Centrum voor Agrobiologisch Onderzoek
Landbouwuniversiteit Wageningen

The soil water balance module SAWAH: description and users guide

Simulation Reports CABO-TT, no. 22

H.F.M. ten Berge, D.M. Jansen, K. Rappoldt and W. Stol

cabo-dlo

Maart 1992

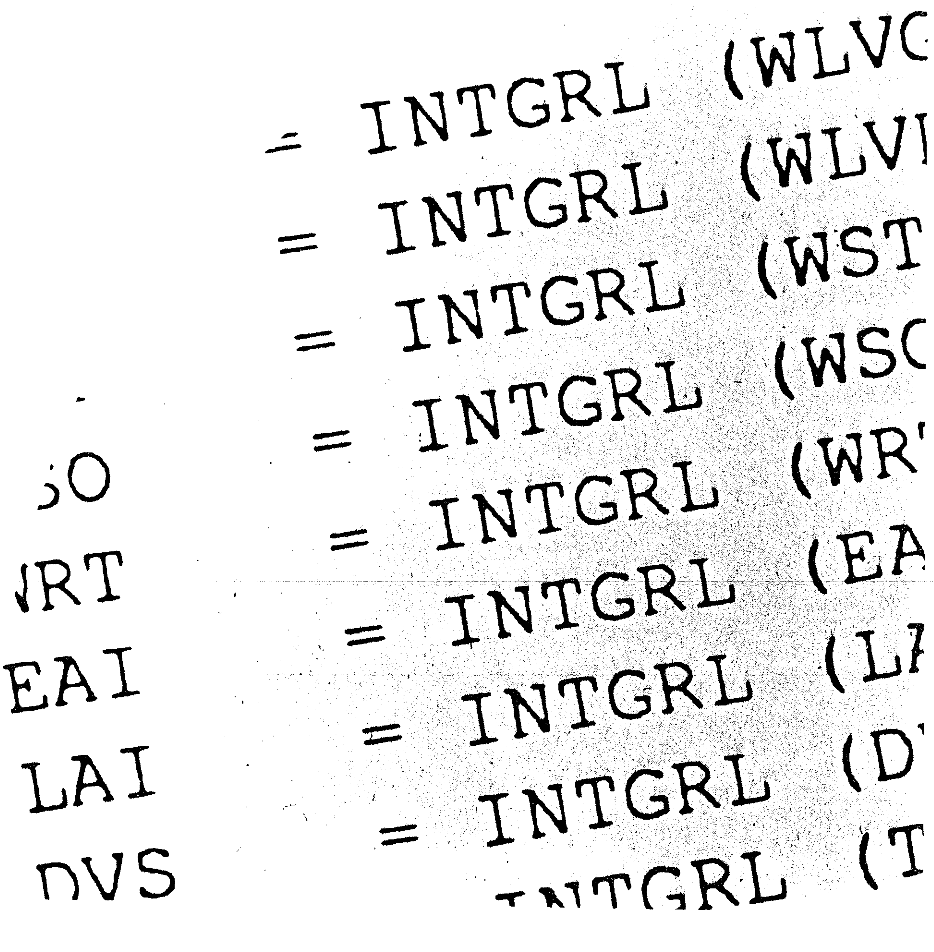
Centrum voor Agrobiologisch Onderzoek
Landbouwuniversiteit Wageningen

The soil water balance module SAWAH: description and users guide

Simulation Reports CABO-TT, no. 22

H.F.M. ten Berge, D.M. Jansen, K. Rappoldt and W. Stol

cabo-dlo



Maart 1992

The soil water balance module SAWAH: description and users guide

H.F.M. ten Berge, D.M. Jansen, K. Rappoldt and W. Stol

Simulation Reports CABO-TT nr. 22

maart 1992

DLO-Centre for Agrobiological Research (CABO-DLO)
P.O. Box 14, 6700 AA Wageningen, The Netherlands
and
Department of Theoretical Production Ecology (TPE), Agricultural University
Postbus 14, 6700 AA Wageningen, The Netherlands

Simulation Reports CABO-TT

Simulation Reports CABO-TT is a series giving supplementary information on agricultural simulation models that have been published elsewhere. Knowledge of those publications will generally be necessary in order to be able to study this material.

