,DP13,DP14,DP15,
*      #DP16,DP17,DP18,NTSURF,NESURF
0006      INCLUDE 'PSI.FTN'
0007 *      COMMON/PSI/NSTEDY,ITOTAL,NQWCP,NQ,MNTRY,ITT,NHBOND,NFDUM,TT(70),
*      #DETAIL(20)
0008      COMMON/TRACE/NPROG1,NPROG2,NBAND,NPRO31,NPROG3,NSUB3
0009      DIMENSION FILE(7),HEAD(2560)
0010      CALL GETMCR (MCR,ISW)
0011      IF (ISW.GT.79) ISW = -40
0012      IF (ISW.LT.0) GO TO 110
0013      DO 100 I=ISW+1,80
0014      100 MCR(I) = ' '
0015      110 IF (ISW.GT.1) ISW = 1
0016      IF (ISW.LT.0) ISW = -1
0017      IF (ISW.NE.1) GO TO 130
0018      IF (MCR(5).EQ.' ' .OR. MCR(5).EQ.0) ISW = 2
0019      IF (ISW.NE.1) GO TO 130
0020      DECODE (45,120,MCR(5)) IUN,NPRINT,NBEGIN,ITT,NREDO,NDIG,NCONS,COELT
0021      120 FORMAT (7I3,E10,3)
0022      IF (NBEGIN.NE.4) NREDO = 0
0023      130 IF (ISW.EQ.1) GO TO 140
0024      WRITE (2,*) 'ENTER FILE=Q DISK UNIT'
0025      READ (2,*) IUN
0026      140 IF (IUN.NE.3) CALL ASNLUN (1,'DP',IUN,ISTAT)
0027      IF (ISW.EQ.1) GO TO 160
0028      WRITE(2,*) 'PRINT OPTION (NORMAL= 1, FOR DEBUG = 2 , NO OUTPUT=0)'
0029      WRITE(2,*) 'NBEGIN(NORMAL= 1 AND FOR RESTART = SUBROUTINE IN MAIN)'
0030      WRITE(2,*) 'IT=TIME STEP TO START (NORMAL=1,RESTART=TIME STEP)'
0031      WRITE(2,*) 'NDIG = OUTPUT DIGITS AFTER DECIMAL IN "F" FORMAT'
0032      WRITE(2,*) 'NCONS = 0 FOR TIME STEPS READ IN PROG31'
0033      WRITE(2,*) '          = 1 FOR CONSTANT TIME STEP FOR COMPUTATION'
0034      WRITE(2,*) ' '
0035      WRITE (2,*) 'NPRINT(0,1,2),NBEGIN(1,2,3,4,5,6,7),IT,NDIG,NCONS'
0036      READ (2,*) NPRINT,NBEGIN,ITT,NDIG,NCONS
0037      WRITE(2,*) 'IS EXPECTED BAND WIDTH TO BE MORE THAN 252(1=Y,0=N)'
0038      READ (2,*) IBAND
0039      IF (NCONS.EQ.0) GO TO 150
0040      WRITE (2,*) 'CONSTANT DELTAT'
0041      READ (2,*) COELT
0042      150 IF (NBEGIN.NE.4) GO TO 160
0043      WRITE (2,*) 'RESTART(1=NORMAL RESTART,2=SYSTEM CRASH)'
0044      WRITE(2,*) 'RESTART OPTION WORKS ONLY WITH BANDWIDTH <252'
0045      WRITE(2,*) 'RESTART WITH NBEGIN=3 IF ERROR DETECTED'
0046      READ (2,*) NREDO
0047      160 CALL DPFILE (1,'TRACE',1.,IPTR18)
0048      CALL DPR (IPTR18,0.,NPROG1,6.)
0049      IF (NSUB3.EQ.0) NSUB3 = 1

```

```

0050      IF (NPROG3.EQ.1) WRITE (2,*) 'PROG3 COMPLETED LAST TIME,THIS IS RE
+RUN'
0051      IF (NPROG1.EQ.1) GO TO 170
0052      WRITE (2,*) 'PLEASE NOTE THAT PROG1 NOT COMPLETED'
0053      NSTOP = 1
0054      170  IF (NPROG2.EQ.1) GO TO 180
0055      WRITE (2,*) 'PROG2 NOT COMPLETED OR RERUN IN PROG1 WAS NOT 1'
0056      NSTOP = 1
0057      180  IF (NRAND.EQ.1) GO TO 190
0058      WRITE (2,*) 'RAND NOT COMPLETED PROG3 STOPPED:NBAND= ',NBAND
0059      NSTOP = 1
0060      190  IF (NPROG3I.EQ.1) GO TO 200
0061      WRITE (2,*) 'OLD VALUES OF PROG3I USED'
0062      200  IF (NSTOP.EQ.0) GO TO 210
0063      WRITE (2,*) 'PROG3 STOPPED BECAUSE PROG1,2, OR RAND NOT COMPLETE'
0064      STOP
0065      210  IF (NBEGIN.NE.4.OR.ISW.NE.1) GO TO 220
0066      NREDO = MOD(NSUB3,10)
0067      NSUB3 = NSUB3/10
0068      IF (NSUB3.EQ.4.AND.NREDO.EQ.0) NSUB3 = 3
0069      220  IF (NREDO.EQ.0.AND.NBEGIN.EQ.4) NBEGIN = 3
0070      IF (NSUB3.LT.NBEGIN) WRITE (2,*) 'NSUB3',NSUB3,'NBEGIN',NBEGIN
0071      IF (NSUB3.LT.NBEGIN) NBEGIN = NSUB3
0072      CALL DPFIL (1,'DESCRIP',25,,IPTR1)
0073      IF (NBEGIN.NE.4) NREDO = 0
0074      CALL DPR (IPTR1,0,LNPT,94,.)
0075      IF (IBBAND.EQ.0) CALL DPFIL (1,'UPBACKUP',DP17,IPTR17)
0076      CALL DPFIL (1,'XYZCOR',DP3,IPTR3)
0077      C      IF (IBBAND.EQ.1)CALL DPFIL(1,'UPBACKUP',DP17*2,IPTR17)
0078      CALL DPFIL (1,'DTANDQ3',DP5,IPTR5)
0079      CALL DPFIL (1,'ELEMATRIX',DP6,IPTR6)
0080      CALL DPFIL (1,'ELEMENT1',DP11,IPTR11)
0081      CALL DPFIL (1,'BOUNDCOND',DP7,IPTR7)
0082      CALL DPFIL (1,'FACTOR',DP89,IPTR89)
0083      CALL DPFIL (1,'NCOL',DP10,IPTR10)
0084      IF (IBBAND.EQ.0) CALL DPFIL (1,'UPTRIANG',DP12,IPTR12)
0085      IF (IBBAND.EQ.1) CALL DPFIL (1,'UPTRIANG',DP12*2,IPTR12)
0086      CALL DPFIL (1,'BACKSUB',DP13,IPTR13)
0087      CALL DPFIL (1,'HEAD',DP14,IPTR14)
0088      CALL DPFIL (1,'BVALUE',DP15,IPTR15)
0089      CALL DPFIL (1,'RESTART',DP16,IPTR16)
0090      CALL DPFIL (1,'ELEMENT0',DP4,IPTR4)
0091      ITSAV = ITT
0092      CALL DPR (IPTR5,0,,NSTEDY,100,.)
0093      ITT = ITSAV
0094      IF (ITOTAL.GT.70) WRITE (2,*) 'ERROR ITOTAL > 50',ITOTAL
0095      ITOTAL = ITOTAL
0096      IF (NPRINT.LE.1) GO TO 230
0097      WRITE (6,250) NTSURF,LNPT,NTT,NBPTC,NBPTC,NSTREM,NPT,NSTIF
0098      WRITE (6,260) NESUR,NSELEM,NELEM,NONZER,MATN,KTYPE,NMAX
0099      230  IF (NSTEDY.EQ.0) GO TO 240
0100      IF (NCONS.EQ.1.AND.ITT.GT.1) ITOTAL = (TT(ITOTAL)-TT(ITT-1))/CDELT
0101      IF (NCONS.EQ.1.AND.ITT.EQ.1) ITOTAL = TT(ITOTAL)/CDELT
0102      240  CALL DPR (IPTR14,(ITT-1)*20.,HEAD,NTT*2,.)
0103      CALL UPW (IPTR89,20.,HEAD,NTT*2,.)
0104      250  FORMAT (/120('*')/,20X,'TOTAL NUMBER OF SURFACE NODES(NTSURF)=',TT

```

```

+0,I10/,20X,'MAXIMUM NUMBER ASSIGNED TO SURFACE NODES(LNPT)'=,T70,I
+10/,20X,'TOTAL NODES IN THE SYSTEM (NTT)'=,T70,I10//,12X,'THE FOLL
+OWING PARAMETERS ARE RELATED TO B.C'//,20X,'TOTAL NUMBER OF POTENTI
+AL B.C. PRESCRIBED (NBPTC)'=,T70,I10/20X,'TOTAL NUMBER OF FLUX B.C
+. PRESCRIBED (NBFPCT)'=,T70,I10/,20X,'TOTAL NUMBER OF STREAMS B.C
+. PRESCRIBED (NSTREM)'=,T70,I10//,12X,'THE FOLLOWING TWO PARAMETER
+S ARE RELATED TO STIFFNESS MATRIX'//,20X,'TOTAL NUMBER OF UNKNOWN(N
+PT)'=,T70,I10/,20X,'MAXIMUM NUMBER OF NON-ZERO COEFFICIENTS(NSTIP)
+*',T70,I10//)
0104      260  FORMAT (12X,'THE FOLLOWING PARAMETERS ARE ASSOCIATED WITH ELEMENTS
+ '//20X,'TOTAL NUMBER OF SURFACE ELEMENTS(NESURF)' =,T70,I10//,20X,
+'MAXIMUM NUMBER ASSIGNED TO SURFACE ELEMENT(NSELEM)'=,T70,I10/20X,
+'TOTAL NUMBER OF SURFACE ELEMENTS(NSELEM)' =,T70,I10/,20X,'TOTAL N
+UMBER OF ELEMENTS(NELEM)' =,T70,I10/,20X,'MAXIMUM NON-ZERO NODES
+IN ANY ELEMENTS(NONZER)' =,T70,I10//,12X,'THE FOLLOWING IS RELATED
+ TO MATERIAL NUMBER ASSIGNED'//,20X,'MAXIMUM NUMBER ASSIGNED TO MAT
+ERIAL(MATN)' =,T70,I10/,20X,'KTYPE-THE FLAG FOR K VALUES'=,T70,I10
+/,20X,'NMAX' =,T70,I10//,120('*')//)
0105      WRITE (2,*) 'NBEGIN',NBEGIN,'NPRINT',NPRINT,'IT',ITT
0106      RETURN
0107      END

```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODE1	003452 917	RW,I,CON,LCL
SPDATA	002016 519	RW,D,CON,LCL
SIDATA	002372 637	RW,D,CON,LCL
SVARS	024172 5181	RW,D,CON,LCL
POINT	000042 17	RW,D,OVR,GBL
MOR	000274 94	RW,D,OVR,GBL
P3I	000570 188	RW,D,OVR,GBL
TRACE	000014 6	RW,D,OVR,GBL

TOTAL SPACE ALLOCATED = 035416 7559

,LP1=START3/CO140

```

C      * ELEMAT.FTN
0001      SUBROUTINE ELEMAT (NSTEDY,MNTRY,DELTAT,IT,NBPTC,NELEM,KTYPE,NMAX,
      +NPRINT)
C-----
C      ELEMENT MATRIX
C-----
0002      COMMON /POINT/IPTR1,IPTR2,IPTR3,IPTR4,IPTR5,IPTR6,IPTR7,IPTR89 ,
      + IPTR10,IPTR11,IPTR12,IPTR13,IPTR14,IPTR15,IPTR16,IPTR17,IPTR18
0003      COMMON/KXYZ/XK(50),YK(50),ZK(50),SPEC(50),THETA(50),ALPHA(50),
      +BETAP(50),PZERO(50)
0004      COMMON/ELEM1/NN,NOD(20),MM1,NQW,SUMQW,NE,NHEAD,XIJCOB(20),
      +XIDLEM(20),PIFJ(400),GIGJ(400),HIHJ(400),XIXJOB(400),IDUM,AREA
0005      COMMON/ELEMAT/NNN,NNE,MM,NHD,AREA1,SUMQ,NODD(20),ZY(20),
      +XIDLE1(20),XIJC01(20),CI(20,20),PIIJ(20,20)
0006      COMMON/TRACE/NPROG1,NPROG2,NBAND,NPROG3,NPROG4,NSUB3
0007      DIMENSION XKK(20),YKK(20),ZKK(20),FACTOR(256),
      +HEAD(256),RHO(256),Z(256),COEF(20)
0008      CALL DPR (IPTR10,0.,NPROG1,6.)
0009      CALL DPR (IPTR2,0.,XK,800.)
0010      NRR = 2000
0011      NMAXH = 256
0012      DO 100 I=1,20
0013      DO 100 J=1,10
0014      PPIJ(I,J) = 0.0
0015      DO 220 IX=1,NELEM
0016      ADR = (IX-1)*13,
0017      CALL DPR (IPTR11,ADR,NN,3310.)
0018      NHD = NHEAD
0019      MN = MM1
0020      IF (MNTRY.NE.0) MN = MNTRY
0021      DO 140 I=1,NN
0022      K = NOD(I)
0023      NODD(I) = K
0024      XIDLE1(I) = XIDLEM(I)
0025      XIJC01(I) = XIJC0B(I)
0026      IF (KTYPE.EQ.0.AND.(NSTEDY.EQ.0.OR,SPEC(MN),NE.0.0)) GO TO 130
0027      NR = (K-1)/NMAX
0028      IF (NR.EQ.NRR) GO TO 110
0029      ADR = NR*2,0
0030      CALL DPR (IPTR89,ADR,FACTOR,512.)
0031      CALL DPR (IPTR89,ADR+20.,HEAD,512.)
0032      CALL DPR (IPTR89,ADR+40.,RHO,512.)
0033      CALL DPR (IPTR3,ADR,Z,512.)
0034      NRR = NR
0035      110 KK = K+NR*NMAX
0036      IF (NSTEDY.EQ.0.OR,SPEC(MN),NE.0.0) GO TO 120
0037      THET = THETA(MN)*(HEAD(KK)-Z(KK)-PZERO(MN))
0038      COEF(I) = ALPHA(MN)+THET*BETAP(MN)
0039      120 IF (KTYPE.EQ.0) GO TO 130
0040      FACT = FACTOR(KK)
0041      XKK(I) = XK(MN)*FACT
0042      YKK(I) = YK(MN)*FACT
0043      ZKK(I) = ZK(MN)*FACT
0044      GO TO 140
0045      130 XKK(I) = XK(MN)
0046      YKK(I) = YK(MN)

```

```

0047      ZKK(I) = ZK(MN)
0048      140  CONTINUE
0049      PIJ = 0.0
0050      DO 200 I=1,NN
0051      IN = (I-1)*NN
0052      DO 180 J=1,NN
0053      IJ = IN+J
0054      CIJ = XKK(J)*FIFJ(IJ)+YKK(J)*GIGJ(IJ)+ZKK(J)*HIHJ(IJ)
0055      IF (NSTEDY.EQ.0) GO TO 170
0056      PPIJ(I,J) = -CIJ
0057      IF (SPEC(MN).NE.0.0) GO TO 150
0058      COF1 = COEF(J)
0059      GO TO 160
0060      150  COF1 = SPEC(MN)
0061      160  PIJ = XIXJOB(IJ)*COF1/DELTAT
0062      170  CI(I,J) = CIJ+PIJ
0063      180  CONTINUE
0064      ZY(I) = 0.0
0065      IF (NPRINT.GE.2.AND.IX.LE.6)PRINT 190, IX,I,ZY(I),(CI(I,J),J=1,NN)
0066      190  FORMAT (' IX',I4,' I',I3,' ZY',E10.3,'CI'/5(10X,10E10.3/))
0067      200  CONTINUE
0068      NNN = NN
0069      IF (NPRINT.GE.2.AND.IX.LE.6) PRINT 210, IX,(I,ZY(I),I=1,NN)
0070      210  FORMAT (' ELEM',I5/6(0(15,E10.3)/))
0071      NNE = NE
0072      MM = MM1
0073      AREA1 = AREA
0074      SUMQ = SUMQW
0075      ADRSN = (IX-1)*7.
0076      CALL DPM (IPTR6,ADRSN,NNN,1748.)
0077      220  CONTINUE
0078      NSUR3 = 1
0079      CALL DPM (IPTR18,0.,NPROG1,6.)
0080      IF (NPRINT.NE.0) WRITE (2,*) 'ELEMENT MATRIX COMPLETED'
0081      RETURN
0082      END

```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODE1	002370 636	RW,I,CON,LCL
SPDATA	000062 25	RW,D,CON,LCL
SIDATA	000220 72	RW,D,CON,LCL
SVARS	010562 2233	RW,D,CON,LCL
STEMPS	000010 4	RW,D,CON,LCL
POINT	000042 17	RW,D,OVR,GRL
KXYZ	003100 800	RW,D,OVR,GRL
ELEM1	014734 3310	RW,D,OVR,GRL
ELEMAT	006650 1748	RW,D,OVR,GRL
TRACE	000014 6	RW,D,OVR,GRL

TOTAL SPACE ALLOCATED = 042446 8851


```

C      SYSMAT,FTN
0001      SUBROUTINE SYSMAT (NELEM,NSTIF,NPT,NPRINT)
C-----
C      SYSTEM MATRIX
C-----
0002      COMMON /POINT/IPTR1,IPTR2,IPTR3,IPTR4,IPTR5,IPTR6,IPTR7,IPTR89 ,
+ IPTR10,IPTR11,IPTR12,IPTR13,IPTR14,IPTR15,IPTR16,IPTR17,IPTR18
0003      COMMON/TRACE/NPROG1,NPROG2,NBAND,NPROG3,NPROG3,NSUB3
0004      COMMON/ELEMAT/NN,NE,MM,MDUM,AR,SUMGW,NOD(20),ZY(20),XIDLEM(20),
+X1JCOB(20),CI(20,20)
0005      DIMENSION NCOL(120),A(120),NODE(2040),IBANDW(2040)
0006      CALL DPR (IPTR10,0.,NPROG1,6.)
0007      CALL DPR (IPTR1,17.,NODE,2040.)
0008      ADR = NPT*2,+1
0009      CALL DPR (IPTR10,ADR,IBANDW,2040.)
0010      DO 100 I=1,120
0011      100  A(I) = 0.
0012      DO 110 I=1,NPT
0013      ADR = I*2,-1
0014      CALL DPR (IPTR10,ADR,NCOL,120.)
0015      110  CALL DPW (IPTR10,ADR,NCOL,384.)
0016      DO 200 IE=1,NELEM
0017      ADR = (IE-1)*7.
0018      CALL DPR (IPTR6,ADR,NN,940.)
0019      IF (NPRINT.GE.2.AND.IE.LE.6) WRITE (6,120) NE,((CI(I,J),J=1,NN),I=
+1,NN)
C
0020      120  FORMAT (' NE',IS,'CI'/8(8E12.4/))
0021      DO 190 K=1,NN
0022      KM = NOD(K)
0023      DO 130 I=1,NPT
0024      IF (KM.EQ.NODE(I)) GO TO 140
0025      130  CONTINUE
0026      GO TO 190
0027      140  KK = I
0028      ADRN = KK*2-1
0029      CALL DPR (IPTR10,ADRN,NCOL,384.)
0030      IBAND = IBANDW(KK)
0031      DO 150 L=1,NN
0032      JL = NOD(L)
0033      DO 150 I=1,NPT
0034      IF (JL.EQ.NODE(I)) GO TO 160
0035      150  CONTINUE
0036      GO TO 180
0037      160  JJ = I
0038      DO 170 LL=1,IBAND
0039      IF (NCOL(LL).NE.JJ) GO TO 170
0040      A(LL) = A(LL)+CI(K,L)
0041      GO TO 180
0042      170  CONTINUE
0043      180  CONTINUE
0044      CALL DPW (IPTR10,ADRN,NCOL,384.)
0045      190  CONTINUE
0046      200  CONTINUE
0047      NSUB3 = 2
0048      CALL DPW (IPTR18,0.,NPROG1,6.)

```

```
0049      IF (NPRINT,NE,0) WRITE (2,*) 'SYSTEM MATRIX FORMED'
0050      RETURN
0051      END
```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODE1	001360	376
SPDATA	000062	25
SIDATA	000134	46
SVARS	021436	4495
STEMPS	000014	6
POINT	000042	17
TRACE	000014	6
ELEMAT	003550	948

TOTAL SPACE ALLOCATED = 027076 5919

,LP1=SYSMAT

```
0001      SUBROUTINE MREAD (NPT,NPRINT)
0002      COMMON /POINT/IPTR1,IPTR2,IPTR3,IPTR4,IPTR5,IPTR6,IPTR7,IPTR89
+ IPTR10,IPTR11,IPTR12,IPTR13,IPTR14,IPTR15,IPTR16,IPTR17,IPTR18
0003      COMMON/TRACE/NPROG1,NPROG2,NBAND,NPROG3,NPROG3,NSUR3
0004      DIMENSION NCOL(128),A(128)
0005      DO 100 I=1,NPT
0006      ADR = I*2-1
0007      CALL DPM (IPTR10,ADR,NCOL,304.)
0008      ADR = (I-1)*3.
0009      CALL DPM (IPTR12,ADR,A,256.)
0010      ADR = ADR+2.
0011      CALL DPM (IPTR12,ADR,NCOL,128.)
0012      NSUR3 = 3
0013      CALL DPM (IPTR18,0.,NPROG1,6.)
0014      IF (NPRINT,NE,0) WRITE (2,*) 'MREAD SUBROUTINE COMPLETED'
0015      RETURN
0016      END
```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODE1	000256	87
SPDATA	000060	24
SIDATA	000050	20
SVARS	001406	387
STEMPS	000002	1
POINT	000042	17
TRACE	000014	6

TOTAL SPACE ALLOCATED = 002074 542

,LP1=MREAD

```

      C      UPPERT.FTN
0001      SUBROUTINE UPPERT (NPRINT,NREDO)
0002      COMMON /POINT/IPTR1,IPTR2,IPTR3,IPTR4,IPTR5,IPTR6,IPTR7,IPTR8,
      + IPTR10,IPTR11,IPTR12,IPTR13,IPTR14,IPTR15,IPTR16,IPTR17,IPTR18
0003      COMMON/TRACE/NPROG1,NPROG2,NBAND,NPROG3,NSUB3
0004      INCLUDE 'HOR.FTN'
0005      * COMMON/HDR/LNPT,NTT,NBPTC,NBFTC,NSTRE,NPT,NSELEM,NELEM,NODMAX,
      * #NONZER,NQO,MATN,KTYPE,NMAX,NSTIF,NTIME,NSTORE,NDUM,TITLE(20),
      * #DP1,DP2,DP3,DP4,DP5,DP6,DP7,DP8,DP9,DP10,DP11,DP12,DP13,DP14,DP15,
      * #DP16,DP17,DP18,NTSURF,NESURF
0006      COMMON/UPPERT/AA(256),NNCOL(252),NC,MAXCOL,MINROW,NFAK
0007      COMMON/BSTORE/IPIVOT(768),IOPROW(768),COEF(768)
0008      COMMON/SYST/A(256),NCOL(252)
0009      COMMON/START/LL,MAXWID,NNTIME,IBAND,NNSTOR,NDU,IBANDW(2048),
      + NSTOPP(2048),NSTART(2048)
0010      COMMON/STORE/AST(3840)
0011      DIMENSION IST(7680)
0012      EQUIVALENCE(AST(1),IST(1))
0013      CALL DPR (IPTR18,0.,NPROG1,6.)
0014      NSUB3 = 40
0015      CALL DPW (IPTR18,0.,NPROG1,6.)
0016      CALL DPR (IPTR1,0.,LNPT,90.)
0017      ADR = 2*NPT+1.
0018      CALL DPR (IPTR10,ADR,IBANDW,2048.)
0019      ADR = ADR+8.
0020      CALL DPR (IPTR10,ADR,NSTART,2048.)
0021      ADR = ADR+8.
0022      CALL DPR (IPTR10,ADR,NSTOPP,2048.)
      C
0023      DOUBLE PRECISION A,AA,B,X,C,AAA,A1,SAVE,ZTEST,DABS
0024      NSTORE = 0
0025      NRQWS = 10
0026      LIMIT = 10000
0027      IF (NPRINT.EQ.1) LIMIT = 100
0028      IF (NPRINT.GE.2) LIMIT = 10
0029      IF (NPRINT.GE.3) LIMIT = 1
0030      NCOUNT = 0
0031      LPRINT = 0
0032      NTIME = 0
0033      ZTEST = 0.
0034      NMAXX = 768
0035      RECORD = 12.
0036      NNN = 253
0037      MAXWID = NSTIF
0038      NBEG = 1
0039      IF (NREDO.EQ.0) GO TO 110
0040      CALL DPR (IPTR16,0.,LL,6150.)
0041      IF (NPRINT.GE.3) PRINT *, 'IBANDW READ',(I,IBANDW(I),I=1,NPT)
0042      NBEG = LL
0043      NSTORE = NNSTOR
0044      NTIME = NNTIME
0045      ADR3 = 16.+NTIME*RECORD
0046      CALL DPR (IPTR15,ADR3,IPIVOT,NMAXX*4.)
0047      NSUR3 = 41
0048      CALL DPW (IPTR18,0.,NPROG1,6.)
0049      WRITE (2,*) 'READING NEW PARAMETERS DONE'
      IF (NREDO.EQ.1) GO TO 110

```

```

0050      NSUB3 = 42
0051      CALL DPW (IPTR18,0,,NPROG1,6.)
0052      NREAD = (NPT-1)/NROWS+1
0053      DO 100 I=1,NREAD
0054      NLEFT = NPT-(I-1)*NROWS
0055      IF (NLEFT.GT.NROWS) NLEFT = NROWS
0056      ADR = (I-1)*NROWS*3.
0057      WRITE (2,*) 'I,ADR,NLEFT',I,ADR,NLEFT
0058      IF (NLEFT.LE.0) GO TO 110
0059      CALL DPR (IPTR17,ADR,AST,768,*NLEFT)
0060      CALL DPW (IPTR12,ADR,AST,768,*NLEFT)
0061      DO 390 LL=NBEG,NPT
0062      CALL SSWTCH (1,ISWT)
0063      IF (ISWT.EQ.1) WRITE (2,*) ' ROW',LL,' LEFT',NPT-LL
0064      CALL SSWTCH (0,ISWT)
0065      IF (ISWT.NE.1) GO TO 120
0066      WRITE (2,*) 'CHECKTHESWITCH"0"ITIS"1".IFERRORTYPE0'
0067      WRITE (2,*) 'OTHERWISETYPE1TOSTOP'
0068      READ (2,*) ISWT
0069      IF (ISWT.NE.1) GO TO 120
0070      NNTIME = NTIME
0071      NNSTOR = NSTORE
0072      CALL DPW (IPTR16,0,,LL,6150.)
0073      ADRS = 16.*NTIME*RECORD
0074      CALL DPW (IPTR15,ADRS,PIVOT,NMAXX*4.)
0075      NSUB3 = 41
0076      CALL DPW (IPTR18,0,,NPROG1,6.)
0077      WRITE (2,*) 'RESTARTPARAMETERSWRITTENONDISK'
0078      STOP
0079      120  NC = 9999
0080      DO 130 I=1,NPT
0081      IC = IBANDW(I)
0082      IF (IC.LE.0) GO TO 130
0083      IF (IC.GE.NC) GO TO 130
0084      MINROW = I
0085      NC = IC
0086      130  CONTINUE
0087      M = MINROW
0088      ADR = (M-1)*3.
0089      CALL DPR (IPTR12,ADR,AA,764.)
0090      NNCOL(NC+1) = 0
0091      A1 = 0.0
0092      DO 140 J=1,NC
C          AAA = DABS(AA(J))
0093      AAA = ABS(AA(J))
0094      IF (AAA.LT.A1) GO TO 140
0095      A1 = AAA
0096      IY = J
0097      140  CONTINUE
0098      MAXCOL = NNCOL(IY)
0099      X = AA(IY)
0100      IF (X.NE.0.0) GO TO 150
0101      WRITE (2,*) 'UPT X=0 AT LL',LL,'MINROW',MINROW,'NCOL',(NNCOL(IK),
+ A(IK),IK=1,NC),'MAXCOL',MAXCOL
0102      STOP
0103      150  DO 160 J=1,NC

```

```

0104      160      AA(J) = AA(J)/X
0105              AA(IY) = 1.0
0106              NSTORE = NSTORE+1
0107              IPIVOT(NSTORE) = M
0108              IOPROW(NSTORE) = M
0109              COEF(NSTORE) = X
0110              IF (NSTORE.LT.NMAXX) GO TO 170
0111              ADRS = 16, +NTIME*RECORD
0112              NTIME = NTIME+1
0113              NSTORE = 0
0114              CALL DPW (IPTR15,ADRS,IPIVOT,NMAXX*4,)
0115      170      ADR = (NPT-LL)*3,
0116              NP1 = NPT-LL
0117              CALL DPW (IPTR13,ADR,AA,768,)
0118              LPRINT = LPRINT+1
0119              IF (LPRINT.LT.LIMIT) GO TO 200
0120              LPRINT = 0
0121              NCOUNT = NCOUNT+1
0122              MORE = NPT-LL
0123              WRITE (6,180) LL,MORE,MAXCOL,NC,M,MAXWID,NTIME,(NNCOL(IJ),AA(IJ),
+IJ=1,NC)
0124      180      FORMAT (' LL',I5,' LEFT',I5,' MAXCOL',I5,' IBAND',I5,' MINROW',I5,
+ ' MAXWID',I5,' NTIME',I5/,12(8(I5,E10,2)/))
0125              IF (NCOUNT.LT.10) GO TO 200
0126              CLOSE (UNIT=6)
0127              IF (NPRINT.NE.0) WRITE (2,*) 'ROW OPERATED',LL,'ROWS LEFT',MORE,'N
+TIME',NTIME
0128              NCOUNT = 0
0129              NREAD = (NPT-1)/NROWS+1
0130              DO 190 LK=1,NREAD
0131              ADR = (LK-1)*NROWS*3,
0132              NLEFT = NPT-(LK-1)*NROWS
0133              IF (NLEFT.GT.NROWS) NLEFT = NROWS
0134              CALL DPR (IPTR12,ADR,AST,768,*NLEFT)
0135      190              CALL DPW (IPTR17,ADR,AST,768,*NLEFT)
0136              NNTIME = NTIME
0137              NNSTOR = NSTORE+1
0138              IF (NNSTOR.LT.0) PRINT *, ' **** NNSTOR NEGATIVE NREAD=2 NOT GOOD'
0139              CALL DPW (IPTR16,0,,LL,6150,)
0140              ADRS = 16, +NTIME*RECORD
0141              CALL DPW (IPTR15,ADRS,IPIVOT,NMAXX*4,)
0142              NSUB3 = 42
0143              CALL DPW (IPTR18,0,,NPROG1,6,)
0144      200              IBANDW(M) = 0
0145              IF (LL.EQ.NPT) GO TO 390
0146              LL1 = NSTART(MAXCOL)
0147              LAST = NSTOPP(MAXCOL)
0148              L = LL1-1
0149      210              L = L+1
0150              IF (L.GT.LAST) GO TO 380
0151              IF (IBANDW(L).EQ.0) GO TO 210
0152              ADR = (L-1)*3,
0153              NLEFT = LAST+1-L
0154              IF (NLEFT.GT.NROWS) NLEFT = NROWS
0155              RWORDS = 768,*NLEFT
0156              L = L-1

```

UPPERT,FTN /TRIRLOCKS/WR

