2F

UNSAT-H, An Unsaturated Soil Water Flow Code for Use at the Hanford Site: Code Documentation

M. J. Fayer
G. W. Gee

October 1985

Prepared for the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Operated for the U.S. Department of Energy
by Battelle Memorial Institute

DISCLAIMER

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

PACIFIC NORTHWEST LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC06-76RLO 1830

Printed in the United States of America
Available from
National Technical Information Service
United States Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22161

NTIS Price Codes
Microfiche A01

Printed Copy

Pages	Price Codes
001-025	A02
026-050	A03
051-075	A04
076-100	A05
101-125	A06
126-150	A07
151-175	A08
176-200	A09
201-225	A010
226-250	A011
251-275	A012
276-300	A013

UNSAT-H, AN UNSATURATED SOIL WATER FLOW
CODE FOR USE AT THE HANFORD SITE:
CODE DOCUMENTATION

M. J. Fayer
G. W. Gee

October 1985

Prepared for
the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Richland, Washington 99352

ABSTRACT

The unsaturated soil moisture flow code, UNSAT-H, which was developed at Pacific Northwest Laboratory for assessing water movement at waste sites on the Hanford site, is documented in this report. This code is used in simulating the water dynamics of arid sites under consideration for waste disposal. The results of an example simulation of constant infiltration show excellent agreement with an analytical solution and another numerical solution, thus providing some verification of the UNSAT-H code. Areas of the code are identified for future work and include runoff, snowmelt, long-term climate and plant models, and parameter measurement.

SUMMARY

The purpose of this report is to document the unsaturated soil moisture flow code, UNSAT-H, which was developed at Pacific Northwest Laboratory for assessing the water dynamics of arid sites under consideration for near-surface waste disposal. This documentation describes a fully implicit, finite-difference method that simulates in one dimension the dynamic processes of infiltration, drainage, redistribution, surface evaporation, and plant water uptake from soil. The soil profile can be homogeneous or layered. The boundary conditions are controlled to reflect actual surface and water-table conditions at a given site.

An example simulation of the infiltration process compares UNSAT-H results with an analytical solution and a numerical solution. The results show excellent agreement for the specific test case, and this agreement provides some verification of the infiltration portion of the UNSAT-H code.

UNSAT-H has several unique features when compared to UNSAT1D and other codes from which UNSAT-H was developed: 1) the code has been modified to describe primarily arid sites (i.e., isothermal vapor flow has been included); 2) additional options for describing soil hydraulic properties have been included; 3) the mass balance error has been improved; and 4) the output has been made more efficient.

Throughout this report, specific areas of UNSAT-H have been identified for future work. One aspect of that work should include the capability to model runoff and snowmelt processes, especially if sites such as the Arid Lands Ecology Reserve are to be modeled. If long-term recharge rates are to be predicted, a stochastic climate model and plant growth model specific for the Hanford site ought to be developed and incorporated into UNSAT-H. Finally, key parameters such as unsaturated hydraulic conductivity require more accurate measurement than they have received to date.

CONTENTS

ABSTRACT	iii
SUMMARY	v
1.0 INTRODUCTION	1.1
2.0 UNSAT-H MODEL OVERVIEW	2.1
2.1 CLIMATIC INPUTS	2.2
2.2 SOIL HYDRAULIC PROPERTIES	2.6
2.3 PLANTS	2.9
3.0 UNSAT-H CODE OPERATION	3.1
3.1 SYSTEM	3.1
3.2 INPUT PROGRAM DATAINH	3.2
3.3 UNSAT-H	3.3
3.4 OUTPUT PROGRAM DATAOUT	3.3
4.0 EXAMPLE SIMULATION	4.1
4.1 PROBLEM DESCRIPTION	4.1
4.2 RESULTS	4.2
5.0 FUTURE NEEDS	5.1
5.1 TEMPORAL VARIABILITY	5.1
5.2 KEY MEASUREMENT PARAMETERS	5.2
5.3 SUMMARY OF NEEDS	5.2
REFERENCES	R.1
APPENDIX A - UNSAT-H INPUT MANUAL: A COMPUTER PROGRAM FOR ONE-DIMENSIONAL UNSATURATED FLOW IN POROUS MEDIA	A.1
APPENDIX B - INPUT AND OUTPUT LISTINGS FOR THE INFILTRATION TEST CASE FOR YOLO LIGHT CLAY.....	B.1
APPENDIX C - UNSAT-H SOURCE COMPUTER CODE	C.1
APPENDIX D - UNSAT-H GLOSSARY	D.1

FIGURES

2.1	Relationship Between Transpiration, Net Radiation, and Day of the Year.....	2.11
2.2	Root Mass at the End of the 1974 Growing Season in Cheatgrass and Bluebunch Wheatgrass Communities.....	2.13
2.3	Actual SINK Term as a Function of the Water Content.....	2.15
3.1	Listing of Sample Command File for Running UNSAT-H in the Batch Mode.....	3.1
4.1	Infiltration Rate and Cumulative Infiltration Versus Time as Determined Using Philip's Solution, the Numerical Code of Haverkamp et al., and UNSAT-H.....	4.3
B.1	Listing of the Sample Input File for the Yolo Light Clay Simulation.....	B.1
B.2	Listing of a Sample Output File for the Yolo Light Clay Simulation.....	B.2

TABLES

2.1	Cheatgrass Root Biomass Data, Root Length, and Root Density Function.....	2.14
4.1	Coefficients and Parameters Used in the Infiltration Simulations.....	4.1

1.0 INTRODUCTION

The prediction of unsaturated water movement at arid sites is difficult. Limited water inputs resulting from uncontrolled sporadic rainfall events, sparse plants subjected to frequent water stress conditions, and highly variable soils with wide ranges of hydraulic properties tend to make simple water flow predictions difficult at best. Although standard deterministic water flow models that use some empirical relationships may be appropriate under certain well-defined conditions (e.g., irrigated agriculture), they require important modifications to simulate arid site water movement.

In this report, we document the major features of the numerical code, UNSAT-H, which has been developed specifically to describe unsaturated flow under arid site conditions. The code is a physically based, deterministic model of soil water movement. Processes such as infiltration, drainage, redistribution, and evapotranspiration are simulated using a pressure head formulation of Richard's equation for soil water movement and a modification of the Penman equation [for potential evapotranspiration (PET)] to account for actual evapotranspiration under dry site conditions.

The report is divided into four sections. Section 2.0 provides an overview of the UNSAT-H model, including a description of how climate, soil hydraulic properties, and plants are integrated into the model. Section 3.0 describes code operation, including the required computer system operation, input and output requirements, and basic features of the computer program UNSAT-H. Section 4.0 provides an example simulation of infiltration and compares the UNSAT-H results with both an analytical solution and another numerical solution. Section 5.0 discusses some of the major issues that will require additional study in order to use UNSAT-H for long-term predictions of water balance at arid sites. An input manual, listings of the example problem's input and output, source code listing, and a glossary are provided in the appendices.

2.0 UNSAT-H MODEL OVERVIEW

The original model on which UNSAT-H is based was developed by Gupta et al. (1978) to model field soil water movement with concurrent soil water extraction by crops. The Gupta et al. code contains a variety of options for plant root growth and water extraction, partitioning PET into transpiration and evaporation, lower boundary conditions, etc., as well as sample runs.

A version of the Gupta et al. code, which was documented for the Electric Power Research Institute (EPRI) by Bond et al. (1984), is called UNSAT1D. It contains most of the original options plus a number of subsidiary programs for fitting polynomials to soil property data and for plotting output data.

UNSAT-H is a departure from its predecessors for several reasons. First, it is designed specifically for arid sites. Most agricultural options such as variable ponding have been removed, and isothermal vapor flow has been included. Because of the dry conditions at the Hanford site, the conductivity of water in liquid form is quite low. In such situations, vapor flow can be a significant component of and may actually dominate the overall flux of water. Secondly, subroutines have been added, which allow for soil hydraulic properties to be described by functions [e.g., those developed by Haverkamp et al. (1977) and Campbell (1974)] other than polynomials.

Thirdly, the new code gives better control of the mass balance error. One of the purposes of the UNSAT-H code will be to simulate recharge with a precision of less than 1 mm/yr. The final new feature is output control. The UNSAT1D code creates several different output files, which consumes computer time. The files take up significant amounts of storage space, thus increasing costs and making it difficult to use the code when storage space is limited. In UNSAT-H, all primary data are output to one binary file, and a post-run processing program (DATAOUT) is used to read the binary output and to analyze the simulation results.

Three variables that control the hydrology of arid sites are the climate, soil properties, and plant behavior. The UNSAT-H code has been designed to use as much of the available data sets at the Hanford site as possible.

2.1 CLIMATIC INPUTS

At the Hanford Meteorology Station (HMS) on the 200 Area plateau, a number of climatic variables are measured on an hourly basis, including rainfall, snowfall, solar radiation, wind speed, air temperature, dew-point temperature, and cloud cover. Most of these data have been summarized by Stone et al. (1983). Because this data set is the most complete climate record available, we have used it to describe the Hanford site climate for simulations related to water movement at waste disposal and protective barrier sites. In addition, climate records from adjacent locations such as the Arid Lands Ecology (ALE) Reserve are also available and can be used. Thorp and Hinds (1977) have described the microclimates of the ALE reserve and related them to the larger record available at the HMS.

Precipitation: The Hanford site 200 Area receives 16.0 cm of precipitation in an average year. Sixty-one percent (9.7 cm) of that total falls during the winter months of October-February. Only 15% (2.4 cm) falls during the summer months of June-August. This seasonal imbalance in the distribution of precipitation plays a pivotal role in the water dynamics of the Hanford site. Gee and Simmons (1979) have shown that winter precipitation contributes to water storage and drainage whereas summer precipitation does not because a large evapotranspiration potential exists during the summer.

On a daily basis, there is no known pattern to rainfall distribution, however, precipitation that falls during the day is more likely to be evaporated than precipitation falling at night. Because of the extensive precipitation data base for the Hanford site 200 Area, hourly precipitation values can be used as input to any modeling scheme. Thus, seasonal and daily precipitation distributions can be preserved.

In the UNSAT-H code, precipitation is entered on an hourly basis. For example, whatever precipitation falls within an hour's time will be totaled and considered to have fallen evenly over the hour. Such a scheme does not preserve actual rainfall intensities; however, in terms of runoff potential, actual intensities are not of great concern at the Hanford site except on steep slopes or when the soil is frozen. Most surface soils at the Hanford site have saturated conductivities (>2.5 cm/h) far in excess of rainfall intensities

normally observed in the waste site areas. In the 200 Area, a storm lasting just 20 min with an intensity of 2.5 cm/h is likely to occur only once in 1000 years (Stone et al. 1983). At sites like the ALE reserve, however, the runoff potential is much greater. Therefore, future improvements to the UNSAT-H code should include an overland flow component.

Snowfall: Snowfall accounts for 25% (4.0 cm) of the total yearly precipitation and 38% (2.3 cm) of the December-February precipitation at the HMS (Stone et al. 1983). To date, most unsaturated modeling of the Hanford site has proceeded on the assumption that snowfall can be treated as equivalent to rainfall. This assumption may be accurate for most situations because snow usually melts fairly quickly after falling (the mean maximum air temperature during the winter is above freezing). However, the ground is covered with an inch or more of snow during a significant number of days. The seasonal average is 19 days, but it can range from 0 to 54 days. This phenomenon is important for two reasons: first, snow on the surface represents a water storage reservoir other than soil. In this position, the snow is subject to melting and direct evaporation (sublimation) or runoff (or runoff for some sites). Thus, little or none of the water from the original snowfall may enter the soil at the site in question. Secondly, snow has a high and variable albedo (0.45-0.8). Most of the incoming solar radiation, the driving force for evaporation, is therefore reflected by the snow surface. This reduction in net shortwave radiation lowers the amount of evaporation that can take place.

Simulation of infiltration and evaporation processes during and after snow events should account for these physical processes. For these reasons, a snow-melt algorithm, such as that of Donigan and Crawford (1976), which is based on degree days, should be added to the UNSAT-H code.

Potential Evapotranspiration: Water vapor is lost from the soil to the atmosphere either by evaporation from the soil surface or by plant transpiration of water from within the soil. This flux of water vapor, usually called evapotranspiration, depends on the vapor density gradient and the resistance to vapor transport. This resistance is offered by the surface boundary layer and plant and soil factors. If the soil is sufficiently wet such that the resistance terms are at a minimum, then evapotranspiration from the plant/soil

system will be at a maximum (i.e., at the potential rate). As the soil dries or the plants close their stomata, the resistance to vapor transport increases, and evapotranspiration falls below the potential rate.

Many calculations of PET have been done at the Hanford site. Wallace (1978) calculated PET for the years 1945-1970 from average monthly weather data from the Hanford Meteorology Station. He used three separate equations (Thornthwaite, Penman, and Morton) and found PET estimates to be quite variable (75.0, 80.0, and 139.0 cm/yr, respectively). Leonhart (1979) used the Thornthwaite equation and a 30-year temperature record from Kennewick, Washington, (just south of the Hanford site) and calculated PET as 73.9 cm/yr. Leonhart also indicated that the annual pan evaporation rate, measured in agricultural areas surrounding the Hanford site, was 152.4 cm/yr. Leonhart states that the pan rate should be reduced by a pan coefficient of 0.7 to get a PET value of 106.6 cm/yr. Gee and Simmons (1979) calculated PET for 1947-1948. Using the Blaney-Criddle and Penman equations (both found in Doorenbos and Pruitt 1977), Gee and Simmons calculated PET values of 147.0 and 181.0 cm (for 1947) and 131.0 and 159.0 cm (for 1948), respectively.

PET estimates for the Hanford site vary widely (73.9 to 181.0 cm/yr). In the case of the Penman equation, several components of the equation are determined empirically. They include the net longwave radiation, conversion of wind speed measured at 15.24 m to 2.0 m, and the wind speed function. Wallace (1978) used the original version of the Penman equation (calibrated to the English climate) and calculated PET to be 80 cm/yr for the Hanford site. When the same data were used with the Penman equation according to Doorenbos and Pruitt (1977), PET was calculated to be 167.1 cm/yr. The difference in the two PET calculations, despite using the "Penman" equation, is caused by the use of different empirical formulations within the equation. Quite clearly, if PET estimates vary by a factor of 2.0, the empirical portions of the equation should be calibrated specifically for Hanford site conditions. Based on the data available at the Hanford site, the empirical formulations in Doorenbos and Pruitt appear to be related to the Hanford site more than are those used in the Penman-type equation used by Wallace (1978).

For water balance modeling at the Hanford site, PET must be estimated daily. The most fundamental approach is a combination equation of the type proposed by Penman (1948), where daily values of net solar radiation, air temperature, humidity, and wind run are used to calculate a daily PET value. The equation is

$$LE = \frac{\Delta(Rn-G) + \gamma f(u)(e_a - e_d)}{\Delta + \gamma} \quad (2-1)$$

where LE = evaporation rate, mm/d

Rn = net radiation, mm/d

G = soil heat flux, mm/d

u = daily wind run, km/d

f(u) = wind function, mm/mbd

e = vapor pressure, mb

Δ = slope of the saturated vapor pressure-temperature curve, mb/°K

γ = psychrometric constant, mb/°K

a,d = vapor pressure subscripts indicating saturation and actual values at the measured air temperature.

The transport coefficients for heat and vapor flow were considered equal in Equation (2-1). In addition, we assumed that the vapor pressure at the evaporating surface was the saturated vapor pressure, although this is not always true in an arid climate. To account for a surface vapor pressure for conditions less than saturated would require plant resistance information that is not available. (See Monteith 1973 for a complete discussion.)

Regardless of which method is used to calculate PET, the UNSAT-H code uses the calculated daily PET to estimate actual evapotranspiration (AET). If the plants and soil combined can deliver water to maintain the PET rate, then AET equals PET. If they cannot supply water at that rate, AET falls below PET. The UNSAT-H model controls evapotranspiration in two ways. During evaporation, when the pressure head at the surface reaches some maximum value (chosen a priori), the surface boundary condition reverts from a flux (i.e., the PET rate) to a fixed pressure head (Nimah and Hanks 1973). Actual evaporation then

equals whatever the soil can supply from below. The second control, plants, depends on the water status of the soil. If the soil dries below some pre-determined water content, transpiration is reduced (see Feddes et al. 1978).

2.2 SOIL HYDRAULIC PROPERTIES

The success of any water dynamics modeling scheme depends heavily on the correct characterization of the soil hydraulic properties, specifically moisture retention and hydraulic conductivity. The properties should be characterized by measuring them in the lab or in situ and mathematically representing the measured properties in the computer code. The UNSAT-H code contains three options for describing soil hydraulic properties: polynomials (Bond et al. 1984), Haverkamp functions (Haverkamp et al. 1977), and Campbell functions (Campbell 1974).

The polynomial option allows up to four polynomials of the forms

$$\theta = a + b \log(h) + c \log^2(h) + d \log^3(h) + e \log^4(h)$$

$$\log K = a + b \log(h) + c \log^2(h) + d \log^3(h) + e \log^4(h)$$

to be used to describe each soil property. θ represents the volumetric moisture content, h the soil water pressure head, and K the hydraulic conductivity.

The UNSAT-H convention is to express the soil water pressure potential in units of head (i.e., the height of a column of water corresponding to the given pressure). The pressure potential (or head) of unsaturated soil is negative, and the magnitude of this value is called suction or tension. Because UNSAT-H will be used almost entirely to model unsaturated flow, we have retained the sign convention of the earlier code by Gupta et al. (1978). That is, the sign of the head values in UNSAT-H is reversed: positive head values refer to suction values, and negative head values refer to positive pressures (greater than atmospheric).

Bond et al. (1984) developed a computer program that can be used to fit polynomials to measured soil hydraulic data and ensure that the fit is

continuous at each of the matching points. The two major advantages of this option are that the user can easily fit any data set and can extend the data into the high pressure head range. A disadvantage of the polynomial option is that it consumes significantly more computer time for representing soil properties than the other options.

In the Haverkamp option (Haverkamp et al. 1977), the soil properties are described by equations of the form

$$\theta = \theta_r + \alpha (\theta_s - \theta_r) / (\alpha + |h|^\beta) \quad (2-2)$$

$$K = K_s A / (A + |h|^B) \quad (2-3)$$

where θ_r is the residual water content, θ_s is the saturated water content, and α , β , A , and B are curve-fitting parameters. [In Equation (2-2), the option exists in UNSAT-H to replace the h term with $\ln(h)$]. McKeon et al. (1983) developed two programs that can be used to fit the Haverkamp functions to measured soil hydraulic data. One of the problems with the Haverkamp option is that the specific water capacity ($d\theta/dh$) and K very quickly tend to approach zero as the head increases. Therefore, less certainty exists about the applicability of this option when the soil is very dry.

In the third available soil property option, the Campbell functions (Campbell 1974), the soil properties are fit with equations of the form

$$\theta = \theta_s (h_e/h)^{(1/b)} \quad (2-4)$$

$$K = K_s (h_e/h)^{(2+3/b)} \quad (2-5)$$

where h_e represents the air-entry pressure head and b is a curve-fitting parameter. The Campbell functions yield finite values of $d\theta/dh$ and K at high head values, which is important in modeling dry soils. A problem arises with layered soils, however, because the soil properties are not described below the h_e values. If two adjacent materials with different h_e values should become

nearly saturated, a situation could arise where flow is artificially induced in the wrong direction. Therefore, this option should be avoided for layered systems.

Vapor flow is ordinarily an insignificant fraction of the overall flux of water in moist soil. When the soil is very dry, however, liquid water conductivity values are so low that vapor flow may dominate the overall water flux, even in the absence of a temperature gradient. Therefore, a method (proposed by Simmons and Gee 1981) of modeling isothermal vapor flow was incorporated into UNSAT-H.

Because the system is considered isothermal, vapor density gradients are equivalent to relative humidity gradients. The relative humidity at any point in the soil can be related to the pressure head by

$$RH = \exp[-hMg/RT] \quad (2-6)$$

where RH = relative humidity

exp = exponential function

M = molar mass of air (18.0 g/mole)

g = gravitational constant (980.7 cm/s²)

R = gas constant (8.314 x 10⁷ erg/mole °K)

T = soil temperature (288.5 °K)

Simmons and Gee (1981) represented vapor flow in terms of conductivity, where the total soil conductivity (K_t) was the sum of the liquid (K_l) and vapor (K_v) conductivities. The vapor conductivity was derived as

$$K_v = \frac{\alpha D_a Mg}{p_w RT} p_s (\phi - \theta) RH \quad (2-7)$$

where α = pore tortuosity (0.66)

D_a = diffusion coefficient of water vapor in air (0.24 cm²/s)

p_w = density of water (1.0 g/cm³)

p_s = saturated vapor density at the soil temperature
 (1.73×10^{-5} g/cm³ at $T = 288.5$ °K)
 ϕ = total porosity

The total flux equation was revised such that

$$q = K_t(h) [dh/dz + K_l(h)/K_t(h)] \quad (2-8)$$

Simmons and Gee (1981) performed a calculation in which they showed that the highest potential value of K_v was approximately 7.4×10^{-9} ϕ cm/h. Therefore, isothermal vapor flow could not make a significant contribution to flow until the liquid water conductivity approached 7.4×10^{-9} ϕ cm/h. For many of the sandy soils at the Hanford site, liquid conductivities near the soil surface often fall far below 7.4×10^{-9} ϕ cm/h as the soil dries out. For much of the time in this area of the soil profile, vapor flow is potentially the dominant mode of water transport.

2.3 PLANTS

The plant community on the Hanford site consists of a mixture of sagebrush and cheatgrass, with lesser amounts of bitterbrush, rabbitbrush, bluegrass, and some Russian thistle (Gee and Heller 1985). The cheatgrass and Russian thistle have average rooting depths of 60 - 70 cm in contrast to the brushes, which have average rooting depths of 250 - 300 cm (Fox et al. 1984). Disturbed areas of the Hanford site tend to become colonized by the cheatgrass and Russian thistle. If construction takes place at the Hanford site, much of the resulting disturbed land may be revegetated by the shallow-rooted plants. This is important from a water-balance standpoint because the shallow-rooted plants will not remove as much total water from the soil profile as would the deeper-rooted plants.

Transpiration from a plant community can be modeled by partitioning PET (as calculated with the Penman equation of Doorenbos and Pruitt 1977) into evaporation and transpiration components using a crop coefficient approach (Ritchie 1972; Wright 1982). There are two methods to partition PET in the

UNSAT-H code. In the first method, potential transpiration (PT) is calculated from the leaf area index (LAI) with the equation

$$PT = PET(-0.12+0.70LAI) \text{ for } 0.1 < LAI < 2.7 \quad (2-9)$$

which was developed by Ritchie and Burnett (1971) for cotton and grain sorghum. Ritchie (1972) notes that PET in the above equation was actually net radiation and not PET calculated with the Penman combination equation; however, the discrepancy is of little consequence. Although LAI data are not currently available for the plant communities at the Hanford site, this option was included in anticipation of work planned for the future. That work may also result in a modification of Ritchie's equation to fit the plant communities at the Hanford site.

The second method for partitioning PET uses local cheatgrass data. Hinds (1975) conducted field experiments with cheatgrass growing in small microlysimeters located in a field plot on the ALE site. During April and May 1972, he measured total and net shortwave radiation, soil heat flux, evaporation and transpiration, and calculated net longwave radiation with an empirical equation. Hinds then related transpiration to the total net radiation and computed what might be considered a crop coefficient. The relationship is shown in Figure 2.1, with the shaded portion representing the ratio of transpiration to net radiation over a 2-month period.

The Hinds data contain a number of deficiencies. First, Hinds related transpiration to the net radiation and not to PET. Indications are that this relationship results in a 36% overestimate of transpiration when converting from PET (i.e., the net radiation represents only about 74% of PET). Secondly, his assumption that evaporation is the same from lysimeters with or without plants is probably suspect. Plants withdraw water from the root zone. Hinds determined that 90% of the cheatgrass roots were in the top 10 cm of soil. Thus, the water withdrawal pattern would be different than that in the lysimeter with just evaporation occurring. The rate of evaporation will therefore necessarily be different (even ignoring the effect of the cheatgrass plants on

FIGURE 2.1. Relationship Between Transpiration, Net Radiation (R_n), and Day of the Year (based on cheatgrass data of Hinds 1975). The cross-hatched area represents the period of measurement during 1972. Shoot biomass production during that period was 220 g/m^2 .

vapor transfer from the soil surface). The net result is that the transpiration numbers reported by Hinds are probably underestimates.

Thirdly, over the 2-month period of Hinds' experiment, no attempt was made to maintain adequate water availability. Therefore, it is unclear whether the transpiration/net radiation ratio is a true crop coefficient or merely reflects the reduction in transpiration caused by a lack of soil moisture. No relationship is available between plant water stress and available moisture for ALE or any of the Hanford waste sites.

Finally, the Hinds' cheatgrass data does not cover the entire cheatgrass growing season. According to Klemmedson and Smith (1964), cheatgrass usually germinates in the fall, remains dormant during the winter, resumes growth in early spring, and flowers and dies by early June (either of maturity or lack of soil moisture). In order to use UNSAT-H to simulate the phenology of cheatgrass, the transpiration relationship (cross-hatched area) in Figure 2.1 must

be extended throughout the growing season. The code user can do this by choosing two dates, NSOW and NHRVST. NSOW is the day when the cheatgrass seeds germinate. NHRVST is the day when the cheatgrass plants cease transpiring. Because the exact dates for these two parameters are unknown, they are left as variables for the code user. As seen in Figure 2.1, the transpiration/net radiation ratio increases linearly between NSOW and the first day of maximum transpiration (Julian Day 90), and it decreases linearly from the last day of maximum transpiration (Julian Day 151) to NHRVST. Despite its deficiencies and until more data become available from future experiments now being planned, the Hinds' cheatgrass data serve as the best available data for partitioning PET into transpiration and evaporation.

From year to year, plant biomass production (and thus transpiration) will vary from that observed during the Hinds' experiment because of the climate or possibly nutrient availability. Hinds (1975) measured a shoot biomass production of 220 g/m^2 over the course of his experiment. Assuming a direct relationship between shoot biomass and transpiration, the UNSAT-H code user can alter the transpiration ratio by specifying a value for shoot biomass other than 220. For example, specifying a shoot biomass of 440 g/m^2 will result in doubling the transpiration/net radiation ratio, with the constraint that the value must fall between 0.0 and 1.0.

Once the fraction of PET that is transpiration is determined, the transpiration demand is applied to the root zone using a volumetric sink term in the flow equation. The sink term of each node is calculated by distributing the transpiration demand over the root zone as a function of the rooting density and modifying the result based on the soil moisture status of each node. Determining the water uptake pattern in this manner is sometimes referred to as the "macroscopic" approach [i.e., treating the roots as a whole rather than individually (Feddes et al. 1978)].

Root density data from the Hanford site are limited to end-of-the-growing-season distributions. Cline et al. (1977) have measured below-ground biomass in two plant communities at ALE, but they did not differentiate between living and dead tissue. Figure 2.2 shows the root distribution for an annual grass community (mostly cheatgrass) located at the 1000-ft elevation and a perennial

FIGURE 2.2. Root Mass (ash free) at the End of the 1974 Growing Season in Cheatgrass and Bluebunch Wheatgrass Communities. (Adapted from Cline et al. 1977)

grass community (mostly bluebunch wheatgrass) located at the 1200-ft elevation. Note that the annual cheatgrass roots were most dense in the top 30 cm of soil. The cheatgrass root biomass data of Figure 2.2 were normalized (Table 2.1). It was assumed that the normalized root biomass could be directly related to root-length density. The root length density could then be related to the depth (Z) below the surface by

$$RHO = a \exp(-bZ) + c \quad (2-10)$$

where a , b , and c are coefficients that optimize the fit ($a = 1.163$, $b = 0.129$, and $c = 0.020$). The relative units of RHO are cm roots/cm soil. To calculate

TABLE 2.1. Cheatgrass Root Biomass Data, Root Length, and Root Density Function

Depth Interval, cm	Root Biomass, g/m ²	Normalized Biomass	RHO cm root/cm soil	RDF, 1/cm	
				Roots Penetrate to 80 cm	Roots Penetrate to 20 cm
0-10	499.4	0.630	0.630	0.0629	0.0770
10-20	145.5	0.184	0.188	0.0188	0.0230
20-30	51.1	0.065	0.066	0.0066	---
30-40	31.5	0.040	0.033	0.0033	---
40-50	25.6	0.032	0.024	0.0024	---
50-60	15.7	0.020	0.021	0.0021	---
60-70	11.8	0.015	0.020	0.0020	---
70-80	11.8	0.015	0.020	0.0020	---
Total	792.4				

the root density function (RDF), the values of RHO are multiplied by their respective depth intervals to obtain the total root length. Each RHO value is then divided by the total root length to obtain the RDF value for each depth. Table 2.1 contains the RDF values calculated for two depths of root penetration.

Numerous schemes for modeling plant water uptake exist (Molz 1981; Alaerts et al. 1985), some of which require knowledge of root and stem water potentials, soil water pressure heads, and conductivities in the vicinity of the roots. Such approaches are appealing because they are physically based, yet they lose their appeal when the data requirements are realized. With the limited amount of plant data available at the Hanford site, using a simplified uptake function of the type described by Feddes et al. (1978) is best. Such a function relates the transpiration rate to the moisture status in the root zone. An example function is shown in Figure 2.3. When the soil moisture content is greater than θ_n , the soil is so wet that it is anaerobic, and the plant ceases transpiration. When the soil moisture content is between θ_d and θ_n , the rate of withdrawal is set to the potential rate. If the soil moisture content is between θ_w and θ_d , the rate of withdrawal is reduced by the appropriate

FIGURE 2.3. Actual SINK Term as a Function of the Water Content (the SINK term represents the rate of water withdrawal)

factor (see Figure 2.3). When the soil moisture content is below θ_w , the roots are considered to have stopped withdrawing water. Currently, choices for moisture content values of θ_w , θ_d , and θ_n are left to the code user because data is lacking.

In contrast to the description in Feddes et al. (1978), this water-uptake function is applied to each node rather than the profile as a whole. Thus, the code sequence is to take the PT rate, partition it among the different nodes (which contain roots) using the root density distribution, then modify the rate of withdrawal (the sink term) from each node based on the moisture status of that node. The actual transpiration rate is the sum of the sink terms multiplied by their respective node thicknesses.

FIGURE 1. Actual (dashed line) and predicted (solid line) concentration vs. time for the system.

Figure 1 shows the actual (dashed line) and predicted (solid line) concentration vs. time for the system. The system is a first-order system with a time constant of 10 minutes. The actual concentration is shown as a dashed line and the predicted concentration is shown as a solid line. The predicted concentration is a smooth curve that follows the general trend of the actual concentration.

In contrast to the description in Figure 1, the actual concentration is shown as a dashed line and the predicted concentration is shown as a solid line. The predicted concentration is a smooth curve that follows the general trend of the actual concentration. The system is a first-order system with a time constant of 10 minutes. The actual concentration is shown as a dashed line and the predicted concentration is shown as a solid line. The predicted concentration is a smooth curve that follows the general trend of the actual concentration.

3.0 UNSAT-H CODE OPERATION

UNSAT-H is coded in VAX FORTRAN Version 4.0,^(a) which is based on American National Standard FORTRAN-77 (ANSI X3.9-1978). Although extensions to FORTRAN-77 are available in VAX FORTRAN, their use is limited to keep UNSAT-H close to a standard FORTRAN-77 form.

3.1 SYSTEM

The UNSAT-H code runs under the VAX/VMS Version 4.0 Operating System.^(b) The operating system provides access to the disk storage devices required by UNSAT-H for its input/output options. The UNSAT-H code is designed to be run in the batch mode by submitting to the batch queue a command file that instructs the system on file location and provides answers to program queries. Figure 3.1 is a listing of a typical command file. (This file was used to run the example simulation described later.)

