R83-3

SWAHP: A SOIL WATER AND HEAT PARAMETERIZATION

by

P. Christopher D. Milly

RALPH M. PARSONS LABORATORY
HYDROLOGY AND WATER RESOURCE SYSTEMS

Report Number 289

This Material is Based Upon Work Supported by the National Science Foundation Under Grant Numbers ATM-7812327 and ATM-8114723

February 1983

MIT

DEPARTMENT
OF
CIVIL
ENGINEERING

SCHOOL OF ENGINEERING
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Cambridge, Massachusetts 02139

R83-3

SWAHP: A SOIL WATER AND HEAT PARAMETERIZATION

by

P. Christopher D. Milly

RALPH M. PARSONS LABORATORY

HYDROLOGY AND WATER RESOURCE SYSTEMS

Report Number 289

This Material is Based Upon Work Supported by the National Science Foundation Under Grant Numbers ATM-7812327 and ATM-8114723

February 1983

SWAHP: A SOIL WATER AND HEAT PARAMETERIZATION

Abstract

SWAHP is an efficient algorithm for prediction of moisture and heat fluxes across a bare soil surface. Surface moisture flux is calculated using the time compression assumption in conjunction with special solutions of the nonlinear moisture diffusion equation. Effects of water vapor and hysteresis are incorporated. The average moisture concentration near the surface is calculated continuously, and redistribution is considered. Potential evaporation is calculated using the surface temperature given by the force-restore procedure.

SWAHP has been programmed in Fortran for computer solution, and the program is documented in detail. Required input consists of soil physical properties and atmospheric forcing data.

ACKNOWLEDGEMENTS

This work was performed with the support of the National Science Foundation under research Grant Numbers ATM-7812327 and ATM-8114723.

The completion of this work was supervised by Dr. Peter S. Eagleson, Professor of Civil Engineering, whose contributions are appreciated.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF VARIABLES

Most of the variables introduced in Chapter 2 and in the Appendices are tabulated here. Not listed are the Fortran names introduced in Chapters 3 and 4. For these, refer to Table 3.2 and the list in Appendix E.

Symbols	Description
A	Soil surface albedo
A_o	Constant term in f_i^*
C	Effective heat capacity of soil
C_H	Heat transfer coefficient
C_w	Vapor transfer coefficient
c_1, c_2, c_3	Force-restore coefficients
c_ℓ	Specific heat of liquid water
c_p	Specific heat of water vapor at constant pressure
D	$K \frac{d\psi}{d\Theta}$, soil moisture diffusivity
D_e	Effective constant D for exfiltration
D_i	Effective constant D for infiltration
$D_{\psi v}$	Effective hydraulic conductivity due to vapor diffusion
E	Rate of evaporation
E_p	Potential rate of evaporation
F_e	Cumulative exfiltration depth
F_e'	Apparent cumulative exfiltration depth
F_i	Cumulative infiltration depth
f	Proportion of annual precipitation reaching deep soil
f_e	Rate of exfiltration

Symbols	Description
f_e^*	Exfiltration capacity
f_i	Rate of infiltration
f_i^*	Infiltration capacity
G	Heat flux into the ground
g	Acceleration of gravity
$I_{\ell d}$	Downward atmospheric longwave radiation
I_s	Incoming solar radiation
K	Hydraulic conductivity
K_r	Relative hydraulic conductivity
L	Latent heat of vaporization of water
L	Monin-Obukhov length
L_o	Value of L at freezing point
N_d	Number of days in year
n	Porosity
P	Precipitation rate
$\bar{P}$	Annual average precipitation as a rate
R	Redistribution parameter
R_s	Surface runoff rate
R_v	Gas constant for water
S_o	Sorptivity in the basic infiltration solution
S_e	Desorptivity
S_i	Sorptivity
T_1	Surface temperature
T_2	Deep soil temperature
T_a	Air temperature

Symbols	Description
t	Time
t_c	Compressed time
t_r	Storm duration
U_a	Windspeed at screen height
Z	Depth of surface wetted zone
z	Distance below surface
z_o	Surface roughness
z_a	Screen height
ε	Emissivity of soil surface
Θ	Volumetric moisture content
Θ_o	Initial Θ in diffusion solutions
Θ_1	Surface Θ in diffusion solutions
Θ_1	Surface moisture content in SWAHP
$\Theta_{pd}(\psi)$	Primary drying scanning curve
Θ_r	Value of Θ at reversal from wetting to drying
Θ_u	Moisture content of soil when freely re-wetted
$\Theta_w(\psi)$	Main wetting curve
$\bar{\Theta}$	Near surface Θ due to infiltration and redistribution
$\hat{\Theta}$	Θ used to evaluate deep thermal properties
λ	Effective thermal conductivity of soil
ρ_o	Saturation absolute humidity
ρ_a	Density of air
ρ_ℓ	Density of liquid water
ρ_{va}	Absolute humidity of air

Symbols	Description
σ	Stefan-Boltzman constant
τ	Period of one day
ψ	Matric potential
$\bar{\psi}$	Value of ψ corresponding to $\bar{\Theta}$
Ω	Tortuosity of air phase

Chapter 1

INTRODUCTION

This document describes a computer program called SWAHP (an acronym for Soil Water and Heat Parameterization). SWAHP is based on an algorithm proposed by Milly and Eagleson (1982) for the calculation of moisture and heat fluxes across a bare land surface under natural atmospheric excitation. The algorithm is an approximate procedure that was developed through a systematic simplification of the detailed partial differential equations governing the vertical flow of water and heat in the soil. A simulation using SWAHP requires orders of magnitude less computer time than the equivalent set of calculations using a more exact finite element model.

SWAHP has been developed for potential use in conjunction with an atmospheric general circulation (GCM), but could also be valuable in any analysis that requires the prediction of land surface moisture and heat fluxes as a response to atmospheric conditions. As presented here, it calculates the dynamic response of the land surface - specifically, evaporation, sensible heat flux and longwave radiation - as a function of past and present atmospheric forcing - temperature and humidity of the air, incoming radiation, etc. Input and output are typically given on an hourly basis. The soil, assumed homogeneous, is described in terms of the conventional variables of soil physics - moisture retention curve, hydraulic conductivity, etc.

The hydraulic behavior of the soil is modeled using nonlinear diffusion theory and the time compression assumption. Thermal behavior, which is essentially linear and which has a dominant diurnal component, is parameterized using the force-restore procedure.

The combination of a sound physical basis and computational efficiency make SWAHP a potential tool in many soil moisture analyses. Examples include parameter and/or state estimation by remote sensing and analysis of spatial variability.

Chapter 2

THE THEORY OF SWAHP

2.1 Intoduction

The theory underlying the computer program SWAHP is presented in this chapter. An earlier presentation of the theory was given by Milly and Eagleson (1982), in portions of a document having wider scope. This chapter is an attempt to bring that material, plus further extensions and clarifications, all together in one convenient reference.

SWAHP is an event-based simulation model. The continuous time dimension is viewed as a connected stream, or sequence, of time intervals during which it is either raining or not raining (Figure 2.1). These intervals are termed, respectively, storm events and interstorm events. Flux of moisture into the soil during a storm event is termed infiltration.

For both types of events, we present an existing solution for the surface moisture flux. Both solutions are based on idealized formulations using the nonlinear equation for diffusion of soil moisture. In each case, the idealization relies partially on very simple initial and boundary conditions.

The two flux solutions are the basic units employed by SWAHP in the prediction of evaporation and infiltration for any series of events. The time compression assumption is the framework within which we apply the idealized infiltration and exfiltration models to situations having boundary conditions different from those assumed in their derivations. No such elegant method exists for the application of our basic solutions to

FIGURE 2.1 Storm and Interstorm Events Horizontal bars indicate durations of storm events, and vertical bars indicate transitions between events. Absence of horizontal bars indicates interstorm events. Five separate, complete events are pictured

problems with non-constant initial conditions. For infiltration during storm events, the initial moisture content is taken here to be zero. For exfiltration between storms, the initial moisture content near the surface is estimated, on the basis solely of the preceeding storm event, using a variation of the basic infiltration solution. Using only this value, however, would eventually lead to errors as time elapses. This would be due first to the actual decrease in near-surface moisture that results from redistribution, or soil drainage, a factor not considered in the derivation of the basic solution. We propose a means for parameterizing this redistribution process and for modifying the basic solution accordingly.

If evaporation continues long enough, a second problem arises as the moisture availability should be governed no longer by the most recent storm, but rather by earlier storms through their cumulative influence on the moisture content at greater depth. A simple memory model is proposed that bases the moisture availibility for evaporation on successively earlier events when an interstorm event continues uninterrupted.

Specification of the surface boundary condition for moisture during any event in SWAHP requires knowledge of the potential evaporation rate. This is dependent upon the surface temperature, so an energy balance is performed. Conduction of heat into the ground is parameterized using the force-restore procedure.

2.2 Infiltration During Storm Events

2.2.1 The Basic Infiltration Solution

Philip (1957) solved the nonlinear moisture diffusion equation

$$\frac{\partial \Theta}{\partial t} = \frac{\partial}{\partial z} \left(D \frac{\partial \Theta}{\partial z}\right) - \frac{\partial K}{\partial z} \tag{2.1}$$

subject to initial and boundary conditions

$$\Theta = \Theta_0 \qquad z > 0 \qquad t = 0 \tag{2.2}$$

$$\Theta = \Theta_1 \qquad z = 0 \qquad t > 0 \tag{2.3}$$

with

$$D = D(\Theta) \tag{2.4}$$

$$K = K(\Theta) \tag{2.5}$$

where Θ is the moisture content, K is the hydraulic conductivity, and Θ_1 is greater than Θ_0. D is a diffusion coefficient given by

$$D = K/\frac{d\Theta}{d\psi} \tag{2.6}$$

where ψ is the matric potential. Because of hysteresis, $\Theta(\psi)$ is not generally unique. In order for (2.4) to hold, Philip (1957) implicitly assumes that the wetting history of the soil is homogeneous in space. His solution yields an expression for the rate of infiltration having the form

$$f_i = \frac{1}{2} A_1 t^{-\frac{1}{2}} + A_2 + K(\Theta_0) + \frac{3}{2} A_3 t^{\frac{1}{2}} + 2 A_4 t + \ldots \tag{2.7}$$

in which the $A_i = A_i(\Theta_0, \Theta_1)$ may be evaluated on the basis of the D and K functions. For practical application, Philip (1957) suggests that the $t^{-\frac{1}{2}}$ and constant terms of (2.7) may be sufficient. Thus

$$f_i = \frac{1}{2} A_1 t^{-\frac{1}{2}} + A_2 + K(\Theta_0) \tag{2.8}$$

One desirable feature of (2.8) is that it is equivalent to (2.7) for early time. A second is that it correctly predicts the existence of an asymptotic, constant value of f_i at large time, although the value predicted, $(A_2 + K(\Theta_0))$, is lower than the actual rate, which is $K(\Theta_1)$ (Philip, 1969, p. 275).

The leading term in (2.8) is the entire solution for f_i in the absence of gravity (i.e., when the final term in (2.1) is excluded). The coefficient A_1 is called the sorptivity and can be expressed in terms of an effective infiltration diffusivity, D_i, as

$$A_1(\Theta_0, \Theta_1) = S_i(\Theta_0, \Theta_1) = 2(\Theta_1 - \Theta_0)[D_i(\Theta_0, \Theta_1)/\pi]^{\frac{1}{2}} \tag{2.9}$$

Crank (1975, p. 250) gives an empirical expression for $D_1(\Theta_0, \Theta_1)$. It is

$$D_i(\Theta_0, \Theta_1) = 1.67(\Theta_1 - \Theta_0)^{-1.67} \int_{\Theta_0}^{\Theta_1} (\Theta - \Theta_0)^{.67} D(\Theta)\, d\Theta \tag{2.10}$$

Philip (1969, p. 251), in discussing the convergence limit of (2.7), notes that A_2 has a value given approximately by

$$A_2 = \frac{1}{2}[K(\Theta_1) - K(\Theta_0)] \tag{2.11}$$

We follow Eagleson (1978) in using this value in the infiltration equation. This yields

$$f_i(t) = \frac{1}{2} S_i(\Theta_0, \Theta_1) t^{-\frac{1}{2}} + \frac{1}{2} [K(\Theta_1) + K(\Theta_0)] \tag{2.12}$$

Our basic solution for infiltration is a special case of (2.12), the case of an initially dry soil and a freely wetted surface, i.e.,

$$\Theta_0 = 0 \tag{2.13}$$

$$\Theta_1 = \Theta_u \tag{2.14}$$

where Θ_u is the maximum value of moisture content upon re-wetting a completely drained soil. It is somewhat less than the porosity, n, due to air entrapment. Mualem (1974) suggests

$$\Theta_u = 0.9\ n \tag{2.15}$$

It follows that $\Theta(\psi)$ should be evaluated along the main wetting curve,

$$\Theta(\psi) \equiv \Theta_w(\psi) \tag{2.16}$$

To summarize, the basic infiltration solution is

$$f_i(t) = \frac{1}{2} S_0\ t^{-\frac{1}{2}} + A_0 \tag{2.17}$$

where we use (2.9), (2.10), (2.6), (2.13), (2.14), and (2.17), together with a change of the integration variable, to define

$$S_0 = \left\{2.13\ \Theta_u^{0.33} \int_{-\infty}^{0} [\Theta_w(\psi)]^{0.67}\ K[\Theta_w(\psi)]d\psi\right\}^{\frac{1}{2}} \tag{2.18}$$

and where we use (2.12), (2.13) and (2.14), together with the fact that K is zero in a dry soil, to obtain

$$A_0 = \frac{1}{2} K(\Theta_u) \tag{2.19}$$

Note that S_0 and A_0 are true soil constants, which need be calculated only once.

The cumulative flux associated with the basic infiltration solution is given by

$$\begin{aligned} F_i(t) &\equiv \int_0^t f_i(\tau)\ d\tau \\ &= S_0\ t^{\frac{1}{2}} + A_0\ t \end{aligned} \tag{2.20}$$

2.2.2 Infiltration Capacity and the Time Compression Assumption

A common model for infiltration during a storm has the form

$$f_i(t) = \min[f_i^*(t), P(t) - E_p(t)] \tag{2.21}$$

in which f_i^* is called the infiltration capacity and P is the precipitation rate. For generality, we have included the potential evaporation rate, E_p, also. We shall apply this model when it is raining and $(P - E_p)$ is non-negative. Occasionally, E_p will exceed P during a storm. This situation is dealt with Section 2.4.1.

The time compression assumption (TCA) specifies that f_i^* is a unique function of F_i, the cumulative infiltration, regardless of the actual time distribution of f_i. The approximate validity of the TCA for infiltration has been demonstrated by Ibrahim and Brutsaert (1968) and by Reeves and Miller (1975).

Since the $f_i^*(F_i)$ relation is assumed to be unique, it may be deduced from any set of boundary conditions. In particular, we may use the case of our basic solution. Since the soil surface was considered to be wetted freely, the solution for f_i is identical to f_i^* for that case. We therefore have, from (2.17) and (2.20),

$$f_i^*(F_i) = A_0 \left\{1 + \left[-1 + \left(1 + \frac{4\ A_0\ F_i}{S_0^2}\right)^{\frac{1}{2}}\right]^{-1}\right\} \tag{2.22}$$

The compressed time, t_c, may be defined as the time that would have elapsed if the present state (F_i or f_i^*) had been reached under the boundary conditions of the basic solution. This is equivalent to t in (2.17) and (2.20). It is given by

$$t_c(t) = \left\{\frac{S_0}{2[f_i^*(t)-A_0]}\right\}^2 \qquad (2.23)$$

or

$$t_c(t) = \left\{\frac{S_0}{2A_0}\left[-1 + \left(1 + \frac{4\ A_0F_i(t)}{S_0^2}\right)^{\frac{1}{2}}\right]\right\}^2 \qquad (2.24)$$

With (2.21) and (2.22), we can compute the infiltration rate during a storm, provided P(t), E(t), A_0, and S_0 are known. Normally, E(t) will be equal to the potential evaporation rate $E_p(t)$, the rare exception (Section 2.4.1) occurring when P(t) is smaller that $E_p(t)$ and the soil surface is very dry. Calculation of $E_p(t)$ is discussed later. Given $f_i(t)$ and ignoring surface depression and detention storage, we can calculate the rate of production of surface runoff, R_s, as

$$R_s(t) = P(t) - E(t) - f_i(t) \qquad (2.25)$$

2.3 Redistribution and Exfiltration During Interstorm Events

2.3.1 Distribution and Redistribution of Soil Moisture Following a Storm

Evaporation and exfiltration between storms are sometimes limited by the rate at which water can reach the surface or simply by the amount of water in storage. In order to quantify these limitations, it is useful to have an estimate of the amount of water stored near the soil surface. Empirical and analytic evidence shows that there is a zone of wet soil near the surface immediately following a storm and that this zone grows deeper and less wet - first rapidly, then more slowly - as time elapses following the storm. In this section, we propose a method for dynamic estimation of the near-surface soil moisture content.

Following a storm event, we conceptualize a soil moisture distribution as depicted in Figure 2.2. There is a uniform moisture content, $\bar{\Theta}_1$, between the surface and depth Z_1. The total equivalent depth of water in this square wave is F_i, the cumulative infiltration depth. The value $\bar{\Theta}_1$ is estimated to be the moisture content that, applied at the surface of a dry soil for the duration t_r of the proceeding storm, would have induced total infiltration equal to F_i. The resulting implicit equation for this initial $\bar{\Theta}$ is, in analogy to (2.20),

$$F_i = S_i(0,\ \bar{\Theta})\ t_r^{\frac{1}{2}} + \frac{1}{2} K(\bar{\Theta}) t_r \tag{2.26}$$

Milly and Eagleson (1982) proposed that redistribution, in the absence of evaporation, be parameterized as a preservation of the square wave distribution, with $\bar{\Theta}$ decreasing, Z increasing, and $\bar{\Theta}Z$ remaining constant (Figure 2.2). They further propose that the downward flux of water at the bottom of the wave be a function only of $\bar{\Theta}$. Thus

$$Z \frac{d\bar{\Theta}}{dt} = -\ g(\bar{\Theta}) \tag{2.27}$$

or

$$\frac{d\bar{\Theta}}{dt} = -\frac{\bar{\Theta}}{F_i}\ g(\bar{\Theta}) \tag{2.28}$$

The function $g(\bar{\Theta})$ should parameterize the effects of gravity and capillary diffusion in transporting water downward following a storm. A truly physically-based formulation of $g(\bar{\Theta})$ in terms of nonlinear diffusion is difficult because both sorption and desorption are occurring simultaneously and because the boundary separating these two processes is moving. We shall, therefore, adopt an alternative, conceptual ap-

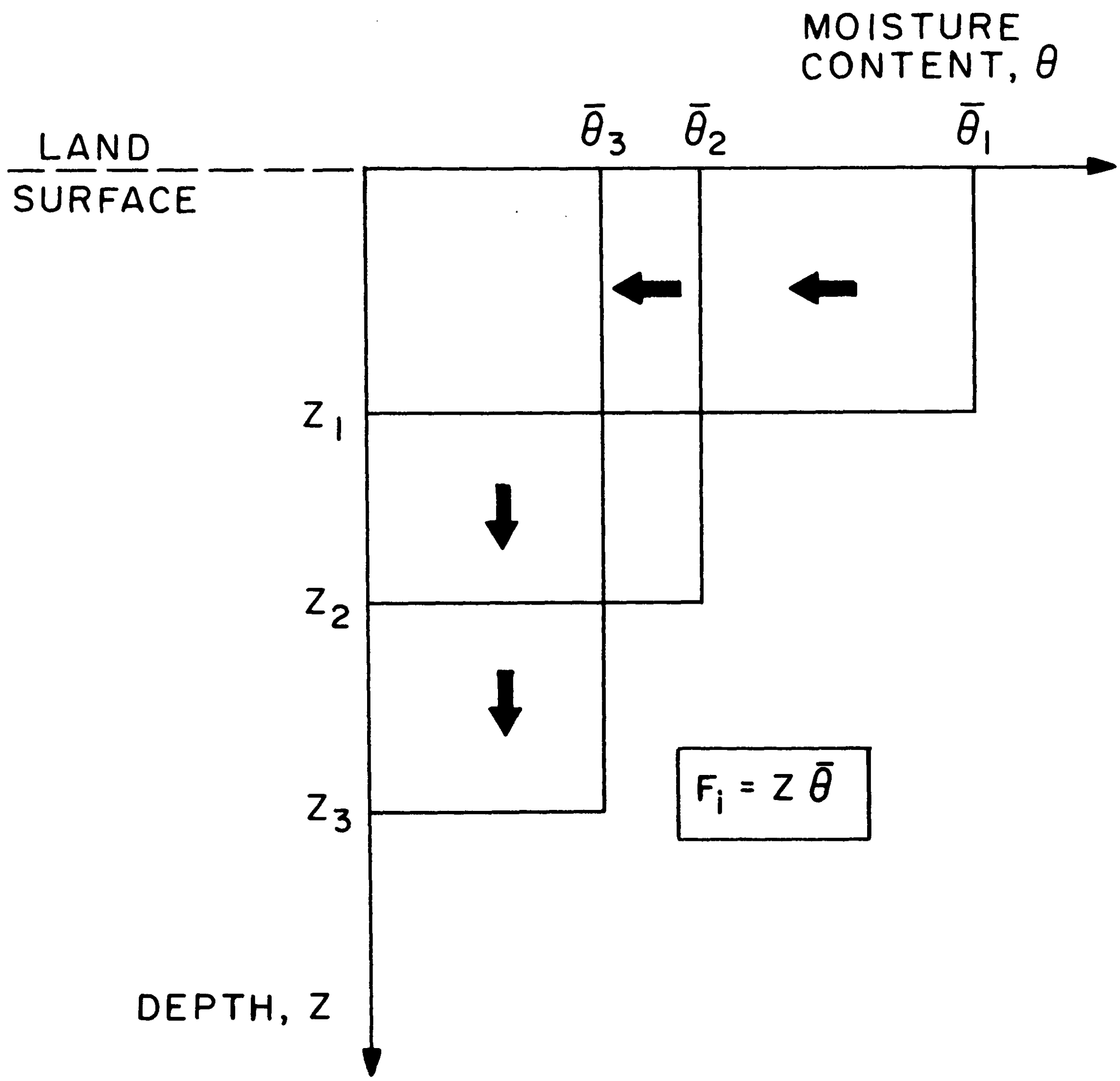

FIGURE 2.2 Soil Moisture Distribution and Redistribution after a Storm Subscript 1 indicates conditions immediately after storm, while 2 and 3 indicate redistribution at later times in the absence of evaporation

proach. Noting the predominant role of the hydraulic conductivity in both diffusion and gravity drainage of soil moisture, we scale the redistribution function $g(\bar{\Theta})$ by the hydraulic conductivity,

$$g(\bar{\Theta}) = R\ K(\bar{\Theta}) \tag{2.29}$$

where R, assumed constant, is called the redistribution parameter. Equations (2.28) and (2.29) yield

$$\frac{d\bar{\Theta}}{dt} = -\frac{R\ \bar{\Theta}\ K(\bar{\Theta})}{F_i} \tag{2.30}$$

Milly and Eagleson (1982) found that SWAHP, used with a value of R equal to 2, gave good predictions of evaporation from a sand and a silt loam under hot/arid and cool/humid atmospheric forcing regimes when compared to the predictions of a detailed finite element model.

2.3.2 The Basic Exfiltration Solution

Eagleson (1978) treats the problem of exfiltration using the formulation of (2.1) through (2.5) with Θ_1 less than Θ_0. By analogy with (2.12), but allowing for the opposition of gravity, Eagleson proposed that the exfiltration rate from a soil initially at moisture content Θ_0, dried with a moisture content of Θ_1 at the surface, be calculated using

$$f_e(t) = \frac{1}{2}\ S_e(\Theta_0,\ \Theta_1)t^{-\frac{1}{2}} - \frac{1}{2}[K(\Theta_1) + K(\Theta_0)] \tag{2.31}$$

in which f_e is the exfiltration rate and S_e is the desorptivity.

As was the case for the infiltration problem, the leading term in the flux equation is the entire solution in the absence of gravity. The desorptivity may also be expressed in terms of an effective diffusion coefficient,

$$S_e(\Theta_0, \Theta_1) = 2(\Theta_0 - \Theta_1) \left[D_e(\Theta_0, \Theta_1)/\pi\right]^{\frac{1}{2}} \tag{2.32}$$

Crank's (1975, p. 251) empirical expression for $D_e(\Theta_0, \Theta_1)$ is

$$D_e(\Theta_0, \Theta_1) = 1.85 \, (\Theta_0 - \Theta_1)^{-1.85} \int_{\Theta_1}^{\Theta_0} (\Theta_0 - \Theta)^{0.85} \, D(\Theta) \, d\Theta \tag{2.33}$$

The basic solution that we shall use for exfiltration is the case of (2.31) when the surface is kept dry and the initial moisture content is equal to $\bar{\Theta}$, the near surface soil moisture content,

$$\Theta_0 = \bar{\Theta} \tag{2.34}$$

$$\Theta_1 = 0 \tag{2.35}$$

For now let us ignore the dynamics of $\bar{\Theta}$; this problem is treated in Section 2.3.4. If we neglect the gravity term, which is usually small, in (2.31), we obtain

$$f_e(t) = \frac{1}{2} S_e(\bar{\Theta}, 0) \, t^{-\frac{1}{2}} \tag{2.36}$$

From (2.32) through (2.35) with a change of the integration variable, we find

$$S_e(\bar{\Theta}, 0) = \left\{2.35\bar{\Theta}^{.15} \int_{-\infty}^{\bar{\psi}} \left[\bar{\Theta} - \Theta_{pd}(\psi, \Theta_r)\right]^{0.85} \left[K(\Theta_{pd}(\psi, \Theta_r)) + D_{\psi v}(\Theta_{pd}(\psi, \Theta_r), \psi)\right] d\psi\right\}^{\frac{1}{2}} \tag{2.37}$$

In (2.37), we have chosen to evaluate the hysteretic $\Theta(\psi)$ relation by assuming that, prior to exfiltration, the soil mass was wetted from dryness to a moisture content Θ_r, at which point desorption began. Thus, $\Theta_{pd}(\psi, \Theta_r)$ is the primary drying scanning curve associated with a reversal at Θ_r. The quantity $D_{\psi v}$ is the effective hydraulic conductivity due to vapor diffusion. It can be expressed (Appendix A) in terms of the soil state and the tortuosity of the air phase, $\Omega(\Theta)$. It is included in (2.37), and not in (2.18), because it can be significant in magnitude relative to K only when Θ is small. In (2.37),

$$\Theta_{pd}(\bar{\psi}, \Theta_r) \equiv \bar{\Theta} \tag{3.38}$$

defines $\bar{\psi}$.

We use the relation of Mualem (1977), based on an independent domains model, to express

$$\Theta_{pd}(\psi, \Theta_r) = \Theta_w(\psi) \left[1 + \frac{\Theta_r - \Theta_w(\psi)}{\Theta_u}\right] \tag{3.39}$$

The desorptivity in (2.36) is, in fact, a function of $\bar{\Theta}$ and Θ_r. If we could assume that redistribution following rain were negligible, then $\bar{\Theta}$ and Θ_r would be identical and equal, say, to the value obtained from (2.26). These values are appropriate immediately after the rain ends, but according to (2.30), $\bar{\Theta}$, and hence S_e, will thereafter decrease due to redistribution. By definition, Θ_r does not change. After considering the application of the TCA to the evaporation problem with a constant desorptivity, we shall propose an extension to allow for a time-varying desorptivity.

2.3.3 Exfiltration Capacity and the Time Compression Assumption

We model evaporation according to

$$E(t) = f_e(t) = \min\,[f_e^*(t),\ E_p(t)] \tag{2.40}$$

when it is not raining and E_p is greater than or equal to zero. The case of dewfall is treated in Section 2.4.2. In (2.40), f_e^* denotes the exfiltration capacity. We shall employ the TCA in order to estimate f_e^*. Thus,

$$f_e^* = f_e^*(F_e) \tag{2.41}$$

where F_e is the cumulative exfiltration depth during the current event.

In order to evaluate $f_e^*(F_e)$, we use the basic exfiltration solution of Section 2.3.2. Combining (2.36) with its integral form, which is

$$F_e(t) = S_e(\bar{\Theta},\ 0)t^{\frac{1}{2}} \tag{2.42}$$

we find

$$f_e^*(F_e) = [S_e(\bar{\Theta},\ 0)]^2/2F_e \tag{2.43}$$

For exfiltration, the compressed time can be found from (2.36) to be

$$t_c = [S_e(\bar{\Theta},\ 0)/2f_e]^2 \tag{2.44}$$

or from (2.42) to be

$$t_c = [F_e/S_e(\bar{\Theta}, 0)]^2 \qquad (2.45)$$

Equations (2.42) through (2.45) are directly applicable only to the situation where S_e is a constant. We next consider an extended formulation for a dynamic S_e.

2.3.4 Desorptivity Decreases Due to Redistribution

The preceeding analysis is based on the assumption of a constant $\bar{\Theta}$, which is then identical to Θ_r. When we hypothesize that $\bar{\Theta}$ decreases with time, independently of evaporation, the analysis is no longer applicable. It is physically unrealistic to use the initial value of S_e, since this will overestimate the ability of the soil to deliver water to the surface. It is possible to use a constant value of S_e based on the value of $\bar{\Theta}$ that results from some fixed period of redistribution, but there appears to be no objective means for the determination of such a duration a priori

Our approach rests on the definition of an effective cumulative exfiltration depth, F_e'. We hypothesize the existence of the state F_e' satisfying the following conditions:

1. The exfiltration capacity is given by

$$f_e^*(F_e') = [S_e(\bar{\Theta}, 0)]^2/2F_e' \qquad (2.46)$$

i.e., independent of the history, current values of $\bar{\Theta}$ and F_e' yield the correct f_e^*. It is because of the similarity between (2.43) and (2.46) that we call F_e' the effective cumulative exfiltration depth.

2. F_e' is a function only of F_e and of (the dynamic) S_e,

$$F_e' = F_e'(F_e, S_e) \tag{2.47}$$

3. The exfiltration capacity given by (2.46), since it is physically controlled only by the slope of the moisture profile at the surface, is not directly influenced by changes in either the bulk moisture content $\bar{\Theta}$ or S_e when there is no actual exfiltration, i.e.,

$$\left.\frac{\partial f_e^*}{\partial S_e}\right|_{F_e} = 0 \tag{2.48}$$

4. For S_e static, f_e^* has the same sensitivity to F_e' as it does to F_e,

$$\left.\frac{\partial f_e^*}{\partial F_e'}\right|_{S_e} = \left.\frac{\partial f_e^*}{\partial F_e}\right|_{S_e} \tag{2.49}$$

We can deduce a prediction equation for F_e' from (2.47),

$$\frac{dF_e'}{dt} = \left.\frac{\partial F_e'}{\partial F_e}\right|_{S_e} \frac{dF_e}{dt} + \left.\frac{\partial F_e'}{\partial S_e}\right|_{F_e} \frac{dS_e}{dt} \tag{2.50}$$

To evaluate the first coefficient, we observe that, according to (2.46),

$$f_e^* = f_e^* (S_e, F_e') \tag{2.51}$$

Hence

$$\left.\frac{\partial f_e^*}{\partial F_e}\right|_{S_e} = \left.\frac{\partial f_e^*}{\partial F_e'}\right|_{S_e} \left.\frac{\partial F_e'}{\partial F_e}\right|_{S_e} \tag{2.52}$$

and then, using (2.49),

$$\left.\frac{\partial F'_e}{\partial F_e}\right|_{S_e} = 1 \tag{2.53}$$

To find the second coefficient in (2.50), we calculate, from (2.46),

$$\left.\frac{\partial f^*_e}{\partial S_e}\right|_{F_e} = \frac{S_e}{F'_e} - \frac{S_e^{\;2}}{2F'^{\,2}_e} \left.\frac{\partial F'_e}{\partial S_e}\right|_{F_e} \tag{2.54}$$

According to (2.48), this must be zero, so

$$\left.\frac{\partial F'_e}{\partial S_e}\right|_{F_e} = 2F'_e/S_e \tag{2.55}$$

Substituting (2.53) and (2.55) into (2.50), we have

$$\frac{dF'_e}{dt} = \frac{dF_e}{dt} + \frac{2F'_e}{S_e}\,\frac{dS_e}{dt} \tag{2.56}$$

or

$$\frac{dF'_e}{dt} = f_e + \frac{2F'_e}{S_e}\,\frac{dS_e}{dt} \tag{2.57}$$

Equation (2.57) gives F'_e, allowing calculation first of f^*_e by (2.46) and then of f_e by (2.40). Meanwhile, S_e is calculated from the dynamic $\bar{\Theta}$ using (2.37). In the evaluation of S_e, Θ_r takes the value of the moisture content $\bar{\Theta}$ in (2.26) that existed immediately following the storm.