Simulation Reports CABO-TT describe improvements of simulation models, new applications or translocations of the programs into other computer languages. Manuscripts or suggestions should be submitted to:
H. van Keulen (CABO-DLO) or J. Goudriaan (TPE)

Simulation Reports CABO-TT are issued by CABO-DLO and TPE and they are available on request. Announcements of new reports will be issued regularly. Addresses of those who are interested in the announcements will be put on a mailing list on request.

Address

Simulation Reports CABO-TT
P.O. Box 14
6700 AA Wageningen
Netherlands

Authors affiliation

H.F.M. ten Berge
DLO-Centre for Agrobiological Research
P.O. Box 14, 6700 AA Wageningen, The Netherlands

D.M. Jansen
Centro Agronómico Tropical de Investigación y Enseñanza
P.O. 34 Turrialba Costa Rica

K. Rappoldt
DLO-Institute for Soil Fertility Research
P.O. Box 30003, 9750 RA Haren, The Netherlands

W. Stol
DLO-Centre for Agrobiological Research
P.O. Box 14, 6700 AA Wageningen, The Netherlands

1 Introduction **Page**

2 Interfacing SAWAH submodel with user-defined MAIN program

3 Program structure of SAWAH

4 The processes: theory and modelling

5 SAWAH subroutines

1 INTRODUCTION

1.1 Submodels

Simulation models are used by scientists to test hypotheses, and to study the behaviour of systems under varying conditions. Also, models serve a purpose in education. The increased use of complex models now makes it more and more difficult to interpret model outputs and to compare results obtained from different models. Often, this problem arises because models are composed of ill-defined submodels, that can not easily be exchanged with alternatives. The modelling efforts in the context of crop soil-water relations are no exception. Soil, crop and atmosphere processes are often all incorporated in a single model, the components of which cannot easily be replaced by alternative formulations. Yet, flexibility in using alternative submodels is essential to efficient research.

Some guidelines have recently been developed within the CABO-TPE group (e.g. van Kraalingen and Rappoldt, 1989). Standardized formats to connect MAIN programs with submodels, and a standard structure of the MAIN algorithm are important, of course. Functions that are speculative, or that have been selected on the basis of personal preference should be easily accessible to allow changes. Model options should be clearly presented. In this study an attempt has been made to follow these guidelines.

1.2 The SAWAH submodel

SAWAH is a Simulation Algorithm for Water flow in Aquic Habitats. It simulates one-dimensional vertical movement of water in soil, and describes on a field scale the dynamics of soil water content, water potential, and water flux at different depths in the soil profile, i.e. between the soil surface and a lower boundary at a user-specified depth.

SAWAH is a deterministic model, based on standard physical soil characteristics (moisture retention curve, hydraulic conductivity curve), measured groundwater levels, and meteorological conditions. It solves the *general flow equation* numerically, under given boundary conditions and root sink strength, and keeps an account of the soil water balance.

No crop processes are included in the model. Root growth and water uptake by the crop must be externally calculated from the current soil water status, provided by SAWAH, and crop characteristics. Uptake is then entered as a sink term into the water balance module.

SAWAH was designed to be connected as a soil water balance module to crop growth simulation models. It may also be applied to cases without crop cover; and it can easily be connected to groundwater models. Emphasis is on robustness and 'transportability', rather than on speed of execution.

SAWAH is offered to the user in the form of an easy-to-use package that can be quickly connected to most dynamic simulation models in FORTRAN or CSMP. It is always 'called' as a subroutine from the user-defined MAIN program.

1.3 Typical applications

A typical feature is SAWAH's ability to handle situations where part of the soil profile is *water-saturated*. This occurs:

- when perched water tables develop, as a result of low-permeability layers;
- when shallow or groundwater is present;
- when alluvial flooding or torrential precipitation are frequent;
- in any combination of these.

The true groundwater level may be *below* the profile depth considered, *in* the profile, or it may be *above* the soil surface. Any continuous transition from one of these situations to another, as occurs in nature, is simulated without user intervention. The same is true for perched water tables, that may be present *within* the profile or may be *above* the surface. The effects of artificial drainage are taken into account, although still in a provisional manner.