On the first line of the command file, the default directory is established. Unless instructed otherwise, the system assumes that the input file is located in the default directory and that the output file is to go there. The next two lines are optional but strongly suggested. They cause copies of the command and input files to be inserted into the *.LOG file of this simulation.

```
$ SET DEFAULT DISK1:[HCODE.UNSATH]
$ TYPE SAMPLE.COM
$ TYPE SAMPLE.INP
$ RUN UNSATH
SAMPLE
$ COST
```

FIGURE 3.1. Listing of Sample Command File (SAMPLE.COM) for Running UNSAT-H in the Batch Mode

-
- (a) "Programming in VAX FORTRAN." 1984. AA-D034D-TE. Digital Equipment Corporation, Maynard, Massachusetts.
(b) "VAX/VMS DCL Dictionary." 1984. Digital Equipment Corporation, Maynard, Massachusetts.

These copies serve as a record of the files used. The RUN UNSATH command on the fourth line actually starts the code running. The code does not necessarily have to be in the default directory (in this case, it is). If not, the RUN UNSATH command should be altered to indicate where the code is located. For example, the command RUN DISK0:[TRIAL]UNSATH indicates that the code is on DISK0 in the main directory TRIAL. The next line of the command file serves as an answer to an UNSAT-H query for the name of the binary input file (*.BIN, with BIN indicating binary form). The lack of a dollar sign at the start of this line indicates that the line is not to be interpreted as a system command but as a response to a program query. The last line causes a cost summary of the simulation run to be written to the *.LOG file.

3.2 INPUT PROGRAM DATAINH

The data necessary to run the UNSAT-H code are first assembled in a file with the filename extension *.INP. The data format for this file is detailed in the UNSAT-H Input Manual, which can be found in Appendix A. A listing of an example input file can be found in Appendix B.

To facilitate operating the UNSAT-H code, the data file is first read by a separate program called DATAINH (similar to DATAIN for the UNSAT1D code of Bond et al. 1984). The DATAINH program reads the data, checks for errors, performs calculations (such as partitioning PET), and then writes the results in binary form to a file with the same name as the input file but with the extension changed to *.BIN. When UNSAT-H is run, it looks for the specified file with the *.BIN extension and reads it. Having DATAINH preprocess the data limits the chance that UNSAT-H will fail to run because of input errors. Thus jobs submitted to run overnight have a greater likelihood of running successfully given that the input data have already been checked by DATAINH.

The DATAINH program has several output options for checking or recording the input data. For all options, the output can be sent to either the screen or a separate file that will be given the same name as the input file but with a *.LIS extension. All error messages are sent to the screen. In addition, if output is going to a file, the error messages will also go to the file.

3.3 UNSAT-H

The UNSAT-H code is designed to run in the batch mode. As such, most of the code operator's work is in pre- and post-processing (Sections 3.2 and 3.4). To run the code in the batch mode, the appropriate command file is submitted to the batch queue. Assuming that the command file SAMPLE.COM (see Figure 3.1) is in the current directory, typing the command

```
SUBMIT/NOPRINTER/LOG=[] SAMPLE
```

will send the command file to the SYSS\$BATCH queue to be processed. Batch jobs normally generate a *.LOG file that contains program output destined for the SYS\$OUTPUT logical unit (on the VAX, this would be the screen or logical unit 6), and a summary of the run characteristics (e.g., total time spent in the queue and computer run time). The *.LOG file is automatically printed out once the job has finished. The NOPRINTER option causes the *.LOG file to remain in the user's main directory with the command file name and a *.LOG extension. The LOG=[] option causes the *.LOG file to go to the default directory that is specified in the command file (see Figure 3.1).

As set up in Figure 3.1, the UNSAT-H code will generate two files: the binary output file (*.RES) and the program *.LOG file. The binary output file will contain a summary of the end-of-day (or end-of-hour) values of pressure head, water content, fluxes, sinks (plant water extraction), mass balance error, time steps, and run time. The *.LOG file will contain listings of the command and input files (if the appropriate commands are given in the command file); any error messages generated during program execution; and a summary of the computer time used, run time, and total run cost.

3.4 OUTPUT PROGRAM DATAOUT

The UNSAT-H code generates a binary output file (*.RES) that cannot be directly viewed. Instead, the *.RES file must first be read and converted to ASCII format. Program DATAOUT performs that function as well as several other output options:

1. Reinitialize. The program goes to the end of the *.RES file and reads the final pressure head values for all nodes. It then generates an output file (called TOSS.OUT) containing the head values in a format ready for inclusion in a *.INP file. The TOSS.OUT file can thus serve as the initial condition for simulating the following year. This option performs the same function as the program REINITIAL does for UNSAT1D.
2. Scan through the data. With this option, the screen shows the user a summary of the initial conditions, any day of the simulation, or the final simulation summary. No hardcopy listings are created here.
3. Hardcopy Listing. Choosing this option causes an *.OUT file to be created. There are two choices for what goes into the file. The first choice yields a summary of the first and last simulation days and the yearly summary. The second choice yields a summary of every day as well as the yearly summary. The *.OUT file generated by the second choice is very similar to the *.OUT file created while UNSAT1D is running.
4. Create data lists. Data files for plotting purposes can be generated with this option.

4.0 EXAMPLE SIMULATION

Haverkamp et al. (1977) performed a number of infiltration simulations using different models of the nonlinear infiltration equation. One of those models, the head-based implicit model [No. 4 in the Haverkamp paper], is the same as that used in UNSAT-H. In order to verify the infiltration component of UNSAT-H and at the same time provide an example of how UNSAT-H works, the Haverkamp et al. (1977) infiltration problem was simulated.

4.1 PROBLEM DESCRIPTION

The two soil types simulated were a sand and a clay (Yolo light clay), both described by Haverkamp et al. (1977). In UNSAT-H, the sand and clay were described using KOPT = 2 and 4, respectively, which indicates the type of soil property descriptions. The various coefficients and input parameters used are listed in Table 4.1.

The total soil profile depths were 90 and 250 cm for the sand and clay, respectively. Depth increments were 1.0 cm. The bottom boundary was treated as impervious. For the upper boundary, the head of the surface node was held constant at 20.73 cm for the sand and 1.0 cm for the clay. The minimum and

TABLE 4.1. Coefficients and Parameters Used in the Infiltration Simulations

<u>Haverkamp Symbols</u>	<u>Sand</u>	<u>Clay</u>	<u>UNSAT-H Symbols</u>
θ_s	0.287	0.495	THET
θ_r	0.075	0.124	THTR
α	1.611×10^6	739	A
β	3.96	4.00	B
---	1.0	1.0	AIRINT
K_s (cm/h)	34	4.428×10^{-2}	SK
A	1.175×10^6	124.6	ALPHA
B	4.74	1.77	BETA
---	1.0	1.0	AIRINK

maximum time steps were 40 and 500 s, respectively, the same used by Haverkamp et al. (1977). The complete input files are located in Appendix B along with samples of the output generated using DATAOUT.

4.2 RESULTS

Figure 4.1 is a reproduction of the Haverkamp et al. (1977) results (their Figure 6). Also plotted is the quasianalytical series solution of Philip (1969). Note that the series solution is only valid for times less than 600 h for the clay and 0.29 h for the sand. The UNSAT-H results have been added to Figure 4.1 as dotted lines. For both soils, the agreement between UNSAT-H, the Haverkamp et al. results, and Philips solutions is excellent. This agreement indicates that the infiltration component of UNSAT-H is performing satisfactorily for this type of problem. Currently, Hanford site work being planned for the next several years will provide opportunities to more fully test all components of the model.

FIGURE 4.1. Infiltration Rate and Cumulative Infiltration Versus Time as Determined Using Philip's Solution (symbols) (Philip 1969), the Numerical Code of Haverkamp et al. (1977) (solid line), and UNSAT-H (dashed line)

FIGURE 1. The effect of temperature on the rate of reaction of benzene with chlorine and bromine. The data were obtained from the work of R. W. B. Lewis and R. D. Burkhart, *Trans. Faraday Soc.*, 40, 405 (1944).

5.0 FUTURE NEEDS

UNSAT-H is limited to the current knowledge of a number of processes and by the availability of data that describe those processes. The UNSAT-H code will be upgraded to reflect new knowledge and data as they become available.

In our modeling effort, two areas of the UNSAT-H model deserve priority attention. First, if the code is to be used to model long-term recharge, the UNSAT-H model should consider long-term climate changes as well as plant community responses. Secondly, a number of parameters and processes require either more accurate measurement than they currently receive (e.g., unsaturated hydraulic conductivity) and/or a basic research effort geared toward understanding the process on a more fundamental level (e.g., uptake of water by plant roots).

5.1 TEMPORAL VARIABILITY

Long-term recharge rates will be addressed using the UNSAT-H code. One approach to modeling long-term recharge is to use the average annual Hanford site rainfall distribution and amount for all simulated years. Although this approach preserves the average distribution and amount, it ignores the degree of change in distribution and amount from year to year. Such vagaries may contribute to drainage in some years and none in others, whereas systems subjected to only the average rainfall pattern may not show this drainage pattern or the total amount. Using a stochastic climate input model similar to that proposed by Richardson (1981) would allow for yearly variations in rainfall distribution and totals consistent with the record at the HMS, while at the same time preserving the overall averages.

Related to the year-to-year variability in climate are the differences in plant community composition and growth behavior. Currently, the entire plant component of UNSAT-H is based on data from two months of cheatgrass growth. This data in no way addresses year-to-year differences in plant growth or long-term changes in plant composition as related to climate changes or other as yet undetermined parameters (e.g., available nitrogen). What is needed is a plant

growth model that ties in plant response at the Hanford site to water availability, weather, and possibly nutrient status. Work is currently being planned to address these concerns. These data should provide a good basis for a plant model representative of Hanford site conditions. However, this long-range goal will only be achieved after several years of carefully planned experiments designed to obtain information about plant phenology and water relations under arid site conditions.

5.2 KEY MEASUREMENT PARAMETERS

For any modeling effort to be successful, the measured input parameters must accurately portray the real system. The soil hydraulic properties are especially critical because they control how much water is evaporated or drained (assuming storage is constant over the long term). It is recognized that the natural landscape at the Hanford site is quite variable; at the present time, however, UNSAT-H is not intended to address soil property heterogeneities. Much of the work for which the UNSAT-H code will be used is for disturbed, homogenized systems (e.g., barriers, lysimeters).

Another key parameter is the uptake of water by plants. Water uptake is heavily dependent on root distribution yet little root data are available at the Hanford site. Of the data available, it is unknown how the root distribution changes over a growing season or how much of the root biomass is living root tissue that actively engages in water uptake. Studies of plant water uptake and root distribution, related to both individual species and mixed communities, should be conducted over the next several years to complement the evapotranspiration studies that are being planned.

5.3 SUMMARY OF NEEDS

Arid sites are attractive for waste disposal because they receive minimal amounts of precipitation that could transport the waste from the site. The UNSAT-H code is intended to help quantify with some precision the potential for waste leaching at these sites. UNSAT-H accounts for the major hydrologic processes, including rainfall, evaporation, soil moisture redistribution, and

transpiration. Throughout this document, however, we have identified a number of areas of the UNSAT-H code and the modeling effort in general that deserve attention. They include the following:

- runoff
- snowmelt
- stochastic climate input for long-term recharge predictions
- plant growth model (including root growth)
- accurate measurements of key parameters.

These topics should be the subject of future research efforts, and the data thus generated would serve as the basis for model verification and validation.

Addressing each of the above areas of concern will be a major step in developing UNSAT-H into a dependable code for the prediction of water dynamics at arid sites.

REFERENCES

- Alaerts, M., M. Badji and J. Feyen. 1985. "Comparing the Performance of Root-Water-Uptake Models." Soil Sci. 139(4):289-296.
- Bond, F. W., C. R. Cole and P. J. Gutknecht. 1984. "Unsaturated Groundwater Flow Model (UNSAT1D) Computer Code Manual." CS-2434-CCM, Electric Power Research Institute, Palo Alto, California.
- Campbell, G. S. 1974. "A Simple Method for Determining Unsaturated Conductivity from Moisture Retention Data." Soil Sci. 177:311-314.
- Cline, J. F., D. W. Uresk and W. H. Rickard. 1977. "Comparison of Soil Water Used by a Sagebrush-Bunchgrass and a Cheatgrass Community." J. Range Manage. 30:199-201.
- Donigian, A. S., and N. H. Crawford. 1976. Modeling Pesticides and Nutrients on Agricultural Lands. EPA-600/2-76-043, U. S. Environmental Protection Agency, Athens, Georgia.
- Doorenbos, J., and W. O. Pruitt. 1977. "Guidelines for Predicting Crop Water Requirements." FAO Irrigation Paper No. 24, 2nd ed., Food and Agricultural Organization of the United Nations, Rome, Italy.
- Feddes, R. A., P. J. Kowalik and H. Zaradny. 1978. Simulation of Field Water Use and Crop Yield. John Wiley and Sons, Inc., New York.
- Fox, T. S., G. D. Tierney and J. M. Williams. 1984. Rooting Depths of Plants on Low-Level Waste Disposal Sites. LA-10253-MS, Los Alamos National Laboratory, Los Alamos, New Mexico.
- Gee, G. W., and P. R. Heller. 1985. Unsaturated Water Flow at the Hanford Site: A Review of Literature and Annotated Bibliography. PNL-5428, Pacific Northwest Laboratory, Richland, Washington.
- Gee, G. W., and C. S. Simmons. 1979. Characterization of the Hanford 300 Area Burial Grounds. Task III - Fluid Transport and Modeling. PNL-2921, Pacific Northwest Laboratory, Richland, Washington.
- Gupta, S. K., K. K. Tanji, D. R. Nielsen, J. W. Biggar, C. S. Simmons and J. L. MacIntyre. 1978. Field Simulation of Soil-Water Movement with Crop Water Extraction. Water Science and Engineering Paper No. 4013. Department of Land, Air, and Water Resource, University of California, Davis.
- Haverkamp, R., M. Vauclin, J. Touma, P. J. Wierenga and G. Vachaud. 1977. "A Comparison of Numerical Simulation Models for One-Dimensional Infiltration." Soil Sci. Soc. Am. J. 41:285-294.

- Hinds, W. T. 1975. "Energy and Carbon Balances in Cheatgrass: An Essay in Autecology." Ecological Monogr. 45:367-388.
- Klemmedson, J. O. and J. G. Smith. 1964. "Cheatgrass (*Bromus Tectorum* L.)." The Botanical Review 30:226-262.
- Leonhart, L. S. 1979. Surface Hydrologic Investigations of the Columbia Plateau Region, Washington. RHO-BWI-ST-6, Rockwell Hanford Operations, Richland, Washington.
- McKeon, T. J., S. W. Tyler, D. W. Mayer and A. E. Reisenauer. 1983. TRUST-II Utility Package: Partially Saturated Soil Characterization, Grid Generation, and Advective Transport Analysis. NUREG/CR-3443, U.S. Nuclear Regulatory Commission, Washington, D.C.
- Molz, F. J. 1981. "Models of Water Transport in the Soil-Plant System: A Review." Water Resour. Res. 17(5):1245-1260.
- Monteith, J. L. 1973. Principles of Environmental Physics. American Elsevier Publishing Co., Inc., New York.
- Nimah, M. N., and R. J. Hanks. 1973. "Model for Estimating Soil Water, Plant, and Atmospheric Interrelations: I. Description and Sensitivity." Soil Sci. Soc. Amer. Proc. 37:522-527.
- Penman, H. L. 1948. "Natural Evaporation from Open Water, Bare Soil, and Grass." Proc. Roy. Soc. A. 193:120-146.
- Philip, J. R. 1969. "Theory of Infiltration." Adv. Hydrosoci. 5:215-296.
- Richardson, C. W. 1981. "Stochastic Simulation of Daily Precipitation, Temperature, and Solar Radiation." Water Resour. Res. 17(1):182-190.
- Ritchie, J. T., and E. Burnett. 1971. "Dryland Evaporative Flux in a Subhumid Climate, 2. Plant Influences." Agron. J. 63:56-62.
- Ritchie, J. T. 1972. "Model for Predicting Evaporation from a Row Crop with Incomplete Cover." Water Resour. Res. 8(5):1204-1212.
- Simmons, C. S., and G. W. Gee. 1981. Simulation of Water Flow and Retention in Earthen Cover Materials Overlying Uranium Mill Tailings. PNL-3877, Pacific Northwest Laboratory, Richland, Washington.
- Stone, W. A., J. M. Thorp, O. P. Gifford and D. J. Hoitink. 1983. Climatological Summary for the Hanford Area. PNL-4622, Pacific Northwest Laboratory, Richland, Washington.
- Thorp, J. M. and W. T. Hinds. 1977. Microclimates of the Arid Lands Ecology Reserve 1968-1975. BNWL-SA-6231, Pacific Northwest Laboratory, Richland, Washington.

Wallace, R. W. 1978. A Comparison of Evapotranspiration Estimates Using DOE Hanford Climatological Data. PNL-2698, Pacific Northwest Laboratory, Richland, Washington.

Wright, J. L. 1982. "New Evapotranspiration Crop Coefficients." TASAE 108(IR2):57-74.

APPENDIX A

UNSAT-H INPUT MANUAL: A COMPUTER PROGRAM FOR ONE-DIMENSIONAL
UNSATURATED FLOW IN POROUS MEDIA

APPENDIX A

UNSAT-H INPUT MANUAL: A COMPUTER PROGRAM FOR ONE-DIMENSIONAL UNSATURATED FLOW IN POROUS MEDIA

ABSTRACT

UNSAT-H is a computer code for simulating unsaturated soil water flow at arid sites. It is based on a computer code that was developed by Gupta et al. (1978) to model soil water movement with concurrent crop water extraction. A version of the Gupta et al. (1978) code, entitled UNSAT1D, was documented by Bond et al. (1984).

Several significant changes to the original code make UNSAT-H unique:

- § isothermal vapor flow based on an "effective" conductivity
- § soil hydraulic property description by either the Campbell or Haverkamp functions (Campbell 1974; Haverkamp et al. 1977)
- § plant parameters specific to arid sites
- § mass balance controls
- § better output control.

INPUT MANUAL

Input for the UNSAT-H code must be in binary form and have the file extension *.BIN. To create the *.BIN file,

1. assemble the data into a *.INP file with the format outlined in this user's manual
2. run program DATAINH, which will read the *.INP file, process the data, create a *.BIN file, and write the processed data to the *.BIN file.

The *.BIN file serves as the input file to UNSAT-H.

The logical unit assignments are

<u>Unit</u>	<u>Variable</u>	<u>Description</u>
1	LUI	Binary input file unit (input to UNSAT-H)
2	LUB	Binary output file unit from DATAIN-H; Binary output file unit from UNSAT-H
5	LUR	Read (interactive input unit)
6	LUS	Screen (interactive)

The UNSAT-H convention is to express the soil water pressure potential in units of head (i.e., the height of a column of water corresponding to the given pressure). The pressure potential (or head) of unsaturated soil is negative, and the magnitude of this value is called suction or tension. Because UNSAT-H will be used almost entirely to model unsaturated flow, we have retained the sign convention of the earlier code by Gupta et al. (1978). That is, the sign of the head values in UNSAT-H is reversed: positive head values refer to suction values; negative head values refer to positive pressures (greater than atmospheric).

OPTIONS, CONSTANTS, AND LIMITS

Card 1 Column No.	Variable	Format (A80) Description
1-80	TITLE	Problem description or identifier. This variable is passed through to the output file.

Card 2		Format (4I5)	
Column No.	Variable	Description	
1-5	IPLANT	Plant option (0) No plants (1) Plants	
6-10	LOWER	Lower boundary condition option (1) Unit gradient (2) Static water table (4) Specified bottom flux (5) Impermeable boundary	
11-15	NGRAV	Orientation of the profile (0) Horizontal (1) Vertical	
16-20	ISWDIF	Option for time step control (0) Check all nodes to see if the percent change in THETA of every node is less than DMAXBA (1) Reduce the time step if the mass balance for the whole profile exceeds DMAXBA	

Card 3		Format (4I5)	
Column No.	Variable	Description	
1-5	NPRINT	Option for level of output (0) Daily summaries and end-of-year summary (1) DELSUB and daily summaries and end-of-year summary. Limited to 15 days because of the amount of output	
6-10	NDAYS	Number of days for which data is provided	
11-15	DAYEND	Ending day of the simulation. DAYEND cannot exceed 365 days	
16-20	NYEARS	Number of years for the simulation to be run. A value of 1 will result in a single-year simulation	

Card 4 Format (3I5)

Column No.	Variable	Description
1-5	NSURPE	Option to allow evaporation when there are no plants (0) No evaporation (1) Evaporation
6-10	NFHOURL	Option to distribute the daily potential evapotranspiration (PET) value over the 24 h of the day (1) User supplies 24 hourly factors (2) Hourly factors are generated with a sine wave function for the hours between 6 A.M. and 6 P.M., while the remaining hourly factors are set equal to 0.01
11-15	ITOPBC	Option for the surface boundary condition (0) Flux (1) Constant head. If this option is chosen, specify the constant surface head value, HTOP

Card 5 Format (3I5)

Column No.	Variable	Description
1-5	KOPT	Options for describing the soil hydraulic properties (1) Polynomial (2) Haverkamp moisture characteristic and conductivity function (Haverkamp et al. 1977) (3) Campbell function (Campbell 1974) (4) Haverkamp functions with head term in moisture characteristic function replaced with $\ln(\text{head})$
6-10	KEST	Option for estimating the conductivity at the midpoint between nodes (1) Arithmetic mean (2) Arithmetic mean with a weighting factor, WTF. If this option is chosen, a value must be supplied for WTF, or else the program uses 0.5 (3) Geometric mean
11-15	IVAPOR	Option to allow vapor flow. If chosen, values must be supplied for TORT, TSOILC, and VAPDIF

Card 6 Format (3F10.0)

Column No.	Variable	Description
1-10	HIRRI	Minimum head to which the soil can wet up
11-20	HDRY	Maximum head to which the soil can dry out (units: cm)
21-30	HTOP	Constant head value of the surface node when the ITOPBC = 1 option is chosen (units: cm)

Card 7		Format (3E10.0)	
Column No.	Variable	Description	
1-10	DMAXBA	Time step control parameter. If ISWDIF = 0, OMAXBA is the maximum allowable percent change in the water content of any node (suggested value is 0.01). If ISWDIF = 1, DMAXBA is the maximum allowable mass balance error (cm) between predicted and calculated soil profile moisture storage values (units: % or cm)	
11-20	DELMAX	Maximum allowable time step, normally 1 h (units: h)	
21-30	DELMIN	Minimum allowable time step (units: h)	

Card 8		Format (3E10.0)	
Column No.	Variable	Description	
1-10	WTF	Weighting factor (from 0.0 to 1.0) used to weight the conductivity of the upstream node in the calculation of midnode conductivities. A value of 0.5 weights the nodes equally	
11-20	RFACT	Maximum time step factor. The time step can be potentially increased by this factor following the completion of a successful time step	
21-30	RAINIF	Rainfall initiation factor. At the start of any water application event, the time step will be reduced by the RAINIF factor	

NODE INFORMATION

Card 9		Format (2I5)	
Column No.	Variable	Description	
1-5	MATN	Number of different soil materials	
6-10	NPT	Number of nodes	

Card 10		Format(4(I4,F10.0))
Column		
No.	Variable	Description
1-5	MAT(1)	Soil material identification number for the given node
6-10	Z(1)	Depth of node below the surface (units: cm)
	.	
	.	
	.	
	MAT(I)	
	Z(I)	

Note: Repeat Card 10 until all data have been entered

SOIL PROPERTY DESCRIPTION Note: Cards 11-19 are entered only if KOPT = 1

Card 11		Format(2I5)
Column		
No.	Variable	Description
1-5	MAXSUB	Maximum number of subdivisions of any given soil property description (limit is 4)
6-10	MAXPOL	Maximum degree of soil property polynomials (limit is 5)

Card 12		Format(A60)
Column		
No.	Variable	Description
1-60	DUMMY	Title describing the soil type and indicating that what follows is moisture characteristic data

Card 13		Format(1X,I5,2F15.0)
Column		
No.	Variable	Description
2-6	NSUBTH	Number of subdivisions of the moisture characteristic for that particular material
7-21	AIRINT	Air entry head (units: cm)
22-36	THET	Saturated water content (units: cm ³ /cm ³)

Card 14 Format(1X,2I5,2F15.0)

Column No.	Variable	Description
2-6	II	Index of the moisture characteristic polynomial
7-11	NDEGTH	Number of polynomial coefficients, or polynomial degree + 1
12-26	XX	Minimum head for which the given polynomial applies (units: cm)
27-41	XDIVTH	Maximum head for which the given polynomial applies (units: cm)

Card 15 Format(5F15.0)

Column No.	Variable	Description
1-15	CREGTH	Moisture characteristic coefficients

Note: Enter NDEGTH values of CREGTH

Note: Repeat Cards 14-15 NSUBTH times

Card 16 Format(A60)

Column No.	Variable	Description
1-60	DUMMY	Title describing the soil type and indicating that what follows is conductivity data

Card 17 Format(1X,I5,2F15.D)

Column No.	Variable	Description
2-6	NSUBKH	Number of subdivisions of conductivity curve for that particular material
7-21	AIRINK	Air entry head (units: cm)
22-36	SK	Saturated hydraulic conductivity (units: cm/h)

Card 18 Format(1X,2I5,2F15.0)

Column No.	Variable	Description
2-6	II	Index of the conductivity polynomial
7-11	NDEGKH	Number of polynomial coefficients, or polynomial degree + 1
12-26	XX	Minimum head for which the given polynomial applies
27-41	XDIVKH	Maximum head for which the given polynomial applies

Card 19 Format(5F15.0)

Column No.	Variable	Description
1-15	CREGKH	Polynomial coefficients for describing conductivity

Note: Enter NDEGKH values of CREGKH

Note: Repeat Cards 18-19 NSUBK times

Cards 20-23 are entered only if KOPT = 2 or 4

Card 20 Format(A60)

Column No.	Variable	Description
1-60	DUMMY	Title describing the soil type and indicating that what follows is moisture characteristic data

Card 21 Format(5F15.0)

Column No.	Variable	Description
1-15	AIRINT	Air entry head (units: cm)
16-30	THET	Saturated water content (units: cm^3/cm^3)
31-45	THTR	Residual water content (units: cm^3/cm^3)
46-60	ALPHA	Coefficient of Haverkamp function (Haverkamp et al. 1977)
61-75	BETA	Coefficient of Haverkamp function

Card 22 Format(A60)

Column No.	Variable	Description
1-60	DUMMY	Title describing the soil type and indicating that what follows is conductivity data

Card 23 Format(4F15.0)

Column No.	Variable	Description
1-15	AIRINK	Air entry head (units: cm)
16-30	SK	Saturated hydraulic conductivity (units: cm/h)
31-45	A	Coefficient of Haverkamp function
46-60	B	Coefficient of Haverkamp function

Enter Cards 24-27 if KOPT = 3

Card 24 Column		Format(A60)
No.	Variable	Description
1-60	DUMMY	Title describing the soil type and indicating that what follows is moisture characteristic data

Card 25 Column		Format(3F15.0)
No.	Variable	Description
1-15	AIRINT	Air entry head (units: cm)
16-30	THET	Saturated water content (units: cm^3/cm^3)
31-45	B	Coefficient of Campbell function (Campbell 1974)

Card 26 Column		Format(A60)
No.	Variable	Description
1-60	DUMMY	Title describing the soil type and indicating that what follows is conductivity data

Card 27 Column		Format(3F15.0)
No.	Variable	Description
1-15	AIRINK	Air entry head (units: cm)
16-30	SK	Saturated hydraulic conductivity (units: cm/h)
31-45	B	Coefficient of Campbell function (Campbell 1974)

Note: Repeat the appropriate soil property descriptions MATN times

Enter Card 28 if IVAPOR = 1

Card 28 Column		Format(3F15.0)
No.	Variable	Description
1-15	TORT	Tortuosity
16-30	TSOILC	Temperature of the soil (units: $^{\circ}\text{C}$)
31-45	VAPDIF	Diffusion coefficient of vapor in air (units: cm^2/s)

INITIAL CONDITIONS

Card 29 Column		Format(I5)
No.	Variable	Description
1-5	NDAY	Simulation starting day, when the initial conditions apply

Card 30 Format(4F14.0)

Column No.	Variable	Description
1-14	H(1)	Initial head (units: cm)
.		
.		
.		
	H(4)	

Note: Repeat Card 30 until NPT values of H have been entered

PLANT INFORMATION

Enter Card 31 if IPLANT = 1

Card 31 Format(6I5)

Column No.	Variable	Description
1-5	LEAF	Option for leaf area index (LAI) (0) LAI values not needed (i.e., NFPET = 2) (1) User supplies LAI values for the year (2) User supplies a subroutine (inoperative)
6-10	NFROOT	Option for root growth (1) Exponential relationship (2) User supplies a subroutine (inoperative)
11-15	NUPTAK	Option for plant water uptake (1) Macroscopic approach of Feddes (Feddes et al. 1978). Requires entry of THETAW, THETAD, and THETAN values (2) User supplies a subroutine (inoperative)
16-20	NFPET	Option for partitioning PET into transpiration (PT) and evaporation (PE) components (1) User supplies daily PET values and program partitions it into PT and PE based on LAI and equation by Ritchie (Ritchie and Burnett 1971) (2) User supplies daily PET values and program partitions it into PT and PE based on the cheatgrass data of Hinds (Hinds 1975)
21-25	NSOW	Day of the year on which seeds are planted
26-30	NHRVST	Number of plant growth days, from NSOW to harvest

Enter Card 32 if LEAF = 1

Card 32		Format(I5)
Column No.	Variable	Description
1-5	NPOINT	Number of changes in the LAI relationship when LEAF = 1

Card 33		Format(5(I5,F5.0))
Column No.	Variable	Description
1-5	NGROW(1)	Day of the year on which the LAI is changed to the corresponding FLAI value
6-10	FLAI(1)	Leaf area index on the corresponding NGROW day
.	.	.
41-45	NGROW(I)	
46-50	FLAI(I)	

Note: Repeat Card 33 until NPOINT pairs of NGROW, FLAI values have been entered

Enter Cards 34-35 if NROOT = 1

Card 34		Format(3F10.0)
Column No.	Variable	Description
1-10	AA	Coefficient in the exponential root growth equation
11-20	B1	Coefficient in the exponential root growth equation
21-30	B2	Coefficient in the exponential root growth equation

Card 35		Format(10I5)
Column No.	Variable	Description
1-5	NROOT(1)	Growth day on which roots reach the corresponding node
.	.	.
46-50	NROOT(I)	

Note: Repeat Card 35 until NPT values of NROOT have been entered

Enter Card 36 if NUPTAK = 1

Card 36 . Format(3F10.0)

Column No.	Variable	Description
1-10	HW	Head corresponding to moisture content below which plants wilt and stop transpiring (units: cm)
11-20	HD	Head corresponding to moisture content below which plant transpiration starts to decrease (units: cm)
21-30	HN	Head corresponding to moisture content above which plants do not transpire because of anaerobic conditions (units: cm)

Enter Card 37 if NFPET = 2

Card 37 Format(2F10.0)

Column No.	Variable	Description
1-10	BIOMAS	Plant shoot biomass scaling factor used in adjusting the transpiration component of PET. The base value is 220 g/m ² . A lower value will result in lower transpiration and higher evaporation, while a higher value will result in higher transpiration and lower evaporation (units: g/m ²)
11-20	BARE	Fraction of soil surface that is bare of plants

BOUNDARY CONDITIONS

Enter Card 38 if NFHOUR = 1

Card 38 Format(8F7.0)

Column No.	Variable	Description
1-8	FPET(1)	Hourly PET distribution factors. The factors should sum to 1.0
.	.	.
50-57	FPET(8)	

Note: Repeat Card 38 until 24 FPET values have been added

Card 39		Format(8F7.0)
Column No.	Variable	Description
1-8	PET(1)	Potential evapotranspiration (units: cm/day)
.	.	.
50-56	PET(8)	

Note: Repeat Card 39 until NDAYS values of PET have been entered

Enter Cards 40-41 if LOWER = 4

Card 40		Format(I5)
Column No.	Variable	Description
1-5	NTLEAK	Number of times the flux lower boundary condition changes during the simulation period

Card 41		Format(5(I5,F5.0))
Column No.	Variable	Description
1-5	NQLEAK(1)	Day on which the bottom boundary flux is changed to the corresponding QLEAK value
6-10	QLEAK(1)	Bottom boundary flux on the corresponding NQLEAK day (units: cm/h)
.	.	.
41-45	NQLEAK(5)	
46-50	QLEAK(5)	

Note: Repeat Card 41 until NTLEAK pairs of data have been entered

Card 42		Format(I5)
Column No.	Variable	Description
1-5	NWATER	Total number of days during which there is water application

Card 43		Format(3I5,F6.0)
Column		
No.	Variable	Description
1-5	IRDAY	Day on which a water application event occurs
6-10	IRTYPE	Option for type of water application (1) rainfall (2) irrigation (3) constant ponding depth less than 1 day (4) constant ponding depth greater than 1 day
11-15	NP	Number of times during the day that the water application rate changes
16-20	EFICEN	Efficiency of the irrigation scheme (i.e., how much of the water actually gets onto the soil surface)

Card 43a		Format(2F7.0)
Column		
No.	Variable	Description
1-7	RTIME(1)	Time of day when the water application rate changes (units: h). The exception is when IRTYPE = 4, and then the units are days. If IRTYPE = 4 and RTIME(NP) = 0, ponding is continued for the remainder simulation (until DAYEND)
8-14	AMOUNT(1)	Rainfall/irrigation amount that falls until the next rate change, RTIME(I+1), or when IRTYPE = 3 or 4, it is the height of water ponded on the surface (units: cm)

Note: Repeat Card 43a NP times

Note: Repeat Cards 43 and 43a NWATER times

REFERENCES

- Bond, F. W., C. R. Cole and P. J. Gutknecht. 1984. "Unsaturated Groundwater Flow Model (UNSAT1D) Computer Code Manual." CS-2434-CCM, Electric Power Research Institute, Palo Alto, California.
- Campbell, G. S. 1974. "A Simple Method for Determining Unsaturated Conductivity from Moisture Retention Data." Soil Sci. 177:311-314.
- Feddes, R. A., P. J. Kowalik and H. Zaradny. 1978. Simulation of Field Water Use and Crop Yield. John Wiley and Sons, Inc., New York.
- Gupta, S. K., K. K. Tanji, D. R. Nielsen, J. W. Biggar, C. S. Simmons and J. L. MacIntyre. 1978. "Field Simulation of Soil-Water Movement with Crop Water Extraction." Water Science and Engineering Paper No. 4013. Department of Land, Air, and Water Resources, University of California, Davis.
- Haverkamp, R., M. Vauclin, J. Touma, P. J. Wierenga and G. Vachaud. 1977. "A Comparison of Numerical Simulation Models for One-Dimensional Infiltration." Soil Sci. Soc. Am. J. 41:285-294.
- Hinds, W. T. 1975. "Energy and Carbon Balances in Cheatgrass: An Essay in Autecology." Ecological Monogr. 45:367-388.
- Ritchie, J. T., and E. Burnett. 1971. "Dryland Evaporative Flux in a Subhumid Climate, 2, Plant Influences." Agron. J. 63:56-62.