2.3.5 Carryover Storage and Reversion to Earlier Interstorm Conditions

It is physically unrealistic to proceed indefinitely in time simply using (2.46) because it does not account for the finite depth, Z, of the square wave of magnitude $\bar{\Theta}$. Eventually F_e (actual exfiltration depth) may far exceed $Z\bar{\Theta}$, the square wave will have actually disappeared, and so there is no significance to the value of $\bar{\Theta}$ being used to evaluate S_e.

Consider the following example, illustrated in Figure 2.3. Rain wets a soil to depth Z_{1i} with moisture content $\bar{\Theta}_{1i}$. Subsequent evaporation amounts to F_{e1}, and redistribution leads to Z_{1f} and $\bar{\Theta}_{1f}$. A new storm wets the soil to depth Z_{2i} with moisture content $\bar{\Theta}_{2i}$. The evaporation event that follows is of sufficient duration for the new moisture wave to be completely consumed. Eventually, the moisture profile will show no sign of the latest storm, but rather appear as though the previous evaporation period had continued uninterrupted. If the duration of the interstorm period is long enough, this wave may also be consumed, and the next earlier profile would become important.

SWAHP allows for this carryover storage effect. Following a storm, calculations proceed as described earlier until the actual cumulative exfiltration is equal to the previous infiltration depth. When and if this point is reached, the values of Z, $\bar{\Theta}$, Θ_r, F_e, F'_e, and S_e are set equal to the values they had at the end of the previous storm. Then $\bar{\Theta}$, S_e, and F_e are integrated forward in time to account for the elapsed time since they were last used. If the new F_e grows to equal $Z\bar{\Theta}$, then values form the next earlier interstorm period are adopted, and so on.

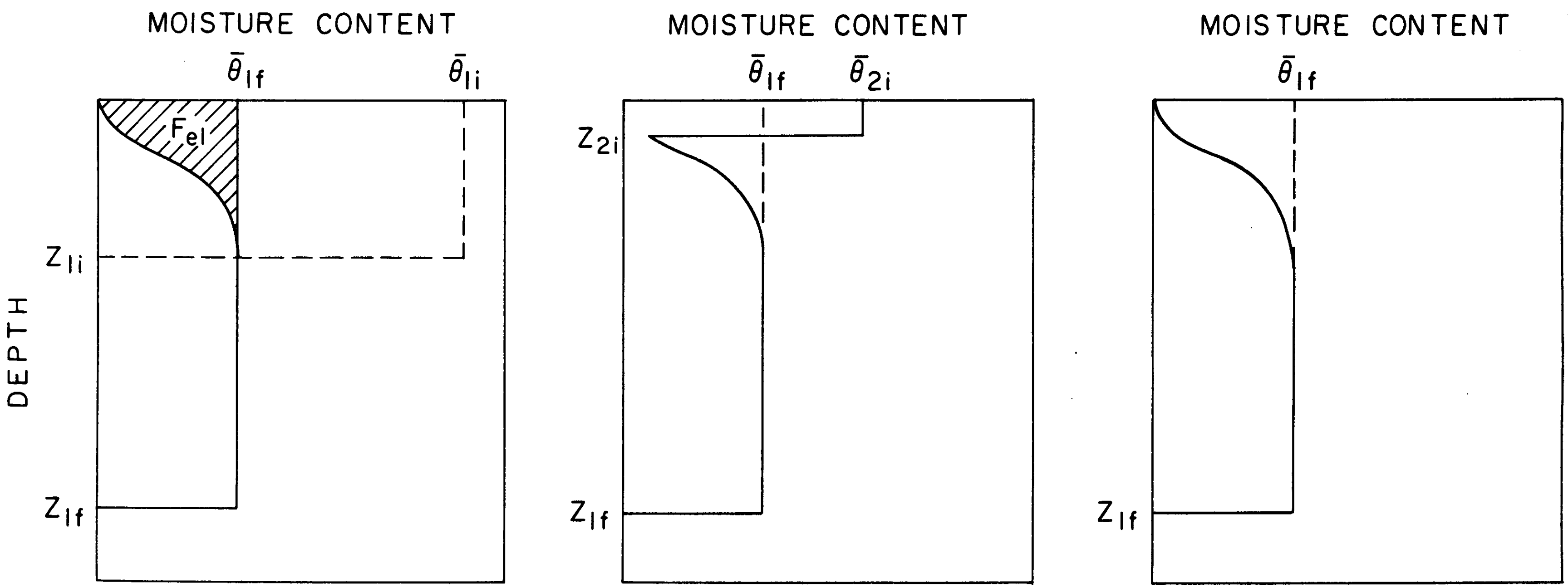

FIGURE 2.3 Illustration of Carryover Storage
See discussion in text.

2.4 The Cases of $E_p > P > 0$ and $E_p < 0 = p$

2.4.1 Potential Evaporation Greater than Precipitation

Equation (2.21) is used when potential evaporation does not exceed precipitation. When this fails to hold at any point during a storm (because P is very low and E_p is positive), then there is effectively no rainfall. If significant infiltration has already occurred during the storm, when the rain was more intense, then it is reasonable to assume that there is sufficient moisture available for potential evaporation from the surface. We shall assume that this earlier rainfall provides the difference between current values of E_p and P. In a slight extension of the TCA, we shall view this as negative infiltration, causing a decrease in the compressed time. Thus, provided F_i is positive,

$$f_i(t) = P(t) - E_p(t) \qquad ; 0 < P < E_p,\ F_i > 0 \tag{2.58}$$

If $P < E_p$ and F_i is zero, either because the storm has just begun or because negative infiltration has equalled positive, then (2.58) should not be used, as it would lead to a negative compressed time and negative cumulative infiltration. In this case, we assume (using the TCA) that whenever F_i goes to zero during a storm, the state of the soil moisture is the same as it was when F_i was last zero during the storm. If E_p then continues to exceed P, the further loss of soil moisture is viewed as exfiltration in a continuation of the previous interstorm evaporation event. Equation (2.40), allowing for nonzero P, becomes

$$f_e(t) = E(t) - P(t) = \min[f_e^*(t),\ E_p(t) - P(t)]; \qquad 0 < P < E_p,\ F_i = 0 \tag{2.59}$$

The exfiltration capacity is still calcualted using (2.46) with the values of F_e', F_e, $\bar{\Theta}$, etc. saved and/or integrated forward in time, as appropriate, from the previous interstorm event.

2.4.2 Dewfall

The other case excluded form the earlier analysis is that of dewfall, or negative evaporation (condensation), between storms. As long as some net exfiltration has occurred prior to the condition $E_p < 0$, we can extend the TCA for exfiltration and consider dewfall as negative exfiltration. Then (2.40) still holds, and simplifies to

$$E(t) = f_e(t) = E_p(t); \; E_p < 0 = P, \; F_e' > 0 \tag{2.60}$$

If F_e' is zero, this approach cannot be applied, since it gives negative compressed time and negative cumulative exfiltartion. In that case, further condensation is added to the mass of soil water associated with the most recent storm event,

$$\bar{\Theta} \frac{dZ}{dt} = E\ (t) = E_p(t) \; ; \; E_p < 0 = P, \; F_e' = 0 \tag{2.61}$$

2.5 The Potential Evaporation Rate and the Surface Temperature

2.5.1 The Potential Evaporation Rate

The potential evaporation rate is the hypothetical rate of evaporation that would occur if free liquid water were made readily available at the soil surface. It can be expressed in terms of atmospheric conditions and surface temperature as

$$E_p \equiv \frac{C_w U_a}{\rho_\ell} \left[\rho_o(T_1) - \rho_{va} \right] \tag{2.62}$$

in which C_w is a stability-dependent bulk transfer coefficient (Appendix B), U_a is the windspeed and ρ_{va} the absolute humidity at screen height Z_a (Appendix B), ρ_ℓ is the density of liquid water, ρ_o is the saturation absolute humidity, and T_1 is the surface temperature.

Since observations of the surface temperature are not normally availabile, its value will be predicted using an energy balance.

2.5.2 The Surface Energy Balance

The net absorption of radiant energy by the soil surface is balanced primarily by losses of latent heat to evaporation, turbulent diffusion of sensible heat to the atmosphere, conduction of heat into the ground (G), and back radiation to the sky. There are also small inputs and outputs of sensible heat associated with rainfall and surface runoff. The energy balance equation can be written

$$(1-A)\ I_s - \varepsilon[\sigma(T_1 + 273)^4 - I_{\ell d})] - \rho_\ell (L + c_\ell T_1)E$$
$$- \rho_a\ c_p\ C_H\ U_a\ (T_1 - T_a) - G + \rho_\ell\ c_\ell\ T_a P$$
$$- \rho_\ell\ c_\ell\ T_1\ R_s = 0 \qquad (2.63)$$

The albedo of the soil is A, ε is the emissivity, I_s is incoming solar radiation, σ is the Stefan-Boltzman constant, $I_{\ell d}$ is incoming (i.e., atmospheric) longwave radiation, L is the latent heat of vaporization of water, c_ℓ is the specific heat of liquid water, ρ_a and c_p are the density and the specific heat at constant pressure of air, C_H is a bulk transfer coefficient for sensible heat (Appendix B), and T_a is the air temperature. Albedo and emissivity may be dependent upon the surface moisture content.

In order to solve (2.63) for T_1, we must develop a second relation between ground heat flux and the surface temperature. One such expression can be derived for an ideal homogeneous medium, giving the surface temperature as a convolution of past heat fluxes. However, this is not convenient to apply in practice. Instead, we resort to the force-restore procedure, described next.

2.5.3 The Force-Restore Equations for Surface Temperature

Deardorff (1978) has shown that the force-restore method is an efficient means of parameterization of the ground surface temperature. Because of its firm physical basis, its accuracy, and its simplicity, we shall employ it in conjunction with the moisture parameterization.

Consider a homogeneous semi-infinite medium subjected to a simple sinusoidal input of heat,

$$G = B \sin(2\pi t/\tau + \pi/4) \tag{2.64}$$

If we impose the condition that

$$T \rightarrow T_2 , \qquad x \rightarrow \infty \tag{2.65}$$

where x is the distance into the medium, then the well-known analytic solution to the heat diffusion equation yields the surface temperature, T_1, after any initial transient has died away,

$$T_1 = T_2 + B\left(\frac{\tau}{2\pi\lambda C}\right)^{1/2} \sin(2\pi t/\tau) \tag{2.66}$$

The essence of the force-restore method is the replacement of the linear partial differential equation of diffusion by an equivalent linear ordinary differential equation of the form

$$\frac{dT_1}{dt} = c_1 G - c_2(T_1 - T_2) \qquad (2.67)$$

This is the simplest dynamic equation for surface temperature that includes both heat input from the atmosphere (G, which forces T_1) and a diffusion-like conduction of heat into the underlying soil (which tends to restore T_1 to the deep temperature, T_2). Furthermore, the constants c_1 and c_2 can be chosen so that (2.67) yields the exact solution, (2.66), to the heat diffusion equation, given the forcing (2.64). This means (2.67) can be tuned to the forcing period τ. The values of c_1 and c_2 are (Deardorff, 1978)

$$c_1 = 2\left(\frac{\pi}{\lambda C \tau}\right)^{\frac{1}{2}} \qquad (2.68)$$

$$c_2 = 2\pi/\tau \qquad (2.69)$$

where λ and C are the thermal conductivity and the volumetric heat capacity of the homogeneous medium.

When G is not sinusoidal, or λ or C are not constant in time and space, (2.67) - (2.69) will no longer yield the exact solution for surface temperature, but they will still yield a good approximation if the forcing is dominated by a particular frequency and if λ and C vary little. This is in fact the case for a soil subjected to natural diurnal forcing by solar radiation and other variables. The period, τ, is one day. We therefore, use (2.67) to predict the surface temperature, T_1, which is required by (2.62) in order to evaluate E_p for use in (2.40) and elsewhere.

In general, T_2 will vary slowly due to the annual cycle of forcing. We model this with a simple linear forcing equation (Deardorff, 1978)

$$\frac{dT_2}{dt} = (\lambda C\ N_d\ \tau)^{-\frac{1}{2}}\ G \qquad (2.70)$$

Deardorff suggests N_d should equal 365, the number of days (recall τ = 1 day) in the annual cycle.

2.5.4 Evaluation of Wetness-Dependent Properties

The products λC used for predicting T_1 and T_2 in (2.67)/(2.68) and (2.70) are effective values over the appropriate depths. For predicting T_2, it is an average over the top ten or twenty meters of soil. This is mostly below the surface zone of rapid fluctuations and large gradients of moisture content. If the water table is deep (> 10 meters), we can evaluate λC using the moisture content at which the hydraulic conductivity is equal to some fraction of the annual average precipitation rate,

$$(\lambda C)_2 = \lambda(\hat{\Theta})C(\hat{\Theta})\ ;\ K(\hat{\Theta}) = f\ \bar{P} \qquad (2.71)$$

The subscript 2 denotes the prediction equation for T_2. $\bar{P}$ is the annual average rainfall depth divided by one year. The coefficient f, which lies between 0 and 1, gives the fraction of rain that can be expected to reach the water table. It can be "guesstimated" for a given location. Large errors in f translate to small errors in (λC), since K is highly sensitive to Θ.

When the water table is high (less than a couple meters deep), $\hat{\Theta}$ may be taken to be equal to the porosity. For intermediate cases, an appropriately weighted average value can be used.

The effective value of λC for use in (2.67)/(2.68) is a weighted average of $(\lambda C)_2$ and the value at the surface,

$$(\lambda C)_1^{\frac{1}{2}} = 0.7(\lambda C)_2^{\frac{1}{2}} + 0.3[\lambda(\Theta_1)\ C(\Theta_1)]^{\frac{1}{2}} \tag{2.72}$$

in which Θ_1 is the estimated value of Θ at the soil surface. This is a simplification of an equation suggested by Deardorff (1978). An admittedly crude estimate takes Θ_1 equal to $\bar{\Theta}$, except when the potential evaporation rate exceeds the exfiltration capacity. The surface is then considered dry,

$$\Theta_1 \begin{cases} 0 & f_e^* \leq E_p \\ \bar{\Theta} & f_e^* > E_p \end{cases} \tag{2.73}$$

This value of Θ_1 is also applicable for the calculation of A and ε when they depend on moisture content.

2.6 Summary of Inputs

Reviewing the contents of this chapter, we see that SWAHP requires specification of the following soil functions:

$K(\Theta)$	hydraulic conductivity
$\Theta_w(\psi)$	main wetting curve
$\Omega(\Theta)$	tortuosity of air phase
$\lambda(\Theta)$	effecitve thermal conductivity
$C(\Theta)$	bulk heat capacity
$A(\Theta)$	surface albedo
$\varepsilon(\Theta)$	surface emissivity

Also needed are the surface roughness and the screen height (Appendix B), and an estimate of $\hat{\Theta}$ or f for (2.71). Given all of the soil parameters necessary to define the functions above, we can then integrate from initial values of T_1, T_2, and $\bar{\Theta}$ to future values at any subsequent time by specifying the atmsopheric forcing during the intervening period. This consists of the following variables:

I_s	incoming solar radiation
$I_{\ell d}$	incoming atmospheric radiation
T_a	air temperature at screen height
ρ_{va}	absolute humidity at screen height
U_a	windspeed at screen height
P	precipitation rate

The computer program used for integration of SWAHP is described in the following chapter.

Chapter 3

PROGRAM DESCRIPTION

3.1 Introduction

In this chapter, we review the structure and functioning of the computer program SWAHP. This is essentially an outline of how the algorithm of Chapter 2 is solved numerically. The emphasis here is on the overall program structure and on some specific features that deserve special note. In addition to this description, there is also fairly extensive documentation of the program steps in the program source itself (Appendix E). For the user who must underestand the program structure and operation in order, say, to make modifications, it is recommended that he read this chapter (after reading Chapter 2) and then refer to the program listing.

SWAHP is written in FORTRAN and was developed on the Honeywell Multics system at M.I.T. In addition to the main program, there are numerous subroutines and function subprograms. These are listed in Table 3.1. Several important variables, referred to in the discussion of the major subroutines (Section 3.3), are tabulated in Table 3.2, along with either descriptions or equivalent symbols. These and most of the other variables are listed in the program itself (Appendix E).

3.2 Overall Structure

The gross structure of SWAHP is illustrated by the flowchart in Figure 3.1. The series of subroutine calls given there corresponds precisely to the brief main program. Subroutine INIT, called only once for a given simulation, performs initial input and output and performs numerous initialization procedures. The three-part sequence of START, PERIOD, and END is

SUBROUTINES		FUNCTIONS	
Name	Called by	Name	Referenced by
INIT	MAIN	ALB	PERIOD
SOILI	INIT	DCADRE*	SORPI
PHII	INIT	DE	DCADRE,SORPI
SORPI	INIT	DI	DCADRE,SORPI
ICSCCU*	SORPI,DSORP	DSORP/DSORPA	PERIOD,REDIST
START	MAIN	EHCAP	INIT,PERIOD
REDIST	START	EMI	PERIOD
ICSEVU*	DSORPA,SORP,DSORP	ETCON	INIT,PERIOD
PERIOD	MAIN	FC	START,PERIOD
FOREST	PERIOD	FINV	START,SORPI,REDIST
END	MAIN	FXBAR	FINV,START
OUTPUT	END	HYCON	DE,DI,FINV,REDIST
		PHI	START
		RHOZ	INIT,PERIOD
		SORP	FXBAR
		TC	PERIOD,END,REDIST
		TORT	DE
		XWET	SOILI,SORPI,FINV,DE,DI

TABLE 3.1

Subprograms used in SWAHP.
DSORPA is an entry point in the function DSORP.
*IMSL subprogram.

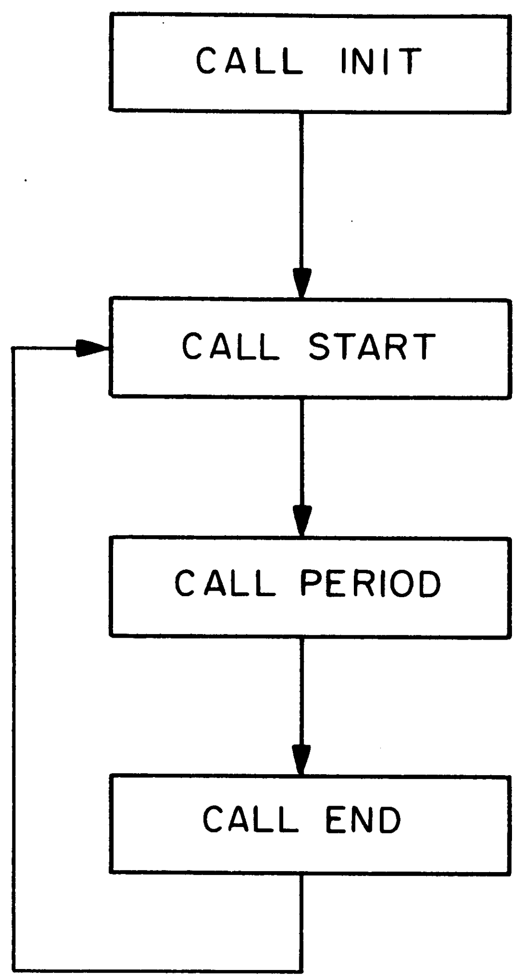

FIGURE 3.1 SWAHP Structure

TABLE 3.2

LIST OF FROTRAN VARIABLES USED IN SECTION 3.3

NAME	EQUIVALENT SYMBOL OR DESCRIPTION
ALBEDO	A
C1	c_1
C3	c_3
CAPEX	min(CAPEX1, CAPEX2)
CAPEX1	Average exfiltration capacity for period
CAPEX2	Equivalent exfiltration capacity due to mass limit
CAPIN	f_i^*
CPNOLD	See text
DEDT	$\left.\frac{dE}{dT_1}\right\|_{T_1} = \text{T1LAST}$
DELT	Duration of period
DELT1	Time until current mass is exhausted
DPART	$0.229\ (1 + \frac{T}{273})^{1.75}\ g\rho_o/R_v(T + 273);\ T = \text{TAVG}$
E	E
EØ	$E\|_{T_1} = \text{T1LAST}$
EMIS	ε
EXFILT	Average rate of exfiltration for the period
FE	F_e
FEP	F'_e
FI	F_i
FILT	Average rate of infiltration for the period
G	G
GORT	$g/R_v(T + 273);\ T = \text{TAVG}$
NPER	Index for simulation period
P	P
Q	Vector containing major fluxes, time in days

TABLE 3.2

(Continued)

RADS	I_s
RADLD	$I_{\ell d}$
RADLU	$\varepsilon\sigma(T_1 + 273)^4$
RADLUØ	$\varepsilon\sigma(T1LAST + 273)^4$
RAIN	Logical variable telling if P>0 this period
RAINB4	Logical variable telling if P>0 last period
RHOVA	ρ_{va}
RLE	$\rho_\ell(L_o + C_p\ T_1)E$
RN	Net radiation
ROOTØ	$[\lambda(0)\ C(0)]^{\frac{1}{2}}$
SE	S_e
SENS	H
UA	U_a
STACK	Array holding old values of exfiltration data
T1LAST	T1 at end of previous period
T2	T_2
TA	T_a
TAVG	Average temperature used to evaluate $D_{\psi v}$
TCE	Compressed time, t_c, for exfiltration
TCI	Compressed time, t_c, for infiltration
TIME	Time elapsed since start of simulation
TRAIN	t_r
TRED	Duration for which redistribution is to be performed
TSTART	Time at start of current rain
XBAR	$\bar{\Theta}$
XR	Θ_r
XZ	Value of F_i for storm that produced the active water mass

executed once for each simulation period. (In our discussion of SWAHP, a simulation period is any finite duration of time during which the atmospheric forcing is specified to be constant. Typically, it may be on the order of one hour in length.) PERIOD performs the actual calculation of evaporation and surface temperature according to the SWAHP algorithm. START and END are mainly devoted to preparation for and follow-up of the subroutine PERIOD. Program execution stops in START when the end of the data is reached. The four major subroutines mentioned here are discussed in more detail in the next section.

3.3 The Major Subroutines

3.3.1 INIT

The main tasks performed by INIT are listed in Figure 3.2. The first statements perform the initial input and output. Next GORT and DPART, two temperature-dependent variables used in the calculation of the vapor conductivity by the function DE, are computed using an input average soil temperature, TAVG.

INIT calls SOILI, which reads the soil parameters and performs calculation of the constants to be used later in the periodic evaluation of various soil functions. The subsequent call to PHII sets up a table of the atmospheric stability ratio versus bulk Richardson number, to be used later for interpolation. Next, INIT calls SORPI, which sets up tables for later interpolation of the desorptivity and the sorptivity.

Final statements in INIT initialize certain constants and variables.

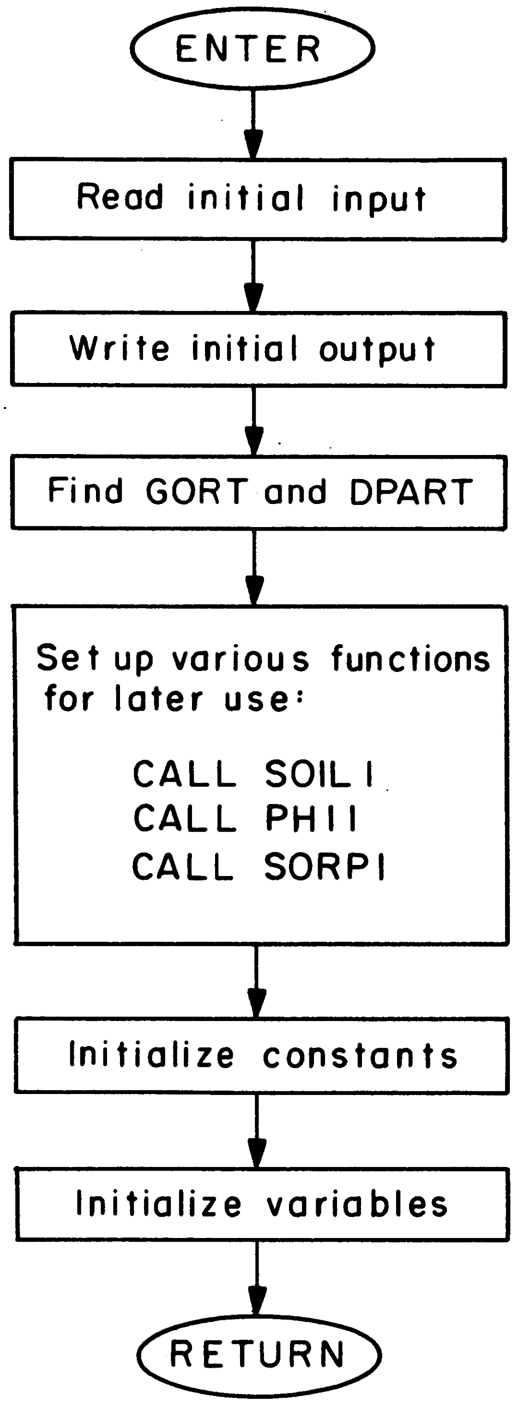

FIGURE 3.2 Flowchart for Subroutine INIT

3.3.2 START

The input for each simulation period - DELT and the forcing variables - is read by START (Figure 3.3). The forcing data consists of P, RHOVA, UA, RADS, RADLD, and TA (Table 3.2). The period duration, DELT, is normally positive. A negative value indicates the end of data and signals the end of calculations. If DELT is positive, NPER and TIME are increased, and forcing-dependent variables are calculated. The logical variable RAIN is set equal to .TRUE. only if P is greater than zero. The value of RAIN from the previous period is saved in RAINB4 ("RAIN-BEFORE"). Subsequent calculations depend upon the values of RAIN and RAINB4, as seen in Figure 3.3.

Let us first consider the case when RAIN is true. There are then two possiblities. The simulation period either is (if not RAINB4) or is not (if RAINB4) the first period of a particular storm. If it is, TSTART is set equal to the elapsed time at the start of rain, and FI (cumulative infiltration) and TCI (compressed infiltration time) are set to zero.

Regardless of the value of RAINB4, START calculates CAPIN and sets TRED when it is raining. CAPIN is the average infiltration capacity (rate) during the current period, given a saturated surface and an initial value of the compressed time. It is given by

$$\text{CAPIN} = \frac{1}{\text{DELT}} \int_{\text{TCI}}^{\text{TCI+DELT}} f_i^*(t_c)\,dt_c$$

$$= \frac{1}{\text{DELT}} \left[F_i(t_c = \text{TCI+DELT}) - F_i(t_c = \text{TCI}) \right]$$

where $f_i^*(t_c)$ and $F_i(t_c)$ correspond to (2.17) and (2.20). After calculating CAPIN, START performs some operations that are executed every period,

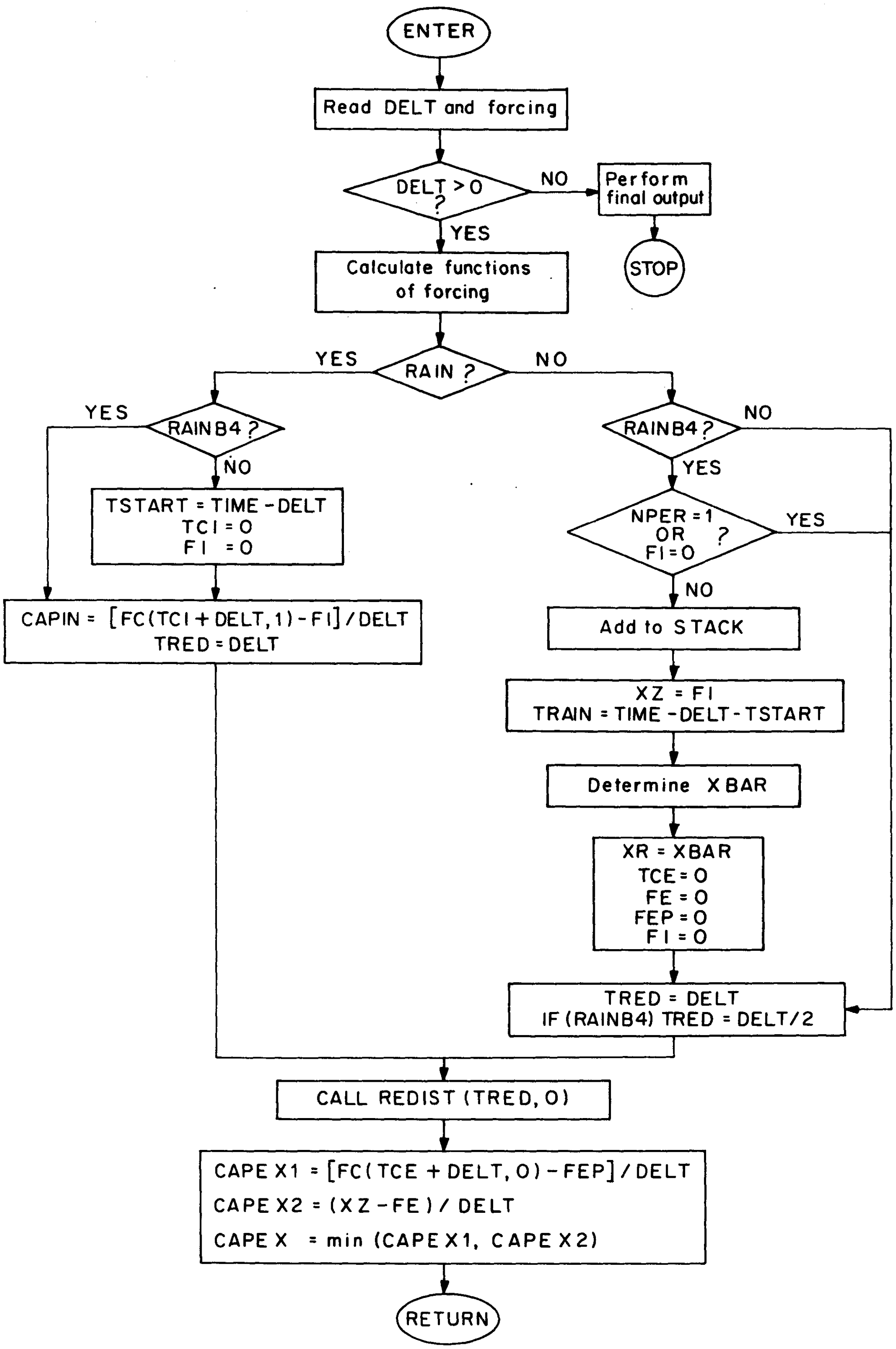

FIGURE 3.3 Flowchart for Subroutine START

whether it rains or not. These are discussed after the following explanation of the interstorm case.