For free-draining soils with high hydraulic conductivity and rapid redistribution to a well-defined field capacity, faster and simpler models are available. SAWAH has no comparative advantage in those cases, because of the small time constants that characterize such permeable systems. Then, models of a more integrated nature (Subsection 1.6.1) are preferable, and a good alternative is the SAHEL submodel.

In crop modelling, SAWAH may be preferred to other models when capillary rise to the root zone is significant, either from true or perched groundwater, and in general when processes related to water transport affect crop growth. Examples are anaerobiosis in the root zone induced by impeded drainage, ponding of water on the surface, solute transport, nitrogen dynamics as affected by soil hydrology and aeration, etc.

Typical cases are the lowland rice systems in the (sub)tropics and the soils in other lowland regions with imperfect drainage, either due to groundwater or to low permeability. Such soils are common throughout The Netherlands.

SAWAH provides several options that can be selected according to the problem simulated. Among these are several forms of moisture characteristics and conductivity curves, and boundary conditions.

1.4 About this manual

This manual is a guide to users of SAWAH. In view of different backgrounds and motivations of potential users, the module is described from different starting points. Therefore, overlap among paragraphs and chapters can not be avoided.

Interfacing with 'calling' (MAIN) programs is discussed in Chapter 2. Chapter 3 addresses the structure of the model. The physical processes are treated in Chapter 4. Chapter 5 gives a short functional description of the subroutines.

The Appendices 1-8 include listings of 'calling sections' of main programs, a list of variables, listings of subroutines with definitions of their input and output variables, and some further treatment of theoretical aspects.

1.5 Nomenclature, program language, and hardware

SAWAH:	Package of 18 subroutines, excluding DRSAWA. The principal subroutine is SUSAWA. The other routines are subject to SUSAWA, except SUWCHK which operates at the same hierarchical level.
DRSAWA:	A driver for SAWAH; written in standard FORTRAN 77. A task-specified subroutine. Interface between FORTRAN MAIN and SAWAH package. Only for FORTRAN users. DRSAWA is also presented in this report.
SUSAWA:	Principal subroutine of SAWAH. Organizing frame for initialization and rate calculations. Under CSMP, SUSAWA is called directly from the MAIN. Under FORTRAN, it is 'called' from DRSAWA.
MAIN:	User-defined main program. This can be a crop model or any other dynamic model that describes processes that are affected by soil water status, soil water fluxes, or floodwater depth.
CSMP:	Simulation language; Continuous System Modelling Program (IBM, 1975).
PCSMP:	PC version of CSMP. Developed for SARP (see below) by Jansen et al. (1988). PCSMP is based on IBM PC FORTRAN V2.0.

All subroutines, including DRSAWA, are written in standard fortran (ANSI FORTRAN 77). Certain expressions, although standard FORTRAN, have been avoided in SUSAWA and all underlying routines, in order to make the package fully compatible with PCSMP, which makes use of an older FORTRAN compiler (IBM PC FORTRAN V2.0; compiles only a subset of FORTRAN 77).

SAWAH has been tested under Microsoft FORTRAN V5.0 on IBM PC AT and compatibles, under Mac FORTRAN/020 V2.3 on Apple Macintosh II, under VAX FORTRAN V4.7 on a microVAX 3100/VMS V5.1-1, and under IBM PC FORTRAN V2.0 on IBM PC AT and compatibles. The driver DRSAWA should not be run under IBM PC FORTRAN V2.0; it is assumed that this FORTRAN version is only used in connection with a CSMP MAIN program which avoids DRSAWA.

DRSAWA makes use of input/output and interpolation facilities from the TTUTIL library. This library (Rappoldt and Van Kraalingen, 1990) must therefore always be linked when DRSAWA is used.

SAWAH has been successfully tested as a submodel under MAIN programs written in standard FORTRAN, CSMP, and PCSMP. Compatibility of DRSAWA and SAWAH with the different programming languages of the MAIN are indicated in Table 1.

Warning 1.	The driver DRSAWA makes use of FORTRAN 77 features that are not allowed in IBM PC FORTRAN V2.0. It is assumed that this older FORTRAN version is used only in connection with a CSMP main program.
Warning 2.	The driver DRSAWA should not be used in CSMP programs. Instead, SUSAWA is called directly.