APPENDIX B

INPUT AND OUTPUT LISTINGS FOR THE INFILTRATION
TEST CASE FOR YOLO LIGHT CLAY

APPENDIX B

INPUT AND OUTPUT LISTINGS FOR THE INFILTRATION TEST CASE FOR YOLO LIGHT CLAY

```

CLAY.INP: SAMPLE simulation run for HCOOE document; 250 nodes
  0   5   1   0           IPLANT,LOWER,NGRAV,ISWDIF
  0  50  50   1           NPRINT,DAYENO,NOAYS,NYEARS
  0   2   1           NSURPE,NFHOUR,ITOPBC
  4   3   0           KOPT,KEST,IVAPOR
0.000E+00 2.000E+04       0.0       HIRRI,HDRY,HTOP
1.000E-02 0.139534 1.111E-02       DMAXBA,DELMAX,DELMIN
      0.5      2.0  1.0E-05       WTF,RFACT,RAINIF
  1 250           MATN,NPT
  1   0.000  1   1.000  1   2.000  1   3.000  MAT,Z
  1   4.000  1   5.000  1   6.000  1   7.000  MAT,Z
*** The intermediate node information has been left out ***
  1 244.000  1 245.000  1 246.000  1 247.000  MAT,Z
  1 248.000  1 249.000           MAT,Z
Yolo Light Clay Moisture Characteristic
      1.00      0.495      0.124      739.0      4.0
Yolo Light Clay Hydraulic Conductivity
      1.00      4.428E-02      124.6      1.77
  0           NDAY
      600.000      600.000      600.000      600.000  Initial Conditions
      600.000      600.000      600.000      600.000
*** The intermediate node information has been left out ***
      600.000      600.000      600.000      600.000
      600.000      600.000
  1           NWATER
  1   5   2 1.000       IRDAY,IRTYPE,NP,EFICEN
0.000 99.000       RTIME(1),AMOUNT(1)
50.000 0.000       RTIME(2),AMOUNT(2)

```

FIGURE B.1. Listing of the Sample Input File (SAMPLE.INP) for the Yolo Light Clay Simulation

UNSAT-H SIMULATION RESULTS

Input Filename: CLAY.BIN

Date: 26-AUG-85

Time: 18:02:09

INITIAL CONDITIONS

NODE	DEPTH	HEAD	THETA	NODE	DEPTH	HEAD	THETA
1	0.000	600.0	0.2376	2	1.000	600.0	0.2376
3	2.000	600.0	0.2376	4	3.000	600.0	0.2376
5	4.000	600.0	0.2376	6	5.000	600.0	0.2376
*** The intermediate node information has been left out ***							
245	244.000	600.0	0.2376	246	245.000	600.0	0.2376
247	246.000	600.0	0.2376	248	247.000	600.0	0.2376
249	248.000	600.0	0.2376	250	249.000	600.0	0.2376

Initial Water Storage = 59.1620 cm

DAILY SUMMARY: Day = 1, Total Simulated Time = 23.9998 h

Depth	29.00	59.00	89.00	119.00	149.00	179.00
Moist	0.2377	0.2376	0.2376	0.2376	0.2376	0.2376
Head	5.982E+02	6.000E+02	6.000E+02	6.000E+02	6.000E+02	6.000E+02
Flux	1.736E-03	1.607E-03	1.607E-03	1.607E-03	1.607E-03	1.607E-03

PRESTOR	INFIL	RUNOFF	EVAPO	TRANS	DRAIN	NEWSTOR	STORAGE
59.1620+	3.9028+	0.0000	- 0.0000-	0.0000-	0.0000	= 63.0648	Versus 63.1923

Mass Balance (cm) = 0.1275, Time steps = 377

DAILY SUMMARY: Day = 50, Total Simulated Time = 23.9998 h

Depth	29.00	59.00	89.00	119.00	149.00	179.00
Moist	0.4950	0.4950	0.4950	0.4950	0.4948	0.4930
Head	2.212E-01	5.114E-01	8.809E-01	1.384E+00	2.256E+00	4.152E+00
Flux	1.070E+00	1.070E+00	1.070E+00	1.070E+00	1.069E+00	1.060E+00

PRESTOR	INFIL	RUNOFF	EVAPO	TRANS	DRAIN	NEWSTOR	STORAGE
118.7326+	1.0701+	0.0000	- 0.0000-	0.0000-	0.0000	= 119.8027	Versus 119.8027

Mass Balance (cm) = 0.0000, Time steps = 172

FIGURE B.2. Listing of a Sample Output File (SAMPLE.LIS) for the Yolo Light Clay Simulation. (This output file was generated using DATAOUT.)

YEAR-END SUMMARY FOR 50 SIMULATION DAYS (units: cm)

Gravity Flow Factor	=	1.0000
Transpiration Scheme Is:	=	0
Total Potential Evapotranspiration (PET)	=	0.00000E+00
Actual Potential Transpiration	=	0.00000E+00
Total Transpiration	=	0.00000E+00
Actual Potential Evaporation	=	0.00000E+00
Total Evaporation	=	0.00000E+00
Evaporation During Growth	=	0.00000E+00
Total Runoff	=	0.00000E+00
Total Infiltration	=	60.513
Total Drainage at Base of Profile	=	0.00000E+00
Total Rainfall	=	0.00000E+00
Actual Rainfall	=	0.00000E+00
Total Irrigation	=	0.00000E+00
Total Final Moisture Storage	=	119.80
Mass Balance Error (cm)	=	-0.12753
Total Time Steps (this run)	=	8805
Total Simulated Time (days)	=	50.005

Total flux (cm) across different depths at the end of 50 days:

DEPTH	FLUX	DEPTH	FLUX	DEPTH	FLUX
0.000	6.0513E+01	0.500	6.0513E+01	1.500	6.0255E+01
2.500	5.9998E+01	3.500	5.9740E+01	4.500	5.9483E+01
5.500	5.9225E+01	6.500	5.8968E+01	7.500	5.8710E+01
*** The intermediate node information has been left out ***					
245.500	2.7283E-01	246.500	1.9200E-01	247.500	1.1414E-01
248.500	3.7949E-02	249.000	0.0000E+00		

FIGURE B.2. (cont)

The final actual mass balance error was 0.0012 cm and not the reported value of 0.1275 cm. The discrepancy is caused by UNSAT-H, which fails to account for the change in moisture content of the top node as its head value goes from 600 to 1 cm after time zero.

APPENDIX C

UNSAT-H SOURCE COMPUTER CODE

APPENDIX C

UNSAT-H SOURCE COMPUTER CODE

```

C** BVAR,INC
C** Real, Integer, and Logical Unit parameters
REAL LOGE
COMMON /RPARA/DMAXBA,DELMAX,DELMIN,HIRRI,HDRY,THMAX,
# SATURK,DRYK,SATURC,DRYC,AA,B1,B2,TMO1ST,LOGE
INTEGER DAYEND
COMMON /IPARA/IPLANT,LOWER,NDAYS,NDAY,KOPT,IVAPOR,
# MATN,NPT,NGRAY,MAXSUB,MAXPOL,NSURPE,
# NPOINT,INC,MXROOT,NWATER,NPRINT
C** Logical unit assignments
C LUI = binary input file unit (input to UNSATH)
C LUB = binary output file unit from DATAINH;
C binary output file unit from UNSATH
C LUR = read (interactive input unit)
C LUS = screen (interactive and additional output unit)
PARAMETER (LUI = 1,
# LUB = 2,
# LUR = 5,
# LUS = 6)
C** M1 = Number of nodes
PARAMETER (M1 = 250)
C** BARRAY,INC
C** Arrayed parameters
COMMON /ARRAY/Z(M1),H(M1),NTROOT(M1),HH(M1),RDF(M1),
# THETA(M1),ZK(M1),C(M1),ZZK(M1),CC(M1),
# FPET(24),THETAM(5),THETAD(5),THETAN(5),
# PTRANS(365),PEVAPO(365),QLEAK(365)
C** BSOIL,INC
C** Soil parameters
COMMON /IHY/MAT(M1),IHY(320)
DIMENSION NSUBTH(5),AIRINT(5),THET(5),
# NDEGTH(20),XDIVTH(20),CREGTH(100),
# NSUBKH(5),AIRINK(5),SK(5),
# NDEGKH(20),XDIVKH(20),CREGKH(100),
# AMOUNT(24),SLOPTH(5),SLOPKH(5)
EQUIVALENCE (IHY(1),NSUBTH),(IHY(6),AIRINT),(IHY(11),THET),
# (IHY(16),NDEGTH),(IHY(36),XDIVTH),(IHY(56),CREGTH),
# (IHY(156),NSUBKH),(IHY(161),AIRINK),(IHY(166),SK),
# (IHY(171),NDEGKH),(IHY(191),XDIVKH),(IHY(211),CREGKH),
# (IHY(311),SLOPTH),(IHY(316),SLOPKH)
C** BTIME,INC
C** Time parameters
COMMON /TIME/DELT,DIFMAX,DELSAV,
# TLEFT,OELSUB,TSUB,ICHECK,TEND,IR,NRATES,RTIME(24),
# TDONE,1DAY,N,SUBEND,1G0150,RFACT,TSTART
0001 PROGRAM DATAINH
0002 C-----
0003 C Last update was 30-AUG-1985 by MJFayer
0004 C
0005 C DATAINH reads the data file *.INP, checks
0006 C the data and performs some calculations, then writes the
0007 C necessary information to the binary output file *.BIN,
0008 C which serves as the input file to the UNSAT-H code
0009 C-----
0010 CHARACTER*80 FILE,OFIL,FILE,TITLE,DUMMY*60,STIME*8,SDATE*9
0011 LOGICAL LUSLW
0012 REAL MOLAR,MGR
0013 INCLUDE 'BVAR,INC/NOLIST'
0014 INCLUDE 'BARRAY,INC/NOLIST'
0015 INCLUDE 'BSOIL,INC/NOLIST'
0016 DIMENSION ZZ(M1),RDF(M1),RTIME(24),QLEAK(M1),QLEAK(M1),
0017 # NGROW(30),FLA1(30),DLA1(365),PET(365)

```

```

0071 FUNED(TD) = EXP(54.878919-(6790.4985/TD)-5.02808*LOG(TD))
0072 LOGE = LOG10(EXP(1.0))
0073 LUSLUW = .TRUE.
0074
0075 NSTOP = 0
0076 CALL TIME(STIME)
0077 CALL DATE(SDATE)
0078 WRITE(LUS,1010)
0079 READ(LUR,1000) FILE
0080 NCHR = INDEX(FILE,' ')
0081 FILE(NCHR:NCHR+3) = '.INP'
0082 OPEN(UNIT=LUI,NAME=FILE,TYPE='OLD',READONLY)
0083 WRITE(LUS,1020)
0084 READ(LUR,*) IPRINT
0085 IF (IPRINT .GT. 0) THEN
0086   WRITE(LUS,1030)
0087   READ(LUR,*) IOUT
0088   IF (IOUT .EQ. 0) THEN
0089     LUW = LUS
0090     LUSLUW = .FALSE.
0091   ELSE
0092     OFILE = FILE
0093     OFILE(NCHR:NCHR+3) = '.LIS'
0094     WRITE(LUS,1040) OFILE
0095     OPEN(UNIT=7,FILE=OFILE,STATUS='NEW')
0096   ENDIF
0097   WRITE(LUW,1050) FILE,SDATE,STIME
0098 ENDIF
0099 1010 FORMAT(/' Enter input filename (without ".INP" extension)',
0100 # ' ==> ',/$)
0101 1020 FORMAT(/' Output options in addition to creating < >.BIN:',/,
0102 # ' (0) = no output',/,
0103 # ' (1) = options, nodes, soil properties',/,
0104 # ' and initial conditions',/,
0105 # ' (2) = (1) plus PET and rain information',/,
0106 # ' (3) = (2) plus leaf and bottom',/,
0107 # ' boundary flux information',/,
0108 # ' Enter choice ==> ',/$)
0109 1030 FORMAT(/' Options for output device:',/,
0110 # ' (0) screen',/,
0111 # ' (1) file',/,
0112 # ' Choose output device option ==> ',/$)
0113 1040 FORMAT(/' The input summary filename is ',A<NCHR+3>/)
0114 1050 FORMAT(/' Input Filename: ',A<NCHR+3>,'T35,'Date: ',A9,T60,
0115 # 'Time: ',A8,/)
0116
0117 C*****
0118 C OPTIONS, CONSTANTS, AND LIMITS
0119 C*****
0120
0121 C-----
0122 C TITLE
0123 C-----
0124 READ(LUI,1000) TITLE
0125 IF (IPRINT .GT. 0) WRITE(LUW,2000) TITLE
0126
0127 C-----
0128 C IPLANT,LOWER,NGRAV,ISWDF
0129 C-----
0130 READ(LUI,1001) IPLANT,LOWER,NGRAV,ISWDF
0131 IF (IPRINT .GT. 0) WRITE(LUW,2010) IPLANT,LOWER,NGRAV,ISWDF
0132 2010 FORMAT(' Options chosen include: '// ' IPLANT = ',I2,T22,
0133 # ' LOWER = ',I2,T42,'NGRAV = ',I2,T62,'ISWDF = ',I2)
0134
0135 C-----
0136 C NPRINT,DAYEND,NDAYS,NYEARS
0137 C-----

```

```

0138 READ(LU1,1001) NPRINT,DAYEND,NDAYS,NYEARS
0139 IF (NPRINT .EQ. 1 .AND. DAYEND .GT. 15) THEN
0140 WRITE(LUS,2015) DAYEND
0141 IF (LUSLUW) WRITE(LUW,2015) DAYEND
0142 DAYEND = 15
0143 ENDIF
0144 IF (IPRINT .GT. 0) WRITE(LUW,2020) NPRINT,DAYEND,NDAYS,NYEARS
0145 2015 FORMAT(/' WARNING: The choice of hourly output (NPRINT = 1)/',
0146 # ' and DAYEND = ',13,' will result in too',/,
0147 # ' large a *.RES file when UNSAT-H is run.',/,
0148 # ' Therefore, DAYEND is reset to 15.',/)
0149 2020 FORMAT(/' NPRINT = ',13,T22,'DAYEND = ',13,T42,'NDAYS = ',13,
0150 # T62,'NYEARS = ',13)
0151 IF (NDAYS .GT. 365) STOP ' Program stopped because NDAYS > 365'
0152
0153 C-----
0154 C NSURPE,NFHOURL,ITOPBC
0155 C-----
0156 READ(LU1,1002) NSURPE,NFHOURL,ITOPBC
0157 IF (IPRINT .GT. 0) WRITE(LUW,2030) NSURPE,NFHOURL,ITOPBC
0158 2030 FORMAT(/' NSURPE = ',13,T22,'NFHOURL = ',13,T42,'ITOPBC = ',13)
0159
0160 C-----
0161 C KOPT,KEST,IVAPOR
0162 C-----
0163 READ(LU1,1001) KOPT,KEST,IVAPOR
0164 IF (IPRINT .GT. 0)
0165 # WRITE(LUW,2040) KOPT,KEST,IVAPOR
0166 2040 FORMAT(/' KOPT = ',13,T22,'KEST = ',13,T42,'IVAPOR = ',13)
0167
0168 C-----
0169 C HIRRI,HDRY,HTOP
0170 C-----
0171 READ(LU1,1003) HIRRI,HDRY,HTOP
0172 IF (IPRINT .GT. 0) WRITE(LUW,2050) HIRRI,HDRY,HTOP
0173 2050 FORMAT(/' HIRRI = ',G10.3,T22,'HDRY = ',G10.3,T42,'HTOP = ',
0174 # G10.3)
0175
0176 C-----
0177 C DMAXBA,DELMAX,DELMIN
0178 C-----
0179 READ (LU1,1003) DMAXBA,DELMAX,DELMIN
0180 IF (IPRINT .GT. 0) WRITE(LUW,2060) DMAXBA,DELMAX,DELMIN
0181 2060 FORMAT(/' DMAXBA = ',1PG10.3,T22,'DELMAX = ',G10.3,T42,
0182 # 'DELMIN = ',G10.3)
0183
0184 C-----
0185 C WTF,RFACT,RAINIF
0186 C-----
0187 READ(LU1,1003) WTF,RFACT,RAINIF
0188 IF (IPRINT .GT. 0) WRITE(LUW,2070) WTF,RFACT,RAINIF
0189 2070 FORMAT(/' WTF = ',1PG10.3,T22,'RFACT = ',G10.3,T42,
0190 # 'RAINIF = ',G10.3)
0191 UP = WTF
0192 DOWN = 1.0-UP
0193
0194 C*****
0195 C NODE INFORMATION: Define geometric details of the problem
0196 C*****
0197
0198 C-----
0199 C MATN,NPT
0200 C-----
0201 READ(LU1,1001) MATN,NPT
0202 IF (IPRINT .GT. 0) WRITE(LUW,3000) MATN,NPT
0203 3000 FORMAT(/' MATN = ',12,T22,'NPT = ',13)
0204 IF (NPT .GT. MI) STOP ' NPT exceeds node dimensions'

```



```

0205
0206 C-----
0207 C   MAT,Z: Material type and depth below the surface
0208 C-----
0209 READ(LUI,3010) (MAT(I), Z(I),I=1,NPT)
0210 3010 FORMAT(4(I4,F10.0))
0211 DO 300 I=1,NPT-1
0212     ZZ(I) = Z(I+1)-Z(I)
0213 300 CONTINUE
0214     ZZ(NPT) = 0.0
0215
0216 C*****
0217 C   SOIL PROPERTY DESCRIPTION
0218 C*****
0219     MAXSUB = 1
0220     MAXPOL = 4
0221     INC = MAXSUB*MAXPOL
0222     IF (KOPT .NE. 1) GO TO 400
0223
0224 C-----
0225 C   KOPT = 1: Polynomial functions
0226 C-----
0227 READ(LUI,1001) MAXSUB,MAXPOL
0228 IF (IPRINT .GT. 0) WRITE(LUW,4000) MAXSUB,MAXPOL
0229 IF (MATN*MAXSUB*MAXPOL .GT. 100)
0230     # STOP ' Dimension of regression coefficients exceeded'
0231 INC = MAXSUB*MAXPOL
0232 DO 410 I=1, MATN
0233 READ(LUI,1000) DUMMY
0234 READ (LUI,4010) NSUBTH(I),AIRINT(I),THET(I)
0235 IF (IPRINT .GT. 0) WRITE(LUW,4020) I,DUMMY,NSUBTH(I),AIRINT(I),
0236     # THET(I)
0237 DO 420 J=1,NSUBTH(I)
0238     ID = (I-1)*MAXSUB+J
0239     IP = (I-1)*INC+(J-1)*MAXPOL
0240 READ(LUI,4030) II,NDEGTH(ID),XX,XDIVTH(ID)
0241 IF (J .GT. 1) THEN
0242     IF (XDIVTH(ID) .LT. XDIVTH(ID-1)) THEN
0243         WRITE(LUS,4040) I,J,XDIVTH(ID),XDIVTH(ID-1)
0244         IF (LUSLUW) WRITE(LUW,4040) I,J,XDIVTH(ID),XDIVTH(ID-1)
0245         NSTOP = 1
0246     ENDIF
0247 ENDIF
0248 READ(LUI,4050) (CREGTH(IP+L),L=1,NDEGTH(ID))
0249 IF (IPRINT .GT. 0) WRITE(LUW,4060) ID,NDEGTH(ID),XDIVTH(ID),
0250     # (CREGTH(IP+L),L=1,NDEGTH(ID))
0251 420 CONTINUE
0252 4000 FORMAT(' KOPT = 1: Polynomial functions for soil hydraulic'
0253     # ' properties'//,
0254     # '          MAXSUB = ',11,' , MAXPOL = ',11/)
0255 4010 FORMAT(1X,15,2F15.0)
0256 4020 FORMAT('/' THETA vs H, MAT ',11,': ',A60, '//,
0257     # ' NSUBTH = ',11,' , AIRINT = ',F9.4,' , THET = ',F6.4, '//
0258     # ' ID NDEGTH XDIVTH CREGTH'//,
0259     # ' -- -----'//)
0260 4050 FORMAT(5F15.0)
0261 4030 FORMAT(1X,215,2F15.0)
0262 4060 FORMAT(0P,T2,I2,T8,I2,T14,1P,E12.4,(T28,E13.6))
0263 4040 FORMAT(' ERROR: Material No. ',11,' , J = ',13,' , XDIVTH(J) = ',
0264     # 'E15.7,' < XDIVTH(J-1) = ',E15.7)
0265 READ(LUI,1000) DUMMY
0266 READ(LUI,4010) NSUBKH(I),AIRINK(I),SK(I)
0267 IF (IPRINT .GT. 0)
0268     # WRITE(LUW,4070) I,DUMMY,NSUBKH(I),AIRINK(I),SK(I)
0269 DO 430 J=1,NSUBKH(I)
0270     ID = (I-1)*MAXSUB+J
0271     IP = (I-1)*INC+(J-1)*MAXPOL

```

```

0272 READ(LUI,4030) I1,NDEGKH(ID),XX,XDIVKH(ID)
0273 IF (J.GT. 1) THEN
0274 IF (XDIVKH(ID).LT. XDIVKH(ID-1)) THEN
0275 WRITE(LUS,4080) I,J,XDIVKH(ID),XDIVKH(ID-1)
0276 IF (LUSLUW) WRITE(LUW,4080) I,J,XDIVKH(ID),XDIVKH(ID-1)
0277 NSTOP = 1
0278 ENDIF
0279 ENOIF
0280 READ(LUI,4050) (CREGKH(IP+L),L=1,NDEGKH(ID))
0281 IF (IPRINT.GT. 0) WRITE(LUW,4060) ID,NDEGKH(ID),XDIVKH(ID),
0282 # (CREGKH(IP+L),L=1,NDEGKH(ID))
0283 430 CONTINUE
0284 410 CONTINUE
0285 4080 FORMAT(' ERROR: Material No. ',I1,' J = ',I3,' XDIVKH(J) = ',
0286 # E15.7,' < XDIVKH(J-1) = ',E15.7)
0287 4070 FORMAT('/ K vs H, MAT ',I1,' : ',A60,/,
0288 # ' NSUBKH = ',I1,' AIRINK = ',F9.4,' SK = ',1P,E12.4,/,
0289 # ' ID NDEGKH XDIVKH CREGKH',
0290 # ' -----')
0291 GO TO 440
0292 400 IF (KOPT.NE. 2 .AND. KOPT.NE. 4) GO TO 450
0293
0294 C-----
0295 C KOPT = 2,4: Haverkamp functions
0296 C-----
0297 IF (IPRINT.GT. 0) WRITE(LUW,4100)
0298 DO 460 I=1,MATN
0299 NSUBTH(I) = 1
0300 NSUBKH(I) = 1
0301 READ(LUI,1000) DUMMY
0302 READ(LUI,4050) AIRINT(I),THET(I),THTR,A,B
0303 IF (IPRINT.GT. 0)
0304 # WRITE(LUW,4110) I,DUMMY,AIRINT(I),A,THET(I),B,THTR
0305 ID = (I-1)*MAXSUB+1
0306 IP = (I-1)*INC
0307 NDEGTH(ID) = 3
0308 XDIVTH(ID) = 2*AIRINT(I)
0309 CREGTH(IP+1) = THTR
0310 CREGTH(IP+2) = A
0311 CREGTH(IP+3) = B
0312 CREGTH(IP+4) = 0.
0313 READ(LUI,1000) DUMMY
0314 READ(LUI,4050) AIRINK(I),SK(I),ALPHA,BETA
0315 IF (IPRINT.GT. 0)
0316 # WRITE(LUW,4120) I,DUMMY,AIRINK(I),ALPHA,SK(I),BETA
0317 NDEGKH(ID) = 2
0318 XDIVKH(ID) = 2.*AIRINK(I)
0319 CREGKH(IP+1) = ALPHA
0320 CREGKH(IP+2) = BETA
0321 CREGKH(IP+3) = 0.0
0322 460 CONTINUE
0323 4100 FORMAT(' KOPT = 2: Haverkamp functions for soil hydraulic',
0324 # ' properties')
0325 4110 FORMAT('/ THETA vs H, MAT ',I1,' : ',A60,/,
0326 # ' AIRINT = ',G12.5,T50,' A = ',G12.5,/,
0327 # ' THET = ',G12.5,T50,' B = ',G12.5,/,
0328 # ' THTR = ',G12.5,/)
0329 4120 FORMAT(' K vs H, MAT ',I1,' : ',A60,/,
0330 # ' AIRINT = ',G12.5,T50,' ALPHA = ',G12.5,/,
0331 # ' SK = ',G12.5,T50,' BETA = ',G12.5)
0332 GO TO 440
0333 450 IF (KOPT.NE. 3) GO TO 470
0334
0335 C-----
0336 C KOPT = 3: Campbell functions
0337 C-----
0338 IF (IPRINT.GT. 0) WRITE(LUW,4200)

```

```

0339      DO 480 I=1,MATN
0340          NSUBTH(I) = 1
0341          NSUBKH(I) = 1
0342          READ(LUI,1000) DUMMY
0343          READ(LUI,4050) AIRINT(I),THET(I),B
0344          IF (IPRINT.GT. 0) WRITE(LUW,4210) I,DUMMY,AIRINT(I),B,THET(I)
0345          ID = (I-1)*MAXSUB+1
0346          IP = (I-1)*INC
0347          NDEGTH(ID) = 1
0348          XDIVTH(ID) = 2*AIRINT(I)
0349          CREGTH(IP+1) = B
0350          CREGTH(IP+2) = 0.0
0351          READ(LUI,1000) DUMMY
0352          READ(LUI,4050) AIRINK(I),SK(I),B
0353          IF (IPRINT.GT. 0) WRITE(LUW,4220) I,DUMMY,AIRINK(I),B,SK(I)
0354          NDEGKH(ID) = 1
0355          XDIVKH(ID) = 2.*AIRINK(I)
0356          CREGKH(IP+1) = B
0357          CREGKH(IP+2) = 0.0
0358      480 CONTINUE
0359      4200 FORMAT(' KOPT = 3: Campbell functions for',
0360          # ' soil hydraulic properties')
0361      4210 FORMAT('/' THETA vs H, MAT '1,1,',' ',A60, '//
0362          # ' AIRINT = ',G12.5,T50,' B = ',G12.5,/,
0363          # ' THET = ',G12.5)
0364      4220 FORMAT(' K vs H, MAT '1,1,',' ',A60, '//
0365          # ' AIRINK = ',G12.5,T50,' B = ',G12.5,/,
0366          # ' SK = ',G12.5)
0367      GO TO 440
0368      470 WRITE(LUS,4230) KOPT
0369          IF (LUSLUW) WRITE(LUW,4230) KOPT
0370      4230 FORMAT(' ERROR: Current KOPT options are 1, 2, 3, and 4.',/,
0371          # ' The input value was ',I2)
0372
0373      440 CONTINUE
0374          H(1) = HIRRI
0375          H(2) = HORY
0376          MAT2 = MAT(2)
0377          MAT(2) = MAT(1)
0378          CALL POLYTH(1,THETA,H)
0379          CALL POLYKH(2,ZK,H)
0380          CALL POLYCH(2,C,H)
0381
0382      C-----
0383      C Estimate THETA and ZK for two points
0384      C-----
0385          THMAX = THETA(1)
0386          SATURK = ZK(1)
0387          DRYK = ZK(2)
0388          SATURC = C(1)
0389          DRYC = C(2)
0390          IF (IPRINT.GT. 0) THEN
0391              WRITE(LUW,4300) HIRRI,THMAX,SATURK,SATURC
0392              WRITE(LUW,4310) HORY,THORY,DRYK,DRYC
0393          ENDIF
0394      4300 FORMAT('/' Surface node hydraulic properties'//,
0395          # ' HIRRI = ',G12.2,' THETA = ',F6.4,' K = ',1P,E12.4,
0396          # ' C = ',E12.4)
0397      4310 FORMAT(' HORY = ',G12.2,' THETA = ',F6.4,' K = ',1P,E12.4,
0398          # ' C = ',E12.4)
0399          MAT(2) = MAT2
0400
0401          MAT1 = MAT(1)
0402      DO 490 I=1,MATN
0403          MAT(I) = 1
0404          H(1) = AIRINT(I)
0405          CALL POLYTH(1,THETA,H)