There are also two cases when it is not raining. If it is the start of an interstorm event, RAINB4 will be .TRUE., and several re-initializations must be made. For the moment, assume NPER > 1 and FI > 0. The exfiltration data associated with the previous storm (FE, FEP, XZ, XBAR, XR) are saved in STACK, in case they are needed in the future (Section 2.3.5). The duration of the storm that just ended (TRAIN) is calculated. The mass of water available for evaporation from that storm (XZ) is set equal to FI. XBAR is calculated according to (2.26), and XR is set to this initial value of XBAR. The variables TCE, FE, FEP, and FI are set to zero.

If this is the first simulation period of the entire run, then there was no previous storm and the re-initialization operations are skipped. Instead, XBAR is left with the value of the input initial moisture content.

Occasionally, a "storm" will result in no additions to soil moisture, as evaporation will equal or exceed precipitation (Section 2.4.1). In that case (FI = 0), the old exfiltration data is not replaced.

For all interstorm periods, the value of TRED is set.

For all simulation periods, START also calls REDIST and sets CAPEX1 and CAPEX2. REDIST (Section 3.4.1) performs the integration of (2.30) for a new value of XBAR; numerically, XBAR and SE are updated step-wise at the start of each simulation period. When the new value of SE is found, (2.57) is integrated forward for the instantaneous change in SE, i.e.,

$$\int_{t_k-\varepsilon}^{t_k+\varepsilon} \frac{1}{F'_e} \frac{dF'_e}{dt}\, dt = \int_{t_k-\varepsilon}^{t_k+\varepsilon} \frac{f_e}{F'_e}\, dt + \int_{t_k-\varepsilon}^{t_k+\varepsilon} \frac{2}{S_e} \frac{dS_e}{dt}\, dt$$

in which t_k is the time at the start of the current period, when the jump in SE occurs, and ε is an infinitesimally small time increment. The first term on the RHS is negligible, and the equation yields

$$(\ell n\ F'_e)_{new} = (\ell n\ F'_e)_{old} + \int_{t_k-\varepsilon}^{t_k+\varepsilon} \frac{2\ (\Delta S)_k\ \delta(t-t_k)}{(S_e)_{old} + (\Delta S)_k u\ (t-t_k)}\, dt$$

in which the "new" and "old" subscripts refer to moments before and after time t_k, $(\Delta S)_k$ is given by

$$(\Delta S)_k = (S_e)_{new} - (S_e)_{old}$$

$u(\cdot)$ is the unit step function, and $\delta(\cdot)$ is its derivative, the Dirac delta function.

This integrates to

$$(\ell n\ F'_e)_{new} = (\ell n\ F'_e)_{old} + 2\left[(\ell n\ S_e)_{new} - (\ell n\ S_e)_{old}\right]$$

or

$$(F'_e)_{new} = (F'_e)_{old} \left[(S_e)_{new} / (S_e)_{old}\right]^2$$

FEP is updated (in REDIST) using this formula, and TCE is re-set using (2.45). These corrections need not be made if the current period is the first of an interstorm event, since FEP and TCE are then zero. They are made during storms, because of the chance that exfiltration may occur.

During a given simulation period, there are two limits on the average evaporation rate, CAPEX1 and CAPEX2. The first is analogous to CAPIN, and is given by

$$\text{CAPEX1} = \frac{1}{\text{DELT}} \int_{\text{TCE}}^{\text{TCE+DELT}} f_e^*(t_c)\, dt_c$$

$$= \frac{1}{\text{DELT}} \left[F_e'(t_c = \text{TCE+DELT}) - F_e'(t_c = \text{TCE}) \right]$$

How is $F_e'(t_c)$ determined? We have already indicated that the second term on the RHS of (2.57) is considered stepwise between periods. During a period, this leaves

$$\frac{dF_e'}{dt} = f_e$$

With $f_e = f_e^*$, t corresponds to t_c. The expression for $f_e^*(F_e')$ appears in (2.46), and yields

$$\frac{dF_e'}{dt_c} = S_e^2/2F_e'$$

which is the desired relation.

CAPEX2 is a second limit on evaporation, accounting for the finite amount of water in the soil from the last storm. Its value is the difference between XZ and the current FE, divided by DELT,

$$\text{CAPEX2} = (\text{XZ} - \text{FE})/\text{DELT}$$

3.3.3 PERIOD

PERIOD (Figure 3.4) solves the force-restore equation for ground surface temperature, T_1, subject to the constraint that evaporation not exceed either of the limits imposed by diffusion resistance (CAPEX1) and by mass conservation (CAPEX2). The subroutine is complicated due to the possibility that, if CAPEX2 is exceeded, new exfiltration data may be extracted from STACK, thereby requiring new values of CAPEX1 and CAPEX2. This complication will be discussed below.

The variables CPNOLD and DELT1 are first set to zero. These are explained in the discussion of CAPEX1 and CAPEX2.

On the first pass, the force-restore equation is solved assuming that evaporation occurs at the potential rate. To evaluate C1, EMIS, and ALBEDO, the surface moisture content is then taken to be XBAR. EØ and DEDT are evaluated as the potential values. The force-restore equation is solved and the resultant E is calculated. If the sum of CAPEX, P (if nonzero) and FI/DELT (if nonzero) is sufficient to meet the demand for evaporation, then the value of E is accepted and control returns to the main program. If not, further calculations depend on which limitation on exfiltration is more servere.

If CAPEX1 is smaller than CAPEX2, then the limitation is due to the slow rate of diffusion, and not to an absolute limit on water. The soil surface is considered to be dry for the evaluation of C1, EMIS, and ALBEDO. EØ is set equal to the total amount of available water, and DEDT is set to zero. This will force a set value of E = EØ, independent of surface temperature. FOREST is called again, and control returns to the main program.

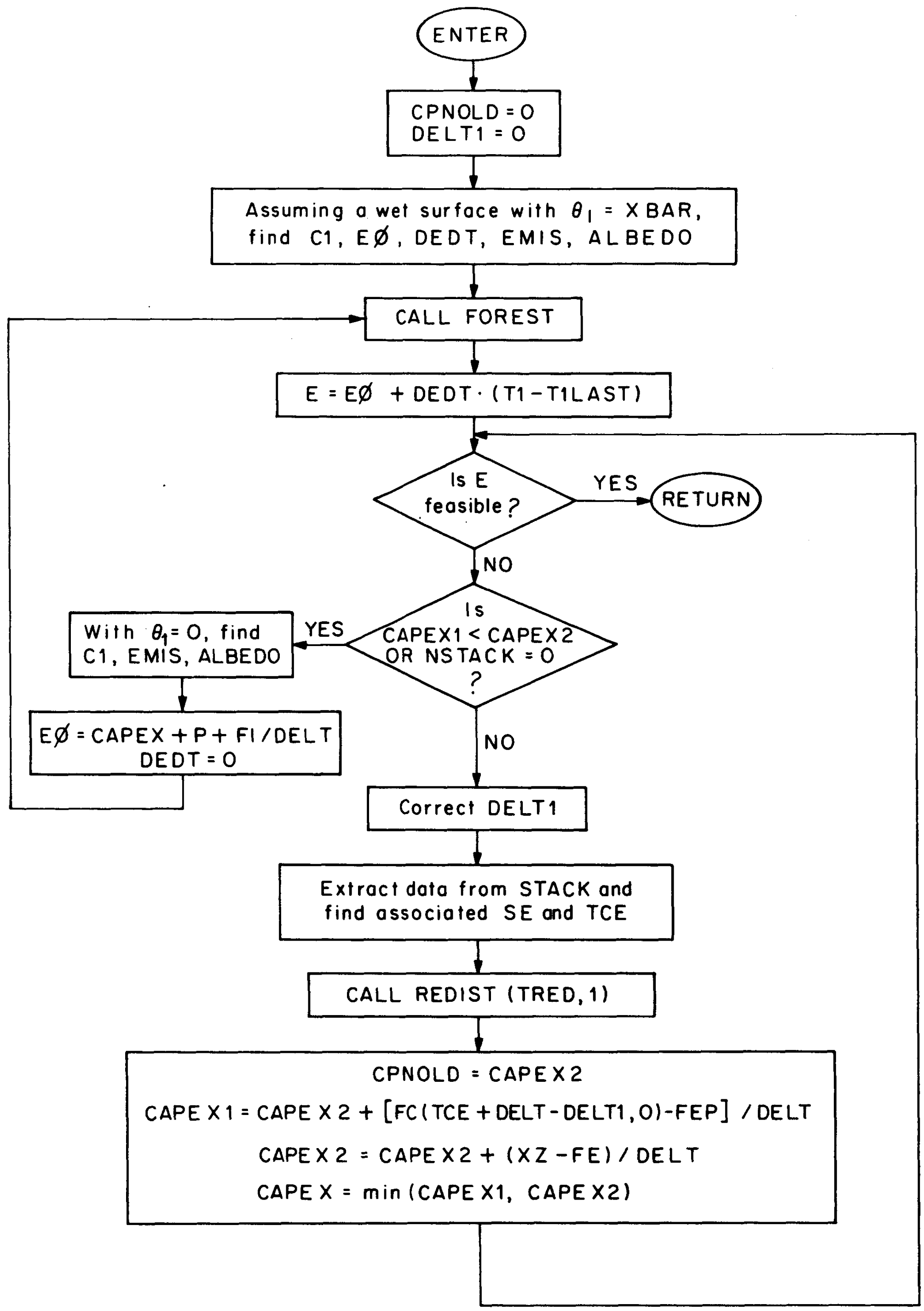

FIGURE 3.4 Flowchart for Subroutine PERIOD

If CAPEX2 is limiting, and there is no further water mass in STACK, then the surface is also considered dry, and the above mentioned procedure is followed.

When CAPEX2 is limiting, but there is still more water (from ealier storms - Section 2.3.5) in STACK then we must account for the availability of this water. We first determine the time, DELT1, that elapses between the start of the period and the moment at which the active mass of water, corresponding to CAPEX2, is depleted. Until this mass is used up, exfiltration occurs at the lesser of CAPEX1 and E_p - P - FI/DELT. Thus,

$$\text{DELT1} = (\text{CAPEX2} \cdot \text{DELT})/\min(\text{CAPEX1},\ E_p - P - \text{FI}/\text{DELT})$$

This equation only applies the first time that data is extracted from STACK during a given period. If further extraction takes place, it must be modified (See below).

Next, the exfiltration associated with the mass of water that was active at the start of the period (CAPEX2) is stored, as a rate, in CPNOLD for future reference. Following this, the new data for exfiltration from an earlier storm is extracted from STACK. The desorptivity and compressed time are calculated. REDIST is called in order to catch up on redistribution that was not performed while the data was in STACK.

CAPEX1 and CAPEX2 were originally calculated in START under the assumption that the mass of soil moisture active then would be active throughout the period. Following extraction of data from STACK, these must be modified.

The total depth of water available, as determined by the diffusion limitations is the sum of the exfiltration depths possible during and after the duration DELT1. By definition of DELT1, the first of these is simply CAPEX2·DELT. The second can be found using the exfiltration equation. The sum of these two depths, divided by DELT, gives the new CAPEX1,

$$\text{CAPEX1} = \frac{1}{\text{DELT}} [\text{CAPEX2}\cdot\text{DELT}$$

$$+ F_e'(t_c = \text{TCI} + \text{DELT} - \text{DELT1}) - F_e'(t_c = \text{TCI})$$

The total depth of water available, as determined by the mass storage limitation, is the sum of the depths during and after the duration DELT1. Then the new CAPEX2 is found as

$$(\text{CAPEX2})_{\text{new}} = \frac{1}{\text{DELT}} [(\text{CAPEX2})_{\text{old}} \cdot \text{DELT} + \text{XZ} - \text{FEP}]$$

Following this, control is transferred back to the point indicated in Figure 3.4. If CAPEX1 and CAPEX2 (together with FI and P) are now sufficient to supply E, control returns to the main program. If not, the check of the relative magnitudes of (the new) CAPEX1 and CAPEX2 is made. If CAPEX1 is now limiting, the dry surface solution is found. If CAPEX2 is again limiting (and there is further data in STACK), then data extraction and updating of CAPEX1 and CAPEX2 must be repeated. Calculation of DELT1 is slightly more complex than already described for the first time through. We still define DELT1 as the time elasped between the start of the period and the exhaustion of the currently active water mass. This is equal to the value of DELT1 already calculated for the first mass, plus the additional time needed to use up the additional water from the second mass.

The amount of that additional water can be seen to be (CAPEX2 - CPNOLD)·DELT, which is the latest increase in CAPEX2, expressed as a depth. Also, CAPEX1 no longer represents the average (over the remainder of the period) infiltration capacity for the current water mass. Instead, it is now given by

$$(FC(TCE + DELT - DELT1, 0) - FE)/(DELT - DELT1)$$

which is equivalent to

$$(CAPEX1 - CPNOLD) \cdot DELT/(DELT - DELT1)$$

We therefore modify the earlier equation for DELT1 to read

$$DELT1 = DELT1 + (CAPEX2 - CPNOLD) \cdot DELT$$
$$/\min[(CAPEX1 - CPNOLD) \cdot DELT/(DELT - DELT1), E - P - FI/DELT]$$

Since CPNOLD and DELT1 are initially set to zero in PERIOD, this equation reduces, on the first time through, to the one presented earlier. It is also valid on subsequent passes, if they occur.

3.3.4 END

END (Figure 3.5) is called after PERIOD, for which it performs follow-up calculations. The changes in FI, FE, FEP, TCI, and TCE are made, fluxes are calculated, and T2 is updated. Output for the period is processed.

The updating to be done depends on the nature of the surface moisture flux. If it rained and evaporation did not exceed precipitation, then the average infiltration rate is given by

$$FILT = \min(P - E, CAPIN)$$

and the resulting surface runoff, if any, is found. On the other hand, if E is greater than P then the exfiltration and/or negative infiltration is

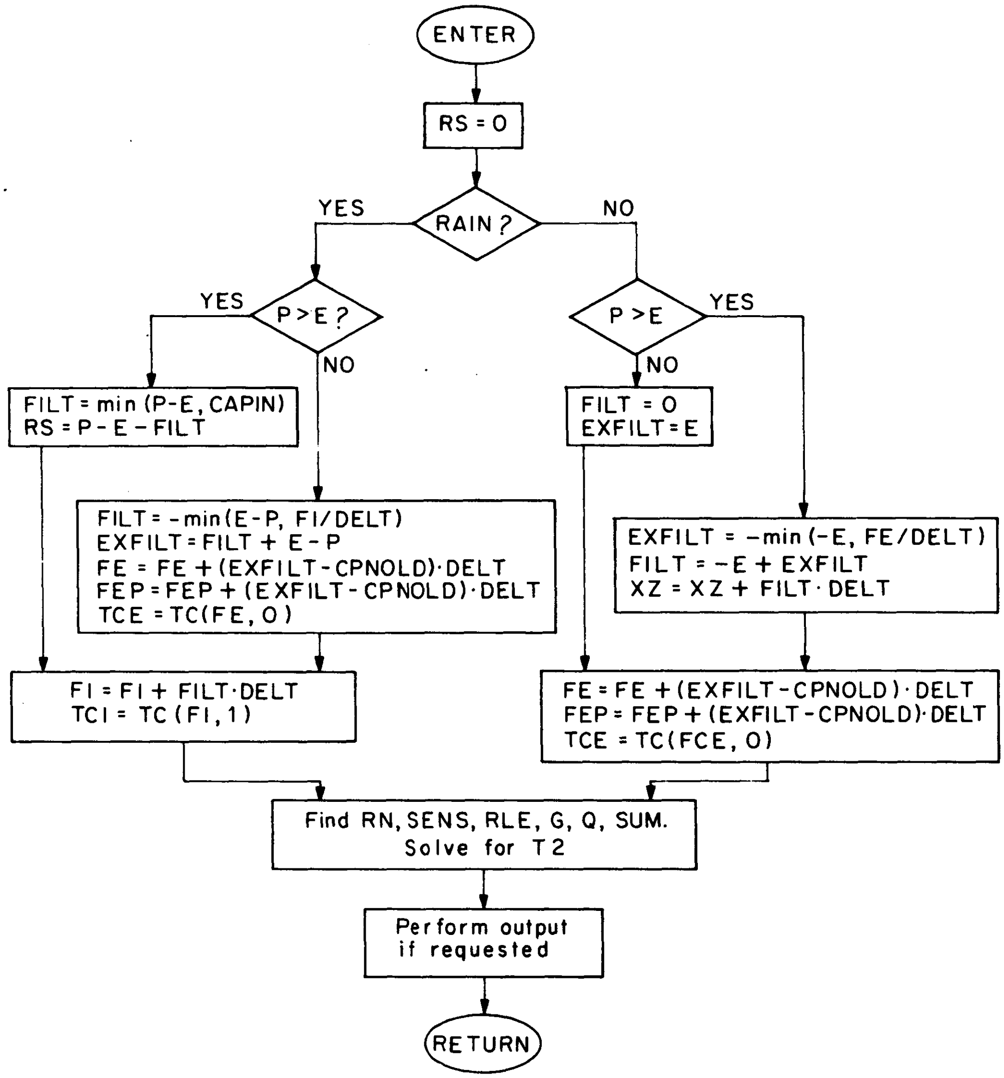

FIGURE 3.5 Flowchart for Subroutine END

found. According to (2.58) and (2.59), negative infiltration occurs first until FI is consumed, and the remainder of the evaporation, if any, comes from exfiltration. The formulae used in this case to find FILT and EXFILT are integrated forms of (2.58) and (2.59), expressed as average rates.

If there is no rain and E is positive, then EXFILT is set equal to E. If E is negative, EXFILT is set equal to the largest portion of E that does not exceed FEP/DELT. This is an averaged form of (2.60) and (2.61). Any remaining portion of E is used to increase XZ.

In all cases, non-zero values of FILT and EXFILT are used to update FI, FE, FEP, TCI, and TCE.

The average rate of net radiation, RN, is calculated as the sum of its three components. The outgoing longwave radiation is given by

$$\text{RADLU}\emptyset + \text{DRADLU}\cdot(\text{T1} - \text{T1LAST})$$

where RADLUØ and DRADLU are the longwave radiation and its temperature derivative calculated by FOREST at the temperature T1LAST. A similar expression is used to find SENS, the rate of sensible diffusion into the air. The latent heat flux, RLE, is calculated as the product of ρ_ℓ, E, and L, the last factor being evaluated at the surface temperature. Heat flux into the ground, G, is given by the balance equation (2.63).

Important heat and water fluxes are stored in the vector Q with the unit of time changed from seconds to days. A cumulative sum of each of these is updated. Equation (2.70) is integrated for a new T_2, and output for the simulation period is performed.

3.4 Other Subroutines

3.4.1 SOILI

At the beginning of a simulation, INIT calls SOILI in orders to read the soil parameters and to initialize several soil constnats. It calculates the constants DAL and DEM for later use by the functions ALB and EMI. It determines the matric potential PRN, separating saturated and unsaturated conditions. This is a constant used by the function XWET.

Several vectors used in the calculation of the relative hydraulic conducitivty in HYCON (Appendix C) are set up in SOILI.

The last section of SOILI performs the initial calculations for the thermal properties. The soil thermal conductivity and soil heat capacity will be calculated in functions ETCON and EHCAP, which use the constants calculated here. These include values of conductivity and heat capacity for the soil when dry, as well as certain partial sums that can be pre-calculated. The thermal conductivity calculations are described in Appendix D.

3.4.2 PHII

This subroutine sets up a table of the stability ratio versus bulk Richardson number, under unstable conditions, for later use by function PHI. The formulation is described in Appendix B.

3.4.3 SORPI

SORPI sets up arrays of $S_i(0, \bar{\Theta})$, for use in (2.26), and of $S_e(0, \bar{\Theta})$ for different values of Θ_r, for use in (2.46) and other equations. Twenty-one (=NX) values of Θ, equally-spaced, approximately from 0 to XU, are defined as XARRAY(I), I = 1, NX. A second vector, YARRAY, is given

NX values approximately from 0 to 1. The range are not exactly (0, XU) or (0, 1) because of some singularities that would cause numerical problems. In this description, we shall ignore the slight discrepancies. They are correctly handled in the program.

If ISOR > 0, desorptivity is calculated for NX values of Θ_r. For each value of Θ_r = XARRAY(I), the desorptivity is found NX times, with the NX $\bar{\Theta}$ values being equally-spaced between 0 and the current value of Θ_r,

$$\bar{\Theta} = \text{YARRAY(J)} \cdot \text{XARRAY(I)}$$

For each pair of Θ_r and $\bar{\Theta}$, the integral in (2.37) is found using the numerical integration algorithm called DCADRE, which is described in Section 3.5.2. The upper limit of integration, $\bar{\psi}$ = P2, is obtained first by solving (2.38) and (2.39) to find $\bar{\Theta}_w(\psi)$, given Θ_r and $\bar{\Theta}$, then by inverting $\bar{\Theta}_w(\psi)$. The relative error in the integral should be less than RERR = 10^{-4}. Given the value of the integral, S_e is determined and stored in SEARAY. When S_e computations are completed, SEARAY(I, J) contains the value of S_e for Θ_r equal to XARRAY(I) and $\bar{\Theta}$ equal to XARRAY(I)·YARRAY(J).

Sorptivity of a dry soil for various values of $\bar{\Theta}$, needed for use in (2.26), is calculated next using a combination of (2.9) and (2.10),

$$S_i(0, \bar{\Theta}) = \left\{2.13\ \bar{\Theta}^{0.33} \int_{-\infty}^{\psi} \left[\Theta_w(\psi)\right]^{0.67} K\left[\Theta_w(\psi)\right] d\psi\right\}^{\frac{1}{2}}$$

in analogy to (2.18). The sorptivity is evaluated for NX values of $\bar{\Theta}$ = XARRAY(I), the result being stored in SIARAY(I). The integral is evaluated using DCADRE, with the upper limit given by $\Theta_w(\bar{\psi}) = \bar{\Theta}$.

The calculated SEARAY and SIARAY are written on file number ISOR.

If ISOR is less than zero, the arrays are not calculated, but instead are read from file number -ISOR. Usually, the first run will be made with ISOR positive, and subsequent runs with it negative, since these calculations are by far the costliest associated with SWAHP.

When the arrays have been either calculated or read, SORPI calls the IMSL subroutine ICSCCU (Section 3.4.7) in order to find the cubic spline coefficients for later interpolation of SIARAY.

The infiltration parameters SØ and AØ are found at the end of SORPI.

3.4.4 REDIST

REDIST has already been mentioned in connection with START, which calls it once during each interstorm simulation period. In REDIST, the $K - \Theta$ relation is fitted to a log-log relation in the neighborhood of $\bar{\Theta}$, i.e.,

$$K = k\Theta^c$$

This facilitates integration of (2.30), which becomes

$$\frac{d\bar{\Theta}}{dt} = - \frac{kR\bar{\Theta}^{c+1}}{F_i}$$

Integration yields

$$(\bar{\Theta})^{-c}_{new} - (\bar{\Theta})^{-c}_{old} = \frac{ckR \cdot TRED}{F_i}$$

where TRED is the duration of the integration. Using the approximation for K, we can write this as

$$(K)_{new} = \left[\frac{1}{(K)_{old}} + \frac{cR \cdot TRED}{F_i}\right]^{-1}$$

Given $(K)_{new}$, the hydraulic conductivity function can be inverted to find $(\bar{\Theta})_{new}$. SWAHP actually uses an estimate of $\bar{\Theta}$ at the middle of the current period. Thus, TRED equals DELT/2 when an interstorm event begins, and DELT thereafter. When data is extracted from STACK, TRED is the amount of time it was in storage.

When $\bar{\Theta}$ has been updated, the new S_e, F'_e and compressed time are found as described in Section 3.3.2.

3.4.5 FOREST

FOREST is called by PERIOD in order to perform a one-step integration of the force-restore equation for T_1. Using (2.59), we can put (2.63) in the form

$$\frac{dT_1}{dt} = g(T_1, T_2)$$

To obtain a solution for T_1 at the end of a period, we employ an implicit finite difference in T_1, and evaluate the slowly-varying T_2 explicitly. Thus

$$T_1^k = T_1^{k-1} + \Delta t \cdot g(T_1^k, T_2^{k-1})$$

or

$$f(T_1^k) = 0$$

where

$$f = -g + \frac{T_1^k - T_1^{k-1}}{\Delta t}$$

and where T_1^k is the only unknown.

In order to solve this, we expand f around the latest value of T_1,

$$f(T_1^k) \simeq f(T_1^{k-1}) + f'(T_1^{k-1}) \cdot (T_1^k - T_1^{k-1})$$

where f' is the derivative of f with respect to T_1. Setting this expression equal to zero, we find

$$T_1^k = T_1^{k-1} - f(T_1^{k-1})/f'(T_1^{k-1})$$

3.4.6 OUTPUT

The OUTPUT call is the last action of the last major subroutine, END. It performs the desired output operations, according to the values of IFOR and IUNF (Chapter 4). Formatted output (i.e., "printout") goes to file number |IFOR|, while unformatted (storage-efficient, machine-readable) output goes to file number IUNF. The user may supplement or otherwise modify the write statements according to his needs.

3.4.7 ICSCCU

ICSCCU is a component of the International Mathematical Subroutine Library (IMSL, 1980) used to find the coefficients for cubic interpolation. It is invoked by

CALL ICSCCU (X, Y, NX, C, IC, IER)

where X contains NX ordered values of the independent variable, Y contains the corresponding values of the function, C returns the spline coefficients, IC is the row dimension of C in the calling program, and IER is an error code. Further information is provided in the IMSL documentation. We employ the spline coefficients only in the call to ICSEVU.

3.4.8 ICSEVU

ICSEVU is an IMSL (1980) subroutine used to perform interpolation using cubic splines. Its usage is

CALL ICSEVU(X, Y, NX, C, IC, U, S, M, IER)

where all of the arguments except U, S, and M are defined in Section 3.4.7. In contrast to ICSCCU, ICSEVU takes the coefficients C as input. In general, U is a vector of length M (may be a scalar if M = 1), that contains the value(s) of the independent variable for which the function values, returned in the vector S, are to be found. Further information is provided by IMSL documentation.

3.5 Function Subprograms

3.5.1 ALB

ALB calculates the soil surface albedo according to

$$ALB = \max(ALDRY + DAL \cdot X, ALWET)$$

where X is the moisture content. Constants are read or calculated in SOILI. This yields a linear decrease of albedo with moisture content increasing from dryness to half-saturation, then a constant value of albedo. This shape is based on the experimental findings of Idso et al. (1975) for Avondale loam.

3.5.2 DCADRE

DCADRE is an IMSL (1980) function that integrates a function using cautions adaptive Romberg extrapolation. The reference

DCADRE(F, A, B, AERR, RERR, ERROR, IER)

returns the integral from A to B of external function F such that

$$\left| \int_A^B F(x)dx \right| \leq \max\left[AERR,\ RERR \left| \int_A^B F(x)dx \right| \right]$$

ERROR is an estimated bound on the absolute error of the value returned. IER is the error code.

3.5.3 DE

DE calculates the integrand in (2.37), given the matric potential, PP (passed as an argument), and the value of XB (passed through the common block SORCOM). DE is called by CDADRE when performing the desorptivity integrations for SORPI.

3.5.4 DI

DI calculates the integrand is Section 3.4.3, given the matric potential, PP. It is called by DCADRE when performing the sorptivity integrations for SORPI.

3.5.5 DSORP

Given the new value of XR (Θ_r), DSORP sets up a vector of desorptivity, SE1, corresponding to that value of Θ_r and to each of the YARRAY values. (See Section 3.4.3.) To obtain each element of SE1, ICSCCU, and ICSEVU are called to interpolate in the Θ_r (or XARRAY) direction.

When the SE1 vector has been filled, it can be used to interpolate for the new desorptivity each time $\bar{\Theta}$ decreases, as long as Θ_r remains the same, i.e., until a new storm comes or until old data is extracted from STACK. Spline coefficients for the interpolation along SE1 are found by ICSCCU.

Given XR, interpolation in SE1 is with respect to the vector YARRAY, with the appropriate value of the independent variable given by $\bar{\Theta}/\Theta_r$, according to the scaling established in Section 3.4.3. The interpolation is perfomed by ICSEVU. Once SE1 has been set up, subsequent entry to this function occurs at DSORPA, skipping the initial calculations, which would be redundant, so long as XR does not change.

3.5.6 EHCAP

EHCAP finds the effective, or bulk, volumetric heat capacity of the soil as a function of moisture content, X, according to

$$EHCAP = HCD + HCAP(1) \cdot X$$

3.5.7 EMI

EMI returns the soil surface emissivity as a function of moisture content, X. A linear dependence is assumed,

$$EMI = EMDRY + DEM \cdot X$$

3.5.8 ETCON

ETCON calculates the effective thermal conductivity of the soil as a function of the moisture content, using the method of deVries (Appendix D).

3.5.9 FC

FC finds the cumulative flux that corresponds to the specified compressed time, TC. FC and TC refer to exfiltration if KOD = 0, to infiltration if KOD = 1. Equations (2.20) for infiltration and (2.42) for exfiltration apply.

3.5.10 FINV

FINV finds a value of X for which

$$\left|\frac{F(X)-Y}{Y}\right| \leq YERR$$

where Y and YERR are input agruments. If METHOD = 1, an original interval (XL, XH), within which the value of X is known to lie, is successively bisected until the convergence condition above is met. The maximum allowable number of iterations is KMAX.

If METHOD = 2, a variation on Newton-Raphson iteration is used. It is assumed that $\ell n(F)$ is more nearly linear than F itself; this allows for rapid convergence with functions F whose range covers several orders of magnitude. (It also restricts usage to functions that stay positive.) Then we can write.

$$\begin{aligned} \ell n\ Y &\cong \ell nF(X_o) + \frac{d}{dX}\left[\ell nF(X_o)\right](X-X_o) \\ &= \ell nF(X_o) + \frac{1}{F(X_o)}\frac{dF(X_o)}{dX}(X-X_o) \end{aligned}$$

where X_o is an initial estimate of X. Solution for X yields

$$X = X_o + F(X_o) \cdot \ln\left[Y/F(X_o)\right] / \frac{dF(X_o)}{dX}$$

This equation is used for iteration until (1) the desired convergence is obtained, (2) the allowable number of iterations is exceeded, or (3) $F(X_o)$ takes a negative value. If either of the latter two occur, FINV switches to METHOD = 1 and starts over.

3.5.11 FXBAR

FXBAR calculates the RHS of (2.26), as a function of moisture content X, for FINV when the latter is called by START to determine the moisture content immediately after a storm.

3.5.12 HYCON

HYCON finds the hydraulic conductivity as a function of moisture content, X. Procedure is described in Appendix C.

3.5.13 PHI

PHI determines the stability factor, STAB, for heat and vapor diffusion, and its derivative with respect to the bulk Richardson number (RI), DSTAB. RI is input. For unstable conditions, linear interpolation for STAB is performed using the arrays RIB and CHRAT that were set up by PHII. DSTAB is calculated directly as the constant slope within the appropriate interval. STAB and DSTAB are placed in the real and imaginary parts of the complex variable PHI. Use of the complex variable allows return of two quantities from a function subprogram.

If conditions are stable, STAB and DSTAB are calculated directly (Appendix B).

3.5.14 RHOZ

RHOZ calculates the saturation value of vapor density, or absolute humidity, in g/cm^3 by interpolation of the array RZ, given the temperature in °C. This value and its temperature derivative are returned via the complex variable RHOZ.

3.5.15 SORP

SORP calls ICSEVU, which performs interpolation for sorptivity using the arrays set up by SORPI.

3.5.16 TC

TC finds the compressed time that corresponds to the specified cumulative flux, FC. FC and TC refer to exfiltration if KOD = 0, to infiltration if KOD = 1. Equations (2.24) for infiltration and (2.45) for exfiltration apply.