Warning 3. There is one flaw in the compatibility of SAWAH with the above-mentioned FORTRAN compilers: the statement SAVE (conform to FORTRAN 77 standard) is compulsory in all subroutines under Mac FORTRAN, whereas it is illegal under IBM PC FORTRAN V2.0. PCSMP users must therefore remove the SAVE statement from all 18 subroutines.

Table 1. Compatibility of MAIN program and submodel (+ indicates full compatibility).

submodel	FORTRAN-77	MAIN program CSMP	PCSMP
DRSAWA + SAWAH	+	not[1] advised	not[2] possible
SAWAH without DRSAWA	not[3] advised	+	+ [4]

1) not advised, because CSMP does not combine well with integrating subroutines (DRSAWA) because of the timing of the integration step.

2) not possible, because FORTRAN V2.0 compiler under PCSMP cannot compile DRSAWA.

3) not advised, because user must adopt many I/O statements in MAIN which are already available in DRSAWA.

4) remove all SAVE statements.

SAWAH was developed in the framework of SARP, a project on Simulation and Systems Analysis for Rice Production, an international activity involving CABO, the Department of Theoretical Production Ecology of Wageningen Agricultural University, the International Rice Research Institute (IRRI, Philippines), and some twenty Asian national rice research centres.

1.6 Conceptual differences with some other models: SWATRE, WOFOST, SAHEL, RICEMOD, IRRIMOD, PADDYWATER

Several models are being used in Wageningen and in the 'IRRI-environment' to simulate the soil water balance, in connection with crop growth simulation. Some models widely used in Wageningen and by collaborating groups outside the rice growing countries are *SWATRE* (Belmans et al., 1983), *WOFOST* (Van Diepen et al., 1988) and *SAHEL* (Stroosnijder, 1982; Penning de Vries et al., 1989). In connection with rice research, notably at IRRI (Los Banos, Philippines), the best known models are incorporated in *RICEMOD* (Mc Mennamy and O'Toole, 1983), *IRRIMOD* (Angus and Zandstra, 1980), and *PADDYWATER* (Bolton and Zandstra, 1981).

Some conceptual differences between SAWAH and these other models will be indicated in this section. The user can then decide whether SAWAH provides a good alternative. This will not always be the case, depending on the availability of soil data and other specific data, and on the desired level of integration (i.e., on the questions to be answered by the modelling effort).

1.6.1 General characteristics; differential and integral models

Differential or Darcy-type models

On a scale from lower to higher integration level, SWATRE, RICEMOD and SAWAH are situated at the lower end. All three models are based on a mechanistic description of water flow, according to the Darcy equation or flux density equation. They are called *Darcy-type* models and can also be characterized as *'differential'* models. The use of the Darcy equation requires the characterization of the soil in terms of *hydraulic conductivity curve* (k-θ) and moisture characteristic (or *pF curve*), or hydraulic conductivity and *diffusivity* (D-θ). Conductivity and diffusivity are expressed as a function of *moisture content* or *pressure head*. The *pF*-curve relates moisture content to pressure head.

'Differential' models are generally preferable when the soil-water system does not reach a state of (near) equilibrium within the period of one time step of the MAIN (crop) model. Typically, transport between upper and lower boundaries of the system is then modelled by dividing the soil into a number of compartments or grid points, and solving the flux density equation and continuity equation (see Chapter 4) numerically. RICEMOD uses an explicit solution scheme, SWATRE is based on an implicit solution of the flow equation, and SAWAH uses both, depending on the moisture status of the soil (saturated/unsaturated).

Integral models

PADDYWATER, IRRIMOD, WOFOST and SAHEL are models of a higher integration level, and are therefore referred to as *'integral'* models. Emphasis is on the water balance of the root zone as a whole. In WOFOST, the modelled soil compartment gradually develops from shallow to deeper with increasing rooted depth. Most 'integral' models do not explicitly consider transport within the root zone. The transport properties of the soil are therefore not explicitly taken into account and the flux density equation is omitted. The use of such models is restricted to certain environmental conditions.

When the soil is treated as a single compartment, the continuity equation is applied to this single 'volume'. *The continuity equation in its finite difference form is then reduced to the overall soil water balance equation.* This allows the application of much simplified and faster computation algorithms as compared to 'differential' models.