```

```

0406      SLOPTH(1) = (THETA(1)-THET(1))/AIRINT(1)
0407      IF (ABS(SLOPTH(1)) .LT. 1.0E-7*THET(1)) SLOPTH(1) = 0.0
0408      IF (SLOPTH(1) .GT. 0.0) THEN
0409          WRITE(LUS,4320) 1,THETA(1),THET(1)
0410          NSTOP = 1
0411      ENDOF
0412      H(1) = AIRINK(1)
0413      CALL POLYKH(1,ZK,H)
0414      SLOPKH(1) = (ZK(1)-SK(1))/AIRINK(1)
0415      IF (ABS(SLOPKH(1)) .LT. 1.0E-7*SK(1)) SLOPKH(1) = 0.0
0416      IF (SLOPKH(1) .GT. 0.0) THEN
0417          WRITE(LUS,4330) 1,ZK(1),SK(1)
0418          NSTOP = 1
0419      ENDOF
0420 490 CONTINUE
0421      MAT(1) = MAT1
0422 4320 FORMAT(/' ERROR: For Material No. ',11,' the air-entry water',/
0423      #      ' content ('F6.4,') is greater than the ',/
0424      #      ' saturated water content ('F6.4,')')
0425 4330 FORMAT(/' ERROR: For Material No. ',11,' the air-entry ',/
0426      #      ' conductivity ('F6.4,') is greater than ',/
0427      #      ' the saturated conductivity ('F6.4,')')
0428
0429 C-----
0430 C      IVAPOR = 1 indicates that vapor flow will be operative.
0431 C
0432 C      TSOIL is the average soil temperature for the period of interest
0433 C      For Hanford, the average annual soil temperature at the 90 cm
0434 C      depth is 15.3 C for the years 1952-1980.
0435 C      TORT is pore tortuosity, which is usually reported as 0.66
0436 C      VAPDIF is the diffusivity of water vapor in air,
0437 C      approximately 0.24 cm cm/s
0438 C-----
0439      MOLAR = 18.0
0440      GRAV = 980.7
0441      GASCON = 8.3143E+07
0442      MGR = MOLAR*GRAV/GASCON
0443      WATDEN = 1.0
0444      IF (IVAPOR .EQ. 1) THEN
0445          READ(LU1,1003) TORT,TSOILC,VAPDIF
0446          TSOIL = TSOILC+273.16
0447          VAPDEN = FUNED(TSOIL)*MOLAR*1000.0/(GASCON*TSOIL)
0448          VAPCOE = 3600.0*MGR*VAPDEN*VAPDIF*TORT/(TSOIL*WATDEN)
0449          IF (IPRINT .GT. 1)
0450              #      WRITE(LUW,4340) TORT,TSOIL,VAPDEN,VAPDIF,MGR,VAPCOE
0451      ENDOF
0452 4340 FORMAT(/' IVAPOR = 1: This option allows vapor flow',/,
0453      #      ' Soil tortuosity = ',F6.4,/,
0454      #      ' Soil temperature = ',F6.2,/,1P,
0455      #      ' Soil vapor density (g/cm3) = ',E10.3,/,
0456      #      ' Vapor diffusivity in air (cm2/s) = ',E10.3,/,
0457      #      ' MOLAR*GRAV/GASCON = ',E10.3,/,
0458      #      ' VAPCOE (consolidated coef., cm/s) = ',E10.3)
0459      IF (IPRINT .GT. 0) WRITE(LUW,1004)
0460
0461 C*****
0462 C      INITIAL CONDITIONS
0463 C*****
0464
0465 C-----
0466 C      Input initial head conditions (cm)
0467 C      Output node, material, depth, and initial condition status
0468 C-----
0469      READ(LU1,5000) NDAY,(H(1),I=1,NPT)
0470      CALL POLYTH(NPT,THETA,H)
0471      CALL POLYCH(NPT,C,H)
0472      CALL POLYKH(NPT,ZK,H)

```

```

0473       IF (IPRINT .GT. 0) THEN
0474         WRITE(LUW,5010) NDAY
0475         WRITE(LUW,5020) (I,Z(I),MAT(I),H(I),ZK(I),C(I),THETA(I),I=1,NPT)
0476       ENDIF
0477 5010 FORMAT(' NDAY = ',I3, '//, T2, 'NODE', T11, 'Z', T16,
0478 # 'MAT', T24, 'HEAD', T34, 'CONDUCTIVITY', T52, 'CAPACITY', T66, 'THETA', '/',
0479 # T2, '-----', T9, '-----', T16, '-----', T23, '-----', T34,
0480 # '-----', T50, '-----', T65, '-----')
0481 5020 FORMAT(T2, I3, T7, F7.2, T17, I1, T20, F9.2, T33, I1, T46, F12.4, T50, F12.4, T65,
0482 # OP, F6.4)
0483       TMOIST = 0.5*(THETA(1)*ZZ(1)+THETA(NPT)*ZZ(NPT-1))
0484       DO 500 I=2,NPT-1
0485         TMOIST = TMOIST+THETA(I)*(ZZ(I)+ZZ(I-1))*0.5
0486 500   CONTINUE
0487       IF (IPRINT .GT. 0) THEN
0488         WRITE(LUW,5030) TMOIST
0489         WRITE(LUW,1004)
0490       ENDIF
0491       IF (IPLANT .EQ. 0) GO TO 690
0492
0493 C-----
0494 C   PLANT DATA (IPLANT = 1)
0495 C
0496 C   Input LEAF,NFROOT,NUPTAK,NFPET,NSOW,NHRVST
0497 C-----
0498       READ(LU1,1001) LEAF,NFROOT,NUPTAK,NFPET,NSOW,NHRVST
0499       IF (IPRINT .GT. 1) WRITE(LUW,6000) LEAF,NFROOT,NUPTAK,NFPET,
0500 # NSOW,NHRVST
0501 6000 FORMAT(' IPLANT = 1'// ' LEAF=',I3, ' NFROOT=',I3, ' NUPTAK=',
0502 # I3, ' NFPET=',I3, ' NSOW=',I3, ' NHRVST=',I3)
0503
0504 C-----
0505 C   LEAF = 1   Leaf area index NGROW,FLAI
0506 C           = 2   User-supplied subroutine for leaf area index
0507 C-----
0508       IF (LEAF .EQ. 1) THEN
0509         READ(LU1,6010) NPOINT,(NGROW(I),FLAI(I),I=1,NPOINT)
0510         IF (NPOINT .GT. 30) THEN
0511           STOP ' NPOINT > 30, DIM OF NGROW, FLAI EXCEEDED'
0512         ENDIF
0513         IF (IPRINT .GT. 1) WRITE(LUW,6020) NPOINT,
0514 # (NGROW(I),FLAI(I),I=1,NPOINT)
0515         J = 1
0516         DO 600 I=1,NDAYS
0517           IF (I .LT. NGROW(1)) THEN
0518             DLAI(I) = 0.0
0519           ELSE IF (I .GT. NGROW(NPOINT)) THEN
0520             DLAI(I) = 0.0
0521           ELSE IF (I .GE. NGROW(1) .AND.
0522 # I .LE. NGROW(NPOINT)) THEN
0523             IF (I .GE. NGROW(J) .AND. J .LT. NPOINT) J = J+1
0524             JJ = J-1
0525             DIFF = (FLAI(J)-FLAI(JJ))/(NGROW(J)-NGROW(JJ))
0526             DLAI(I) = FLAI(JJ)+DIFF*(I-NGROW(JJ))
0527           ENDIF
0528 600   CONTINUE
0529         IF (IPRINT .GT. 2) WRITE(LUW,6030) (I,DLAI(I),I=1,NDAYS)
0530       ELSE IF (LEAF .EQ. 2) THEN
0531         WRITE(LUS,*) ' Subroutine MYLAI was not provided'
0532         NSTOP = 1
0533       ELSE IF (LEAF .GT. 2) THEN
0534         WRITE(LUS,6040) LEAF
0535         IF (LUSLUW) WRITE(LUW,6040) LEAF
0536         NSTOP = 1
0537       ENDIF
0538
0539       IF (NFROOT .NE. 1 .AND. NFROOT .NE. 2) THEN

```

```

0540      WRITE(LUS,6050) NFROOT
0541      IF (LUSLUW) WRITE(LUW,6050) NFROOT
0542      NSTOP = 1
0543      GO TO 610
0544  ENDIF
0545      IF (NFROOT .EQ. 2) GO TO 620
0546
0547  C-----
0548  C      NFROOT = 1      NTRoot,AA,B1,B2
0549  C-----
0550      READ (LU1,1003) AA,B1,B2
0551      IF (IPRINT .GT. 1) WRITE(LUW,6060) AA,B1,B2
0552      READ (LU1,1001) (NTRoot(i),i=1,NPT)
0553      I = 1
0554  630  IF (NTRoot(i) .LT. NHRVST .AND. I .LE. NPT) THEN
0555          I = I+1
0556          GO TO 630
0557      ENDIF
0558      MDEPTH = I-1
0559      MXROOT = MDEPTH
0560      IF (IPRINT .LT. 2) GO TO 640
0561      WRITE(LUW,6070)
0562  6070  FORMAT(/' Root depth, density, and weight/node versus depth'/)
0563      WRITE(LUW,6080)
0564
0565  C-----
0566  C      Estimate root density factor (RDF) at each node of root depth
0567  C-----
0568      ROOTS = 0.
0569      ZZ0 = 0.
0570      RDFF(1) = 0.
0571      DO 650 I=2,MDEPTH
0572          RDFF(I) = AA*EXP(-B1*Z(I))+B2
0573          ROOTS = ROOTS+RDFF(I)*(Z(I)+ZZ0)*0.5
0574          ZZ0 = Z(I)
0575  650  CONTINUE
0576      IF (MDEPTH .GT. 1) ROOTS = ROOTS-RDFF(MDEPTH)*ZZ0*0.5
0577      DO 660 I=1,MDEPTH
0578          RDF(I) = RDFF(I)/ROOTS
0579  660  CONTINUE
0580      WRITE(LUW,6090) (NTRoot(i),Z(i),RDFF(i),RDF(i),i=1,MDEPTH)
0581  640  IF (IPRINT .GT. 0) THEN
0582          WRITE(LUW,6100) MXROOT
0583          WRITE(LUW,1004)
0584      ENDIF
0585  6100  FORMAT(/' MXROOT (deepest node to which roots penetrate) = ',I2)
0586      GO TO 610
0587
0588  C-----
0589  C      NFROOT = 2      User-defined root density
0590  C-----
0591  620  WRITE(LUS,*) ' Subroutine "MYROOT" was not provided'
0592      WRITE(LUW,*) ' Subroutine "MYROOT" was not provided'
0593
0594  610  IF (NUPTAK .EQ. 1) THEN
0595
0596  C-----
0597  C      NUPTAK = 1      HW,HD,HN
0598  C-----
0599      IF (IPRINT .GT. 1) WRITE(LUW,6110)
0600      MATSAY = MAT(1)
0601      DO 670 I=1,MATN
0602          MAT(I) = 1
0603          READ (LU1,1003) HW,HD,HN
0604          CALL POLYTH(1,THETAW(I),HW)
0605          CALL POLYTH(1,THETAD(I),HD)
0606          CALL POLYTH(1,THETAN(I),HN)

```

```

0607         IF (IPRINT .GT. 1) WRITE(LUW,6120) I, THETA(I), THETA(I),
0608         # THETA(I)
0609 670 CONTINUE
0610         MAT(I) = MATSAV
0611         ELSE IF (NUPTAK .EQ. 2) THEN
0612
0613 C-----
0614 C      NUPTAK = 2   User-defined MYSINK
0615 C-----
0616         WRITE(LUS,*) ' User subroutine "MYSINK" was not provided'
0617         IF (LUSLUW)
0618         # WRITE(LUW,*) ' User subroutine "MYSINK" was not provided'
0619         NSTOP = 1
0620         ELSE
0621         WRITE(LUS,6130) NUPTAK
0622         IF (LUSLUW) WRITE(LUW,6130) NUPTAK
0623         NSTOP = 1
0624         ENDIF
0625         IF (IPRINT .GT. 1) WRITE(LUW,1004)
0626         IF (NFPET .EQ. 2) READ (LUI,1003) BIOMAS,BARE
0627
0628 C*****
0629 C      BOUNDARY CONDITIONS
0630 C*****
0631
0632 690 IF (IPLANT .EQ. 0) THEN
0633
0634 C-----
0635 C      IPLANT = 0   No plants      NSURPE
0636 C-----
0637         IF (IPRINT .GT. 1) THEN
0638         IF (NSURPE .EQ. 0) WRITE(LUW,7000) NSURPE
0639         IF (NSURPE .EQ. 1) WRITE(LUW,7010) NSURPE
0640         ENDIF
0641         IF (NSURPE .EQ. 0) THEN
0642         IF (IPRINT .GT. 0) WRITE(LUW,1004)
0643         GO TO 700
0644         ENDIF
0645         ENDIF
0646 7000 FORMAT(' NSURPE = ',11,' : There will be no surface',
0647         # ' evaporation')
0648 7010 FORMAT(' NSURPE = ',11,' : There will be surface',
0649         # ' evaporation')
0650
0651 C-----
0652 C      Input FPET (NFHOUR = 1) or calculate FPET (NFHOUR = 2)
0653 C-----
0654         IF (NFHOUR .EQ. 1) THEN
0655         READ(LUI,1006) (FPET(I),I=1,24)
0656         IF (IPRINT .GT. 1) WRITE(LUW,7120)
0657         ELSE IF (NFHOUR .EQ. 2) THEN
0658         CALL MYHRLY(FPET)
0659         IF (IPRINT .GT. 1) WRITE(LUW,7130)
0660         ELSE
0661         WRITE(LUW,7140) NFHOUR
0662         STOP
0663         ENDIF
0664         SUM = 0.0
0665         DO 710 I=1,24
0666         SUM = SUM+FPET(I)
0667 710 CONTINUE
0668         IF (SUM .LT. 0.999 .OR. SUM .GT. 1.001) THEN
0669         WRITE(LUW,7150) SUM
0670         WRITE(LUW,7160) (FPET(I),I=1,24)
0671         STOP
0672         ELSE
0673         DO 720 I=1,24

```

```

0674         FPET(I) = FPET(I)/SUM
0675 720      CONTINUE
0676         IF (IPRINT .GT. 1) THEN
0677             WRITE(LUM,7160) (FPET(I),I=1,24)
0678             WRITE(LUM,1004)
0679         ENDIF
0680     ENDIF
0681 7150     FORMAT(' ERROR: Sum of diurnal PET distribution factors is '
0682 #         ,F6.4,'. It should be 1.0')
0683
0684     IF (IPLANT .EQ. 0) THEN
0685 C-----
0686 C     IPLANT = 0    No plants, read in PEVAP0
0687 C-----
0688         READ(LU1,1006) (PEVAP0(I),I=1,NDAYS)
0689         IF (IPRINT .GT. 1) THEN
0690             WRITE(LUM,7170)
0691             WRITE(LUM,7180) (I,PEVAP0(I),I=1,NDAYS)
0692         ENDIF
0693         IF (IPRINT .GT. 0) WRITE(LUM,1004)
0694         GO TO 700
0695     ENDIF
0696 7170     FORMAT(5(' DAY PEVAP0 ')/5(' --- ----- '))
0697 7180     FORMAT(5(I4,F8.4,3X))
0698
0699     IF (NFPET .EQ. 1) THEN
0700 C-----
0701 C     NFPET = 1    Ritchie method of partitioning PET
0702 C-----
0703         READ (LU1,1006) (PET(I),I=1,NDAYS)
0704         IF (IPRINT .GT. 1) THEN
0705             WRITE(LUM,7200)
0706             WRITE(LUM,7210)
0707         ENDIF
0708         DO 730 I=1, NDAYS
0709             ALAI = DLAI(I)
0710             PETT = PET(I)
0711             PTRANS(I) = PETT*(-.21+0.70*SORT(ALAI))
0712             IF (ALAI .LT. 0.1) PTRANS(I) = 0.0
0713             IF (ALAI .GT. 2.7) PTRANS(I) = PETT
0714             PEVAP0(I) = PETT-PTRANS(I)
0715             TOTPET = TOTPET+PETT
0716             TOTPTR = TOTPTR+PTRANS(I)
0717             TOTPEV = TOTPEV+PEVAP0(I)
0718 730      CONTINUE
0719         IF (IPRINT .GT. 0) THEN
0720             IF (IPRINT .GT. 1) WRITE(LUM,7220) (IDAY,PET(IDAY),
0721 #                 PTRANS(IDAY),PEVAP0(IDAY),IDAY=1,NDAYS)
0722             WRITE(LUM,7230) TOTPET,TOTPTR,TOTPEV
0723             WRITE(LUM,1004)
0724         ENDIF
0725     ELSE IF (NFPET .EQ. 2) THEN
0726 C-----
0727 C     NFPET = 2    Daily PET and Hinds transpiration function to define
0728 C     PT and PE from PET (relation found by Ted Hinds).
0729 C     In addition, the user can scale the data according
0730 C     to biomass (BIOMAS). The base biomass is 220 g/m**2
0731 C-----
0732         IF (NSOW .GE. 90 .AND. NSOW .LE. NHRVST)
0733             #     WRITE(LUS,7235) NSOW
0734             IF (NHRVST .LT. 152 .OR. NHRVST .GT. 365)
0735                 #     WRITE(LUS,7236) NHRVST
0736             IF (BIOMAS .EQ. 0) BIOMAS = 220.
0737             IF (IPRINT .GT. 1) THEN
0738                 WRITE(LUM,7240) BIOMAS,BARE
0739             ENDIF
0740

```



```

0741      WRITE(LUW,7210)
0742      ENDIF
0743      READ (LUI,1006) (PET(I),I=1,NDAYS)
0744      DO 740 IDAY=1,NDAYS
0745          IF (NSOW .LE. IDAY) THEN
0746              NGROWD = IDAY-NSOW+1
0747          ELSE
0748              NGROWD = 365+IDAY-NSOW+1
0749          ENDIF
0750          IF (NGROWD .GT. 0 .AND. NGROWD .LE. NHRVST) THEN
0751              RATIO = 0.300
0752              IF (NSOW .LT. 90) THEN
0753                  IGROWD = 90-NSOW
0754                  RATIO = RATIO*NGROWD/IGROWD
0755              ELSE IF (NSOW .GT. 151) THEN
0756                  IGROWD = 365+90-NSOW
0757              ENDIF
0758              RATIO = MIN(1.0,RATIO*BIOMAS/220.0)
0759              PTRANS(IDAY) = (1.-BARE)*RATIO*PET(IDAY)
0760              IF (NGROWD .LT. 5) PTRANS(IDAY) = 0.0
0761              PEVAPO(IDAY) = PET(IDAY)-PTRANS(IDAY)
0762          ELSE
0763              PEVAPO(IDAY) = PET(IDAY)
0764          ENDIF
0765          TOTPET = TOTPET+PET(IDAY)
0766          TOTPTR = TOTPTR+PTRANS(IDAY)
0767          TOTPEV = TOTPEV+PEVAPO(IDAY)
0768      740 CONTINUE
0769          IF (IPRINT .GT. 0) THEN
0770              IF (IPRINT .GT. 1) WRITE(LUW,7220) (IDAY,PET(IDAY),
0771              # PTRANS(IDAY),PEVAPO(IDAY),IDAY=1,NDAYS)
0772              WRITE(LUW,7230) TOTPET,TOTPTR,TOTPEV
0773              WRITE(LUW,1004)
0774          ENDIF
0775          ELSE
0776              WRITE(LUS,7250) NFPET
0777              IF (LUSLUW) WRITE(LUW,7250) NFPET
0778              STOP
0779          ENDIF
0780
0781      700 IF (LOWER .EQ. 3) STOP ' LOWER = 3 option is inoperative'
0782      IF (LOWER .EQ. 4) THEN
0783
0784      C-----
0785      C LOWER = 4: Flux at lower boundary. NTLEAK,NQLEAK,QLEAK
0786      C-----
0787      READ (LUI,7300) NTLEAK,(NQLEAK(I),QLEAK(I),I=1,NTLEAK)
0788      IF (IPRINT .GT. 1) THEN
0789          WRITE(LUW,7310)
0790          WRITE(LUW,7320) NTLEAK,(NQLEAK(I),QLEAK(I),I=1,NTLEAK)
0791      ENDIF
0792      IF (NTLEAK .GT. 50) STOP ' NQLEAK,QLEAK dimensions exceeded'
0793      JJ = 1
0794      DO 750 I=1,NDAYS
0795          IF (I .GT. NQLEAK(JJ)) JJ = JJ+1
0796          J = JJ-1
0797          WQLEAK(I) = QLEAK(J)+(QLEAK(JJ)-QLEAK(J))*(1-NQLEAK(JJ))/
0798          # (NQLEAK(JJ)-NQLEAK(J))
0799      750 CONTINUE
0800      IF (IPRINT .GT. 2) THEN
0801          WRITE(LUW,7330)
0802          WRITE(LUW,7320) (I,WQLEAK(I),I=1,NDAYS)
0803          WRITE(LUW,1004)
0804      ENDIF
0805      ELSE IF (LOWER .LT. 1 .OR. LOWER .GT. 5) THEN
0806          WRITE(LUS,7340) LOWER
0807          IF (LUSLUW) WRITE(LUW,7340) LOWER

```

```

0808         NSTOP = 1
0809     ENDIF
0810
0811 C-----
0812 C   NWATER
0813 C-----
0814     READ (LUI,1001) NWATER
0815     IF (IPRINT .GT. 0 .AND. NWATER .EQ. 0)
0816     #   WRITE(LUW,*) ' NWATER = 0 means there is no rain/irrigation'
0817
0818     IF (NSTOP .NE. 0) THEN
0819         WRITE(LUS,7400)
0820         IF (LUSLUW) WRITE(LUW,7400)
0821         STOP
0822     ENDIF
0823 7400 FORMAT('/! Program stopped due to input error ** See output!')
0824
0825 C-----
0826 C   Write binary file for input to UNSAT-H
0827 C-----
0828     FILE(NCHR:NCHR+3) = '.BIN'
0829     OPEN(UNIT=LUB,NAME=FILE,TYPE='NEW',FORM='UNFORMATTED')
0830     WRITE(LUB) IPLANT,LOWER,NDAYS,NDAY,NPRINT,ITOPBC,HTOP,
0831     #       KOPT,KEST,UP,DOWN,IVAPOR,DAYEND,NYEARS,ISWDIF,
0832     #       MATN,NPT,NGRAV,MAXSUB,MAXPOL,NSURPE,NSOW,NHRVST,
0833     #       INC,MXROOT,NWATER
0834     WRITE(LUB) DMAXBA,DELMAX,DELMIN,RAINIF,RFACT,HIRR1,HDRY,
0835     #       THMAX,SATURK,DRYK,SATURC,DRYC,AA,B1,B2,
0836     #       TMOIST,LOGE,SLOPTH,SLOPKH
0837     WRITE(LUB) TITLE,Z,H,NROOT,THETA,THETAD,THETAN,RDF,FPET,THETA,
0838     #       ZK,C,MGR,VAPCOE,TSOIL,PTRANS,PEVAPO,WQLEAK
0839     WRITE(LUB) MAT,IHY
0840     IF (NWATER .EQ. 0) GO TO 760
0841
0842 C-----
0843 C   Water input:  IRDAY,IRTPY,NP,EFICEN,TIME,AMOUNT
0844 C-----
0845     NSTOP = 0
0846     IF (IPRINT .GT. 1) WRITE(LUW,7410) NWATER
0847     PIRDAY = 0
0848     DO 770 I=1,NWATER
0849         READ(LUI,7420,END=775) IRDAY,IRTYPE,NP,EFICEN,
0850         #       (RTIME(J),AMOUNT(J),J=1,NP)
0851         IF (EFICEN .EQ. 0.0) EFICEN = 1.0
0852         DO 780 J=1,NP-1
0853             TOTALR = TOTALR+AMOUNT(J)
0854             IF (RTIME(J+1) .LE. RTIME(J) .AND. IRTYPE .NE. 4) THEN
0855                 WRITE(LUS,7430) IRDAY
0856                 IF (LUSLUW) WRITE(LUW,7430) IRDAY
0857                 NSTOP = 1
0858             ENDIF
0859 780     CONTINUE
0860         IF (IRDAY .LE. PIRDAY) THEN
0861             WRITE(LUS,7440) IRDAY
0862             IF (LUSLUW) WRITE(LUW,7440) IRDAY
0863             NSTOP = 1
0864         ENDIF
0865         PIRDAY = IRDAY
0866         IF (IPRINT .GT. 1) THEN
0867             WRITE(LUW,7450) IRDAY,RTIME(1),AMOUNT(1),IRTYPE,EFICEN,NP
0868             WRITE(LUW,7460) (RTIME(J),AMOUNT(J),J=2,NP)
0869         ENDIF
0870         IF (NSTOP .EQ. 0) WRITE(LUB) IRDAY,IRTYPE,EFICEN,NP,
0871         #       (RTIME(J),AMOUNT(J),J=1,NP)
0872 770     CONTINUE
0873         IF (N .NE. NP) THEN
0874 775     WRITE(LUS,7470) NWATER,I-1

```

```

0875      IF (LUSLUW) WRITE(LUW,7470) NWATER,1-1
0876      ENDIF
0877 7470 FORMAT(/' ERROR: NWATER is ',13,' but only ',13,' events read')
0878      IF (IPRINT .GT. 0) THEN
0879          WRITE(LUW,7480) TOTALR
0880      ENDIF
0881      IF (NSTOP .EQ. 1) THEN
0882          WRITE(LUS,7490)
0883          IF (LUSLUW) WRITE(LUW,7490)
0884      ENDIF
0885 7430 FORMAT(' ERROR: Time of rain/irrigation in ascending order on',/
0886      # ' day',14)
0887 7440 FORMAT(' ERROR: Day of rain/irrigation not in ascending order',/
0888      # ' on day',14)
0889 7490 FORMAT(' ERROR: Rain data not written to the binary file')
0890
0891 760 CLOSE (UNIT=1)
0892
0893 C-----
0894 C      Formats
0895 C-----
0896 1000 FORMAT(A80)
0897 1001 FORMAT(10I5)
0898 1002 FORMAT(3I5,F5.0)
0899 1003 FORMAT(8F10.0)
0900 1004 FORMAT(/80(' ')/)
0901 1006 FORMAT(8F7.0)
0902 2000 FORMAT(80(' ')/,1X,A80/,80(' ')/)
0903 5000 FORMAT(15/(4F14.0))
0904 5030 FORMAT(/' Total Initial Moisture Content = ',F7.3,' cm')
0905 6010 FORMAT(15/(5(15,F5.0)))
0906 6020 FORMAT(/
0907      # ' Total no. of data points growth days-LAI',15/,10X,
0908      # ' Growth Days',10X,' Leaf Area Index',100(15X,15,15X,F10.3))
0909 6030 FORMAT(6(' DAY',15,' LAI',F5.3))
0910 6040 FORMAT(/' ERROR: Current LEAF options are 0 and 1. Input value',
0911      # ' was ',12/)
0912 6050 FORMAT(/' ERROR: Current NROOT options are 1. Input',
0913      # ' value was ',12)
0914 6060 FORMAT(80(' ')/' NROOT = 1:',
0915      # ' Negative exponential representation of root growth'//
0916      # ' AA (Intersection of the curve at z=0 with abscissa) = ',
0917      # ' F10.3,/
0918      # ' B1 (coefficient defining degree of curvature) = ',
0919      # ' F10.5,/
0920      # ' B2 (coefficient that determines the value of asymptote = ',
0921      # ' F10.3)
0922 6080 FORMAT(T4,'DAY',T14,'MAX',T25,'ROOT',T33,'NORMALIZED',/
0923      # ' T10,'ROOT DEPTH',T23,'DENSITY',T35,'DENSITY',/
0924      # ' T23,'(cm/cm)',T35,'(1/cm)',/
0925      # ' T4,'---',T10,'-----',T23,'-----',T35,'-----')
0926 6090 FORMAT(T4,13,T12,F7.2,T20,F9.3,T34,F7.4)
0927 6110 FORMAT(' NUPTAK = 1:',
0928      # ' Feddes moisture dependent sink term')
0929 6120 FORMAT(/' For Material No. ',11,/,
0930      # ' THETAW (wilting point moisture content) = ',F10.4/,
0931      # ' THETAD (lower limit of optimum moisture content) = ',F10.4/,
0932      # ' THETAN (upper limit of optimum moisture content) = ',F10.4)
0933 6130 FORMAT(/' ERROR: Current NUPTAK options are 1. Input',
0934      # ' value was ',12)
0935 7120 FORMAT(' NFHOUR = 1: Hourly PET distribution values',
0936      # ' are entered by the user')
0937 7130 FORMAT(' NFHOUR = 2: User subroutine for hourly',
0938      # ' PET distribution')
0939 7140 FORMAT(/' ERROR: Current NFHOUR options are 1 and 2. Input',
0940      # ' value was ',12)
0941 7160 FORMAT(3(/8(3X,F6.4))

```

```

0942 7200 FORMAT(' NFPET = 1: ',/
0943 # ' PET is partitioned into PT and PE according'//
0944 # ' to the relationship developed by Ritchie ' /
0945 # ' (1972)'//)
0946 7210 FORMAT(2(6X,'DAY',4X,'PET',3X,'PTRANS',2X,'PEVAP0'),/,
0947 # 2(6X,'-----',2X,'-----',2X,'-----',2X,'-----'))
0948 7220 FORMAT(2(6X,I3,3F8.4))
0949 7235 FORMAT(' ERROR: NSOW was ',I3,' which is between 90 and NHRVST'//)
0950 7236 FORMAT(' ERROR: NHRVST was ',I3,' which is less than 151'//)
0951 7230 FORMAT(/' Totals: PET = ',F8.4,/,
0952 # ' PTRANS = ',F8.4,/,
0953 # ' PEVAP0 = ',F8.4)
0954 7240 FORMAT(' NFPET = 2: PET is partitioned into PT and PE according'//
0955 # ' to the relationship developed by Ted Hinds'//
0956 # ' Biomass, g/m**2 (BIOMAS) = ',F7.2,/,
0957 # ' Day of max. Transp. (TRMAX) = ',F7.2,/,
0958 # ' Bare soil fraction (BARE) = ',F7.2,/)
0959 7250 FORMAT(/' ERROR: Current NFPET options are 1 and 2. Input',
0960 # ' value was ',I2/)
0961 7300 FORMAT(15/,100(5(15,F5.0)))
0962 7310 FORMAT(' LOWER = 4: Bottom boundary fluxes (cm/hour)'//,
0963 # ' NQLEAK = ',I2,' flux changes'//,
0964 # ' Input Values (cm/day)'//)
0965 7320 FORMAT(5(' Day Flux '),5(' --- --- '),/(5(14,F8.4)))
0966 7330 FORMAT(' Output Values (cm/day) interpolated from input values'//)
0967 7340 FORMAT(/' ERROR: Current LOWER options are 1,2,4, and 5. ',
0968 # ' Input value was ',I2)
0969 7410 FORMAT(' NWATER (number of days of rain/irrigation) = '
0970 # ' 13,/,T17,'Rainfall/Irrigation Details'//,
0971 # ' T2,'DAY',T9,'TIME',T16,'AMOUNT',T25,'APPLICATION',
0972 # ' T39,'EFFICIENCY',T51,'CHANGES IN',/,T9,'(HR)',T17,'(CM)',
0973 # ' T29,'TYPE',T52,'RATE/HEAD',/,T2,'---',T9,'---',T16,'---',
0974 # ' T25,'---',T39,'---',T51,'---')
0975 7420 FORMAT(3I5,F6.0/,(2F7.0))
0976 7450 FORMAT(T2,I3,T7,F6.3,T16,F6.3,T30,I1,T41,F5.3,T55,I2)
0977 7460 FORMAT(T7,F6.3,T17,F5.3)
0978 7480 FORMAT(/' Water App. = ',F7.3,' cm')
0979 STOP
0980 END
0001
0002
0003 SUBROUTINE MYHRLY(FPT)
0004 C-----
0005 C Calculates hourly distribution of PET based on a sine wave
0006 C-----
0007 DIMENSION FPT(24)
0008 DATA P1/3.14159265/
0009 TPI = 2.*P1/24.
0010 DO 100 IHOUR=1,24
0011 IF (IHOUR .LT. 7 .OR. IHOUR .GE. 19) THEN
0012 FPT(IHOUR) = 0.01
0013 ELSE
0014 TEMP = -0.5*COS(TPI*(IHOUR-6))+0.5*COS(TPI*(IHOUR-7))
0015 FPT(IHOUR) = 0.88*TEMP
0016 ENDIF
0017 100 CONTINUE
0018 RETURN
0019 END
0001 SUBROUTINE POLYCH(NPTS,C,H)
0002 C-----
0003 C Latest Update 30-AUG-1985 by MJFayer
0004 C
0005 C This subroutine calculates the specific moisture capacity (C)
0006 C of each node as a function of the head value (H) for that node
0007 C-----
0008
0009 INCLUDE 'BYAR,INC/NOLIST'