3.5.17 TORT

TORT calculates the tortuosity of the air phase, for use in DE, as a function of volumetric air content, Θ_a. We use (Lai et al., 1976)

$$\mathrm{TORT} = \Theta_a^{2/3}$$

3.5.18 XWET

XWET returns the value of the main wetting function, given the argument POT, matric potential. The empirical expression used can be fit well for most soils. It is

$$XWET = AA \cdot (POT/BB)^{CC} + DD \cdot (7 - PF) + EE$$

where PF is $\log_{10}$(-POT) and AA through EE are parameters. If POT is greater than or equal to PRN (Section 3.4.1), then XWET = XU.

3.6 BLOCK DATA

BLOCK DATA is used to initialize several variables that belong to COMMON blocks. These are the arrays RZ, TCON, SF, and HCAP, as well as the physical constants CL, CP, DF, ELØ, RHOCP, RHOL, RLCL, and SIG.

Chapter 4

HOW TO USE SWAHP

4.1 Introduction

This chapter describes how to use the computer program called SWAHP. Section 4.2 summarizes the data requirements of the program. Special attention is given in Section 4.3 to the problem of computing and saving the desorptivity and sorptivity arrays for later use. Section 4.4 describes the current options for output from the program. The proper format of the input file is presented in Section 4.5, and the procedure for using the program on Multics at M.I.T. is described in Section 4.6. The chapter concludes with a sample problem in Section 4.7.

4.2 Data Requirements

4.2.1 Note on Units

All variables should be input in terms of centimeters, grams, and seconds. Energy storage and fluxes are defined in terms of calories and langleys. Use of any other system of units, even if self-consistent, result in errors since some of the program constants are assigned values in these units.

All computations and output are also performed in this system, with the exception of the arrays Q and SUM, whose components are fluxes in terms of days instead of seconds. SUM is printed at the end of a simulation.

4.2.2 System Parameters

Several parameters of the soil system are read by SWAHP as input. The Fortran names of these are R, the redistribution parameter in (2.29); TAVG, a space-time average soil temperature used to evaluate GORT and DPART in (A.6); ENDAYS, N_d in (2.70); ZØ, the surface roughness Z_o in (B.3); ZA, the screen height in (B.3) at which meteorological forcing is measured; and XHAT, the $\hat{\Theta}$ of (2.71). Those describing moisture retention are POR, the porosity n; XU, the re-wetted moisture content Θ_u; AA, BB, CC, DD, and EE, the parameters that define the main wetting function $\Theta_w(\psi)$. CKSAT is the hydraulic conductivity at $\Theta = \Theta_u$ (Appendix C). Parameters used for calculation of λ and C are XK, Θ_k in Appendix D, the limit of liquid continuity; VPER(3), VPER(4), and VPER(5), the volumetric fractions of soil constituents Θ_i, i = 3, 4, 5, used in Appendix D. Parameters relating to reflection and emission of radiation are ALDRY and ALWET, the extreme albedo values in Section 3.5.1; EMDRY and EMWET, the extreme emissivity values.

Each of these 21 parameters is discussed below. We deal with them in the order outlined above.

The redistribution parameter, R, determines the rate of soil moisture redistribution. Milly and Eagleson (1982) suggest a value of R equal to 2. (See Section 2.3.1 of this report.) They have noted, however, that a larger value would be more appropriate for a fine-grained soil that is usually very dry at depth, since this would hasten redistribution. Although the question appears to deserve further research, the value of 2 seems reasonable for general application.

TAVG is the temperature (degrees Celsius) used in the evaluation of $D_{\psi v}$ to find the desorptivity. Computed results are not overly sensitive to this parameter. Milly and Eagleson (1982, p. 118) simply use an initial soil temperature for the analogous problem in finite element simulations and find that it introduces insignificant errors. We recommend setting TAVG = TINIT (Section 4.2.2)

ENDAYS should normally be set equal to 365 (Section 2.5.3). Milly and Eagleson (1982, p. 156) consider an artificial situation where the soil is initially isothermal throughout. They show that a better value for ENDAYS in that situation is approximately the number of days in the simulation, if much less than 365.

ZØ is the soil surface roughness (centimeters) determined, e.g., to fit the surface wind profile,

$$U = \frac{U_*}{k} \ell n(Z/Z_o)$$

where U_* is the friction velocity and k = 0.4 is von Karman's constant.

ZA is the height (centimeters) of observation of UA, RHOVA, and TA (Section 4.2.3), typically 100 to 200 cm.

XHAT has a value determined by (2.71) with a reasonable estimate of f in that equation.

POR is the porosity, the proporation of the soil occupied by "void" space, i.e., air and water.

XU is the moisture content when the dry soil is wetted freely. It is equal to POR minus the proportion of volume occupied by entrapped air. A value of 0.9•POR (Mualem, 1974) can be used in the absence of specific measurements.

The parameters AA, BB, CC, DD, and EE can be found by fitting soil sorption data to the main wetting function given in (3.5.18). Desorption data can be converted to equivalent sorption data using (2.39) with Θ_r equal to Θ_u. BB is in centimeters.

CKSAT (centimeters/second) can be measured by standard methods or estimated by empirical procedures.

The value of XK is not critical. It can be set using (Milly and Eagleson, 1982, p. 60).

$$K(\Theta_k) = 0.1 \cdot D_{\psi v}(\Theta_k)$$

or (Rose, 1963) it can be set equal to the moisture content in equilibrium with a relative humidity of 0.6.

VPER(3), VPER(4), and VPER(5) are the volumetric proportions of quartz, other minerals, and organic matter. (Their sum equals one minus the porosity.)

The radiation parameters ALDRY, ALWET, EMDRY, and EMWET can either be measured or chosen from values given by Sellers (1965) and Eagleson (1970).

4.2.3 Initial Conditions

In its current form, SWAHP accepts no input for the initial condition on moisture. The variables XZ, FI, FE, and FEP one all set to zero initially, implicitly invoking an initial condition of total dryness. (The facts that XBAR is not set to zero and that XZ is slightly non-zero result from computational considerations and are not inconsistent with this initial conditions.) Certainly, before the first simulated storm, predicted evaporation (which can then result only from dewfall) may be incorrect, and the error may persist beyond the first storm if the STACK is emptied. SWAHP was developed without consideration of non-stationary (i.e., seasonal) input sequences, and the dry initial condition was adequate in those cases. In situations where long-term carryover storage of water from before the start of the simulation is of interest, the initial condition should be modified. For example, SWAHP can be modified slightly to read, as input, the initial XBAR and XZ. Their values would be the average moisture content and total water depth in the soil at the start of the simulation, e.g., the start of the "dry season." Alternatively, the preceeding "wet season" could be appended to the start of the simulation.

Initial values of the temperatures T1 and T2 must be specified. Because of the greater thermal inertia associated with it, T2 is the more critical value. The input value of T2 should be the measures soil temperature at a depth where the diurnal temperature wave becomes insignificant. This is normally less than one meter in depth. The initial value of the surface temperature, T1, can be set equal to T2; it will reach dynamic equilibrium with T2 and the atmospheric forcing quickly.

4.2.4 Atmospheric Forcing

The values of the meteorologic forcing variables must be specified for the duration of the simulation. The time scale of their definition should be fine enough to include the diurnal variations for which the force-restore model is intended. A duration of one-half to three hours is suggested for forcing input. Each simulation period, the variables read as input are P, the precipitation rate (centimeters/second); RHOVA, the absolute humidity of the air (gram/cm^3) at screen height; UA, the windspeed (cm/sec) at screen height; RADS, the incoming solar radiation flux (langley/sec); RADLD, the incoming atmospheric radiation flux (langly/sec); TA, the air temperature (°C) at screen height; and DELT, the duration for which these values apply. Standard parameterizations are available for estimation of the radiation terms in the absence of direct measurements (e.g., Eagleson, 1970).

4.3 Evaluation of Sorptivities and Desorptivities

The previous section dealt with most of the input variables, those having physical significance for the system being modeled. A few other variables are input to provide further information about the simulation. One of these is ISOR, which specifies whether the arrays SEARAY and SIARAY are to be calculated and written on a file at the start of a simulation (ISOR > 0) or they are to be read from a file (ISOR < 0). In either case, the absolute value of ISOR is the appropriate file number. For the first run with a particular soil and TAVG, it is necessary to use ISOR > 0. In subsequent runs, it is desirable to use ISOR < 0 in order not to re-calculate the arrays. The integrations for the elements of these arrays consume roughly as much time as twelve months of simulation with one-hour periods (on the order of 100 seconds CPU time at MIT on the Honeywell 68/DPS, Multics system).

4.4 Output Options

Four other input variables control the processing of output from SWAHP. These are IFOR, IUNF, IRAIN, and ISTK.

Output from SWAHP is sequential, and can be either formatted or unformatted. Formatted output can be read directly by the program user, while unformatted output can only be read by the computer as input for other programs. Because of its relative efficiency, the latter is the form that should be used if the user wishes to perform any computer-based post-processing or analysis of the SWAHP output (e.g., plotting).

Formatted output is controlled by IFOR, IRAIN, and ISTK. If IFOR = 0, there is no formatted output. If IFOR is non-zero, output is written on file number |IFOR| every simulation period, following a little initial output to the same file. For IFOR > 0, the variables printed every period are TIME, P, E, XBAR, and T1. For IFOR < 0, they are NPER, TIME, P, EP, E, DFIL, RS, RN, G, SENS, RLE, XBAR, XR, XZ, FEP, FE, FI, TCE, TCI, T1, and T2. These options may be changed by the user, if he so desires, in the subroutine OUTPUT.

If IRAIN = 1 and IFOR ≠ 0, then SWAHP also prints a message whenever rain starts or stops. The time is noted and, if rain is stopping, the depth of infiltration, XZ, and the new XBAR are written.

If ISTK = 1 and IFOR ≠ Ø, a message is printed whenever data is stored in or extracted from STACK. The time and the number of events stored in STACK are printed. When storage occurs, the stored values of XZ (amount of infiltration form the relevant storm) and FE (amount of XZ that has evaporated) are also printed.

Unformatted output is written on file number IUNF if IUNF is positive. Initially, subroutine INIT writes a list of the input variables on IUNF. Each period, subroutine OUTPUT will then also write the list of variables given above for IFOR < 0. At the end of the simulation, START writes an integer -1 on IUNF. Programs reading from IUNF for post-processing should have READ statements corresponding exactly to the WRITE statements in INIT and OUTPUT (Appendix E). Each time a new NPER is read, its value should be checked to see if it is positive. If it is, the rest of the variables can be read. If it is not, this signals the end of the data stored in the file.

4.5 Format of the Input File

The format of the input file (file number 15) is given in Table 4.1. All of the variables, except TITLE (which may be any 80-character string), are discussed in the preceeding sections.

4.6 Running SWAHP on Multics at MIT

To run SWAHP on Multics at MIT, several commands must be given. These are listed below:

1. Add the IMSL library to your search rules, using the following Multics command:

   ```
   add_search_rules >libraries>imsl -after >am
   ```

   This needs to be given only once per process. It can be included in the start_up.ec.

2. Obtain a copy of the SWAHP program source, "swahp.fortran".

3. Compile SWAHP using the "card" and "optimize" options:

   ```
   ft  swahp  -card  -ot
   ```

   This need not be repeated ever unless the object code, a segment named "swahp", is destroyed or damaged.

TABLE 4.1

Card No.	READ Format	Variable(s)
1	A80	TITLE
2	5I4	IFOR, IUNF, IRAIN, ISTK, ISOR
3	8F10.0	R, TAVG, ENDAYS, ZØ, ZA, XHAT, T1, T2
4	8F10.0	POR, XU, CKSAT, AA, BB, CC, DD, EE
5	8F10.0	XK, VPER(3), VPER(4), VPER(5), ALDRY, ALWET, EMDRY, EMWET
6, 7, 8...	7F10.0	DELT, P, RHOVA, UA, RADS, RADLD, TA

Format of the input file. The last card should have the format of 6, 7, 8..., and specify -1. for DELT and 0. for the rest of the variables.

4. Create the input file according to the prescribed format. Let us assume the input file is called "input.data" .

5. Attach the input file:

io attach file15 vfile_ input.data

6. Attach the output files, where appropriate. IF IFOR ≠ 0, then type

io attach filexx vfile_ for.data

where xx is the two-digit numerical value of |IFOR| specified in the input file, and for.data is the (arbitrary) name of the formatted output file. If direct terminal printout is desired, specify IFOR = 6 in the input file and do not execute the command above. By default, the output will then be directed to the terminal. If xx is not 06 and the command above is not given, then a file called filexx will be created and used for output.

If UNF ≠ 0, issue the command

io attach fileyy vfile_ unf.data

where yy is the two-digit numerical value of IUNF being input, and unf.data is the (arbitrary) name of the unformatted output file. If this command is not given, any unformatted output will go to a file called fileyy.

7. Run the program by typing

swahp

If for any reason it is necessary to hit BREAK and to cancel program execution, be sure to follow the BREAK by

rl -a

and

cf -a

before trying to start the program again. The latter command ensures that reading and writing will start back at the beginning of the appropriate file.

8. Any files that were attached should generally be detached after execution, unless the user understands the implications of leaving them attached. To detach file15, type

io detach file15

To detach filexx, type

io detach filexx,

and so on.

9. At this point, formatted output, if any, will either have been printed at the terminal or will be available in for.data or in whatever segment was attached as filexx, or in filexx if no attach command was given. Similarly, unformatted output will be in unf.data or in whatever segment was attached to fileyy, or in fileyy if no attach command was given.

The formatted file may be printed or viewed through an editor. The unformatted file can be used as input for any post-processers.

10. If the program is to be re-run, steps 1, 2, and 3 are skipped. If the new input file has the same soil type and the same TAVG, be sure to change ISOR to a negative value for subsequent runs, in order to avoid re-calculation of sorptivities and desorptivities. Continue as before from step 4.

4.7 Using SWAHP on Other Computer Systems

We assume that users of SWAHP on other systems have working knowledge of the system, including Fortran program compilation and execution, and management of input and output files. In this section we mention two particular problems that may arise on other systems. The first is due to minor differences in the Fortran language from system to system. The second is due to the inability to access the IMSL library programs.

Although the Fortran language is fairly standard, small differences often make a program developed at one site cause errors at another. The problem, if one arises, will ususally be apparent from the error message, and the appropriate modifications can be made. If the error is more obscure, it is best to consult a programmer who is familiar with the intricacies of Fortran on your system.

SWAHP was developed using three IMSL subprograms to perform some of the calculations, but IMSL is by no means universally available. If you do not have access to it, other equivalent software can be used. This may be available in another software library or it may be written specially for this application.

DCADRE, the IMSL numerical integration routine, is documented in Section 3.5.2. Any other integration routine that returns the value of the integral of a function F between limits A and B, subject to similar error criteria, can be used instead of DCADRE in subroutine SORPI.

The other IMSL subroutines used are ICSCCU and ICSEVU, documented in Sections 3.4.7 and 3.4.8. These may be replaced by equivalent cubic spline subprograms, or the user may wish to use another method of interpolation (e.g., simply linear) of SIARAY and SEARAY to find sorptivity and desorptivity.

4.8 Sample Problem

A sample problem is presented here as a test for the user to see if the program is functioning properly and as an example of some of the output.

The input file for the sample problem appears in Appendix F. The soil parameters are reasonable ones for a silt loam. Hourly forcing is specified for four and one half days, beginning at midnight. There is rain during the first 15 hours, and later for two hours just before the end of the third day. (Note: ISOR = -82 since sorptivities and desorptivities had already been evaluated and stored in file 82.)

The output obtained for the sample problem appears in Appendix G. The first rain ends shortly after noon, after which evaporation (at the potential rate) and T1 decrease. From midnight until about dawn, there is dewfall and T1 stabilizes around 10°C. Dewfall occurs again at the start of the third day. Meanwhile, XBAR decreases from 0.31 following the storm to 0.25 by the time the second storm begins.

Whenever rain ends and the exfiltration variables are re-initialized, their values are first saved in STACK. This happens at 0.54 days and at 3.0 days in the example.

The second storm causes only 0.03 cm cumulative infiltration. This is soon consumed by evaporation in the morning of the fourth day (at 3.42 days), at which point the older exfiltration data is extracted from STACK. Soon after that, the water from the earlier storm is also consumed, and the next older data is extracted. (Note that the value of XBAR after extraction corresponds to the value before storage of the same data in the past.) Because the initial value of XZ was specified as 0, there is practically no more evaporation after this point. The exception occurs in the early morning (4.29, 4.33

days) when E_p becomes positive following dewfall. The evaporation then is entirely supplied by the preceeding dewfall.

Summary data on fluxes appears at the end of the output. Average net radiation of 188.0 langleys/day is partitioned among heat flux into the ground (39.4) and into the atmosphere (26.1), and evaporation (122.5).

Average evaporation equals average precipitation. Average potential evaporation is high enough to consume all precipitation, even though evaporation is sometimes limited by diffusion in the soil, as evidenced by the steadily and gradually declining evaporation rate from 2.38 days to 2.79 days. Average E can be no greater than average P, because the initial value of XZ is given as zero.

The potential evaporation exceeds the actual by a factor of about 2.5, and surface runoff is zero.

REFERENCES

Anderson, E. A., A point energy and mass balance model of a snow cover, NOAA T. R. NWS19, 1976.

Crank, J., The Mathematics of Diffusion, 2nd ed., Oxford University Press, 1975.

Deardorff, J. W., Dependence of Air-Sea Transfer coefficients on Bulk Stability, J. Geophys. Res., 73(8), 2549-2557, 1968.

Deardorff, J. W., Efficient prediction of ground surface temperature and moisture, with inclusion of a layer of vegetation, J. Geophys. Res., 83(C4), 1889-1903, 1978.

Eagleson, P. S., Dynamic Hydrology, McGraw-Hill, New York, 1970.

Eagleson, P. S., Climate, soil, and vegetation (7 parts), Water Resources Research, 14(5), 705-776, 1978.

Edlefsen, N. A. and A. B. C. Anderson, Thermodynamics of soil moisture, Hilgardia, 15(2), 1943.

Ibrahim, H. A. and W. Brutsaert, Intermittent infiltration into soils with hysteresis, J. Hydraulics Div. Proc. ASCE, 94(HY1), 113-137, 1968.

Idso, S. B., R. D. Jackson, R. J. Reginato, D. A. Kimball, and F. S. Nakayama, The dependence of bare soil albedo on soil water content, J. Appl. Met., 14, 109-113, 1975.

International Mathematical and Statistical Libraries, Incorporated, IMSL Library Reference Manual, Edition 8, 1980.

Kimball, B. A., R. D. Jackson, R. J. Reginato, F. S. Nakayama, and S. B. Idso, Comparison of field-measured and calculated soil-heat fluxes, Soil Sci. Soc. Am. Proc., 40(1), 18-25, 1976.

Lai, S., J. M. Tiedje, and A. E. Erickson, In situ measurement of gas diffusion coefficient in soils, Soil Sci. Soc. Am. Proc., 40(1), 3-6, 1976.

Milly, P. C. D. and P. S. Eagleson, Parameterization of moisture and heat fluxes across the land surface for use in atmospheric general circulation models, Tech. Rep. 279, R. M. Parsons Laboratory, Department of Civil Engineering, M.I.T., Cambridge, Massachusetts, 1982

Monin, A. S. and A. M. Obukhov, Osnovnye zakonomernosti turbulentnogo peremeshivaniya v prizemnom sloye atmosfery, Akad. Nauk SSSR, Trudy Geofiz. Inst. No. 24(151), 163-187, 1954.

Mualem, Y., A conceptual model of hysteresis, Water Resour. Res. 10(3), 514-520, 1974.

Mualem, Y., A new model for predicting the hydraulic conductivity of unsaturated porous media, Water Resour. Res., 12(3), 513-522, 1976.

Mualem, Y., Extension of the similarity hypothesis used for modeling the soil water characteristics, Water Resources Research, 13(4), 773-780, 1977.

Philip, J. R., The theory of infiltration: 1. The infiltration equation and its solution, Soil Sci., 83(5), 345-357, 1957.

Philip, J. R., The theory of infiltration, in Advances in Hydroscience, Vol. 5, edited by V. T. Chow, pp. 215-296, Academic, New York, 1969.

Philip, J. R. and D. A. deVries, Moisture movement in porous materials under temperature gradients, Trans. Amer. Geophys. Union, 38(2), 222-232, 1957.

Reeves, M. and E. E. Miller, Estimating infiltration for erratic rainfall, Water Resour. Res., 11(1), 102-110, 1975.

Rose, D. A., Water movement in porous materials: Part 2 - The separation of the components of water meovement, Brit. J. Appl. Phys., 14, 491-496, 1963.

Sellers, W. D., Physical Climatology, University of Chicago Press, 1965.

deVries, D. A., Thermal properties of soils, in Physics of Plant Environment, ed. W. R. Van Wijk, North-Holland Publishing Company, Amsterdam, 1966.

Appendix A

The Effective Hydraulic Conductivity due to Vapor Diffusion

Isothermal moisture flow occurs in the liquid and vapor phases. The vapor flow can be expressed as (Philip and deVries, 1957)

$$\rho_\ell \, q_v = -\nu \, D_a \, \Omega \, (n - \Theta) \, \frac{\partial \rho_v}{\partial z} \tag{A.1}$$

where $\rho_\ell \, q_v$ is the mass flux rate of vapor in the z-direction, ν is the mass flow factor (taken as unity), D_a is the molecular diffusivity of water-vapor in air given by Kimball et al. (1976) as

$$D_a = 0.229(1 + \frac{T}{273})^{1.75} \text{ cm}^2 \text{ s}^{-1} \tag{A.2}$$

Ω is the tortuosity of the air phase, n is porosity, Θ is moisture content, and ρ_v is absolute humidity.

Expansion of (A.1) in terms of $\frac{\partial \psi}{\partial z}$ and taking ν equal to unity gives

$$\begin{aligned} \rho_\ell \, q_v &= -0.229(1 + \frac{T}{273})^{1.75} \, \Omega \, (n - \Theta) \, \frac{d\rho_v}{d\psi} \frac{d\psi}{dz} \\ &= -D_{\psi v} \frac{d\psi}{dz} \end{aligned} \tag{A.3}$$

where $D_{\psi v}$ is the desired vapor analog of hydraulic conductivity. The absolute humidity is (Edlefson and Anderson, 1943, p. 145).

$$\rho_v(\psi, T) = \rho_o(T) \, \exp[\psi g / R_v(T + 273)] \tag{A.4}$$

in which ρ_o is the saturation value, g is the acceleration of gravity, and R_v is the gas constant for water vapor. Using (A.3) and (A.4), we find

$$D_{\psi v} = 0.229 \left(1 + \frac{T}{273}\right)^{1.75} \Omega \, (n - \Theta) \cdot \frac{g\rho_o(T)}{R_v(T+273)} \exp[\psi g/R_v(T + 273)] \tag{A.5}$$

In SWAHP, the temperature factors are evaluated at the start of the simulation using the input typical soil temperature, so (A.5) is later evaluated in DE as

$$D_{\psi v} = \text{DPART} \cdot \Omega \cdot (n - \Theta) \cdot \exp(\psi \cdot \text{GORT}) \tag{A.6}$$

where DPART and GORT are constants.

Appendix B

Transfer Parameterization for the Atmosphere

Equations (2.62) and (implicitly) (2.63) express the potential evaporation rate, E_p, and the sensible heat flux, H, as

$$\rho_\ell E_p = C_w U_a[\rho_o(T_1) - \rho_{va}] \tag{B.1}$$

and

$$H = \rho_a c_p C_H U_a(T_1 - T_a) \tag{B.2}$$

in which the bulk transfer coefficients C_w and C_H are dependent upon atmospheric stability. In order to quantify this dependence, we borrow the relations used by Anderson (1976), which in turn are based heavily on those of Deardorff's (1968) application of the similarity theory of Monin and Obukhov (1954).

Under conditions of neutral stability, the coefficients are given by

$$(C_M)_N = (C_H)_N = (C_W)_N = \frac{k^2}{(\ln \frac{z_a}{z_o})^2} \tag{B.3}$$

where the N subscript denotes neutral conditions, k is von Karman's constant (=0.4), z_a is the screen height, and z_o is the surface roughness length. C_M is the drag coefficient.

Under unstable conditions, the bulk transfer coefficients are related to their neutral values through.

$$\frac{C_M}{(C_M)_N} = \left\{1 - \frac{(C_M)_N^{\frac{1}{2}}}{k}\left[\ln\left(\frac{1+x^2}{2}\right) + 2\ \ln\left(\frac{1+x}{2}\right) - 2\ \tan^{-1}(x) + \frac{\pi}{2}\right]\right\}^{-2} \tag{B.4}$$

and

$$\frac{C_H}{(C_H)_N} = \frac{C_W}{(C_W)_N} = \left[\frac{C_M}{(C_M)_N}\right]^{\frac{1}{2}}\left[1 - \frac{2}{k}\ (C_M)_N^{\frac{1}{2}}\ \ln\left(\frac{1+x^2}{2}\right)\right]^{-1} \tag{B.5}$$

in which x is a function of stability,

$$x = (1 - 16\ z_a/L)^{\frac{1}{4}} \tag{B.6}$$

where L is the Monin-Obukhov length. It is related to the bulk Richardson number,

$$(Ri)_B \equiv \frac{2g\ z_a(T_a - T_1)}{(T_a + T_1)\ U_a^2} \tag{B.7}$$

T_1 being the surface temperature, by the expression

$$\frac{z_a}{L} = \frac{k\ \dfrac{C_H}{(C_M)_N}}{(C_M)_N^{\frac{1}{2}}\left[\dfrac{C_M}{(C_M)_N}\right]^{3/2}}\ (Ri)_B \tag{B.8}$$

For stable conditions, assuming equality of the stability functions,

$$\frac{C_M}{(C_M)_N} = \frac{C_H}{(C_H)_N} = \frac{C_W}{(C_W)_N} = \begin{cases} \left(1 - \frac{(Ri)_B}{Ri_{cr}}\right)^2 & , (Ri)_B < Ri_{cr} \\ 0 & , (Ri)_B \geq Ri_{cr} \end{cases} \tag{B.9}$$

where Ri_{cr} is the cirtical Richardson number, above which there is no turbulence. We employ the value $Ri_{cr} = 0.2$.

Appendix C

Calculation of the Hydraulic Conductivity

The hydraulic conductivity is expressed as

$$K = CKSAT \cdot K_r(\Theta) \tag{C.1}$$

where $K_r(\Theta)$ is the relative hydraulic conductivity (unity when soil is fully wetted), and CKSAT is a constant, the hydraulic conductivity of the fully wetted soil. For isothermal simulations of non isothermal natural conditions, Milly and Eagleson (1982) suggest evaluating CKSAT at the average daily maximum air temperature.

The relative hydraulic conductivity is expressed using Mualem's (1976) formulation,

$$K_r(\Theta) = \left(\frac{\Theta}{\Theta_u}\right)^{\frac{1}{3}} \left[\int_0^{\Theta} \frac{d\Theta'}{\psi(\Theta')}\right]^2 \left[\int_0^{\Theta_u} \frac{d\Theta}{\psi(\Theta)}\right]^{-2} \tag{C.2}$$

Because there is little hysteresis in the $K_r(\Theta)$ curve, the relation between ψ and Θ is arbitrarily taken to be the main wetting curve to evaluate the integrals.

The integrals are evaluated numerically by approximating $\psi(\Theta)$ with a piece-wise linear relation between $pF = \log(-\psi)$ and Θ_w.

Thus

$$pF = PFR_k - SS_k \cdot (\Theta_w - XR_k) \tag{C.3}$$

for Θ_w in the range (XR_k, XR_{k+1}) and pF in the range (PFR_{k+1}, PFR_k) where the intercepts PRF_k are equally spaced between log(-PRN) and 7. Then the XR_k are given by

$$XR_k = \Theta_w(PFR_k) \tag{C.4}$$

and the slopes are given by

$$SS_k = \frac{PFR_k - PFR_{k+1}}{XR_{k+1} - XR_k} \tag{C.5}$$

We also define

$$PR_k = -10^{PFR_k} \tag{C.6}$$

Using this representations for $\psi(\Theta_w)$, we may evaluate the integrals in (C.2). In general,

$$\int_0^{\Theta_w} \frac{d\Theta}{\psi(\Theta)} = - \int_0^{\Theta} 10^{-pF}\, d\Theta \tag{C.7}$$

Substituting our piecewise expression for pF and integrating,

$$\int_0^{\Theta_w} \frac{d\Theta}{\psi(\Theta)} = \left(\Theta_u \ \ell n\ 10\right)^{-1} \cdot \left\{ \sum_{i=1}^{M-1} SS_i^{-1} \cdot \left(PR_{1+1}^{-1} - PR_i^{-1}\right) + SS_m^{-1} \left(PWET^{-1} - PR_m^{-1}\right)\right\} \tag{C.8}$$

for PWET in the range (PR_{m+1}, PR_m) where PWET is defined as

$$PWET = \psi(\Theta_w) \tag{C.9}$$

For convenience, we define

$$ENT_m = \sum_{i=1}^{m} SS_i^{-1}(PR_{i+1}^{-1} - PR_i^{-1}) \tag{C.10}$$

Then (C.2) can be expressed as

$$K_r(\Theta) = (\Theta/\Theta u)^{\frac{1}{2}} \cdot \left\{ \left[ENT_{m-1} + SS_m^{-1} \cdot \left(PWET^{-1} - PR_m^{-1} \right) \right] \cdot ENT_N^{-1} \right\}^2 \qquad \text{(C.11)}$$

for PWET in the range (PR_{m+1}, PR_m). The subscript N indicates the highest value in the range, the one for which

$$PFR_N = \log(-PRN) \qquad \text{(C.12)}$$

and

$$XR_N = \Theta_u \qquad \text{(C.13)}$$

The values of the PFR, PR, SS, and ENT are found at the start of a simulation in SOILI. Equation (C.11) is then applied for the calculation of hydraulic conductivity by the function HYCON.

Appendix D

Calculation of the Thermal Conductivity

The thermal conductivity of a moist soil is calculated using a weighted average (deVries, 1966),

$$\lambda = \frac{\sum_{i=1}^{5} k_i \, \Theta_i \, \lambda_i}{\sum_{i=1}^{5} k_i \, \Theta_i} \qquad \text{(D.1)}$$

where λ_i is the thermal conductivity of the i'th constituent (table below), k_i is the ratio of the average temperature gradient in the i'th constituent to the average temperature gradient of the bulk medium, and Θ_i is the volumetric fraction of constituent i. A conceptual model yields

$$k_i = \frac{2}{3}\left[1 + \left(\frac{\lambda_i}{\lambda_1} - 1\right) g_i\right]^{-1} + \frac{1}{3}\left[1 + \left(\frac{\lambda_i}{\lambda_1} - 1\right)\left(1 - 2g_i\right)\right]^{-1} \qquad \text{(D.2)}$$

where the liquid phase is considered continuous and g_i is the "shape factor" of the i'th constituent. For the solid particles, constant values as given in the table below are assumed. No value is needed for g_1, since its coefficient is zero. The value of g_2 is considered a function of moisture content as follows (Kimball et al., 1976):

$$g_2 = \begin{cases} .013 + \left(\dfrac{.022}{\Theta_w(\text{pF}=4.2)} + \dfrac{.298}{n}\right)\Theta & \Theta < \Theta_w(\text{pF}=4.2) \\ .035 + \dfrac{.298}{n}\,\Theta & \Theta_w(\text{pF}=4.2) < \Theta \end{cases} \qquad \text{(D.3)}$$

where Θ_w(pF=4.2) is the wilting point.

The vapor distillation effect, which enhances the thermal conductivity of the air phase, is ignored. Milly and Eagleson (1982) found that this vapor contribution did not affect surface fluxes.