Hybrids

SAHEL might be classified as a hybrid with both integral and differential characteristics, because it considers more than one single soil compartment, while still avoiding the use of the flow equation. This is accomplished by assuming an instantaneous (i.e., within the time resolution of the model) distribution of soil water. SAHEL therefore makes use of the *field capacity* concept. The assumptions limit the applicability of SAHEL to well-drained soils with high conductivity and relatively rapid redistribution.

Limiting soil water storage

In the integral models, soil water storage is just the 'closing term' in the overall water balance. All fluxes into and out of the system are quantified, and result in the net rate of water storage in the soil. Of course, all models have to set a limit to this storage, as exists in reality. To cope with this imperative, *deep drainage* results in SAHEL and WOFOST when supply exceeds the limited storage capacity. In PADDYWATER and IRRIMOD, *percolation rate* (deep drainage) is a constant. There, the problem of limited storage capacity in the soil is solved by allowing *surface storage*. When surface storage capacity is exceeded, *runoff* takes care of excess water.

Darcy-type models limit soil water storage implicitly by calculating the boundary fluxes as a function of internal and external conditions. They may or may not incorporate the concept of surface storage.

1.6.2 Boundary conditions

Differential models

In the 'differential' models SWATRE, RICEMOD and SAWAH, the fluxes at the upper and lower boundary of the system (*infiltration, evaporation, capillary rise and deep percolation*) are calculated from the physical soil properties and the conditions at the boundaries. In the field, where *fluxes* are rarely imposed as forcing functions, the boundary conditions actually imposed on the soil profile are usually *state variable* boundary conditions (pressure head, moisture content).

In RICEMOD, the lower boundary condition is the *moisture content* in the lower soil compartment; it is defined by a forcing function. In SAWAH and SWATRE, the *pressure head* at the lower boudary is defined externally. The latter two models have an option for free drainage: essentially a *constant force* condition, induced by gravity at the lower boundary. The resulting flux then is derived internally as a function of the *state* (i.e. moisture content).

Roughly, the same rules apply at the upper boundary: fully externally imposed flux conditions do rarely occur in the field. RICEMOD does not describe infiltration explicitly: ponded water and water in the first soil layer are merged into one single 'volume'. See Section 4.3. for details on SAWAH.

Integral models

In the integral models PADDYWATER and IRRIMOD, *flux values* at the system boundaries are defined externally: infiltration and deep percolation. For dry root zones, capillary supply from groundwater is then calculated on the basis of root zone water status, using an integral form of the flux equation, or using tabulated values for a given soil type and groundwater depth. The former solution is also applied in WOFOST. In SAHEL, infiltration is a fixed fraction of rainfall, and deep percolation is the excess water that cannot be stored in the profile (Subsection 1.6.1). Being designed for dry conditions with occasional showers, SAHEL must allow for temporal variations in deep percolation rate; and capillary rise is assumed absent. Under the 'paddyfield' conditions modelled by IRRIMOD and PADDYWATER, a constant percolation rate is more likely to occur and is therefore assumed.

1.6.3 Synchronization, time steps and computation time

Synchronization

PADDYWATER, IRRIMOD, SAHEL, WOFOST, SWATRE, and RICEMOD all have a so-called *'synchronous'* water balance: the time step at which the soil water balance is evaluated is equal to the time step in the calling (MAIN) crop model. For all but SWATRE, this is one day. SAWAH, in contrast, can be qualified as *'asynchronous'*.

Time step size

In SWATRE and SAWAH the time steps are small and variable. Numerical solution of the flow equation requires time steps in the order of 0.0001-0.1 day for most soil-water systems. The actual step size depends on soil type and compartment sizes. Small time steps do not combine very well with the large time steps commonly used in crop models. In SAWAH this problem is solved by running the water balance 'asynchronously' from the main. In SWATRE, alternatively, the small time steps required for flow calculations are imposed on the crop component of the model too.

It is at least peculiar that RICEMOD, as a Darcy based model, does not consider the 'small time step' requirement in solving the flow equation. This may be one reason for reported arithmetic computation failures.

CPU time

Darcy-type models are 'expensive' in computation; not only because all computations are repeated many times per day, but also because more statements are executed for each time step, than in *integral* models that avoid the use of the flux density equation.