```

```

0040      INCLUDE 'BSOIL.INC/NOLIST'
0057
0058      DIMENSION C(NPTS),H(NPTS)
0059      WCOMP = -4.707E-8
0060
0061      DO 1000 I=1,NPTS
0062          M = MAT(I)
0063          X = H(I)
0064          IF (X .LT. AIRINT(M)) THEN
0065              IF (X .LE. 0.0) THEN
0066                  C(I) = WCOMP
0067              ELSE
0068                  C(I) = SLOPTH(M)
0069              ENDIF
0070              GO TO 1000
0071          ENDIF
0072          IF (X .GT. HTRY) X = HTRY
0073          GO TO (100,200,300,400), KOPT
0074
0075      C-----
0076      C      KOPT = 1: Polynomial function
0077      C-----
0078      100      XX = X
0079              IPREV = (M-1)*MAXSUB
0080              IMAX = IPREV + NSUBTH(M)
0081              J = IPREV+1
0082      110      IF (X .GT. XDIVTH(J) .AND. J .LT. IMAX) THEN
0083                  J = J+1
0084                  GO TO 110
0085              ENDIF
0086              ID = J
0087              IP = (M-1)*INC + (J-IPREV-1)*MAXPOL
0088              X = LOG10(X)
0089              X0 = X
0090              SUM = CREGTH(IP+2)
0091              IF(NDEGTH(ID) .LE. 2) GO TO 130
0092              DO 120 K=3,NDEGTH(ID)
0093                  SUM = SUM + (K-1)*CREGTH(IP+K)*X
0094                  X = X*X0
0095      120          CONTINUE
0096      130          C(I) = LOGE*SUM/XX
0097              GO TO 500
0098
0099      C-----
0100      C      KOPT = 2: Haverkamp description
0101      C-----
0102      C-----
0103      200      IP = (M-1)*INC
0104              THTR = CREGTH(IP+1)
0105              ALPHA = CREGTH(IP+2)
0106              BETA = CREGTH(IP+3)
0107              XBETA = X*BETA
0108              XBM1 = X*(BETA-1)
0109              C(I) = -XBM1*ALPHA*BETA*(THET(M)-THTR)/((ALPHA+XBETA)
0110              #      *(ALPHA+XBETA))
0111              GO TO 500
0112
0113      C-----
0114      C      KOPT = 3: Campbell description
0115      C-----
0116      C-----
0117      300      IP = (M-1)*INC
0118              B = CREGTH(IP+1)
0119              TEM = (X/AIRINT(M))**(-1-(1/B))
0120              C(I) = -TEM*THET(M)/(B*AIRINT(M))
0121              GO TO 500
0122      C-----

```

```

0123 C      KOPT = 4: Haverkamp function, LOG-based
0124 C-----
0125 400    IP = (M-1)*INC
0126        THTR = CREGTH(IP+1)
0127        ALPHA = CREGTH(IP+2)
0128        BETA = CREGTH(IP+3)
0129        XBETA = LOG(X)**BETA
0130        XBM1 = LOG(X)**(BETA-1)
0131        C(I) = -XBM1*ALPHA*BETA*(THET(M)-THTR)/(X*(ALPHA+XBETA)**2)
0132
0133 500    IF (C(I) .GT. 0.0) THEN
0134        IERROR = IERROR + 1
0135        WRITE(LUS,6000)M,-C(I),H(I),AIRINT(M)
0136    ENDIF
0137
0138 1000   CONTINUE
0139        IF (IERROR .GT. 0) THEN
0140        WRITE(LUS,6010) IERROR
0141        STOP
0142    ENDIF
0143
0144    RETURN
0145
0146 C-----
0147 C      FORMAT STATEMENTS
0148 C-----
0149 6000   FORMAT(' ERROR: Capacity term negative, MAT No. ',I1,' C = ',
0150 #       G12.5,' H = ',G12.5,' AIRINT = ',G12.5)
0151 6010   FORMAT(' Program execution stopped in subroutine POLYCH  '/
0152 #       ' because of negative capacity term(s) for ',I3,' nodes')
0153
0154    END
0001    SUBROUTINE POLYKH(NPTS,ZK,H)
0002 C-----
0003 C      Latest Update 30-AUG-1985 by MJFayer
0004 C
0005 C      This subroutine calculates the conductivity (ZK) of each node
0006 C      as a function of the head value (H) for that node
0007 C-----
0008
0009    INCLUDE 'BVAR.INC/NOLIST'
0040    INCLUDE 'BSOIL.INC/NOLIST'
0057
0058    DIMENSION ZK(NPTS),H(NPTS)
0059
0060    DO 1000 I=1,NPTS
0061        M = MAT(I)
0062        X = H(I)
0063        IF (X .LT. AIRINT(M)) THEN
0064            IF (X .LE. 0.0) THEN
0065                ZK(I) = SK(M)
0066            ELSE
0067                ZK(I) = SLOPKH(M)*X+SK(M)
0068            ENDIF
0069        GO TO 1000
0070    ENDIF
0071    IF (X .GT. HDRY) X = HORY
0072    GO TO (100,200,300,200), KOPT
0073
0074 C-----
0075 C      KOPT = 1: Polynomial function
0076 C-----
0077 100    IPREV = (M-1)*MAXSUB
0078        IMAX = IPREV + NSUBKH(M)
0079        J = IPREV+1
0080 110    IF (X .GT. XD1YKH(J) .AND. J .LT. IMAX) THEN
0081        J = J+1

```

```

0082      GO TO 110
0083      ENDIF
0084      ID = J
0085      IP = (M-1)*INC + (J-1PREV-1)*MAXPOL
0086      X = LOG10(X)
0087      X0 = X
0088      SUM = CREGKH(IP+1)
0089      DO 120 K = 2,NDEGKH(ID)
0090          SUM = SUM + CREGKH(IP+K)*X
0091          X = X*X0
0092 120    CONTINUE
0093      ZK(1) = 10.**SUM
0094      GO TO 1000
0095
0096  C-----
0097  C      KOPT = 2,4: Haverkamp conductivity function
0098  C-----
0099 200    IP = (M-1)*INC
0100        A = CREGKH(IP+1)
0101        B = CREGKH(IP+2)
0102        ZK(1) = SK(M)*A/(A+X**B)
0103        GO TO 1000
0104
0105  C-----
0106  C      KOPT = 3: Campbell conductivity function
0107  C-----
0108 300    IP = (M-1)*INC
0109        B = CREGKH(IP+1)
0110        ZK(1) = SK(M) * (AIRINT(M)/X)**(2.+(3./B))
0111 1000   CONTINUE
0112       RETURN
0113       END
0001     SUBROUTINE POLYTH(NPTS,THETA,H)
0002  C-----
0003  C      Latest Update 30-AUG-1985 by MJFayer
0004  C
0005  C      This subroutine calculates the moisture content (THETA) of each
0006  C      node as a function of the head (H) value for that node
0007  C-----
0008
0009      INCLUDE 'BVAR,INC/NOLIST'
0040      INCLUDE 'BSOIL,INC/NOLIST'
0057
0058      DIMENSION THETA(NPTS),H(NPTS)
0059
0060      DO 1000 I=1,NPTS
0061          M = MAT(I)
0062          X = H(I)
0063          IF (X .LT. AIRINT(M)) THEN
0064              IF (X .LE. 0.0) THEN
0065                  THETA(I) = THET(M)
0066              ELSE
0067                  THETA(I) = SLOPTH(M)*X+THET(M)
0068              ENDIF
0069              GO TO 1000
0070          ENDIF
0071          IF (X .GT. HDRY) X = HDRY
0072          GO TO (100,200,300,400), KOPT
0073
0074  C-----
0075  C      KOPT = 1: Polynomial description
0076  C-----
0077 100    IPREV = (M-1)*MAXSUB
0078          IMAX = IPREV + NSUBTH(M)
0079          J = IPREV+1
0080 110    IF (X .GT. XDIVTH(J) .AND. J .LT. IMAX) THEN
0081          J = J+1

```

```

0082          GO TO 110
0083      ENDIF
0084      ID = J
0085      IP = (M-1)*INC + (J-IPREV-1)*MAXPOL
0086      X = LOG10(X)
0087      X0 = X
0088      SUM = CREGTH(IP+1)
0089      DO 120 K = 2, NDEGTH(ID)
0090          SUM = SUM + CREGTH(IP+K)*X
0091          X = X*X0
0092 120      CONTINUE
0093      THETA(I) = SUM
0094      GO TO 1000
0095
0096  C-----
0097  C      KOPT = 2: Haverkamp THETA function
0098  C-----
0099 200      IP = (M-1)*INC
0100          THTR = CREGTH(IP+1)
0101          ALPHA = CREGTH(IP+2)
0102          BETA = CREGTH(IP+3)
0103          THETA(I) = ALPHA*(THET(M)-THTR)/(ALPHA+X**BETA) + THTR
0104          GO TO 1000
0105
0106  C-----
0107  C      KOPT = 3: Campbell THETA function
0108  C-----
0109 300      IP = (M-1)*INC
0110          B = CREGTH(IP+1)
0111          THETA(I) = THET(M) * (X/AIRINT(M))**(-1.0/B)
0112          GO TO 1000
0113
0114  C-----
0115  C      KOPT = 4: Haverkamp THETA function, LN based
0116  C-----
0117 400      IP = (M-1)*INC
0118          THTR = CREGTH(IP+1)
0119          ALPHA = CREGTH(IP+2)
0120          BETA = CREGTH(IP+3)
0121          THETA(I) = ALPHA*(THET(M)-THTR)/(ALPHA+(LOG(X)**BETA)) + THTR
0122
0123 1000      CONTINUE
0124          RETURN
0125          END
0001      PROGRAM UNSATH
0002  C-----
0003  C      Updated August 1985, MJF
0004  C
0005  C      *** Multiple Year Simulation ***
0006  C
0007  C      Units: 1 = Binary Input file *.BIN (from DATAINH)
0008  C              2 = Binary result file *.RES (UNSATH output)
0009  C              5 = Screen Input
0010  C              6 = Log output
0011  C-----
0012      CHARACTER*80 IFILE, OFILE, TITLE, STIME*8, SDATE*9
0013      LOGICAL CHEKEM
0014      INTEGER SUBSTP, DAYSTP, TSTP, UPPER, WINDEX
0015      INCLUDE 'BVAR.INC/NOLIST'
0046      INCLUDE 'BARRAY.INC/NOLIST'
0054      INCLUDE 'BTIME.INC/NOLIST'
0061      INCLUDE 'BSOIL.INC/NOLIST'
0078      REAL MGR, KV(M1), KLMID
0079      DIMENSION TTHETA(M1), ZKMID(M1), ZZKMID(M1), Q(M1), THICK(M1),
0080      #          ZZ(M1), A1(M1), A2(M1), A3(M1), ZY(M1), RATE(24),
0081      #          SUBQ(M1), DAYQ(M1), TQ(M1), SINK(M1), DAYSINK(M1),
0082      #          SUBSNK(M1), GMOD(M1), GGMOD(M1)

```



```

0083      CALL TIME(STIME)
0084      CALL DATE(SOATE)
0085
0086      C-----
0087      C      IFILE = Binary input filename
0088      C-----
0089      WRITE(LUS,*) ' Enter binary input filename',
0090      #           ' (program appends ".BIN")'
0091      READ(LUR,9,END=999) IFILE
0092      9      FORMAT(A80)
0093      NCHR = INOEX(IFILE,' ')
0094      IFILE(NCHR:NCHR+3) = '.BIN'
0095      WRITE(LUS,10) IFILE
0096      10     FORMAT(' Input filename: ',A<NCHR+3>,:T35,'Date: ',A9,T60,
0097      #         'Time: ',A8,/)
0098      OPEN(UNIT=LUI,NAME=IFILE,TYPE='OLD',FORM='UNFORMATTED',READONLY)
0099
0100
0101      C-----
0102      C      Read in the DATAINH input file
0103      C-----
0104      READ(LUI) IPLANT,LOWER,NDAYS,NDAY,NPRINT,ITOPBC,HTOP,
0105      #           KOPT,KEST,UP,DOWN,IVAPOR,DAYEND,NYEARS,ISWDF,
0106      #           MATN,NPT,NGRAV,MAXSUB,MAXPOL,NSURPE,NSOW,NHRVST,
0107      #           INC,MXROOT,NWATER
0108      READ(LUI) DMAXBA,DELMAX,DELMIN,RAINIF,RFACT,HIRRI,HORY,
0109      #           THMAX,SATURK,DRYK,SATURC,DRYC,AA,B1,B2,
0110      #           TMOIST,LOGE,SLOPTH,SLOPKH
0111      READ(LUI) TITLE,Z,H,NTROOT,THETAH,THETAD,THETAN,RDF,FPET,THETA,
0112      #           ZK,C,MGR,VAPCOE,TSOIL,PTRANS,PEVAPO,WLEAK
0113      READ(LUI) MAT,iHY
0114      IF (NWATER .GT. 0) THEN
0115          READ(LUI) IRDAY,IRTYPE,EFICEN,NRATES,
0116      #           (RTIME(J),AMOUNT(J),J=1,NRATES)
0117          WINDEX = 1
0118      ENDIF
0119
0120      OFILE = IFILE
0121      OFILE(NCHR:NCHR+3) = '.RES'
0122      IRL = MAX(30,(IPLANT+3)*NPT+15)
0123      140    OPEN(UNIT=LUB,NAME=OFILE,TYPE='NEW',FORM='UNFORMATTED',
0124      #         ACCESS='DIRECT',RECL=IRL,RECORDTYPE='FIXED')
0125      DELT = DELMIN
0126      G      = NGRAV
0127      NTOTAL = 24./DELMAX
0128      DELSUB = 24./NTOTAL
0129
0130      C-----
0131      C      Write initial conditions to binary output file
0132      C-----
0133      IDAY = 0
0134      HOUR = 0.0
0138      STORE = TMOIST
0139      TPREV = TMOIST
0140      WRITE(LUB,REC=1) NPT,IPLANT,(Z(I),I=1,NPT),DAYEND,NPRINT,NSURPE,
0141      #           NTOTAL,IFILE,SDATE,STIME
0142      IF (IPLANT .EQ. 0) THEN
0143          WRITE(LUB,REC=2) IDAY,HOUR,(H(J),THETA(J),DAYQ(J),J=1,NPT),
0144      #           DAYINF,DAYRAN,DAYE,DAYTRA,DAYRUN,TPREV,
0145      #           TMOIST,DAYSTP,PE,PT,DAYTIM
0146      ELSE
0147          WRITE(LUB,REC=2) IDAY,HOUR,(H(J),THETA(J),DAYQ(J),DAYSNK(J),
0148      #           J=1,NPT),DAYINF,DAYRAN,DAYE,DAYTRA,DAYRUN,
0149      #           TPREV,TMOIST,DAYSTP,PE,PT,DAYTIM
0150      ENDIF
0151      IREC = 2
0152      IF (ITOPBC .EQ. 1) H(1) = HTOP

```

```

0153
0154 C-----
0155 C   Storing initial values as previous values
0156 C   A double first letter implies "previous value" (IE., HH, TTHETA)
0157 C-----
0158       DO 182 I=1,NPT
0159           HH(I) = H(I)
0160           TTHETA(I) = THETA(I)
0161           ZZK(I) = ZK(I)
0162           CC(I) = C(I)
0163           GMOD(I) = 1.0
0164           GGMOD(I) = GMOD(I)
0165 182   CONTINUE
0166
0167       ZZ(1) = Z(2)-Z(1)
0168       THICK(1) = 0.5*ZZ(1)
0169       THICK(NPT) = 0.5*(Z(NPT)-Z(NPT-1))
0170       DO 184 I=2,NPT-1
0171           ZZ(I) = Z(I+1)-Z(I)
0172           THICK(I) = 0.5*(ZZ(I-1)+ZZ(I))
0173 184   CONTINUE
0174       ISTEP = 0
0175
0176 C-----
0177 C   IRTYPE = 4, constant ponding > 1 day,
0178 C-----
0179       IF (IRTYPE .EQ. 5) THEN
0180           NEWPON = RTIME(2)
0181           IF (NEWPON .EQ. 0) NEWPON = DAYEND
0182           HHPOND = AMOUNT(1)
0183           ITPOND = 2
0184       ENDIF
0185       IF (NDAY .EQ. 0) NDAY = 1
0186
0187 C-----
0188 C   If NDAY is greater than 1, go through the rain data until the
0189 C   appropriate starting day is reached
0190 C-----
0191       IF (NWATER .GT. 0 .AND. IRDAY .LT. NDAY) THEN
0192 188   IF (WINDEX .LT. NWATER) THEN
0193           READ(LUI) IRDAY,IRTYPE,EFICEN,NRATES,(RTIME(J),
0194           #           AMOUNT(J),J=1,NRATES)
0195           WINDEX = WINDEX+1
0196           GO TO 188
0197       ENDIF
0198       DO 190 I=NRATES+1,24
0199           RTIME(I) = 0.0
0200 190   CONTINUE
0201       ENDIF
0202
0203 C*****
0204 C           MULTIPLE YEAR LOOP
0205 C
0206 C   HEAD values are retained from end of previous year
0207 C   Precipitation data file is rewound and repeated each year
0208 C*****
0209       DO 32709 IYEAR=1,NYEARS
0210           IF (IYEAR.GT.1) THEN
0211               REWIND 1
0212               DO 200 I=1,4
0213                   READ(LUI)
0214 200   CONTINUE
0215               READ (LUI) IRDAY,IRTYPE,EFICEN,NRATES,
0216               #           (RTIME(J),AMOUNT(J),J=1,NRATES)
0217               NDAY = 1
0218               WINDEX = 1
0219               CLOSE(UNIT=LUB)

```

```

0220      STORE = TMOIST
0221      TPREV = TMOIST
0222      OPEN (UNIT=LUB,NAME=OFILE,TYPE='NEW',FORM='UNFORMATTED',
0223      #      ACCESS='DIRECT',RECL=IRL,RECORDTYPE='FIXED')
0224      WRITE(LUB,REC=1) NPT,IPLANT,(Z(I),I=1,NPT),DAYEND,NPRINT,NSURPE,
0225      #      NTOTAL,IFILE,SDATE,STIME
0226      IF (IPLANT.EQ. 0) THEN
0227      WRITE(LUB,REC=2) IDAY,HOUR,(H(J),THETA(J),DAYQ(J),
0228      #      J=1,NPT),DAYINF,DAYRAN,DAYE,DAYTRA,DAYRUN,
0229      #      TPREV,TMOIST,DAYSTP,PE,PT,DAYTIM
0230      ELSE
0231      WRITE(LUB,REC=2) IDAY,HOUR,(H(J),THETA(J),DAYQ(J),
0232      #      DAYSJK(J),J=1,NPT),DAYINF,DAYRAN,DAYE,DAYTRA,
0233      #      DAYRUN,TPREV,TMOIST,DAYSTP,PE,PT,DAYTIM
0234      ENDIF
0235      IREC = 2
0236      ENDIF
0237
0238      C-----
0239      C      Initialize yearly cumulatives
0240      C-----
0241      APLIED = 0.
0242      IDAYLS = NDAY-1
0243      TE = 0.
0244      TETRAN = 0.
0245      TINF = 0.
0246      TPE = 0.
0247      TPET = 0.
0248      TPT = 0.
0249      TRAIN = 0.
0250      TRUNOF = 0.
0251      TSTP = 0.
0252      TTIM = 0.
0253      TTIRRI = 0.
0254      TTRA = 0.
0255
0256      C*****
0257      C      DAY LOOP
0258      C
0259      C      IDAY = Day number
0260      C      NDAY = Starting day
0261      C      DAYEND = Ending day
0262      C*****
0263      DO 32706 IDAY=NDAY,DAYEND
0264      TPREV = TMOIST
0265      IF (LOWER.EQ. 4) QLAST = WQLEAK(IDAY)
0266      TOTALR = 0.
0267      IF (IRTYPE.NE. 4) THEN
0268      IIRRI = 0
0269      IRAIN = 0
0270      ENDIF
0271      ICHECK = 0
0272
0273      C-----
0274      C      Check for rain on current day and set times
0275      C
0276      C      ICHECK = 0 No water application
0277      C      = 1 Irrigation day and time of application > current SUBEND
0278      C      = 2 Irrigation day and TSTART is within current SUBEND
0279      C      = 3 Irrigation occurs on the current SUB time step
0280      C-----
0281      IF (IRDAY.NE. IDAY) GO TO 330
0282
0283      C-----
0284      C      ISATUK set to 1 to initiate a saturated initial condition for a
0285      C      rainfall event
0286      C-----

```

```

0287      ISATUK = 1
0288      ICHECK = 1
0289      IF (IRTYPE .EQ. 1) THEN
0290          IRAIN = 1
0291      ELSE
0292          IRR1 = 1
0293      ENDIF
0294      TSTART = RTIME(1)
0295      TSTOP = RTIME(NRATES)
0296      TEND = RTIME(2)
0297
0298      C-----
0299      C   Determines the water input rate (cm/h)
0300      C   IRTYPE = 1 is PPT, = 2 is IRR1
0301      C-----
0302      IF (IRTYPE .EQ. 1 .OR. IRTYPE .EQ. 2) THEN
0303          DO 320 I=1,NRATES-1
0304              TOTALR = TOTALR+AMOUNT(I)
0305              RATE(I) = AMOUNT(I)/(RTIME(I+1)-RTIME(I))
0306              IF (IRTYPE .EQ. 2) RATE(I) = EFICEN#RATE(I)
0307          CONTINUE
0308          APLIED = APLIED+TOTALR
0309      ELSE IF (IRTYPE .EQ. 3) THEN
0310          HPOND = AMOUNT(1)
0311      ELSE IF (IRTYPE .EQ. 4) THEN
0312          HPOND = AMOUNT(ITPOND)
0313          IHPOND = 1
0314          IF (IDAY .GT. NEWPON) THEN
0315              ITPOND = ITPOND+1
0316              NEWPON = RTIME(ITPOND)
0317          IF (NEWPON .EQ. 0) NEWPON = DAYEND
0318      ENDIF
0319  ENDIF
0320
0321      C-----
0322      C   If IPLANT = 0, PTRANS = 0 and PET becomes PEVAPO
0323      C   If IPLANT = 1, PET becomes PTRANS + PEVAPO
0324      C-----
0325      330  PE = PEVAPO(IDAY)
0326          IF (IPLANT .EQ. 0) THEN
0327              PET = PE
0328          ELSE
0329              PT = PTRANS(IDAY)
0330              PET = PE + PT
0331          ENDIF
0332
0333          IF (IPLANT .NE. 0) THEN
0334              IF (IDAY .GE. NSOW) THEN
0335                  NGROW = IDAY-NSOW+1
0336              ELSE
0337                  NGROW = 365+IDAY-NSOW+1
0338              ENDIF
0339              IF (NGROW .GT. NHRVST) GO TO 435
0340
0341      C-----
0342      C   Determine to which depth roots have gone
0343      C-----
0344          I = 1
0345      400  IF (NGROW .GT. NTROOT(I) .AND. I .LT. MXROOT) THEN
0346          I = I+1
0347          GO TO 400
0348      ENDIF
0349          MDEPTH = I
0350          IF (MDEPTH .EQ. 1) GO TO 435
0351
0352      C-----
0353      C   Estimate root density factor (RDF) at each node of root depth

```

```

0354 C-----
0355     ROOTS = 0.
0356     RDF(1) = 0.
0357     DO 410 I=2,MDEPTH
0358         RDF(I) = RHO(Z(I),AA,B1,B2)
0359         ROOTS = ROOTS+RDF(I)*THICK(I)
0360 410     CONTINUE
0361     IF (MDEPTH.GT.1.AND.MDEPTH.LT.NPT) THEN
0362         ROOTS = ROOTS-RDF(MDEPTH)*THICK(MDEPTH)*0.5
0363         RDF(MDEPTH) = 0.5*RDF(MDEPTH)
0364     ENDIF
0365     DO 420 I=1,MDEPTH
0366         RDF(I) = RDF(I)/ROOTS
0367 420     CONTINUE
0368     ENDIF
0369
0370 C-----
0371 C    Initialize daily cumulative totals
0372 C-----
0373 435     IR      = 1
0374         DAYSTP = 0
0375         DAYTIM = 0.0
0376         DAYE   = 0.0
0377         OAYPE  = 0.0
0378         DAYTRA = 0.0
0379         DAYPT  = 0.0
0380         DAYINF = 0.0
0381         DAYIRR = 0.0
0382         DAYRUN = 0.0
0383         DAYRAN = 0.0
0384         TPREV  = TMOIST
0385         DO 440 I=1,NPT
0386             DAYSNK(I) = 0.0
0387             DAYQ(I)   = 0.0
0388 440     CONTINUE
0389         PTSUB = 0.0
0390         PESUB = 0.0
0391
0392 C*****
0393 C          DELSUR LOOP
0394 C
0395 C    Program will remain in this loop until 1 day has been simulated.
0396 C*****
0397     DO 32672 N=1,NTOTAL
0398         SPREV = TMOIST
0399         IF (IPLANT.EQ.1.OR.NSURPE.EQ.1) THEN
0400             NHR = INT(1.0+(N-1)*24./NTOTAL)
0401             PTSUB = PT*FPET(NHR)
0402             PESUB = PE*FPET(NHR)
0403         ENDIF
0404
0405 C-----
0406 C    Initialize total values for each DELMAX (max time step)
0407 C-----
0408     SUBE   = 0.0
0409     SUBIRR = 0.0
0410     SUBINF = 0.0
0411     SUBRAN = 0.0
0412     SUBRUN = 0.0
0413     SUBSTP = 0
0414     SUBTIM = 0.0
0415     SUBTRA = 0.0
0416     DO 450 I=1,NPT
0417         SUBSNK(I) = 0.0
0418         SUBQ(I)   = 0.0
0419 450     CONTINUE
0420

```

```

0421 C-----
0422 C   Start of estimations for each minimum time step
0423 C   TDONE = TIME 0, SUBEND = TIME 1
0424 C-----
0425 C   TDONE = (N-1)*DELSUB
0426 C   SUBEND = TDONE+DELSUB
0427 C   IF (ICHECK.EQ. 1 .AND. TSTART.LT. SUBEND .AND.
0428 C   #   TSTART.GE. TDONE) ICHECK=2
0429
0430 C-----
0431 C   Time stepping for the DELSUB loop
0432 C   TLEFT = time left till either the end of the DELSUB period or
0433 C   the time of the next water application rate change
0434 C-----
0435 C   IF (ICHECK.EQ. 0 .OR. ICHECK.EQ. 1) THEN
0436 C     TLEFT = DELSUB
0437 C   ELSE IF (ICHECK.EQ. 2) THEN
0438 C     TLEFT = TSTART-TDONE
0439 C     IF (TLEFT.LT. 0.001) THEN
0440 C       ICHECK = 3
0441 C       GO TO 460
0442 C     ENDIF
0443 C   ELSE IF (ICHECK.EQ. 3) THEN
0444 460 C     TLEFT = AMIN1(TEND-TDONE,DELSUB)
0445 C     IF (TLEFT.LT. 0.001) THEN
0446 C
0447 C-----
0448 C   IR   = Index of rainfall events during one day
0449 C   NRATES = Number of rainfall events during one day
0450 C-----
0451 C     IR = IR+1
0452 C     IF (IR.GE. NRATES) THEN
0453 C       ICHECK = 0
0454 C       IRAIN = 0
0455 C       IRR1 = 0
0456 C       TLEFT = DELSUB
0457 C     ELSE
0458 C       TEND = RTIME(IR+1)
0459 C       TLEFT = AMIN1(TEND-TDONE,DELSUB)
0460 C     ENDIF
0461 C   ENDIF
0462 C   ENDIF
0463
0464 C-----
0465 C   Start of the DELSUB loop. The program remains here until DELSUB
0466 C   time has elapsed. (i.e. TSUB = DELSUB)
0467 C-----
0468 C   TSUB = 0.0
0469 32649 IF(.NOT.(DELSUB-TSUB.GT.MIN(.0001,DELMIN))) GO TO 32648
0470 C   IF (DELT.LT. DELMIN) DELT = DELMIN
0471 C   DELSVL = DELSAV
0472 C   DELSAV = DELT
0473 C   TSUB = TSUB+DELSAV
0474 C   ELTIME = TDONE+TSUB
0475 C   IRYES = 0
0476
0477 C-----
0478 C   UPPER = 0, Flux surface bc; UPPER = 1, constant head surface bc
0479 C-----
0480 C   UPPER = 0
0481 C   NSINK = 0
0482 C   IF (IPLANT.EQ. 1) NSINK = 1
0483 C   PRAIN = 0.0
0484 C   PIRRI = 0.0
0485
0486 C-----
0487 C   Surface flux initially set to the evaporation rate