Values of λ obtained using this model are good down to a moisture content approximately equal to Θ_k, the content at which the liquid phase becomes discontinuous. At total dryness, the same theory may be applied by substituting λ_2 for λ_1 in (D.2), i.e., by taking air as the continuous phase. In this case, a correction factor for λ of 1.25 is required. Between $\Theta = 0$ and $\theta = \theta_k$, values of λ are usually estimated by an interpolation scheme. In our scheme, we first evaluate λ at Θ_k in the usual manner, and then we use the known value of λ for $\Theta = 0$ to find λ at Θ by means of linear interpolation.

The equations presented above are used by ETCON to find thermal conductivity as a function of moisture content, XXX. From (D.1),

$$\mathrm{ETCON} = \frac{\mathrm{PS1 + XXX \cdot TCON(1) + (POR\text{-}XXX) \cdot TCON(2) \cdot GRAT}}{\mathrm{PS2 + XXX + (POR\text{-}XXX) \cdot GRAT}} \tag{D.4}$$

where GRAT = k_2 is given by (D.2) and (D.3). The partial sums, PS1 and PS2, are pre-calculated in SOILI. They are

$$\mathrm{PS1} = \sum_{i=3}^{5} k_i \, \Theta_i \, \lambda_i \tag{D.5}$$

$$\mathrm{PS2} = \sum_{i=1} k_i \, \Theta_i \tag{D.6}$$

The conductivity (cal cm^{-1} s^{-1} $°K^{-1}$) and shape factor of each constituent is listed below:

Constituent	i	λi	g_i
liquid water	1	$1.37x10^{-3}$	-
air	2	$6x10^{-5}$	See text.
quartz	3	$2.1x10^{-2}$	0.125
other minerals	4	$7x10^{-3}$	0.125
organic matter	5	$6x10^{-4}$	0.5