```

      C      WRITE(6,*)'NLEFT,RWORDS,ADR',NLEFT,RWORDS,ADR
0157      CALL DPR (IPTR12,ADR,AST,RWORDS)
0158      NWRITE = 0
0159      DO 370 IROW=1,NLEFT
0160      L = L+1
0161      I = L
0162      NC = IBANDW(I)
0163      IF (NC.EQ.0) GO TO 370
0164      IBASEA = (IROW-1)*384
0165      IBASEN = (IROW-1)*768+512
0166      DO 220 LT=1,NC
0167      NCOL(LT) = IST(IBASEN+LT)
0168      220  A(LT) = AST(IBASEA+LT)
0169      NCOL(NC+1) = 0
0170      DO 340 J=1,NC
0171      IF (NCOL(J)=MAXCOL) 340,230,370
0172      230  NOPROW = I
0173      NWRITE = 1
0174      JKOP = 1
0175      JKPI = 1
0176      C = -A(J)
0177      NSTORE = NSTORE+1
0178      IPIVOT(NSTORE) = M
0179      IOPROW(NSTORE) = I
0180      COEF(NSTORE) = C
0181      IF (NSTORE.LT.NMAXX) GO TO 240
0182      ADRS = 16,NTIME*RECORD
0183      NTIME = NTIME+1
0184      NSTORE = 0
0185      CALL DPR (IPTR13,ADRS,IPIVOT,NMAXX*4,)
0186      240  CONTINUE
0187      IF (NNCOL(JKPI).EQ.0) GO TO 350
0188      IF (NCOL(JKOP).EQ.0) GO TO 250
0189      IF (NNCOL(JKPI)=NCOL(JKOP)) 250,270,330
0190      250  II = IBANDW(I)+1
0191      NCOL(II+1) = 0
0192      IF (MAXWID.LT,II) MAXWID = II
0193      IBANDW(I) = II
0194      IF (MAXWID.GT,NNN) GO TO 420
0195      N9 = NNCOL(JKPI)
0196      IF (NSTART(N9).GT,I) NSTART(N9) = I
0197      IF (NSTOPP(N9).LT,I) NSTOPP(N9) = I
0198      JKL = JKOP+1
0199      260  IX = II-1
0200      A(IX) = A(IX)
0201      NCOL(IX) = NCOL(IX)
0202      II = IX
0203      IF (IX.GE,JKL) GO TO 260
0204      A(JKOP) = AA(JKPI)*C
0205      NCOL(JKOP) = NNCOL(JKPI)
0206      IX = NCOL(JKOP)
0207      GO TO 320
0208      270  IX = NCOL(JKOP)
0209      IF (IX.EQ,MAXCOL) GO TO 280
0210      X = AA(JKPI)*C+A(JKOP)
0211      A(JKOP) = X

```

```

0212      ATEST = ABS(X)-ZTEST
          C      ATEST = DABS(X) - ZTEST
0213      IF (ATEST.GT.0.) GO TO 320
0214      200  IBANDW(NOPROW) = IBANDW(NOPROW)-1
0215      IF (IBANDW(NOPROW)) 290,290,300
0216      290  PRINT 440, MINROW,MAXCOL,NOPROW
0217      STOP
0218      300  IX = IBANDW(NOPROW)
0219      DO 310 NK=JKOP,IX
0220      A(NK) = A(NK+1)
0221      310  NCOL(NK) = NCOL(NK+1)
0222      IX = IX+1
0223      NCOL(IX) = 0
0224      A(IX) = 0.0
0225      JKPI = JKPI+1
0226      GO TO 240
0227      320  JKPI = JKPI+1
0228      330  JKOP = JKOP+1
0229      GO TO 240
0230      340  CONTINUE
0231      350  NC = IBANDW(I)
0232      DO 360 LT=1,NC
0233      IST(IBASE+LT) = NCOL(LT)
0234      360  AST(IBASE+LT) = A(LT)
0235      370  CONTINUE
0236      IF (NWRITE.EQ.1) CALL DPW (IPTR12,ADR,AST,768,*NLEFT)
0237      GO TO 210
0238      380  CONTINUE
0239      390  CONTINUE
0240      BALL = NTIME
0241      BALL = BALL*NMAXX+NSTORE
0242      CALL DPW (IPTR1,0,,LNPT,90,)
0243      IF (NPRINT.GE.2) WRITE (6,400) BALL
0244      400  FORMAT (' TOTAL NUMBER OF B STORE',F10.0)
0245      IF (NSTORE.EQ.0) GO TO 410
0246      ADRS = 16.*NTIME*RECORD
0247      CALL DPW (IPTR15,ADRS,PIVOT,NMAXX*4,)
0248      410  IF (NPRINT.GE.1) WRITE (6,460) MAXWID
0249      GO TO 430
0250      420  PRINT 450, I,LL
0251      STOP
0252      430  NSUB3 = 4
0253      CALL DPW (IPTR10,0,,NPROG1,6,)
0254      IF (NPRINT.NE.0) WRITE (2,*) 'UPPER TRIANGLE DECOMPOSITION DONE'
0255      RETURN
0256      440  FORMAT (120('#')/5X,'MATRIX SINGULAR',//5X,'MINROW',I5,' MAXCOL',I5
+,' NOPROW',I5' PROGRAM STOPPED',//120('#')//)
0257      450  FORMAT (215,'DIMEN OF A , NCOL AND ASSOCIATED PARAMETERS EXCEEDED'
+)
0258      460  FORMAT (' MAXWID',I10)
0259      END

```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
------	------	------------

SCODE1	006744	1778	RW,I,CON,LCL
SPDATA	000510	164	RW,D,CON,LCL
SIDATA	000634	206	RW,D,CON,LCL
SVAR8	000164	58	RW,D,CON,LCL
STEMPS	000020	8	RW,D,CON,LCL
POINT	000042	17	RW,D,OVR,GBL
TRACE	000014	6	RW,D,OVR,GBL
HDR	000274	94	RW,D,OVR,GBL
UPPERT	003000	768	RW,D,OVR,GBL
BSTORE	014000	3072	RW,D,OVR,GBL
SYST	002770	764	RW,D,OVR,GBL
START	030014	6150	RW,D,OVR,GBL
STORE	036000	7680	RW,D,OVR,GBL

TOTAL SPACE ALLOCATED = 121072 20765

,LP:=UPPERT

```

0001      SUBROUTINE MREADS (NPT,NPRINT)
0002      COMMON /POINT/IPTR1,IPTR2,IPTR3,IPTR4,IPTR5,IPTR6,IPTR7,IPTR8,
+ IPTR10,IPTR11,IPTR12,IPTR13,IPTR14,IPTR15,IPTR16,IPTR17,IPTR18
0003      COMMON/TRACE/NPROG1,NPROG2,NBAND,NPROG3,NPROG4,NPROG5,NSUB3
0004      DIMENSION NCOL(128),A(128)
0005      DO 100 I=1,NPT
0006          ADR = I*2-1
0007          CALL DPR (IPTR10,ADR,NCOL,384,)
0008          ADR = (I-1)*6,
0009          CALL DPW (IPTR12,ADR,A,256,)
0010          ADR = ADR+4,
0011      100 CALL DPW (IPTR12,ADR,NCOL,128,)
0012          NSUB3 = 3
0013          CALL DPW (IPTR18,0,,NPROG1,6,)
0014          IF (NPRINT.NE.0) WRITE (2,*) 'MREAD SUBROUTINE COMPLETED'
0015          RETURN
0016      END

```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES	
SCODE1	000256	87	RW,I,CON,LCL
SPDATA	000060	24	RW,D,CON,LCL
SIDATA	000050	20	RW,D,CON,LCL
SVAR8	001406	387	RW,D,CON,LCL
STEMPS	000002	1	RW,D,CON,LCL
POINT	000042	17	RW,D,OVR,GBL
TRACE	000014	6	RW,D,OVR,GBL

TOTAL SPACE ALLOCATED = 002074 542

,LP:=MREADS


```

C      UPERTB.FTN
0001      SUBROUTINE UPERTB (NPRINT,NREDO)
0002      COMMON /POINT/IPTR1,IPTR2,IPTR3,IPTR4,IPTR5,IPTR6,IPTR7,IPTR89 ,
      + IPTR10,IPTR11,IPTR12,IPTR13,IPTR14,IPTR15,IPTR16,IPTR17,IPTR18
0003      COMMON/TRACE/NPROG1,NPROG2,NBAND,NPROG3,NPROG3,NSUB3
0004      INCLUDE 'HDR.FTN'
0005      * COMMON/HDR/LNPT,NTT,NBPTC,NBFTC,NSTREH,NPT,NSELEH,NELEH,NODMAX,
      * #NONZER,NQD,MATN,KTYPE,NMAX,NSTIF,NTIME,NSTORE,NDUM,TITLE(20),
      * #DP1,DP2,DP3,DP4,DP5,DP6,DP7,DP89,DP10,DP11,DP12,DP13,DP14,DP15,
      * #DP16,DP17,DP18,NTSURF,NE9URF
0006      COMMON/UPPERT/AA(256),NMCOL(252),NC,MAXCOL,MINROW,NFAK
0007      COMMON/BSTORE/IPIVOT(768),IOPROW(768),COEF(768)
0008      COMMON/SYST/A(512),NCOL(512)
0009      COMMON/START/LL,MAXWID,NNTIME,IBAND,NNSTOR,NDU,IBANDW(2048),
      + NSTOPP(2048),NSTART(2048)
0010      COMMON/STORE/AST(3072)
0011      DIMENSION IST(6144)
0012      EQUIVALENCE(AST(1),IST(1))
0013      CALL DPR (IPTR10,0.,NPROG1,6.)
0014      NSIHS = 40
0015      CALL DPW (IPTR10,0.,NPROG1,6.)
0016      CALL DPR (IPTR1,0.,LNPT,90.)
0017      ADR = 2*NPT+1.
0018      CALL DPR (IPTR10,ADR,IBANDW,2048.)
0019      ADR = ADR+8.
0020      CALL DPR (IPTR10,ADR,NSTART,2048.)
0021      ADR = ADR+8.
0022      CALL DPR (IPTR10,ADR,NSTOPP,2048.)
C      DOUBLE PRECISION A,AA,B,X,C,AAA,A1,SAVE,ZTEST,DABS
0023      NSTORE = 0
0024      NROWS = 4
0025      LIMIT = 10000
0026      IF (NPRINT.EQ.1) LIMIT = 100
0027      IF (NPRINT.GE.2) LIMIT = 10
0028      IF (NPRINT.GE.3) LIMIT = 1
0029      NCOUNT = 0
0030      LPRINT = 0
0031      NTIME = 0
0032      ZTEST = 0.
0033      NMAXX = 768
0034      RECORD = 12.
0035      NNN = 513
0036      MAXWID = NSTIF
0037      NBEG = 1
C-----
C      THE BACK UP IS NOT USED TO FILEQ SPACE
C-----
0038      DO 130 LL=NBEG,NPT
0039      CALL SSWTCH (1,ISWT)
0040      IF (ISWT.EQ.1) WRITE (2,*) ' ROW',LL,' LEFT',NPT-LL
0041      NC = 9999
0042      DO 130 I=1,NPT
0043      IC = IRANDW(I)
0044      IF (IC.LE.0) GO TO 130
0045      IF (IC.GE.NC) GO TO 130
0046      MINROW = I

```

```

0047      NC = IC
0048      130  CONTINUE
0049          IF (NC,LE,252) GO TO 140
0050          WRITE (2,*) 'NC',NC,'LL',LL,'>THAN 252 PROGRAM STOPPED'
0051          WRITE (6,*) 'NC',NC,'LL',LL,'>THAN 252 PROGRAM STOPPED'
0052          STOP
0053      140  M = MINROW
0054          ADR = (M-1)*6.
0055          CALL DPR (IPTR12,ADR,AA,504.)
0056          CALL DPR (IPTR12,ADR+4,NNCOL,252.)
0057          NNCOL(NC+1) = 0
0058          A1 = 0.0
0059          DO 150 J=1,NC
C              AAA = DABS(AA(J))
0060          AAA = ABS(AA(J))
0061          IF (AAA,LT,A1) GO TO 150
0062          A1 = AAA
0063          IY = J
0064      150  CONTINUE
0065          MAXCOL = NNCOL(IY)
0066          X = AA(IY)
0067          IF (X,NE,0.0) GO TO 160
0068          WRITE (2,*) 'UPTX=0ATLL',LL,'MINROW',MINROW,'NCOL',(NNCOL(IK),AA(I
+K),IK=1,NC),'MAXCOL',MAXCOL
0069          STOP
0070      160  DO 170 J=1,NC
0071      170  AA(J) = AA(J)/X
0072          AA(IY) = 1.0
0073          NSTORE = NSTORE+1
0074          IPIVOT(NSTORE) = M
0075          IOPROW(NSTORE) = M
0076          COEF(NSTORE) = X
0077          IF (NSTORE,LT,NMAXX) GO TO 180
0078          ADRS = 16. + NTIME*RECORD
0079          NTIME = NTIME+1
0080          NSTORE = 0
0081          CALL DPW (IPTR13,ADRS,IPIVOT,NMAXX*4.)
0082      180  ADR = (NPT-LL)*3.
0083          NPI = NPT-LL
0084          CALL DPW (IPTR13,ADR,AA,768.)
0085          LPRINT = LPRINT+1
0086          IF (LPRINT,LT,LIMIT) GO TO 200
0087          LPRINT = 0
0088          NCOUNT = NCOUNT+1
0089          MORE = NPT-LL
0090          WRITE (6,190) LL,MORE,MAXCOL,NC,M,MAXWID,NTIME,(NNCOL(IJ),AA(IJ),I
+J=1,NC)
0091      190  FORMAT (' LL',I5,' LEFT',I5,' MAXCOL',I5,' IBAND',I5,' MINROW',I5,
+' MAXWID',I5,' NTIME',I5/,12(8(I5,E10,2)/))
0092          IF (NCOUNT,LT,10) GO TO 200
0093          CLOSE (UNIT*6)
0094          IF (NPRINT,NE,0) WRITE (2,*) 'ROW OPERATED',LL,'ROWS LEFT',MORE,'N
+TIME',NTIME,'BAND',NC
0095          NCOUNT = 0
0096          NNTIME = NTIME
0097          NNSTOR = NSTORE-1

```

```

0098      IF (NNSTOR.LT.0) PRINT *, ' **** NNSTOR NEGATIVE NREDO=2 NOT GOOD'
0099      CALL DPW (IPTR16,0,,LL,6150,)
0100      AORS = 16,+NTIME*RECORD
0101      CALL DPW (IPTR15,AORS,IPIVOT,NMAXX*4,)
0102      NSUR3 = 02
0103      CALL DPW (IPTR16,0,,NPROG1,6,)
0104      200  IRANDW(M) = 0
0105           IF (LL.EQ.NPT) GO TO 390
0106           LL1 = NSTART(MAXCOL)
0107           LAST = NSTOPP(MAXCOL)
0108           L = LL1-1
0109      210  L = L+1
0110           IF (L.GT.LAST) GO TO 380
0111           IF (IRANDW(L).EQ.0) GO TO 210
0112           ADR = (L-1)*6,
0113           NLEFT = LAST+1-L
0114           IF (NLEFT.GT.NROWS) NLEFT = NROWS
0115           RWORDS = 1536,*NLEFT
0116           L = L-1
0117           CALL DPR (IPTR12,ADR,AST,RWORDS)
0118           NWRITE = 0
0119           DO 370 IROW=1,NLEFT
0120           L = L+1
0121           I = L
0122           NC = IRANDW(I)
0123           IF (NC.EQ.0) GO TO 370
0124           IBASEA = (IROW-1)*768
0125           IBASEN = (IROW-1)*1536+1024
0126           DO 220 LT=1,NC
0127           NCOL(LT) = 1ST(IBASEN+LT)
0128      220  A(LT) = AST(IBASEA+LT)
0129           NCOL(NC+1) = 0
0130           DO 340 J=1,NC
0131           IF (NCOL(J)=MAXCOL) 340,230,370
0132      230  NOPROW = I
0133           NWRITE = 1
0134           JKOP = 1
0135           JKPI = 1
0136           C = -A(J)
0137           NSTORE = NSTORE+1
0138           IPIVOT(NSTORE) = M
0139           IOPROW(NSTORE) = I
0140           COEF(NSTORE) = C
0141           IF (NSTORE.LT.NMAXX) GO TO 240
0142           AORS = 16,+NTIME*RECORD
0143           NTIME = NTIME+1
0144           NSTORE = 0
0145           CALL DPW (IPTR15,AORS,IPIVOT,NMAXX*4,)
0146      240  CONTINUE
0147           IF (NNCOL(JKPI).EQ.0) GO TO 350
0148           IF (NCOL(JKOP).EQ.0) GO TO 250
0149           IF (NNCOL(JKPI)=NCOL(JKOP)) 250,270,330
0150      250  II = IRANDW(I)+1
0151           NCOL(II+1) = 0
0152           IF (MAXWID.LT.II) MAXWID = II
0153           IRANDW(I) = II

```

```

0154      IF (MAXWID.GT,NNN) GO TO 420
0155      N9 = NNCOL(JKPI)
0156      IF (NSTART(N9).GT,I) NSTART(N9) = I
0157      IF (NSTOPP(N9).LT,I) NSTOPP(N9) = I
0158      JKL = JKOP+1
0159  260    IX = II-1
0160      A(II) = A(IX)
0161      NCOL(II) = NCOL(IX)
0162      II = IX
0163      IF (IX.GE,JKL) GO TO 260
0164      A(JKOP) = AA(JKPI)*C
0165      NCOL(JKOP) = NNCOL(JKPI)
0166      IX = NCOL(JKOP)
0167      GO TO 320
0168  270    IX = NCOL(JKOP)
0169      IF (IX.EQ,MAXCOL) GO TO 280
0170      X = AA(JKPI)*C+A(JKOP)
0171      A(JKOP) = X
0172      ATEST = ABS(X)-ZTEST
      C      ATEST = DABS(X) - ZTEST
0173      IF (ATEST.GT,0.) GO TO 320
0174  280    IBANDW(NOPROW) = IBANDW(NOPROW)+1
0175      IF (IBANDW(NOPROW)) 290,290,300
0176  290    PRINT 440, MINROW,MAXCOL,NOPROW
0177      STOP
0178  300    IX = IBANDW(NOPROW)
0179      DO 310 NK=JKOP,IX
0180      A(NK) = A(NK+1)
0181  310    NCOL(NK) = NCOL(NK+1)
0182      IX = IX+1
0183      NCOL(IX) = 0
0184      A(IX) = 0.0
0185      JKPI = JKPI+1
0186      GO TO 240
0187  320    JKPI = JKPI+1
0188  330    JKOP = JKOP+1
0189      GO TO 240
0190  340    CONTINUE
0191  350    NC = IBANDW(I)
0192      DO 360 LT=1,NC
0193      IST(IBASEN+LT) = NCOL(LT)
0194  360    AST(IBASEA+LT) = A(LT)
0195  370    CONTINUE
0196      IF (NWRITE.EQ,1) CALL DPW (IPTR12,ADR,AST,1536,*NLEFT)
0197      GO TO 210
0198  380    CONTINUE
0199  390    CONTINUE
0200      BALL = NTIME
0201      BALL = BALL*NMAXX+NSTORE
0202      CALL DPW (IPTR1,0.,LNPT,90.)
0203      IF (NPRINT.GE,2) WRITE (6,400) BALL
0204  400    FORMAT (' TOTAL NUMBER OF B STORE',F10.0)
0205      IF (NSTORE.EQ,0) GO TO 410
0206      ADRS = 16.*NTIME+RECORD
0207      CALL DPW (IPTR15,ADRS,PIVOT,NMAXX*4,)
0208  410    IF (NPRINT.GE,1) WRITE (6,460) MAXWID

```

```

0209      GO TO 430
0210      420  PRINT 450, I,LL
0211          WRITE (2,450) I,LL
0212          STOP
0213      430  NSUR3 = 4
0214          CALL OPW (IPTR10,0,,NPROG1,6.)
0215          IF (NPRINT.NE.0) WRITE (2,*) 'UPPER TRIANGLE DECOMPOSITION DONE'
0216          RETURN
0217      440  FORMAT (120('#')/5X,'MATRIX SINGULAR'/,5X,'MINROW',I5,' MAXCOL',I5
+,' NOPROW',I5' PROGRAM STOPPED'/,120('#')//)
0218      450  FORMAT (' ROW OPERATED',I4,' ROWS DONE',I4,' A AND NCOL DIMENSION
+EXCEEDED ***** PROGRAM STOPPED DUE TO ABOVE')
0219      460  FORMAT (' MAXWID',I10)
0220      END

```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODE1	005630 1484	RW,I,CON,LCL
SPDATA	000342 113	RW,D,CON,LCL
SIDATA	000704 226	RW,D,CON,LCL
SVARS	000160 56	RW,D,CON,LCL
STEMPS	000020 8	RW,D,CON,LCL
POINT	000042 17	RW,D,OVR,G8L
TRACE	000014 6	RW,D,OVR,G8L
HDR	000274 94	RW,U,OVR,G8L
UPPERT	003000 768	RW,D,OVR,G8L
BSTORE	014000 3072	RW,D,OVR,G8L
SYST	006000 1536	RW,D,OVR,G8L
START	030014 6150	RW,D,OVR,G8L
STORE	030000 6144	RW,U,OVR,G8L

TOTAL SPACE ALLOCATED = 114664 19674

,LP1=UPERTB.

```

C      BVALUE,FTN
0001      SUBROUTINE BVALUE (IT,NQWCP,NQ,NBPTC,NSTREH,NMAX,NPT,NPRINT,NELEM
      +,NSELEM,NSTEDY,LNPT,NBPTC,NHBOUND)
C-----
C      B VALUES ARE ESTIMATED
C-----
0002      COMMON /POINT/IPTR1,IPTR2,IPTR3,IPTR4,IPTR5,IPTR6,IPTR7,IPTR89 ,
      + IPTR10,IPTR11,IPTR12,IPTR13,IPTR14,IPTR15,IPTR16,IPTR17,IPTR18
0003      COMMON/TRACE/NPROG1,NPROG2,NBAND,NPRO3I,NPROG3,NBUB3
0004      COMMON/ELEMAT/MN,NE,MM,NHEAD,AREA,SUMQW,NOD(20),ZY(20),XIDLEM(20),
      + XIJCON(20),CI(20,20),PPIJ(20,20)
0005      COMMON /KXYZ/ XK(50),YK(50),ZK(50)
0006      DIMENSION BIVF(128),STREAM(126),NFLUX(768),LENGTH(768),ELEV(768),
      +NODBF(128),Z(256),B(2048),HEAD(256),BIV(768),BIVP(768),
      +IDENT(768),NODE(2048),SUMQ(2560),Q(768),HEAD0(20),NODB(768)
0007      EQUIVALENCE (LENGTH,NODB),(ELEV,BIV),(SUMQ,Z),(Q,BIVP)
0008      CALL DPR (IPTR10,0.,NPROG1,6.)
0009      CALL DPR (IPTR1,17.,NODE,2048.)
0010      CALL DPR (IPTR1,10.,IDENT,768.)
0011      CALL DPR (IPTR7,9.,NFLUX,768.)
0012      CALL DPR (IPTR7,26.,NODBF,128.)
0013      CALL DPR (IPTR7,27.,BIVF,256.)
0014      CALL DPR (IPTR7,0.,NODB,768.)
0015      IF (NSTEDY.NE.0) GO TO 100
0016      CALL DPR (IPTR7,3.,BIV,768.)
0017      GO TO 120
C-----
C      SINCE SIMULATION FOR CHANGE IS DONE BIV IS READ ONLY
C      IF THERE IS VARIABLE HELD B,C.
C-----
0018      100 IF (NHRUND.EQ.0) GO TO 120
0019      WORD8 = NBPTC*2.
0020      IF (IT.EQ.1) CALL DPR (IPTR7,3.,BIVP,WORD8)
0021      IF (IT.GT.1) CALL DPR (IPTR5,(IT-2)*6.+1021.,BIVP,WORD8)
0022      CALL DPR (IPTR5,(IT-1)*6.+1021.,BIV,WORD8)
0023      DO 110 I=1,NBPTC
0024      BIV(I) = BIV(I)-BIVP(I)
0025      110 BIVP(I) = 0.
0026      120 IF (NPRINT.LT.2) GO TO 130
0027      WRITE (6,*) 'NODBINBVALUE',NBPTC,(NODB(I),BIV(I),I=1,NBPTC)
0028      WRITE (6,*) 'NBPTC',NBPTC,(NODBF(I),BIVF(I),I=1,NBPTC)
0029      130 NRHH = 0
0030      NMAXH = 256
0031      ADRH = 20.
0032      CALL DPR (IPTR89,ADRH,HEAD,512.)
0033      GO TO (170,140,150), NQ
0034      140 ADR = 1401.
0035      GO TO 160
0036      150 ADR = (IT-1)*6.+1401.
0037      160 CALL DPR (IPTR5,ADR,Q,1536.)
0038      IF (NPRINT.GE.1) PRINT 490, (I,Q(I),I=1,NSELEM)
0039      170 GO TO (210,180,190), NQWCP
0040      180 ADR = 1.
0041      GO TO 200
0042      190 ADR = (IT-1)*20.+1.
0043      200 CALL DPR (IPTR5,ADR,SUMQ,5120.)

```

```

C-----
C      START OF ELEMENT LOOP
C-----
0044 210 DO 300 IX=1,NELEM
0045     ADR = (IX-1)*7.
0046     CALL DPR (IPTR6,ADR,NN,1740.)
0047     IF (SUMQW,NE,0.0,AND,NPRINT,NE,0) PRINT *, 'NE',NE,'SUMQW',SUMQW
0048     IF (NSTEDY,EQ,0) GO TO 240

C-----
C      FOR TRANSIENT CASE THE ASSOCIATED HEAD VALUES ARE READ FROM DISK
C-----
0049     DO 230 I=1,NN
0050     K = NOD(I)
0051     NRH = (K-1)/NMAXH
0052     IF (NRH,EQ,NRHH) GO TO 220
0053     ADRH = 20.+NRH*2.
0054     CALL DPR (IPTR89,ADRH,HEAD,512.)
0055     NRHH = NRH
0056 220 KK = K-NRH*NMAXH
0057     HEAD(I) = HEAD(KK)
0058 230 CONTINUE
0059 240 IF (NHEAD,EQ,0) GO TO 260

C-----
C      ELEMENT HAS HELD POTENTIAL NODES
C-----
0060     DO 270 I=1,NN
0061     NRQ = NOD(I)
0062     DO 260 K=1,NBPTC
0063     KK = NOD(K)
0064     IF (NRQ,NE,KK) GO TO 260
0065     BIVD = BIV(K)
0066     ZY(I) = BIVD
0067     DO 250 M=1,NN
0068 250 ZY(M) = ZY(M)-CI(M,I)*BIVD
0069     GO TO 270
0070 260 CONTINUE
0071 270 CONTINUE
0072 280 DO 370 I=1,NN
0073     K = NOD(I)
0074     DO 290 J=1,NPT
0075     IF (K,NE,NODE(J)) GO TO 290
0076     GO TO 300
0077 290 CONTINUE
0078     GO TO 370
0079 300 PIJ = 0.0

C-----
C      TRANSIENT CASE - TERMS ASSOCIATED WITH PREVIOUS TIME STEPS
C      CALCULATED AND BROUGHT ON RIGHT HAND SIDE
C-----
0080     IF (NSTEDY,EQ,0) GO TO 320
0081     DO 310 L=1,NN
0082 310 PIJ = PIJ+PPIJ(I,L)*HEAD(L)
0083     ZY(I) = ZY(I)+PIJ
0084 320 IF (NE,GT,999) GO TO 330
0085     IF (NQ,EQ,1) GO TO 330
0086     ZY(I) = ZY(I)+Q(NE)*XIDLEM(I)

```