```

```

0488 C-----
0489 SFLUX = -PESUB
0490 IF (IRRI .EQ. 1 .AND. IRTYPE .GT. 2) THEN
0491   IF (HPOND .GT. 0) THEN
0492     H(1) = -HPOND
0493     IRYES = 1
0494     UPPER = 1
0495   ELSE
0496     IRYES = 0
0497     HPOND = 0
0498     IRR1 = 0
0499   ENDIF
0500   GO TO 32647
0501 ELSE
0502 C-----
0503 C   During watering, surface evaporation set to zero
0504 C-----
0505   IF (.NOT.((IRAIN .NE. 0 .OR. IRR1 .NE. 0) .AND. ICHECK .GT. 0))
0506     # GO TO 32647
0507   ELPAST = ELTIME-DELSAV
0508 C-----
0509 C   Are we within a rainfall period or across a boundary of one?
0510 C-----
0511   CHEKEM = (ELPAST.GE. RTIME(1).AND.
0512     # ELTIME.LE. RTIME(NRATES)).OR. ICHECK.EQ.3
0513   IF (CHEKEM) THEN
0514 C-----
0515   ISATUK = 1 indicates the start of a rainfall event. Once the
0516   rain has started, ISATUK = 2
0517 C-----
0518   IF (ISATUK .EQ. 1 .AND. RATE(IR) .GT. 0.0) THEN
0519 C-----
0520 C   For the first time step of a rainfall event, C and K are set to
0521 C   their values at saturation and DELT is reduced by RAINIF
0522 C-----
0523   C(1) = SATURC
0524   ZK(1) = SATURK
0525   TSUB = TSUB-DELT
0526   DELT = DELT*RAINIF
0527   IF (DELT .LT. DELMIN) DELT = 1.5*DELMIN
0528   TSUB = TSUB+DELT
0529   ISATUK = 2
0530   ENDIF
0531   IF (IR .LT. NRATES .AND. RATE(IR) .GT. 0.) THEN
0532     NSINK = 0
0533     IF (IRRI .EQ. 1) PIRRI = RATE(IR)
0534     IF (IRAIN .EQ. 1) PRRAIN = RATE(IR)
0535     IRYES = 1
0536     SFLUX = RATE(IR)
0537   ENDIF
0538   ENDIF
0539   32647 IF (IPLANT .NE. 1) GO TO 32639
0540   TRANSP = 0.0
0541   DO 500 I=1,MDEPTH
0542     SINK(I) = 0.
0543   500 CONTINUE
0544 C** Transpiration is assumed to be zero during irrigation or rainfall
0545   IF (PTSUB .EQ. 0.0) NSINK = 0
0546 C-----
0547

```

```

0555 C Estimate sink term by a modified FEDDES method
0556 C-----
0557 IF (NSINK.NE. 1) GO TO 32639
0558 DO 520 I=1,MDEPTH
0559 MATI = MAT(I)
0560 TH = (THETA(I)+TTHETA(I))*0.5
0561 THW = THETA(MATI)
0562 THD = THETA(MATI)
0563 THN = THETA(MATI)
0564 IF (TH.GT. THW.AND. TH.LE. THN) THEN
0565 IF (TH.LT. THD) THEN
0566 ALPHAF = (TH-THW)/(THD-THW)
0567 ELSE
0568 ALPHAF = 1.0
0569 ENDIF
0570 ELSE
0571 ALPHAF = 0.0
0572 ENDIF
0573 SINK(I) = PTSUB*ALPHAF*RDF(I)
0574 TRANSP = TRANSP+SINK(I)*THICK(I)
0575 520 CONTINUE
0576
0577 C*****
0578 C CONVERGENCE LOOP FOR THE CURRENT TIME STEP
0579 C
0580 C*****
0581 32639 ISAVUP = UPPER
0582 IF (ITOPBC.EQ. 1) THEN
0583 UPPER = 1
0584 ISAVUP = UPPER
0585 ENDIF
0586 IF (IVAPOR.EQ. 1) THEN
0587 DO 145 I=1,NPT
0588 TH = (THETA(I)+TTHETA(I))*0.5
0589 KV(I) = VAPCOE*(THETA(MATI)-TH)*EXP(-H(I)*MGR/TSOIL)
0590 145 CONTINUE
0591 ENDIF
0592
0593 C** Start of the "decrease DELT" loop for times when the soil dries out
0594 C** or saturates, thus changing the surface boundary condition
0595
0596 150 DELSVL = DELSAV
0597 DELSAV = DELT
0598 NPPT = NPT-1
0599 160 IF (DELSAV.LE. 0.) STOP ' DELT=0. AT 160'
0600
0601 C*****
0602 C PREDICTOR-CORRECTOR LOOP (while "IN" < 2)
0603 C
0604 C This implicit scheme solves the partial differential equation
0605 C for the change in head
0606 C*****
0607 IN = 0
0608 32626 IF (.NOT.(IN.LT. 2)) GO TO 32625
0609 IN = IN+1
0610 DEL = 1./DELSAV
0611
0612 C-----
0613 C Calculate internodal conductivities
0614 C-----
0615 DO 560 I=1,NPPT
0616 IF (KEST.EQ. 1) THEN
0617 KLMID = (ZK(I)+ZK(I+1))*0.5
0618 ZKMID(I) = (ZK(I)+KV(I)+ZK(I+1)+KV(I+1))*0.5
0619 ELSE IF (KEST.EQ. 2) THEN
0620 IF (-(H(I)+Z(I)).GT. -(H(I+1)+Z(I+1))) THEN
0621 KLMID = ZK(I)*UP+ZK(I+1)*DOWN

```



```

0622      ZKMID(1) = (ZK(1)+KV(1))*UP + (ZK(1+1)+KV(1+1))*DOWN
0623      ELSE
0624      KLMID = ZK(1)*DOWN+ZK(1+1)*UP
0625      ZKMID(1) = (ZK(1)+KV(1))*DOWN + (ZK(1+1)+KV(1+1))*UP
0626      ENDIF
0627      ELSE IF (KEST .EQ. 3) THEN
0628      KLMID = (ZK(1)*ZK(1+1))*0.5
0629      ZKMID(1) = ((ZK(1)+KV(1)) * (ZK(1+1)+KV(1+1)))*0.5
0630      ENDIF
0631      IF (IVAPOR .EQ. 1) GMOD(1) = KLMID/ZKMID(1)
0632      IF (ISTEP .EQ. 0) THEN
0633      ZZKMID(1) = ZKMID(1)
0634      GGMOD(1) = GMOD(1)
0635      ENDIF
0636 560 CONTINUE
0637
0638      ISTEP = 1
0639      IF (ITOPBC .EQ. 1) THEN
0640      UPPER = 1
0641      H(1) = HTOP
0642      ENDIF
0643
0644  C-----
0645  C   Calculate coefficients for simulation of change in head
0646  C-----
0647      DO 570 I1=2,NPPT
0648      I = I1-1
0649      I11 = I1+1
0650      Z31 = Z(I11)-Z(I)
0651      FZZ1 = 1./(Z31*ZZ(I))
0652      FZZ2 = 1./(Z31*ZZ(I1))
0653      ZK1 = ZKMID(I)*FZZ1
0654      ZK2 = ZKMID(I1)*FZZ2
0655      ZZK1 = ZZKMID(I)*FZZ1
0656      ZZK2 = ZZKMID(I1)*FZZ2
0657      AVC = 0.5*(C(I1)+CC(I1))
0658      A2(I1) = AVC*DEL-ZK1-ZK2
0659      A1(I1) = ZK1
0660      A3(I1) = ZK2
0661      ZY(I1) = ZK1*(HH(I1)-HH(I)+ZZ(I)*G*GMOD(I))
0662      # -ZK2*(HH(I11)-HH(I1)+ZZ(I1)*G*GMOD(I1))
0663      # +AVC*HH(I1)*DEL-SINK(I1)+ZK1*ZZ(I)*G*GMOD(I)
0664      # -ZK2*ZZ(I1)*G*GMOD(I1)
0665 570 CONTINUE
0666      IF (UPPER .EQ. 1) THEN
0667
0668      C** Surface node equation for constant head b.c.
0669
0670      ISTART = 2
0671      ZY(2) = ZY(2)-A1(2)*H(1)
0672      A1(2) = 0.
0673      ELSE
0674
0675      C** Surface node equation for flux b.c.
0676
0677      Z1 = ZZ(1)
0678      Z11 = ZKMID(1)/(Z1*Z1)
0679      Z111 = ZZKMID(1)/(Z1*Z1)
0680      AVC = (C(1)+CC(1))*0.5
0681      A1(1) = 0.0
0682      A2(1) = AVC*DEL-Z11
0683      A3(1) = Z11
0684      ZY(1) = AVC*DEL*HH(1)-ZKMID(1)*G*GMOD(1)/Z1-Z111*(HH(2)-HH(1)
0685      # +Z1*G*GMOD(1))+2.*SFLUX/Z1-SINK(1)
0686      ISTART = 1
0687      ENDIF
0688

```

```

0689 C-----
0690 C   LOWER: 1=semi, 2=static, 4=dynamic, 5=impermeable
0691 C-----
0692 IF (LOWER .EQ. 1 .OR. LOWER .EQ. 4 .OR. LOWER .EQ. 5) THEN
0693   ZN = ZZ(NPPT)
0694   ZNN = ZKMID(NPPT)/(ZN*ZN)
0695   ZNNN = ZZKMID(NPPT)/(ZN*ZN)
0696   AVC = 0.5*(C(NPT)+CC(NPT))
0697   A3(NPT) = 0.0
0698   IF (LOWER .EQ. 1) THEN
0699     QLAST = 0.5*(ZK(NPT)+ZZK(NPT))
0700     A1(NPT) = ZNN
0701     A2(NPT) = AVC*DEL-ZNN
0702     ZY(NPT) = AVC*HH(NPT)*DEL
0703     #           +ZNNN*(HH(NPT)-HH(NPPT)+ZN*G*GMOD(NPPT))
0704     #           -2.*QLAST/ZN+ZKMID(NPPT)*G*GMOD(NPPT)/ZN
0705   ENDIF
0706   IF (LOWER .EQ. 4) THEN
0707     A1(NPT) = ZNN
0708     A2(NPT) = AVC*DEL-ZNN
0709     ZY(NPT) = AVC*HH(NPT)*DEL
0710     #           +ZNNN*(HH(NPT)-HH(NPPT)+ZN*G*GMOD(NPPT))
0711     #           -2.*QLAST/ZN+ZKMID(NPPT)*G*GMOD(NPPT)/ZN
0712   ENDIF
0713   IF (LOWER .EQ. 5) THEN
0714     QLAST = 0.
0715     A1(NPT) = ZNN
0716     A2(NPT) = AVC*DEL-ZNN
0717     ZY(NPT) = AVC*HH(NPT)*DEL
0718     #           +ZNNN*(HH(NPT)-HH(NPPT)+ZN*G*GMOD(NPPT))
0719     #           -2.*QLAST/ZN+ZKMID(NPPT)*G*GMOD(NPPT)/ZN
0720   ENDIF
0721   NPPT = NPT
0722 ENDIF
0723
0724 C-----
0725 C   Solution of tridiagonal matrix
0726 C-----
0727 CALL TRIDAG(ISTART,NPPT,A1,A2,A3,ZY,H)
0728
0729 C*****
0730 C   SOLUTION ESTIMATE COMPLETE FOR THIS ITERATION
0731 C*****
0732 NPPT = NPPT-1
0733
0734 C** Check whether surface b,c. changed due to solution with given
0735 C** flux. If so, repeat as constant surface head b,c.
0736
0737 IGO150 = 0
0738 IF (UPPER .EQ. 0) THEN
0739   IF (H(1) .LT. HIRRI) THEN
0740     H(1) = HIRRI
0741     UPPER = 1
0742     ZK(1) = SATURK
0743     C(1) = SATURC
0744     IGO150 = 1
0745   ENDIF
0746   IF (H(1) .GT. HDRY) THEN
0747     H(1) = HDRY
0748     ZK(1) = DRYK
0749     C(1) = DRYC
0750     UPPER = 1
0751     IGO150 = 1
0752   ENDIF
0753   IF (IGO150 .EQ. 1) THEN
0754     DO 782 I=2,NPT
0755       H(I) = HH(I)

```

```

0756 782      CONTINUE
0757      GO TO 150
0758      ENDIF
0759      ENDIF
0760
0761 C-----
0762 C      Endd of convergence check on upper boundary condition
0763 C      ==> Passed
0764 C
0765 C      Check whether results are less than HDRY. If so, set up for
0766 C      return to top of predictor-corrector loop
0767 C-----
0768      DO 784 I=1,NPT
0769          IF (H(I) .GT. HDRY) H(I) = HDRY
0770 784      CONTINUE
0771          IF (IN .EQ. 2) CALL POLYTH(NPT,THETA,H)
0772          IF (IN .EQ. 1) THEN
0773              CALL POLYKH(NPT,ZK,H)
0774              CALL POLYCH(NPT,C,H)
0775          ENDIF
0776          GO TO 32626
0777
0778 C*****
0779 C      End of the PREDICTOR-CORRECTOR LOOP
0780 C*****
0781
0782 C**      Check for convergence on THETA
0783
0784 32625 IF (ISWDIF .NE. 0) GO TO 32594
0785      DIFMAX = 0.0
0786      DO 788 I=1,NPT
0787          THMAX = MAX(THETA(I),TTHETA(I),0.01)
0788          THDIF = ABS((THETA(I)-TTHETA(I))/THMAX)
0789          DIFMAX = MAX(DIFMAX,THDIF)
0790 788      CONTINUE
0791      IGO150 = 0
0792
0793 C**      If time step reaches DELMIN, accept the % THETA change error,
0794 C**      even if it is greater than DMAXBA, and go on
0795
0796      IF (DELT .GT. DELMIN .OR. DIFMAX .LT. DMAXBA)
0797          # CALL DELCHK(IDAYLS)
0798          IF (IGO150 .EQ. 1) THEN
0799              DO 790 I=1,NPT
0800                  THETA(I) = TTHETA(I)
0801                  ZK(I) = ZZK(I)
0802                  C(I) = CC(I)
0803                  H(I) = HH(I)
0804 790      CONTINUE
0805                  IF (ISATUK .EQ. 2) THEN
0806                      ZK(I) = SATURK
0807                      C(I) = SATURC
0808                  ENDIF
0809                  UPPER = ISAVUP
0810                  GO TO 150
0811          ENDIF
0812
0813 C**      Fluxes are calculated at all column segments
0814
0815 32594 DO 792 I=2,NPT
0816      I = I-1
0817      ZI = ZZ(I)
0818      Q(I) = DELSAV*0.5*(ZKMID(I)*(H(I)-H(I)+ZI*G*GMOD(I))+
0819          # ZZKMID(I)*(HH(I)-HH(I)+ZI*G*GMOD(I)))/ZI
0820 792      CONTINUE
0821
0822 C-----

```

```

0823 C      Surface flux (SSFLUX)
0824 C      Estimate infiltration from THETA and geometry if saturated
0825 C      (i.e., UPPER .EQ. 1)
0826 C      If the infiltration was from rainfall, however, do not let it
0827 C      exceed the actual rainfall (i.e., ISAVUP .NE. 1)
0828 C-----
0829 C      IF (UPPER .EQ. 1) THEN
0830 C          SSFLUX = Q(1)+(THETA(1)-TTHETA(1))*THICK(1)
0831 C          IF (ISAVUP .NE. 1 .AND. H(1) .EQ. HIRRI)
0832 C              # SSFLUX = MIN(SSFLUX,SFLUX*DELSAV)
0833 C          IF (ISAVUP .NE. 1 .AND. H(1) .EQ. HDRY)
0834 C              # SSFLUX = MAX(SSFLUX,SFLUX*DELSAV)
0835 C          ELSE
0836 C              SSFLUX = SFLUX*DELSAV
0837 C          ENDIF
0838 C
0839 C**      Deep drainage flux
0840 C
0841 C      IF (LOWER .EQ. 2) THEN
0842 C          Q(NPT) = Q(NPPT)-(THETA(NPT)-TTHETA(NPT))*THICK(NPT)
0843 C      ELSE
0844 C          Q(NPT) = QLAST*DELSAV
0845 C      ENDIF
0846 C
0847 C-----
0848 C      Estimate total water storage in the profile (SMOIST) and the
0849 C      mass balance error (DIFFMA)
0850 C-----
0851 C      SMOIST = 0.5*(THETA(1)*ZZ(1)+THETA(NPT)*ZZ(NPT-1) )
0852 C      DO 794 I=2,NPT-1
0853 C          SMOIST = SMOIST+THETA(I)*0.5*(ZZ(I-1)+ZZ(I) )
0854 C      794 CONTINUE
0855 C      DIFFMA = ABS( (SMOIST+SSFLUX-TRANSP*DELSAV-Q(NPT)-SMOIST) )
0856 C
0857 C-----
0858 C      Reduce the time step if the mass balance error is not within
0859 C      limits (when ISWDIF .EQ. 1)
0860 C-----
0861 C      IF (ISWDIF .EQ. 1 .AND. DIFFMA .GT. DMAXBA) THEN
0862 C          IF (DELT .NE. DELMIN) THEN
0863 C              DELT = DELT*0.5
0864 C              DO 796 I=1,NPT
0865 C                  THETA(I) = TTHETA(I)
0866 C                  ZK(I) = ZZK(I)
0867 C                  C(I) = CC(I)
0868 C                  H(I) = HH(I)
0869 C      796 CONTINUE
0870 C          UPPER = ISAVUP
0871 C          IF (ISATUK .EQ. 2) THEN
0872 C              ZK(1) = SATURK
0873 C              C(1) = SATURC
0874 C          ENDIF
0875 C          TSUB = TSUB-DELT
0876 C          !GO150 = 1
0877 C          IF (DELT .LT. DELMIN) THEN
0878 C              TSUB = TSUB-DELT
0879 C              DELT = DELMIN
0880 C              TSUB = TSUB+DELMIN
0881 C              IF (IDAY .NE. IDAYLS) THEN
0882 C                  IDAYLS = IDAY
0883 C                  WRITE(LUS,250) IDAY,N,TSUB,DELT,DIFFMA
0884 C              ENDIF
0885 C          ENDIF
0886 C          GO TO 150
0887 C      ENDIF
0888 C      ENDIF
0889 C      250 FORMAT(' IDAY',14,' N',12,' TSUB',G10.3,' DELT',G10.3,

```

```

0890      #      ' DIFFMA',G10.3,/' DELT SMALLER THAN DELMIN ** ',
0891      #      ' PROGRAM CONTINUED WITH DELT=DELMIN' )
0892      IF ((ISWDIF.EQ. 1 .AND. DIFFMA.LT. DMAXBA*.25) .AND.
0893      #      DELT.LT. DELMAX) THEN
0894          DELT = DELT*RFACT
0895          IF (DELTA.GT. DELMAX) DELT = DELMAX
0896          CALL TTLEFT
0897          IF (TLEFT.LT. 10.*DELTA) THEN
0898              LEFT = TLEFT/DELTA+.49999
0899              IF (LEFT.LE. 1) THEN
0900                  DELT = TLEFT
0901              ELSE
0902                  DELT = TLEFT/LEFT
0903              ENDIF
0904          ENDIF
0905          IF (TLEFT.LE. 1.5*DELTA .OR. DELTA.GT. TLEFT) DELT = TLEFT
0906      ENDIF
0907
0908      C*****
0909      C      END OF CONVERGENCE LOOP
0910      C      Time step successful, all conditions satisfied. New values
0911      C      are made as previous values and hourly totals are updated
0912      C*****
0913      IF (ISATUK.EQ. 2) ISATUK = 0
0914      CALL POLYKH(NPT,ZK,H)
0915      CALL PDLYCH(NPT,C,H)
0916      SUBSTP = SUBSTP+1
0917      SUBTIM = SUBTIM+DELSAV
0918      DO 800 I=1,NPT
0919          SUBQ(I) = SUBQ(I)+Q(I)
0920      800 CONTINUE
0921      SUBTRA = SUBTRA+TRANSP*DELSAV
0922      DO 805 I=1,MDEPTH
0923          SUBSNK(I) = SUBSNK(I)+SINK(I)*DELSAV
0924      805 CONTINUE
0925      IF (SSFLUX.GT. 0.0) THEN
0926          SUBINF = SUBINF+SSFLUX
0927          IF (TTIM.EQ. 0) THEN
0928              IF (SUBSTP.EQ. 1 .AND. ITOPBC.EQ. 1)
0929              #          SUBINF = SUBINF+(THETA(1)-TTHETA(1))*THICK(1)
0930          ENDIF
0931      ENDIF
0932      IF (SSFLUX.LT. 0.0) SUBE = SUBE-SSFLUX
0933      IF ((IRR1.EQ. 1 .OR. IRAIN.EQ. 1) .AND. IRYES.EQ. 1) THEN
0934          SUBRAN = SUBRAN+DELSAV*PRAIN
0935          SUBIRR = SUBIRR+DELSAV*PIRRI
0936          RUNOFF = 0.0
0937          IF (PIRRI*DELSAV.GT. SSFLUX) RUNOFF = PIRRI*DELSAV-SSFLUX
0938          IF (PRAIN*DELSAV.GT. SSFLUX) RUNOFF = PRAIN*DELSAV-SSFLUX
0939          IF (RUNOFF.GT. 0.0) SUBRUN = SUBRUN+RUNOFF
0940      ENDIF
0941      DO 810 I=1,NPT
0942          TTHETA(I) = THETA(I)
0943          ZZK(I) = ZK(I)
0944          CC(I) = C(I)
0945          ZZKMID(I) = ZKMID(I)
0946          GGMOD(I) = GMOD(I)
0947      810 CONTINUE
0948
0949      C-----
0950      C      Extrapolate the HEAD for property estimation
0951      C-----
0952      DO 815 I=1,NPT
0953          HTMP = H(I)
0954          H(I) = HTMP+(HTMP-HH(I))*DELTA/DELSAV
0955          HH(I) = HTMP
0956      815 CONTINUE

```

```

0957      CALL POLYKH(NPT,ZK,H)
0958      CALL POLYCH(NPT,C,H)
0959      IF (IVAPOR.EQ. 1) CALL POLYTH(NPT,THETA,H)
0960      DO 820 I=1,NPT
0961          H(I) = HH(I)
0962      820  CONTINUE
0963      TMOIST = SMOIST
0964      GO TO 32649
0965
0966      C *****
0967      C      END OF THE DELSUB LOOP. (TSUB = DELSUB)
0968      C      Daily totals are updated
0969      C *****
0970
0971      32648  DAYE  = DAYE+SUBE
0972            DAYINF = DAYINF+SUBINF
0973            DAYIRR = DAYIRR+SUBIRR
0974            DAYPE  = DAYPE+PESUB
0975            DAYPT  = DAYPT+PTSUB
0976            DAYRAN = DAYRAN+SUBRAN
0977            DAYRUN  = DAYRUN+SUBRUN
0978            DAYSTP  = DAYSTP+SUBSTP
0979            DAYTIM  = DAYTIM+SUBTIM
0980            DAYTRA  = DAYTRA+SUBTRA
0981            SSFLUX  = SUBINF-SUBE
0982            DO 825 I=1,NPT
0983                DAYSNK(I) = DAYSNK(I)+SUBSNK(I)
0984                DAYQ(I)  = DAYQ(I)+SUBQ(I)
0985      825  CONTINUE
0986
0987      C-----
0988      C      Write DELSUB summary to binary output file
0989      C-----
0990      IF (NPRINT.EQ. 1) THEN
0991          IREC = IREC+1
0992          HOUR = N*DELMAX
0993          IF (IPLANT.EQ. 0) THEN
0994              WRITE(LUB,REC=IREC) IDAY,HOUR,(H(J),TTHETA(J),SUBQ(J),
0995              #                      J=1,NPT),SUBINF,SUBRAN,SUBE,SUBTRA,
0996              #                      SUBRUN,SPREV,TMOIST,SUBSTP,PESUB,PTSUB,SUBTIM
0997          ELSE
0998              WRITE(LUB,REC=IREC) IDAY,HOUR,(H(J),TTHETA(J),SUBQ(J),
0999              #                      SUBSNK(J),J=1,NPT),SUBINF,SUBRAN,SUBE,SUBTRA,
1000              #                      SUBRUN,SPREV,TMOIST,SUBSTP,PESUB,PTSUB,SUBTIM
1001          ENDIF
1002      ENDIF
1003
1004      C-----
1005      C      Estimate the average DELT for the next hour
1006      C-----
1007      DELT = MIN(DELSUB,2.0*DELSUB/SUBSTP)
1008      32672  CONTINUE
1009
1010      C*****
1011      C      END OF THE DELSUB LOOP FOR THIS DAY
1012      C*****
1013
1014      N = N-1
1015      HOUR = N*DELMAX
1016
1017      C-----
1018      C      Add daily total to seasonal total
1019      C-----
1020      858  TSTP  = TSTP+DAYSTP
1021            TTIM  = TTIM+DAYTIM
1022            TPE   = TPE+DAYPE
1023            TE    = TE+DAYE

```

```

1024      IF (DAYTRA .GT. 0.) TETRA = TETRA+DAYE
1025      TTIRRI = TTIRRI+DAYIRR
1026      TINF = TINF+DAYINF
1027      TRUNOF = TRUNOF+DAYRUN
1028      TTRA = TTRA+DAYTRA
1029      TPT = TPT+DAYPT
1030      TRAI N = TRAI N+DAYRAN
1031      TPET = TPET+PET
1032      DO 860 I=1,NPT
1033          TQ(I) = TQ(I)+DAYQ(I)
1034 860 CONTINUE
1035      IF (NWATER .GT. WINDEX .AND. IDAY .EQ. IRDAY) THEN
1036          READ(LUI) IRDAY,IRTYPE,EFICEN,NRATES,(RTIME(J),
1037          #          AMOUNT(J),J=1,NRATES)
1038          WINDEX = WINDEX+1
1039      ENOIF
1040
1041  C-----
1042  C   Write daily summary to binary output file
1043  C-----
1044      IREC = IREC+1
1045      IF (IPLANT .EQ. 0) THEN
1046          WRITE(LUB,REC=IREC) IDAY,HOUR,(H(J),THETA(J),DAYQ(J),
1047          #          J=1,NPT),DAYINF,DAYRAN,DAYE,DAYTRA,
1048          #          DAYRUN,TPREV,TMOIST,DAYSTP,PE,PT,DAYTIM
1049      ELSE
1050          WRITE(LUB,REC=IREC) IDAY,HOUR,(H(J),THETA(J),DAYQ(J),
1051          #          DAYSINK(J),J=1,NPT),DAYINF,DAYRAN,DAYE,DAYTRA,
1052          #          DAYRUN,TPREV,TMOIST,DAYSTP,PE,PT,DAYTIM
1053      ENDIF
1054 32706 CONTINUE
1055
1056  C*****
1057  C   End of the DAY LOOP for this year
1058  C*****
1059
1060  C-----
1061  C   Write final simulation summary to binary output file
1062  C-----
1063      TERR = STORE+TINF-TE-TTRA-TQ(NPT)-TMOIST
1064      IREC = IREC+1
1065      TTIM = TTIM/24.0
1066      WRITE(LUB,REC=IREC) IDAY,G,IPLANT,TPET,TPT,TTRA,TPE,TE,TETRA,
1067      #          TRUNOF,TINF,TTIM,TRAIN,APPLIED,TTIRRI,TMOIST,TERR,
1068      #          TSTP,(TQ(I),I=1,NPT)
1069
1070 32709 CONTINUE
1071      CLOSE(UNIT=LUB)
1072 999 STOP
1073
1074      END
0001
0002
0003  C-----
0004  C   Root Density Function for sink term
0005  C-----
0006      FUNCTION RHO(Z,AA,B1,B2)
0007      RHO = AA*EXP(-B1*Z)+B2
0008      RETURN
0009      END
0001      SUBROUTINE DELCHK(IDAYLS)
0002
0003  C-----
0004  C   This subroutine adjusts the time step according to the mass
0005  C   balance error. Specifically, it will increase or decrease the
0006  C   time step according to the ratio of
0007  C   DMAXBA = Maximum allowed mass balance error for any node

```

```

0008 C      DIFMAX = Maximum observed mass balance error for any node
0009 C-----
0010      INCLUDE 'BVAR.INC/NOLIST'
0041      INCLUDE 'BTIME.INC/NOLIST'
0048
0049      IF (DIFMAX .LE. 0.001) THEN
0050          R = 20.
0051      ELSE
0052          R = DMAXBA/DIFMAX
0053      ENDIF
0054      IF (R .LE. 1.) THEN
0055
0056 C-----
0057 C      Decrease the time step
0058 C-----
0059          RPRIM = R*R
0060          IF (RPRIM .LT. 0.5) RPRIM = 0.5
0061          TSUB = TSUB - DELT
0062          DELT = DELT * RPRIM
0063          IF (DELT .LT. DELMIN) THEN
0064              IF (IDAY .NE. IDAYLS) THEN
0065                  IDAYLS=IDAY
0066                  WRITE(LUS,6000) IDAY,DELT,R,DIFMAX,TDONE+TSUB
0067              ENDIF
0068              DELT = DELMIN
0069          ELSE
0070              IF (DELT .EQ. DELSAV) DELT = 0.75*DELT
0071          ENDIF
0072          IGO150 = 1
0073          TSUB = TSUB+DELT
0074          GO TO 1000
0075      ELSE
0076
0077 C-----
0078 C      Increase the time step
0079 C-----
0080          RPRIM = 0.5*(1+R)
0081          IF (RPRIM .GT. RFACT) RPRIM = RFACT
0082          IF (R .GE. RSAVE .OR. R .EQ. 20.) DELT = DELT*RPRIM
0083          IF (DELT .GT. DELSUB) DELT = DELSUB
0084          CALL TTLEFT
0085          IF (TLEFT .GT. 2*DELT) GO TO 900
0086          IF (TLEFT .LE. DELT) THEN
0087              DELT = MAX(DELMIN,TLEFT)
0088          ELSE
0089              DELT = TLEFT/2.0
0090          ENDIF
0091      ENDIF
0092      900  RSAVE = R
0093      1000 RETURN
0094
0095 C-----
0096 C      Format statements
0097 C-----
0098      6000  FORMAT(' IDAY',I3,' DELT',E12.5,' R',F8.4,' DIFMAX=',G12.5,
0099      #      ' TIME=',G12.5,/, ' Simulation continued with DELMIN')
0100      END
0001      SUBROUTINE TRIDAG( IF,L,A,B,C,D,V)
0002 C-----
0003 C      SUBROUTINE FOR SOLUTION OF TRIDIAGONAL MATRIX (TRIDAG)
0004 C-----
0005      DIMENSION A(L),B(L),C(L),D(L),V(L),BETA(250),GAMMA(250)
0006
0007      BETA(1F) = B(1F)
0008      GAMMA(1F) = D(1F)/BETA(1F)
0009      IFP1 = 1F+1
0010      DO 100 I=IFP1,L

```



```

0011      I1 = I - 1
0012      BETA(I) = B(I) - A(I)*C(I)/BETA(I1)
0013      GAMMA(I) = (D(I)-A(I)*GAMMA(I1))/BETA(I)
0014 100  CONTINUE
0015      V(L) = GAMMA(L)
0016      LAST = L - 1 IF
0017      DO 200 K = 1,LAST
0018          I = L - K
0019          V(I) = GAMMA(I) - C(I)*V(I+1)/BETA(I)
0020 200  CONTINUE
0021      RETURN
0022      END
0001      SUBROUTINE TTLEFT
0002
0003  C-----
0004  C   Calculates the "time left" until the end of the next time
0005  C   period or flow rate change.
0006  C-----
0007      INCLUDE 'BVAR.INC/NOLIST'
0008      INCLUDE 'BTIME.INC/NOLIST'
0009
0010      TSUBB = TSUB
0011      IF (TSUB .EQ. DELSUB) THEN
0012          SUBEND = SUBEND + DELSUB
0013          TDONE = TDONE + DELSUB
0014          TSUBB = 0
0015      ENDIF
0016      IF (ICHECK .EQ. 1 .AND. TSTART .LT. SUBEND .AND. TSTART .GE. TDONE)
0017          # ICHECK=2
0018
0019      GO TO (100,200,300,400), ICHECK+1
0020
0021 100  TLEFT = DELSUB - TSUBB
0022      GO TO 1000
0023
0024 200  TLEFT = DELSUB-TSUBB
0025      IF (ABS(TLEFT) .GT. 0.001) GO TO 1000
0026      ICHECK = 2
0027
0028 300  TLEFT = TSTART - TSUBB
0029      IF (TLEFT .GT. 0.0) GO TO 1000
0030      ICHECK = 3
0031
0032 400  IF (TEND .GE. SUBEND) THEN
0033      IF (TSUBB .NE. DELSUB) THEN
0034          TLEFT = DELSUB - TSUBB
0035      ELSE
0036          TLEFT = DELSUB
0037      ENDIF
0038      ELSE
0039      TLEFT = TEND - TSUBB
0040      IF (TLEFT .GT. 0.) GO TO 1000
0041      IR = IR + 1
0042      IF (IR .LE. NRATES) THEN
0043          TENO = RTIME(IR+1)
0044          IF (TENO .EQ. 0 .OR. TENO .LT. TDONE+TSUBB) THEN
0045              IF (TEND .GT. SUBEND) THEN
0046                  TLEFT = DELSUB -TSUBB
0047              ELSE
0048                  TLEFT = TEND -TSUBB
0049              ENDIF
0050          ELSE
0051              WRITE(LUS,6000) 1 DAY,N,TSUBB,TEND,IR,NRATES
0052              ENDOF
0053          ELSE
0054              ICHECK = 0
0055              TLEFT = DELSUB - TSUBB