Appendix E

Program Source Code

```
C*****************************************************************************
C
C                              SWAHP
C                              =====
C
C               SOIL WATER AND HEAT PARAMETERIZATION
C               ------------------------------------
C
C          THIS IS A SIMULATION MODEL OF MOISTURE AND HEAT FLOW ACROSS A
C     BARE SOIL SURFACE.  SURFACE MOISTURE FLUXES ARE CALCULATED USING
C     THE TIME COMPRESSION ASSUMPTION IN CONJUNCTION WITH SPECIAL
C     SOLUTIONS OF THE NONLINEAR MOISTURE DIFFUSION EQUATION. EFFECTS
C     OF WATER VAPOR AND HYSTERESIS ARE INCORPORATED.  A NEAR SURFACE
C     MOISTURE CONCENTRATION IS CALCULATED CONTINUOUSLY, AND REDISTRI-
C     BUTION IS CONSIDERED.
C
C          POTENTIAL EVAPORATION IS CALCULATED USING THE SURFACE TEMP-
C     ERATURE GIVEN BY THE FORCE-RESTORE PROCEDURE.  THE TEMPERATURE AND
C     THE MOISTURE FLUX EQUATIONS ARE SOLVED SIMULTANEOUSLY, THEIR COUP-
C     LING COMING MAINLY THROUGH THE EVAPORATION RATE.
C
C     WRITTEN BY P. CHRISTOPHER D. MILLY.  FOR FURTHER DOCUMENTATION,
C     REFER TO THE FOLLOWING TECHNICAL REPORTS OF M.I.T.'S RALPH
C     M. PARSONS LABORATORY:
C
C     1. MILLY, P.C.D., SWAHP: THEORY AND COMPUTER PROGRAM, T.R. 2XX,
C          1982.  PRESENTS THE THEORY BEHIND SWAHP IN DETAIL. GIVES
C          A FULL DESCRIPTION OF THE STRUCTURE AND OPERATION OF THE
C          COMPUTER PROGRAM. ALSO HAS DETAILS ON INPUT FILES AND
C          PROGRAM USAGE, INCLUDING A TEST PROBLEM.
C
C     2. MILLY, P.C.D., AND P. S. EAGLESON, PARAMETERIZATION OF MOISTURE
C          AND HEAT FLUXES ACROSS THE LAND SURFACE FOR USE IN ATMOSPHERIC
C          GENERAL CIRCULATION MODELS, T.R. 279, 1982.  GIVES AN OUTLINE
C          OF AN EARLIER VERSION OF SWAHP. PRESENTS COMPARISONS OF
C          SWAHP CALCULATIONS WITH THOSE OF A DETAILED FINITE ELEMENT
C          SOLUTION OF THE COMPLETE GOVERNING EQUATIONS.
C
C*****************************************************************************
C
C     COPYRIGHT PAUL CHRISTOPHER DAMIAN MILLY AND THE MASSACHUSETTS INSTITUTE
C     OF TECHNOLOGY 1982.
C
C*****************************************************************************
C
C     THE TEXT THAT FOLLOWS CONSISTS OF THE FOLLOWING ITEMS:
C
C          LIST OF VARIABLES
C          MAIN PROGRAM
C          SUBROUTINES
C             INIT
C             START
C             PERIOD
C             END
C             SOILI
C             SORPI
C             PHII
C             FOREST
C             REDIST
C             OUTPUT
```

```
C         FUNCTIONS
C            ALB
C            DE
C            DI
C            DSORP
C            EHCAP
C            EMI
C            ETCON
C            FC
C            FINV
C            FXBAR
C            HYCON
C            PHI
C            RHOZ
C            SORP
C            TC
C            TORT
C            XWET
C         BLOCK DATA
C
C*****************************************************************************
C
C     LIST OF VARIABLES
C     =================
C
C     NAME          DESCRIPTION
C     ====          ===========
C
C    AO         CONSTANT TERM IN INFILTRATION CAPACITY
C    AA         FIRST PARAMETER OF MAIN WETTING CURVE
C    ALBEDO     ALBEDO OF SURFACE
C    ALDRY      ALBEDO OF SURFACE WHEN TOTALLY DRY
C    ALWET      ALBEDO OF SURFACE WHEN TOTALLY WET
C    BB         SECOND PARAMETER OF MAIN WETTING CURVE
C    C1         FIRST FORCE-RESTORE PARAMETER
C    C1PART     CONSTANT FACTOR IN C1
C    C2         SECOND FORCE-RESTORE PARAMETER
C    C3         THIRD FORCE-RESTORE PARAMETER
C    CAPEX      MINIMUM OF CAPEX1 AND CAPEX2
C    CAPEX1     AVERAGE EXFILTRATION CAPACITY FOR THE PERIOD, DETERMINED BY
C               DIFFUSION LIMITATION
C    CAPEX2     EXFILTRATION LIMIT DUE TO FINITE WATER STORAGE, EXPRESSED
C               AS A RATE
C    CAPIN      AVERAGE INFILTRATION CAPCITY FOR THE PERIOD
C    CC         THIRD PARAMETER OF MAIN WETTING CURVE
C    CD         DRAG COEFFICIENT
C    CDN        NEUTRAL DRAG COEFFICIENT
C    CDRAT      CD/CDN IN SUBROUTINE PHII
C    CDU        PRODUCT OF CDN, UA, AND STAB
C    CDUOR      CDU/RHOL
C    CH         TURBULENT TRANSFER COEFFICIENT FOR HEAT
C    CHRAT      VECTOR OF STAB VALUES
C    CKSAT      HYDRAULIC CONDUCTIVITY FOR MOISTURE CONTENT EQUAL TO XU
C    CL         SPECIFIC HEAT OF LIQUID WATER
C    CONO       THERMAL CONDUCTIVITY OF DRY SOIL
C    CP         SPECIFIC HEAT OF WATER AT CONSTANT PRESSURE
C    CPNOLD     SAVES THE LATEST VALUE OF CAPIN FROM CURRENT PERIOD.
C    DAL        DERIVATIVE OF ALBEDO W.R.T. X FOR ALBEDO GREATER THAN ALWET
C    DAYS       ELAPSED TIME IN DAYS AT END OF CURRENT PERIOD
C    DD         FOURTH PARAMETER OF MAIN WETTING CURVE
```

```
C    DEDT       DERIVATIVE OF EVAPORATION WITH RESPECT TO T1
C    DELT       DURATION OF THE SIMULATION PERIOD
C    DELT1      TIME FROM START OF PERIOD UNTIL CURRENT MASS WOULD BE EXHAUSTED
C    DEM        DERIVATIVE OF EMISSIVITY W.R.T. MOISTURE CONTENT
C    DF         NUMBER OF SECONDS PER DAY (86400)
C    DFIL       FILT-EXFILT
C    DFDT       DERIVATIVE OF F W.R.T. T1
C    DPART      A CONSTANT FACTOR IN THE VAPOR CONDUCTIVITY
C    DR         DERIVATIVE OF SATURATION ABSOLUTE HUMIDITY W.R.T. T1
C    DRADLU     DERIVATIVE OF RADLU WITH RESPECT TO T1
C    DRI        DERIVATIVE OF RI W.R.T. T1
C    DSTAB      DERIVATIVE OF STAB W.R.T. RI
C    DSENS      DERIVATIVE OF SENS W.R.T. T1
C    E          EVAPORATION RATE
C    EO         E EVAULATED AT T1LAST
C    EE         FIFTH PARAMETER OF MAIN WETTING CURVE
C    ELO        LATENT HEAT OF VAPORIZATION AT FREEZING POINT
C    EMDRY      EMISSIVITY OF SOIL WHEN TOTALLY DRY
C    EMIS       EMISSIVITY OF SOIL
C    EMWET      EMISSIVITY OF SOIL WHEN TOTALLY WET
C    ENDAYS     NUMBER OF DAYS USED IN C3 PARAMETER
C    ENT(25)    INTEGRAL USED IN EVALUATION OF HYCON
C    EP         POTENTIAL EVAPORATION RATE
C    EXFILT     AVERAGE EXFILTRATION RATE DURING PERIOD
C    F          THE RESIDUAL IN THE HEAT BALANCE EQUATION WHEN EVALUATED AT T1LAST
C    FE         CUMULATIVE EXFILTRATION DEPTH
C    FEP        APPARENT CUMULATIVE EXFILTRATION DEPTH
C    FI         CUMULATIVE INFILTRATION DEPTH
C    FILT       AVERAGE RATE OF INFILTRATION FOR THE PERIOD
C    G          HEAT FLUX INTO THE GROUND
C    GA1        Y-INTERCEPT FOR CALCULATION OF AIR SHAPE FACTOR WHEN X IS GREATER
C               THAN XWILT
C    GA2        Y-INTERCEPT FOR CALCULATION OF AIR SHAPE FACTOR WHEN X IS LESS
C               THAN XWILT
C    GB1        SLOPE CORRESPONDING TO GA1
C    GB2        SLOPE CORRESPONDING TO GA2
C    GORT       ACCELERATION OF GRAVITY DIVIDED BY WATER GAS CONSTANT AND KELVIN
C               TEMPERATURE
C    GRAT       RATIO OF THERMAL GRADIENT IN A PARTICULAR PHASE TO THE BULK
C               THERMAL GRADIENT
C    HCAP(5)    VECTOR WITH SPECIFIC HEAT OF EACH PHASE
C    HCD        BULK VOLUMETRIC HEAT CAPCITY OF SOIL WHEN TOTALLY DRY
C    ICO        ROW DIMENSION OF SICO AND SEECO
C    IEOF       END OF OUTPUT FILE MARKER (=-1)
C    IER        ERROR CODE USED BY THE IMSL SUBPROGRAMS
C    IFOR       FORMATTED OUTPUT CODE
C    IRAIN      CODE FOR PRINTING OF RAIN INFORMATION
C    IRPT       INDICATOR WITH VALUE OF O FOR FIRST FORCE-RESTORE SOLUTION (E=EP)
C               AND 1 FOR SECOND
C    ISOR       CODE FOR SORPTIVITY/DESORPTIVITY CALCULATION
C    ISTK       CODE FOR PRINTOUT OF STACK INFORMATION
C    IUNF       CODE FOR UNFORMATTED OUTPUT
C    MAXSTK     MAXIMUM NUMBER OF PREVIOUS SETS OF EXFILTRATION DATA THAT CAN BE
C               SAVED IN STACK (=10)
C    NPER       SEQUENCE NUMBER OF CURRENT SIMULATION PERIOD
C    NSS        NUMBER OF SEGMENTS IN THE APPROXIMATE RETENTION CURVE USED TO
C               EVALUATE HYDRAULIC CONDUCTIVITY INTEGRALS
C    NSTACK     NUMBER OF SETS OF DATA CURRENTLY STORED IN STACK
C    NX         NUMBER OF ELEMENTS IN XARRAY, YARRAY, ETC. (=21)
C    P          RATE OF PRECIPITATION
```

```
C    P1          LOWER LIMIT OF PSI FOR SORPTIVITY AND DESORPTIVITY INTEGRATIONS
C    P2          UPPER LIMIT OF PSI FOR SORPTIVITY AND DESORPTIVITY INTEGRATIONS
C    PFRM        MAXIMUM VALUE OF LOG(-PSI) (=7)
C    PFRN        VALUE OF LOG(-PSI) SEPARATING SATURATION AND UNSATURATION
C    PFR(25)     VECTOR OF INTERCEPTS FOR SEGMENTS OF THE APPROXIMATE RETENTION
C                CURVE USED TO EVALUATE THE HYDRAULIC CONDUCTIVITY INTEGRALS
C    POR         POROSITY
C    PP          MATRIC POTENTIAL IN DE AND DI
C    PR(25)      VALUES OF PSI CORRESPONDING TO PFR(25)
C    PRN         VALUE OF PSI CORRSPONDING TO PFRN
C    PS1         PARTIAL SUM IN NUMERATOR OF EFFECTIVE THERMAL CONDUCTIVITY
C    PS2         PARTIAL SUM IN DENOMINATOR OF EFFECTIVE THERMAL CONDUCTIVITY
C    Q(8)        VECTOR OF FLUX RATES FOR CURRENT PERIOD, TIME IN DAYS
C    R           REDISTRIBUTION PARAMETER
C    RADLD       LONGWAVE RADIATION FROM SKY
C    RADLUO      BACK LONGWAVE RADIATION EVALUATED AT T1LAST
C    RADS        INCOMING SOLAR RADIATION
C    RAIN        LOGICAL VARIABLE THAT TELLS IF IT'S RAINING
C    RAINB4      VALUE OF RAIN FROM LAST PERIOD
C    RCCDU       RHOCP TIMES CDU
C    RH          ABSOLUTE HUMIDITY AT TEMPERATURE T1LAST
C    RHOCP       PRODUCT OF DENSITY OF AIR AND SPECIFIC HEAT OF AIR AT CONSTANT
C                PRESSURE
C    RHOL        DENSITY OF LIQUID WATER
C    RHOVA       ABSOLUTE HUMIDITY OF AIR AT SCREEN HEIGHT
C    RI          BULK RICHARDSON NUMBER
C    RIB         VECTOR OF RI VALUES FOR INTERPOLATION OF STAB
C    RIPART      A FACTOR IN RI
C    RLCL        PRODUCT OF RHOL AND CL
C    RLE         PRODUCT OF RHOL, E, AND (LO+CP*T1)
C    RLL         PRODUCT OF RHOL AND LATENT HEAT OF VAPORIZATION OF WATER
C    RN          NET RADIATION
C    ROOTO       SQUARE ROOT OF PRODUCT OF HEAT CAPACITY AND THERMAL CONDUCTIVITY,
C                EVALUATED FOR DRY SOIL
C    ROOT1       SAME AS ROOTO, EVALUATED USING XBAR
C    ROOT2       SAME AS ROOTO, EVALUATED USING XHAT
C    RS          RATE OF SURFACE RUNOFF
C    SO          SORPTIVITY USED IN THE BASIC INFILTRATION EQUATION
C    SE          DESORPTIVITY
C    SE1         VECTOR OF DESORPTIVITY FOR CURRENT XR AND VARIOUS XBAR
C    SEARAY      2-D ARRAY OF DESORPTIVITY VALUES
C    SEE         TEMPORARY VECTOR OF DESORPTIVITY USED IN SETTING UP SE1
C    SEECO       BI-CUBIC SPLINE COEFFICIENT ARRAY CORRESPONDING TO SE1 (ALSO
C                TO SEE TEMPORARILY)
C    SENS        RATE OF SENSIBLE HEAT DIFFUSION TO THE ATMOSPHERE
C    SENSO       VALUE OF SENS EVALUATED AT T1=T1LAST
C    SF(5)       VECTOR OF SHAPE FACTOR FOR EACH CONSTITUENT
C    SIARAY      VECTOR OF SORPTIVITY VALUES
C    SICO        CUBIC SPLINE COEFFICIENTS FOR INTERPOLATION OF SIARRAY AND XARRAY
C    SIG         STEFAN-BOLTZMAN CONSTANT
C    SS(25)      VECTOR OF SLOPES CORRESPONDING TO PFR(25)
C    STAB        RATIO OF ACTUAL TO NEUTRAL TRANSFER COEFFICIENT FOR TURBULENT
C                TRANSFER OF HEAT AND VAPOR
C    STACK(6,10) ARRAY FOR STORAGE OF PAST VALUES OF FE, XBAR, XZ, FEP, XR,
C                TOGETHER WITH THE TIME THEY WERE STORED
C    SUM(8)      VECTOR CONTAINING RUNNING SUM OF FLUX RATES FOR PERIODS, TIME
C                IN DAYS
C    T1          SURFACE TEMPERATURE
C    T1LAST      VALUE OF T1 AT START OF PERIOD AND FIRST GUESS FOR NEW VALUE
C    T2          DEEP (ABOUT ONE METER) SOIL TEMPERATURE
```

```
C    TA         AIR TEMPERATURE AT SCREEN HEIGHT
C    TAVG       AN AVERAGE SOIL TEMPERATURE (INPUT) FOR EVALUATION OF VAPOR
C               TERMS DPART AND GORT
C    TCE        COMPRESSED TIME OF EXFILTRATION
C    TCI        COMPRESSED TIME OF INFILTRATION
C    TCON(5)    VECTOR OF THERMAL CONDUCTIVITY OF EACH SOIL CONSTITUENT
C    TIME       ELAPSED TIME AT END OF CURRENT SIMULATION PERIOD
C    TITLE      ANY 80-CHARACTER STRING
C    TK         TEMPERATURE, DEGREES KELVIN
C    TRAIN      DURATION OF STORM
C    TRED       DURATION OF REDISTRIBUTION INTERVAL
C    TSTAB      DERIVATIVE OF STAB W.R.T. T1, DIVIDED BY STAB
C    TSTART     TIME AT WHICH LAST RAIN BEGAN
C    UA         WINDSPEED AT SCREEN HEIGHT
C    VK         VON KARMAN CONSTANT
C    VPER(5)    VECTOR OF VOLUMETRIC FRACTION OF EACH SOIL CONSTITUENT
C    X          SOIL MOISTURE CONTENT IN SOME FUNCTION SUBPROGRAMS
C    X1         MOISTURE CONTENT AT PSI=P1
C    X2         MOISTURE CONTENT AT PSI=P2
C    XARRAY     VECTOR OF MOISTURE CONTENT USED FOR INTERPOLATION OF SORPTIVITY
C               AND DESORPTVITY
C    XB         'INITIAL' MOISTURE CONTENT IN INFILTRATION AND EXFILTRATION
C               SOLUTIONS
C    XBAR       CONCEPTUAL NEAR-SURFACE SOIL MOISTURE CONTENT RESULTING FROM
C               INFILTRATION AND REDISTRIBUTION
C    XHAT       AN AVERAGE DEEP-SOIL MOISTURE CONTENT
C    XI(25)     VALUES OF MOISTURE CONTENT CORRESPONDING TO PFR(25)
C    XIM        VALUE OF MOISTURE CONTENT CORRESPONDING TO PFRM
C    XK         VALUE OF MOISTURE CONTENT WHEN LIQUID FLOW BECOMES NEGLIGIBLE
C    XR         VALUE OF XBAR IMMEDIATELY AFTER A STORM, THE REVERSAL VALUE
C    XU         MOISTURE CONTENT WHEN FREELY WETTED FROM DRYNESS
C    XW         IN SORPI, THE VALUE OF THE MAIN WETTING FUNCTION CORRESPONDING
C               TO THE VALUES OF XWR AND XB
C    XWILT      MOISTURE CONTENT AT THE WILTING POINT
C    XWR        VALUE OF MOISTURE CONTENT AT DRYING REVERSAL FOR CALCULATION
C               OF DESORPTVITY
C    XZ         THE DEPTH OF INFILTRATION FROM THE LATEST STORM
C    Y          RATIO OF XBAR/XR
C    YARRAY     ARRAY OF Y VALUES USED FOR INTERPOLATION OF DESORPTIVITY
C    ZO         SURFACE ROUGHNESS LENGTH
C    ZA         SCREEN HEIGHT FOR TA, UA, AND RHOVA
C    ZL         RATIO OF ZA TO MONIN-OBUKHOV LENGTH
C
C
C
C***********************************************************************
C
C      THE MAIN PROGRAM FOLLOWS:
C
C=======================================================================
C
C    INITIALIZATION PHASE
C
       CALL INIT
C
C=======================================================================
C
C    SIMULATION PHASE
C
   10  CALL START
```

```
      CALL PERIOD
      CALL END
      GO TO 10
      END
C
C****** END OF MAIN PROGRAM ***************************************************
C****** SUBROUTINE AND FUNCTION SUBPROGRAMS FOLLOW. ***************************
C/////////////////////////////////////////////////////////////////////////////
C/////////////////////////////////////////////////////////////////////////////
      SUBROUTINE INIT
C/////////////////////////////////////////////////////////////////////////////
C/////////////////////////////////////////////////////////////////////////////
C
C         THIS SUBROUTINE IS CALLED ONCE AT THE BEGINNING OF A SIMULATION
C    TO PERFORM INITIAL INPUT AND OUTPUT OPERATIONS, TO INITIALIZE
C    CONSTANTS AND VARIABLES, AND TO SET UP VARIOUS FUNCTIONS.
C
C=============================================================================
      LOGICAL*1 RAIN,RAINB4
      COMMON /BIG/  STACK(6,10), SUM(8), Q(8),
     * ALBEDO,      C1, C1PART,      C2,      C3,  CAPEX, CAPEX1,
     * CAPEX2,  CAPIN,     CDN,  CDUOR,      CL,      CP, CPNOLD,    DEDT,
     *   DELT,      DF,   DFIL, DRADLU,  DSENS,       E,      EO,     ELO,
     *   EMIS,      EP,     FE,     FEP,      FI,       G,    SENS,      P,
     *      R,  RADLD, RADLUO,   RADS,  RCCDU,  RHOCP,   RHOL,  RHOVA,
     *   RLCL,     RLE,     RN,      RS,  ROOTO,  ROOT1,  ROOT2,  SENSO,
     *    SIG,      T1, T1LAST,   T1OLD,      T2,      TA,    TCE,    TCI,
     *   TIME,  TSTAB,    XBAR,  XINIT,      XR,      XZ,     ZA,
     *   IFOR,  IRAIN,    ISTK,    IUNF, MAXSTK,    NPER, NSTACK,
     *   RAIN, RAINB4
      COMMON /VAPCOM/ DPART,GORT
      COMPLEX RHOZ
      CHARACTER*80 TITLE
C=============================================================================
C
C     READ INITIAL INPUT.
C
C=============================================================================
      READ(15,1010) TITLE
      READ(15,1020) IFOR,IUNF,IRAIN,ISTK,ISOR
      READ(15,1030) R,TAVG,ENDAYS,ZO,ZA,XHAT,T1,T2
      XBAR=XHAT
      XR=XHAT
C=============================================================================
C
C   WRITE THE INITIAL OUTPUT.
C
C=============================================================================
      IF(IFOR.EQ.0) GO TO 10
      ITEM=IABS(IFOR)
      WRITE(ITEM,2010)
      WRITE(ITEM,2020) TITLE
      WRITE(ITEM,2030) IFOR,IUNF
      WRITE(ITEM,2040) R,TAVG,ENDAYS,ZO,ZA,XHAT,T1,T2
      IF(IFOR.GT.0) WRITE(ITEM,2050)
      IF(IFOR.LT.0) WRITE(ITEM,2060)
   10 IF(IUNF.GT.0) WRITE(IUNF) TITLE,R,TAVG,ENDAYS,ZO,ZA,XHAT,T1,T2
C=============================================================================
C
C   GIVEN AN AVERAGE SOIL TEMPERATURE, CALCULATE TEMPERATURE-
```

```
C   DEPENDENT TERMS IN THE EFFECTIVE VAPOR CONDUCTIVITY COEFFICIENT.
C   THESE ARE FOR USE BY FUNCTION DE WHEN SORP PERFORMS THE INTEGRATION
C   FOR DESORPTIVITY.
C
C===========================================================================
      RH=REAL(RHOZ(TAVG))
      TK=TAVG+273.16
      GORT=2.13E-4/TK
      DPART=5.8E-7*TK**2.3*RH*GORT/RHOL
C===========================================================================
C
C   SOILI WILL READ THE SOIL PROPERTY DATA AND COMPUTE
C   THE RELEVANT DERIVED CONSTANTS FOR USE IN THE SOIL FUNCTIONS.
C
C===========================================================================
      CALL SOILI
C===========================================================================
C
C     CALCULATE NEUTRAL DRAG COEFFICIENT AND SET UP TABLE OF STABILITY
C     FACTOR VERSUS BULK RICHARDSON NUMBER.
C
C===========================================================================
      CDN=0.16*(ALOG(ZA/ZO))**-2.
      CALL PHII(CDN)
C===========================================================================
C
C     SET UP ARRAYS OF DESORPTIVITY AND SORPTIVITY AS FUNCTIONS OF MOISTURE
C     CONTENT.
C
C===========================================================================
      CALL SORPI(ISOR)
C===========================================================================
C
C   INITIALIZE CONSTANTS.
C
C===========================================================================
      ROOTO=SQRT(ETCON(0.)*EHCAP(0.))
      ROOT2=SQRT(ETCON(XHAT)*EHCAP(XHAT))
      C1PART=2.*SQRT(3.14159/DF)
      C2=2.*3.14159/DF
      C3=1./(ROOT2*SQRT(ENDAYS*DF))
C===========================================================================
C
C   INITIALIZE VARIABLES.
C
C===========================================================================
      TIME=0.0
      NPER=0
      FEP=0.
      FE=0.
      FI=0.
      XZ=1.E-6
      TCE=0.
      TCI=0.
      NSTACK=0
      DO 20 I=1,8
  20  SUM(I)=0.
C===========================================================================
1010  FORMAT(A80)
1020  FORMAT(5I4)
```

```
1030  FORMAT(8F10.0)
2010  FORMAT(///,80(1H*)///32X,17(1H*)/32X,1H*,15X,1H*/32X,1H*,
     1  '       SWAHP      ',1H*/32X,1H*,15X,1H*/32X,17(1H*)//)
2020  FORMAT(//1X,A80)
2030  FORMAT(//5X,'FORMATTED OUTPUT GOING TO FILE NUMBER',I3/
     * 5X,'UNFORMATTED OUTPUT TO                    ',I3)
2040  FORMAT(//5X,'INPUT:'/5X,' R =',E10.2,2X,'TAVG =',E10.2,2X,
     * 'ENDAYS =',E10.2,
     * '  ZO =',E10.2,/5X,'ZA =',E10.2,'  XHAT =',E10.2,6X,'T1 =',E10.2,
     * '  T2 =',E10.2)
2050  FORMAT(////5X,'OUTPUT FOR EACH PERIOD FOLLOWS.'//
     * 14X,'DAYS',12X,'P',12X,'E',12X,'XBAR',10X,'T1'/)
2060  FORMAT(////5X,'DETAILED OUTPUT FOR EACH PERIOD FOLLOWS.',
     * ' THE FORMAT IS SHOWN HERE:'//
     * 9X,'NPER',6X,'TIME',6X,'P   ',6X,'EP  ',6X,'E   ',6X,'DFIL',
     * 6X,'RS  '/ 9X,'RN  ',6X,'G   ',6X,'H   ',6X,'RLE ',6X,'XBAR',6X,
     * 'XR  ',6X,'XZ  '/  9X,'FEP ',6X,'FE  ',6X,'FI  ',6X,'TCE ',6X,
     * 'TCI ',6X,'T1  ',6X,'T2  ')
      RETURN
      END
C/////////////////////////////////////////////////////////////////////////////
C/////////////////////////////////////////////////////////////////////////////
      SUBROUTINE START
C/////////////////////////////////////////////////////////////////////////////
C/////////////////////////////////////////////////////////////////////////////
C
C        THIS SUBROUTINE READS DATA DEFINING THE BOUNDARY CONDITIONS
C   FOR A NEW SIMULATION PERIOD. PREPARATIONS FOR THE SUBROUTINE PERIOD
C   ARE MADE. WHEN RAIN STARTS OR ENDS, THE APPROPRIATE STATE VARIABLES
C   ARE RE-INITIALIZED. WHEN IT IS RAINING, THE INFILTRATION CAPACITY
C   IS FOUND. IN EITHER CASE, THE EXFILTRATION LIMITS ARE FOUND.
C   REDISTRIBUTION OF SOIL MOISTURE IS CALCULATED FOR THE PERIOD.
C
C=============================================================================
      EXTERNAL FXBAR
      COMPLEX PHI,ZZ
      LOGICAL*1 RAIN,RAINB4
      COMMON /BIG/  STACK(6,10), SUM(8), Q(8),
     * ALBEDO,     C1, C1PART,     C2,     C3,  CAPEX, CAPEX1,
     * CAPEX2,  CAPIN,    CDN,  CDUOR,     CL,     CP, CPNOLD,   DEDT,
     *   DELT,     DF,   DFIL, DRADLU,  DSENS,      E,     EO,    ELO,
     *   EMIS,     EP,     FE,    FEP,     FI,      G,   SENS,      P,
     *      R,  RADLD, RADLUO,   RADS,  RCCDU,  RHOCP,   RHOL,  RHOVA,
     *   RLCL,    RLE,     RN,     RS,  ROOTO,  ROOT1,  ROOT2,  SENSO,
     *    SIG,     T1, T1LAST,  T1OLD,     T2,     TA,    TCE,    TCI,
     *   TIME,  TSTAB,   XBAR,  XINIT,     XR,     XZ,     ZA,
     *   IFOR,  IRAIN,   ISTK,   IUNF, MAXSTK,   NPER, NSTACK,
     *   RAIN, RAINB4
      COMMON /BLOCK1/ AA,BB,CC,DD,EE,PRN,XU
      COMMON /FXCOM/ TRAIN
C=============================================================================
C
C     READ PERIOD DURATION AND ATMOSPHERIC FORCING. PROGRAM EXECUTION
C     TERMINATES WHEN A NEGATIVE DURATION IS SPECIFIED.
C
C=============================================================================
      READ(15,1010) DELT,P,RHOVA,UA,RADS,RADLD,TA
      IF (DELT.GT.O.) GO TO 20
C=============================================================================
C
```

```
C      END OF SIMULATION. CALCULATE AND PRINT AVERAGE FLUXES.
C      MARK THE END OF THE UNFORMATTED FILE.
C
C==========================================================================
      IEOF=-1
      IF(IUNF.GT.O) WRITE(IUNF) IEOF
      DO 10 I=1,8
   10 SUM(I)=SUM(I)/NPER
      IF(IFOR.NE.O) WRITE(IABS(IFOR),2010) SUM
      IF(IFOR.EQ.O) WRITE(6,2010) SUM
      STOP
C==========================================================================
C
C      CALCULATE FUNCTIONS OF THE FORCING.
C
C==========================================================================
   20 NPER=NPER+1
      TIME=TIME+DELT
      RAINB4=RAIN
      RAIN=.FALSE.
      IF(P.GT.O.) RAIN=.TRUE.
      IF(NPER.EQ.1) RAINB4=.NOT.RAIN
      RIPART=981.*ZA/((TA+273.16)*UA*UA)
      DRI=-RIPART
      RI=RIPART*(TA-T1)
      ZZ=PHI(RI)
      STAB=REAL(ZZ)
      DSTAB=AIMAG(ZZ)
      TSTAB=O.
      IF(STAB.NE.O.) TSTAB=DSTAB*DRI/STAB
      CDU=STAB*CDN*UA
      RCCDU=RHOCP*CDU
      CDUOR=CDU/RHOL
      T1LAST=T1
C==========================================================================
C
C    SUBSEQUENT CALCULATIONS DEPEND ON WHETHER OR NOT IT'S RAINING.
C
C==========================================================================
      IF(.NOT.RAIN) GO TO 40
C==========================================================================
C
C    IT IS RAINING. DID IT RAIN LAST PERIOD?
C
C==========================================================================
      IF(RAINB4) GO TO 30
C==========================================================================
C
C      THE RAIN JUST BEGAN. RESET TCI AND FI, AND RECORD TSTART.
C
C==========================================================================
      TSTART=TIME-DELT
      TCI=O.
      FI=O.
C==========================================================================
C
C    EVERY RAINY PERIOD, CALCULATE THE AVERAGE INFILTRATION CAPACITY OVER
C    THE COMING PERIOD AND SET THE REDISTRIBUTION TIME.
C
C==========================================================================
```

```
   30  CAPIN=(FC(TCI+DELT,1)-FI)/DELT
       TRED=DELT
       GO TO 90
C==========================================================================
C
C    IT IS NOT RAINING. DID IT RAIN LAST PERIOD?
C
C==========================================================================
C
   40  IF(.NOT.RAINB4) GO TO 80
C
C==========================================================================
C
C    THE RAIN JUST ENDED. STORE CURRENT EXFILTRATION DATA IN THE STACK,
C    RECORD TRAIN AND V, AND CALCULATE XBAR. RESET XR, TCE, FEP, FE,
C    AND FI. DO NOT PERFORM STORAGE IF SIMULATION JUST BEGAN (NPER=1).
C    IF FI IS ZERO, INDICATING THAT ALL OF THE PRECIPITATION OF THE LAST
C    STORM EVAPORATED BEFORE THE END OF THE STORM, THEN DO NOT PERFORM
C    STORAGE OR RESET, BUT RATHER KEEP THE OLD VALUES.
C
C==========================================================================
       IF(NPER.EQ.1) GO TO 80
       IF(FI.EQ.O.) GO TO 80
       NSTACK=MINO(NSTACK+1,MAXSTK)
       IF(ISTK.NE.1.OR.IFOR.EQ.O) GO TO 50
       DAYS=(TIME-DELT)/DF
       WRITE(IABS(IFOR),2020) DAYS,NSTACK
   50  IF(NSTACK.EQ.1) GO TO 70
       DO 60 I=2,NSTACK
       J=NSTACK+2-I
       DO 60 K=1,6
   60  STACK(K,J)=STACK(K,J-1)
   70  STACK(1,1)=FE
       STACK(2,1)=XBAR
       STACK(3,1)=XZ
       STACK(4,1)=FEP
       STACK(5,1)=XR
       STACK(6,1)=TIME-DELT
       XZ=FI
       TRAIN=TIME-DELT-TSTART
       XBAR=FINV(FXBAR,XZ,O.OO1,XU*O.999,1.E-3,2,30)
       XR=XBAR
       TCE=O.
       FE=O.
       FEP=O.
       FI=O.
C==========================================================================
C
C      EVERY NON-RAINY PERIOD, SET THE REDISTRIBUTION TIME.
C
C==========================================================================
   80  TRED=DELT
       IF(RAINB4) TRED=O.5*DELT
C==========================================================================
C
C    EACH PERIOD, CALULATE SOIL MOISTURE REDISTRIBUTION AND FIND
C    THE AVERAGE EXFILTRATION CAPACITY OF THE SOIL OVER THE PERIOD.
C
C==========================================================================
   90  CALL REDIST(TRED,O)
```

```
      CAPEX1=(FC(TCE+DELT,O)-FEP)/DELT
      CAPEX2=(XZ-FE)/DELT
      CAPEX=AMIN1(CAPEX1,CAPEX2)
      IF(RAIN.EQ.RAINB4.OR.NPER.EQ.1.OR.IFOR.EQ.O.OR.IRAIN.EQ.O) RETURN
      DAYS=(TIME-DELT)/DF
      IF(RAIN) WRITE(IABS(IFOR),2030) DAYS
      IF(.NOT.RAIN) WRITE(IABS(IFOR),2040) DAYS,XZ,XBAR
      RETURN
C=============================================================================
1010  FORMAT(7F10.0)
2010  FORMAT(///5X,'SIMULATION COMPLETED. AVERAGE FLUXES:'//10X,
     * 'RN',5X,F6.1,' LANGLEYS/DAY'/10X,'G',6X,F6.1,'     "'/10X,
     * 'SENS',3X,F6.1,'     "'/10X,'RLE',4X,F6.1,'     "'//10X,
     * 'P',1X,F11.3,' CENTIMETERS/DAY'/10X,'EP',F11.3,'        "'/10X,
     * 'E ',F11.3,'        "'/10X,'RS',F11.3,'        "')
2020  FORMAT(/ 'AT DAYS =',F6.2,', ADDING PREVIOUS DATA TO STACK. NOW NS
     *TACK =',I3)
2030  FORMAT(/'AT DAYS =',F6.2,', RAIN STARTS'/)
2040  FORMAT(/'AT DAYS =',F6.2,', RAIN ENDS. FI=',F6.2,
     * '  AND RESULTANT XBAR=',F7.4/)
      END
C/////////////////////////////////////////////////////////////////////////////
C/////////////////////////////////////////////////////////////////////////////
      SUBROUTINE PERIOD
C/////////////////////////////////////////////////////////////////////////////
C/////////////////////////////////////////////////////////////////////////////
C
C   PERIOD SETS UP AND SOLVES THE FORCE-RESTORE EQUATION, TAKING INTO ACCOUNT
C   THE POSSIBLE SOIL CONTROL OF MOISTURE FLUX. THE EVAPORATION RATE IS
C   FOUND SIMULTANEOUSLY.
C
C=============================================================================
      LOGICAL*1 RAIN,RAINB4
      COMMON /BIG/  STACK(6,10), SUM(8), Q(8),
     * ALBEDO,     C1, C1PART,      C2,      C3,  CAPEX, CAPEX1,
     * CAPEX2,  CAPIN,     CDN,  CDUOR.      CL,     CP, CPNOLD,    DEDT,
     *   DELT,     DF,    DFIL, DRADLU,  DSENS,      E,      EO,     ELO,
     *   EMIS,     EP,      FE,    FEP,      FI,      G,   SENS,       P,
     *      R, RADLD,  RADLUO,    RADS,  RCCDU,  RHOCP,    RHOL,  RHOVA,
     *   RLCL,    RLE,      RN,      RS,  ROOTO,  ROOT1,   ROOT2,  SENSO,
     *    SIG,     T1, T1LAST,  T1OLD,      T2,     TA,     TCE,     TCI,
     *   TIME,  TSTAB,    XBAR,  XINIT,      XR,     XZ,      ZA,
     *   IFOR,  IRAIN,    ISTK,   IUNF, MAXSTK,   NPER, NSTACK,
     *   RAIN, RAINB4
      COMMON /BLOCK5/ ALDRY,ALWET,DAL
      COMMON /BLOCK6/ EMDRY,DEM
      COMMON /PARAM/ SE,SO,AO
      COMPLEX RHOZ,ZZ
C=============================================================================
C
C   PREPARE TO SOLVE THE FORCE-RESTORE EQUATION FOR SURFACE TEMPERATURE.
C   ASSUME THAT EVAPORATION OCCURS AT THE POTENTIAL RATE FOR THE FIRST
C   TRY. CALCULATE C1, ALBEDO, AND EMISSIVITY ASSUMING THAT THE
C   SURFACE IS WET, WITH MOISTURE CONTENT EQUAL TO DEPTH-AVERAGED VALUE.
C
C=============================================================================
      IRPT=0
      CPNOLD=0.
      DELT1=0.
      ROOT1=SQRT(EHCAP(XBAR)*ETCON(XBAR))
```

```
      C1=C1PART/(0.7*ROOT2+0.3*ROOT1)
      ZZ=RHOZ(T1)
      RH=REAL(ZZ)
      DR=AIMAG(ZZ)
      EO=CDUOR*(RH-RHOVA)
      DEDT=CDUOR*(DR+TSTAB*(RH-RHOVA))
      ALBEDO=ALB(XBAR)
      EMIS=EMI(XBAR)
      GO TO 20
C========================================================================
C
C   IF CONTROL IS TRANSFERRED HERE, IT IS IN ORDER TO REPEAT THE FORCE-
C   RESTORE SOLUTION WITH A SPECIFIED VALUE OF EVAPORATION LESS THAN THE
C   POTENTIAL EVAPORATION RATE. THE SURFACE IS ASSUMED TO BE DRY IN
C   ORDER TO FIND C1, ALBEDO, AND EMISSIVITY.
C
C========================================================================
   10 C1=C1PART/(0.7*ROOT2+0.3*ROOT0)
      EO=CAPEX+P+FI/DELT
      DEDT=0.
      EMIS=EMDRY
      ALBEDO=ALDRY
      IRPT=1
C========================================================================
C
C   PERFORM THE FORCE-RESTORE INTEGRATION.
C
C========================================================================
   20 CALL FOREST
C========================================================================
C
C   GIVEN THE SURFACE TEMPERATURE, CALCULATE THE EVAPORATION RATE.
C   IS THIS VALUE ACCEPTABLE? IF IT DOES NOT EXCEED THE AVAILABLE AMOUNT
C   OF WATER, IT IS ACCEPTABLE, AND CONTROL RETURNS TO THE MAIN PROGRAM.
C
C========================================================================
      E=EO+DEDT*(T1-T1LAST)
      IF(IRPT.EQ.0) EP=E
   30 IF(E.LE.CAPEX+P+FI/DELT) RETURN
C========================================================================
C
C   THE POTENTIAL EVAPORATION RATE EXCEEDS THE AVAILABLE WATER SUPPLY.
C   IF THIS IS DUE TO A DIFFUSION LIMITATION, OR IF THERE IS NO FURTHER
C   MASS IN STACK, REPEAT THE SOLUTION WITH A DRY SURFACE AND SOIL CONTROL
C   OF EVAPORATION.
C
C========================================================================
      IF(CAPEX1.LE.CAPEX2.OR.NSTACK.EQ.0) GO TO 10
C========================================================================
C
C   THERE IS A LIMITATION ON EVAPORATION DUE TO THE AMOUNT OF WATER IN
C   THE CURRENTLY ACTIVE MOISTURE MASS. DETERMINE HOW FAR INTO THE
C   PERIOD (DELT1) THIS RESTRICTION IS FELT.
C
C========================================================================
      FCAP=(CAPEX1-CPNOLD)*DELT/(DELT-DELT1)
      DELT1=DELT1+(CAPEX2-CPNOLD)*DELT/AMIN1(FCAP,
     * E-P-FI/DELT)
C========================================================================
C
```

```
C   REMOVE EXFILTRATION DATA FROM TOP OF STACK. CALCULATE REDISTRIBUTION TIME.
C
C===============================================================================
      FE=STACK(1,1)
      XBAR=STACK(2,1)
      XZ=STACK(3,1)
      FEP=STACK(4,1)
      XR=STACK(5,1)
      TRED=TIME-STACK(6,1)
      IF(NSTACK.EQ.1) GO TO 50
C===============================================================================
C
C   SHIFT OLDER DATA UP ONE POSITION IN THE STACK.
C
C===============================================================================
      DO 40 I=2,NSTACK
      DO 40 K=1,6
   40 STACK(K,I-1)=STACK(K,I)
   50 NSTACK=NSTACK-1
      DAYS=TIME/DF
      IF(ISTK.EQ.1.AND.IFOR.NE.0) WRITE(IABS(IFOR),2010) DAYS,NSTACK
C===============================================================================
C
C   CALCULATE SE AND TCE. PERFORM REDISTRIBUTION FOR THE TIME THAT THIS
C   SET OF DATA WAS STORED IN STACK.
C
C===============================================================================
      SE=DSORP(XBAR,XR)
      TCE=TC(FEP,0)
      CALL REDIST(TRED,1)
C===============================================================================
C
C   UPDATE THE VALUES OF CAPEX1 AND CAPEX2. GO BACK TO SEE IF MOISTURE
C   SUPPLY IS NOW ADEQUATE.
C
C===============================================================================
      CAPEX1=CAPEX2+(FC(TCE+DELT-DELT1,0)-FEP)/DELT
      CPNOLD=CAPEX2
      CAPEX2=CAPEX2+(XZ-FE)/DELT
      CAPEX=AMIN1(CAPEX1,CAPEX2)
      GO TO 30
C===============================================================================
2010  FORMAT(/'AT DAYS =',F6.2,', EXTRACTING FROM STACK. NSTACK=',I3/)
      END
C///////////////////////////////////////////////////////////////////////////////
C///////////////////////////////////////////////////////////////////////////////
      SUBROUTINE END
C///////////////////////////////////////////////////////////////////////////////
C///////////////////////////////////////////////////////////////////////////////
C
C     END PERFORMS FOLLOW-UP CALCULATIONS FOR PERIOD. THE CHANGES IN THE
C     STATE VARIABLES OF INFILTRATION AND EXFILTRATION ARE RECORDED, FLUXES
C     ARE CALCULATED, AND T2 IS UPDATED. THE OUTPUT FOR THE PERIOD IS
C     PROCESSED.
C
C===============================================================================
      LOGICAL*1 RAIN,RAINB4
      COMMON /BIG/  STACK(6,10), SUM(8), Q(8),
     * ALBEDO,     C1, C1PART,     C2,     C3,  CAPEX, CAPEX1,
     * CAPEX2,  CAPIN,     CDN,  CDUOR,     CL,     CP, CPNOLD,   DEDT,
```

```
     *   DELT,      DF,   DFIL, DRADLU,  DSENS,      E,     EO,    ELO,
     *   EMIS,      EP,     FE,    FEP,     FI,      G,   SENS,      P,
     *      R,  RADLD, RADLUO,   RADS,  RCCDU,  RHOCP,   RHOL,  RHOVA,
     *   RLCL,     RLE,     RN,     RS,  ROOTO,  ROOT1,  ROOT2, SENSO,
     *    SIG,      T1, T1LAST,  T1OLD,     T2,     TA,    TCE,    TCI,
     *   TIME,  TSTAB,   XBAR,  XINIT,     XR,     XZ,     ZA,
     *   IFOR,  IRAIN,   ISTK,   IUNF, MAXSTK,   NPER, NSTACK,
     *   RAIN, RAINB4
C=========================================================================
C
C     EVAPORATION AND PRECIPITATION ARE KNOWN. HOW THEY ARE DIVIDED
C     AMONG INFILTRATION AND EXFILTRATION DEPENDS ON WHETHER OR NOT
C     IT WAS RAINING AND WHETHER OR NOT PRECIPITATION RATE EXCEEDED
C     EVAPORATION RATE.
C
C=========================================================================
      RS=0.
C=========================================================================
C
C     DID IT RAIN?
C
C=========================================================================
      IF(RAIN) GO TO 30
C=========================================================================
C
C     IT WAS NOT RAINING. WAS EVAPORATION POSITIVE (NORMAL) OR NEGATIVE
C     (DEWFALL)?
C
C=========================================================================
      IF(P.GT.E) GO TO 10
C=========================================================================
C
C     NORMAL EVAPORATION CONDITIONS HELD. EVAPORATION COMES FROM
C     EXFILTRATION.
C
C=========================================================================
      FILT=0.
      EXFILT=E
      GO TO 20
C=========================================================================
C
C     DEWFALL OCCURRED. TO THE EXTENT POSSIBLE, THIS IS TREATED AS
C     NEGATIVE EXFILTRATION. ANY EXTRA WATER IS ADDED TO THE PREVIOUS
C     RAINFALL MASS.
C
C=========================================================================
   10 EXFILT=-AMIN1(-E,FEP/DELT)
      FILT=-E+EXFILT
      XZ=XZ+FILT*DELT
C=========================================================================
C
C     FOR ALL INTERSTORM PERIODS, EXFILTRATION IS ADDED TO THE ACTUAL AND
C     THE APPARENT CUMULATIVE EXFILTRATION DEPTHS AND THE COMPRESSED TIME
C     IS UPDATED.
C
C=========================================================================
   20 FEP=FEP+(EXFILT-CPNOLD)*DELT
      FE=FE+(EXFILT-CPNOLD)*DELT
      TCE=TC(FEP,0)
      GO TO 60
```

```
C=========================================================================
C
C     THIS WAS A STORM PERIOD. DID PRECIPITATION RATE EXCEED EVAPORATION?
C
C=========================================================================
   30 IF(P.LT.E) GO TO 40
C=========================================================================
C
C     PRECIPITATION EXCEEDED EVAPORATION. THE PORTION OF THE EXCESS THAT
C     IS NOT GREATER THAN THE INFILTRATION CAPACITY IS INFILTRATION. THE
C     REMAINDER IS SURFACE RUNOFF.
C
C=========================================================================
      FILT=AMIN1(P-E,CAPIN)
      RS=P-E-FILT
      GO TO 50
C=========================================================================
C
C     EVAPORATION EXCEEDED PRECIPITATION RATE. TO THE EXTENT POSSIBLE,
C     EXTRA EVAPORATION COMES OUT AS NEGATIVE INFILTRATION. ANY OTHER
C     COMES AS EXFILTRATION IN A CONTINUATION OF THE PREVIOUS INTER-
C     STORM EVENT.
C
C=========================================================================
   40 FILT=-AMIN1(E-P,FI/DELT)
      EXFILT=FILT+E-P
      FEP=FEP+(EXFILT-CPNOLD)*DELT
      FE=FE+(EXFILT-CPNOLD)*DELT
      TCE=TC(FEP,0)
C=========================================================================
C
C     FOR ALL STORM PERIODS, CUMULATIVE INFILTRATION AND COMPRESSED TIME
C     ARE UPDATED.
C
C=========================================================================
   50 FI=FI+FILT*DELT
      IF(FI.LT.1.E-8) FI=0.
      TCI=TC(FI,1)
C=========================================================================
C
C     FLUX UPDATING COMPLETED. FIND ENERGY FLUXES AND WATER FLUXES.
C     STORE THEM IN SUM.
C
C=========================================================================
   60 RN=(1.-ALBEDO)*RADS+EMIS*RADLD-(RADLUO+DRADLU*(T1-T1LAST))
      SENS=SENSO+DSENS*(T1-T1LAST)
      RLE=E*RHOL*(ELO+CP*T1)
      G=RN-SENS-RLE+(P*TA-RS*T1)*RLCL
      DFIL=FILT-EXFILT
      Q(1)=DF*RN
      Q(2)=DF*G
      Q(3)=DF*(SENS-RLCL*(P*TA-RS*T1))
      Q(4)=DF*RLE
      Q(5)=P*DF
      Q(6)=EP*DF
      Q(7)=E*DF
      Q(8)=RS*DF
      DO 70 I=1,8
   70 SUM(I)=SUM(I)+Q(I)
C=========================================================================
```

```
C
C   CALCULATE DEPTH-AVERAGED TEMPERATURE.
C
C==============================================================================
      T2=T2+DELT*C3*(RN-SENS-RLE+RLCL*P*(TA-T1))
C==============================================================================
C
C      PERFORM THE OUTPUT FOR THE PERIOD.
C
C==============================================================================
      IF(IFOR.NE.O.OR.IUNF.NE.O) CALL OUTPUT
      RETURN
      END
C//////////////////////////////////////////////////////////////////////////////
C//////////////////////////////////////////////////////////////////////////////
      SUBROUTINE SOILI
C//////////////////////////////////////////////////////////////////////////////
C//////////////////////////////////////////////////////////////////////////////
C
C      THIS SUBROUTINE READS THE SOIL PARAMETERS FROM THE INPUT FILE.
C      IT ALSO COMPUTES DERIVED SOIL PARAMETERS THAT WILL BE USED BY
C      SOME OF THE FUNCTION SUBPROGRAMS.
C
C==============================================================================
      COMMON /BLOCK1/ AA,BB,CC,DD,EE,PRN,XU
      COMMON /BLOCK2/ CKSAT,ENT(25),PFR(25),PR(25),SS(25),
     *  XI(25),XIM,NSS
      COMMON /BLOCK3/ HCD,HCAP(5)
      COMMON /BLOCK4/ CONO,GA1,GA2,GB1,GB2,POR,PS1,PS2,TCON(5),
     * VPER(5),XK,XWILT,ZETAO,SF(5)
      COMMON /BLOCK5/ ALDRY,ALWET,DAL
      COMMON /BLOCK6/ EMDRY,DEM
C==============================================================================
C
C      READ SOIL PARAMETERS.
C
C==============================================================================
      READ(15,1010) POR,XU,CKSAT,AA,BB,CC,DD,
     * EE,XK,(VPER(K),K=3,5),ALDRY,ALWET,EMDRY,EMWET
C==============================================================================
C
C      FIND ALBEDO AND EMISSIVITY CONSTANTS FOR LATER USE BY ALB AND EMI.
C
C==============================================================================
      DAL=2.*(ALWET-ALDRY)/POR
      DEM=(EMWET-EMDRY)/XU
C==============================================================================
C
C      FIND MOISTURE RETENTION CONSTANTS FOR LATER USE BY XWET. THE INPUT
C      VALUE OF EE IS REPLACED BY A COMPUTED ONE THAT ENSURES XWET GOES
C      TO ZERO WHEN PF GOES TO PFRM=7, ITS MAXIMUM VALUE. THE VALUE OF PF
C      SEPARATING SATURATION AND DESATURATION, PFRN, IS NEXT FOUND ITERATIVELY.
C      ITS LOGARITHM IS CALLED PRN.
C
C==============================================================================
      PFRM=7.
      EE=-AA*(-1.E+7/BB)**CC
      PFRN=O.
   10 PFOLD=PFRN
      PFRN=ALOG10(-BB*((XU-DD*(7.-PFOLD)-EE)/AA)**(1./CC))
```

```
      IF(ABS(PFOLD-PFRN).GE.1.E-6)  GO TO 10
      PFRN=PFRN+1.E-6
      PRN=-10.**PFRN
      XIM=XWET(-1.E+7)
C==========================================================================
C
C     FIT THE FIRST DRYING CURVE TO A PIECEWISE LINEAR RELATION BETWEEEN
C     MOISTURE CONTENT AND PF. THIS APPROXIMATION IS USED ONLY IN THE
C     NUMERICAL INTEGRATION OF THE RELATIVE HYDRAULIC CONDUCTIVITY FUNCTION
C     BY HYCON.
C
C==========================================================================
      NSS=21
      XI(1)=XIM
      PFR(1)=PFRM
      PR(1)=-10.**PFRM
      DP=(PFRM-PFRN)/NSS
      DO 20 K=1,NSS
      PFR(K+1)=PFRM-K*DP
      PR(K+1)=-10.**PFR(K+1)
      XI(K+1)=XWET(PR(K+1))
   20 SS(K)=(PFR(K)-PFR(K+1))/(XI(K+1)-XI(K))
C==========================================================================
C
C   DO INITIAL INTEGRATION OF THE RELATIVE HYDRAULIC CONDUCTIVITY
C   FUNCTION.
C
C==========================================================================
      DO 30 K=1,NSS
      ENT(K)=(1./PR(K+1)-1./PR(K))/SS(K)
      IF(K.EQ.1) GO TO 30
      ENT(K)=ENT(K)+ENT(K-1)
   30 CONTINUE
C==========================================================================
C
C     INITIALIZE THERMAL CONSTANTS FOR EHCAP AND ETCON.
C
C==========================================================================
      XWILT=XWET(-1.5E+4)
      GA1=.035
      GA2=.013
      GB1=.298/POR
      GB2=(GA1-GA2)/XWILT+GB1
C==========================================================================
C
C     CALCULATE THE VALUES OF THERMAL CONDUCTIVITY AND HEAT CAPACITY AT
C     COMPLETE DRYNESS.
C
C==========================================================================
      S1=POR*TCON(2)
      S2=POR
      HCD=0.
      DO 40 I=3,5
      FAC=TCON(I)/TCON(2)-1.
      GRAT=0.667/(1.+FAC*SF(I))+0.333/(1.+FAC*(1.-2.*SF(I)))
      S1=S1+GRAT*TCON(I)*VPER(I)
      S2=S2+GRAT*VPER(I)
   40 HCD=HCD+HCAP(I)*VPER(I)
      CONO=1.25*S1/S2
C==========================================================================
```

```
C
C     INITIALIZE THE PARTIAL SUMS THAT WILL BE USED LATER BY ETCON.
C
C=============================================================================
      PS1=0.
      PS2=0.
      DO 50 I=3,5
      FAC=TCON(I)/TCON(1)-1.
      GRAT=0.667/(1.+FAC*SF(I))+0.333/(1.+FAC*(1.-2.*SF(I)))
      TERM=GRAT*VPER(I)