```

0087 330 GO TO (340,350,360), NQWCP
0088 340 IF (SUMQW.EQ.0,0) GO TO 360
0089 ZY(I) = ZY(I)-SUMQW*XIJCOB(I)
0090 GO TO 360
0091 350 ZY(I) = ZY(I)-SUMQ(IX)*XIJCOB(I)
0092 360 B(J) = B(J)+ZY(I)
0093 370 CONTINUE
0094 IF (NPRINT.GE.2.AND.IX.LT.6) PRINT *, 'ZY ',(I,ZY(I),I=1,NN)
0095 380 CONTINUE
0096 IF (NPRINT.GE.2) WRITE (6,*) 'B AFTER ELEMAT'
0097 IF (NPRINT.GE.2) WRITE (6,480) (B(I),I=1,NPT)
0098 IF (NBPTC.EQ.0) GO TO 420
0099 DO 410 I=1,NBPTC
0100 K = NONBF(I)
0101 DO 390 J=1,NPT
0102 IF (K.EQ.NODE(J)) GO TO 400
0103 390 CONTINUE
0104 GO TO 410
0105 400 B(J) = B(J)+BIVF(I)
0106 410 CONTINUE
0107 420 IF (NSTREH.EQ.0) GO TO 470
0108 CALL DPR (IPTR2,0,,KK,380,)
0109 CALL DPR (IPTR7,12,,LENGTH,768,)
0110 CALL DPR (IPTR7,15,,ELEV,1536,)
0111 CALL DPR (IPTR89,20,,HEAD,512,)
0112 CALL DPR (IPTR3,60,,Z,512,)
0113 NMM = 0
0114 DO 460 I=1,LNPT
0115 IF (NFLUX(I).EQ.0) GO TO 460
0116 J = IDENT(I)
0117 DO 430 K=1,NPT
0118 IF (NODE(K).EQ.J) GO TO 440
0119 430 CONTINUE
0120 GO TO 460
0121 440 NM = (J-1)/NMAX
0122 IF (NM.EQ.NMM) GO TO 450
0123 NMM = NM
0124 ADR = NM*2.
0125 CALL DPR (IPTR89,ADR+20,,HEAD,512.)
0126 CALL DPR (IPTR3,ADR+60,,Z,512.)
0127 450 M = J-NM*NMAX
0128 HEAD(M) = HEAD(M)
0129 IF (HEAD(M).LE.ELEV(I)) GO TO 460
0130 QF = LENGTH(I)*ZK(1)*(HEAD(M)-ELEV(I))/(ELEV(I)-Z(M))
0131 IF (QF.LE.0,0) GO TO 460
0132 IF (NPRINT.GE.1) WRITE (6,500) I,HEAD(M),ELEV(I),QF,LENGTH(I)
0133 B(K) = B(K)-QF
0134 460 CONTINUE
0135 470 CALL DPW (IPTR15,0,,B,4096,)
0136 IF (NPRINT.GE.2) WRITE (6,510) (B(I),I=1,NPT)
0137 NSUB3 = 5
0138 CALL DPW (IPTR18,0,,NPROG1,6.)
0139 IF (NPRINT.NE.0) WRITE (2,*) 'B ESTIMATION DONE'
0140 RETURN
0141 480 FORMAT (12E10,3)
0142 490 FORMAT (8(14,G10,3))

```


FORTRAN IV-PLUS V02-510
BVALUE,FTN /TRIBLOCKS/WR

16:10105

13-SEP-79

PAGE 4

```
0143    500    FORMAT (' NODE',I5,'HEAD=',F10.2,' ELEV.=',F10.2,' QP=',F10.2,' L
+ENGTH',I8)
0144    510    FORMAT (' B VALUES AFTER FLUX CONDITIONS CONSIDERED',//,120(12E10.3/
+))
0145                END
```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODE1	004712 1253	RW,I,CON,LCL
SPOATA	000276 95	RW,D,CON,LCL
SIOATA	000544 176	RW,D,CON,LCL
SVARS	105572 17853	RW,D,CON,LCL
STEMPS	000022 9	RW,D,CON,LCL
POINT	000042 17	RW,D,OVR,GBL
TRACE	000014 6	RW,D,OVR,GBL
ELEMAT	006650 1748	RW,D,OVR,GBL
KXYZ	001130 300	RW,D,OVR,GBL

TOTAL SPACE ALLOCATED = 123646 21459

,LP:=BVALUE

```

      C      BACKSU.FTN
0001      SUBROUTINE BACKSU (NPRINT)
0002      COMMON /POINT/IPTR1,IPTR2,IPTR3,IPTR4,IPTR5,IPTR6,IPTR7,IPTR8 ,
      * IPTR10,IPTR11,IPTR12,IPTR13,IPTR14,IPTR15,IPTR16,IPTR17,IPTR18
0003      COMMON/TRACE/NPROG1,NPROG2,NBAND,NPROG3,NPROG4,NSUB3
0004      INCLUDE 'HDR.FTN'
0005 *      COMMON/HDR/LNPT,NTT,NBPTC,NBFTC,NSTREM,NPT,NSELEM,NELEM,NODMAX,
      * #NONZER,NOD,MATN,KTYPE,NMAX,NSTIF,NTIME,NSTORE,NDOH,TITLE(20),
      * #DP1,DP2,DP3,DP4,DP5,DP6,DP7,DP8,DP9,DP10,DP11,DP12,DP13,DP14,DP15,
      * #DP16,DP17,DP18,NTSURF,NESURF
0006      COMMON/UPPERT/AA(256),NNCOL(252),NC,MAXCOL,MINROW,NFAK1
0007      COMMON/BSTORE/IPIVOT(768),IOPROW(768),COEF(768)
0008      DIMENSION B(2048),BA(2048)
0009      CALL DPR (IPTR18,0.,NPROG1,6.)
0010      RMAXX = 768.
0011      RECORD = 12.
0012      CALL DPR (IPTR1,0.,LNPT,90.)
0013      CALL DPR (IPTR15,0.,B,4096.)
0014      BALL = NTIME
0015      BALL = BALL*RMAXX+NSTORE
0016      IF (NPRINT.GT.1) WRITE (6,*) ' B ALL IN BACKSU',BALL,'NS',NSTORE,
      +NT',NTIME
0017      IF (NPRINT.GE.2) WRITE (6,*) ' B READ IN BACKSUB',(B(I),I=1,NPT)
0018      READD = RMAXX
0019      NTIME = NTIME+1
0020      DO 120 ITX=1,NTIME
0021      AMORE = BALL-(ITX-1)*RMAXX
0022      IF (AMORE.GT.RMAXX) AMORE = RMAXX
0023      ADR = 16.+(ITX-1)*RECORD
0024      CALL DPR (IPTR15,ADR,IPIVOT,RMAXX*4.)
0025      DO 110 IN=1,AMORE
0026      M = IPIVOT(IN)
0027      MM = IOPROW(IN)
0028      IF (M.EQ.MM) GO TO 100
0029      B(MM) = B(MM)+B(M)*COEF(IN)
0030      GO TO 110
0031      100      B(M) = B(M)/COEF(IN)
0032      110      CONTINUE
0033      120      CONTINUE
0034      IF (NPRINT.GE.2) WRITE (6,*) 'B AFTER OPERA. IN BACK',(B(I),I=1,NP
      +T)
0035      DO 150 I=1,NPT
0036      ADR = (I-1)*3.
0037      II = NPT+1-I
0038      CALL DPR (IPTR13,ADR,AA,768.)
0039      RB = R(MINROW)
0040      DO 140 J=1,NC
0041      NN = NNCOL(J)
0042      IF (NN.EQ.MAXCOL) GO TO 130
0043      RB = RB-BA(NN)*AA(J)
0044      GO TO 140
0045      130      IJ = J
0046      140      CONTINUE
0047      BA(MAXCOL) = RB
0048      150      CONTINUE
0049      CALL DPR (IPTR15,0.,BA,4096.)

```

```

0050      IF (NPRINT.EQ.2) WRITE (6,*) 'B AFTER BACKSUB',(B(I),I=1,NPT)
0051      NSUB3 = 6
0052      CALL DPW (IPTR18,0.,NPROG1,6,)
0053      IF (NPRINT.NE.0) WRITE (2,*) 'BACK SUB DONE'
0054      RETURN
0055      END
  
```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODE1	001620 456	RW,I,CON,LCL
SPDATA	000170 60	RW,D,CON,LCL
SIDATA	000074 30	RW,D,CON,LCL
SVARS	040056 215	RW,D,CON,LCL
STEMPS	000010 4	RW,D,CON,LCL
POINT	000042 17	RW,D,OVR,GBL
TRACE	000014 6	RW,D,OVR,GBL
HDR	000274 94	RW,D,OVR,GBL
UPPERT	003000 768	RW,D,OVR,GBL
RSTORE	014000 3072	RW,D,OVR,GBL

TOTAL SPACE ALLOCATED = 061544 12722

,LP1=BACKSU

```

C      **** HWRITE,FTN
0001      SUBROUTINE HWRITE (NPRINT,IT,NDIG,DELTAT,NCONS,TTIME,NWRITE,NSI1)
0002      COMMON /POINT/IPTR1,IPTR2,IPTR3,IPTR4,IPTR5,IPTR6,IPTR7,IPTR89 ,
      + IPTR10,IPTR11,IPTR12,IPTR13,IPTR14,IPTR15,IPTR16,IPTR17,IPTR18
0003      COMMON/TRACE/NPROG1,NPROG2,NBAND,NPROG3,NPROG3,NSUB3
0004      INCLUDE 'HDR,FTN'
0005      * COMMON/HDR/LNPT,NTT,NBPTC,NBFTC,NSTREM,NPT,NSELEM,NELEM,NODMAX,
      * #NONZER,NQD,MATN,KTYPE,NHAX,NSTIF,NTIME,NSTORE,NDOUM,TITLE(20),
      * #OP1,OP2,OP3,OP4,OP5,OP6,OP7,OP89,OP10,OP11,OP12,OP13,OP14,OP15,
      * #OP16,OP17,OP18,NTSURF,NESURF
0006      INCLUDE 'P31,FTN'
0007      * COMMON/P31/NSTEDY,ITOTAL,NQWCP,NQ,MNTRY,ITT,NHBOND,NFDOUM,TT(70),
      * #DETAIL(20)
0008      COMMON /PRINT/NUG(2,1024)
0009      DIMENSION NUOE(2048),NUMBER(2560),Z(2560),HEAD(2560),A(2048)
0010      EQUIVALENCE (NUOE(1),NUG(1,1))
0011      CALL DPR (IPTR18,0.,NPROG1,6.)
0012      CALL DPR (IPTR1,0.,LNPT,90.)
0013      CALL DPR (IPTR1,1.,NUMBER,2560.)
0014      CALL DPR (IPTR1,17.,NUOE,2048.)
0015      CALL DPR (IPTR3,0.,Z,5120.)
0016      CALL DPR (IPTR15,0.,A,4096.)
0017      IF (NSTEDY.NE.0) GO TO 110
0018      IF (NPRINT.GT.0) NPP = 1
0019      CALL DPR (IPTR14,0.,HEAD,NTT*2.)
0020      DO 100 I=1,NPT
0021      100      HEAD(NOE(I)) = A(I)
0022      CALL DPW (IPTR14,20.,HEAD,NTT*2.)
0023      IF (NPP.NE.0) WRITE (6,230)
0024      GO TO 130
0025      110      NPP = 0
0026      IF (NPRINT.NE.0.AND.NWRITE.EQ.1) NPP = 1
0027      CALL DPR (IPTR89,20.,HEAD,5120.)
0028      DO 120 I=1,NPT
0029      120      K = NOE(I)
0030      HEAD(K) = A(I)+HEAD(K)
0031      CALL DPW (IPTR89,20.,HEAD,5120.)
0032      IF (NWRITE.EQ.1) CALL DPW (IPTR14,IT*20.,HEAD,5120.)
0033      IF (NWRITE.EQ.1.AND.NCONS.EQ.1) IT = IT+1
0034      IF (NPP.EQ.0) GO TO 220
0035      IF (NSTEDY.EQ.0) GO TO 130
0036      WRITE (6,240) IT,TTIME,DELTAT
0037      WRITE (2,*) 'TIMESTEP',IT,'ELAPSEOTIME',TTIME
0038      130      DO 140 I=1,2048
0039      140      NUOE(I) = 0
0040      NSN = 0
0041      DO 150 I=1,NTT
0042      150      N = NUMBER(I)
0043      IF (N.LT.1000) NSN = NSN+1
0044      IF (N.LT.1000) NUG(1,NSN) = I
0045      NUG(2,NSN) = NUG(2,NSN)+1
0046      WRITE (6,250) ,TITLE,DETAIL
0047      NPAG = NSN/4
0048      IREM = NSN-NPAG*4
0049      IF (IREM.GT.0) NPAG = NPAG+1
0050      PRINT 270

```

```

0051      NLines = 19
0052      DO 210 IPAG=1,NPAG
0053      JS = (IPAG-1)*4+1
0054      INUM = 4
0055      IF (IREM.GT.0.AND.IPAG.EQ.NPAG) INUM = IREM
0056      IMX = MAX(NUG(2,IS),NUG(2,IS+1),NUG(2,IS+2),NUG(2,IS+3))
0057      IF (NLines+IMX.LE.97) GO TO 170
0058      NLines = 14
0059      IF (IPAG.NE.1) PRINT 160
0060 160    FORMAT (1H1)
0061      PRINT 250, TITLE,DETAIL
0062      PRINT 270
0063 170    CONTINUE
0064      NLines = NLines+IMX+2
0065      DO 200 J=1,IMX
0066      DO 180 I=1,INUM
0067      ISPAC = 1+(I-1)*33
0068      ISS = NUG(2,IS-1+I)
0069      IF (J.GT.ISS) GO TO 180
0070      INDX = NUG(1,IS-1+I)+J-1
0071      PRINT 260, NUMRER(INDX),Z(INDX),HEAD(INDX)
0072 180    CONTINUE
0073      PRINT 190
0074 190    FORMAT (1H )
0075 200    CONTINUE
0076      PRINT 270
0077 210    CONTINUE
0078      IF (NSTEDY.EQ.0) GO TO 220
0079 220      NSUR3 = 7
0080      CALL OPW (IPTR18,0.,NPROG1,6.)
0081      RETURN
0082 230      FORMAT (1H1//,10X,'STEADY STATE SOLUTION',//)
0083 240      FORMAT (1H1//,10X,'RESULTS OF HEAD ESTIMATION AT TIME STEP',I4,'
+ELAPSED TIME',G12.4,' DELTAT=',G12.4//)
0084 250      FORMAT (//5X,20A4,//10X,20A4//128('-',)//T5,'NODE',T12,'DEPTH',T25
+, 'HEAD',T38,'NODE',T45,'DEPTH',T58,'HEAD',T71,'NODE',T78,'DEPTH',T91
+, 'HEAD',T104,'NODE',T111,'DEPTH',T124,'HEAD',//128('-',)//)
0085 260      FORMAT (1H+,<ISPAC>X,1X,IS,1X,G11.4,1X,F11.4NDIG>)
0086 270      FORMAT (/)
0087      END

```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODEF1	002246 595	RW,I,CON,LCL
SPDATA	000076 31	RW,D,CON,LCL
SIDATA	000634 206	RW,D,CON,LCL
SVARS	102040 16912	RW,D,CON,LCL
STEMPS	000010 4	RW,D,CON,LCL
POINT	000042 17	RW,D,OVR,GRL
TRACE	000014 6	RW,D,OVR,GRL
HDR	000274 94	RW,D,OVR,GRL
P3T	000570 158	RW,D,OVR,GRL
PRINT	010000 2048	RW,D,OVR,GRL

TOTAL SPACE ALLOCATED = 116412 20101

,LP:=HWRITE

9.6 OTHER SUPPORTIVE PROGRAMS, PROGRAM LISTINGS

FORTRAN IV-PLUS V02-510
PLOTTEL,FTN /TRIALLOCKS/WR

16111116

13-SEP-79

PAGE 1

```

C      PLOTTEL,FTN
C-----
C      MAIN TO PLOT THE NODES,ELEMENTS,LOGS, AND HEAD
C-----
0001      BYTE INOUT(768)
0002      DIMENSION WT(768),TITLE(16),YD(2),XD(2),FILE(7)
0003      WRITE (2,*) 'HEAD TO BE PLOTTED (1=YES,0=NO) ',NPRINT'
0004      READ (2,*) NHEAD,NPRINT
0005      IF (NHEAD.EQ.1) CALL HREAD (WT,ZMAX,BASE,TITLE,NPRINT,INOUT)
0006      WRITE (2,*) 'ENTER PLOTTING DEVICE 0=TG 1=XY 2=DISK'
0007      READ (2,*) IDEV
0008      IF (IDEV.EQ.0) CALL ASNLUN (3,'TG',0,ISTAT)
0009      IF (IDEV.EQ.1) CALL ASNLUN (3,'XY',0,ISTAT)
0010      IF (IDEV.NE.2) GO TO 100
0011      CALL ASNLUN (3,'DP',0,ISTAT)
0012      WRITE (2,*) 'ENTER PLOT FILE NAME'
0013      READ (2,110) FILE
0014      N = ICHR(FILE)
0015      OPEN (UNIT=3,NAME=FILE,TYPE='NEW',FORM='UNFORMATTED')
0016  100  CALL PLOTS (0,0,3)
0017      CALL LINES (1,1,1,2,XD,YD)
0018      CALL NUMBER (X,YP,SZNOO,AL,0,0,-1)
0019      CALL SYMBOL (X,YP,,07,15,0,0,-1)
0020      CALL PLMAIN (NHEAD,WT,NPRINT,INOUT)
0021      STOP
0022  110  FORMAT (20A4)
0023      END

```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODE1	000440 144	RW,I,CON,LCL
SPDATA	000226 75	RW,D,CON,LCL
SIDATA	000212 69	RW,D,CON,LCL
SVARS	007616 1991	RW,D,CON,LCL

TOTAL SPACE ALLOCATED = 010716 2279

NO FPP INSTRUCTIONS GENERATED

,LP:=PLOTTEL

```

0001      C      PLMAIN,FTN
          SUBROUTINE PLMAIN (NHEAD,HFAD,NPRINT,INOUT)
          C-----
          C      TO PROVIDE MEANS TO CHECK CORRECTNESS OF INPUT DATA USED IN THREE-
          C      DIMENSIONAL FORMULATION OF LARGE MULTILAYERED GROUND WATER SYSTEM
          C      THIS PROGRAM IS DEVELOPED TO PLOT
          C
          C      1. NODE LOCATIONS
          C      2. TWO DIMENSIONAL SURFACE ELEMENT
          C      3. VERTICAL LOGS AT EACH NODE. DOTTED PART OF THE LOG REPRESENTS
          C      LOCATION OF BED IN SEA.
          C      VARIOUS HYDROLOGIC UNITS ENCOUNTERED ARE MARKED BY HORIZONTAL
          C      TICK MARK AND INTEGER VALUES BETWEEN TICK MARKS IS THE
          C      IDENTIFICATION NUMBER OF THE MATERIAL.
          C-----
0002      RYTE INOUT(768),TITLE
0003      DIMENSION XX(768),YY(768),NOD(16),IN(16),MIXNOD(4),ZZD(20),
          +XMIN(500),YMIN(500),NAME(500),HCC(2),LL(20),ZZ(20),YD(2),A(2),
          + CONCEN(20),FILE(7),XD(2),HEAD(768),LIST(400),LOGST(400)
0004      REAL NINE32,NINEA
          C
0005      COMMON /BOX/ XMIN,YMIN,XMAX,YMAX,SF,LOGS,ISO,ZMAX,FACTZ,TITLE(80)
0006      WRITE (2,*) 'PLOTCONTROLFILE'
0007      READ (2,690) FILE
0008      N = ICHR(FILE)
0009      OPEN (UNIT=4,NAME=FILE,TYPE='OLD',READONLY)
0010      READ (4,690) FNAME
          C-----
          C      THIS FILE IS THE SAME AS USED FOR 3-D FINITE ELEMENT SOLUTION
          C-----
0011      READ (4,690) FILE
0012      N = ICHR(FILE)
0013      OPEN (UNIT=5,NAME=FILE,TYPE='OLD',READONLY)
0014      READ (4,690) SIZES
0015      READ (4,*) NDEX,NELEMP,NDEL,SZNDN,SZELE,KELEM
          C-----
          C      FACTOR = CONVERSION FACTOR IN ADDITION TO "XYCONV" FOR X- AND Y-
          C      COORDINATES.
          C      FACTZ = CONVERSION FACTOR IN ADDITION TO "ZCONV" FOR VERTICAL
          C      COORDINATES.
          C      FACTOR AND FACTZ SHOULD BE ASSIGNED TO FIT THE PLOTTINGS ON
          C      AVAILABLE DEVICES.
          C      NELEMP = FLAG FOR PLOTTING THE ELEMENTS
          C      = 0 NO ELEMENTS ARE PLOTTED,PROVIDED FOR CHECKING ONLY NODE
          C      LOCATIONS OR TO DEVELOP THE ELEMENTS.
          C      = 1 PLOTS THE SURFACE ELEMENTS.
          C      LOGS = FLAG FOR PLOTTING THE VERTICAL LOGS.
          C      = 0 NO LOGS ARE PLOTTED( MAY BE USED FOR CHECKING ELEMENTS
          C      ONLY).
          C      = 1 PLOTS VERTICAL LOGS.
          C-----
0016      READ (4,690) FACT
0017      READ (4,*) FACTOR,XXMIN,YYMIN
0018      READ (4,690) DLOGS
0019      READ (4,100) LOGS,MSL,ISO,ISDU,FACTZ,ZMINUS,P30,YUP,LAYERP

```

```

0020      100  FORMAT (4I5,4E10,3,I5)
0021          IF (YUP.EQ.0) YUP = 1.5
0022          READ (4,690) FPENS
0023          READ (4,*) NPEN1,NPEN2,NPEN3
0024          READ (4,690) SURD
0025          READ (4,750) LSURD,NSEGM,NTOTAL,NOLOGS
0026          IF (LSURD.EQ.0) GO TO 130
0027          IF (NTOTAL.EQ.0) GO TO 110
0028          READ (4,*) (LIST(I),I=1,NTOTAL)
0029      110  IF (NSEGM.EQ.0) GO TO 130
0030          DO 120 I=1,NSEGM
0031          READ (4,*) NSTART,NSTOP,INC
0032          DO 120 J=NSTART,NSTOP,INC
0033          NTOTAL = NTOTAL+1
0034      120  LIST(NTOTAL) = J
0035      130  IF (NOLOGS.EQ.0) GO TO 160
0036          READ (4,690) DUMMY
0037          READ (4,*) NSEGM,NNLOGS
0038          IF (NNLOGS.NE.0) READ (4,*) (LOGST(I),I=1,NNLOGS)
0039          IF (NSEGM.EQ.0) GO TO 160
0040          DO 150 I=1,NSEGM
0041          READ (4,*) NSTART,NSTOP,INC
0042          DO 140 J=NSTART,NSTOP,INC
0043          NNLOGS = NNLOGS+1
0044      140  LOGST(NNLOGS) = J
0045      150  CONTINUE
0046      160  SZLOG = SZMOD*0.5
0047          FACTY = FACTOR
0048          IF (ISQ.EQ.0) GO TO 170
0049          IF (P30.EQ.0.0) P30 = 3.14/6.
0050          COSP30 = COS(P30)
0051          SINP30 = SIN(P30)
0052          IF (ISQ.EQ.1) FACTY = FACTOR*YUP
0053      170  FIVEA = 5./0.
0054          CALL NEWPEN (NPEN2)
0055          NINE32 = 9./32.
0056          XMIN = 10000.
0057          YMIN = 10000.
0058          READ (5,685) TITLE
0059          READ (5,700) NZELEV,KTYPE,PERCON,XYCONV,ZCONV,HCONV,NSEA
0060          IF (NPRINT.NE.0) PRINT 700, NZELEV,KTYPE,PERCON,XYCONV,ZCONV,HCONV
0061          +,NSEA
0062          IF (XYCONV.EQ.0.) XYCONV = 1.
0063      180  READ (5,710) NSEC
0064          IF (NPRINT.NE.0) PRINT 750, NSEC
0065          IF (NSEC.NE.0) GO TO 190
0066          READ (5,730) DUMMY
0067          GO TO 140
0068      190  READ (5,720) NSEC,K,X,Y,WTT,BED,NRB,LMAX,(LL(J),ZZ(J),J=1,LMAX)
0069          IF (NPRINT.NE.0) PRINT 720, NSEC,K,X,Y,WTT,BED,NRB,LMAX,(LL(J),ZZ(
0070          +J),J=1,LMAX)
0071          IF (NZELEV.EQ.1) GO TO 198
0072          ZSUM = 0.0
0073          DO 192 J = 2,LMAX
0074          ZSUM = ZSUM - ZZ(J-1)
0075      192  ZZ(J) = ZSUM

```



```

0074      ZZ(1) = 0.0
0075      190  IF (NSEC,NE,0) GO TO 350
0076      IF (KTYPE,EQ,2) READ (5,730) (CONCEN(J),J=1,LMAX)
0077      IF (LSURD,EQ,0) GO TO 230
0078      IF (LSURD,GT,0) GO TO 210
0079      DO 200 I=1,NTOTAL
0080      IF (K,EQ,LIST(I)) GO TO 190
0081      200  CONTINUE
0082      GO TO 230
0083      210  DO 220 I=1,NTOTAL
0084      IF (K,EQ,LIST(I)) GO TO 230
0085      220  CONTINUE
0086      GO TO 190
0087      230  IF (NPRINT,NE,0) PRINT *, 'NODE',K
0088      CALL NEWPEN (NPEN2)
0089      IF (LOGS,EQ,0) GO TO 270
0090      LYES = 1
0091      IF (NNLOGS,EQ,0) GO TO 250
0092      DO 240 I=1,NNLOGS
0093      IF (K,NE,LOGST(I)) GO TO 240
0094      LYES = 0
0095      GO TO 270
0096      240  CONTINUE
0097      250  IF (HSL,EQ,0) GO TO 250
0098      DO 252 J = 2, LMAX
0099      252  ZZ(J) = ZZ(J) + ZZ(J-1)
0100      ZZ(1) = 0.0
0101      ZZ(1) = 0.0
0102      DO 254 J = 2, LMAX
0103      254  ZZ(J) = ZZ(J)*FACTZ+ZZ(J-1)
0104      GO TO 270
0105      258  DO 260 J=1,LMAX
0106      260  ZZ(J) = (ZZ(J)-ZMINUS)*FACTZ+ZCONV
0107      270  IF (NPRINT,NE,0) PRINT 280, K,X,Y,LMAX,(LL(J),ZZ(J),J=1,LMAX)
0108      280  FORMAT (' K',I4,' X',G10.3,' Y',G10.3,' LMAX',I3,' LL+ZZ',/10(I5,F7
+ ,1))
0109      X = (X-XXMIN)*XYCONV*FACTOR
0110      Y = (Y-YYMIN)*XYCONV*FACTY
0111      IF (NPRINT,NE,0) WRITE (2,*) 'K,X,Y',K,X,Y
0112      IF (ISO,EQ,0,OR,ISO,EQ,2) GO TO 290
0113      X = COSP30*(X-Y)
0114      Y = SINP30*(X+Y)
0115      290  NTT = NTT+1
0116      XX(K) = X
0117      YY(K) = Y
0118      IF (XMAX,LT,X) XMAX = X
0119      IF (YMAX,LT,Y) YMAX = Y
0120      IF (XMIN,GT,X) XMIN = X
0121      IF (YMIN,GT,Y) YMIN = Y
0122      IF (LOGS,EQ,0,OR,LYES,EQ,0) GO TO 330
0123      CALL PLOT (XX(K),YY(K),3)
0124      IF (ZZ(1),EQ,0,0) GO TO 300
0125      YD(2) = YY(K)+ZZ(1)

```

```

C-----
C   INCLUDE OPTION FOR SLANTING LOGS
C-----

```

```

0126      YD(1) = YY(K)
0127      XD(1) = XX(K)
0128      XD(2) = XX(K)
0129      CALL LINES (1,1,1,2,XD,YD)
0130      IF (ZZ(1).GT.0.0) CALL PLOT (XX(K),YY(K),3)
0131      YD(1) = YY(K)+ZZ(LMAX)
0132      CALL PLOT (XX(K),YD(1),2)
0133      X = XX(K)
0134      Y = YY(K)
0135      XP = X+.055
0136      ZDIF = (ZZ(1)-ZZ(LMAX))
0137      IF (ZMAX.LT.ZDIF) ZMAX = ZDIF
0138      DO 320 L=1,LMAX
0139      LK = LMAX+1-L
0140      YP = Y+ZZ(LK)
0141      CALL SYMBOL (X,YP,.07,15,0.0,-1)
0142      IF (LK.NE.1) GO TO 310
0143      IF (NDEX.EQ.0) GO TO 320
0144      X = X+.1
0145      AL = K
0146      IF (ZZ(LMAX).GT.0.0) YP = Y
0147      CALL NUMBER (X,YP,8ZNOD,AL,0.0,-1)
0148      GO TO 320
0149      310 IF (LAYERP.EQ.0) GO TO 320
0150      LJ = LL(LK)/100
0151      ALJ = LJ
0152      YP = Y+.5*(ZZ(LK)-ZZ(LK-1))
0153      CALL NUMBER (XP,YP,8ZLOG,ALJ,0.0,-1)
0154      320 CONTINUE
0155      GO TO 190
0156      330 X = XX(K)
0157      Y = YY(K)
0158      IF (NDEX.EQ.0.AND.NHEAD.EQ.0) GO TO 190
0159      CALL SYMBOL (X,Y,.07,1,0.0,-1)
0160      IF (NDEX.EQ.0) GO TO 340
0161      AL = K
0162      X = X+.05
0163      CALL NUMBER (X,Y,8ZNOD,AL,0.0,-1)
0164      340 IF (NHEAD.EQ.0) GO TO 190
0165      IF (INOUT(K).EQ.0) GO TO 190
0166      AL = HEAD(K)
0167      X2 = X+8ZNOD*3
0168      IF (NDEX.EQ.0) X2 = XX(K)
0169      Y = YY(K)
0170      CALL NEWPEN (NPEN3)
0171      CALL NUMBER (X2,Y,8ZNOD,AL,0.0,1)
0172      GO TO 190
0173      350 IF (NELEMP.EQ.0) GO TO 680
0174      360 READ (5,710) NSEC
0175      IF (NSEC.EQ.0) GO TO 360
0176      IF (NSEC.EQ.3.OR.NSEC.EQ.4) GO TO 360
0177      IF (NSEC.EQ.6) GO TO 680
0178      DO 660 IE=1,10000
0179      CALL NEWPEN (NPEN1)
0180      X = 0.0
0181      Y = 0.0

```