1.6.4 Topography and groundwater

Toposequence

In IRRIMOD and PADDYWATER the modelled field is considered as part of a toposequence. This is necessary to incorporate lateral inflow from and outflow to adjacent fields. Maximum groundwater depth and percolation rate are defined as functions of topographic position. In IRRIMOD also lateral flow through the saturated subsoil is calculated, and groundwater changes as a result of deep percolation. In this sense it is 'more complete' than any of the other water models discussed here. It will be clear, however, that the relations required for modelling groundwater development are very site-specific and can only be obtained by full-scale field experiments.

SAWAH requires daily values of measured groundwater depth as input, thus taking groundwater level as the net result of all lateral and vertical flow processes. The same holds for WOFOST, RICEMOD and SWATRE. SAHEL, without adaptation, cannot deal with groundwater.

Long time series

Groundwater depth is largely determined by rainfall regime. When, however, a model must predict probabilities in future soil hydrology on the basis of available rainfall statistics, the lack of corresponding time series on groundwater may present a problem. Yet, in agro-ecological studies models are frequently applied in that way. An attractive option then seems to express groundwater depth as an *empirical* and *site-specific* function of rainfall. Obviously, establishing such relations requires long series of simultaneous observations of rainfall and groundwater depth. For most applications, no realistic results can be expected from linking soil water models with models that predict the groundwater level from flow processes.

1.6.5 Transpiration

In each model, transpiration is taken into account in a specific way. Usually, the water status of the rooted section of the profile determines the ratio of actual to potential transpiration. In all models discussed here, transpiration is the only direct link between soil and crop subsystems. In all of these, except SAWAH, transpiration is calculated from crop and soil conditions and parameters, and external variables.

In SAWAH, root water uptake is treated as an input, provided by the 'calling' crop model; it is externally calculated from soil water status and potential transpiration. The differences among models with regard to transpiration are therefore not discussed. Crop response to water availability is not an aspect of a water balance model in the strict sense.

1.6.6 Saturation

The models SWATRE, WOFOST, and SAHEL were not designed to include saturation of the soil profile with water. Not only is water flow in saturated soil constrained by a condition of zero flux-convergence and is the expression for the pressure potential different from unsaturated soil, a major difficulty is the transition from unsaturated to saturated flow. Therefore, in these models difficulties are encountered if groundwater rises into the profile, or when perched water (saturated soil overlying unsaturated soil) develops.

IRRIMOD, PADDYWATER and SAWAH, in contrast, were developed to take into account saturation. In the former two models, the whole profile is either identified as saturated or unsaturated. RICEMOD is not suitable for saturated conditions: it has no facility to prevent 'oversaturation' of soil compartments, a time step problem (the top compartment is an exception). This leads to computational errors and instability. Moreover, in RICEMOD the pressure term of the water potential is not considered when the soil becomes saturated. In the absence of groundwater, therefore, the percolation rate through a perched section may be grossly underestimated; in the presence of groundwater, the reverse will occur.

When groundwater or perched water result in partial saturation of the profile, the latter is separated in SAWAH into saturated and unsaturated domains. Subsequently, the flow equation is solved for both sections, taking into account the boundary conditions that each section imposes on the other, at their joint interface.

2 Interfacing SAWAH submodel with user-defined MAIN program

2.1 MAIN programs in FORTRAN, CSMP, and PCSMP

MAIN

The user-supplied MAIN program will be one that dynamically simulates processes in which the soil water balance plays a role. It will contain a section where state variables are initialized by a CALL to the submodel, followed by a time-loop in which dynamic processes are simulated, and by a terminal section where final calculations are performed.

Interface

The interface (i.e. the communication) between the MAIN program and the SAWAH submodel differs depending on the language in which the MAIN is written: either FORTRAN, or CSMP/PCSMP. Differences between CSMP and PCSMP are not relevant in the current context, so further reference will be made only to CSMP.
The differences in interface originate from differences between CSMP and FORTRAN with respect to input and output formats. Differences also result from the specific CSMP integration and initialization procedures. The submodel SAWAH *itself* - i.e. SUSAWA and underlying subroutines - is, however, independent of the MAIN program's language.