```

```

0092         ENDIF
0093     ENDIF
0094
0095     1000 RETURN
0096
0097 C-----
0098 C   Format statements
0099 C-----
0100     6000 FORMAT(' ERROR AT 45 IDAY',15,' N',13,' TSUBB=',G12.5,' TEND=',
0101     #           G12.5,' IR=',13,' NRATES=',15)
0102
0103     END
C**
C** BOUT.INC
C** Variables, Arrays, and Common Blocks for DATAOUT
CHARACTER*80 IFILE,OFIL,RFIL,STIME*8,SDATE*9
INTEGER DAYEND,DAYSTP,TSTP
PARAMETER (M1=250)
DIMENSION DAYQ(M1),DAYSNK(M1),H(M1),THETA(M1),Z(M1),TQ(M1),NODE(6)
COMMON /INTEG/ DAYENO,DAYSTP,TSTP,LUI,LUR,LUS,LUH,NPT,IPLANT,
#           NPRINT,NSURPE,NTOTAL,MAXREC,IREC,IERROR,NCHR,1 DAY
COMMON /REAL/  HOUR,DAYINF,DAYRAN,DAYE,DAYTRA,DAYRUN,TPREV,TMOIST,
#           PE,PT,DAYTIM
COMMON /ARRAY/ DAYQ,DAYSNK,H,THETA,Z,TQ,NODE
COMMON /CHAR/  IFILE,OFIL,RFIL,STIME,SDATE
DATA LUI,LUR,LUS,LUH/1,5,6,7/
PROGRAM DATAOUT
0001
0002 C-----
0003 C   Latest Update 1-SEP-1985 by MJFayer
0004 C
0005 C   This program processes the binary output file that is
0006 C   generated by UNSAT-H.
0007 C-----
0008     INCLUDE 'BOUT.INC/NOLIST'
0022
0023     WRITE(LUS,6000)
0024     20  WRITE(LUS,6010)
0025     READ(LUR,5000) IFILE
0026     NCHR = INDEX(IFILE,' ')
0027     IFILE(NCHR:NCHR+3) = '.RES'
0028     OPEN(UNIT=LUI,NAME=IFILE,TYPE='OLD',FORM='UNFORMATTED',READONLY,
0029     #     ACCESS='DIRECT',RECORDTYPE='FIXED')
0030     READ(LUI,REC=1) NPT,IPLANT,(Z(I),I=1,NPT),DAYEND,NPRINT,NSURPE,
0031     #     NTOTAL,RFIL,SDATE,STIME
0032     MAXREC = 3+DAYEND*(1+NTOTAL*NPRINT)
0033     IREC = 2
0034     WRITE(LUS,*)
0035     50  WRITE(LUS,6020)
0036     READ(LUR,*) IOPT
0037     IF (IOPT.LT. 1 .OR. IOPT.GT. 5) IOPT = 0
0038
0039     GO TO (999,100,200,300,400,500), IOPT+1
0040
0041     100  CALL REINIT
0042     GO TO 50
0043     200  CALL SCAN
0044     GO TO 50
0045     300  CALL HARDCOPY
0046     GO TO 50
0047     400  CALL LISTDATA
0048     GO TO 50
0049     500  CLOSE (LUI)
0050     GO TO 20
0051
0052     999  STOP ' '
0053
0054 C-----
0055 C   FORMAT Statements

```

```

0056 C-----
0057 5000 FORMAT(A80)
0058 6000 FORMAT(10(/),
0059 # '      Program DATAOUT ',/,
0060 # '      Version 1-SEP-1985',/,
0061 # '      Contact: MJ Fayer 6-B326',/,
0062 # '      Program DATAOUT provides several options for',/,
0063 # '      processing the data output file (*.RES) generated',/,
0064 # '      of UNSAT-H'/)
0065 6010 FORMAT('/ Enter data filename (without ".RES" ext.) ==> ', $)
0066 6020 FORMAT(' COMMAND LEVEL: The processing options are...',/,
0067 # '      0) Exit the program',/,
0068 # '      1) Reinitialize',/,
0069 # '      2) Scan the data',/,
0070 # '      3) Create hardcopy output',/,
0071 # '      4) Create data files for plotting',/,
0072 # '      5) Change the current *.RES file',/,
0073 # '      Enter the number of your choice ==> ', $)
0074 END
0001
0002 SUBROUTINE READREC
0003 C-----
0004 C Latest Update 1-SEP-1985 by MJFayer
0005 C
0006 C Reads the *.RES file (from UNSAT-H), either DELSUB or daily
0007 C summary records
0008 C-----
0009 INCLUDE 'BOUT.INC/NOLIST'
0023 IERROR = 0
0024 IF (IPLANT .EQ. 0) THEN
0025 READ(LUI,REC=IREC,ERR=200) IDAY,HOUR,(H(J),THETA(J),DAYQ(J),
0026 # J=1,NPT),DAYINF,DAYRAN,DAVE,DAYTRA,DAYRUN,
0027 # TPREV,TMDIST,DAYSTP,PE,PT,DAYTIM
0028 ELSE
0029 READ(LUI,REC=IREC,ERR=200) IDAY,HOUR,(H(J),THETA(J),DAYQ(J),
0030 # DAYSIN(J),J=1,NPT),DAYINF,DAYRAN,DAVE,DAYTRA,
0031 # DAYRUN,TPREV,TMDIST,DAYSTP,PE,PT,DAYTIM
0032 ENDIF
0033 100 RETURN
0034
0035 200 WRITE(LUS,6000) IREC
0036 IERROR = 1
0037 GO TO 100
0038
0039 C-----
0040 C FORMAT Statements
0041 C-----
0042 6000 FORMAT('/ Record No. ',15,' could not be read'/)
0043
0044 END
0001 SUBROUTINE REINIT
0002 C-----
0003 C Latest Revision 1-SEP-1985 by MJFayer
0004 C
0005 C Auxillary Program to DATAOUT
0006 C Reads the *.RES file from the completed UNSAT-H run and
0007 C outputs a file (TOSS.OUT) with the final head values.
0008 C This file can serve as the initial conditions for the next
0009 C simulation
0010 C-----
0011 INCLUDE 'BOUT.INC/NOLIST'
0025 WRITE(LUS,6000)
0026 IREC = MAXREC-1
0027 CALL READREC
0028 IF (IERROR .EQ. 1) THEN
0029 WRITE(LUS,6010)
0030 GO TO 100

```

```

0031      ENDIF
0032      OPEN (UNIT=2,NAME='TOSS.OUT',TYPE='NEW')
0033      NDAY = 0
0034      WRITE(2,6020) NDAY,(H(1),I=1,NPT)
0035      WRITE(LUS,6030) IDAY
0036      WRITE(LUS,6000)
0037      CLOSE (2)
0038      100 RETURN
0039
0040      C-----
0041      C   FORMAT Statements
0042      C-----
0043      6000 FORMAT(/B0(' ')/)
0044      6010 FORMAT(' The chosen file could not be reinitialized')
0045      6020 FORMAT(15,T60,'NDAY'/4F14,3,T60,'Initial Conditions'/(4F14,3))
0046      6030 FORMAT(' File TOSS.OUT has been created.',/,
0047      #      ' It contains head values for the end of day ',13)
0048      END
0001      SUBROUTINE SCAN
0002      C-----
0003      C   Latest Update 1-SEP-1985 by MJFayer
0004      C
0005      C   Auxiliary Program to DATAOUT
0006      C   Allows the user to scan the end-of-DELSUB and end-of-day
0007      C   summaries for any day of a simulation, including the initial
0008      C   conditions and the year-end summary
0009      C-----
0010      INCLUDE 'BOUT.INC/NOLIST'
0024      LUM = 6
0025      WRITE(LUS,6000)
0026      WRITE(LUS,6010) NPT
0027      READ(LUR,*) NDEPTH,(NODE(I),I=1,NDEPTH)
0028      WRITE(LUS,*)
0029      IF (NDEPTH .LE. 0) THEN
0030          NDEPTH = 1
0031          NODE(1) = 1
0032      ENDIF
0033      ID = 1
0034      IF (DAYEND .GT. 9) ID = 2
0035      IF (DAYEND .GT. 99) ID = 3
0036
0037      C**   Choose a day for scanning
0038
0039      120 WRITE(LUS,6020) DAYEND,DAYEND+1
0040      READ(LUR,*) JDAY
0041      IF (JDAY .LE. -1) GO TO 200
0042      IF (JDAY .GT. DAYEND+1) THEN
0043          WRITE(LUS,6030) JDAY,DAYEND+1
0044          GO TO 120
0045      ENDIF
0046      IF (JDAY .EQ. DAYEND+1) THEN
0047          IREC = MAXREC
0048          CALL SUMMARY
0049          WRITE(LUS,6000)
0050          GO TO 120
0051      ENDIF
0052
0053      C**   Choose a DELSUB period for scanning
0054
0055      IF (NPRINT .EQ. 1 .AND. JDAY .NE. 0) THEN
0056          WRITE(LUS,6040) NTOTAL/NPRINT,1+(NTOTAL/NPRINT)
0057          READ(LUR,*) JH
0058      ENDIF
0059      IREC = 2+JDAY+(JH*(JDAY-1)*NTOTAL-1)*NPRINT
0060      CALL READREC
0061      IF (IERROR .EQ. 1) GO TO 120
0062

```

```

0063 C** Output the Initial conditions
0064
0065 IF (JOAY .EQ. 0) THEN
0066     NCHR1 = INDEX(RFILE,' ')
0067     WRITE(LUS,6050) RFILE,SDATE,STIME
0068     WRITE(LUS,6060)
0069     WRITE(LUS,6070) (1,Z(1),H(1),THETA(1),I=1,NPT)
0070     WRITE(LUS,6080) TMOIST
0071     WRITE(LUS,6000)
0072     GO TO 120
0073 ENDIF
0074
0075 C** Output DELSUB/daily summaries
0076
0077 SSFLUX = DAYINF-DAYE
0078 TNEW = TPREV+DAYINF+DAYRUN-DAYE-DAYTRA-DAYQ(NPT)
0079 DAYERR = ABS(TNEW-TMOIST)
0080 IF (NPRINT .EQ. 0 .OR. JH .EQ. 1+(NTOTAL/NPRINT)) THEN
0081     WRITE(LUS,6090) 1DAY,HOUR
0082 ELSE
0083     WRITE(LUS,6100) 1DAY,HOUR
0084 ENDIF
0085 WRITE(LUS,6110) (Z(NODE(I)),I=1,NDEPTH)
0086 WRITE(LUS,6120) (THETA(NODE(I)),I=1,NDEPTH)
0087 WRITE(LUS,6130) (H(NODE(I)),I=1,NDEPTH)
0088 WRITE(LUS,6140) (DAYQ(NODE(I)),I=1,NDEPTH)
0089 IF (IPLANT .GT. 0) WRITE(LUS,6150) (DAYSXK(NODE(I)),I=1,NDEPTH)
0090 WRITE(LUS,6160)
0091 WRITE(LUS,6170) TPREV,DAYINF,DAYRUN,DAYE,
0092 # DAYTRA,DAYQ(NPT),TNEW,TMOIST
0093 WRITE(LUS,6180) DAYERR,DAYSTP
0094 IF (NSURPE .EQ. 1) WRITE(LUS,6190) PE,DAYE
0095 IF (IPLANT .EQ. 1) WRITE(LUS,6200) PT,DAYTRA
0096 WRITE(LUS,6000)
0097 GO TO 120
0098
0099 200 WRITE(LUS,6000)
0100 LUW = 7
0101 RETURN
0102
0103 C-----
0104 C FORMAT Statements
0105 C-----
0106 6000 FORMAT(/80(' ')/)
0107 6010 FORMAT(' SCAN (Option No. 2)',/,
0108 # ' When scanning the results, data for up to 6 ',/,
0109 # ' individual nodes can be viewed.',/,
0110 # ' Enter the number of nodes desired (1 to 6) and ',/,
0111 # ' the node numbers (1 to ',13,') ==> ',)
0112 6020 FORMAT(' The SCAN options are:',/,
0113 # ' -1) Return to DATAOUT command level',/,
0114 # ' 0) Look at the Initial conditions',/,
0115 # '<9-ID>X, '1 to ',1<ID>, ' Look at the end-of-day summary for',
0116 # ' chosen day',/,
0117 # '<9-ID>X, ' ',1<ID>, ' Look at the end-of-year summary',/,
0118 # ' Enter the desired option ==> ',)
0119 6030 FORMAT(' The chosen day ( ',13,') is greater than DAYEND+1 ( ',
0120 # '13,'). Try again...')
0121 6040 FORMAT(' This file contains hourly data. The options are:',/,
0122 # ' T6,1 to ',15,T15') summary of the chosen time period',/,
0123 # ' T10,15,T15') overall summary of the chosen day',/,
0124 # ' Enter the desired choice ==> ',)
0125 6050 FORMAT('///T26'INITIAL CONDITIONS'/T26,18(' ')/,
0126 # ' Input Filename: ',A<NCHR1-1>,T35,' Date: ',A9,T6 0,
0127 # ' Time: ',A8,/80(' '))
0128 6060 FORMAT(2(5X,'NODE',4X,'DEPTH',6X,'HEAD',4X,'THETA'),/,
0129 # 2(5X,'-----',3X,'-----',4X,'-----',2X,'-----'))

```

```

0130 6070 FORMAT((2(6X,13,F10.3,F10.1,F8.4)))
0131 6080 FORMAT('/ Initial Water Storage = ',F8.4,' cm')
0132 6090 FORMAT('/ DAILY SUMMARY: Day = ',13,' Total Simulated Time = ',
0133 # F6.3,' hours',/80('-'))
0134 6100 FORMAT('/ HOURLY SUMMARY: Day = ',14,' Hour = ',
0135 # F7.4,'/80('-'))
0136 6110 FORMAT(' Depth ',6F12.2)
0137 6120 FORMAT(' Moist ',6(F12.4))
0138 6130 FORMAT(' Head ',5(1PE12.3))
0139 6140 FORMAT(' Flux ',5(1PE12.3))
0140 6150 FORMAT(' Slnk ',5(1PE12.3))
0141 6160 FORMAT('/ PRESTOR',X,' INFIL',X,' RUNOFF ',X,' EVAPO ',
0142 # X,' TRANS',2X,' DRAIN',X,' NEWSTOR',10X,' STORAGE')
0143 6170 FORMAT(F9.4,'+',F7.4,'+',F7.4,'x','-',
0144 # F7.4,'-',F7.4,'-',F8.4,' = ',F8.4,' Versus',F9.4/)
0145 6180 FORMAT(' Mass Balance (cm) = ',F7.4,' Time steps = ',15)
0146 6190 FORMAT('/ Evaporation: Potential = ',F7.4,' Actual = ',F7.4)
0147 6200 FORMAT('/ Transpiration: Potential = ',F7.4,' Actual = ',F7.4)
0148
0149 END
0001 SUBROUTINE SUMMARY
0002 C-----
0003 C Lastest Update 1-SEP-1985 MJFayer
0004 C
0005 C Auxiliary program to DATAOUT
0006 C This subroutine reads and outputs the year-end summary data
0007 C found in the last record of an UNSAT-H *.RES file
0008 C-----
0009 INCLUDE 'BOUT.INC/NOLIST'
0023 READ(LUI,REC=MAXREC,ERR=2D0) IDAY,G,IPLANT,TPET,TPT,TTRA,TPE,TE,
0024 # PEGROW,TRUNOF,TINF,TTIM,TRAINN,APLIED,TTIRRI,
0025 # TMOIST,TERR,TSTP,(TQ(1),I=1,NPT)
0026 WRITE(LUW,6000) IDAY-1,IPLANT,TPET,TPT,TTRA,TPE,TE,PEGROW,
0027 # TRUNOF,TINF,TQ(NPT),TRAINN,APLIED,TTIRRI,TMOIST,
0028 # TERR,TSTP,TTIM
0029 SFLOW = TINF-TE
0030 WRITE(LUW,6020) IDAY-1
0031 WRITE(LUW,6030) Z(1),SFLOW,(0.5*(Z(1)+Z(1+1)),TQ(1),I=1,NPT-1),
0032 # Z(NPT),TQ(NPT)
0033 100 RETURN
0034
0035 200 WRITE(LUS,6040) MAXREC
0036 GO TO 100
0037
0038 C-----
0039 C FORMAT Statements
0040 C-----
0041 6000 FORMAT('/ YEAR-END SUMMARY FOR ',13,' SIMULATION DAYS ',
0042 # '(Units: cm)',/,
0043 # ' Transpiration Scheme is: =',14/
0044 # ' Potential Evapotranspiration =',F9.4/
0045 # ' Potential Transpiration =',F9.4/
0046 # ' Actual Transpiration =',F9.4/
0047 # ' Potential Evaporation =',F9.4/
0048 # ' Actual Evaporation =',F9.4/
0049 # ' Evaporation during Growth =',F9.4/
0050 # ' Total Runoff =',F9.4/
0051 # ' Total Infiltration =',F9.4/
0052 # ' Total Drainage at Base of Profile =',F9.4/
0053 # ' Total Rainfall =',F9.4/
0054 # ' Actual Rainfall =',F9.4/
0055 # ' Total Irrigation =',F9.4/
0056 # ' Total Final Moisture Storage =',F9.4/
0057 # ' Mass Balance Error =',F9.4/
0058 # ' Total Time Steps (this run) =',19/
0059 # ' Total Time Actually Simulated (days) =',F9.4)
0060 6020 FORMAT('/ Total flux (cm) across different depths at the ',

```

```

0061      #      ' end of ',13,' days: '//
0062      #      3(5X,'DEPTH',8X,'FLUX',3X),/,
0063      #      3(4X,'-----',3X,'-----'))
0064 6030  FORMAT(3(4X,OP,F7.3,1P,E14.4))
0065 6040  FORMAT('/' The end-of-year summary record ('',15,'') could not',
0066      #      ' be read. SUMMARY aborted,'',/))
0067
0068      END
0001  SUBROUTINE LISTDATA
0002  C-----
0003  C      Latest Update 1-SEP-1985 by MJFayer
0004  C
0005  C      Auxiliary Program to DATAOUT
0006  C      Allows the user to create data lists, such as head values at
0007  C      the end of each day, for review on the screen or for output
0008  C      to a separate file to be used for plotting purposes.
0009  C-----
0010      INCLUDE 'BOUT.INC/NOLIST'
0024      WRITE(LUS,6000)
0025      WRITE(LUS,6010)
0026
0027  C**  Choose UNSAT-H variable for output
0028
0029  100  NOUT = 1
0030      WRITE(LUS,6020)
0031      READ(LUR,*) CHOICE
0032      IF (CHOICE .LE. -1) GO TO 200
0033      IF (NPRINT .EQ. 1 .AND. CHOICE .EQ. 2) THEN
0034          WRITE(LUS,6030) NTOTAL
0035          READ(LUR,*) NOUT
0036          IF (NOUT .LT. 1) NOUT = 1
0037          IF (NOUT .GT. NTOTAL) NOUT = NTOTAL
0038          ICOUNT = 1
0039      ENDIF
0040
0041  C**  Choose output device option, either screen or file
0042
0043      WRITE(LUS,6040)
0044      READ(LUR,*) IOUT
0045      IF (IOUT .EQ. 1) THEN
0046          OFILE = IFILE
0047          OFILE(NCHR:NCHR+3) = '.LIS'
0048          OPEN(UNIT=LUM,FILE=OFILE,STATUS='NEW')
0049      ELSE
0050          LUM = 6
0051      ENDIF
0052
0053  C**  Enter node numbers for desired output
0054
0055      IF (CHOICE .EQ. 0 .OR. CHOICE .EQ. 1) THEN
0056          WRITE(LUS,6050) NPT
0057          READ(LUR,*) NDEPTH,(NODE(I),I=1,NDEPTH)
0058          WRITE(LUS,*)
0059          IF (NDEPTH .LE. 0) THEN
0060              NDEPTH = 1
0061              NODE(1) = 1
0062          ENDIF
0063      ENDIF
0064      WRITE(LUM,6060) RFILE,SDATE,STIME,IFILE
0065      IF (CHOICE .EQ. 0 .OR. CHOICE .EQ. 1) THEN
0066          WRITE(LUM,6070) (NODE(I),I=1,NDEPTH),(Z(NODE(I)),I=1,NDEPTH)
0067      ELSE
0068          WRITE(LUM,6080)
0069      ENDIF
0070
0071  C**  Read variables and write to screen or file
0072

```

```

0073 DO 130 IREC=2,MAXREC-1
0074 IF (NPRINT .EQ. 1 .AND. IREC .EQ. (2+IDAY*(NTOTAL+1))) GO TO 130
0075 CALL READREC
0076 IF (IERROR .EQ. 1) GO TO 135
0077 DAY = IDAY-1.0+HOUR/24.0
0078 IF (IDAY .EQ. 0) DAY = 0.0
0079 IF (CHOICE .EQ. 0) THEN
0080 WRITE(LUW,6090) DAY,(H(NODE(1)),I=1,NDEPTH)
0081 ELSE IF (CHOICE .EQ. 1) THEN
0082 WRITE(LUW,6100) DAY,(THETA(NODE(1)),I=1,NDEPTH)
0083 ELSE IF (CHOICE .EQ. 2) THEN
0084 IF (NPRINT .GE. 1 .AND.
0085 # IREC .EQ. 2+IDAY*(1+NTOTAL/NPRINT)) GO TO 130
0086 IF (IDAY .GT. 0) RATINF = DAYINF/DAYTIM
0087 CUMINF = CUMINF+DAYINF
0088 IF (ICOUNT .EQ. NOUT) THEN
0089 WRITE(LUW,6110) DAY,RATINF,CUMINF
0090 ICOUNT = 0
0091 ENDIF
0092 ICOUNT = ICOUNT+1
0093 ENDIF
0094 130 CONTINUE
0095 135 IF (IOUT .EQ. 1) THEN
0096 WRITE(LUS,6120) OFILE
0097 CLOSE (LUW,STATUS='KEEP')
0098 ENDIF
0099 LUW = 7
0100 WRITE(LUS,6000)
0101 GO TO 100
0102
0103 200 WRITE(LUS,6000)
0104 RETURN
0105
0106 C-----
0107 C FORMAT Statements
0108 C-----
0109 6000 FORMAT(/80(' ')/)
0110 6010 FORMAT(' LISTDATA (Option No. 4)'/)
0111 6020 FORMAT(' The LISTDATA options are: ',/,
0112 # ' -1) Return to DATAOUT command level',/,
0113 # ' 0) Look at HEAD values',/,
0114 # ' 1) Look at THETA values',/,
0115 # ' 2) Look at Infiltration (rate and cum)',/,
0116 # ' Enter the desired option ==> ',)
0117 6030 FORMAT(/' Options for reducing the amount of infil. output',/,
0118 # ' 1) output every DELSUB period',/,
0119 # ' 2 to 15) output reduced by a factor equal to ',/,
0120 # ' the selected choice',/,
0121 # ' Choose output reduction option ==> ',)
0122 6040 FORMAT(/' Options for output device: ',/,
0123 # ' 0) screen',/,
0124 # ' 1) file',/,
0125 # ' Choose output device option ==> ',)
0126 6050 FORMAT(/' When listing data, data for up to 7 ',/,
0127 # ' Individual nodes can be viewed.',/,
0128 # ' Enter the number of nodes desired (1 to 7) and ',/,
0129 # ' the node numbers (1 to 13,) ==> ',)
0130 6060 FORMAT(/' Run Input Filename: ',A50,/,
0131 # ' Date of Run: ',A9,/,
0132 # ' Time of Run: ',A8,/,
0133 # ' Results Filename: ',A50,/)
0134 6070 FORMAT(T5,'NODE = ',T15,<NDEPTH>(12,8X),/,
0135 # T4,'DEPTH = ',<NDEPTH>(F8.2,2X),/,
0136 # ' DAY'/80(' ')/)
0137 6080 FORMAT(T13,'Rate of T24 Cumulative'/
0138 # T5'Day'T11'Infil(cm/hr)'T24'Infil (cm)'/80(' ')/)
0139 6090 FORMAT(F10.5,7F10.1)

```



```

0140 6100 FORMAT(F10.5,7(2X,F6.4,2X))
0141 6110 FORMAT(F10.5,1P2E12.4)
0142 6120 FORMAT(/' The data has been written to file ',A<NCHR+3>)
0143
0144      END
0001      SUBROUTINE HARDCOPY
0002 C-----
0003 C      Latest Update 1-SEP-1985 by MJFayer
0004 C
0005 C      Auxiliary Program to DATAOUT
0006 C      Allows the user to create a hard copy of the results of an
0007 C      UNSAT-H simulation, including the initial conditions and the
0008 C      year-end summary. The user can choose to have the copy saved
0009 C      as a file on disk or have the copy printed directly without
0010 C      saving it on disk.
0011 C-----
0012      INCLUDE 'BOUT.INC/NOLIST'
0026      WRITE(LUS,6000)
0027      WRITE(LUS,6010)
0028      OFILE = 'FILE'
0029      OFILE(NCHR:NCHR+3) = '.OUT'
0030      OPEN(UNIT=LUM,FILE=OFILE,STATUS='NEW')
0031      WRITE(LUS,6020)
0032      READ(LUR,*) OPTION
0033      WRITE(LUS,6030) NPT
0034      READ(LUR,*) NDEPTH,(NODE(I),I=1,NDEPTH)
0035      WRITE(LUS,*)
0036      IF (NDEPTH .LE. 0) THEN
0037          NDEPTH = 1
0038          NODE(1) = 1
0039      ENDIF
0040      IREC = 2
0041      JDAY = 0
0042 100 IF (OPTION .EQ. 1 .AND. JDAY .EQ. 2) IREC = MAXREC-1
0043      CALL READREC
0044      IF (IERROR .EQ. 1) THEN
0045          CLOSE (LUM,STATUS='KEEP')
0046          WRITE(LUS,6040)
0047          GO TO 300
0048      ENDIF
0049      IF (IREC .EQ. 2) THEN
0050
0051 C**      Initial Condition summary
0052
0053          NCHR1 = INDEX(RFILE,' ')
0054          WRITE(LUM,6050) RFILE,SDATE,STIME
0055          WRITE(LUM,6060)
0056          WRITE(LUM,6070) (I,Z(I),H(I),THETA(I),I=1,NPT)
0057          WRITE(LUM,6080) TMO1ST
0058      ELSE
0059
0060 C**      Daily summary
0061
0062          SSFLUX = DAYINF-DAYE
0063          TNEW = TPREV+DAYINF+DAYRUN-DAYE-DAYTRA-DAYQ(NPT)
0064          DAYERR = ABS(TNEW-TMO1ST)
0065          WRITE(LUM,6090) IDAY,HOUR
0066          WRITE(LUM,6100) (Z(NODE(I)),I=1,NDEPTH)
0067          WRITE(LUM,6110) (THETA(NODE(I)),I=1,NDEPTH)
0068          WRITE(LUM,6120) (H(NODE(I)),I=1,NDEPTH)
0069          WRITE(LUM,6130) (DAYQ(NODE(I)),I=1,NDEPTH)
0070          IF (IPLANT .GT. 0) WRITE(LUM,6140) (DAYSNK(NODE(I)),I=1,NDEPTH)
0071          WRITE(LUM,6150)
0072          WRITE(LUM,6160) TPREV,DAYINF,DAYRUN,DAYE,
0073          #      DAYTRA,DAYQ(NPT),TNEW,TMO1ST
0074          WRITE(LUM,6170) DAYERR,DAYSTP
0075          IF (NSURPE .EQ. 1) WRITE(LUM,6180) PE,DAYE

```

```

0076      IF (IPLANT.EQ. 1) WRITE(LUW,6190) PT, DAYTRA
0077      ENDIF
0078      IF (IREC.EQ. MAXREC-1) GO TO 200
0079      JDAY = JDAY+1
0080      IREC = 2+(NTOTAL*NPRINT+1)*JDAY
0081      GO TO 100
0082
0083      C** Year-end Summary
0084
0085      200  WRITE(LUW,6200)
0086           IREC = MAXREC
0087           CALL SUMMARY
0088
0089      C** Output to either a file or the printer
0090
0091      300  WRITE(LUS,6210)
0092           READ(LUR,*) IOPT
0093           IF (IOPT.EQ. 1) THEN
0094               CLOSE (LUW,STATUS='KEEP')
0095               WRITE(LUS,6220) OFILE
0096           ELSE
0097               CLOSE (LUW,STATUS='PRINT/DELETE')
0098           ENDIF
0099
0100      400  WRITE(LUS,6000)
0101           RETURN
0102
0103      C-----
0104      C      FORMAT Statements
0105      C-----
0106      6000  FORMAT(/80(' ')/)
0107      6010  FORMAT(' HARDCOPY (Option No. 3)')
0108      6020  FORMAT(' The HARDCOPY options are: '//,
0109           #      '          1) Initial conditions, summaries of '//,
0110           #      '          the first and last days, and the '//,
0111           #      '          year-end summary',/,
0112           #      '          2) Same as option (1) but includes a '//,
0113           #      '          summary of each and every day',/,
0114           #      ' Enter the desired option ==> ', $)
0115      6030  FORMAT(/' When printing the results, data for up to 6 '//,
0116           #      ' individual nodes can be included.',/,
0117           #      ' Enter the number of nodes desired (1 to 6) and ',/,
0118           #      ' the node numbers (1 to 13,') ==> ', $)
0119      6040  FORMAT(/' HARDCOPY stopped'/)
0120      6050  FORMAT(T22'UNSAT-H SIMULATION RESULTS'//,
0121           #      ' Input Filename: ', A<NCHR1-1>, T35, 'Date: ', A9, T60,
0122           #      ' Time: ', A8, /80(' ')/)
0123           T26'INITIAL CONDITIONS'/T26, 18(' ')/)
0124      6060  FORMAT(2(5X, 'NODE', 4X, 'DEPTH', 6X, 'HEAD', 4X, 'THETA'),/,
0125           #      2(5X, '-----', 3X, '-----', 4X, '-----', 2X, '-----'))
0126      6070  FORMAT(2(6X, 13, F10.3, F10.1, F8.4))
0127      6080  FORMAT(/' Initial Water Storage = ', F8.4, ' cm')
0128      6090  FORMAT(/80(' ')/' DAILY SUMMARY: Day = ', I3, ' Total ',
0129           #      ' Simulated Time = ', F7.4, ' hours'//, 80(' '))
0130      6100  FORMAT(' Depth ', 6F12.2)
0131      6110  FORMAT(' Moist ', 6(F12.4))
0132      6120  FORMAT(' Head ', 6(1PE12.3))
0133      6130  FORMAT(' Flux ', 6(1PE12.3))
0134      6140  FORMAT(' Sink ', 6(1PE12.3))
0135      6150  FORMAT(/' PRESTOR', X, ' INFIL', X, ' RUNOFF ', X, ' EVAPO ',
0136           #      ' X, ' TRANS', 2X, ' DRAIN', X, ' NEWSTOR', 10X, 'STORAGE')
0137      6160  FORMAT(F9.4, '+', F7.4, '+', F7.4, 1X, '-',
0138           #      ' F7.4, '-', F7.4, '-', F8.4, ' =', F8.4, ' Versus', F9.4/)
0139      6170  FORMAT(' Mass Balance (cm) = ', F7.4, ' Time steps = ', I6)
0140      6180  FORMAT(/' Evaporation: Potential = ', F7.4, ' Actual = ', F7.4)
0141      6190  FORMAT(/' Transpiration: Potential = ', F7.4, ' Actual = ', F7.4)
0142      6200  FORMAT(1H1)

```

```

0143 6210 FORMAT(/' Output specification:',//,
0144      #      '      1) to a file on disk',//,
0145      #      '      2) directly to the printer',//,
0146      #      ' Enter the desired choice ==> ', $)
0147 6220 FORMAT(/' HARDCOPY output can be found in file ', A<NCHR+3>)
0148
0149      END

```

REFERENCES

- Ritchie, J. T. 1972. "Model for Predicting Evaporation from a Row Crop with Incomplete Cover." Water Resour. Res. 8(5):1204-1212.