```

0182      DO 370 J=1,16
0183 370      NOD(J) = 0
0184      READ (5,740) NSEC,NE,N1,NN,(NOD(J),J=1,NN)
0185      IF (NPRINT.NE.0) PRINT 740, NSEC,NE,N1,NN,(NOD(J),J=1,NN)
0186      IF (NSEC.NE.0) GO TO 670
0187      IF (N1.EQ.0) READ (5,750) (MIXNOD(J),J=1,4)
0188      IF (KELEM.EQ.1) READ (5,380) XK,YK,ZK,SC
0189 380      FORMAT (8F10.0)
0190      IF (LSUBN.EQ.0) GO TO 430
0191      DO 420 I=1,NN
0192      JJ = NOD(I)
0193      IF (LSUBN.GT.0) GO TO 420
0194      DO 390 J=1,NTOTAL
0195      IF (JJ.EQ.LIST(J)) GO TO 660
0196 390      CONTINUE
0197      GO TO 420
0198 420      DO 410 J=1,NTOTAL
0199      IF (JJ.EQ.LIST(J)) GO TO 420
0200 410      CONTINUE
0201      GO TO 660
0202 420      CONTINUE
0203 430      J = 1
0204      IN(J) = NOD(1)
0205      IF (NOD(9).EQ.0) GO TO 440
0206      J = J+1
0207      IN(J) = NOD(9)
0208 440      IF (NOD(5).EQ.0) GO TO 450
0209      J = J+1
0210      IN(J) = NOD(5)
0211 450      IF (NOD(10).EQ.0) GO TO 460
0212      J = J+1
0213      IN(J) = NOD(10)
0214 460      J = J+1
0215      IN(J) = NOD(2)
0216      IF (NOD(15).EQ.0) GO TO 470
0217      J = J+1
0218      IN(J) = NOD(15)
0219 470      IF (NOD(8).EQ.0) GO TO 480
0220      J = J+1
0221      IN(J) = NOD(8)
0222 480      IF (NOD(16).EQ.0) GO TO 490
0223      J = J+1
0224      IN(J) = NOD(16)
0225 490      J = J+1
0226      IN(J) = NOD(3)
0227      IF (NOD(12).EQ.0) GO TO 500
0228      J = J+1
0229      IN(J) = NOD(12)
0230 500      IF (NOD(4).EQ.0) GO TO 510
0231      J = J+1
0232      IN(J) = NOD(6)
0233 510      IF (NOD(11).EQ.0) GO TO 520
0234      J = J+1
0235      IN(J) = NOD(11)
0236 520      J = J+1
0237      IN(J) = NOD(4)

```

```

0238      IF (NOD(14).EQ.0) GO TO 530
0239      J = J+1
0240      IN(J) = NOD(14)
0241      530  IF (NOD(7).EQ.0) GO TO 540
0242      J = J+1
0243      IN(J) = NOD(7)
0244      540  IF (NOD(13).EQ.0) GO TO 550
0245      J = J+1
0246      IN(J) = NOD(13)
0247      550  LAST = J
0248      J = IN(LAST)
0249      CALL PLOT (XX(J),YY(J),3)
0250      DO 560 I=1,LAST
0251      J = IN(I)
0252      CALL PLOT (XX(J),YY(J),2)
0253      560  CONTINUE
0254      IF (NDEL.EQ.0) GO TO 660
0255      X = 0.0
0256      Y = 0.0
0257      DO 650 I=1,NN
0258      K = NOD(I)
0259      IF (K.EQ.0) GO TO 650
0260      IF (I.GT.4) GO TO 620
0261      BETA = 0.0
0262      MMM = N1
0263      IF (N1.NE.0) GO TO 570
0264      MCODE = MIXNOD(I)
0265      MCC(1) = MCODE/100
0266      MCC(2) = MCODE/10+MCC(1)*10
0267      570  DO 610 LT=1,2
0268      IF (N1.EQ.0) MMM = MCC(LT)
0269      GO TO (580,590,600), MMM
0270      580  BETA = BETA+0.5
0271      GO TO 610
0272      590  BETA = BETA-0.5
0273      GO TO 610
0274      600  BETA = BETA-FIVE0
0275      610  CONTINUE
0276      FACT = BETA*0.25
0277      GO TO 640
0278      620  IF (I.GT.8) GO TO 630
0279      FACT = 0.5
0280      GO TO 640
0281      630  FACT = NINE32
0282      640  IF (K.LE.0.OR.K.GT.600) PRINT *, 'K',K,IE,(NOD(J),J=1,NN)
0283      X = X+XX(K)*FACT
0284      Y = Y+YY(K)*FACT
0285      650  CONTINUE
0286      CALL NEWPEN (NPEN3)
0287      AJ = NE
0288      X = X-.15
0289      IF (NE.GT.9) X = X-.05
0290      IF (NE.GT.100) X = X-.075
0291      CALL NIMRER (X,Y,57ELE,AJ,0.0,-1)
0292      660  CONTINUE
0293      670  NELEM = IE-1

```

```

0294      IF (NPRINT.NE.0) PRINT *, NELEM,'TOTAL ELEMENTS'
0295      680  CALL NEWPEN (NPEN2)
0296      SF = FACTOR
0297      CALL OUTRFX
0298      CALL PLOTND
0299      STOP
0300      685  FORMAT (80A1)
0301      690  FORMAT (20A4)
0302      700  FORMAT (2I5,4F10.0,I5)
0303      710  FORMAT (I1)
0304      720  FORMAT (I1,I4,4F10.2,3I5,/5(6(F8.1,I5)/))
0305      730  FORMAT (16F5.0)
0306      740  FORMAT (I1,I4,15I5/16I5)
0307      750  FORMAT (16I5)
0308      END

```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODE1	012036 2575	RW,I,CON,LCL
\$PDATA	000156 55	RW,D,CON,LCL
\$IDATA	000550 100	RW,D,CON,LCL
\$VARS	032142 6705	RW,D,CON,LCL
\$TEMPS	000016 7	RW,D,CON,LCL
BOX	000160 56	RW,D,OVR,GBL

TOTAL SPACE ALLOCATED = 045324 9578

```

0001      SUBROUTINE OUTRAY
C-----
C      ROUTINE TO DRAW A BOX AROUND THE PLOT 1 INCH WIDER ON EACH SIDE,
C      1 INCH ABOVE THE TOP AND 2 INCHES BELOW THE BOTTOM, WITH LABELS
C      AND SCALE.
C-----
0002      BYTE ITITLE
0003      REAL LTRSZ
0004      COMMON /BOX/XMIN,YMIN,XMAX,YMAX,SF,LOGS,ISO,ZMAX,FACTZ,ITITLE(80)
C-----
C      REQUEST PLOT TITLE.
C-----
0005      LTRSZ = 0.14
0006      WRITE (2,*) 'SELECT TITLE OPTION'
0007      WRITE (2,*) '0 = NO TITLE, 1 = USE PREV.TITLE, 2 = NEW TITLE,'
0008      READ (2,100) IANSR
0009      100  FORMAT (I1)
0010      IF (IANSR.EQ. 0) GO TO 125
0011      IF (IANSR.NE.2) GO TO 120
0012      105  WRITE (2,*) 'ENTER PLOT TITLE'
0013      READ (2,110) ITITLE
0014      110  FORMAT (80A1)
0015      120  CONTINUE
0016      HITE = LTRSZ * 6.0 / 7.0
0017      XSPACE = (XMAX + 1.0) - (XMIN - 1.0)
0018      TSPACE = XSPACE * 0.2
0019      TEMP = TSPACE / HITE
0020      MAXCHR = TEMP
0021      NNN = ICHR2(ITITLE) - 1
0022      XNN = NNN
0023      IF (NNN.LE. MAXCHR) GO TO 125
0024      XADD = (NNN - MAXCHR) * HITE
0025      XOFD2 = XADD / 2.0
0026      WRITE (2,121) MAXCHR
0027      121  FORMAT (' TITLE TOO LONG - LIMIT IS ',I3,' CHARS.')
0028      WRITE (2,99) LTRSZ
0029      99    FORMAT (' LETTER SIZE IS ',F6,3)
0030      WRITE (2,*) ' SELECT OPTION - 1 = INCREASE BOX SIZE,'
0031      WRITE (2,*) '                2 = ENTER NEW TITLE '
0032      WRITE (2,*) '                3 = DECREASE LETTER SIZE'
0033      READ (2,*) IANSR2
0034      GO TO (122,105,123) , IANSR2
0035      122  CONTINUE
C-----
C      INCREASE BOX FOR LONG TITLE.
C-----
0036      XMIN = XMIN - XOFD2
0037      XMAX = XMAX + XOFD2
0038      GO TO 125
0039      123  CONTINUE
C-----
C      DECREASE TITLE LETTER SIZE
C-----
0040      WRITE (2,*) ' ENTER NEW LETTER SIZE FOR TITLE,'

```

```

0041      READ (2,*) LTRISZ
0042      GO TO 120
C-----
C      DRAW OUTER BOX AROUND PLOT.
C-----
0043      125  CONTINUE
0044          XMIN = XMIN-1.
0045          YMIN = YMIN-2.
0046          IF (LOGS.EQ.1) YMIN = YMIN-ZMAX
0047          IF (ISO.EQ.0) YMIN = YMIN+1
0048          CALL PLOT (XMIN,YMIN,3)
0049          YMAX = YMAX+1.
0050          CALL PLOT (XMIN,YMAX,2)
0051          XMAX = XMAX+1.
0052          CALL PLOT (XMAX,YMAX,2)
0053          CALL PLOT (XMAX,YMIN,2)
0054          CALL PLOT (XMIN,YMIN,2)
0055          CALL PLOT (0.,0.,3)

0056          XMX2 = XMAX
0057          YMY2 = YMIN
C-----
C      DRAW LABELS.
C-----
0058          IF (IANSR.EQ.0) GO TO 130
0059          NNN = ICHR2(ITITLE)
0060          IF (NNN.LE.1) GO TO 130
0061          CALL SYMBOL (XMIN+0.2,YMIN+0.4,LTRISZ,ITITLE,0.0,NNN)
0062      130  CONTINUE
C-----
C      DRAW SCALE 2 INCHES LONG.
C-----
0063          CALL SYMBOL (XMX2-1.5,YMY2+1.7,0.1,'SCALE',0.0,5)
0064          FPZ = 0.0
0065          FPS = 1.0/SF
0066          FPS2 = 2*FPS
0067          YIP55 = 1.55+YMIN
0068          YIP5 = 1.5+YMIN
0069          YIP4 = 1.4+YMIN
0070          NDIG = -1
0071          IF (FPS.LT.10.)NDIG = 2
0072          IF (FPS.LT.1.)NDIG = 3
0073          IF (FPS.LT.0.1)NDIG = 4
0074          CALL NUMBER (XMX2-2.4,YIP55,0.1,FPZ,0.0,1)
0075          CALL PLOT (XMX2-2.4,YIP5,3)
0076          CALL PLOT (XMX2-2.4,YIP4,2)
0077          CALL PLOT (XMX2-1.4,YIP4,2)
0078          CALL PLOT (XMX2-1.4,YIP5,2)
0079          CALL NUMBER (XMX2-1.45,YIP55,0.1,FPS,0.0,NDIG)
0080          CALL PLOT (XMX2-1.4,YIP4,3)
0081          CALL PLOT (XMX2-0.4,YIP4,2)
0082          CALL PLOT (XMX2-0.4,YIP5,2)
0083          CALL NUMBER (XMX2-0.5,YIP55,0.1,FPS2,0.0,NDIG)
0084          IF (FACTZ.LE.0) RETURN
C-----
C      DRAW VERTICAL SCALE 2 INCHES LONG.

```

```

C-----
0085      CALL SYMBOL (XMXP2=1.75,YMIN+1.04,0.1,'VERT, SCALE',0.0,11)
0086      FPZ = 0.0
0087      FPS = 1.0/FACTZ
0088      FPS2 = 2*FPS
0089      YP89 = 0.89+YMIN
0090      YP84 = 0.84+YMIN
0091      YP74 = 0.74+YMIN
0092      NDIG = -1
0093      IF(FPS.LT.10.)NDIG = 2
0094      IF(FPS.LT.1.)NDIG = 3
0095      IF(FPS.LT.,1)NDIG = 4
0096      CALL NUMBER (XMXP2=2.4,YP89,0.1,FPZ,0.0,1)
0097      CALL PLOT (XMXP2=2.4,YP84,3)
0098      CALL PLOT (XMXP2=2.4,YP74,2)
0099      CALL PLOT (XMXP2=1.4,YP74,2)
0100      CALL PLOT (XMXP2=1.4,YP84,2)
0101      CALL NUMBER (XMXP2=1.45,YP89,0.1,FPS,0.0,NDIG)
0102      CALL PLOT (XMXP2=1.4,YP74,3)
0103      CALL PLOT (XMXP2=0.4,YP74,2)
0104      CALL PLOT (XMXP2=0.4,YP84,2)
0105      CALL NUMBER (XMXP2=0.5,YP89,0.1,FPS2,0.0,NDIG)
0106      RETURN
0107      END

```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODE1	002726 747	RW,I,CON,LCL
SPDATA	000532 173	RW,D,CON,LCL
SIDATA	000502 161	RW,D,CON,LCL
SVARS	000126 43	RW,D,CON,LCL
STEMPS	000024 10	RW,D,CON,LCL
BOX	000160 56	RW,D,OVR,GRL

TOTAL SPACE ALLOCATED = 000514 1190

DP2:PLMAIN,LP:PLMAIN


```

C      HREADPL.FTN
0001      SUBROUTINE HREAD (WT,ZMAX,BASE,TITLE,NPRINT,INOUT)
0002      BYTE INOUT(768)
0003      DIMENSION HEAD(2560),NUMBER(2560),IDENT(768),WT(768),FILE(7),
+TITLE(16),DETAIL(20)
0004      WRITE (2,*) 'NAME OF BINARY FILE HAVING HEAD DATA'
0005      READ (2,140) FILE
0006      WRITE (2,*) 'MAT#( REGIONAL TOP OR BOTTOM = 0)'
0007      WRITE (2,*) 'NTOP ( 0 = TOP AND 1 = BOTTOM) OF SELECTED MAT#'
0008      READ (2,*) NMAT,NTOP
0009      N = ICHR(FILE)
0010      OPEN (UNIT=1,NAME=FILE,TYPE='OLD',FORM='UNFORMATTED',READONLY)
0011      READ (1) (DETAIL(I),I=1,20),LNPT,(IDENT(I),I=1,LNPT)
0012      READ (1) NTT,(NUMBER(I),HEAD(I),I=1,NTT)
0013      WRITE (2,140) DETAIL
0014      CLOSE (UNIT=1)
0015      IF (NPRINT.GE.1) PRINT 150, LNPT,(IDENT(I),I=1,LNPT)
0016      IF (NPRINT.GE.1) PRINT 160, (NUMBER(I),HEAD(I),I=1,NTT)
0017      BASE = 1.0E10
0018      ZMAX = -1.0E10
0019      DO 100 I=1,LNPT
0020      100   WT(I) = 0.0
0021      DO 120 I=1,LNPT
0022      N = IDENT(I)
0023      IF (N.EQ.0) GO TO 120
0024      NN = N+20
0025      IF (NPRINT.GE.1) WRITE (2,*) 'N=',N,'LNPT=',LNPT
0026      DO 110 L=N,NN
0027      ND = NUMBER(L)
0028      LLL = ND/1000
0029      NSUR = MOD(ND,1000)
0030      IF (NSUR.NE.1) GO TO 120
0031      IF (LLL.LT.NMAT) GO TO 110
0032      IF (LLL.EQ.NMAT) INOUT(I) = 1
0033      WT(I) = HEAD(L)
0034      IF (NTOP.EQ.0.AND.NMAT.NE.0) WT(I) = HEAD(L-1)
0035      IF (NPRINT.EQ.2) PRINT 130, I,NUMBER(L),HEAD(L)
0036      IF (BASE.GT.WT(I)) BASE = WT(I)
0037      IF (ZMAX.LT.WT(I)) ZMAX = WT(I)
0038      GO TO 120
0039      110   CONTINUE
0040      120   CONTINUE
0041      RETURN
0042      130   FORMAT (' SURFACE NODE',I5,' NODE SELECTED',I5,' HEAD',G10.3)
0043      140   FORMAT (20A4)
0044      150   FORMAT (' LNPT=',I5/200(20I5/))
0045      160   FORMAT (' HEAD VALUES READ FROM DISK',/200(8(I5,G10.3)/))
0046      END

```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODE1	001702 481	RW,I,CON,LCL
SPDATA	000214 70	RW,D,CON,LCL
FORTRAN IV-PLUS V02-510		
HREADPL.FTN	/TRIBLOCKS/WR	
SIDATA	000300 96	RW,D,CON,LCL
SVAR5	041202 8513	RW,D,CON,LCL
STEMPS	000004 2	RW,D,CON,LCL

TOTAL SPACE ALLOCATED = 043624 9162

,LPI=HREADPL

```

C      GRIDIT.FTN
C-----
C      THIS PROGRAM USES THE INPUT DECK OF THREE-DIMENSIONAL FINITE ELEME
C      GROUND WATER MODEL, FOR ESTIMATION OF LOCAL COORDINATES OF EACH GR
C      POINT.
C
C      FOR A GIVEN PROBLEM, THIS PROGRAM IS EXECUTED ONLY ONCE, USING THE
C      BINARY OUTPUT FILE OF THIS , GRIDH,GRIDV AND GRIDIN PROGRAMS
C      ESTIMATE THE GRID VALUES OF THE PARAMETER OF INTEREST.
C
C      USING APPROPRIATE VALUE OF "FACTOR" TO CONVERT COORDINATES TO
C      PLOTTING DEVICE USER CAN DEVELOP DIFFERENT INTENSITY OF GRID
C
C      THE PRODUCT OF X- AND Y- GRIDS SHOULD BE LESS THAN 6750.
C
C      IMPORTANT TO NOTE THAT THIS PROGRAM CAN HANDLE ONLY LINEAR
C      ELEMENTS,IF THE GIVEN PROBLEM HAS BEEN DISCRITIZED IN TO HIGHER OR
C      OR MIXED ORDER ELEMENTS, USING THE NODE LOCATIONS OF THE MODEL
C      THE GIVEN REGION OF INTEREST HAS TO BE REDEFINED BY LINEAR ELEMENT
C      ALL OTHER INFORMATION CAN BE LEFT IN THE SAME FORM.
C-----
0001      BYTE FILE(40)
0002      DIMENSION X(768),Y(768),NOD(4),XQ(4),YQ(4),HW(4),CONCEN(20),
+LL(20),ZZ(20),SCC(400),ETT(400),LIST(400),LNODES(768),
+ NXY(600), NYY(600) ,Z(768),NUMBER(768),FML(10)
0003  100  FORMAT (00A1)
0004      WRITE (2,*) 'CONTROL FILE WITHOUT "EXT" PROG ATTACHES "GRD"'
0005      READ (2,100) FILE
0006      N = ICHR(FILE)
0007      N = N+1
0008      FILE(N) = '.'
0009      FILE(N+1) = 'G'
0010      FILE(N+2) = 'H'
0011      FILE(N+3) = 'D'
0012      OPEN (UNIT=4,NAME=FILE,TYPE='OLD',READONLY)
0013      FILE(N+3) = 'O'
0014      OPEN (UNIT=3,NAME=FILE,FORM='UNFORMATTED')
0015      FILE(N+1) = 'N'
0016      FILE(N+2) = 'O'
0017      FILE(N+3) = 'O'
0018      OPEN (UNIT=1,NAME=FILE,TYPE='NEW',FORM='UNFORMATTED')
0019      READ (4,100) FINPUT
0020      READ (4,100) FILE
0021      N = ICHR(FILE)
0022      OPEN (UNIT=5,NAME=FILE,TYPE='OLD',READONLY)
0023      READ (4,*) LSUBD,LSEGM,NTOTAL
0024      IF (LSUBD.EQ.0) GO TO 130
0025      IF (NTOTAL.EQ.0) GO TO 110
0026      READ (4,*) (LIST(I),I=1,NTOTAL)
0027  110  IF (LSEGM.EQ.0) GO TO 130
0028      DO 120 I=1,LSEGM
0029      READ (4,*) NSTART,NSTOP,INC
0030      DO 120 J=NSTART,NSTOP,INC
0031      NTOTAL = NTOTAL+1
0032  120  LIST(NTOTAL) = J
C-----

```

```

C      SINCE NORMALLY MIXED ORDER OR HIGHER ORDER ELEMENTS ARE USED FOR
C      SIMULATION AND FOR PLOTTING PURPOSES THE GIVEN REGION HAS TO BE
C      REDEFINED IN LINEAR ELEMENTS, THROUGH THE FLAG "NFILE" THAT FILE
C      CAN BE READ DIRECTLY. IF "NFILE" = 0 IT IS ASSUMED THAT THE INPUT
C      FILE FOR SIMULATION CONSISTS OF LINEAR ELEMENTS AND NO NFILE IS
C      ASKED FOR.
C-----
0033 130 READ (4,100) DUMMY
0034      READ (4,*) NFILE,KELEM,FACTOR
C-----
C      FACTOR IS THE CONVERSION FACTOR OF X - AND Y- COORDINATES .
C      PROGRAM SEARCHES THE VALUES AT CORNER OF EACH INTEGER VALUES
C      THROUGH "FACTOR",THE INTENSITY OF GRID IS CHANGED.
C-----
0035      IF (NFILE.EQ.0) GO TO 140
0036      READ (4,100) FILE
0037      CLOSE (UNIT=4)
0038      N = ICHR(FILE)
0039      OPEN (UNIT=4,NAME=FILE,TYPE='OLD',READONLY)
0040      LN = 4
0041 140 IF (NFILE.EQ.0) LN = 5
0042      NTT = 2560
0043      WRITE (2,*) 'NPRINT'
0044      READ (2,*) NPRINT
C-----
C      CHANGE THIS NUMBER AS PER THE NUMBER OF TOTAL NODES
C-----
0045      READ (5,150) DUMMY
0046 150 FORMAT (A4)
0047      READ (5,160) NPPP,NZELEV,KTYPE,PERCON,XYCONV,ZCONV,HCONV
0048 160 FORMAT (I5,I3,I2,7F10.0)
0049      XYCONV = XYCONV*FACTOR
0050      XMIN = 1.0E10
0051      YMIN = 1.0E10
0052      XMAX = 0.0
0053      YMAX = 0.0
0054      DO 190 I=1,3000
0055      READ (5,170) NSEC
0056 170 FORMAT (I1)
0057 180 FORMAT (I1,I4,4F10.2,3I5,/2(4(F8.1,I5)/))
0058      IF (NSEC.NE.0) GO TO 200
0059      READ (5,150) DUMMY
0060 190 CONTINUE
0061 200 PRINT 210, NSEC
0062 210 FORMAT (' NSEC ',I2)
0063      NODES = 0
0064      DO 260 INODE=1,3000
0065      READ (5,180) NSEC,K,XX,YY,WT,RED,NBB,LMAX,(LL(J),ZZ(J),J=1,LMAX)
0066      IF (NSEC.NE.0) GO TO 270
0067      IF (KTYPE.EQ.2) READ (5,690) (CONCEN(I),I=1,LMAX)
0068      IF (LSUBD.EQ.0) GO TO 250
0069      IF (LSUBD.GT.0) GO TO 230
0070      DO 220 I=1,NTOTAL
0071      IF (K.EQ.LIST(I)) GO TO 260
0072 220 CONTINUE
0073      GO TO 250

```

```

0074      230      DO 240 I=1,NTOTAL
0075              IF (K.EQ.LIST(I)) GO TO 250
0076      240      CONTINUE
0077              GO TO 260
0078      250      XX = XY*XYCONV
0079              YY = YY*XYCONV
0080              IF (XX.LT.XMIN) XMIN = XX
0081              IF (YY.LT.YMIN) YMIN = YY
0082              IF (XX.GT.XMAX) XMAX = XX
0083              IF (YY.GT.YMAX) YMAX = YY
0084              X(K) = XX
0085              Y(K) = YY
0086              NODES = NODES+1
0087              LNODES(NODES) = K
0088              WRITE (1) K,X(K),Y(K),WTT,BED,LMAX,(LL(J),ZZ(J),J=1,LMAX)
0089      260      CONTINUE
0090      270      IF (NSEC,NE,1) GO TO 290
0091              READ (5,260) FMT
0092      280      FORMAT (10A4)
0093              READ (5,FMT) (M,I=1,NTT)
0094      290      IF (KTYPE,NE,2) GO TO 300
0095              READ (5,260) FMT
0096              READ (5,FMT) (C,I=1,NTT)
0097      300      IF (NSEC,NE,2) GO TO 320
0098              DO 310 I=1,1000
0099              READ (5,180) NSEC
0100              IF (NSEC,NE,0) GO TO 320
0101      310      CONTINUE
0102      320      IF (NSEC,NE,3) GO TO 340
0103              DO 330 I=1,1000
0104              READ (5,180) NSEC
0105              IF (NSEC,NE,0) GO TO 340
0106      330      CONTINUE
0107      340      IF (NSEC,NE,4) GO TO 360
0108              DO 350 I=1,1000
0109              READ (5,180) NSEC
0110              IF (NSEC,NE,0) GO TO 360
0111      350      CONTINUE
0112      360      IF (NSEC,NE,5) GO TO 610
0113              PRINT 670, XMAX,XMIN,YMAX,YMIN
0114              NXMIN = INT(XMIN)
0115              XMAX = XMAX-NXMIN
0116              NXMAX = INT(XMAX+0.5)+1
C-----
C      NXMAX IS SHIFTED BY 1.0
C-----
0117              YMIN = YMIN
0118              NYMIN = INT(YMIN)
0119              YMAX = YMAX-NYMIN
C-----
C      NYMAX IS SHIFTED BY 1.0
C-----
0120              NYMAX = INT(YMAX+0.5)+1
0121              WRITE (3) NXMAX,NXMIN,NYMAX,NYMIN,NODES,(LNODES(I),I=1,NODES)
0122              IF (NPRINT,NE,0) PRINT 680, NXMAX,NXMIN,NYMAX,NYMIN
0123              WRITE (2,680) NXMAX,NXMIN,NYMAX,NYMIN

```

```

0124      IF (NFILE,NE,0) CLOSE (UNIT=5)
0125      CLOSE (UNIT=1)
0126      DO 590 I=1,10000
0127      READ (LN,650) NSEC,NE,N1,NN,(NOD(I),I=1,NN)
0128      IF (NPRINT,NE,0) WRITE (2,*) NE
0129      IF (NSEC,NE,0) GO TO 600
0130      IF (KELEM,EQ,1) READ (5,280) DD
0131      IF (NN,NE,4) GO TO 630
0132      IF (II.LT,10,AND,NPRINT,NE,0) PRINT 370, NE,NOD
0133      370  FORMAT (' ELEM ',I3,' NODES',I5)
0134      IF (LSUBO,EQ,0) GO TO 420
0135      DO 410 I=1,NN
0136      JJ = NOD(I)
0137      IF (LSUBO,GT,0) GO TO 390
0138      DO 380 J=1,NTOTAL
0139      IF (JJ,EQ,LIST(J)) GO TO 590
0140      380  CONTINUE
0141      GO TO 420
0142      390  DO 400 J=1,NTOTAL
0143      IF (JJ,EQ,LIST(J)) GO TO 410
0144      400  CONTINUE
0145      GO TO 590
0146      410  CONTINUE
0147      420  ITIME = 0
0148      430  DO 440 J=1,4
0149      JJ = NOD(J)
0150      XQ(J) = X(JJ)
0151      440  YQ(J) = Y(JJ)
0152      450  D = (XQ(1)+XQ(2)+XQ(3)+XQ(4))*0.25
0153      DD = (YQ(1)+YQ(2)+YQ(3)+YQ(4))*0.25
0154      A = (XQ(1)-XQ(2)-XQ(3)+XQ(4))*0.25
0155      AA = (YQ(1)-YQ(2)-YQ(3)+YQ(4))*0.25
0156      B = (XQ(1)+XQ(2)-XQ(3)-XQ(4))*0.25
0157      BB = (YQ(1)+YQ(2)-YQ(3)-YQ(4))*0.25
0158      C = (XQ(1)-XQ(2)+XQ(3)-XQ(4))*0.25
0159      CC = (YQ(1)-YQ(2)+YQ(3)-YQ(4))*0.25
0160      A1 = A*CC-C*AA
0161      C  WRITE (2,*) 'D',D,'DD',DD,'A',A,'AA',AA,'B',B,'BB',BB,C,CC
0162      LINEAR = 0
0163      IF (A1,EQ,0,AND,C,EQ,0,AND,CC,EQ,0,0) LINEAR = 1
0164      IF (A1,NE,0,OR,LINEAR,EQ,1) GO TO 460
0165      XQ(2) = XQ(2)+.0001
0166      YQ(2) = YQ(2)+.0001
0167      GO TO 450
0168      460  R1 = A*BB-AA*B-C*DD+CC*D
0169      C1 = BB*D-B*DD
0170      A2 = A1+2,0
0171      AA1 = CC*B-BB*C
0172      BB1 = CC*D-C*DD+B*AA-A*BB
0173      CC1 = AA*D-A*DD
0174      LINET = 0
0175      IF (AA1,EQ,0,0) LINET = 1
0176      GO TO 490

```