Driver

The connection MAIN-to-submodel is through a direct CALL to the subroutine SUSAWA for CSMP MAIN programs, while a FORTRAN MAIN makes use of a *program driver*, the subroutine DRSAWA. This driver can be regarded as a shell, wrapped around the SAWAH module. It performs various tasks, that will be discussed below. (The use of a driver under a FORTRAN MAIN is not compulsory, but it is strongly recommended because it avoids the necessity to expand the MAIN.) For the CSMP user no such driver is provided, because of CSMP's independent initialization and integration procedures. (Section 1.5, Table 1).

*A COMPLETE LISTING OF STATEMENTS THAT THE USER SHOULD ADOPT IN THE MAIN PROGRAM CAN BE FOUND IN APPENDICES 1 (***FORTRAN MAIN***) AND 2 (***CSMP MAIN***). A full example is worked out in Appendices 1-5 for a particular case.*

2.2 Integrals, rates and initial values

Interaction with environment

The soil subsystem interacts with its environment - be it a crop, a hydrological catchment area, a groundwater system or the atmosphere - through *fluxes* over the subsystem's boundaries. These depend on the system's *state*, its properties, and the external conditions. The variables describing the state of the soil system are *integrals*: the water contents of soil compartments, the total amount of water in the soil, the depth of the dry surface layer, and the depth of the water layer ponded on the surface.

Rates and states; arguments

The value of each integral is calculated from an *initial value* and a *rate-of-change*, by integrating rate over time. Under CSMP, the principal arguments exchanged between the user-supplied MAIN program and the submodel are rate variables (output to MAIN) and state variables (input from MAIN). This is because, in CSMP, integration of rates is performed by the MAIN program. Under FORTRAN, in contrast, integration of rates is performed by DRSAWA. The resulting states are then outputs from DRSAWA to the MAIN. They may or may not be used there for other purposes, at the user's choice.

Some variables other than rates or states are also exchanged between MAIN and submodel, for example the above-mentioned fluxes. A full description of SAWAH input and output is given in Sections 2.5 and 2.6 and in Appendices 1-5.

Types of integrals

Two groups of integrals are distinguished (Table 2):
(1) state variables that represent the physical state of the system;
(2) variables that keep track of cumulative values for overall balance purposes; these do not represent a true state of the system in the physical sense, but rather a history for which no 'physical memory' is available.

Table 2. Integrals in the SAWAH model

Group (1)	*integral (= state)*	*rate*	*initial condition*
(1) soil moisture content	WCLQT(10)	WCLCH(10)	WCLQTI(10)
depth of water layer on field	WLØQT	WLØCH	WLØQTI
depth of evaporation front	ZEQT	ZECH	ZEQTI
total water stored in profile	WCUM	WCUMCH	WCUMI
(2) contribution of surface-stored water to overall balance	WLØCO	SURREL	0.
contribution of stored soil water to overall balance	WCUMCO	PROREL	0.
cumulative fluxes	FLXCU(11)	FLXQT(11)	0.
cumulative rain **	RAINCU	RAIN	0.
cumulative runoff *	RNOFCU	RUNOF	0.
cumulative transpiration *	TRWCU	TRW	0.
cumulative evaporation *	EVSWCU	EVSW	0.
cumulative uprise over lower soil boundary	UPRICU	UPRISE	0.
cumulative drainage from drains *	DRAICU	DRAIN	0.

N.B.1 Figures in brackets indicate maximum array lengths

N.B.2 All variables in group (2) represent values accumulated since initialization.

N.B.3 Of the integrals, only WCLQT, ZEQT and WLØQT are inputs to SUSAWA. Together they define the state of the system, on the basis of which rates are calculated.

N.B.4 For the integral variables of group (2), and their corresponding rate variables, the convention has been maintained that 'gain terms' to the water balance have a positive value, whereas 'losses' are negative. Variables marked with ** are always positive or zero, those marked with * are always negative or zero. Following this convention, contributions from a 'reservoir' (WCUMCO and WLØCO for soil-stored and surface-stored water, respectively) are positive if the reservoir is being depleted, i.e. water is being released. Therefore, SURREL = -WLØCH * 1000. and PROREL = - WCUMCH (Factor 1000. to convert m to mm).

N.B.5 Fluxes over compartment interfaces (FLXQT) are defined positive if downward, negative if upward.

2.3 Integration, dummy integration, synchronization, DELT and DT

Some brief remarks about the procedures followed in rate calculations and integration are made here. For further details reference is made to Chapter 3.