APPENDIX D

UNSAT-H GLOSSARY

APPENDIX D

UNSAT-H GLOSSARY

The UNSAT-H convention is to express soil water pressure potential in units of head (i.e., the height of a column of water corresponding to the given pressure). The pressure potential (or head) of unsaturated soil is negative and the magnitude of this value is called suction or tension. Because UNSAT-H will be used almost entirely to model unsaturated flow, we have retained the sign convention of the earlier code by Gupta et al. (1978). That is, the sign of the head values in UNSAT-H is reversed: positive head values refer to suction values; negative head values refer to positive pressures (greater than atmospheric).

VARIABLES

A	Coefficient in Haverkamp conductivity function
AA	Coefficient in the exponential root growth equation
ALPHA	Coefficient in Haverkamp moisture characteristic description
ALPHAF	Sink reduction factor, based on the moisture content of each node (Feddes method)
APPLIED	Running total of rainfall/irrigation amounts as they are input
AVC	Average specific moisture capacity over a time step (units: 1/cm)
B	Coefficient in Campbell functions and Haverkamp conductivity function
BARE	Fraction of soil surface that is bare of plants [in DATAINH]
BETA	Coefficient in Haverkamp moisture characteristic description
BIOMAS	Plant shoot biomass scaling factor used in adjusting transpiration [in DATAINH] (units: g/m ²)
B1	Coefficient in the exponential root growth equation
B2	Coefficient in the exponential root growth equation
CHEKEM	Logical. A .TRUE. condition indicates rainfall has occurred earlier in the day or is occurring presently and rainfall will occur at some time before the end of the day. Before the first and after the last rain events of the day, CHEKEM is .FALSE.
DAYE	Running total of evaporation for a given day (units: cm)
DAYEND	Ending simulation day
DAYINF	Running total of infiltration for a given day (units: cm)
DAYIRR	Running total of irrigation for a given day (units: cm)
DAYPE	Total potential evaporation for a given day (units: cm)
DAYPT	Total potential transpiration for a given day (units: cm)
DAYRAN	Running total of rain for a given day (units: cm)
DAYRUN	Running total of runoff for a given day (units: cm)
DAYSTP	Running total of time steps taken during a given day

DAYRAN	Running total of rain for a given day (units: cm)
DAYRUN	Running total of runoff for a given day (units: cm)
DAYSTP	Running total of time steps taken during a given day
DAYTIM	Actual number of hours simulated in a particular day (units: h)
DAYTRA	Running total of transpiration for a given day (units: cm)
DEL	A shortened version of 1/DELSAV for use in calculating the coefficients for the solver (in the p-c loop) (units: 1/h)
DELMAX	Maximum time step (normally 1 h) (units: h)
DELMIN	Minimum allowable time step (units: h)
DELSAV	The time step of the current p-c loop (units: h)
DELSUB	Length of a time subdivision of a day, normally equal to DELMAX (units: h)
DELSVL	Value of the last successful time step (units: h)
DELT	The projected time step (units: h)
DIFFMA	Mass balance error that is calculated at the end of each successful time step when ISWDIF = 1 (units: cm)
DIFMAX	Maximum observed percent change in THETA for any node when ISWDIF = 0 (units: cm ³ /cm ³)
DMAXBA	Time step control parameter. If ISWDIF = 0, DMAXBA is the maximum allowable percent change in the water content of any node. If ISWDIF = 1, DMAXBA is the maximum allowable difference between predicted and calculated soil profile water storage values (units: cm ³ /cm ³ or cm, respectively)
DOWN	Downstream weighting factor for estimating conductivity at the midpoint between nodes; DOWN = 1.0 - UP
DRYC	Limit of specific moisture capacity when H is set to HDRY
DRYK	Limit of conductivity when H is set to HDRY
DUMMY	Dummy character variable used for input [in DATAINH]
EFICEN	Efficiency of the irrigation scheme (i.e., how much of the water actually gets onto the soil surface)
ELPAST	Running total of simulated time in a day, calculated up to the last successfully completed time step
ELTIME	Running total of simulated time in day, assuming the present time step will be successful
FZZ1	Part of solution matrix
FZZ2	Part of solution matrix
G	Option to orient the soil profile (0) horizontally or (1) vertically
HD	Head corresponding to moisture content below which plant transpiration starts to decrease below the potential (units: cm)
HDRY	Maximum head to which the soil can dry out (units: cm)
HPOND	Initial ponding height when IRTYPE = 3 or 4 (units: cm)
HIRRI	Minimum head that can occur during rain or irrigation. A negative number indicates a positive pressure (units: cm)
HN	Head corresponding to moisture content above which plants do not transpire because of anaerobic conditions (units: cm)
HOURL	Running total of time simulated during the day (units: h)
HPOND	Current ponding height when IRTYPE = 3 or 4 (units: cm)
HTMP	Temporary storage location for head values during extrapolation of head values for property estimation (units: cm)
HTOP	Head value of the surface node when ITOPBC is set to (1) for a constant head boundary condition (units: cm)

HW Head corresponding to moisture content below which plants wilt and stop transpiring (units: cm)
 I Integer index
 ICHECK Identifies a water application day
 (0) no
 (1) yes, but start of water application is not in the current DELSUB period
 (2) yes, and the starting time of a water application period (TSTART) is within the current DELSUB period
 (3) yes, already within a water application period
 ID Index identifying a soil property polynomial in POLYCH, POLYKH, and POLYTH
 IDAY Day number
 IDAYLS Flag that indicates the last day that a detailed print was generated when DELT went below DELMIN. Occurs in the time step control loops. Allows only 1 print per day, then IDAYLS is reset to IDAY
 IERROR Sum of nodes for which the capacity was calculated in POLYCH as negative. If greater than zero, the program stops
 IFILE Name of the input file (with *.BIN extension)
 IGO150 Flag in the p-c loop:
 (0) indicates nothing
 (1) indicates a p-c iteration must be repeated because of a water balance problem
 IHPOND Flag indicating that there will be ponding during that particular day
 II Integer index
 III Integer index
 IN Indicates location within the predictor-corrector loop, (0) before the loop, (1) after the predictor but before the corrector, and (2) after the corrector
 INC Potential number of coefficients for completely describing any one soil hydraulic property, found in POLYCH, POLYKH, and POLYTH
 IP Index of soil property coefficients, found in POLYCH, POLYKH, and POLYTH
 IPLANT Option to include plants
 (0) no plants
 (1) plants
 IR Index number of each rainfall event during a given day
 IRAIN Flag indicating water application in the form of rain on day IRDAY=IDAY
 IRDAY Day on which a rain/irrigation event occurs
 IREC Record counter for control of binary output
 IRL Binary record length that depends on the number of nodes
 IRR1 Flag indicating water application in the form of irrigation or ponding on day IRDAY=IDAY
 IRTYPE Option for water application
 (1) rain
 (2) irrigation
 (3) constant ponding depth less than 1 day
 (4) constant ponding depth greater than 1 day
 IRYES Flag indicating that there is surface infiltration during that time step

ISATUK Flag (normally 0) that is set to (1) to initiate a saturated surface condition at the start of a water application event and (2) once the event has been initiated

ISAVUP Saves the UPPER switch so it can be reset correctly if it is changed during the p-c loop

ISTART Array index passed to subroutine TRIDAG
 (1) flux surface boundary condition
 (2) constant head surface boundary condition

ISTEP It is (0) on the first time step and sets ZKKMID and GGMOD equal to ZKMID and GMOD, respectively, on the first time. After the first time step, ISTEP is always (1) and ZKKMID and GGMOD are always equal to the ZKMID and GMOD values at the end of the previous time step

ISWDIF Option for time step control
 (0) maximum percent change in THETA
 (1) mass balance

ITOPBC Option for the surface boundary condition
 (0) flux
 (1) constant head, for which the program will require a value for HTOP, the head at the surface

ITPOND Index of the next time for changing the ponding depth when IRTYPE = 4

IVAPOR Option to include isothermal vapor flow; requires values for TORT, VAPDIF, and TSOIL

IYEAR Index of year being simulated

J Index

KEST Options for estimating the conductivity between nodes
 (1) arithmetic mean
 (2) upstream weighting of arithmetic mean
 (3) geometric mean

KLMID The liquid water conductivity at the midpoint between nodes (units: cm/h)

KOPT Options available to describe soil hydraulic properties
 (1) polynomials
 (2) Haverkamp functions
 (3) Campbell functions
 (4) Haverkamp functions except that head in the moisture retention function is replaced by $\ln(\text{head})$

LEAF Option for entering leaf area index (LAI) (if IPLANT = 1) [in DATAINH]
 (1) User supplies LAI values for the year
 (2) User supplies a subroutine

LEFT Time remaining until the start of a water application period or the end of the current DELSUB period, whichever comes first

LOGE Shortened version of $\log(\exp(1))$; used in POLYCH

LOWER Options to describe the lower boundary condition
 (1) semi-infinite profile with a unit gradient
 (2) static water table
 (3) inoperative
 (4) dynamic water table, where the flux at the base is entered for each day
 (5) impermeable boundary

MATI Short version of MAT(I)

MATN	Number of different soil materials
MAXPOL	Maximum degree of soil property polynomials + 1
MAXSUB	Maximum number of subdivisions of any given soil property description. In other words, the maximum number of polynomials needed to completely describe the soil property
MDEPTH	Node to which roots have penetrated on a given day
MGR	Lumped parameter; molar weight of water times the gravity constant, divided by the universal gas constant (units: °K/cm)
MXROOT	Deepest node to which roots will penetrate during the year
N	Index of present DELSUB period
NCHR	Index of the '.' character position in a file name
NDAY	Starting day
NDAYS	Total number of days of available data
NEWPON	Time (ITPOND) when the present ponding height should be changed (units: h)
NFHOURL	Option for distributing the daily potential evapotranspiration (PET) value over the course of the day [in DATAINH] (1) User supplies 24 hourly values that must sum to 1.0 (2) 24 hourly values are generated using a sine wave function from 0600 to 1800 and assuming the factors during the remaining hours are 0.01; all factors summing to 1.0
NFPET	Option for defining PET [in DATAINH] (1) User supplies daily PET values and partitions it into PE and PT based on LAI and equation by Ritchie (1971) (2) User supplies daily PET values and partitions it into PE and PT based on the cheatgrass data of Hinds (1975)
NFROOT	Option for defining root growth [in DATAINH] (1) Exponential relationship (2) User supplies a subroutine
NGRAV	Orientation of the profile (0) horizontal (1) vertical
NGROW	Growth day, starting from NSOW (units: day)
NHR	Integer indicating the hour of the day
NHRVST	Number of days from NSOW to harvest (units: day)
NPOINT	Number of changes in the LAI relationship when LEAF = 1 [in DATAINH]
NPPT	NPT-1
NPRINT	Option for level of output (0) daily summaries with end-of-year summary (1) hourly and daily summaries with end-of-year summary, limited to 15 days total because of size of output file
NPT	Number of nodes
NRATES	Total number of water application rate changes on a given day
NSINK	Flag indicating that plants are transpiring (1) or not (0) during the present time step
NSOW	Day of the year on which seeds are planted
NSTOP	For output control [in DATAINH]
NSURPE	Option for surface evaporation (0) no evaporation (1) evaporation

NTLEAK Number of times the flux lower boundary condition changes during the simulation period (LOWER = 4) [in DATAINH]
 NTOTAL Number of time subdivisions in a day. Normally 24 but can be more if DELMAX < 1
 NUPTAK Option for plant water uptake [in DATAINH]
 (1) Feddes approach (Feddes et al. 1978), requires THETAW, THETAD, and THETAN
 (2) User supplies a subroutine
 NWATER Total number of days on which rainfall or irrigation occurs
 NYEARS Number of years that the simulation will be repeated
 OFILE Output file name
 PE Potential evaporation for the day, equivalent to PEVAPO(IDAY) (units: cm/day)
 PESUB Hourly potential evaporation rate (units: cm/h)
 PET Daily PET, the sum of PEVAPO and PTRANS (units: cm)
 PIRRI Irrigation rate (units: cm/h)
 PRAIN Rainfall rate (units: cm/h)
 PT Potential transpiration for the day, equivalent to PTRANS(IDAY) (units: cm/day)
 PTSUB Hourly potential transpiration rate (units: cm/h)
 QLAST Flux rate imposed at the bottom boundary (units: cm/h)
 R Ratio of DMAXBA to DIFMAX; calculated in DELCHK and used to adjust the time step
 RAINIF Factor used to reduce the time step at the start of a rain or irrigation event
 RFACT Maximum factor by which the time step can increase following a successful time step
 ROOTS Total length of roots in the profile
 RSAVE Last value of R that was greater than 1
 RUNOFF Runoff during a time step (units: cm)
 SATURC Specific water capacity corresponding to a pressure head of HIRRI (units: 1/cm)
 SATURK Hydraulic conductivity corresponding to a pressure head of HIRRI (units: cm/h)
 SDATE Character variable which contains the date the simulation was started
 SFLUX Potential surface flux during a time step, either infiltration or evaporation (units: cm/h)
 SMOIST Profile moisture storage at the end of a time step and before a mass balance check is made. Calculated as the sum of THETA(I)*THICK(I)
 SPREV Profile moisture storage at the beginning of the current DELSUB period (units: cm)
 SSFLUX Actual surface flux (units: cm)
 STIME Character variable which contains the time the simulation was started
 STORE Initial soil moisture storage value, TMOIST (units: cm)
 SUBE Running total of evaporation during a DELSUB time period (units: cm)
 SUBEND The ending time for a given time subdivision (units: h)
 SUBINF Running total of infiltration during the given DELSUB period (units: cm)
 SUBIRR Running total of irrigation during a DELSUB period (units: cm)
 SUBRAN Running total of rainfall during a DELSUB period (units: cm)

SUBRUN	Running total of runoff over a given DELSUB period (units: cm)
SUBSTP	Running total of time steps during a given DELSUB period
SUBTIM	Running total of actual simulated time (units: h)
SUBTRA	Running total of transpiration over a given DELSUB period (units: cm)
TDONE	Time that has been simulated (done) in a day (units: h)
TE	Running total of total yearly evaporation (units: cm)
TEND	Ending time of the current water application rate (units: h)
TERR	Mass balance error at the end of the simulation (units: cm)
TETRA	Running total of evaporation on days when transpiration occurs (units: cm)
TH	Temporary storage for THETA(I)
THD	Short version of THETAD(MAT(I))
THDIF	The percent change in THETA for a given node over a time step; used to control the time step
THMAX	The maximum of THETA and TTHETA, used to calculate THDIF
THN	Short version of THETAN(MAT(I))
THTR	Residual water content, used in Haverkamp moisture characteristic description (Haverkamp et al. 1977) (units: cm^3/cm^3)
THW	Short version of THETA(MAT(I))
TINF	Running total of yearly infiltration (units: cm)
TITLE	Character variable containing the problem title from the input file
TLEFT	Time left until the end of the next time period or flow rate change (units: h)
TMOIST	Profile moisture storage at the end of an accepted time step and the end of the day, referred to as STORAGE in the output (units: cm)
TORT	Pore tortuosity, used when IVAPOR = 1 [in DATAINH]
TOTALR	Total water application amount (as it is input) for the day (units: cm)
TPE	Running total of yearly potential evaporation during non-infiltration periods (units: cm)
TPET	Running total of daily PET values (units: cm)
TPREV	Profile moisture storage (TMOIST) of the previous day, referred to as PRESTOR in the output (units: cm)
TPT	Running total of potential transpiration during non-infiltration periods (units: cm)
TRAIN	Running total of daily rainfall as applied to the surface (units: cm)
TRANSP	Transpiration rate (units: cm/h)
TRUNOF	Running total of daily runoff amounts (units: cm)
TSOIL	Average soil temperature (units: °K)
TSTART	Starting time of the first rainfall/irrigation event for that day (units: h)
TSTOP	Stopping time of the last rainfall/irrigation event for that day (units: h)
TSTP	Running total of time steps taken since the beginning of the simulation, obtained by summing NSTEP values at the end of each day
TSUB	Running total of time completed in a given DELSUB loop (units: h)
TTIM	Running total of actual time simulated (units: h)
TTIRRI	Running total of daily irrigation as applied to the surface (units: cm)
TTRA	Running total of daily transpiration (units: cm)

UP	Upstream weighting factor for estimating the conductivity at nodal midpoints, used when KEST = 2
UPPER	Flag that indicates the surface boundary condition; it can change while in the p-c loop (0) flux (1) constant head
VAPCOE	Combination of variables used in vapor flow calculations (units: cm/h)
VAPDEN	Saturated vapor density at the soil temperature TSOIL (units: g/cm ³) [in DATAINH]
VAPDIF	Diffusion coefficient of water vapor in air (units: cm ² /s) [in DATAINH]
WATDEN	Density of water at the soil temperature TSOIL (units: g/cm ³) [in DATAINH]
WCOMPR	Compressibility of water, found in POLYCH (units: 1/cm)
WINDEX	Index of water application day
Z1	Part of the solution matrix
Z11	Part of the solution matrix
Z111	Part of the solution matrix
Z31	Part of the solution matrix
ZI	Shortened version of ZZ(I)
ZK1	Part of the solution matrix
ZK2	Part of the solution matrix
ZN	Part of the solution matrix
ZNN	Part of the solution matrix
ZNNN	Part of the solution matrix
ZZK1	Part of the solution matrix
ZZK2	Part of the solution matrix

ARRAYS

A1	Part of the solution matrix
A2	Part of the solution matrix
A3	Part of the solution matrix
AIRINK	Air entry pressure of the conductivity function (units: cm)
AIRINT	Air entry pressure of the soil moisture characteristic (units: cm)
AMDUNT	Water application amount (units: cm)
C	Present specific water capacity (units: 1/cm)
CC	Specific water capacity at the end of the previous time step (units: 1/cm)
CREGKH	Regression coefficients for polynomials used to describe the hydraulic conductivity, KEST = 1
CREGTH	Regression coefficients for polynomials used to describe the soil moisture characteristic, KEST = 1
DAYQ	Running total of flux amounts between nodes during the day (units: cm)
DAYSINK	Running total of the sink terms for each node during the day (units: 1)
FLAI	Leaf area index on the corresponding NGROW day [in DATAINH]
FPET	Hourly PET distribution factors

GGMOD	Value of GMOD at the end of the last successful time step
GMOD	Ratio of liquid conductivity to total conductivity when IVAPOR = 1
H	Present soil moisture pressure head value (units: cm)
HH	Soil moisture pressure head value at the end of the previous time step (units: cm)
IHY	Array containing all of the soil hydraulic property data
KV	Vapor conductivity when IVAPOR = 1 (units: cm/h)
MAT	Identifies the soil material of the specified node
NDEGKH	Number of polynomial coefficients for describing hydraulic conductivity in POLYKH
NDEGTH	Number of polynomial coefficients for describing the moisture characteristic in POLYCH and POLYTH
NGROW	Day of the year on which the LAI is changed to the corresponding FLAI value [in DATAINH] (units: day)
NQLEAK	Day on which the bottom boundary flux is changed to the corresponding QLEAK value [in DATAINH] (units: day)
NSUBKH	Number of polynomials used to describe hydraulic conductivity in POLYKH
NSUBTH	Number of polynomials used to describe the soil moisture characteristic curve in POLYCH and POLYTH
NROOT	Growth day on which roots reach the given node
PEVAPO	Potential evaporation rate for the day (units: cm/day)
PTRANS	Potential transpiration rate for the day (units: cm/day)
Q	Flux amounts between nodes for each DELSUB period (units: cm)
QLEAK	Bottom boundary flux on the corresponding NQLEAK day [in DATAINH] (units: cm/h)
RATE	Rate of rainfall for rainfall event IR (units: cm/h)
RDF	Root density function (Units: 1/cm)
RTIME	Time of day when the water application rate changes (units: h)
SINK	Rate of water removal by plants from a given node (units: 1/h)
SK	Saturated hydraulic conductivity (units: cm/h)
SLOPKH	Linear slope between the conductivity values at heads of zero and airentry; used in POLYKH (units: 1/h)
SLOPTH	Linear slope between the water content at heads of zero and airentry; used in POLYCH and POLYTH (units: 1/cm)
SUBQ	Running total of flux amounts between nodes during the present DELSUB period (units: cm)
SUBSNK	Hourly summary of sink term for each node (units: 1)
THET	Saturated water content (units: cm ³ /cm ³)
THETA	Water content (units: cm ³ /cm ³)
THETAD	Water content below which roots will remove water at less than the optimum rate (NUPTAK = 1) (units: cm ³ /cm ³)
THETAN	Water content above which anaerobic conditions exist and no water will be withdrawn by plants (NUPTAK = 1) (units: cm ³ /cm ³)
THETAW	Water content below which roots will not withdraw water (NUPTAK = 1) (units: cm ³ /cm ³)
THICK	Node thickness (units: cm)
TQ	Yearly summary of flux amounts (OAYQ) between nodes (units: cm)
TTHETA	Soil moisture content at the end of the previous time step (units: cm ³ /cm ³)
WQLEAK	Specified bottom boundary flux when LOWER = 4 (units: cm/h)

XDIVKH	Upper head boundary of region where given polynomial is valid (units: cm)
XDIVTH	Upper head boundary of region where given polynomial is valid (units: cm)
Z	Depth of node below the surface (units: cm)
ZK	Hydraulic conductivity (units: cm/h)
ZKMID	Hydraulic conductivity at the midpoint between nodes (units: cm/h)
ZY	Part of the solution matrix
ZZ	Distance from node I to node I+1 (units: cm)
ZZK	Hydraulic conductivity at the end of the previous time step (units: cm/h)
ZZKMID	Hydraulic conductivity at the midpoint between nodes at the end of the previous time step (units: cm/h)

PARAMETER CONSTANTS

LUB	Logical unit for binary output
LUI	Logical unit for binary input
LUR	Logical unit for interactive input
LUS	Logical unit for interactive output
M1	Dimension of node arrays

FUNCTIONS AND SUBROUTINES REFERENCED

DATE	Function that returns the date (specific to VAX Fortran)
DELCHK	Subroutine to adjust the time step according to the maximum percent change in THETA (ISWDIF = 0)
FUNED	Function that returns the saturated vapor pressure (g/cm^3) at the given temperature ($^{\circ}\text{K}$) [in DATAINH]
POLYCH	Subroutine to calculate specific water capacities
POLYKH	Subroutine to calculate hydraulic conductivities
POLYTH	Subroutine to calculate water contents
RHO	Function to calculate the root density
TIME	Function that returns the time (specific to VAX Fortran)
TRIDAG	Subroutine containing the matrix solver
TTLEFT	Subroutine to calculate the time left until the end of the next time period or flow rate change

REFERENCES

- Feddes, R. A., P. J. Kowalik and H. Zaradny. 1978. Simulation of Field Water Use and Crop Yield. John Wiley and Sons, Inc., New York.
- Gupta, S. K., K. K. Tanji, D. R. Nielsen, J. W. Biggar, C. S. Simmons and J. L. MacIntyre. 1978. "Field Simulation of Soil-Water Movement with Crop Water Extraction." Water Science and Engineering Paper No. 4013. Department of Land, Air, and Water Resource, University of California, Davis.
- Haverkamp, R., M. Vauclin, J. Touma, P. J. Wierenga and G. Vachaud. 1977. "A Comparison of Numerical Simulation Models for One-Dimensional Infiltration." Soil Sci. Soc. Am. J. 41:285-294.
- Hinds, W. T. 1975. "Energy and Carbon Balances in Cheatgrass: An Essay in Autecology." Ecological Monogr. 45:367-388.
- Ritchie, J. T., and E. Burnett. 1971. "Dryland Evaporative Flux in a Subhumid Climate, 2, Plant Influences." Agron. J. 63:56-62.

DISTRIBUTION

No. of
Copies

No. of
Copies

OFFSITE

M. J. Barainca, Program Manager
Low-Level Waste Management
Program
Idaho Operations Office
U.S. Department of Energy
550 Second Street
Idaho Falls, ID 83401

R. Boland
Waste Management Project
Office
Nevada Operations Office
U.S. Department of Energy
P.O. Box 14100
Las Vegas, NV 89114

T. C. Chee
R&D and Byproducts Division
DP-123 (GTN)
U.S. Department of Energy
Washington, DC 20545

B. W. Church, Director
Health Physics Division
Nevada Operations Office
U.S. Department of Energy
P.O. Box 14100
Las Vegas, NV 89114

F. Gorup, Program Manager
Nuclear Waste Management Group
Chicago Operations Office
U.S. Department of Energy
9800 South Cass Avenue
Argonne, IL 60439

J. J. Jicha, Director
R&D and Byproducts Division
DP-123 (GTN)
U.S. Department of Energy
Washington, DC 20545

E. A. Jordan
Low-Level Waste Program
Manager
Division of Storage and
Treatment Projects
NE-25 (GTN)
U.S. Department of Energy
Washington, DC 20545

D. B. Leclaire, Director
Office of Defense Waste
and Byproducts Management
DP-12 (GTN)
U.S. Department of Energy
Washington, DC 20545

O. D. Markham
U.S. Department of Energy
Idaho Operations Office
Radiological and Environmental
Sciences Laboratory
Idaho Falls, ID 83401

R. L. Walters
Ecological Research Division
ER-75
Office of Health and Environ-
mental Research, OER
U.S. Department of Energy
Washington, DC 20545

30 DOE Technical Information Center

Technical Library
Argonne National Laboratory
Argonne, IL 60439

Peter Colombo, Group Leader
Nuclear Waste Research
Brookhaven National Laboratory
Building 701
Upton, NY 11973

No. of
Copies

Marty Mifflin
Desert Research Institute
1500 E. Tropicana Ave.
Suite 210
Las Vegas, NV 89109

E. A. Jennrich
Low-Level Management
Program
EG&G Idaho, Inc.
P.O. Box 1625
Idaho Falls, Idaho 83415

Environmental Protection Agency
Technology Assessment Division
Office of Radiation Programs
Washington, D.C. 20460

Technical Library
Idaho National Engineering
Laboratory
Idaho Falls, Idaho 83401

T. E. Hakonson
Los Alamos National Laboratory
Los Alamos, NM 87545

J. W. Nyhan
Los Alamos National Laboratory
P. O. Box 1663
Los Alamos, NM 87545

J. G. Steger
Los Alamos National Laboratory
P. O. Box 1663
Los Alamos, NM 87545

Technical Library
Los Alamos National Laboratory
Los Alamos, New Mexico 87545

National Academy of Sciences
National Research Council
2101 Constitution Avenue
Washington, DC 20418

No. of
Copies

Dan Stephens
Dept. of Geosciences
New Mexico Tech
Socorro, NM 87801

J. S. Baldwin
Low-Level Waste Management
Program
Oak Ridge National Laboratory
P.O. Box X
Oak Ridge, TN 37830

P. J. Wierenga
Dept. of Soil Science
New Mexico State University
Las Cruces, NM 88004

L. J. Mezga, Program Manager
Low-Level Waste Management
Program
Oak Ridge National Laboratory
P.O. Box X
Oak Ridge, TN 37830

Technical Library
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37830

E. W. Kendall
Waste Management Project Manager
REECO
P.O. Box 642
Mercury, NV 89023

J. J. Blakeslee, Program Manager
Nuclear Waste Processing
Rockwell International
P.O. Box 464
Golden, CO 80401

C. E. Wickland
Manager, Waste Operations
Rockwell International
Rocky Flats Plant
P.O. Box 464
Golden, CO 80401

No. of
Copies

J. C. Corey
Savannah River National
Laboratory
P.O. Box A
Aiken, SC 29801

I. W. Marine
Savannah River National
Laboratory
Box A
Aiken, SC 29801

Technical Library
Savannah River National
Laboratory
Aiken, SC 29081

E. M. Romney
University of California
at Los Angeles
Westwood, CA 96137

W. A. Jury
Department of Soils
University of California
at Riverside
Riverside, CA 92502

J. W. Cary
U.S. Department of Agriculture
Snake River Conservation
Research Center
Route 1, Box 186
Kimberly, ID 83341

Jonathan Herrmann
Office of Solid Waste
U.S. Environmental Protection
Agency
Municipal Environmental Research
Laboratory
Cincinnati, OH 45268

No. of
Copies

Jack Fischer
Low-Level Radioactive Waste
Program
U.S. Geological Survey
Water Resources Division
12201 Sunrise Valley Drive
Reston, VA 22092

I. J. Winograd
U.S. Geological Survey
National Center - Mail Stop 432
Reston, VA 22092

E. P. Weeks
U.S. Geological Survey
Federal Center Mail Stop 413
Denver, CO 80225

E. O'Donnell
Earth Sciences Branch
Division of Health, Siting
and Waste Management Research
U.S. Nuclear Regulatory
Commission
Washington, DC 20555

F. Swanberg
Division of Health, Siting
and Waste Management Research
U.S. Nuclear Regulatory
Commission
Washington, DC 20555

T. J. Nicholson
U.S. Nuclear Regulatory
Commission
Div. Health, Siting and Waste
Management
MS 1130 SS
Washington, DC 20555

No. of
Copies

Jeanne Rensel
Hi-Level Waste Management
Washington State Department
of Ecology
Pu II
Olympia, WA 98504

G. S. Campbell
Washington State University
Pullman, WA 99164

ONSITE

Hanford Engineering Development
Laboratory

R. E. Lerch

7 Richland Operations Office

E. A. Bracken
R. A. Holten
N. T. Karagianes
H. E. Ransom
J. D. White
J. K. W. Wukelic

21 Rockwell Hanford Operations

M. R. Adams
J. F. Albaugh
S. Baker
G. A. Beitel
L. C. Brown
W. H. Chapman-Riggsbee
C. DeFigh-Price
W. F. Heine
W. A. Jordan
G. G. Meade
S. J. Phillips
R. D. Prosser
J. F. Relyea
J. H. Roecker

No. of
Copies

R. C. Routson
D. L. Sanders
W. W. Schulz
J. C. Sonnichsen
S. A. Wiegman
G. F. Williamson
D. E. Wood

United Nuclear Industries

D. R. Pratt

47 Pacific Northwest Laboratory

P. A. Beedlow
J. B. Burnham
D. W. Dragnich
C. E. Elderkin
M. J. Fayer (10)
G. W. Gee (5)
M. J. Graham
H. A. Haerer
J. M. Hales
M. S. Hanson
J. N. Hartley
P. C. Hays
P. R. Heller
T. L. Jones
C. T. Kincaid
R. R. Kirkham
G. V. Last
I. C. Nelson
A. E. Reisenauer
L. E. Rogers
R. J. Serne
C. S. Simmons
D. R. Simpson
R. L. Skaggs
J. A. Stottlemire
G. P. Streile
R. E. Wildung
Technical Information (5)
Publishing Coordination (2)