```

C-----
C  NEXT SECTION IS NOT OPERATED AT PRESENT
C-----

```

```

0176 465 ITIME = ITIME+1
0177 IF (ITIME,GE,4) GO TO 470
0178 NSAVE = NOD(1)
0179 NOD(1) = NOD(2)
0180 NOD(2) = NOD(3)
0181 NOD(3) = NOD(4)
0182 NOD(4) = NSAVE
0183 GO TO 430
0184 470 PRINT 480, NE,C1,B,BB
0185 480 FORMAT ('ELEM',I5,' C1 =',E10.3,' B =',E10.3,' BB =',E10.3)
0186 GO TO 590
0187 490 CALL POINTS (NE,XQ,YQ,1,NXX,NYY,NPOINT,NPRINT)
0188 IF (NPOINT,EQ,0) GO TO 590
0189 DO 540 LZ=1,NPOINT
0190 ET = 0.0
0191 NX = NXX(LZ)
0192 NY = NYY(LZ)
C IF (NX,LT,NXMIN) NXMIN = NX
0193 C2 = C1+B*NY-BB*NX
0194 B2 = B1+C*NY-CC*NX
0195 BB2 = BB1+NY*C-CC*NX
0196 CC2 = CC1+A*NY-AA*NX
0197 IF (LINEAR,EQ,1) GO TO 510
0198 SIGN = 1.0
0199 ITIME = 0
0200 IF (A2,EQ,0.) WRITE (2,*) 'L',LINEAR,'A2',A2
0201 500 SC = (-B2+SIGN*SQRT(B2*B2-4.*A1*C2))/A2
0202 IF (ABS(SC).LE.1.00001) GO TO 540
0203 ITIME = ITIME+1
0204 IF (ITIME,EQ,2) GO TO 570
0205 SIGN = -1
0206 GO TO 500
0207 510 SC = -C2/B2
0208 IF (ABS(SC).LE.1.00001) GO TO 520
0209 GO TO 570
0210 520 ET = -CC2/BB2
0211 IF (ABS(ET).LE.1.00001) GO TO 530
0212 GO TO 570
0213 530 IF (ET,GT,1.0) ET = 1.0
0214 IF (ET,LT,-1.0) ET = -1.0
0215 GO TO 560
0216 540 DIV = BB+SC*CC
0217 NFLAG = 0
0218 IF (DIV,NE,0.) GO TO 550
0219 WRITE (6,*) 'DIV=0.FORNE',NE,'BB',BB,'SC',SC,'CC',CC,(NOD(I),XQ(I)
+,YQ(I),I=1,4)
0220 DIV = .000001
C-----
C THIS IS DONE TO AVOID DIVISION BY 0.0 NEED FURTHER IMPROVEMENT
C-----
0221 PRINT *, 'NX,NY,DD,AA',NX,NY,DD,AA
0222 NFLAG = 1
0223 550 ET = (NY-DD-SC*AA)/DIV
0224 IF (NFLAG,EQ,1) PRINT *, 'ET AFTER SETTING DIV = .00001',ET
0225 IF (ABS(ET).LE.1.00001) GO TO 560
0226 SIGN = -1

```

```

0227      GO TO 500
0228      560    NX = NX+1-NXMIN
0229      NY = NY+1-NYMIN
0230      WRITE (3) NX,NY,NOD,SC,ET
0231      GO TO 500
0232      570    IF (NPRINT.NE.0) PRINT 660, NE,NX,NY
0233      580    CONTINUE
0234      590    CONTINUE
0235      600    CONTINUE
0236      GO TO 700
0237      610    PRINT 620
0238      620    FORMAT (' PROGRAM STOPPED ***** THERE WAS NO INSEC =5) IN DECK')
0239      GO TO 700
0240      630    PRINT 640, NE
0241      640    FORMAT (' PROGRAM STOPPED SINCE FOR ELEM ',I5,' NODES ARE MORE
+ THAN 4'//,' FOR THIS PROGRAM LINEAR ELEMENTS ARE REQUIRED')
0242      650    FORMAT (I1,I4,I5I5/16I5)
0243      660    FORMAT (' CONVERSION NOT ACHIEVED IN ELEMENT',I5,' NX',I5,' NY',I5,
+)
0244      670    FORMAT (' XMAX=',F10.1,' XMIN=',F10.1,' YMAX=',F10.1,' YMIN=',F10.
+1)
0245      680    FORMAT (' NXMAX=',I5,' NXMIN=',I5,' NYMAX=',I5,' NYMIN=',I5)
0246      690    FORMAT (9E10.3)
0247      700    STOP
0248      END

```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCONE1	010150 2100	RW,I,CON,LCL
SPDATA	000226 75	RW,D,CON,LCL
SIDATA	001124 290	RW,D,CON,LCL
SVARS	045476 9631	RW,D,CON,LCL
STEMPS	000042 17	RW,D,CON,LCL

TOTAL SPACE ALLOCATED = 057262 12121

```

0001      SUBROUTINE POINTS (NE,XX,YY,INCR,NXX,NYY,NPOINT,NPRINT)
0002      DIMENSION XX(4), YY(4), A(4), B(4), YB(4), NXX(600), NYY(600),
+        XLMAX(4), XLMIN(4), YLMAX(4), YLMIN(4), YBPTR(4)
0003      INTEGER X, Y
0004      DO 120 I=1,4
0005        II = I+1
0006        IF (II.EQ.5) II = 1
0007        XXX = XX(II)-XX(I)
0008        IF (XXX.NE.0.0) GO TO 100
0009        A(I) = 1.0E8
0010        B(I) = 1.0E8
0011        GO TO 110
0012      100  YYY = YY(II)-YY(I)
0013           A(I) = YYY/XXX
0014           B(I) = YY(I)-A(I)*XX(I)
0015      110  XLMIN(I) = XX(I)
0016           IF (XX(II).LT.XX(I)) XLMIN(I) = XX(II)
0017           XLMAX(I) = XX(I)
0018           IF (XX(II).GT.XX(I)) XLMAX(I) = XX(II)
0019           YLMIN(I) = YY(I)
0020           IF (YY(II).LT.YY(I)) YLMIN(I) = YY(II)
0021           YLMAX(I) = YY(I)
0022           IF (YY(II).GT.YY(I)) YLMAX(I) = YY(II)
0023      120  CONTINUE
0024           XGMAX = -1.0E8
0025           XGMIN = 1.0E8
0026           YGMAX = -1.0E8
0027           YGMIN = 1.0E8
0028           DO 130 I=1,4
0029             IF (XX(I).LT.XGMIN) XGMIN = XX(I)
0030             IF (XX(I).GT.XGMAX) XGMAX = XX(I)
0031             IF (YY(I).LT.YGMIN) YGMIN = YY(I)
0032             IF (YY(I).GT.YGMAX) YGMAX = YY(I)
0033      130  CONTINUE
0034           MINX = XGMIN+.9999
0035           MAXX = XGMAX
0036           MINY = YGMIN+.9999
0037           MAXY = YGMAX
0038           NPOINT = 0
0039           DO 220 X=MINX,MAXX,INCR
0040             NN = 0
0041             DO 150 I=1,4
0042               II = I-1
0043               IF (ABS(X-XLMIN(II)).LT.1.0E-4) GO TO 140
0044               IF (ABS(X-XLMAX(II)).LT.1.0E-4) GO TO 140
0045               IF (X.LT.XLMIN(II)) GO TO 150
0046               IF (X.GT.XLMAX(II)) GO TO 150
0047      140  IF (A(II).EQ.1.0E8) GO TO 150
0048             NN = NN+1
0049             YB(NN) = A(II)*X+B(II)
0050             YB(NN) = YB(NN)*1.0E2
0051             YB(NN) = INT(YB(NN)+0.5)
0052             YB(NN) = YB(NN)/1.0E2
0053             YBPTR(NN) = I
0054             IF (NN.LE.1) GO TO 150
0055             IF (X.EQ.XLMIN(II).AND.X.EQ.XLMAX(II).AND.A(II).NE.1.0E8) NN = NN-1

```



```

0056      IF (X.EQ.XLMAX(I),AND,X.EQ.XLMIN(I),AND,A(I),NE,1.0E8) NN = NN-1
0057      IF (I,NE,4) GO TO 150
0058      IF (X.EQ.XLMIN(4),AND,X.EQ.XLMAX(1),AND,A(4),NE,1.0E8,AND,A(1),NE,
+1.0E8) NN = NN-1
0059      IF (X.EQ.XLMAX(4),AND,X.EQ.XLMIN(1),AND,A(4),NE,1.0E8,AND,A(1),NE,
+1.0E8) NN = NN-1
0060      150  CONTINUE
0061      NNM = NN-1
0062      DO 170 J=1,NNM
0063      DO 160 I=1,NNM
0064      IF (YB(I).LT,YB(I+1)) GO TO 160
0065      STORE = YB(I)
0066      YB(I) = YB(I+1)
0067      YB(I+1) = STORE
0068      STORE = YBPTR(I)
0069      YBPTR(I) = YBPTR(I+1)
0070      YBPTR(I+1) = STORE
0071      160  CONTINUE
0072      170  CONTINUE
0073      DO 210 Y=MINY,MAXY,INCR
0074      DO 180 I=1,NN
0075      IF (A(YBPTR(I)),EQ,1.0E4,AND,Y.GT,YLMIN(YBPTR(I)),AND,Y.LT,
+YLMAX(YBPTR(I))) GO TO 200
0076      IF (YB(I),EQ,Y) GO TO 200
0077      IF (YB(I).LT,Y) GO TO 180
0078      ISECTS = I-1
0079      GO TO 190
0080      180  CONTINUE
0081      ISECTS = NN
0082      190  IF (ISECTS/2*2,EQ,ISECTS) GO TO 210
0083      200  NPOINT = NPOINT+1
0084      NXX(NPOINT) = X
0085      NYY(NPOINT) = Y
0086      210  CONTINUE
0087      220  CONTINUE
0088      IF (NPRINT,GE,2) PRINT *, 'NE',NE,'# OF POINTS',NPOINTS,'MAX,MIN',
+XGMAX,XGMIN,YGMAX,YGMIN,'NXX',(NXX(IP),IP=1,NPOINT),'NYY',(NYY(IP)
+,IP=1,NPOINT)
0089      RETURN
0090      END

```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODE1	003050	788 RW,I,CON,LCL
SPDATA	000064	26 RW,D,CON,LCL
SIDATA	000050	20 RW,D,CON,LCL
SVARS	000266	91 RW,D,CON,LCL
STEMPS	000020	8 RW,D,CON,LCL

TOTAL SPACE ALLOCATED = 003512 933

,LP:=GRIDIT

```

C      GRIDIN.FTN
C-----
C      THIS PROGRAM ESTIMATES VALUES AT REGULAR GRID OF INPUT PARAMETERS
C      LIKE TOP, BOTTOM, OR THICKNESS OF ANY GIVEN HYDROGEOLOGIC UNIT.
C
C      BINARY FILE *.NOD OF GRIDIT.FTN ARE USED
C
C      BINARY OUTPUT FILE "VALUE" CONTAINS GRID VALUES TO BE PLOTTED.
C-----
0001      INTEGER SCA,ETA
0002      BYTE INOUT(1000),FILE(40)
0003      DIMENSION LL(7),ZZ(7),WT(1000),VALUE(6750),NOD(4),SCA(4),
+      ETA(4),TITLE(16),X(768),Y(768)
0004      DATA SCA/1,-1,-1,1/
0005      DATA ETA/1,1,-1,-1/
0006      WRITE (2,*) 'OUTPUT FILE OF GRID WITHOUT EXTENTION PROGRAM APENDS
+      * *.GRD" AND * *.GRD"'
0007      READ (2,100) FILE
0008      FORMAT (40A1)
0009      N = ICHR(FILE)
0010      N = N+1
0011      FILE(N) = '.'
0012      FILE(N+1) = 'N'
0013      FILE(N+2) = 'O'
0014      FILE(N+3) = 'D'
0015      OPEN (UNIT=1,NAME=FILE,TYPE='OLD',FORM='UNFORMATTED',READONLY)
0016      FILE(N+1) = 'G'
0017      FILE(N+2) = 'R'
0018      FILE(N+3) = 'O'
0019      OPEN (UNIT=3,NAME=FILE,TYPE='OLD',FORM='UNFORMATTED',READONLY)
0020      WRITE (2,*) 'FORWHOLEDOMAINTHICKNESSMAT#0'
0021      WRITE (2,*) 'NFLAG=0NORMAL=1FORUSINGNEARESTNODEVALUE'
0022      WRITE (2,*) ' '
0023      WRITE (2,*) 'NPRINT,MAT#,NTOP(0=T,1=BOTTOM,2=THICKNESS),NFLAG(0,1)'
0024      READ (2,*) NPRINT,NMAT,NTOP
0025      BASE = 1.0E10
0026      ZMAX = -1000.
0027      DO 150 I=1,1000
0028      READ (1,END=160) NO,XX,YY,W,BED,LMAX,(LL(J),ZZ(J),J=1,LMAX)
0029      IF (LNPT.LT.NO) LNPT = NO
0030      X(NO) = XX
0031      Y(NO) = YY
0032      DO 140 J=1,LMAX
0033      LLL = LL(J)/100
0034      IF (NMAT.NE.0) GO TO 110
C-----
C      MATERIAL NUMBER IS ZERO
C-----
0035      IF (NTOP.EQ.0) WT(I) = ZZ(1)
0036      IF (NTOP.EQ.1) WT(I) = ZZ(LMAX)
0037      IF (NTOP.EQ.2) WT(I) = ZZ(1)-ZZ(LMAX)
0038      INOUT(I) = 1
0039      GO TO 150
0040      110 IF (LLL.LT.NMAT) GO TO 140
0041      IF (LLL.EQ.NMAT) INOUT(I) = 1

```

```

0042      IF (NTOP.NE.2) GO TO 120
0043      WT(ND) = ZZ(J-1)-ZZ(J)
0044      IF (WT(ND).LT.0.0) WT(ND) = 0.0
0045      GO TO 130
0046 120    WT(ND) = ZZ(J-1)
0047      IF (NTOP.EQ.1) WT(ND) = ZZ(J)
0048 130    IF (BASE.GT.WT(ND)) BASE = WT(ND)
0049      IF (ZMAX.LT.WT(ND)) ZMAX = WT(ND)
0050      GO TO 150
0051 140    CONTINUE
0052 150    CONTINUE
0053 160    CLOSE (UNIT=1)
      C
0054      ZMAX = INT(ZMAX+10)
0055      IF (NPRINT.NE.0) PRINT *, BASE,ZMAX
0056      READ (3) NXMAX,NXMIN,NYMAX,NYMIN
0057      NEED = NXMAX*NYMAX
0058      IF (NEED.LT.6750) GO TO 170
0059      WRITE (2,*) 'NXMAX*NYMAX>6750.PROGRAM STOPPED.'
0060      WRITE (2,*) 'REDUCESCALE SIZE IN FILE"',GRD,'"AND RERUN GRIDIT'
0061      STOP
0062 170    DO 210 INODE=1,9000
0063      READ (3,END=220) NX,NY,NOD,SC,ET
0064      IF (NPRINT.GE.2) PRINT 180, INODE,NX,NY,NOD,SC,ET
0065 180    FORMAT (' INODE',7I5,2E12.3)
0066      SUM = 0.0
0067      IFLAG = 0
0068      DO 190 J=1,4
0069 190    IF (INOUT(NOD(J)),EQ.1) IFLAG = IFLAG+1
0070      IF (IFLAG.LE.2) GO TO 210
0071      DO 200 J=1,4
0072      K = NOD(J)
0073      SUM = SUM+.25*(1+SC*SCA(J))*(1+ET*ETA(J))*WT(K)
0074 200    CONTINUE
0075      VALUE((NY-1)*NXMAX+NX) = SUM
0076 210    CONTINUE
0077 220    CLOSE (UNIT=3)
0078      IF (NFLAG.EQ.0) GO TO 240
0079      DO 230 I=1,LNPT
0080      IF (X(I).EQ.0.0.AND.Y(I).EQ.0.0) GO TO 230
0081      NX = INT(X(I)+0.5)-NXMIN+1
0082      NY = INT(Y(I)+0.5)-NYMIN
0083      IJ = NY*NXMAX+NX
0084      VALUE(IJ) = WT(I)
0085 230    CONTINUE
0086 240    ZMAX = -1.0E10
0087      ZMIN = 1.E10
0088      DO 250 I=1,NYMAX
0089      NPLUS = (I-1)*NXMAX
0090      DO 250 J=1,NXMAX
0091      VAL = VALUE(NPLUS+J)
0092      IF (VAL.LT.ZMIN) ZMIN = VAL
0093 250    IF (VAL.GT.ZMAX) ZMAX = VAL
0094      IF (NPRINT.NE.0) PRINT 260
0095 260    FORMAT (' READING OF FILE 2 COMPLETED'//)
0096      IF (NPRINT.LE.1) GO TO 290
      DO 270 I=1,NXMAX

```

```

0097      NSTART = (I-1)*NXMAX+1
0098      NSTOP = NSTART+NXMAX-1
0099      270  PRINT 280, I, (VALUE(J), J=NSTART, NSTOP)
0100      280  FORMAT (/15,10G10.3,/200(5X,10G10.3/))
0101      290  WRITE (2,*) 'NXMAX,NYMAX,ZMIN,ZMAX', NXMAX, NYMAX, ZMIN, ZMAX
0102      IF (NPRINT.NE.0) PRINT *, NXMIN, NXMAX, NYMIN, NYMAX, ZMIN, ZMAX
0103      OPEN (UNIT=1, NAME='VALUE', TYPE='NEW', FORM='UNFORMATTED')
0104      DO 300 J=1, NYMAX
0105      NPLUS = (J-1)*NXMAX
0106      300  WRITE (1) (VALUE(NPLUS+I), I=1, NXMAX)
0107      CLOSE (UNIT=1)
0108      STOP
0109      END
  
```

```

C      GRIDH,FTN
C-----
C      THIS PROGRAM ESTIMATES VALUES AT REGULAR GRID OF HEAD AT TOP OR
C      BOTTOM VARIOUS HYDROGEOLOGIC UNITS.
C
C      BINARY FILE OF PROGRAM GRIDIT IS USED FOR IDENTIFYING
C      THE ELEMENTS ASSOCIATED WITH THE GIVEN GRID AND ALSO THE LOCAL
C      COORDINATES OF GRID POINT AS RELATED TO THE GIVEN ELEMENT.
C
C      USER SPECIFIES THE INPUT BINARY FILE CONTAINING HEAD VALUES.
C
C      BINARY OUTPUT FILE "VALUE" CONTAINS GRID VALUES FOR PLOTS,
C      1. NXMAX(WIDTH IN X DIRECTION),NYMAX(WIDTH IN Y-DIRECTION),
C      NUMBER OF CHARACTERS IN TITLE,TITLE
C      2. NUMBER OF NYMAX RECORDS EACH CONTAINING NXMAX VALUES
C-----
0001      BYTE INOUT,FILE
0002      COMMON/A/NFLAG,NPRINT,NXMAX,NXMIN,NYMAX,NYMIN,NPT,NTOTAL,WT(768),
+ZMAX,BASE,TITLE(16),FILE(40),LIST(768),NEED,INOUT(768)
0003      WRITE (2,*) 'GRIDIT FILE NAME (WITHOUT".EXT")'
0004      READ (2,100) FILE
0005      100  FORMAT (80A1)
0006      N = ICHR(FILE)
0007      N = N+1
0008      FILE(N) = '.'
0009      FILE(N+1) = 'G'
0010      FILE(N+2) = 'R'
0011      FILE(N+3) = 'O'
0012      OPEN (UNIT=3,NAME=FILE,TYPE='OLO',FORM='UNFORMATTED',READONLY)
0013      FILE(N+1) = 'N'
0014      FILE(N+2) = 'O'
0015      FILE(N+3) = 'O'
0016      OPEN (UNIT=7,NAME=FILE,TYPE='ULD',FORM='UNFORMATTED',READONLY)
0017      READ (3) NXMAX,NXMIN,NYMAX,NYMIN,NTOTAL,(LIST(I),I=1,NTOTAL)
0018      NEED = NXMAX*NYMAX
0019      NFLQ = 1
0020      IF (NEED.GT.6750) NFLQ = 2
0021      CALL HREAD
0022      GO TO (110,120), NFLQ
0023      110  CALL GRID
0024      GO TO 130
0025      120  WRITE (2,*) 'FILE Q DISK'
0026      READ (2,*) IUN
0027      CALL ASNLUN (1,'DP',IUN,ISTAT)
0028      CALL DPFILE (1,'PLOT',512.,IPTR1)
0029      CALL GRDFLG
0030      130  STOP
0031      END

```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
------	------	------------

FORTRAN IV-PLUS V02-510
GRIDH,FTN /TRIBLOCKS/WR

12152132 16-OCT-79

PAGE 2

SCODE1	000572	189	RW,I,CON,LCL
SPDATA	000106	35	RW,D,CON,LCL
SIDATA	000122	41	RW,D,CON,LCL
SVARS	000014	6	RW,D,CON,LCL
A	012602	2753	RW,D,OVR,BBL

TOTAL SPACE ALLOCATED = 013648 3024

NO FPP INSTRUCTIONS GENERATED

,LPI=GRIDH

HREAD.FTN

/TRI8LOCKS/WR

```

      C      HREAD.FTN
0001      SUBROUTINE HREAD
0002      BYTE INOUT,FILE
0003      COMMON/A,NFLAG,NPRINT,NXMAX,NXMIN,NYMAX,NYMIN,NPT,NTOTAL,WT(768),
      *HMAX,HMIN,TITLE(16),FILE(40),LIST(768),NEED,INOUT(768)
0004      DIMENSION HEAD(2560),NUMBER(2560),IDENT(768),DETAIL(20),FILE2(20)
      C      -----
      C      VERTICAL LOG DATA ASSUMING UNIFORM LOG IN THE REGION
      C      -----
0005      WRITE (2,*) 'NAME OF BINARY FILE HAVING HEAD DATA'
0006      READ (2,160) FILE2
0007      WRITE (2,*) 'NPRINT,MAT#,NTOP(0=T,1=B),NFLAG(00R1)'
0008      READ (2,*) NPRINT,NMAT,NTOP,NFLAG
0009      N = ICHR(FILE2)
0010      OPEN (UNIT=1,NAME=FILE2,TYPE='OLD',FORM='UNFORMATTED',READONLY)
0011      READ (1) (DETAIL(I),I=1,20),LNPT,(IDENT(I),I=1,LNPT)
0012      READ (1) NTT,(NUMBER(I),HEAD(I),I=1,NTT)
0013      WRITE (2,170) DETAIL
0014      CLOSE (UNIT=1)
0015      IF (NPRINT.GE.1) PRINT 180, LNPT,(IDENT(I),I=1,LNPT)
0016      IF (NPRINT.GE.1) PRINT 190, (NUMBER(I),HEAD(I),I=1,NTT)
0017      HMIN = 1.0E10
0018      HMAX = -1.0E10
0019      DO 100 I=1,768
0020      100      WT(I) = 0.0
0021      DO 140 I=1,LNPT
0022      N = IDENT(I)
0023      IF (N.EQ.0) GO TO 140
0024      NN = N+25
0025      DO 130 L=N,NN
0026      ND = NUMBER(L)
0027      LLL = ND/1000
0028      NSUR = MOD(ND,1000)
0029      IF (NSUR.EQ.I) GO TO 110
0030      IF (NMAT.NE.0.OR,NTOP.NE.1) GO TO 140
0031      WT(I) = HEAD(L-1)
0032      INOUT(I) = 1
0033      IF (NPRINT.EQ.2) PRINT 150, I,NUMBER(L-1),WT(I)
0034      GO TO 120
0035      110      IF (LLL.LT.NMAT) GO TO 130
0036      IF (LLL.EQ.NMAT) INOUT(I) = 1
0037      WT(I) = HEAD(L)
0038      IF (NTOP.EQ.0.AND,NMAT.NE.0) WT(I) = HEAD(L-1)
0039      IF (NPRINT.EQ.2) PRINT 150, I,NUMBER(L),WT(I)
0040      120      IF (HMIN.GT.WT(I)) HMIN = WT(I)
0041      IF (HMAX.LT.WT(I)) HMAX = WT(I)
0042      GO TO 140
0043      130      CONTINUE
0044      140      CONTINUE
0045      RETURN
0046      150      FORMAT (' SURFACE NODE',I5,' NODE SELECTED',I5,' HEAD',G10.3)
0047      160      FORMAT (20A4)
0048      170      FORMAT (20A4)
0049      180      FORMAT (' LNPT=',I5/200(20I5/))
0050      190      FORMAT (' HEAD VALUES READ FROM DISK',/200(8(I5,G10.3)/))
0051      END

```

FORTRAN IV-PLUS V02-S1D
HREAD,FTN /TRIBLOCKS/HR

12156131 16-OCT-79

PAGE 2

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODE1	001736 495	RW,I,CON,LCL
SPDATA	000116 39	RW,D,CON,LCL
SIDATA	000246 83	RW,D,CON,LCL
SVARS	041266 8539	RW,D,CON,LCL
STEMPS	000004 2	RW,D,CON,LCL
A	012602 2753	RW,D,OVR,GRL

TOTAL SPACE ALLOCATED = 056416 11911

,LF1#HREAD


```

C      GRID.FTN
0001      SUBROUTINE GRID
0002      INTEGER SCA,ETA
0003      BYTE INOUT,FILE
0004      COMMON/A,NFLAG,NPRINT,NXMAX,NXMIN,NYMAX,NYMIN,NPT,NTOTAL,WT(768),
+ZMAX,BASE,TITLE(16),FILE(40),LIST(768),NEED,INOUT(768)
0005      DIMENSION VALUE(6750),NOD(4),SCA(4),ETA(4)
0006      DATA SCA/1,-1,-1,1/
0007      DATA ETA/1,1,-1,-1/
0008      DO 100 I=1,NEED
0009      100  VALUE(I) = 0.
0010      DO 140 INODE=1,9000
0011      READ (3,END=150) NX,NY,NOD,SC,ET
0012      IF (NPRINT.GE.3) PRINT 110, INODE,NX,NY,NOD,SC,ET
0013      110  FORMAT (' INODE',7I5,2E12.3)
0014      SUM = 0.0
0015      IFLAG = 0
0016      DO 120 J=1,4
0017      120  IF (INOUT(NOD(J)).EQ.1) IFLAG = IFLAG+1
0018      IF (IFLAG.LE.2) GO TO 140
0019      DO 130 J=1,4
0020      K = NOD(J)
0021      SUM = SUM+.25*(1+SC*SCA(J))*(1+ET*ETA(J))*WT(K)
0022      130  CONTINUE
0023      VALUE((NY-1)*NXMAX+NX) = SUM
0024      140  CONTINUE
0025      150  CLOSE (UNIT=3)
0026      IF (NFLAG.EQ.0) GO TO 170
0027      DO 160 I=1,NTOTAL
0028      READ (7,END=170) K,X,Y
0029      NX = INT(X+0.5)-NXMIN+1
0030      NY = INT(Y+0.5)-NYMIN
0031      IJ = NY*NXMAX+NX
0032      IF (NPRINT.GE.1) WRITE (6,*) 'NODE',K,'AV.HEAD',VALUE(IJ),'NODE H'
+WT(K)
0033      160  VALUE(IJ) = WT(K)
0034      170  ZMAX = -1.0E10
0035      ZMIN = 1.0E10
0036      DO 180 I=1,NYMAX
0037      NPLUS = (I-1)*NXMAX
0038      DO 180 J=1,NXMAX
0039      VAL = VALUE(NPLUS+J)
0040      IF (VAL.LT.ZMIN) ZMIN = VAL
0041      180  IF (VAL.GT.ZMAX) ZMAX = VAL
0042      IF (NPRINT.NE.0) PRINT 190
0043      WRITE (2,*) 'READING OFF FILE POINTS COMPLETED'
0044      OPEN (UNIT=5,NAME='VALUE',TYPE='NEW',FORM='UNFORMATTED')
0045      190  FORMAT (' READING OF FILE "POINTS" COMPLETED')
0046      IF (NPRINT.LE.1) GO TO 220
0047      DO 200 I=1,NYMAX
0048      NSTART = (I-1)*NXMAX+1
0049      NSTOP = NSTART+NXMAX-1
0050      200  PRINT 210, I,(VALUE(J),J=NSTART,NSTOP)
0051      210  FORMAT (/I5,10G10.3/,20(5X,10G10.3/))
0052      220  IF (NPRINT.NE.0) PRINT *, ' POINTS IN X-DIRECTION',NXMAX,' Y-DIREC
+TION',NYMAX,' MINIMUM VALUE OF THE PARAMETER',ZMIN,' MAXIMUM VALUE

```

```

      +,ZMAX
0053      WRITE (2,*) 'NXMAX,NYMAX,ZMIN,ZMAX',NXMAX,NYMAX,ZMIN,ZMAX
0054      DO 230 J=1,NYMAX
0055      NPLUS = (J-1)*NXMAX
0056      230 WRITE (5) (VALUE(NPLUS+I),I=1,NXMAX)
0057      CLOSE (UNIT=5)
0058      RETURN
0059      END
  
```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODE1	002214 582	RW,I,CON,LCL
SPDATA	000256 87	RW,D,CON,LCL
SIDATA	000162 57	RW,D,CON,LCL
SVARS	064702 13537	RW,D,CON,LCL
A	012602 2753	RW,D,OVR,GBL

TOTAL SPACE ALLOCATED = 102360 17016

,LP1=GRID

```

      C      GRDFLG,FTN
0001      SUBROUTINE GRDFLG
0002      BYTE INOUT,FILE
0003      COMMON/A/NFLAG,NPRINT,NXMAX,NXMIN,NYMAX,NYMIN,NPT,NTOTAL,WT(768),
      +ZMAX,BASE,TITLE(16),FILE(40),LIST(768),NEED,INOUT(768)
0004      INTEGER SCA,ETA
0005      DIMENSION NOD(4),SCA(4),ETA(4),VALUE(2560)
0006      DATA SCA/1,-1,-1,1/
0007      DATA ETA/1,1,-1,-1/
0008      CALL DPR (IPTR1,1,,VALUE,5120.)
0009      NRR = 1.
0010      DO 140 INODE=1,9000
0011      READ (3,END=150) NX,NY,NOD,SC,ET
0012      IF (NPRINT.GE.3) PRINT 100, INODE,NX,NY,NOD,SC,ET
0013      100  FORMAT (' INODE',7I5,2E12.3)
0014      SUM = 0.0
0015      IFLAG = 0
0016      DO 110 J=1,4
0017      110  IF (INOUT(NOD(J)).EQ.1) IFLAG = IFLAG+1
0018      IF (IFLAG.LE.2) GO TO 140
0019      DO 120 J=1,4
0020      K = NOD(J)
0021      SUM = SUM+.25*(1+SC*SCA(J))*(1+ET*ETA(J))*WT(K)
0022      120  CONTINUE
0023      NR = 1+(NY-1)/10
0024      IF (NR.EQ.NRR) GO TO 130
0025      CALL DPW (IPTR1,(NRR-1)*20.+1,VALUE,5120.)
0026      CALL DPR (IPTR1,(NR-1)*20.+1,VALUE,5120.)
0027      NRR = NR
0028      130  KK = (NY-(NR-1)*10.)*256.+NX
0029      VALUE(KK) = SUM
0030      140  CONTINUE
0031      150  CLOSE (UNIT=3)
0032      IF (NFLAG.EQ.0) GO TO 180
0033      DO 170 I=1,NTOTAL
0034      READ (7,END=180) K,X,Y
0035      PRINT *, 'K,X,Y',K,X,Y
0036      NX = INT(X+0.5)-NXMIN+1
0037      NY = INT(Y+0.5)-NYMIN
0038      NR = 1+NY/10
0039      PRINT *, 'NX,NY,NR',NX,NY,NR
0040      IF (NR.EQ.NRR) GO TO 160
0041      CALL DPW (IPTR1,1+(NRR-1)*20.,VALUE,5120.)
0042      CALL DPR (IPTR1,1+(NR-1)*20.,VALUE,5120.)
0043      NRR = NR
0044      160  KK = (NY-(NR-1)*10)*256.+NX
0045      IF (NPRINT.GE.1) WRITE (2,*) 'NODE',K,'AV.HEAD',VALUE(KK),'NODE H'
      +,WT(K)
0046      170  VALUE(KK) = WT(K)
0047      180  CALL DPW (IPTR1,(NRR-1)*20+1.,VALUE,5120.)
0048      ZMAX = -1.0E10
0049      ZMIN = 1.0E10
0050      DO 190 J=1,NYMAX
0051      CALL DPR (IPTR1,(J-1)*2+1.,VALUE,512.)
0052      DO 190 I=1,NXMAX
0053      VAL = VALUE(I)

```