      PS1=PS1+TERM*TCON(I)
   50 PS2=PS2+TERM
      RETURN
C=============================================================================
 1010 FORMAT(8F10.0)
      END
C/////////////////////////////////////////////////////////////////////////////
C/////////////////////////////////////////////////////////////////////////////
      SUBROUTINE SORPI(ISOR)
C/////////////////////////////////////////////////////////////////////////////
C/////////////////////////////////////////////////////////////////////////////
C
C     SORPI CALCULATES SORPTIVITY AND DESORPTIVITY ARRAYS. THE DESORPTIVITY
C     IS FOUND AS A FUNCTION OF (1) THE MOISTURE CONTENT AT WHICH THE
C     REVERSAL TO DRYING OCCURRED AND (2) THE RATIO OF THE 'INITIAL'
C     MOISTURE CONTENT FOR DESORPTIVITY (XBAR) TO THIS VALUE.
C     THE SORPTIVITY IS FOUND AS A FUNCTION OF THE IMPOSED MOISTURE
C     CONTENT AT THE SURFACE OF A DRY SOIL.
C
C=============================================================================
      COMMON /SORCOM/ XARRAY(21),YARRAY(21),SIARAY(21),
     *  SEARAY(21,21),SICO(20,3),SEE(21),SE1(21),SEECO(20,3),
     *  XB,XWR,ICO,NX
      COMMON /BLOCK1/ AA,BB,CC,DD,EE,PRN,XU
      COMMON /BLOCK2/ CKSAT,ENT(25),PFR(25),PR(25),SS(25),
     *  XI(25),XIM,NSS
      EXTERNAL DI
      EXTERNAL DE
      EXTERNAL XWET
      COMMON /PARAM/ SE,SO,AO
C=============================================================================
C
C     SET UP XARRAY AND YARRAY.
C
C=============================================================================
      NX=21
      ICO=20
      DX=0.9999*(XU-0.001)/(NX-1.)
      AERR=0.
      RERR=1.E-4
      DO 10 I=1,NX
      XARRAY(I)=(I-1.)*DX+0.0009999
   10 YARRAY(I)=0.001 + 0.999*(FLOAT(I-1)/FLOAT(NX-1))
C=============================================================================
C
C     IF SORPTIVITIES AND DESORPTIVITIES ARE ALREADY STORED IN A FILE,
C     THEN THEIR COMPUTATION IS SKIPPED.
C
C=============================================================================
      IF(ISOR.LT.0) GO TO 40
```

```
C==========================================================================
C
C   CALCULATE THE DESORPTIVITY ASSOCIATED WITH AN INITIAL MOISTURE CONTENT
C   OF XB AND A BOUNDARY CONDITION OF DRYNESS. RELATION BETWEEN MOISTURE
C   CONTENT AND MATRIC POTENTIAL GIVEN ASSUMING A DRYING REVERSAL AFTER
C   WETTING TO XARRAY(I) FROM DRYNESS. NOTE XARRAY(I) GREATER THAN OR
C   EQUAL TO XB.
C
C==========================================================================
      WRITE(6,2010)
      P1=-1.E7-1.
      DO 20 I=1,NX
      XWR=XARRAY(I)
      ARG=0.5*(XU+XWR)
      PMIN=-1.E7
      DO 20 J=1,NX
      XB=(XARRAY(I)-XARRAY(1))*YARRAY(J)+XARRAY(1)
      XW=ARG-SQRT(ARG*ARG-XU*XB)
      P2=FINV(XWET,XW,PMIN,0.,1.E-6,1,30)
      VALUE=DCADRE(DE,P1,P2,AERR,RERR,ERROR,IER)
      IF(IER.LT.100) GO TO 20
      IF(IER.EQ.131) WRITE(6,2040) XWR,XB
      IF(IER.EQ.132) WRITE(6,2050) XWR,XB
      IF(IER.EQ.133) WRITE(6,2060) XWR,XB,RERR
   20 SEARAY(I,J)=SQRT(2.36*VALUE*XB**0.15)
C==========================================================================
C
C   CALCULATE THE SORPTIVITY ASSOCIATED WITH WETTING FROM DRYNESS TO XARRAY(I).
C
C==========================================================================
      WRITE(6,2020)
      P1=-1.E7
      X1=0.
      PMIN=-1.E7
      DO 30 I=1,NX
      P2=FINV(XWET,XARRAY(I),PMIN,0.,1.E-6,1,30)
      PMIN=P2
      X2=XARRAY(I)
      XB=X1
      VALUE=DCADRE(DI,P1,P2,AERR,RERR,ERROR,IER)
      IF(IER.LT.100) GO TO 30
      IF(IER.EQ.131) WRITE(6,2070) XB
      IF(IER.EQ.132) WRITE(6,2080) XB
      IF(IER.EQ.133) WRITE(6,2090) XB,RERR
   30 SIARAY(I)=SQRT(2.12*VALUE*(X2-X1)**0.3333)
      WRITE(6,2030)
C==========================================================================
C
C     WRITE THE CALCULATED ARRAYS ONTO A FILE.
C
C==========================================================================
      WRITE(ISOR) ((SEARAY(I,J),I=1,NX),J=1,NX),(SIARAY(I),I=1,NX)
      GO TO 50
C==========================================================================
C
C     THE ARRAYS HAVE ALREADY BEEN SET UP. READ THEM FROM A FILE.
C
C==========================================================================
   40 READ(-ISOR) ((SEARAY(I,J),I=1,NX),J=1,NX),(SIARAY(I),I=1,NX)
C==========================================================================
```

```
C
C     CALCULATE CUBIC SPLINE COEFFICIENTS FOR INTERPOLATION OF
C     DESORPTIVITY AND SORPTIVITY. SET THE SORPTIVITY USED TO FIND
C     THE INFILTRATION CAPACITY.
C
C=========================================================================
   50 CALL ICSCCU(XARRAY,SIARAY,NX,SICO,ICO,IER)
      IF(IER.EQ.0) GO TO 60
      IF(IER.EQ.129) WRITE(6,2100) NX,ICO
      IF(IER.EQ.130) WRITE(6,2110) NX
      IF(IER.EQ.131) WRITE(6,2120)
      STOP
   60 SO=SIARAY(NX)
      AO=0.5*CKSAT
      RETURN
C=========================================================================
 2010 FORMAT(1X,'BEGINNING DESORPTIVITY EVALUATION.')
 2020 FORMAT(1X,'FINISHED. BEGINNING SORPTIVITY EVALUATION.')
 2030 FORMAT(1X,'FINISHED.')
 2040 FORMAT(/1X,'**** ERROR. IER=131 USING DCADRE TO EVALUATE DESORPTIV
     *ITY IN SUBROUTINE SORPI.'/1X,'XWR=',F10.6,' AND XB=',F10.6/
     * 1X,'THIS IS INDICATIVE OF INSUFFICIENT INTERNAL WORKING STORAGE.'
     *)
 2050 FORMAT(/1X,'**** ERROR. IER=132 USING DCADRE TO EVALUATE DESORPTIV
     *ITY IN SUBROUTINE SORPI.'/1X,'XWR=',F10.6,' AND XB=',F10.6/
     * 1X,'THIS IS INDICATIVE OF FAILURE DUE TO EXCESSIVE NOISE IN THE'/
     * 1X,'FUNCTION RELATIVE TO AERR OR RERR OR DUE TO AN ILL-BEHAVED'/
     * 1X,'INTEGRAND.')
 2060 FORMAT(/1X,'**** ERROR. IER=133 USING DCADRE TO EVALUATE DESORPTIV
     *ITY IN SUBROUTINE SORPI.'/1X,'XWR=',F10.6,' AND XB=',F10.6/
     * 1X,'RERR=',E10.2,' IS OUTSIDE THE ALLOWABLE RANGE (0.,0.1).')
 2070 FORMAT(/1X,'**** ERROR. IER=131 USING DCADRE TO EVALUATE SORPTIVIT
     *Y IN SUBROUTINE SORPI.'/1X,'XB=',F10.6/
     * 1X,'THIS IS INDICATIVE OF INSUFFICIENT INTERNAL WORKING STORAGE.'
     *)
 2080 FORMAT(/1X,'**** ERROR. IER=132 USING DCADRE TO EVALUATE SORPTIVIT
     *Y IN SUBROUTINE SORPI.'/1X,'XB=',F10.6/
     * 1X,'THIS IS INDICATIVE OF FAILURE DUE TO EXCESSIVE NOISE IN THE'/
     * 1X,'FUNCTION RELATIVE TO AERR OR RERR OR DUE TO AN ILL-BEHAVED'/
     * 1X,'INTEGRAND.')
 2090 FORMAT(/1X,'**** ERROR. IER=133 USING DCADRE TO EVALUATE SORPTIVIT
     *Y IN SUBROUTINE SORPI.'/1X,'XB=',F10.6/
     * 1X,'RERR=',E10.2,' IS OUTSIDE THE ALLOWABLE RANGE (0.,0.1).')
 2100 FORMAT(/1X,'**** ERROR. IER=129 USING ICSCCU IN SORPI.'/
     * 1X,'NX=',I3,' CANNOT EXCEED ICO=',I3,' BY MORE THAN ONE.')
 2110 FORMAT(/1X,'**** ERROR. IER=130 USING ICSCCU IN SORPI.'/
     * 1X,'NX=',I3,' MUST BE AT LEAST TWO.')
 2120 FORMAT(/1X,'**** ERROR. IER=131 USING ICSCCU IN SORPI.'/
     * 1X,'VECTOR XARRAY IS NOT ORDERED.')
      END
C/////////////////////////////////////////////////////////////////////////
C/////////////////////////////////////////////////////////////////////////
      SUBROUTINE PHII(CDN)
C/////////////////////////////////////////////////////////////////////////
C/////////////////////////////////////////////////////////////////////////
C
C     THIS SUBROUTINE SETS UP A TABLE (A PAIR OF VECTORS) GIVING THE
C     STABILITY RATIO FOR HEAT AND MOISTURE TRANSFER IN THE ATMOSPHERIC
C     BOUNDARY LAYER AS A FUNCTION OF THE BULK RICHARDSON NUMBER,
C     WHEN CONDITIONS ARE UNSTABLE. THE VECTORS, STORED IN COMMON BLOCK
```

```
C     STBLTY, ARE USED FOR INTERPOLATION BY THE FUNCTION PHI.
C     (BASED ON SOFTWARE WRITTEN BY D. C. CURTIS, NATIONAL WEATHER SERVICE)
C
C==========================================================================
      COMMON /STBLTY/ RIB(21),CHRAT(21)
C==========================================================================
C
C     LOOP THROUGH SEVERAL VALUES OF ZA/L, WHERE L IS THE MONIN-OBUKHOV
C     LENGTH. FOR EACH ONE, FIND THE BULK RICHARDSON NUMBER AND THE STABILITY
C     RATIO.
C
C==========================================================================
      VK=0.4
      DO 10 I=1,16
C==========================================================================
C
C     VALUES OF ZA/L ARE CHOSEN AS MEMBERS OF A GEOMETRIC PROGRESSION.
C
C==========================================================================
      ZL=-.0001*1.95**(I-1)
C==========================================================================
C
C     FIND THE STABILITY RATIO FOR MOMENTUM, AND THE ACTUAL DRAG
C     COEFFICIENT, GIVEN CDN AND ZL.
C
C==========================================================================
      X = (1.00 - 16.0*ZL)**0.25
      CDRAT = (1.00 - (CDN**0.50/VK)*
     1(ALOG((1 + X*X)/2.00) + 2.00*ALOG((1+X)/2.00)
     2-2.00*ATAN(X) + 3.14159/2.00))**(-2.00)
      CD = CDRAT*CDN
C==========================================================================
C
C     FIND THE STABILITY RATIO FOR HEAT AND VAPOR, GIVEN CDRAT, CDN,
C     AND X(ZL).
C
C==========================================================================
      CHRAT(I) = CDRAT**0.5*(1.00/(1.00 - 2.00/VK*
     1CDN**0.5*ALOG((1.00 + X*X)/2.00)))
C==========================================================================
C
C     FIND THE TRANSFER COEFFICIENT FOR HEAT AND VAPOR.
C
C==========================================================================
      CH = CHRAT(I)*CDN
C==========================================================================
C
C     GIVEN ZL AND THE TRANSFER COEFFICIENTS, FIND THE CORRESPONDING
C     BULK RICHARDSON NUMBER.
C
C==========================================================================
   10 RIB(I) = CDRAT**1.5*CDN**1.5*ZL/(VK*CH)
      RETURN
      END
C//////////////////////////////////////////////////////////////////////////
C//////////////////////////////////////////////////////////////////////////
      SUBROUTINE FOREST
C//////////////////////////////////////////////////////////////////////////
C//////////////////////////////////////////////////////////////////////////
C
```

```
C   THIS SUBROUTINE PERFORMS THE INTEGRATION OF THE FORCE RESTORE EQUATION
C   FOR GROUND SURFACE TEMPERATURE. A SINGLE NEWTON-RAPHSON STEP IS PERFORMED
C   ON THE EQUATION. T1LAST IS THE VALUE OF T1 AT THE START OF THE CURRENT
C   SIMULATION PERIOD.
C
C===============================================================================
      LOGICAL*1 RAIN,RAINB4
      COMMON /BIG/  STACK(6,10), SUM(8), Q(8),
     * ALBEDO,     C1, C1PART,     C2,      C3,  CAPEX, CAPEX1,
     * CAPEX2,  CAPIN,    CDN,  CDUOR,      CL,     CP, CPNOLD,    DEDT,
     *   DELT,     DF,   DFIL, DRADLU,  DSENS,      E,     EO,     ELO,
     *   EMIS,     EP,     FE,    FEP,      FI,      G,   SENS,       P,
     *      R,  RADLD, RADLUO,   RADS,  RCCDU,  RHOCP,   RHOL,   RHOVA,
     *   RLCL,    RLE,     RN,     RS,  ROOTO,  ROOT1,  ROOT2,   SENSO,
     *    SIG,     T1, T1LAST,  T1OLD,      T2,     TA,    TCE,     TCI,
     *   TIME,  TSTAB,   XBAR,  XINIT,      XR,     XZ,     ZA,
     *   IFOR,  IRAIN,   ISTK,   IUNF, MAXSTK,   NPER, NSTACK,
     *   RAIN, RAINB4
C===============================================================================
      T1=T1LAST
      RLL=RHOL*(ELO+(CP-CL)*T1)
      SENSO=RCCDU*(T1-TA)
      DSENS=RCCDU*(1.+TSTAB*(T1-TA))
      TK=T1+273.16
      RADLUO=EMIS*SIG*TK*TK*TK*TK
      DRADLU=4.*RADLUO/TK
C===============================================================================
C
C     CALCULATE FORCE-RESTORE EQUATION RESIDUAL AND ITS DERIVATIVE.
C
C===============================================================================
      F= C2*(T1-T2)-C1*((1.-ALBEDO)*RADS+
     * EMIS*RADLD-RADLUO-SENSO-RLL*EO+RLCL*P*(TA-T1))
      DFDT=1./DELT+C2+C1*(DRADLU+DSENS+RLL*DEDT+RHOL*(CP-CL)*EO
     * +RLCL*P)
C===============================================================================
C
C     MAKE NEWTON-RAPHSON CORRECTION.
C
C===============================================================================
      T1=T1-F/DFDT
      RETURN
      END
C///////////////////////////////////////////////////////////////////////////////
C///////////////////////////////////////////////////////////////////////////////
      SUBROUTINE REDIST(TRED,KOD)
C///////////////////////////////////////////////////////////////////////////////
C///////////////////////////////////////////////////////////////////////////////
C
C     PERFORM THE REDISTRIBUTION OF SOIL MOISTURE. TRED IS THE DURATION OF
C     THE REDISTRIBUTION INTERVAL. KOD IS EQUAL TO UNITY IF REDISTRIBUTION
C     IS BEING PERFORMED ON NEWLY-EXTRACTED DATA, AND IS ZERO OTHERWISE.
C
C===============================================================================
      EXTERNAL HYCON
      LOGICAL*1 RAIN,RAINB4
      COMMON /BIG/  STACK(6,10), SUM(8), Q(8),
     * ALBEDO,     C1, C1PART,     C2,      C3,  CAPEX, CAPEX1,
     * CAPEX2,  CAPIN,    CDN,  CDUOR,      CL,     CP, CPNOLD,    DEDT,
     *   DELT,     DF,   DFIL, DRADLU,  DSENS,      E,     EO,     ELO,
```

```
     *   EMIS,      EP,      FE,     FEP,      FI,       G,   SENS,       P,
     *      R,  RADLD, RADLUO,    RADS,  RCCDU,  RHOCP,   RHOL,   RHOVA,
     *   RLCL,     RLE,      RN,      RS,  ROOTO,  ROOT1,  ROOT2,  SENSO,
     *    SIG,      T1, T1LAST,   T1OLD,      T2,      TA,    TCE,     TCI,
     *   TIME,   TSTAB,    XBAR,  XINIT,      XR,      XZ,     ZA,
     *   IFOR,   IRAIN,    ISTK,   IUNF, MAXSTK,    NPER, NSTACK,
     *   RAIN, RAINB4
      COMMON /PARAM/ SE,SO,AO
C========================================================================
C
C     DETERMINE THE SLOPE, C, OF THE LOG-K VERSUS LOG-THETA CURVE.
C     INTEGRATE THE REDISTRIBUTION EQUATION ANALYTICALLY TO FIND THE
C     NEW VALUE OF XBAR.
C
C========================================================================
      XA=XBAR
      XB=1.001*XBAR
      CKA=HYCON(XA)
      CKB=HYCON(XB)
      C=ALOG10(CKB/CKA)/ALOG10(XB/XA)
      ZED=1./(1./CKA+C*TRED*R/XZ)
      XBAR=XA*(ZED/CKA)**(1./C)
      XBAR=FINV(HYCON,ZED,XBAR,XBAR,1.E-3,2,30)
C========================================================================
C
C     IF DATA HAS JUST BEEN EXTRACTED (KOD=1), THEN THIS IS A CONTINUATION OF
C     AN EARLIER EXFILTRATION EVENT, SO TCE AND FEP MUST BE UPDATED WHEN
C     A NEW SE IS FOUND. THIS IS ALSO TRUE ANY OTHER TIME, EXCEPT WHEN
C     A NEW INTERSTORM PERIOD HAS JUST BEGUN (THEN TCE=FEP=0), OR WHEN
C     THE ENTIRE SIMULATION HAS JUST BEGUN.
C
C========================================================================
      IF(KOD.EQ.1) GO TO 10
      IF(NPER.EQ.1) GO TO 20
      IF((.NOT.RAIN).AND.RAINB4) GO TO 20
   10 SEOLD=SE
      SE=DSORPA(XBAR,XR)
      FEP=FEP*(SE*SE)/(SEOLD*SEOLD)
      TCE=TC(FEP,0)
      RETURN
C========================================================================
C
C     NOTE THAT THE FIRST CALL TO FIND A DESORPTIVITY WITHIN AN INTERSTORM
C     EVENT IS MADE THROUGH 'DSORP' WHICH SETS UP INTERPOLATION ARRAYS FOR
C     A GIVEN XR. SUBSEQUENT CALLS ARE TO THE ENTRY POINT 'DSORPA',
C     WHICH AVOIDS RE-CALCULATION OF THE ARRAYS.
C
C========================================================================
   20 SE=DSORP(XBAR,XR)
      RETURN
      END
C////////////////////////////////////////////////////////////////////////
C////////////////////////////////////////////////////////////////////////
      SUBROUTINE OUTPUT
C////////////////////////////////////////////////////////////////////////
C////////////////////////////////////////////////////////////////////////
C
C     PERFORMS DESIRED OUTPUT OPERATIONS EACH PERIOD.
C
C========================================================================
```

```
      LOGICAL*1 RAIN,RAINB4
      COMMON /BIG/  STACK(6,10), SUM(8), Q(8),
     * ALBEDO,      C1, C1PART,      C2,       C3,  CAPEX, CAPEX1,
     * CAPEX2,  CAPIN,     CDN,  CDUOR,       CL,      CP, CPNOLD,    DEDT,
     *   DELT,      DF,   DFIL, DRADLU,   DSENS,       E,      EO,     ELO,
     *   EMIS,      EP,     FE,    FEP,      FI,       G,    SENS,       P,
     *      R,  RADLD, RADLUO,    RADS,   RCCDU,  RHOCP,    RHOL,   RHOVA,
     *   RLCL,     RLE,     RN,      RS,   ROOTO,  ROOT1,   ROOT2,   SENSO,
     *    SIG,      T1, T1LAST,  T1OLD,      T2,      TA,     TCE,     TCI,
     *   TIME,  TSTAB,    XBAR,  XINIT,      XR,      XZ,      ZA,
     *   IFOR,  IRAIN,    ISTK,   IUNF,  MAXSTK,    NPER, NSTACK,
     *   RAIN, RAINB4
C=============================================================================
      IF(IFOR.LT.O) WRITE(-IFOR,2010) NPER,TIME,P,EP,E,DFIL,RS,
     * RN,G,SENS,RLE,XBAR,XR,XZ,FEP,FE,FI,TCE,TCI,T1,T2
      IF(IFOR.LE.O) GO TO 10
      DAYS=TIME/DF
      WRITE(IFOR,2020) DAYS,P,E,XBAR,T1
   10 IF(IUNF.EQ.O) RETURN
      WRITE(IUNF) NPER
      WRITE(IUNF) TIME,P,EP,E,DFIL,RS,RN,G,SENS,RLE,XBAR,XR,XZ,FEP,FE,
     * FI,TCE,TCI,T1,T2
      RETURN
C=============================================================================
 2010 FORMAT(/5X,I7,3X,6(E10.2)/5X,7E10.2/5X,7E10.2)
 2020 FORMAT(10X,F8.2,6X,4(E10.2,4X))
      END
C/////////////////////////////////////////////////////////////////////////////
C/////////////////////////////////////////////////////////////////////////////
      FUNCTION ALB(X)
C/////////////////////////////////////////////////////////////////////////////
C/////////////////////////////////////////////////////////////////////////////
C
C     FIND THE ALBEDO AS A FUNCTION OF THE MOISTURE CONTENT.
C
C=============================================================================
      COMMON /BLOCK5/ ALDRY,ALWET,DAL
C=============================================================================
      ALB=AMAX1(ALDRY+DAL*X,ALWET)
      RETURN
      END
C/////////////////////////////////////////////////////////////////////////////
C/////////////////////////////////////////////////////////////////////////////
      FUNCTION DE(PP)
C/////////////////////////////////////////////////////////////////////////////
C/////////////////////////////////////////////////////////////////////////////
C
C     COMPUTE INTEGRAND IN EFFECTIVE EXFILTRATION DIFFUSIVITY.
C
C=============================================================================
      COMMON /VAPCOM/ DPART,GORT
      COMMON /SORCOM/ XARRAY(21),YARRAY(21),SIARAY(21),
     *  SEARAY(21,21),SICO(20,3),SEE(21),SE1(21),SEECO(20,3),
     *  XB,XWR,ICO,NX
      COMMON /BLOCK1/ AA,BB,CC,DD,EE,PRN,XU
      COMMON /BLOCK4/ CONO,GA1,GA2,GB1,GB2,POR,PS1,PS2,TCON(5),
     * VPER(5),XK,XWILT,ZETAO,SF(5)
C=============================================================================
      XW=XWET(PP)
      X=XW*(1.+(XWR-XW)/XU)
```

```
      DX=AMAX1(XB-X,O.)
      IF(DX.EQ.O.) GO TO 10
      DE=(DX)**O.85*(HYCON(X)+EXP(PP*GORT)*DPART*(POR-X)*TORT(POR-X))
      RETURN
   10 DE=O.
      RETURN
      END
C///////////////////////////////////////////////////////////////////////////
C///////////////////////////////////////////////////////////////////////////
      FUNCTION DI(PP)
C///////////////////////////////////////////////////////////////////////////
C///////////////////////////////////////////////////////////////////////////
C
C     COMPUTE INTEGRAND IN EFFECTIVE INFILTRATION DIFFUSIVITY.
C
C===========================================================================
      COMMON /SORCOM/ XARRAY(21),YARRAY(21),SIARAY(21),
     *  SEARAY(21,21),SICO(20,3),SEE(21),SE1(21),SEECO(20,3),
     *  XB,XWR,ICO,NX
C===========================================================================
      X=XWET(PP)
      DI=(ABS(X-XB))**O.6667*HYCON(X)
      RETURN
      END
C///////////////////////////////////////////////////////////////////////////
C///////////////////////////////////////////////////////////////////////////
      FUNCTION DSORP(XBAR,XR)
C///////////////////////////////////////////////////////////////////////////
C///////////////////////////////////////////////////////////////////////////
C
C     WHEN DSORP IS FIRST CALLED, THE VALUE OF SE FOR A FIXED VALUE OF
C     XR AND FOR EACH VALUE OF XBAR=XARRAY(I) IS FOUND BY CUBIC SPLINE
C     INTERPOLATION. THESE VALUES ARE SAVED IN SE1.
C
C===========================================================================
      COMMON /SORCOM/ XARRAY(21),YARRAY(21),SIARAY(21),
     *  SEARAY(21,21),SICO(20,3),SEE(21),SE1(21),SEECO(20,3),
     *  XB,XWR,ICO,NX
C===========================================================================
      DO 30 J=1,NX
      DO 10 I=1,NX
   10 SEE(I)=SEARAY(I,J)
      CALL ICSCCU(XARRAY,SEE,NX,SEECO,ICO,IER)
      IF(IER.EQ.O) GO TO 20
      IF(IER.EQ.129) WRITE(6,2010) NX,ICO
      IF(IER.EQ.130) WRITE(6,2020) NX
      IF(IER.EQ.131) WRITE(6,2030)
      STOP
   20 CALL ICSEVU(XARRAY,SEE,NX,SEECO,ICO,XR,DESORP,1,IER)
      IF(IER.EQ.O) GO TO 30
      IF(IER.EQ.33) WRITE(6,2040) XR,XARRAY(1)
      IF(IER.EQ.34) WRITE(6,2050) XR,XARRAY(NX)
      STOP
   30 SE1(J)=DESORP
C===========================================================================
C
C     THE CUBIC SPLINE COEEFICIENTS FOR YARRAY AND SE1 ARE SET UP FOR
C     SUBSEQUENT CALCULATION OF DSORP BELOW. WHEN NEW VALUES OF THE
C     DESORPTIVITY ARE NEEDED LATER, AND XR HAS NOT YET CHANGED, THIS
C     FUNCTION IS ENTERED AT ENTRY POINT DSORPA, SKIPPING THE CALCULATIONS
```

```
C     ABOVE.
C
C==========================================================================
      CALL ICSCCU(YARRAY,SE1,NX,SEECO,ICO,IER)
      IF(IER.EQ.0) GO TO 40
      IF(IER.EQ.129) WRITE(6,2010) NX,ICO
      IF(IER.EQ.130) WRITE(6,2020) NX
      IF(IER.EQ.131) WRITE(6,2030)
      STOP
C//////////////////////////////////////////////////////////////////////////
      ENTRY DSORPA(XBAR,XR)
C//////////////////////////////////////////////////////////////////////////
   40 Y=(XBAR-XARRAY(1))/(XR-XARRAY(1))
      CALL ICSEVU(YARRAY,SE1,NX,SEECO,ICO,Y,DSORP,1,IER)
      IF(IER.EQ.0) RETURN
      IF (IER.EQ.33) WRITE(6,2060) Y,YARRAY(1)
      IF (IER.EQ.34) WRITE(6,2070) Y,YARRAY(NX)
      STOP
 2010 FORMAT(/1X,'**** ERROR. IER=129 USING ICSCCU IN DSORP TO FIND SEEC
     *O.'/ 1X,'NX=',I3,' CANNOT EXCEED ICO=',I3,' BY MORE THAN ONE.')
 2020 FORMAT(/1X,'**** ERROR. IER=130 USING ICSCCU IN DSORP TO FIND SEEC
     *O.'/ 1X,'NX=',I3,' MUST BE AT LEAST TWO.')
 2030 FORMAT(/1X,'**** ERROR. IER=131 USING ICSCCU IN DSORP TO FIND SEEC
     *O.', 1X,'VECTOR XARRAY IS NOT ORDERED.')
 2040 FORMAT(/1X,'**** ERROR. IER=33 USING ICSEVU IN DSORP TO FIND "DESO
     *RP".'/1X, 'XR=',F10.6,' IS LESS THAN XARRAY(1)=',F10.6)
 2050 FORMAT(/1X,'**** ERROR. IER=34 USING ICSEVU IN DSORP TO FIND "DESO
     *RP".'/1X, 'XR=',F10.6,' IS GREATER THAN XARRAY(NX)=',F10.6)
 2060 FORMAT(/1X,'**** ERROR. IER=33 USING ICSEVU IN DSORPA TO FIND "DSO
     *RP".'/1X, 'Y=',F10.6,' IS LESS THAN YARRAY(1)=',F10.6)
 2070 FORMAT(/1X,'**** ERROR. IER=34 USING ICSEVU IN DSORPA TO FIND "DSO
     *RP".'/1X, 'Y=',F10.6,' IS GREATER THAN YARRAY(NX)=',F10.6)
      END
C//////////////////////////////////////////////////////////////////////////
C//////////////////////////////////////////////////////////////////////////
      FUNCTION EHCAP(X)
C//////////////////////////////////////////////////////////////////////////
C//////////////////////////////////////////////////////////////////////////
C
C     COMPUTE THE EFFECTIVE VOLUMETRIC HEAT CAPACITY OF THE POROUS MEDIUM
C     AS A SIMPLE WEIGHTED AVERAGE OF THE CAPACITIES OF THE CONSTITUENTS.
C     THE CONSTANT PART OF THIS WEIGHTED AVERAGE WAS PRE-COMPUTED BY SOILI.
C
C==========================================================================
      COMMON /BLOCK3/ HCD,HCAP(5)
C==========================================================================
      EHCAP=HCD+HCAP(1)*X
      RETURN
      END
C//////////////////////////////////////////////////////////////////////////
C//////////////////////////////////////////////////////////////////////////
      FUNCTION EMI(X)
C//////////////////////////////////////////////////////////////////////////
C//////////////////////////////////////////////////////////////////////////
C
C     FIND THE EMISSIVITY AS A FUNCTION OF THE MOISTURE CONTENT.
C
C==========================================================================
      COMMON /BLOCK6/ EMDRY,DEM
      EMI=EMDRY+DEM*X
```

```
      RETURN
      END
C////////////////////////////////////////////////////////////////////////////////
C////////////////////////////////////////////////////////////////////////////////
      FUNCTION ETCON(X)
C////////////////////////////////////////////////////////////////////////////////
C////////////////////////////////////////////////////////////////////////////////
C
C     FIND THE EFFECTIVE THERMAL CONDUCTIVITY OF THE SOIL USING DE VRIES
C     MODEL. XXX IS SET EQUAL TO X UNLESS X IS LESS THAN XK, THE MOISTURE
C     CONTENT AT WHICH LIQUID CONTINUITY FAILS. THE THERMAL CONDUCITIVITY
C     AT XXX IS FOUND. IF X EQUALS XXX, THAT VALUE IS RETURNED. IF X WAS
C     LESS THAN XK, LINEAR INTERPOLATION IS USED TO ESTIMATE ETCON BETWEEN
C     ITS VALUES AT DRYNESS AND AT X=XK.
C
C===============================================================================
      COMMON /BLOCK4/ CONO,GA1,GA2,GB1,GB2,POR,PS1,PS2,TCON(5),
     * VPER(5),XK,XWILT,ZETAO,SF(5)
C===============================================================================
      XXX=X
      IF(X.LT.XK) XXX=XK
      IF(XXX.LT.XWILT) GO TO 10
      SF(2)=GA1+GB1*XXX
      GO TO 20
   10 SF(2)=GA2+GB2*XXX
   20 FAC=TCON(2)/TCON(1)-1.
      GRAT=0.667/(1.+FAC*SF(2))+0.333/(1.+FAC*(1.-2.*SF(2)))
      ETCON=(PS1+XXX*TCON(1)+(POR-XXX)*GRAT*
     * TCON(2))/(PS2+XXX+(POR-XXX)*GRAT)
      IF(X.EQ.XXX)  RETURN
      FA=X/XXX
      ETCON=CONO+(ETCON-CONO)*FA
      RETURN
      END
C////////////////////////////////////////////////////////////////////////////////
C////////////////////////////////////////////////////////////////////////////////
      FUNCTION FC(TC,KOD)
C////////////////////////////////////////////////////////////////////////////////
C////////////////////////////////////////////////////////////////////////////////
C
C   FIND THE CUMULATIVE FLUX (FC) THAT CORRESPONDS TO THE CURRENT COMPRESSED
C   TIME (TC). KOD=O SIGNIFIES EXFILTRATION, AND KOD=1 IS INFILTRATION.
C
C===============================================================================
      COMMON /PARAM/ SE,SO,AO
C===============================================================================
      IF(KOD.EQ.1) GO TO 10
C===============================================================================
C
C   EXFILTRATION.
C
C===============================================================================
      FC=SE*SQRT(TC)
      RETURN
C===============================================================================
C
C   INFILTRATION.
C
C===============================================================================
   10 FC=AO*TC+SO*SQRT(TC)
```

```
      RETURN
      END
C///////////////////////////////////////////////////////////////////////////
C///////////////////////////////////////////////////////////////////////////
      FUNCTION FINV(F,Y,XL,XH,YERR,METHOD,KMAX)
C///////////////////////////////////////////////////////////////////////////
C///////////////////////////////////////////////////////////////////////////
C
C     FINV FINDS A VALUE OF X FOR WHICH THE ABSOLUTE VALUE OF
C     (F(X)-Y)/Y IS LESS THAN OR EQUAL TO YERR. IF METHOD=1, AN ORIGINAL
C     INTERVAL (XL,XH), WITHIN WHICH THE VALUE OF X IS KNOWN TO LIE,
C     IS SUCCESSIVELY BISECTED UNTIL THE DESIRED CONVERGENCE IS
C     ACHIEVED. IF METHOD=2, A NEWTON-RAPHSON SCHEME IS APPLIED. IN
C     THAT CASE, THE AVERAGE OF XL AND XH IS USED AS AN INITIAL GUESS.
C     KMAX IS THE MAXIMUM ALLOWABLE NUMBER OF ITERATIONS IN EITHER CASE.
C     IF THE NEWTON-RAPHSON METHOD FAILS TO CONVERGE, THE BISECTION
C     METHOD IS ATTEMPTED. THE FUNCTION F IS AN EXTERNAL, USER-SUPPLIED
C     FUNCTION.
C
C===========================================================================
      EXTERNAL F
C===========================================================================
      XLO=XL
      XHI=XH
      K=0
C===========================================================================
C
C     WHICH METHOD?
C
C===========================================================================
      IF(METHOD.EQ.2) GO TO 70
C===========================================================================
C
C   METHOD 1: SUCCESSIVE BISECTION.
C
C===========================================================================
   10 YLO=F(XLO)
      YHI=F(XHI)
      IF(YLO.LT.Y.AND.YHI.GT.Y) GO TO 20
      IF(YLO.GT.Y.AND.YHI.LT.Y) GO TO 40
      WRITE(6,2010)
      STOP
C===========================================================================
C
C   CASE WHEN DF/DX IS POSITIVE.
C
C===========================================================================
   20 K=K+1
      IF(K.GT.KMAX) GO TO 60
      FINV=0.5*(XLO+XHI)
      YM=F(FINV)
      IF(ABS(Y-YM).LE.YERR*Y) RETURN
      IF(YM.GT.Y) GO TO 30
      XLO=FINV
      GO TO 20
   30 XHI=FINV
      GO TO 20
C===========================================================================
C
C   CASE WHEN DF/DX IS NEGATIVE.
```

```
C
C==========================================================================
   40  K=K+1
       IF(K.GT.KMAX) GO TO 60
       FINV=0.5*(XLO+XHI)
       YM=F(FINV)
       IF(ABS(Y-YM).LE.YERR*Y) RETURN
       IF(YM.LT.Y) GO TO 50
       XLO=FINV
       GO TO 40
   50  XHI=FINV
       GO TO 40
   60  WRITE(6,2020)
       STOP
C==========================================================================
C
C      METHOD 2: NEWTON-RAPHSON ITERATION. F ASSUMED TO BE POSITIVE AND
C      EXPONENTIAL IN X.
C
C==========================================================================
   70  XLO=0.5*(XLO+XHI)
   80  K=K+1
       IF(K.GT.KMAX) GO TO 90
       FINV=XLO
       XHI=1.0001*XLO
       FLO=F(XLO)
       FHI=F(XHI)
       IF(FHI.LT.0..OR.FLO.LT.0.) GO TO 90
       DFDX=(FHI-FLO)/(XHI-XLO)
       XLO=XLO+FLO*ALOG(Y/FLO)/DFDX
       IF(XLO.LE.1.E-3) XLO=1.E-3
       IF(ABS(Y-FLO).LE.YERR*Y) RETURN
       IF(K.LT.KMAX) GO TO 80
C==========================================================================
C
C      NEWTON-RAPHSON FAILS, EITHER BECAUSE F WAS NEGATIVE OR BECAUSE
C      ALLOWABLE NUMBER OF ITERATIONS WAS EXCEEDED. TRY BISECTION.
C
C==========================================================================
   90  XLO=XL
       XHI=XH
       K=0
       GO TO 10
 2010  FORMAT(//1X,'**** ERROR. FINV, BISECTION, INVOKED WHEN (F(XLO)-Y)
      *AND (F(XHI)-Y) HAVE IDENTICAL SIGN')
 2020  FORMAT(//1X,'**** ERROR. FINV, BISECTION DOES NOT CONVERGE')
       END
C//////////////////////////////////////////////////////////////////////////
C//////////////////////////////////////////////////////////////////////////
       FUNCTION FXBAR(XBAR)
C//////////////////////////////////////////////////////////////////////////
C//////////////////////////////////////////////////////////////////////////
C
C      FXBAR CALCULATES THE INFILTRATION DEPTH RESULTING FROM A SURFACE
C      MOISTURE CONTENT OF XBAR BEING MAINTAINED ABOVE A PREVIOUSLY DRY
C      SOIL FOR DURATION TRAIN. THIS FUNCTION IS CALLED BY FINV.
C
C==========================================================================
       COMMON /FXCOM/ TRAIN
C==========================================================================
```

```
      FXBAR=0.5*TRAIN*HYCON(XBAR)+SORP(XBAR)*SQRT(TRAIN)
      RETURN
      END
C///////////////////////////////////////////////////////////////////////////
C///////////////////////////////////////////////////////////////////////////
      FUNCTION HYCON(X)
C///////////////////////////////////////////////////////////////////////////
C///////////////////////////////////////////////////////////////////////////
C
C     FIND THE HYDRAULIC CONDUCTIVITY GIVEN THE MOISTURE CONTENT, X.
C
C===========================================================================
      COMMON /BLOCK1/ AA,BB,CC,DD,EE,PRN,XU
      COMMON /BLOCK2/ CKSAT,ENT(25),PFR(25),PR(25),SS(25),
     *  XI(25),XIM,NSS
C===========================================================================
C
C     FIRST FIND THE INTERVAL, NK, WITHIN WHICH THE VALUE OF X LIES.
C
C===========================================================================
      DO 10 K=1,NSS
      NK=NSS+1-K
      IF(X.GT.XI(NK)) GO TO 20
      IF(K.EQ.NSS) GO TO 30
   10 CONTINUE
C===========================================================================
C
C     THEN FIND THE VALUE OF MATRIC POTENTIAL THIS CORRESPONDS TO BY
C     USING THE LINEAR APPROXIMATION BETWEEN MOISTURE CONTENT AND PF.
C     WITH THIS VALUE, EVALUATE THE EXPRESSION FOR HYDRAULIC CONDUCTIVITY.
C
C===========================================================================
   20 PFM=PFR(NK)-SS(NK)*(X-XI(NK))
      IF(PFM.LT.1.E-04) PFM=0.0
      PWET=-10.**PFM
      CK1=0.
      IF(NK.GT.1) CK1=ENT(NK-1)
      CKPART=(CK1+(1./PWET-1./PR(NK))
     */SS(NK))/ENT(NSS)
      CKPART=CKPART*CKPART
      HYCON=CKSAT*SQRT((X-XIM)/(XU-XIM))*CKPART
      IF(HYCON.LT.0.) HYCON=0.
      RETURN
   30 HYCON=0.
      RETURN
      END
C///////////////////////////////////////////////////////////////////////////
C///////////////////////////////////////////////////////////////////////////
      FUNCTION PHI(RI)
C///////////////////////////////////////////////////////////////////////////
C///////////////////////////////////////////////////////////////////////////
C
C     THIS SUBROUTINE DETERMINES THE STABILITY RATIO FOR ATMOSPHERIC
C     DIFFUSION OF HEAT AND VAPOR. FOR UNSTABLE CONDITIONS, THIS FUNCTION
C     USES THE VECTORS RIB (BULK RICHARDSON NUMBER) AND CHRAT (STABILITY
C     FACTOR FOR HEAT AND VAPOR TRANSFER) TO FIND THE DESIRED VALUE, STAB.
C     FOR STABLE CONDITIONS, AN EXPLICIT FORMULA IS USED. IN ALL CASES,
C     THE DERIVATIVE OF STAB WITH RESPECT TO RIB IS ALSO FOUND. STAB AND
C     DSTAB ARE RETURNED AS THE REAL AND IMAGINARY PARTS OF PHI.
C
```

```
C===========================================================================
      COMPLEX PHI
      COMMON /STBLTY/ RIB(21),CHRAT(21)
C===========================================================================
C
C     IS ATMOSPHERE STABLE?
C
C===========================================================================
      IF(RI.GE.RIB(1)) GO TO 30
C===========================================================================
C
C     UNSTABLE. INTERPOLATE FOR PHI INSIDE THE APPROPRIATE INTERVAL.
C
C===========================================================================
      DO 10 I=1,15
      II=I+1
      IF(RI.GE.RIB(II)) GO TO 20
   10 CONTINUE
   20 DELTA=RIB(II-1)-RIB(II)
      DSTAB=(CHRAT(II-1)-CHRAT(II))/(RIB(II-1)-RIB(II))
      STAB=CHRAT(II)+DSTAB*(RI-RIB(II))
      PHI=STAB+(0.,1.)*DSTAB
      RETURN
C===========================================================================
C
C     STABLE. IS RIB LESS THAN OR GREATER THAN ITS CRITICAL VALUE?
C
C===========================================================================
   30 IF(RI.GT..2) GO TO 40
C===========================================================================
C
C     RIB IS BELOW THE CRITICAL VALUE.
C
C===========================================================================
      TERM=(1.-5.*RI)
      DSTAB=-10.*TERM
      STAB=TERM*TERM
      PHI=STAB+(0.,1.)*DSTAB
      RETURN
C===========================================================================
C
C     RIB IS ABOVE THE CRITICAL VALUE. THERE IS NO TURBULENT TRANSFER.
C
C===========================================================================
   40 DSTAB=0.
      STAB=0.
      PHI=STAB+(0.,1.)*DSTAB
      RETURN
      END
C///////////////////////////////////////////////////////////////////////////
C///////////////////////////////////////////////////////////////////////////
      FUNCTION RHOZ(TEE)
C///////////////////////////////////////////////////////////////////////////
C///////////////////////////////////////////////////////////////////////////
C
C     THIS SUBPROGRAM CALCULATES THE SATURATION VAPOR DENSITY AND ITS
C     DERIVATIVE AS FUNCTIONS OF TEMPERATURE. IN ORDER TO USE THE FUNCTION
C     FORMAT, WE STORE THESE TWO VARIABLES AS THE REAL AND IMAGINARY
C     PARTS OF A COMPLEX VALUE. RZ(I+1) CONTAINS THE VALUE OF SATURATION
C     ABSOLUTE HUMIDITY AT TEMPERATURE I (DEGREES CELSIUS). LINEAR
```

```
C      INTERPOLATION IS USED.
C
C=========================================================================
       COMPLEX RHOZ
       COMMON /ABSHU/ RZ(81)
C=========================================================================
       L=TEE
       EXTRA=TEE-FLOAT(L)
       DRDT=RZ(L+2)-RZ(L+1)
       RHOZ=RZ(L+1)+(EXTRA+(0.,1.))*DRDT
       RETURN
       END
C/////////////////////////////////////////////////////////////////////////
C/////////////////////////////////////////////////////////////////////////
       FUNCTION SORP(XS)
C/////////////////////////////////////////////////////////////////////////
C/////////////////////////////////////////////////////////////////////////
C
C      USE INTERPOLATION BY CUBIC SPLINES TO CALCULATE SORPTIVITY, GIVEN
C      THE MOISTURE CONTENT XS.
C
C=========================================================================
       COMMON /SORCOM/ XARRAY(21),YARRAY(21),SIARAY(21),
      *  SEARAY(21,21),SICO(20,3),SEE(21),SE1(21),SEECO(20,3),
      *  XB,XWR,ICO,NX
C=========================================================================
       CALL ICSEVU(XARRAY,SIARAY,NX,SICO,ICO,XS,SORP,1,IER)
       IF(IER.EQ.0) RETURN
       IF(IER.EQ.33) WRITE(6,2010) XS,XARRAY(1)
       IF(IER.EQ.34) WRITE(6,2020) XS,XARRAY(NX)
       STOP
2010   FORMAT(/1X,'**** ERROR. IER=33 USING ICSEVU IN SORP.'
      */1X, 'XS=',F10.6,' IS LESS THAN XARRAY(1)=',F10.6)
2020   FORMAT(/1X,'**** ERROR. IER=34 USING ICSEVU IN SORP.'
      * /1X, 'XS=',F10.6,' IS GREATER THAN XARRAY(NX)=',F10.6)
       END
C/////////////////////////////////////////////////////////////////////////
C/////////////////////////////////////////////////////////////////////////
       FUNCTION TC(FC,KOD)
C/////////////////////////////////////////////////////////////////////////
C/////////////////////////////////////////////////////////////////////////
C
C      DETERMINE THE COMPRESSED TIME (TC) THAT CORRESPONDS TO THE CUMULATIVE
C      FLUX DEPTH (FC). KOD=0 SIGNIFIES EXFILTRATION, KOD=1 IS INFILTRATION.
C
C=========================================================================
       COMMON /PARAM/ SE,SO,AO
C=========================================================================
       IF(KOD.EQ.1) GO TO 20
C=========================================================================
C
C    EXFILTRATION.
C
C=========================================================================
       IF(SE.EQ.0.) GO TO 10
       FAC=FC/SE
       TC=FAC*FAC
       RETURN
   10  TC=0.
       RETURN
```

```
C===========================================================================
C
C   INFILTRATION.
C
C===========================================================================
  20  ROOTT=0.5*(-SO+SQRT(SO*SO+4.*AO*FC))/AO
      TC=ROOTT*ROOTT
      RETURN
      END
C///////////////////////////////////////////////////////////////////////////
C///////////////////////////////////////////////////////////////////////////
      FUNCTION TORT(XA)
C///////////////////////////////////////////////////////////////////////////
C///////////////////////////////////////////////////////////////////////////
C
C     FIND THE TORTUOSITY OF THE AIR PHASE AS A FUNCTION OF THE VOLUMETRIC
C     AIR CONTENT.
C
C===========================================================================
      TORT=XA**0.667
      RETURN
      END
C///////////////////////////////////////////////////////////////////////////
C///////////////////////////////////////////////////////////////////////////
      FUNCTION XWET(POT)
C///////////////////////////////////////////////////////////////////////////
C///////////////////////////////////////////////////////////////////////////
C
C     THIS FUNCTION FINDS THE VALUE OF THE MAIN WETTING FUNCTION, GIVEN
C     THE SOIL PARAMETERS AND THE MATRIC POTENTIAL, POT.
C
C===========================================================================
      COMMON /BLOCK1/ AA,BB,CC,DD,EE,PRN,XU
C===========================================================================
      DATA DLN10 /2.302585093/
      IF(POT.GE.PRN) GO TO 10
      IF(POT.LE.-1.E7) GO TO 20
      PF=ALOG10(-POT)
      ARG1=POT/BB
      ARG2=AA*ARG1**CC
      XWET=ARG2+DD*(7.-PF)+EE
      RETURN
  10  XWET=XU
      RETURN
  20  XWET=1.E-8
      RETURN
      END
C///////////////////////////////////////////////////////////////////////////
C///////////////////////////////////////////////////////////////////////////
      BLOCK DATA
C///////////////////////////////////////////////////////////////////////////
C///////////////////////////////////////////////////////////////////////////
C
C     INITIALIZE THE ARRAYS RZ, TCON, SF, AND HCAP, AS WELL AS
C     SOME PHYSICAL CONSTANTS.
C
C===========================================================================
      COMMON /ABSHU/ RZ(81)
      LOGICAL*1 RAIN,RAINB4
      COMMON /BIG/  STACK(6,10), SUM(8), O(8),
```

```
     * ALBEDO,      C1, C1PART,     C2,       C3,  CAPEX, CAPEX1,
     * CAPEX2,  CAPIN,    CDN,  CDUOR,       CL,     CP, CPNOLD,   DEDT,
     *   DELT,     DF,   DFIL, DRADLU,  DSENS,      E,      EO,    ELO,
     *   EMIS,     EP,     FE,    FEP,       FI,     G,    SENS,      P,
     *      R,  RADLD, RADLUO,   RADS,  RCCDU,  RHOCP,    RHOL,  RHOVA,
     *   RLCL,    RLE,     RN,     RS,  ROOTO,  ROOT1,   ROOT2,  SENSO,
     *    SIG,     T1, T1LAST,  T1OLD,      T2,     TA,     TCE,    TCI,
     *   TIME,  TSTAB,   XBAR,  XINIT,      XR,     XZ,      ZA,
     *   IFOR,  IRAIN,   ISTK,   IUNF, MAXSTK,   NPER, NSTACK,
     *   RAIN, RAINB4
      COMMON /BLOCK3/ HCD,HCAP(5)
      COMMON /BLOCK4/ CONO,GA1,GA2,GB1,GB2,POR,PS1,PS2,TCON(5),
     * VPER(5),XK,XWILT,ZETAO,SF(5)
C==============================================================================
       DATA RZ /0.4831E-05,0.5178E-05,0.5545E-05,0.5933E-05,0.6344E-05,
     1          0.6779E-05,0.7240E-05,0.7728E-05,0.8246E-05,0.8793E-05,
     2          0.9373E-05,0.9984E-05,0.1063E-04,0.1131E-04,0.1203E-04,
     3          0.1279E-04,0.1358E-04,0.1443E-04,0.1532E-04,0.1626E-04,
     4          0.1724E-04,0.1827E-04,0.1936E-04,0.2050E-04,0.2170E-04,
     5          0.2297E-04,0.2429E-04,0.2567E-04,0.2713E-04,0.2865E-04,
     6          0.3025E-04,0.3193E-04,0.3368E-04,0.3552E-04,0.3744E-04,
     7          0.3945E-04,0.4154E-04,0.4374E-04,0.4602E-04,0.4841E-04,
     8          0.5091E-04,0.5352E-04,0.5624E-04,0.5907E-04,0.6203E-04,
     9          0.6511E-04,0.6831E-04,0.7165E-04,0.7512E-04,0.7874E-04,
     A          0.8250E-04,0.8641E-04,0.9049E-04,0.9472E-04,0.9910E-04,
     1          0.1036E-03,0.1084E-03,0.1133E-03,0.1184E-03,0.1237E-03,
     2          0.1292E-03,0.1349E-03,0.1408E-03,0.1469E-03,0.1533E-03,
     3          0.1598E-03,0.1666E-03,0.1737E-03,0.1809E-03,0.1884E-03,
     4          0.1963E-03,0.2042E-03,0.2126E-03,0.2212E-03,0.2301E-03,
     5          0.2393E-03,0.2488E-03,0.2585E-03,0.2686E-03,0.2791E-03,
     6          0.2898E-03/
      DATA TCON /1.37E-3, 6.E-5, 2.1E-2, 7.E-3, 6.E-4/
      DATA SF /0., 0., .125, .125, .5/
      DATA HCAP /.998, .0003, .46, .46, .6/
      DATA CL,CP,DF,ELO,RHOCP,RHOL,RLCL,SIG /1.,.448,86400.,597.3,
     * 2.88E-4,0.998,0.998,1.375E-12/
      DATA MAXSTK /10/
      END
```