```

0054      IF (VAL.LT.ZMIN) ZMIN = VAL
0055      190  IF (VAL.GT.ZMAX) ZMAX = VAL
0056      IF (NPRINT.NE.0) PRINT 200
0057      WRITE (2,*) 'READING OF FILE POINTS COMPLETED'
0058      200  FORMAT (' READING OF FILE "POINTS" COMPLETED'/)
0059      IF (NPRINT.LE.1) GO TO 210
0060      210  IF (NPRINT.NE.0) PRINT *, ' POINTS IN X-DIRECTION',NXMAX,' Y-DIREC
+TION',NYMAX,' MINIMUM VALUE OF THE PARAMETER',ZMIN,' MAXIMUM VALUE
+ ',ZMAX
0061      WRITE (2,*) 'NXMAX,NYMAX,ZMIN,ZMAX',NXMAX,NYMAX,ZMIN,ZMAX
0062      RETURN
0063      END

```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODE1	002510 676	RW,I,CON,LCL
SPDATA	000310 100	RW,D,CON,LCL
SIDATA	000144 50	RW,D,CON,LCL
SVAR3	020112 5157	RW,D,CON,LCL
STEMPS	000004 2	RW,D,CON,LCL
A	012602 2753	RW,D,OVR,GBL

TOTAL SPACE ALLOCATED = 042104 8738

*CP1=GRDFLO

```

C      PLOTBC.FTN
C-----
C      THIS PROGRAMS PLOTS THE BOUNDARY CONDITIONS SPECIFIED FOR 3D MODEL
C-----
0001      REAL NINE32,NINE8
0002      INCLUDE '311,3]HDR.FTN'
0003      *      COMMON/HDR/LNPT,NTT,NBPTC,NBPTC,NSTREM,NPT,NSELEM,NELEM,NODMAX,
      *      #NONZER,NQD,MATN,KTYPE,NMAX,NSTIF,NTIME,NSTORE,NDUM,TITLE(20),
      *      #DP1,DP2,DP3,DP4,DP5,DP6,DP7,DP89,DP10,DP11,DP12,DP13,DP14,DP15,
      *      #DP16,DP17,DP18,NTSURF,NE8URF
0004      INCLUDE '311,3]P3I.FTN'
0005      *      COMMON/P3I/NSTEDY,ITOTAL,NQWCP,NQ,MNTRY,ITT,NHBOND,NFDUM,TT(70),
      *      #DETAIL(20)
0006      DIMENSION XX(1280),YY(1280),ZZ(1280),YD(2),A(2),NOD8(768),
      +BIV(768),CONCEN(7),FILE(7),XD(2),TDATA(20),IDENT(768),
      +NUM(2560),TIME(4)
0007      DATA TIME/'TIME','STE','P',' ',' '
0008      WRITE (2,*) 'PLOT TITLE MAXIMUM OF 80 CHARACTERS'
0009      READ (2,180) TDATA
0010      WRITE (2,*) 'ENTER FILE-Q DISK UNIT'
0011      READ (2,*) IUN
0012      IF (IUN.NE.3) CALL ASNLUN (1,'DP',IUN,ISTAT)
0013      CALL DPFILE (1,'DESCRIP',25,,IPTR1)
0014      CALL DPR (IPTR1,0,,LNPT,92,)
0015      CALL DPR (IPTR1,1,,NUM,2560,)
0016      CALL DPR (IPTR1,11,,IDENT,768,)
0017      CALL DPFILE (1,'DTANDQS',DP5,IPTR5)
0018      CALL DPFILE (1,'BOUNDCOND',DP7,IPTR7)
0019      CALL DPR (IPTR5,0,,NSTEDY,188,)
0020      IF (NHBOND.EQ.0) GO TO 100
0021      WRITE (2,*) 'VARIABLE HELD8C= PLOTALL(=0) OR SPECIFIC(=TIMESTEP)'
0022      READ (2,*) ISTEPS
0023      IF (ISTEPS.EQ.0) ISTEPS = ITOTAL
0024      100      CALL DPR (IPTR7,0,,NOD8,768,)
0025      CALL DPFILE (1,'XYZCOR',DP3,IPTR3)
0026      CALL DPR (IPTR3,0,,ZZ,2560,)
0027      CALL DPR (IPTR3,20,,XX,2560,)
0028      CALL DPR (IPTR3,40,,YY,2560,)
0029      WRITE (2,*) 'PLOTING SCALE = (1)HORIZONTAL (2)VERTICAL'
0030      READ (2,*) FACTOR,FACTZ
0031      WRITE (2,*) 'ENTER PLOTING DEVICE 0=TO 1=XY 2=DISK'
0032      READ (2,*) IDEV
0033      IF (IDEV.EQ.0) CALL ASNLUN (3,'TG',0,ISTAT)
0034      IF (IDEV.EQ.1) CALL ASNLUN (3,'XY',0,ISTAT)
0035      IF (IDEV.NE.2) GO TO 110
0036      CALL ASNLUN (3,'DP',0,ISTAT)
0037      WRITE (2,*) 'ENTER PLOTFILE NAME'
0038      READ (2,180) FILE
0039      N = ICHR(FILE)
0040      OPEN (UNIT=3,NAME=FILE,TYPE='NEW',FORM='UNFORMATTED')
0041      110      FIVE8 = 5./8.
0042      NINE32 = 9./32.
0043      CALL PLOTS (0,,0,3)
0044      CALL NEWPEN (NPEN2)
0045      WRITE (2,*) 'NBPTC',NBPTC
0046      IBEG = 1

```

```

0047      IF (ISTEPS.NE.0) IBEG = ISTEPS
0048      DO 170 IT=IBEG,ISTEPS
0049      IF (NHCOND.EQ.1) CALL DPR (IPTR5,(IT-1)*6,+1821.,BIV,NBPTC+2.)
0050      IF (NHCOND.EQ.0) CALL DPR (IPTR7,3.,BIV,NBPTC+2.)
0051      LAST = 999
0052      NRR = 0
0053      MAX = 1280
0054      XMIN = 1.E18
0055      YMIN = 1.E18
0056      DO 140 I=1,NBPTC
0057      J = MOD8(I)
0058      IF (J.EQ.0) GO TO 150
0059      NR = (J-1)/MAX
0060      IF (NR.EQ.NRR) GO TO 120
0061      ADD = NR*10.
0062      CALL DPR (IPTR3,0.,+ADD,ZZ,2560.)
0063      CALL DPR (IPTR3,20.,+ADD,XX,2560.)
0064      CALL DPR (IPTR3,40.,+ADD,YY,2560.)
0065      NRR = NR
0066      120  N = NUM(J)
0067      JJ = J-NR*MAX
0068      ADD = ZZ(JJ)*FACTZ
0069      X = XX(JJ)*FACTOR
0070      Y = YY(JJ)*FACTOR+ADD
0071      IF (X.LT.XMIN) XMIN = X
0072      IF (Y.LT.YMIN) YMIN = Y
0073      NP = MOD(N,1000)
0074      IF (NP.NE.LAST) GO TO 130
0075      CALL PLOT (XSAVE,YSAVE,3)
0076      CALL PLOT (X,Y,2)
0077      130  XSAVE = X
0078      YSAVE = Y
0079      CALL SYMBOL (X,Y,.07,1,0.0,-1)
0080      AJ = N
0081      X = X+.08
0082      CALL NUMBER (X,Y,.07,AJ,0.0,-1)
0083      X = X+.3
0084      AJ = BIV(I)
0085      CALL NUMBER (X,Y,.07,AJ,0.0,1)
0086      LAST = NP
0087      140  CONTINUE
0088      150  N = ICHR2(TDATA)
0089      YMIN = YMIN-1.0
0090      CALL SYMBOL (XMIN,YMIN,.07,TDATA,0.0,N)
0091      IF (NHCOND.EQ.0) GO TO 160
0092      CALL SYMBOL (XMIN,YMIN=.2,.07,TIME,0.0,9)
0093      AL = IT
0094      CALL NUMBER (1.0,YMIN-1.,.07,AL,0.0,-1)
0095      160  CALL PLOTND
0096      170  CONTINUE
0097      STOP
0098      180  FORMAT (20A4)
0099      END

```

PROGRAM SECTIONS

FORTRAN JV-PLUS V02-510
PLOTBC.FTN /TRIBLOCKS/WR

12151131 16-OCT-79

PAGE 3

NAME	SIZE	ATTRIBUTES
SCODE1	002370 636	RW,I,CON,LCL
SPDATA	000552 181	RW,D,CON,LCL
SIDATA	000562 185	RW,D,CON,LCL
SVARS	064420 13448	RW,D,CON,LCL
STEPS	000004 2	RW,D,CON,LCL
HDR	000274 94	RW,D,OVR,GBL
P3I	000570 188	RW,D,OVR,GBL

TOTAL SPACE ALLOCATED = 071434 14734

,LP1=PLOTBC

```

C      PLFLUX.FTN
C-----
C      PROGRAM TO PLOT SURFACE AND PRESCRIBED FLUX RATES IN VARIOUS LAYER
C-----
0001      REAL NINE32,NINE8
0002      COMMON/HDR/LNPT,NTT,NBPTC,NBFTC,NSTREM,NPT,NSELEH,NELEM,NODMAX,
0003      +NONZER,NG1,MATN,KTYPE,NMAX,NSTIF,NTIME,NSTORE,NDUM,TITLE(20)
0004      DIMENSION XX(600),YY(600),NOD(16),IN(16),MIXNOD(4),
0005      +XMID(500),YMID(500),NAME(500),MCC(2),LL(7),ZZ(7),YD(2),A(2),
0006      +CONCEN(7),FILE(7),XD(2),Q(400),DETAIL(20)
0007      WRITE (2,*) 'TYPETITLEOF PLOT MAXIMUM OF 80 CHARACTERS'
0008      READ (2,450) DETAIL
0009      WRITE (2,*) 'SURFACE FLUX(=0) OR ELEMENT FLUX(=1)'
0010      READ (2,*) NFX
0011      IF (NFX.EQ.0) GO TO 120
C      PLOTTING THE SUMQ VALUES
0012      OPEN (UNIT=4,NAME='STRESS.CRD',TYPE='OLD',READONLY)
0013      WRITE (2,*) 'N YEAR,NMAT(1=GLACIAL,2=MAGOTHY)'
0014      READ (2,*) N YEAR,NMAT
0015      DO 100 K=1,N YEAR
0016      READ (4,110) Q
0017      IF (N YEAR,EQ.K,AND,NMAT,EQ.1) GO TO 130
0018      100 READ (4,110) Q
0019      GO TO 130
0020      110 FORMAT (12F10.0)
0021      120 WRITE (2,*) 'ENTER THE FILE-Q DISK UNIT'
0022      READ (2,*) IUN
0023      IF (IUN.NE.3) CALL ASNLUN (1,'OP',IUN,ISTAT)
0024      CALL DPFIL (1,'DISCRIP',25,,IPTR1)
0025      CALL DPR (IPTR1,0,,LNPT,50,)
0026      CALL DPFIL (1,'DTAND008',1301,,IPTR5)
0027      CALL DPR (IPTR5,1001,,Q,800,)
0028      130 WRITE (2,*) 'ENTER INPUT FILE NAME'
C-----
C      THIS FILE IS THE SAME AS USED FOR 3-D FINITE ELEMENT SOLUTION
C-----
0029      READ (2,450) FILE
0030      N = ICHR(FILE)
0031      OPEN (UNIT=5,NAME=FILE,TYPE='OLD',READONLY)
0032      WRITE (2,*) 'NPRINT(0,1),NELEMP(0=NO,1=YES)'
0033      READ (2,*) NPRINT,NELEMP
C-----
C      FACTOR = CONVERSION FACTOR IN ADDITION TO "XYCONV" FOR X- AND Y-
C      COORDINATES.
C-----
0034      WRITE (2,*) 'FACTOR'
0035      READ (2,*) FACTOR
0036      WRITE (2,*) 'NPEN1,NPEN2,NPEN3'
0037      READ (2,*) NPEN1,NPEN2,NPEN3
0038      PRINT *, ' HORIZONTAL FACTOR=',FACTOR
0039      WRITE (2,*) 'ENTER PLOTTING DEVICE 0=TG 1=XY 2=DISK'
0040      READ (2,*) IDEV
0041      IF (IDEV.EQ.0) CALL ASNLUN (3,'TG',0,ISTAT)
0042      IF (IDEV.EQ.1) CALL ASNLUN (3,'XY',0,ISTAT)
0043      IF (IDEV.NE.2) GO TO 140
0044      CALL ASNLUN (3,'OP',0,ISTAT)

```



```

0042      WRITE (2,*) 'ENTER PLOT FILE NAME'
0043      READ (2,450) FILE
0044      N = ICHR(FILE)
0045      OPEN (UNIT=3,NAME=FILE,TYPE='NEW',FORM='UNFORMATTED')
0046      140 FIVE8 = 5./8.
0047          NINE32 = 9./32.
0048      CALL PLOTS (0.,0.,3)
0049      CALL NEWPEN (NPEN2)
0050      YD(1) = FACTOR*49000.0
0051      CALL PLOT (YD(2),YD(1),2)
0052      YD(2) = FACTOR*135000.
0053      CALL PLOT (YD(2),YD(1),2)
0054      YD(1) = 0.0
0055      CALL PLOT (YD(2),YD(1),2)
0056      YD(2) = 0.0
0057      CALL PLOT (YD(2),YD(1),2)
0058      READ (5,450) TITLE
0059      READ (5,460) NZELEV,KTYPE,PERCON,XYCONV,ZCONV,HCONV,NSEA
0060      150 READ (5,470) NSEC
0061          IF (NSEC.NE.0) GO TO 160
0062      READ (5,490) DUMMY
0063      GO TO 150
0064      160 READ (5,480) NSEC,K,X,Y,NTT,BED,NBB,LMAX,(LL(J),ZZ(J),J=1,LMAX)
0065          IF (NSEC.NE.0) GO TO 180
0066      IF (NPRINT.NE.0) PRINT 170, K,X,Y,LMAX,(LL(J),ZZ(J),J=1,LMAX)
0067      170 FORMAT (' K',I4,' X',F8.2,' Y',F8.2,' LMAX',I3,' LL+ZZ',/7(I5,F8.1)
0068          +)
0069      IF (KTYPE.EQ.2) READ (5,490) (CONCEN(J),J=1,LMAX)
0070      XX(K) = X*XYCONV*FACTOR
0071      YY(K) = Y*XYCONV*FACTOR
0072      NTT = NTT+1
0073      GO TO 160
0074      180 READ (5,470) NSEC
0075          IF (NSEC.EQ.0) GO TO 180
0076          IF (NSEC.EQ.3.OR.NSEC.EQ.4) GO TO 180
0077          DO 430 IE=1,10000
0078              CALL NEWPEN (NPEN1)
0079              X = 0.0
0080              Y = 0.0
0081              DO 190 J=1,16
0082                  NOD(J) = 0
0083              READ (5,500) NSEC,NE,N1,NN,(NOD(J),J=1,NN)
0084              IF (NSEC.NE.0) GO TO 440
0085              IF (N1.EQ.0) READ (5,510) (MIXNOD(J),J=1,4)
0086              J = 1
0087              IN(J) = NOD(1)
0088              IF (NOD(9).EQ.0) GO TO 200
0089              J = J+1
0090              IN(J) = NOD(9)
0091              200 IF (NOD(5).EQ.0) GO TO 210
0092              J = J+1
0093              IN(J) = NOD(5)
0094              210 IF (NOD(10).EQ.0) GO TO 220
0095              J = J+1
0096              IN(J) = NOD(10)
0097              220 J = J+1

```

```

0097      IN(J) = NOD(2)
0098      IF (NOD(15).EQ.0) GO TO 230
0099      J = J+1
0100      IN(J) = NOD(15)
0101  230    IF (NOD(8).EQ.0) GO TO 240
0102      J = J+1
0103      IN(J) = NOD(8)
0104  240    IF (NOD(16).EQ.0) GO TO 250
0105      J = J+1
0106      IN(J) = NOD(16)
0107  250    J = J+1
0108      IN(J) = NOD(3)
0109      IF (NOD(12).EQ.0) GO TO 260
0110      J = J+1
0111      IN(J) = NOD(12)
0112  260    IF (NOD(6).EQ.0) GO TO 270
0113      J = J+1
0114      IN(J) = NOD(6)
0115  270    IF (NOD(11).EQ.0) GO TO 280
0116      J = J+1
0117      IN(J) = NOD(11)
0118  280    J = J+1
0119      IN(J) = NOD(4)
0120      IF (NOD(14).EQ.0) GO TO 290
0121      J = J+1
0122      IN(J) = NOD(14)
0123  290    IF (NOD(7).EQ.0) GO TO 300
0124      J = J+1
0125      IN(J) = NOD(7)
0126  300    IF (NOD(13).EQ.0) GO TO 310
0127      J = J+1
0128      IN(J) = NOD(13)
0129  310    LAST = J
0130      CALL NEWPEN (NPEN3)
0131      J = IN(LAST)
0132      IF (NELEMP.EQ.0) GO TO 330
0133      CALL PLOT (XX(J),YY(J),3)
0134      DO 320 I=1,LAST
0135      J = IN(I)
0136      CALL PLOT (XX(J),YY(J),2)
0137  320    CONTINUE
0138  330    DO 420 I=1,NW
0139      K = NOD(I)
0140      IF (K.EQ.0) GO TO 420
0141      IF (I.GT.4) GO TO 390
0142      BETA = 0.0
0143      MMM = N1
0144      IF (N1.NE.0) GO TO 340
0145      MCODE = MIXNOD(I)
0146      MCC(1) = MCODE/100
0147      MCC(2) = MCODE/10-MCC(1)*10
0148  340    DO 380 LT=1,2
0149      IF (N1.EQ.0) MMM = MCC(LT)
0150      GO TO (350,360,370), MMM
0151  350      BETA = BETA+0.5
0152      GO TO 380

```

```

0153 360 BETA = BETA-0.5
0154      GO TO 380
0155 370 BETA = BETA-FIVE8
0156 380 CONTINUE
0157      FACT = BETA*0.25
0158      GO TO 410
0159 390 IF (I.GT.0) GO TO 400
0160      FACT = 0.5
0161      GO TO 410
0162 400 FACT = NINE32
0163 410 IF (K.LE.0.OR.K.GT.600) PRINT *, 'K',K,IE,(NOD(J),J=1,NN)
0164      X = X+XX(K)*FACT
0165      Y = Y+YY(K)*FACT
0166 420 CONTINUE
0167      CALL NEWPEN (NPEN2)
0168      AJ = NE
      C      X = X - .15
      C      IF(NE.GT.9)X = X -.05
      C      IF(NE.GT.180)X = X -.075
0169      CALL NUMBER (X,Y,.05,AJ,0.0,-1)
0170      IF (Q(NE).EQ.0.0) GO TO 430
0171      QJ = Q(NE)*36500
0172      IF (NFX,NE.0) QJ = Q(NE)
0173      Y = Y-.1
0174      CALL NEWPEN (NPEN2)
0175      CALL NUMBER (X,Y,.07,QJ,0.0,1)
0176 430 CONTINUE
0177 440 NELEM = IE-1
0178      PRINT *, NELEM,'TOTAL ELEMENTS'
0179      N = ICHR2(DETAIL)
0180      CALL SYMBDL (.2,.2,.07,DETAIL,0.0,N)
0181      CALL PLOTND
0182      STOP
0183 450 FORMAT (20A4)
0184 460 FORMAT (2I5,4F10.0,I5)
0185 470 FORMAT (I1)
0186 480 FORMAT (I1,I4,4F10.2,3I5,/2(6(F8.1,I5)/))
0187 490 FORMAT (16F5.0)
0188 500 FORMAT (I1,I4,15I5/16I5)
0189 510 FORMAT (16I5)
0190      ENN

```

PROGRAM SECTIONS

NAME	SIZE	ATTRIBUTES
SCODE1	005456 1431	RW,I,CON,LCL
SPDATA	000664 218	RW,D,CON,LCL
SIDATA	000576 191	RW,D,CON,LCL
SVARS	027036 5903	RW,D,CON,LCL
STEMPS	000002 1	RW,D,CON,LCL
HDR	000164 58	RW,D,OVR,GRL

TOTAL SPACE ALLOCATED = 036364 7602

10.0 SIMPLE ILLUSTRATIVE EXAMPLES OF INPUT DATA AND EXECUTION OF FE3DGW

10.1 GENERAL

To acquaint users with the options and input data for FE3DGW and its interactive execution, four examples with various combinations of boundary conditions are expounded upon using one simplified element. Extension of the problem, accomplished by addition of further elements, is illustrated in the fifth example using four radial elements with three layers. The following discussions include usage of the FE3DGW programs for estimation of hydraulic head. For the supportive programs, user's manuals are described in Chapter 8.0.

The PDP 11/45 computers provides the option of installing a task, designated by three letters. The subprograms of FE3DGW, which are used repeatedly, are installed as PR1, PR2, BND, P3I and PR3 corresponding to the subprograms, PROG1, PROG2, BAND, PROG3I, and PROG3 respectively (Table 10.1). Subprogram PR1 and P3I require an input file in addition to interactive input. The PR1 input file contains titles, conversion factors, material properties, nodal coordinates, boundary conditions and element details (Figure 8.1). P3I reads (Figure 8.2) the options for steady state or transient simulation, time steps for transient solution, surface and element source/sink terms (time invariant or variant) and also the option to define time variant held boundary conditions. For all the illustrative examples, the input files for PBI (designated with extension ".PR1") and PR1 (designated with extension ".P3I") are included along with copy of the interactive input.

10.2 LINEAR ELEMENT WITH HELD BOUNDARY CONDITIONS AT TOP AND BOTTOM

This example illustrates the use of a two dimensional surface with vertical logs to generate three-dimensional elements (Figure 10.1). Each material layer constitutes an individual three-dimensional element. In this example, the surface element (100 ft x 100 ft) is defined by four nodes, 1(0,0), 2(100,0), 3(100,100), and 4(0,100) and the vertical log at each node

FIGURE 10.1. Linear Element 4 Layers with Held B.C. At. Surface and Base

TABLE 10.1. Copy of Terminal Output for Installation of Tasks

```
MCR>INS DP1:[311,3]PROG1/TASK=...PR1
MCR>INS DP1:[311,3]PROG2/TASK=...PR2
MCR>INS DP1:[311,3]BAND/TASK=...END
MCR>INS DP1:[311,3]PROG3I/TASK=...P3I
MCR>INS DP1:[311,3]PROG3/TASK=...PR3
```

is described by four material layers (5 interfaces). The node numbers assigned to the base of each log are 4001, 4002, 4003, and 4004 (see Figure 9.3 for details). The total hydraulic head at the surface (nodes 1 to 4) correspond to the surface elevation, 400 ft and similarly, the hydraulic head at the base of the last element (nodes 4001 to 4004) is held at 0 ft.

Table 10.2 is the input data file (LINEAR.PR1) for PR1. For the steady state solution, the input data file, HSTEADY.P3I (Table 10.3) is used in P3I. Table 10.4 illustrates the interactive input stream required to run FE3DGW. Table 10.5 and 10.6 illustrates the output of PR1 and PR3.

The transient simulation of this problem is obtained by using TRANSIENT.P3I (Table 10.7) in P3I and then executing PR3 (Table 10.8). Because TRANSIENT.P3I specifies variable time steps, PR3 executes the subroutines ELEMAT (element matrix), SYSMAT (system matrix), MREAD (creating file for equation solver), UPPERT (decomposition into Upper Triangle), BVALUE (right-hand B vector estimation for $AX = B$ system), BACKSUB (back substitution) and HWRITE (update simulation results and printing) for each time step.

Alternatively, transient solutions can be obtained by using constant time step in PR3. The formation and decomposition of the system matrix is done only once, because the resultant system coefficient matrix remains the same in a constant time step. The solution is obtained by estimation of the right hand vector (BVALUE) and back substitution (BACKSUB). Only the results, which match with the elapsed time steps specified in P3I are saved on the Disk. The interactive execution of the transient case with a constant time step is given in Table 10.9.

TABLE 10.2. One Linear Element (LINEAR.PR1)

FILE NAME= LINEAR.PR1

LINEAR ELEMENT 4 LAYERS WITH HELD 0,C. AT, SURFACE AND BASE

```

1 0 1.0 1.0 1.0 1.0
0 1 IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY
1.0 1.0 1.0 .25 .000001 0.0 .000001 .000001
0 2 IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY
1.0 1.0 1.0 .25 .000001 0.0 .000001 .000001
0 3 IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY
1.0 1.0 1.0 .25 .000001 0.0 .000001 .000001
0 4 IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY
1.0 1.0 1.0 .25 .000001 0.0 .000001 .000001
0 5 IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY
1.0 1.0 1.0 .25 .000001 0.0 .000001 .000001
1 SWITCH FOR NODAL DATA AND END OF PROPERTIES
0 1 0.0 0.0 0.0 0.0 1 5 01
400.0 0102 300.0 0203 200.0 0304 100.0 0400 0.0
0 2 100.0 0.0 0.0 0.0 1 5 01
400.0 0102 300.0 0203 200.0 0304 100.0 0400 0.0
3 100.0 100.0 0.0 0.0 1 5 01
400.0 0102 300.0 0203 200.0 0304 100.0 0400 0.0
0 4 0.0 100.0 0.0 0.0 1 5 01
400.0 0102 300.0 0203 200.0 0304 100.0 0400 0.0
2
1 400.0
2 400.0
3 400.0
4 400.0
04001 0.0
04002 0.0
04003 0.0
04004 0.0
5
0 1 1 4 1 2 3 4
6 SWITCH FOR END OF DATA

```

TABLE 10.3. PROG3I Input File (HSTEADY.P3I) for Steady State Simulation

FILE NAME= HSTEADY.P3I

PROG3I FILE**STEADY STATE FLOW ---- SURFACE AND ELEMENT FLUX = 0.0
0 0 1 1 0 0 0;NSTEADY,ITOTAL,NQWCP,NQ,MNTRY,NHBOND,NTEXP

TABLE 10.4. Interactive Execution of Steady State Using LINEAR.PR1
and HSTEADY.P3I

```

MCR>PR1
  ENTER FILE-Q DISK UNIT
  2
  ENTER INPUT FILE NAME
  LINEAR.PR1
  NPRINT(0,5),NPINCH(0=MISSING MATERIAL BY WEDGED ELEM,1=OTHERWISE)
  3,0
  NEW PROBLEM(=0) OR INPUT MODIFICATION(=1)
  0
  ...PR1  --  STOP

MCR>PR2
  ENTER FILE-Q DISK UNIT
  2
  ENTER PRINT OPTION (0,1,2)
  0
  ...PR2  --  STOP

MCR>BND
  ENTER FILE-Q DISK UNIT
  2
  ENTER PRINT OPTION(0,1,2)
  0
  ...BND  --  STOP

MCR>P3I
  ENTER FILE-Q DISK UNIT
  2
  ENTER INPUT FILE NAME
  HSTEADY.P3I
  ENTER PRINT OPTION(0,1,2)
  0
  NEW TITLE(1=Y,0=N)
  0
  NSTEDY      0 ITOTAL      0 NQWCP      1 NQ      1 MENTRY      0 NHBOND
    0 NTEXP      0
  ...P3I  --  STOP

MCR>PR3
  ENTER FILE-Q DISK UNIT
  2
  PRINT OPTION (NORMAL= 1, FOR DEBUG = 2 , NO OUTPUT=0)
  NBEGIN(NORMAL= 1 AND FOR RESTART = SUBROUTINE IN MAIN)
  IT=TIME STEP TO START (NORMAL=1,RESTART=TIME STEP)
  NDIG = OUTPUT DIGITS AFTER DECIMAL IN "F" FORMAT
  NCONS = 0 FOR TIME STEPS READ IN PROG3I
          = 1 FOR CONSTANT TIME STEP FOR COMPUTATION

  NPRINT(0,1,2),NBEGIN(1,2,3,4,5,6,7),IT,NDIG,NCONS
  1,1,1,2,0
  IS EXPECTED BAND WIDTH TO BE MORE THAN 252(1=Y,0=N)
  0
  NBEGIN      1 NPRINT      1 IT      1
  ELEMENT MATRIX COMPLETED
  SYSTEM MATRIX FORMED
  MREAD SUBROUTINE COMPLETED
  UPPER TRIANGLE DECOMPOSITION DONE
  B ESTIMATION DONE
  BACK SUB DONE
  ...PR3  --  STOP

MCR>

```

TABLE 10.5. PROG1 Output of LINEAR.PR1

TIME= 13:53:35 DATE= 13-SEP-79

 LINEAR ELEMENT 4 LAYERS WITH NODAL FLUX AT SURFACE AND HELD B.C. AT BASE

 VALUES OF CONVERSION FACTORS SPECIFIED

CONVERSION FACTOR FOR PERMEABILITY DATA	1.0000
CONVERSION FACTOR FOR XY-COORDINATES	1.0000
CONVERSION FACTOR FOR Z-COORDINATES	1.0000
CONVERSION FACTOR FOR HEAD(BOTH INITIAL AND B.C.)	1.00
KTYPE =	0
FLAG FOR CONTROLLING THE LEVEL OF PRINT OUTPUT=	2

TABLE 10.5. (contd)

PROPERTIES FOR MATERIAL 1 *****

IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY

K=X=	1.00000	L/T
K=Y=	1.00000	L/T
K=Z=	1.00000	L/T
THETA=	0.250000	
MEDIUM COMPRESSIBILITY	0.100000E+05	1/L
FLUID COMPRESSIBILITY	0.100000E+05	1/L
REFERENCE PRESSURE HEAD OF THETA	0.00	L
SPECIFIC STORAGE COEFF=	0.100000E+06	1/L

PROPERTIES FOR MATERIAL 2 *****

IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY

K=X=	1.30000	L/T
K=Y=	1.00000	L/T
K=Z=	1.00000	L/T
THETA=	0.250000	
MEDIUM COMPRESSIBILITY	0.100000E+05	1/L
FLUID COMPRESSIBILITY	0.100000E+05	1/L
REFERENCE PRESSURE HEAD OF THETA	0.00	L
SPECIFIC STORAGE COEFF=	0.100000E+06	1/L

PROPERTIES FOR MATERIAL 3 *****

IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY

K=X=	1.00000	L/T
K=Y=	1.00000	L/T
K=Z=	1.00000	L/T
THETA=	0.250000	
MEDIUM COMPRESSIBILITY	0.100000E+05	1/L
FLUID COMPRESSIBILITY	0.100000E+05	1/L
REFERENCE PRESSURE HEAD OF THETA	0.00	L
SPECIFIC STORAGE COEFF=	0.100000E+06	1/L

PROPERTIES FOR MATERIAL 4 *****

IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY

K=X=	1.00000	L/T
K=Y=	1.00000	L/T
K=Z=	1.00000	L/T
THETA=	0.250000	
MEDIUM COMPRESSIBILITY	0.100000E+05	1/L
FLUID COMPRESSIBILITY	0.100000E+05	1/L
REFERENCE PRESSURE HEAD OF THETA	0.00	L
SPECIFIC STORAGE COEFF=	0.100000E+06	1/L

PROPERTIES FOR MATERIAL 5 *****

IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY

K=X=	1.00000	L/T
K=Y=	1.30000	L/T
K=Z=	1.00000	L/T
THETA=	0.250000	
MEDIUM COMPRESSIBILITY	0.100000E+05	1/L
FLUID COMPRESSIBILITY	0.100000E+05	1/L
REFERENCE PRESSURE HEAD OF THETA	0.00	L
SPECIFIC STORAGE COEFF=	0.100000E+06	1/L

TABLE 10.5. (contd)

COORDINATES AND OTHER DETAILS OF EACH NODE									
INTERNAL NODE#	USER NODE#	X-COORD.	Y-COORD.	Z-COORD.	INITIAL HEAD	TEMP C	DEG	RHO	XMEW CONCENTRATION FACTOR PPM
1	1	0.0	0.0	400.0	0.0				
6	2	100.0	0.0	400.0	0.0				
11	3	100.0	100.0	400.0	0.0				
16	4	0.0	100.0	400.0	0.0				
TOTAL NODES IN SYSTEM=									20
NUMBER OF SURFACE NODES									4
MAXIMUM NUMBER ASSIGNED TO THE SURFACE NODE									4
TOTAL NUMBER OF NODES WITH PRESCRIBED HEAD B.C.ARE=									
4									
NODE#	4001	HEAD=	0.00	NODE#	4002	HEAD=	0.00	NODE#	4003
NODE#				NODE#	4004	HEAD=	0.00	NODE#	4004
NODE#				NODE#				NODE#	
TOTAL NUMBER OF NODES WITH FLUX B.C.=									
4									
NODE#	1	FLUX=	0.250E+04	NODE#	2	FLUX=	0.250E+04	NODE#	3
NODE#				NODE#	4	FLUX=	0.250E+04	NODE#	
NODE#				NODE#				NODE#	
ELEM#, 1 ORDER ONE MAT# 1 TOTAL NODES 8 CORNER NODES									
1				2				3	4
1001	1002	1003	1004						

TABLE 10.5. (contd)

TOTAL NUMBER OF UNKNOWNNS ARE=				16 THESE NODES ARE AS GIVEN BELOW											

1	1001	2001	3001	2	1002	2002	3002	3	1003	2003	3003	4	1004	2004	
3004															

DIMENSIONS IN OTHER PROGRAMS HAVE TO BE SET FOR FOLLOWING MAXIMUM VALUES															
TOTAL NUMBER OF SURFACE NODES(NTSURF)=												4			
MAXIMUM NUMBER ASSIGNED TO SURFACE NODES(LNPT)=												4			
TOTAL NODES IN THE SYSTEM (NTT)=												20			
THE FOLLOWING PARAMETERS ARE RELATED TO B.C															
TOTAL NUMBER OF POTENTIAL B.C. PRESCRIBED (NBPTC)=												4			
TOTAL NUMBER OF FLUX B.C. PRESCRIBED (NBFTC)=												4			
TOTAL NUMBER OF STREAMS B.C. PRESCRIBED (NSTREM)=												0			
THE FOLLOWING TWO PARAMETERS ARE RELATED TO STIFFNESS MATRIX															
TOTAL NUMBER OF UNKNOWN(NPT)=												16			

THE FOLLOWING PARAMETERS ARE ASSOCIATED WITH ELEMENTS															
TOTAL NUMBER OF SURFACE ELEMENTS(NESURF) =												1			
MAXIMUM NUMBER ASSIGNED TO SURFACE ELEMENT(NSELEM)=												1			
TOTAL NUMBER OF ELEMENTS(NELEM) =												4			
MAXIMUM NUMBER OF NODES IN ANY ELEMENT(NODMAX)=												8			
MAXIMUM NON-ZERO NODES IN ANY ELEMENTS(NONZER) =												8			
THE FOLLOWING IS RELATED TO MATERIAL NUMBER ASSIGNED															
MAXIMUM NUMBER ASSIGNED TO MATERIAL(MATN) =												5			

TIME= 13153142 DATE= 13-SEP-79

TABLE 10.6. PROG3 Steady State Output of LINEAR.PR1

STEADY STATE SOLUTION

LINEAR ELEMENT 4 LAYERS WITH HELD B,C, AT, SURFACE AND BASE

PROG31 FILE**STEADY STATE FLOW ---- SURFACE AND ELEMENT FLUX = 0.0

-----			-----			-----			-----		
NODE	DEPTH	HEAD	NODE	DEPTH	HEAD	NODE	DEPTH	HEAD	NODE	DEPTH	HEAD

1	400.0	400.00	2	400.0	400.00	3	400.0	400.00	4	400.0	400.00
1001	300.0	300.00	1002	300.0	300.00	1003	300.0	300.00	1004	300.0	300.00
2001	200.0	200.00	2002	200.0	200.00	2003	200.0	200.00	2004	200.0	200.00
3001	100.0	100.00	3002	100.0	100.00	3003	100.0	100.00	3004	100.0	100.00
4001	0.0000	0.00	4002	0.0000	0.00	4003	0.0000	0.00	4004	0.0000	0.00

TABLE 10.7. Use of TRANSIENT.P3I for Transient Simulation with Three Variable Time Steps

FILE NAME= TRANSIENT.P3I

TRANSIENT SOLUTION OF PROBLEM WITH SPECIFIED 4 TIME STEPS

1 3 1 1 0 0 0/INSTEDY,ITOTAL,NQ,NQWCP,MNTRY,NHBOND,NTEXP
.2,.5,1.

TABLE 10.8. Interactive Execution of LINEAR.PR1 with
TRANSIENT.P3I and Variable Time Steps

```

MCR>P3I
  ENTER FILE-Q DISK UNIT
2
  ENTER INPUT FILE NAME
TRANSIENT.P3I
  ENTER PRINT OPTION(0,1,2)
1
  NEW TITLE(1=Y,0=N)
0
  NSTEDY      1 ITOTAL      4 NQWCP      1 NQ      1 MNTRY      0 NHBOND
    0 NTEXP      0
...P3I -- STOP

MCR>PR3
  ENTER FILE-Q DISK UNIT
2
  PRINT OPTION (NORMAL= 1, FOR DEBUG = 2 , NO OUTPUT=0)
  NBEGIN(NORMAL= 1 AND FOR RESTART = SUBROUTINE IN MAIN)
  IT=TIME STEP TO START (NORMAL=1,RESTART=TIME STEP)
  NDIG = OUTPUT DIGITS AFTER DECIMAL IN 'F' FORMAT
  NCONS = 0 FOR TIME STEPS READ IN PROG3I
          = 1 FOR CONSTANT TIME STEP FOR COMPUTATION

  NPRINT(0,1,2),NBEGIN(1,2,3,4,5,6,7),IT,NDIG,NCONS
1,1,1,4,0
  IS EXPECTED BAND WIDTH TO BE MORE THAN 252(1=Y,0=N)
0
  NBEGIN      1 NPRINT      1 IT      1
  ELEMENT MATRIX COMPLETED
  SYSTEM MATRIX FORMED
  MREAD SUBROUTINE COMPLETED
  UPPER TRIANGLE DECOMPOSITION DONE
  B ESTIMATION DONE
  BACK SUB DONE
  TIMESTEP      1 ELAPSEDTIME  0.2000000
  ELEMENT MATRIX COMPLETED
  SYSTEM MATRIX FORMED
  MREAD SUBROUTINE COMPLETED
  UPPER TRIANGLE DECOMPOSITION DONE
  B ESTIMATION DONE
  BACK SUB DONE
  TIMESTEP      2 ELAPSEDTIME  0.5000000
  ELEMENT MATRIX COMPLETED
  SYSTEM MATRIX FORMED
  MREAD SUBROUTINE COMPLETED
  UPPER TRIANGLE DECOMPOSITION DONE
  B ESTIMATION DONE
  BACK SUB DONE
  TIMESTEP      3 ELAPSEDTIME  1.0000000
...PR3 -- STOP

MCR>

```

TABLE 10.9. Interactive Transient Execution of PROG3 with
Constant Time Step

```

PR3
ENTER FILE-Q DISK UNIT
2
PRINT OPTION (NORMAL= 1, FOR DEBUG = 2 , NO OUTPUT=0)
NBEGIN(NORMAL= 1 AND FOR RESTART = SUBROUTINE IN MAIN)
IT=TIME STEP TO START (NORMAL=1,RESTART=TIME STEP)
NDIG = OUTPUT DIGITS AFTER DECIMAL IN "F" FORMAT
NCONS = 0 FOR TIME STEPS READ IN PROG3I
      = 1 FOR CONSTANT TIME STEP FOR COMPUTATION

NPRINT(0,1,2),NBEGIN(1,2,3,4,5,6,7),IT,NDIG,NCONS
1,1,1,4,1
IS EXPECTED BAND WIDTH TO BE MORE THAN 252(1=Y,0=N)
0
CONSTANT DELTAT
.1
PROG3 COMPLETED LAST TIME.THIS IS RERUN
NBEGIN      1 NPRINT      1 IT      1
ELEMENT MATRIX COMPLETED
SYSTEM MATRIX FORMED
MREAD SUBROUTINE COMPLETED
UPPER TRIANGLE DECOMPOSITION DONE
B ESTIMATION DONE
BACK SUB DONE
B ESTIMATION DONE
BACK SUB DONE
TIMESTEP      2 ELAPSED TIME  0.2000000
B ESTIMATION DONE
BACK SUB DONE
B ESTIMATION DONE
BACK SUB DONE
B ESTIMATION DONE
BACK SUB DONE
TIMESTEP      3 ELAPSED TIME  0.5000000
B ESTIMATION DONE
BACK SUB DONE
B ESTIMATION DONE
BACK SUB DONE
B ESTIMATION DONE
BACK SUB DONE
B ESTIMATION DONE
BACK SUB DONE
TIMESTEP      4 ELAPSED TIME  1.0000000
B ESTIMATION DONE
BACK SUB DONE
TIMESTEP      5 ELAPSED TIME  1.1000000
...PR3 -- STOP
MCR>

```


10.3 MIXED ORDER ELEMENT WITH SURFACE FLUX AT TOP AND HELD POTENTIAL AT BASE

For illustrative purposes, this element is formulated with sides of mixed order (Figure 10.2): one cubic (nodes 1,9,10,2), two quadratic (nodes 2,8,3 and 4,7,1) and one linear (nodes 3,4). Table 10.10 is a listing of the input data file (MIXEDELEM.PR1).

FIGURE 10.2. Mixed Order Element - Flux (Read in PROG3I)
At Top - Held Potentials at Base

TABLE 10.10. MIXELEM.PR1 for Mixed Order Element

```

FILE NAME= MIXELEM.PR1

MIXED ORDER ELEMENT - FLUX(READ IN PROGSI) AT TOP - HELD POTENTIALS AT BASE
  1  0  1.0  1.0  1.0  1.0
0  1  1 IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY
  1.0  1.0  1.0  .25  .000001  0.0  .000001  .0000001
0  2  1 IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY
  1.0  1.0  1.0  .25  .000001  0.0  .000001  .0000001
0  3  1 IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY
  1.0  1.0  1.0  .25  .000001  0.0  .000001  .0000001
0  4  1 IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY
  1.0  1.0  1.0  .25  .000001  0.0  .000001  .0000001
0  5  1 IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY
  1.0  1.0  1.0  .25  .000001  0.0  .000001  .0000001

1
0  1  0.0  0.0  0.0  1.0  1  5  01
400.0  0102  300.0  0203  200.0  0304  100.0  0400  0.0
0  2  100.0  0.0  0.0  1.0  1  5  01
400.0  0102  300.0  0203  200.0  0304  100.0  0400  0.0
  3  101.0  100.0  0.0  1.0  1  5  01
400.0  0102  300.0  0203  200.0  0304  100.0  0400  0.0
0  4  1.0  100.0  0.0  1.0  1  5  01
400.0  0102  300.0  0203  200.0  0304  100.0  0400  0.0
  7  0.0  56.  0.0  1.0  1  5  01
400.0  0102  300.0  0203  200.0  0304  100.0  0400  0.0
0  8  100.0  55.0  0.0  1.0  1  5  01
400.0  0102  300.0  0203  200.0  0304  100.0  0400  0.0
  9  39.2  0.0  0.0  1.0  1  5  01
400.0  0102  300.0  0203  200.0  0304  100.0  0400  0.0
0  10  09.0  0.0  0.0  1.0  1  5  01
400.0  0102  300.0  0203  200.0  0304  100.0  0400  0.0
2

04001  1.0
04002  1.0
04003  1.0
04004  1.0
04007  1.0
04008  1.0
04009  1.0
04010  1.0
5
0  1  0  10  1  2  3  4  0  0  7  8  9  10
  321  321  121  121
6

```

As regards the boundary conditions, base nodes (4001 to 4010) of the last element are held at 0 ft elevation and the time invariant surface flux (identical to infiltration) 1 cu ft/day/sq ft is entered in P3I using QSURFACE.P3I (Table 10.11). Since unit hydraulic properties are assigned to all materials, the resultant hydraulic head is also equivalent to the elevation of the nodes. Table 10.12 illustrates the interactive solution of this problem.

TABLE 10.11. QSURFACE.P3I for Surface Flux

```
FILE NAME= QSURFACE.P3I
PRESCRIBING SURFACE FLUX IN PROG3I
      0      0      1      2      0
1.0,      CONVERSION FACTOR
1.0,      INFILTRATION OF ELEM#1 1. L/T
```

10.4 MIXED ORDER ELEMENT WITH HELD AND FLUX BOUNDARY CONDITIONS AT TOP AND HELD POTENTIALS AT BASE

In finite element simulation of the ground water basin, along with a specified surface recharge, one or two nodes of a element may be held nodes (denoting a river, sea or lake shore). In FE3DGW the system matrix of the nodes with unknown potentials only is formed. This permits use of the mixed boundary conditions at the surface.

To illustrate the same, this example (Table 10.13 and Figure 10.3) differs from the preceeding mixed order element by one more held node at the top surface (node #1, head = 400 ft). The interactive execution is illustrated in Table 10.14.

10.5 LINEAR ELEMENT WITH POINT SOURCE AT TOP AND HELD POTENTIALS AT BASE

This example (Figure 10.4 and Table 10.15) differ from the first example by prescribing a point source of 2500 cuft/day at each of the top nodes (1 to 4). Interaction execution is illustrated in Table 11.16.

10.6 RADIAL ELEMENTS WITH EXPONTIALLY INCREASING HORIZONTAL COORDINATES

For simulation of well draw down in a three layered system (one aquifer between two aquitards), radial elements with exponentially increasing horizontal coordinates are usually used.

TABLE 10.12. Interactive Execution of Mixed Order Problem with
MIXELEM.PR1 and QSURFACE.P3I

```

MCR>PR1
  ENTER FILE-Q DISK UNIT
2
  ENTER INPUT FILE NAME
MIXELEM.PR1
  NPRINT(0,5),NPINCH(0=MISSING MATERIAL BY WEDGED ELEM,1=OTHERWISE)
1,0
  NEW PROBLEM(=0) OR INPUT MODIFICATION(=1)
0
...PR1  --  STOP

MCR>PR2
  ENTER FILE-Q DISK UNIT
2
  ENTER PRINT OPTION (0,1,2)
0
...PR2  --  STOP

MCR>BND
  ENTER FILE-Q DISK UNIT
2
  ENTER PRINT OPTION(0,1,2)
0
...BND  --  STOP

MCR>P3I
  ENTER FILE-Q DISK UNIT
2
  ENTER INPUT FILE NAME
QSURFACE.P3I
  ENTER PRINT OPTION(0,1,2)
1
  NEW TITLE(1=Y,0=N)
0
  NSTEDY      0 ITOTAL      0 NQWCP      1 NQ      2 MENTRY      0 NHBOND
    0 NTEXP      0
...P3I  --  STOP

MCR>PR3
  ENTER FILE-Q DISK UNIT
2
  PRINT OPTION (NORMAL= 1, FOR DEBUG = 2 , NO OUTPUT=0)
  NBEGIN(NORMAL= 1 AND FOR RESTART = SUBROUTINE IN MAIN)
  IT=TIME STEP TO START (NORMAL=1,RESTART=TIME STEP)
  NDIG = OUTPUT DIGITS AFTER DECIMAL IN "F" FORMAT
  NCONS = 0 FOR TIME STEPS READ IN PROG3I
          = 1 FOR CONSTANT TIME STEP FOR COMPUTATION

  NPRINT(0,1,2),NBEGIN(1,2,3,4,5,6,7),IT,NDIG,NCONS
1,1,1,2,0
  IS EXPECTED BAND WIDTH TO BE MORE THAN 252(1=Y,0=N)
0
  NBEGIN      1 NPRINT      1 IT      1
  ELEMENT MATRIX COMPLETED
  SYSTEM MATRIX FORMED
  MREAD SUBROUTINE COMPLETED
  UPPER TRIANGLE DECOMPOSITION DONE
  B ESTIMATION DONE
  BACK SUB DONE
...PR3  --  STOP

MCR>

```

FIGURE 10.3. Mixed Order Elem-Bottom and One Top Held Head - Rest Surface Flux B. C.

TABLE 10.13. ELMIXBC.PR1 for Mixed Order Element with Mixed Boundary Conditions at Surface

FILE NAME= ELMIXBC.PR1

MIXED ORDER ELEM-BOTTOM AND ONE TOP HELD HEAD + REST SURFACE FLUX B.C.

1	0	1.0	1.0	1.0	1.0		
0	1	IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY					
	1.0	1.0	1.0	.25	.000001	0.0	.000001 .0000001
0	2	IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY					
	1.0	1.0	1.0	.25	.000001	0.0	.000001 .0000001
0	3	IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY					
	1.0	1.0	1.0	.25	.000001	0.0	.000001 .0000001
0	4	IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY					
	1.0	1.0	1.0	.25	.000001	0.0	.000001 .0000001
0	5	IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY					
	1.0	1.0	1.0	.25	.000001	0.0	.000001 .0000001

1	0	1	0.0	0.0	0.0	0.0	1	5	01
	400.0	0102	300.0	0203	200.0	0304	100.0	0400	0.0
0	2	100.0	0.0	0.0	0.0	0.0	1	5	01
	400.0	0102	300.0	0203	200.0	0304	100.0	0400	0.0
0	3	100.0	100.0	0.0	0.0	0.0	1	5	01
	400.0	0102	300.0	0203	200.0	0304	100.0	0400	0.0
0	4	0.0	100.0	0.0	0.0	0.0	1	5	01
	400.0	0102	300.0	0203	200.0	0304	100.0	0400	0.0
0	7	0.0	56.0	0.0	0.0	0.0	1	5	01
	400.0	0102	300.0	0203	200.0	0304	100.0	0400	0.0
0	8	100.0	55.0	0.0	0.0	0.0	1	5	01
	400.0	0102	300.0	0203	200.0	0304	100.0	0400	0.0
0	9	39.2	0.0	0.0	0.0	0.0	1	5	01
	400.0	0102	300.0	0203	200.0	0304	100.0	0400	0.0
0	10	89.0	0.0	0.0	0.0	0.0	1	5	01
	400.0	0102	300.0	0203	200.0	0304	100.0	0400	0.0

2

1	400.0
9	400.0
04001	0.0
04002	0.0
04003	0.0
04004	0.0
04007	0.0
04008	0.0
04009	0.0
04010	0.0

5

0	1	0	10	1	2	3	4	0	0	7	8	9	10
	321	321	121	121									

6

TABLE 10.14. Interactive Execution with ELMIXBC.PR1
and QSURFACE.P3I

```

MCR>PR1
  ENTER FILE-Q DISK UNIT
2
  ENTER INPUT FILE NAME
ELMIXBC.PR1
  NPRINT(0,5),NFINCH(0=MISSING MATERIAL BY WEIGED ELEM,1=OTHERWISE)
3,0
  NEW PROBLEM(=0) OR INPUT MODIFICATION(=1)
0
...PR1 -- STOP

MCR>PR2
  ENTER FILE-Q DISK UNIT
2
  ENTER PRINT OPTION (0,1,2)
0
...PR2 -- STOP

MCR>BND
  ENTER FILE-Q DISK UNIT
2
  ENTER PRINT OPTION(0,1,2)
0
...BND -- STOP

MCR>P3I
  ENTER FILE-Q DISK UNIT
2
  ENTER INPUT FILE NAME
QSURFACE.P3I
  ENTER PRINT OPTION(0,1,2)
0
  NEW TITLE(1=Y,0=N)
0
  NSTEDY      0 ITOTAL      0 NQWCP      1 NQ      2 MNTRY      0 NHBOND
    0 NTEXP      0
...P3I -- STOP

MCR>PR3
  ENTER FILE-Q DISK UNIT
2
  PRINT OPTION (NORMAL= 1, FOR DEBUG = 2 , NO OUTPUT=0)
  NBEGIN(NORMAL= 1 AND FOR RESTART = SUBROUTINE IN MAIN)
  IT=TIME STEP TO START (NORMAL=1,RESTART=TIME STEP)
  NDIG = OUTPUT DIGITS AFTER DECIMAL IN 'F' FORMAT
  NCONS = 0 FOR TIME STEPS READ IN PROG3I
         = 1 FOR CONSTANT TIME STEP FOR COMPUTATION

  NPRINT(0,1,2),NBEGIN(1,2,3,4,5,6,7),IT,NDIG,NCONS
1,1,1,2,0
  IS EXPECTED BAND WIDTH TO BE MORE THAN 252(1=Y,0=N)
0
  NBEGIN      1 NPRINT      1 IT      1
  ELEMENT MATRIX COMPLETED
  SYSTEM MATRIX FORMED
  MREAD SUBROUTINE COMPLETED
  UPPER TRIANGLE DECOMPOSITION DONE
  R ESTIMATION DONE
  BACK SUB DONE
...PR3 -- STOP

MCR>

```

FIGURE 10.4. Linear Element 4 Layers with Nodal Flux at Surface and Held B.C. at Base

TABLE 10.15. LINODALQ.PR1 for a Linear Element with Point Source

FILE NAME= LINODALQ.PR1

LINEAR ELEMENT 4 LAYERS WITH NODAL FLUX AT SURFACE AND HELD B.C. AT BASE

```

1 0 1.0 1.0 1.0 1.0
0 1 IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY
1.0 1.0 1.0 .25 .000001 0.0 .000001 .0000001
0 2 IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY
1.0 1.0 1.0 .25 .000001 0.0 .000001 .0000001
0 3 IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY
1.0 1.0 1.0 .25 .000001 0.0 .000001 .0000001
0 4 IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY
1.0 1.0 1.0 .25 .000001 0.0 .000001 .0000001
0 5 IMAGINARY MATERIAL WITH ALL PROPERTIES EQUAL TO UNITY
1.0 1.0 1.0 .25 .000001 0.0 .000001 .0000001

```

1 SWITCH FOR NODAL DATA AND END OF PROPERTIES

```

0 1 0.0 0.0 0.0 0.0 0.0 1 5 01
400.0 0102 300.0 0203 200.0 0304 100.0 0400 0.0
0 2 100.0 0.0 0.0 0.0 0.0 1 5 01
400.0 0102 300.0 0203 200.0 0304 100.0 0400 0.0
0 3 100.0 100.0 0.0 0.0 0.0 1 5 01
400.0 0102 300.0 0203 200.0 0304 100.0 0400 0.0
0 4 0.0 100.0 0.0 0.0 0.0 1 5 01
400.0 0102 300.0 0203 200.0 0304 100.0 0400 0.0
2

```

```

04001 0.0
04002 0.0
04003 0.0
04004 0.0

```

```

4
1 2500.0
2 2500.0
3 2500.0
4 2500.0

```

```

5
0 1 1 0 1 2 3 4
6 SWITCH FOR END OF DATA

```

TABLE 10.16. Interactive Execution with LINODALQ.PR1 and HSTEADY.P3I

```

MCR>PR1
  ENTER FILE-Q DISK UNIT
  2
  ENTER INPUT FILE NAME
  LINODALQ.PR1
  NPRINT(0,5),NPINCH(0=MISSING MATERIAL BY WEDGED ELEM,1=OTHERWISE)
  2,0
  NEW PROBLEM(=0) OR INPUT MODIFICATION(=1)
  0
  ...PR1 -- STOP

MCR>PR2
  ENTER FILE-Q DISK UNIT
  2
  ENTER PRINT OPTION (0,1,2)
  0
  ...PR2 --- STOP

MCR>BND
  ENTER FILE-Q DISK UNIT
  2
  ENTER PRINT OPTION(0,1,2)
  0
  ...BND -- STOP

MCR>P3I
  ENTER FILE-Q DISK UNIT
  2
  ENTER INPUT FILE NAME
  HSTEADY.P3I
  ENTER PRINT OPTION(0,1,2)
  0
  NEW TITLE(1=Y,0=N)
  0
  NSTEDY      0 ITOTAL      0 NQWCP      1 NQ      1 MENTRY      0 NHBOND
    0 NTEXP      0
  ...P3I --- STOP

MCR>PR3
  ENTER FILE-Q DISK UNIT
  2
  PRINT OPTION (NORMAL= 1, FOR DEBUG = 2 , NO OUTPUT=0)
  NBEGIN(NORMAL= 1 AND FOR RESTART = SUBROUTINE IN MAIN)
  IT=TIME STEP TO START (NORMAL=1,RESTART=TIME STEP)
  NDIG = OUTPUT DIGITS AFTER DECIMAL IN 'F' FORMAT
  NCONS = 0 FOR TIME STEPS READ IN PROG3I
          = 1 FOR CONSTANT TIME STEP FOR COMPUTATION

  NPRINT(0,1,2),NBEGIN(1,2,3,4,5,6,7),IT,NDIG,NCONS
  1,1,1,2,0
  IS EXPECTED BAND WIDTH TO BE MORE THAN 252(1=Y,0=N)
  0
  NBEGIN      1 NPRINT      1 IT      1
  ELEMENT MATRIX COMPLETED
  SYSTEM MATRIX FORMED
  MREAD SUBROUTINE COMPLETED
  UPPER TRIANGLE DECOMPOSITION DONE
  B ESTIMATION DONE
  BACK SUB DONE
  ...PR3 -- STOP