Appendix F

Sample Input

```
*****************************************************************************

                            ****************
                            *              *
                            *    SWAHP     *
                            *              *
                            ****************


                 SAMPLE SWAHP SIMULATION

     FORMATTED OUTPUT GOING TO FILE NUMBER  8
     UNFORMATTED OUTPUT TO                  0

     INPUT:
      R =  0.20E+01  TAVG =  0.14E+02  ENDAYS =  0.37E+03  ZO =  0.10E+00
     ZA =  0.20E+03  XHAT =  0.16E+00      T1 =  0.14E+02  T2 =  0.14E+02


     OUTPUT FOR EACH PERIOD FOLLOWS.
```

DAYS	P	E	XBAR	T1
0.04	0.42E-05	0.59E-06	0.12E+00	0.12E+02
0.08	0.42E-05	0.20E-06	0.11E+00	0.12E+02
0.13	0.42E-05	-0.14E-07	0.11E+00	0.11E+02
0.17	0.42E-05	-0.96E-07	0.11E+00	0.11E+02
0.21	0.34E-04	-0.58E-07	0.11E+00	0.11E+02
0.25	0.34E-04	0.97E-07	0.10E+00	0.11E+02
0.29	0.34E-04	0.66E-06	0.10E+00	0.12E+02
0.33	0.34E-04	0.17E-05	0.10E+00	0.14E+02
0.38	0.34E-04	0.32E-05	0.10E+00	0.15E+02
0.42	0.34E-04	0.48E-05	0.10E+00	0.17E+02
0.46	0.84E-05	0.64E-05	0.99E-01	0.18E+02
0.50	0.84E-05	0.76E-05	0.98E-01	0.19E+02
0.54	0.84E-05	0.82E-05	0.97E-01	0.20E+02

AT DAYS = 0.54, ADDING PREVIOUS DATA TO STACK. NOW NSTACK = 1

AT DAYS = 0.54, RAIN ENDS. FI= 0.76 AND RESULTANT XBAR= 0.3055

0.58	0.00E+00	0.13E-04	0.31E+00	0.23E+02
0.63	0.00E+00	0.13E-04	0.30E+00	0.23E+02
0.67	0.00E+00	0.11E-04	0.30E+00	0.22E+02
0.71	0.00E+00	0.81E-05	0.29E+00	0.20E+02
0.75	0.00E+00	0.50E-05	0.29E+00	0.18E+02
0.79	0.00E+00	0.26E-05	0.29E+00	0.16E+02
0.83	0.00E+00	0.16E-05	0.29E+00	0.14E+02
0.88	0.00E+00	0.97E-06	0.29E+00	0.13E+02
0.92	0.00E+00	0.52E-06	0.28E+00	0.12E+02

3.13	0.00E+00	-0.59E-07	0.18E+00	0.11E+02
3.17	0.00E+00	-0.83E-07	0.18E+00	0.11E+02
3.21	0.00E+00	-0.33E-07	0.18E+00	0.11E+02
3.25	0.00E+00	0.13E-06	0.18E+00	0.11E+02
3.29	0.00E+00	0.69E-06	0.18E+00	0.12E+02
3.33	0.00E+00	0.17E-05	0.18E+00	0.14E+02
3.38	0.00E+00	0.32E-05	0.18E+00	0.16E+02

AT DAYS = 3.42, EXTRACTING FROM STACK. NSTACK= 1

3.42	0.00E+00	0.38E-05	0.25E+00	0.18E+02
3.46	0.00E+00	0.31E-05	0.25E+00	0.20E+02
3.50	0.00E+00	0.30E-05	0.25E+00	0.22E+02
3.54	0.00E+00	0.29E-05	0.25E+00	0.24E+02
3.58	0.00E+00	0.29E-05	0.25E+00	0.29E+02
3.63	0.00E+00	0.28E-05	0.25E+00	0.30E+02
3.67	0.00E+00	0.27E-05	0.25E+00	0.29E+02

AT DAYS = 3.71, EXTRACTING FROM STACK. NSTACK= 0

3.71	0.00E+00	0.11E-05	0.81E-01	0.26E+02
3.75	0.00E+00	-0.18E-14	0.81E-01	0.23E+02
3.79	0.00E+00	0.00E+00	0.81E-01	0.20E+02
3.83	0.00E+00	0.00E+00	0.81E-01	0.17E+02
3.88	0.00E+00	0.00E+00	0.81E-01	0.15E+02
3.92	0.00E+00	0.00E+00	0.81E-01	0.14E+02
3.96	0.00E+00	0.00E+00	0.81E-01	0.12E+02
4.00	0.00E+00	0.00E+00	0.81E-01	0.11E+02
4.04	0.00E+00	-0.20E-06	0.81E-01	0.11E+02
4.08	0.00E+00	-0.38E-06	0.81E-01	0.10E+02
4.13	0.00E+00	-0.49E-06	0.81E-01	0.98E+01
4.17	0.00E+00	-0.53E-06	0.81E-01	0.97E+01
4.21	0.00E+00	-0.51E-06	0.81E-01	0.98E+01
4.25	0.00E+00	-0.35E-06	0.81E-01	0.10E+02
4.29	0.00E+00	0.53E-06	0.81E-01	0.12E+02
4.33	0.00E+00	0.19E-05	0.81E-01	0.16E+02
4.38	0.00E+00	0.00E+00	0.81E-01	0.21E+02
4.42	0.00E+00	0.00E+00	0.81E-01	0.26E+02
4.46	0.00E+00	0.00E+00	0.81E-01	0.30E+02
4.50	0.00E+00	0.00E+00	0.81E-01	0.33E+02

SIMULATION COMPLETED. AVERAGE FLUXES:

RN	188.0	LANGLEYS/DAY
G	39.4	"
SENS	26.1	"
RLE	122.5	"

P	0.202	CENTIMETERS/DAY
EP	0.419	"
E	0.202	"
RS	-0.000	"

Appendix G

Sample Output

```
                    SAMPLE SWAHP SIMULATION
8    0    1    1 -82
     2.        14.       365.       0.1       200.    .16079        14.       14.
0.46000    0.41400 0.125E-03  0.21000-495.00000  -0.14700   0.00000  -0.04890
0.11000    0.16000   0.33000   0.05000   0.20000   0.10000   0.90000   0.95000
  3600. 0.421E-05 0.100E-04 0.261E+03 0.000E+00 0.809E-02 0.119E+02
  3600. 0.421E-05 0.100E-04 0.261E+03 0.000E+00 0.800E-02 0.113E+02
  3600. 0.421E-05 0.100E-04 0.268E+03 0.000E+00 0.796E-02 0.111E+02
  3600. 0.421E-05 0.100E-04 0.281E+03 0.000E+00 0.800E-02 0.113E+02
  3600. 0.337E-04 0.100E-04 0.299E+03 0.000E+00 0.809E-02 0.119E+02
  3600. 0.337E-04 0.100E-04 0.322E+03 0.110E-03 0.824E-02 0.127E+02
  3600. 0.337E-04 0.100E-04 0.347E+03 0.110E-02 0.845E-02 0.139E+02
  3600. 0.337E-04 0.100E-04 0.373E+03 0.255E-02 0.869E-02 0.153E+02
  3600. 0.337E-04 0.100E-04 0.398E+03 0.404E-02 0.895E-02 0.167E+02
  3600. 0.337E-04 0.100E-04 0.421E+03 0.534E-02 0.922E-02 0.181E+02
  3600. 0.843E-05 0.100E-04 0.439E+03 0.628E-02 0.948E-02 0.195E+02
  3600. 0.843E-05 0.100E-04 0.452E+03 0.678E-02 0.971E-02 0.207E+02
  3600. 0.843E-05 0.100E-04 0.459E+03 0.678E-02 0.989E-02 0.215E+02
  3600. 0.000E+00 0.100E-04 0.459E+03 0.150E-01 0.891E-02 0.221E+02
  3600. 0.000E+00 0.100E-04 0.452E+03 0.128E-01 0.895E-02 0.223E+02
  3600. 0.000E+00 0.100E-04 0.439E+03 0.967E-02 0.891E-02 0.221E+02
  3600. 0.000E+00 0.100E-04 0.421E+03 0.610E-02 0.881E-02 0.215E+02
  3600. 0.000E+00 0.100E-04 0.398E+03 0.263E-02 0.865E-02 0.207E+02
  3600. 0.000E+00 0.100E-04 0.373E+03 0.263E-03 0.845E-02 0.195E+02
  3600. 0.000E+00 0.100E-04 0.347E+03 0.000E+00 0.822E-02 0.181E+02
  3600. 0.000E+00 0.100E-04 0.322E+03 0.000E+00 0.798E-02 0.167E+02
  3600. 0.000E+00 0.100E-04 0.299E+03 0.000E+00 0.774E-02 0.153E+02
  3600. 0.000E+00 0.100E-04 0.281E+03 0.000E+00 0.752E-02 0.139E+02
  3600. 0.000E+00 0.100E-04 0.268E+03 0.000E+00 0.734E-02 0.127E+02
  3600. 0.000E+00 0.100E-04 0.261E+03 0.000E+00 0.721E-02 0.119E+02
  3600. 0.000E+00 0.100E-04 0.261E+03 0.000E+00 0.712E-02 0.113E+02
  3600. 0.000E+00 0.100E-04 0.268E+03 0.000E+00 0.710E-02 0.111E+02
  3600. 0.000E+00 0.100E-04 0.281E+03 0.000E+00 0.712E-02 0.113E+02
  3600. 0.000E+00 0.100E-04 0.299E+03 0.000E+00 0.721E-02 0.119E+02
  3600. 0.000E+00 0.100E-04 0.322E+03 0.263E-03 0.734E-02 0.127E+02
  3600. 0.000E+00 0.100E-04 0.347E+03 0.263E-02 0.752E-02 0.139E+02
  3600. 0.000E+00 0.100E-04 0.373E+03 0.610E-02 0.774E-02 0.153E+02
  3600. 0.000E+00 0.100E-04 0.398E+03 0.967E-02 0.798E-02 0.167E+02
  3600. 0.000E+00 0.100E-04 0.421E+03 0.128E-01 0.822E-02 0.181E+02
  3600. 0.000E+00 0.100E-04 0.439E+03 0.150E-01 0.845E-02 0.195E+02
  3600. 0.000E+00 0.100E-04 0.452E+03 0.162E-01 0.865E-02 0.207E+02
  3600. 0.000E+00 0.100E-04 0.459E+03 0.162E-01 0.881E-02 0.215E+02
  3600. 0.000E+00 0.100E-04 0.459E+03 0.150E-01 0.891E-02 0.221E+02
  3600. 0.000E+00 0.100E-04 0.452E+03 0.128E-01 0.895E-02 0.223E+02
  3600. 0.000E+00 0.100E-04 0.439E+03 0.967E-02 0.891E-02 0.221E+02
  3600. 0.000E+00 0.100E-04 0.421E+03 0.610E-02 0.881E-02 0.215E+02
  3600. 0.000E+00 0.100E-04 0.398E+03 0.263E-02 0.865E-02 0.207E+02
  3600. 0.000E+00 0.100E-04 0.373E+03 0.263E-03 0.845E-02 0.195E+02
  3600. 0.000E+00 0.100E-04 0.347E+03 0.000E+00 0.822E-02 0.181E+02
  3600. 0.000E+00 0.100E-04 0.322E+03 0.000E+00 0.798E-02 0.167E+02
  3600. 0.000E+00 0.100E-04 0.299E+03 0.000E+00 0.774E-02 0.153E+02
  3600. 0.000E+00 0.100E-04 0.281E+03 0.000E+00 0.752E-02 0.139E+02
  3600. 0.000E+00 0.100E-04 0.268E+03 0.000E+00 0.734E-02 0.127E+02
  3600. 0.000E+00 0.100E-04 0.261E+03 0.000E+00 0.721E-02 0.119E+02
  3600. 0.000E+00 0.100E-04 0.261E+03 0.000E+00 0.712E-02 0.113E+02
  3600. 0.000E+00 0.100E-04 0.268E+03 0.000E+00 0.710E-02 0.111E+02
  3600. 0.000E+00 0.100E-04 0.281E+03 0.000E+00 0.712E-02 0.113E+02
  3600. 0.000E+00 0.100E-04 0.299E+03 0.000E+00 0.721E-02 0.119E+02
  3600. 0.000E+00 0.100E-04 0.322E+03 0.263E-03 0.734E-02 0.127E+02
  3600. 0.000E+00 0.100E-04 0.347E+03 0.263E-02 0.752E-02 0.139E+02
```

3600.	0.000E+00	0.100E-04	0.373E+03	0.610E-02	0.774E-02	0.153E+02
3600.	0.000E+00	0.100E-04	0.398E+03	0.967E-02	0.798E-02	0.167E+02
3600.	0.000E+00	0.100E-04	0.421E+03	0.128E-01	0.822E-02	0.181E+02
3600.	0.000E+00	0.100E-04	0.439E+03	0.150E-01	0.845E-02	0.195E+02
3600.	0.000E+00	0.100E-04	0.452E+03	0.162E-01	0.865E-02	0.207E+02
3600.	0.000E+00	0.100E-04	0.459E+03	0.162E-01	0.881E-02	0.215E+02
3600.	0.000E+00	0.100E-04	0.459E+03	0.150E-01	0.891E-02	0.221E+02
3600.	0.000E+00	0.100E-04	0.452E+03	0.128E-01	0.895E-02	0.223E+02
3600.	0.000E+00	0.100E-04	0.439E+03	0.967E-02	0.891E-02	0.221E+02
3600.	0.000E+00	0.100E-04	0.421E+03	0.610E-02	0.881E-02	0.215E+02
3600.	0.000E+00	0.100E-04	0.398E+03	0.263E-02	0.865E-02	0.207E+02
3600.	0.000E+00	0.100E-04	0.373E+03	0.263E-03	0.845E-02	0.195E+02
3600.	0.000E+00	0.100E-04	0.347E+03	0.000E+00	0.822E-02	0.181E+02
3600.	0.000E+00	0.100E-04	0.322E+03	0.000E+00	0.798E-02	0.167E+02
3600.	0.000E+00	0.100E-04	0.299E+03	0.000E+00	0.774E-02	0.153E+02
3600.	0.421E-05	0.100E-04	0.281E+03	0.000E+00	0.845E-02	0.139E+02
3600.	0.421E-05	0.100E-04	0.268E+03	0.000E+00	0.824E-02	0.127E+02
3600.	0.000E+00	0.100E-04	0.261E+03	0.000E+00	0.809E-02	0.119E+02
3600.	0.000E+00	0.100E-04	0.261E+03	0.000E+00	0.800E-02	0.113E+02
3600.	0.000E+00	0.100E-04	0.268E+03	0.000E+00	0.796E-02	0.111E+02
3600.	0.000E+00	0.100E-04	0.281E+03	0.000E+00	0.800E-02	0.113E+02
3600.	0.000E+00	0.100E-04	0.299E+03	0.000E+00	0.809E-02	0.119E+02
3600.	0.000E+00	0.100E-04	0.322E+03	0.110E-03	0.824E-02	0.127E+02
3600.	0.000E+00	0.100E-04	0.347E+03	0.110E-02	0.845E-02	0.139E+02
3600.	0.000E+00	0.100E-04	0.373E+03	0.255E-02	0.869E-02	0.153E+02
3600.	0.000E+00	0.100E-04	0.398E+03	0.404E-02	0.895E-02	0.167E+02
3600.	0.000E+00	0.100E-04	0.421E+03	0.534E-02	0.922E-02	0.181E+02
3600.	0.000E+00	0.100E-04	0.439E+03	0.628E-02	0.948E-02	0.195E+02
3600.	0.000E+00	0.100E-04	0.452E+03	0.678E-02	0.971E-02	0.207E+02
3600.	0.000E+00	0.100E-04	0.459E+03	0.678E-02	0.989E-02	0.215E+02
3600.	0.000E+00	0.100E-04	0.459E+03	0.150E-01	0.891E-02	0.221E+02
3600.	0.000E+00	0.100E-04	0.452E+03	0.128E-01	0.895E-02	0.223E+02
3600.	0.000E+00	0.100E-04	0.439E+03	0.967E-02	0.891E-02	0.221E+02
3600.	0.000E+00	0.100E-04	0.421E+03	0.610E-02	0.881E-02	0.215E+02
3600.	0.000E+00	0.100E-04	0.398E+03	0.263E-02	0.865E-02	0.207E+02
3600.	0.000E+00	0.100E-04	0.373E+03	0.263E-03	0.845E-02	0.195E+02
3600.	0.000E+00	0.100E-04	0.347E+03	0.000E+00	0.822E-02	0.181E+02
3600.	0.000E+00	0.100E-04	0.322E+03	0.000E+00	0.798E-02	0.167E+02
3600.	0.000E+00	0.100E-04	0.299E+03	0.000E+00	0.774E-02	0.153E+02
3600.	0.000E+00	0.100E-04	0.281E+03	0.000E+00	0.752E-02	0.139E+02
3600.	0.000E+00	0.100E-04	0.268E+03	0.000E+00	0.734E-02	0.127E+02
3600.	0.000E+00	0.100E-04	0.261E+03	0.000E+00	0.721E-02	0.119E+02
3600.	0.000E+00	0.100E-04	0.261E+03	0.000E+00	0.712E-02	0.113E+02
3600.	0.000E+00	0.100E-04	0.268E+03	0.000E+00	0.710E-02	0.111E+02
3600.	0.000E+00	0.100E-04	0.281E+03	0.000E+00	0.712E-02	0.113E+02
3600.	0.000E+00	0.100E-04	0.299E+03	0.000E+00	0.721E-02	0.119E+02
3600.	0.000E+00	0.100E-04	0.322E+03	0.263E-03	0.734E-02	0.127E+02
3600.	0.000E+00	0.100E-04	0.347E+03	0.263E-02	0.752E-02	0.139E+02
3600.	0.000E+00	0.100E-04	0.373E+03	0.610E-02	0.774E-02	0.153E+02
3600.	0.000E+00	0.100E-04	0.398E+03	0.967E-02	0.798E-02	0.167E+02
3600.	0.000E+00	0.100E-04	0.421E+03	0.128E-01	0.822E-02	0.181E+02
3600.	0.000E+00	0.100E-04	0.439E+03	0.150E-01	0.845E-02	0.195E+02
3600.	0.000E+00	0.100E-04	0.452E+03	0.162E-01	0.865E-02	0.207E+02
-1.	0.	0.	0.	0.	0.	