MCR>

```

In this example (Figure 10.5 and Table 10.17) four elements with held potentials at the surface (top of first aquitard), base (bottom of second aquitard) and also the extreme end (away from well) are used. Withdrawal from the well is defined as a point sink (at nodes 2001, 2002, 3001, 3002) of the pumped aquifer. Exponential time steps ($\Delta T_t = \Delta T_{t-1} * \text{Factor}$) are designed by input data file, QTRANS.P3I (Table 10.18). The interactive execution is illustrated in Table 10.19. Table 10.20 includes some of the output of PR3.

Because the illustrative simulated region is only a fraction of the total, the initial time step prescribed is extremely small (1.0×10^{-6} day). For the complete simulation of well drawdown in an extensive aquifer system, the four elements in this example have to be expanded with additional elements such that within the simulated time period, drawdown curve does not reach the boundary.

TABLE 10.17. WELL.PR1 for Three Layered and Four Element
Pie Shaped Region

FILE NAME= WELL.PR1

RADIAL 4 ELEMENTS THREE LAYERED AND EXPONENTIAL INCREASING DISTANCE

1 0 1.00 1.00 1.00 1.00 1.00
1 MATERIAL OF AQUIFER IN LAYER A
0.100E+02 0.100E+02 0.100E+02 0.100E+00 0.100E+01 0.100E+01 0.100E+01 0.200E+03
2 MATERIAL OF AQUITARD (LAYER B)
0.100E-03 0.100E-03 0.100E-03 0.100E+00 0.100E+01 0.100E+01 0.100E+01 0.200E+04
3 MATERIAL OF PUMPED AQUIFER (LAYER C)
0.100E+03 0.100E+03 0.100E+03 0.100E+00 0.100E+01 0.100E+01 0.100E+01 0.100E-03
4 MATERIAL OF AQUITARD (LAYER D)
0.100E-04 0.100E-04 0.100E-04 0.100E+00 0.100E+01 0.100E+01 0.100E+01 0.200E-04
5 MATERIAL OF AQUIFER (LAYER E)
0.100E+02 0.100E+02 0.100E+02 0.100E+00 0.100E+01 0.100E+01 0.100E+01 0.200E+03

1
1 1.000 0.0000 0.0000 -50.00 1 4 2
0.0 203 -20.0 304 -30.0 400 -50.0
20.9239 0.3027 0.0000 -50.00 1 4 2
0.0 203 -20.0 304 -30.0 400 -50.0
3 1.649 0.0000 0.0000 -50.00 1 4 2
0.0 203 -20.0 304 -30.0 400 -50.0
4 1.523 0.6329 0.0000 -50.00 1 4 2
0.0 203 -20.0 304 -30.0 400 -50.0
5 2.718 0.0000 0.0000 -50.00 1 4 2
0.0 203 -20.0 304 -30.0 400 -50.0
6 2.511 1.040 0.0000 -50.00 1 4 2
0.0 203 -20.0 304 -30.0 400 -50.0
7 4.482 0.0000 0.0000 -50.00 1 4 2
0.0 203 -20.0 304 -30.0 400 -50.0
8 4.141 1.715 0.0000 -50.00 1 4 2
0.0 203 -20.0 304 -30.0 400 -50.0
9 7.389 0.0000 0.0000 -50.00 1 4 2
0.0 203 -20.0 304 -30.0 400 -50.0
10 6.027 2.029 0.0000 -50.00 1 4 2
0.0 203 -20.0 304 -30.0 400 -50.0

2
0
09 0.000000
2009 0.000000
3009 0.000000
4009 0.000000
10 0.000000
2010 0.000000
3010 0.000000
4010 0.000000
1 0.000000
4001 0.000000
2 0.000000
4002 0.000000
3 0.000000
4003 0.000000
4 0.000000
4004 0.000000
5 0.000000
4005 0.000000
6 0.000000
4006 0.000000
7 0.000000
4007 0.000000
8 0.000000
4008 0.000000
10 0.000000

4
2001=1.562
3001=1.562
3002=1.562
2002=1.562

5
1 1 4 9 10 8 7
2 1 4 7 8 6 5
3 1 4 5 6 4 3
4 1 4 3 4 2 1

6

TABLE 10.18. PTRANS.P3I for Exponentially Incremented Time Step

FILE NAME= PTRANS.P3I

TRANSIENT SOLUTION WITH GENERATED EXPONENTIAL TIME STEP (PTRANS,P3I)

1 5 1 1 0 0 1 NSTEDY, ITOTAL, NQ, NQWCP, MNTRY, NHBOND, EXP
1,E-6 1.5 INITIAL TIME AND FACTOR FOR INCREASING NEXT

TABLE 10.19. Interactive Execution of Radial Problem

```

MCR>PR1
  ENTER FILE-Q DISK UNIT
2
  ENTER INPUT FILE NAME
WELL.PR1
  NPRINT(0,5),NFINCH(0=MISSING MATERIAL BY WEDGED ELEM,1=OTHERWISE)
3,0
  NEW PROBLEM(=0) OR INPUT MODIFICATION(=1)
0
...PR1 -- STOP

MCR>PR2
  ENTER FILE-Q DISK UNIT
2
  ENTER PRINT OPTION (0,1,2)
0
...PR2 -- STOP

MCR>BND
  ENTER FILE-Q DISK UNIT
2
  ENTER PRINT OPTION(0,1,2)
0
...BND -- STOP

MCR>P3I
  ENTER FILE-Q DISK UNIT
2
  ENTER INPUT FILE NAME
PTRANS.P3I
  ENTER PRINT OPTION(0,1,2)
1
  NEW TITLE(1=Y,0=N)
0
  NSTEDY      1 ITOTAL      5 NQWCP      1 NQ      1 MNTRY      0 NHBOND
    0 NTEXP      1
...P3I -- STOP

MCR>PR3
  ENTER FILE-Q DISK UNIT
2
  PRINT OPTION (NORMAL= 1, FOR DEBUG = 2 , NO OUTPUT=0)
  NBEGIN(NORMAL= 1 AND FOR RESTART = SUBROUTINE IN MAIN)
  IT=TIME STEP TO START (NORMAL=1,RESTART=TIME STEP)
  NDIG = OUTPUT DIGITS AFTER DECIMAL IN "F" FORMAT
  NCONS = 0 FOR TIME STEPS READ IN PROG3I
         = 1 FOR CONSTANT TIME STEP FOR COMPUTATION

```

TABLE 10.19. (contd)

```

NPRINT(0,1,2),NBEGIN(1,2,3,4,5,6,7),IT,NDIG,NCONS
1,1,1,2,0
IS EXPECTED BAND WIDTH TO BE MORE THAN 252(1=Y,0=N)
0
NBEGIN      1 NPRINT      1 IT      1
ELEMENT MATRIX COMPLETED
SYSTEM MATRIX FORMED
MREAD SUBROUTINE COMPLETED
UPPER TRIANGLE DECOMPOSITION DONE
B ESTIMATION DONE
BACK SUB DONE
TIMESTEP      1 ELAPSED TIME  1.0000000E-06
ELEMENT MATRIX COMPLETED
SYSTEM MATRIX FORMED
MREAD SUBROUTINE COMPLETED
UPPER TRIANGLE DECOMPOSITION DONE
B ESTIMATION DONE
BACK SUB DONE
TIMESTEP      2 ELAPSED TIME  2.5000002E-06
ELEMENT MATRIX COMPLETED
SYSTEM MATRIX FORMED
MREAD SUBROUTINE COMPLETED
UPPER TRIANGLE DECOMPOSITION DONE
B ESTIMATION DONE
BACK SUB DONE
TIMESTEP      3 ELAPSED TIME  4.7500002E-06
ELEMENT MATRIX COMPLETED
SYSTEM MATRIX FORMED
MREAD SUBROUTINE COMPLETED
UPPER TRIANGLE DECOMPOSITION DONE
B ESTIMATION DONE
BACK SUB DONE
TIMESTEP      4 ELAPSED TIME  8.1250009E-06
ELEMENT MATRIX COMPLETED
SYSTEM MATRIX FORMED
MREAD SUBROUTINE COMPLETED
UPPER TRIANGLE DECOMPOSITION DONE
B ESTIMATION DONE
BACK SUB DONE
TIMESTEP      5 ELAPSED TIME  1.3187501E-05
...PR3  --  STOP
MCR>

```

TABLE 10.20. Sample Output of P3I and PR3 for WELL.PR1 and QTRANS.P3I

 TRANSIENT SOLUTION WITH GENERATED EXPONENTIAL TIME STEP (PTRANS,P3I)

NSTEDY 1 ITOTAL 5 NQWCP 1 NO 1 MNTY 0 NHBOND
 0 NTEXP 1
 TRANSIENT RUN OF MODEL PRESCRIBED

TOTAL NUMBER OF TIME STEPS(ITOTAL) 5

1	TIME STEP	0.1000E-05	ELAPSED TIME	0.1000E-05
2	TIME STEP	0.1500E-05	ELAPSED TIME	0.2500E-05
3	TIME STEP	0.2250E-05	ELAPSED TIME	0.4750E-05
4	TIME STEP	0.3375E-05	ELAPSED TIME	0.8125E-05
5	TIME STEP	0.5063E-05	ELAPSED TIME	0.1319E-04

NO NEW SOURCE OR SINK FOR EACH ELEMENT* IS READ IN. SOLUTION BASED ON SOURCE/SINK READ* IN PROGRAM 1

NO SURFACE FLUX IS PRESCRIBED

TABLE 10.20. (contd)

RESULTS OF HEAD ESTIMATION AT TIME STEP 1 ELAPSED TIME 0.1000E-05 DELTAT= 0.1000E-05

RADIAL 4 ELEMENTS SINGLE LAYER EXPONENTIAL INCREASING DISTANCE

TRANSIENT SOLUTION WITH GENERATED EXPONENTIAL TIME STEP (PTRANS,P3I)

NODE	DEPTH	HEAD	NODE	DEPTH	HEAD	NODE	DEPTH	HEAD	NODE	DEPTH	HEAD
1	0.0000	0.00000	2	0.0000	0.00000	3	0.0000	0.00000	4	0.0000	0.00000
2001	-20.00	-0.00954	2002	-20.00	-0.00954	2003	-20.00	-0.00358	2004	-20.00	-0.00358
3001	-30.00	-0.00954	3002	-30.00	-0.00954	3003	-30.00	-0.00358	3004	-30.00	-0.00358
4001	-50.00	0.00000	4002	-50.00	0.00000	4003	-50.00	0.00000	4004	-50.00	0.00000
5	0.0000	0.00000	6	0.0000	0.00000	7	0.0000	0.00000	8	0.0000	0.00000
2005	-20.00	-0.00076	2006	-20.00	-0.00076	2007	-20.00	-0.00004	2008	-20.00	-0.00004
3005	-30.00	-0.00076	3006	-30.00	-0.00076	3007	-30.00	-0.00004	3008	-30.00	-0.00004
4005	-50.00	0.00000	4006	-50.00	0.00000	4007	-50.00	0.00000	4008	-50.00	0.00000
9	0.0000	0.00000	10	0.0000	0.00000						
2009	-20.00	0.00000	2010	-20.00	0.00000						
3009	-30.00	0.00000	3010	-30.00	0.00000						
4009	-50.00	0.00000	4010	-50.00	0.00000						

TABLE 10.20. (contd)

RESULTS OF HEAD ESTIMATION AT TIME STEP 2 ELAPSED TIME 0.2500E-03 DELTAT= 0.1500E-05

RADIAL 4 ELEMENTS SINGLE LAYER EXPONENTIAL INCREASING DISTANCE

TRANSIENT SOLUTION WITH GENERATED EXPONENTIAL TIME STEP (PTRANS,P3I)

NODE	DEPTH	HEAD	NODE	DEPTH	HEAD	NODE	DEPTH	HEAD	NODE	DEPTH	HEAD
1	0.0000	0.00000	2	0.0000	0.00000	3	0.0000	0.00000	4	0.0000	0.00000
2001	-20.00	-0.01443	2002	-20.00	-0.01443	2003	-20.00	-0.00744	2004	-20.00	-0.00745
3001	-30.00	-0.01443	3002	-30.00	-0.01443	3003	-30.00	-0.00744	3004	-30.00	-0.00745
4001	-50.00	0.00000	4002	-50.00	0.00000	4003	-50.00	0.00000	4004	-50.00	0.00000
5	0.0000	0.00000	6	0.0000	0.00000	7	0.0000	0.00000	8	0.0000	0.00000
2005	-20.00	-0.00257	2006	-20.00	-0.00257	2007	-20.00	-0.00039	2008	-20.00	-0.00039
3005	-30.00	-0.00257	3006	-30.00	-0.00257	3007	-30.00	-0.00039	3008	-30.00	-0.00039
4005	-50.00	0.00000	4006	-50.00	0.00000	4007	-50.00	0.00000	4008	-50.00	0.00000
9	0.0000	0.00000	10	0.0000	0.00000						
2009	-20.00	0.00000	2010	-20.00	0.00000						
3009	-30.00	0.00000	3010	-30.00	0.00000						
4009	-50.00	0.00000	4010	-50.00	0.00000						

TABLE 10.20. (contd)

RESULTS OF HEAD ESTIMATION AT TIME STEP 3 ELAPSED TIME 0.4750E-05 DELTAT= 0.2250E-05

RADIAL 4 ELEMENTS SINGLE LAYER EXPONENTIAL INCREASING DISTANCE

TRANSIENT SOLUTION WITH GENERATED EXPONENTIAL TIME STEP (PTRANS,P3I)

NODE	DEPTH	HEAD	NODE	DEPTH	HEAD	NODE	DEPTH	HEAD	NODE	DEPTH	HEAD
1	0.0000	0.00000	2	0.0000	0.00000	3	0.0000	0.00000	4	0.0000	0.00000
2001	-20.00	-0.01023	2002	-20.00	-0.01023	2003	-20.00	-0.01092	2004	-20.00	-0.01093
3001	-30.00	-0.01023	3002	-30.00	-0.01023	3003	-30.00	-0.01092	3004	-30.00	-0.01093
4001	-50.00	0.00000	4002	-50.00	0.00000	4003	-50.00	0.00000	4004	-50.00	0.00000
5	0.0000	0.00000	6	0.0000	0.00000	7	0.0000	0.00000	8	0.0000	0.00000
2005	-20.00	-0.00494	2006	-20.00	-0.00494	2007	-20.00	-0.00127	2008	-20.00	-0.00127
3005	-30.00	-0.00494	3006	-30.00	-0.00494	3007	-30.00	-0.00127	3008	-30.00	-0.00127
4005	-50.00	0.00000	4006	-50.00	0.00000	4007	-50.00	0.00000	4008	-50.00	0.00000
9	0.0000	0.00000	10	0.0000	0.00000						
2009	-20.00	0.00000	2010	-20.00	0.00000						
3009	-30.00	0.00000	3010	-30.00	0.00000						
4009	-50.00	0.00000	4010	-50.00	0.00000						

TABLE 10.20. (contd)

RESULTS OF HEAD ESTIMATION AT TIME STEP 4 ELAPSED TIME 0.0125E-05 DELTAT= 0.3375E-05

RADIAL 4 ELEMENTS SINGLE LAYER EXPONENTIAL INCREASING DISTANCE

TRANSIENT SOLUTION WITH GENERATED EXPONENTIAL TIME STEP (PTRANS,P31)

NODE	DEPTH	HEAD	NODE	DEPTH	HEAD	NODE	DEPTH	HEAD	NODE	DEPTH	HEAD

1	0.0000	0.00000	2	0.0000	0.00000	3	0.0000	0.00000	4	0.0000	0.00000
2001	-20.00	-0.02161	2002	-20.00	-0.02161	2003	-20.00	-0.01415	2004	-20.00	-0.01415
3001	-30.00	-0.02161	3002	-30.00	-0.02161	3003	-30.00	-0.01415	3004	-30.00	-0.01415
4001	-50.00	0.00000	4002	-50.00	0.00000	4003	-50.00	0.00000	4004	-50.00	0.00000
5	0.0000	0.00000	6	0.0000	0.00000	7	0.0000	0.00000	8	0.0000	0.00000
2005	-20.00	-0.00753	2006	-20.00	-0.00753	2007	-20.00	-0.00263	2008	-20.00	-0.00263
3005	-30.00	-0.00753	3006	-30.00	-0.00753	3007	-30.00	-0.00263	3008	-30.00	-0.00263
4005	-50.00	0.00000	4006	-50.00	0.00000	4007	-50.00	0.00000	4008	-50.00	0.00000
9	0.0000	0.00000	10	0.0000	0.00000						
2009	-20.00	0.00000	2010	-20.00	0.00000						
3009	-30.00	0.00000	3010	-30.00	0.00000						
4009	-50.00	0.00000	4010	-50.00	0.00000						

TABLE 10.20. (contd)

RESULTS OF HEAD ESTIMATION AT TIME STEP 5 ELAPSED TIME 0.1319E-04 DELTAT= 0.5063E-05

RADIAL 4 ELEMENTS SINGLE LAYER EXPONENTIAL INCREASING DISTANCE

TRANSIENT SOLUTION WITH GENERATED EXPONENTIAL TIME STEP (PTRANS,P3I)

NODE	DEPTH	HEAD	NODE	DEPTH	HEAD	NODE	DEPTH	HEAD	NODE	DEPTH	HEAD
1	0.0000	0.00000	2	0.0000	0.00000	3	0.0000	0.00000	4	0.0000	0.00000
2001	-20.00	-0.02466	2002	-20.00	-0.02466	2003	-20.00	-0.01711	2004	-20.00	-0.01711
3001	-30.00	-0.02466	3002	-30.00	-0.02466	3003	-30.00	-0.01711	3004	-30.00	-0.01711
4001	-50.00	0.00000	4002	-50.00	0.00000	4003	-50.00	0.00000	4004	-50.00	0.00000
5	0.0000	0.00000	6	0.0000	0.00000	7	0.0000	0.00000	8	0.0000	0.00000
2005	-20.00	-0.01008	2006	-20.00	-0.01008	2007	-20.00	-0.00419	2008	-20.00	-0.00419
3005	-30.00	-0.01008	3006	-30.00	-0.01008	3007	-30.00	-0.00419	3008	-30.00	-0.00419
4005	-50.00	0.00000	4006	-50.00	0.00000	4007	-50.00	0.00000	4008	-50.00	0.00000
9	0.0000	0.00000	10	0.0000	0.00000						
2009	-20.00	0.00000	2010	-20.00	0.00000						
3009	-30.00	0.00000	3010	-30.00	0.00000						
4009	-50.00	0.00000	4010	-50.00	0.00000						

11.0 REFERENCES

1. Bredehoeft, J. D. and G. F. Pinder. 1970. Digital analysis of areal flow in multiaquifer ground water sytem: A quasi three-dimensional model. Water Resour. Res. 6(3):883-888.
2. Saleem, Z. A. 1973. Method for numerical solution of flow in multiaquifer systems. Water Resour. Res. 9(5):1465-1469.
3. Jacob, C. E. 1950. Chapter 5, IN: Engineering Hydraulics. (Ed.) H. Rouse, John Wiley and Sons, New York. p. 321-386.
4. Hantush, M. S. 1960. Modification of theory of leaky aquifers. J. Geophys. Res. 65:3713-3725.
5. DeWiest, R. J. M. 1966. On the storage coefficient and the equations of ground water flow. J. Geophys. Res. 71:1117-1122.
6. Cooper, Hilton H., Jr. 1966. The equation of ground water flow in fixed and deforming coordinates. J. Geophys. Res. 71(20):4785-4790.
7. Pinder, G. F. and E. O. Frind. 1972. Application of Galerkin's procedure to aquifer analysis. Water Resour. Res. 8(1):108-120.
8. Gupta, S. K., K. K. Tanji and J. N. Luthin. 1975. A Three- dimensional finite element ground water model, California Water Resource Center, Univ. of California, Davis, Contribution #152, p. 119.
9. Price, H. S., J. C. Cavendish and R. S. Varga. 1968. Numerical methods of higher order accuracy for diffusion-convection equations. Soc. Petrol. Eng. J. 8(3):293-303.
10. Cavendish, J. E., H. S. Price and R. S. Varga. 1968. Variational methods for numerical solution of boundary value problems. Paper presented at Symposium on Numerical Simulation of Reservoir Performance, Soc. Petrol. Eng., Dallas, Texas. April, 22-23.
11. Culham, W. E. and R. S. Varga. 1970. Numerical methods for time dependent nonlinear boundary value problems. Paper presented at 2nd Symposium on Numerical Simulation of Reservoir Performance, Soc. Petrol. Eng., Dallas, Texas.
12. Zienkiewicz, O. C. and Y. K. Cheung. 1965. Finite elements in the solutions of field problems. The Engineer, September.

13. Zienkiewicz, O. C. 1971. The Finite Element Method in Engineering Science. McGraw-Hill, London. p. 129-153.
14. Gupta, S. K., and G. F. Pinder. 1978. Three-Dimensional Finite Element Model for Multilayered Ground-Water Reservoir of Long-Island, New York. Dept. of Civil Engineering, Princeton University, Princeton, 100 p.
15. Gupta, S. K. and K. K. Tanji. 1977. Computer Program for Solution of Large, Sparse Unsymmetric Systems of Linear Equations. Int. J. Num. Meth. in Eng., Vol. 11, p. 1251-1259.
16. Nelson, R. W., and J. A. Schur. 1978. A Preliminary Evaluation Capability for Some Two-Dimensional Ground-Water Contamination Problems. BCSR-38, Boeing Computer Services, Richland, WA.
17. Hantush, M. S., "Nonsteady Flow to Flowing Wells in Leaky Aquifers," Journal of Geophysical Research, Volume 64, No. 8, pp 1043-1052, 1959.
18. Curtin, G. 1971. Hydrogeology of the Sutter Basin, Sacramento Valley, California. M.S. Thesis, Univ. of Arizona. 83 pages.
19. Gupta, S. K. and K. K. Tanji. 1976. A three-dimensional Galerkin finite element solution of flow through mutliaquifers in Sutter Basin, California 12(2) 155-162P.

DISTRIBUTION

<u>No. of Copies</u>		<u>No. of Copies</u>	
	<u>OFFSITE</u>		J. E. Campbell Div. 54.3 Sandia Laboratories Albuquerque, NM 87115
	A. A. Churm DOE Chicago Patent Group 9800 South Cass Avenue Argonne, IL 60439		
27	DOE Technical Information Center	20	Wayne A. Carbiener Office of Nuclear Waste Isolation Battelle Memorial Institute 505 King Avenue Columbus, OH 43201
2	Argonne National Laboratory Reference Library 9800 South Cass Avenue Argonne, IL 60439		H. C. Claiborne Union Carbide Corp. P.O. Box Y Oak Ridge, TN 39380
	G. E. Barr Div. 1141 Sandia Laboratories Albuquerque, NM 87115		Neville G. W. Cook Dept. of Materials Science and Mineral Engineering Hearst Mining Building University of California Berkeley, CA 94720
	Battelle Memorial Institute Office of Nuclear Waste Isolation Attn: Beverly Rawles 505 King Avenue Columbus, OH 43201		Carl R. Cooley DOE Office of Waste Management Washington, D. C. 20545
	John Bird Geology Department Cornell University Ithaca, NY 14853		Jared Davis Nuclear Regulatory Commission Washington, DC 20555
	J. D. Bredehoeft U.S. Geological Survey Water Resources Division Reston, VA 22092		George DeBucharanne U. S. Geological Survey Reston, VA 22092
2	Brookhaven National Laboratory Reference Section Information Division Upton, Long Island, NY 11973		R. T. Dillon Div. 5413 Sandia Laboratories Albuquerque, NM 87115

No. of
Copies

J. O. Duguid
Office of Nuclear Waste
Isolation
Battelle Memorial Institute
505 King Ave.
Columbus, OH 43201

Environmental Protection Agency
Office of Radiation Programs
Technical Assessment Division
AW559
Washington, DC 20460

Geraghty and Miller, Inc.
44 Sintsink Dr. East
Port Washington, NY 11050

Robert M. Garrells
Dept. of Geologic Sciences
Northwestern University
Princeton, NY 08540

W. G. Gray
Dept. of Civil Engineering
Princeton University
Princeton, NJ 08540

D. B. Grove
U.S. Geological Survey
WRD
Mail Stop 413
Denver Federal Center
Denver, CO 80225

John Handin, Director
Center for Tectonophysics
Texas A>M University
College Station, TX 77840

Colin A. Heath
DOE Division of Waste Management
Washington, DC 20545

William M. Hewitt
Office of Nuclear Waste
Isolation
Battelle Memorial Institute
505 King Avenue
Columbus, OH 43201

No. of
Copies

Peter L. Hofmann
Office of Nuclear Waste
Isolation
Battelle Memorial Institute
505 King Avenue
Columbus, OH 43201

Muzaffer Kehnemuyi
Office of Nuclear Waste
Isolation
Battelle Memorial Institute
505 King Avenue
Columbus, OH 43201

John F. Kircher
Office of Nuclear Waste
Isolation
Battelle Memorial Institute
505 King Avenue
Columbus, OH 43201

Ronald B. Lantz
Inter-Environmental Consultants
11511 Katy Freeway
Houston, TX 77079

2 Lawrence Berkeley Laboratory
Reference Library
University of California
Berkeley, CA 94720

2 Lawrence Livermore Laboratory
Reference Library
P. O. Box 808
Livermore, CA 94550

D. I. Leap
U.S. Geological Survey
Water Resources Division
MS 416
Denver Federal Center
Denver, CO 80225

S. E. Logan
Los Alamos Technical
Associates, Inc.
P.O. Box 410
Los Alamos, NM 87544

No. of
Copies

No. of
Copies

2 Los Alamos Scientific Laboratory
Reference Library
P.O. Box 1663
Los Alamos, NM 87544

John Lyons
Dept. of Earth Sciences
Dartmouth College
Hanover, NH 03755

R. D. MacNish
U.S. Geological Survey
Water Resources Div.
301 W. Congress
Tucson, AR 85701

J. B. Martin
Asst. Director for Radioactive
Waste Mgmt Branch
NRC Division of Materials and
Fuel Cycle Facility Licensing
Washington, DC 20555

John T. McGinnis
Office of Nuclear Waste
Isolation
Battelle Memorial Institute
505 King Avenue
Columbus, OH 43201

Sheldon Meyers
DOE Office of Nuclear Waste
Management
Washington, DC 20545

P. A. Mote
Bechtel National, Inc.
50 Beale St.
San Francisco, CA 94119

2 Barry Naft
NUS Corporation
4 Research Place
Rockville, MD 20805

T. N. Narasimhan
Lawrence Berkeley Laboratory
Berkeley, CA 94720

J. O. Neff
Department of Energy
Columbus Program Office
505 King Avenue
Columbus, OH 43201

2 Neil A. Norman
Environmental Sciences
Department
Bechtel National Inc.
P.O. Box 3965
San Francisco, CA 94105

2 Oak Ridge National Laboratory
Central Research Library
Document Reference Section
Oak Ridge, TN 37830

Frank L. Parker
Dept. of Environmental Eng.
Vanderbilt University
Nashville, TN 37235

L. W. Picking
Stone and Webster Eng. Corp.
245 Summer St.
Boston, MA 02107

George Pinder
Dept. of Civil Engineering
Princeton University
Princeton, NJ 08540

Gary Robbins
U.S. Nuclear Regulatory
Commission
Division of Waste Management
Washington, DC 20555

2 Savannah River Laboratory
Reference Library
Aiken, SC 29801

G. Segol
Bechtel National, Inc.
50 Beale St.
San Francisco, CA 94119

No. of
Copies

Raymond Siever
Dept. of Geological Sciences
Harvard University
Cambridge, MA 02138

Howard P. Stephens
Sandia Laboratories
P.O. Box 5800
Albuquerque, NM 87115

David B. Stewart
U.S. Geological Survey
National Center 959
Reston, VA 22092

Lewis D. Thorson
Lawrence Livermore Laboratory
Box 803, L-224
Livermore, CA 94550

N. P. Timofeef
State University of New York
Binghamton, NY 13901

Wendal Weart
Division 1140
Sandia Laboratories
Albuquerque, NM 87115

Robert Williams
Electric Power Research Inst.
3412 Hillview Avenue
P.O. Box 10412
Palo Alto, CA 94304

Paul A. Witherspoon
Lawrence Berkeley Laboratory
University of California
Berkeley, CA 94710

FOREIGN

D'Allessandro Avogadro
Commission of European
Communities
Joint Research Centre
I-21020 Ispra (Varese)
ITALY

No. of
Copies

V. K. Barwell
Environmental Research Branch
Atomic Energy of Canada, Ltd.
Chalk River, Ontario, K0J1J0
CANADA

Leif Carlsson
Chief Hydrologist
Geological Survey of Sweden
Bredgrand 4
S-753 20 Uppsala
SWEDEN

J. A. Cherry
Dept. of Earth Sciences
Waterloo, Ontario
CANADA

K. W. Dormuth
Storage and Disposal Branch
Atomic Energy of Canada
Research Company
Whiteshell Nuclear Research
Establishment
Pinawa, Manitoba R0E 1L10
CANADA

Dr. John E. Gale
Department of Earth Sciences
University of Waterloo
Waterloo, Ontario N24 3G1
CANADA

Ferruccio Gera
Radiation Protection and
Waste Management Division
Nuclear Energy Agency/OECD
38 boulevard Suchet
75016 Paris
FRANCE

K. H. Hubenthal
Bundesministerium fur Forschung
und Technologie
Stressemannstrasse 2
Postfach 200706
D-5300 Bonn
F.R. of GERMANY

No. of
Copies

2 INIS Clearinghouse
International Atomic Energy
Agency
P.O. Box 590
A-1011, Vienna
AUSTRIA

Klaus Kuhn
Institute fur Tiefagerung
Wissenschaftliche Abteilung
Berliner Strasse 2
3392 Clausthal - Zellerfeld
F.R. of GERMANY

Hans W. Levi
Hahn-Meitner-Institut fur
Kernforschung
Glienicker Strasse 100
1000 Berlin 39
F.R. of GERMANY

Library
Studsvik Energiteknik AB
S-611 01 Nykoping
SWEDEN

Franz Peter Oesterle
Physikalisch-Chemische
Bundesanstalt
Bundesallee 100
D-3300 Braunschweig
F.R. of GERMANY

Tonis Papp
Karnbranslesakerhet
Fack. 10240
Stockholm
SWEDEN

Tjalle Vandergraff
Atomic Energy of Canada Limited
Whiteshell Nuclear Research
Establishment
Pinawa, Manitoba ROE 1LQ
CANADA

No. of
Copies

Seppo Vaori
Technical Research Centre
of Finland
Nuclear Research Laboratory
P.O.B. 169, SF-00181
Helsinki 18
FINLAND

Egbert Schapermeier
Battelle-Institute e.V.
Am Romerhof 35
D-6000 Frankfurt am Main 90
F.R. of GERMANY

Jukka-Pekka Salo
Technical Research Laboratory
P.O.B. 169, SF-00181
Helsinki 18
FINLAND

ONSITE

4 DOE Richland Operations Office

O. J. Elgert
H. E. Ransom
J. J. Schreiber
F. R. Standerfer

2 Rockwell Hanford Operations

R. C. Arnett
R. A. Deju
Rockwell Document Control

No. of
Copies

51 Pacific Northwest Laboratory

G. L. Benson
D. L. Bradley
A. Brandstetter (10)
R. W. Bond
J. S. Burlison
D. B. Cearlock
M. O. Cloninger
C. R. Cole
F. H. Dove
D. W. Dragnich
D. R. Friedrichs
S. R. Gupta
M. A. Harwell
B. W. Howes
J. H. Jarrett
F. E. Kaszeta
M. R. Kreiter
J. S. Light
R. W. Nelson
A. M. Platt
J. R. Raymond (10)
A. E. Reisenauer
R. J. Serne
C. S. Simmons
J. K. Soldat
J. F. Washburn
Technical Information Library
(5)
Publishing Coordination Ro (2)
Water and Land Resources
Department Library (10)

RECIPIENTS OF PNL-2929:

If you wish to be on the distribution list for future model computer code and report revisions, please fill out the form below and return it to Dr. Albin Brandstetter.

REQUEST FOR MODEL REPORT REVISION

Name _____
Title and Affiliation _____
Address _____
Date _____ Signature _____

Please fill out this form and mail to:

Dr. Albin Brandstetter
Project Manager, AEGIS
Battelle, Pacific Northwest Laboratory
P.O. Box 999
Richland, WA 99352
U.S.A.