Dummy integration

At each rate CALL (ITASK=2), SUSAWA receives the current values of state variables as inputs from the MAIN (under CSMP MAIN) or from the driver routine (under FORTRAN MAIN), and returns the rates-of-change as output. These, in fact, are *time-averaged* rates. They result from a series of so-called *dummy* integrations during a time interval of DELT, the time step employed in the MAIN. This is necessary because SAWAH simulates processes on an independent time axis, with time steps much smaller than DELT. In crop models usually time steps DELT of the order of one day are applied. Numerical solution of the water flow equation, however, requires time steps DT several orders of magnitude smaller (Section 3.6.3).

Synchronization

Such a construction, where the time steps of MAIN and submodel are different, is called *asynchronous* coupling. Asynchronous coupling is usually associated with *dummy* integrations in the submodel, as in the model presented here. In SAWAH the time interval DELT is divided into many smaller steps DT and for each DT one 'loop' is made. In this *internal time loop*, rates-of-change are calculated, and these rates are subsequently integrated to states. This is repeated until a full time-equivalent of DELT, e.g. one day, is completed. At that point, the sum of all time steps DT, passed since entering the subroutine SUSAWA from the MAIN, equals DELT. At the completion of DELT, the new values of the state variables are compared with the preceding values received from the MAIN, i.e. the values at the start of the submodel's time loop. The difference between the two values, divided by DELT, is the *time-averaged rate*. This rate is returned to the MAIN or to the program driver, where it is integrated over DELT.

Duplication

Evidently, the values of the state variables thus obtained after each DELT through integration by the MAIN or by the driver, are equal to those already attained in SUSAWA at the completion of the internal time loop. Integration in the MAIN therefore seems redundant. From a programming point of view, however, it is correct - especially in larger models with several submodels - to first perform all rate calculations, and then integrate all rates. This eliminates the possibility that values of different state variables, used in rate calculations, correspond to different *time* values.

SAWAH: a rate calculating module

Separation of *rate* and *integration* calls is not a specific feature of asynchronous coupling. The use of dummy integrations, in contrast, is. The user is not confronted with the asynchronous character of SAWAH: the whole submodel is treated like any other rate-calculating routine. For interfacing with the MAIN, the only distinction is that time step sizes in MAIN and submodel must be communicated. This information is transferred to the submodel through the argument list.

2.4 Interface structure: task-specified calls

2.4.1 FORTRAN MAIN

Tasks

Figure 1 illustrates the relation between the SAWAH submodel and a FORTRAN MAIN. Four types of CALLs to the program driver DRSAWA can be distinguished. Each type corresponds to a particular task to be performed by DRSAWA. The requested task is identified with the help of an integer variable (ITASK) that enters DRSAWA as an input argument at the CALL statement. The following tasks exist:

ITASK

(1) *initialization*: data reading; calculation of soil moisture content at specified pF-values for output to transpiration-calculating routine; initialization of water content, surface water depth, depth of evaporation front, and time-cumulated variables; opening of output files and writing of headers.
(2) calculation of *rates of change* of all state variables; calculation of fluxes; writing of output to files.
(3) *integration of rates*; water balance check.
(4) *terminal* calculations: finishing output; preparation for new initialization. The 'terminal' CALL is essential when SAWAH is used within a loop (e.g. in calibration procedures).

CALL frequency

The *initial* and *terminal* CALL occur only once per simulation run. The *rate* and *integration* CALL are repeated for each time step DELT in the dynamic section of the MAIN program. Figure 1, where the tasks are indicated in brackets, illustrates that the driver DRSAWA in turn makes a CALL to SUSAWA; it does so, however, only at the *initial* (1) and *rate* (2) CALL from the MAIN, not at the *integration* (3) and *terminal* (4) CALL. Hence, the SAWAH submodel proper (i.e. excluding the driver routine DRSAWA) is essentially a *rate-calculating* module, which also *initializes* state variables, but does not perform *integrations*.

The structured use of *multiple-task* routines in dynamic simulation models has been outlined by Van Kraalingen and Rappoldt (1989). It allows user-friendly programming and it is strongly recommended for designing modular simulation algorithms.

2.4.2 CSMP MAIN

Figure 2 presents the relation between the submodel and a CSMP MAIN program. The procedure is roughly the same as in FORTRAN, with two important differences:

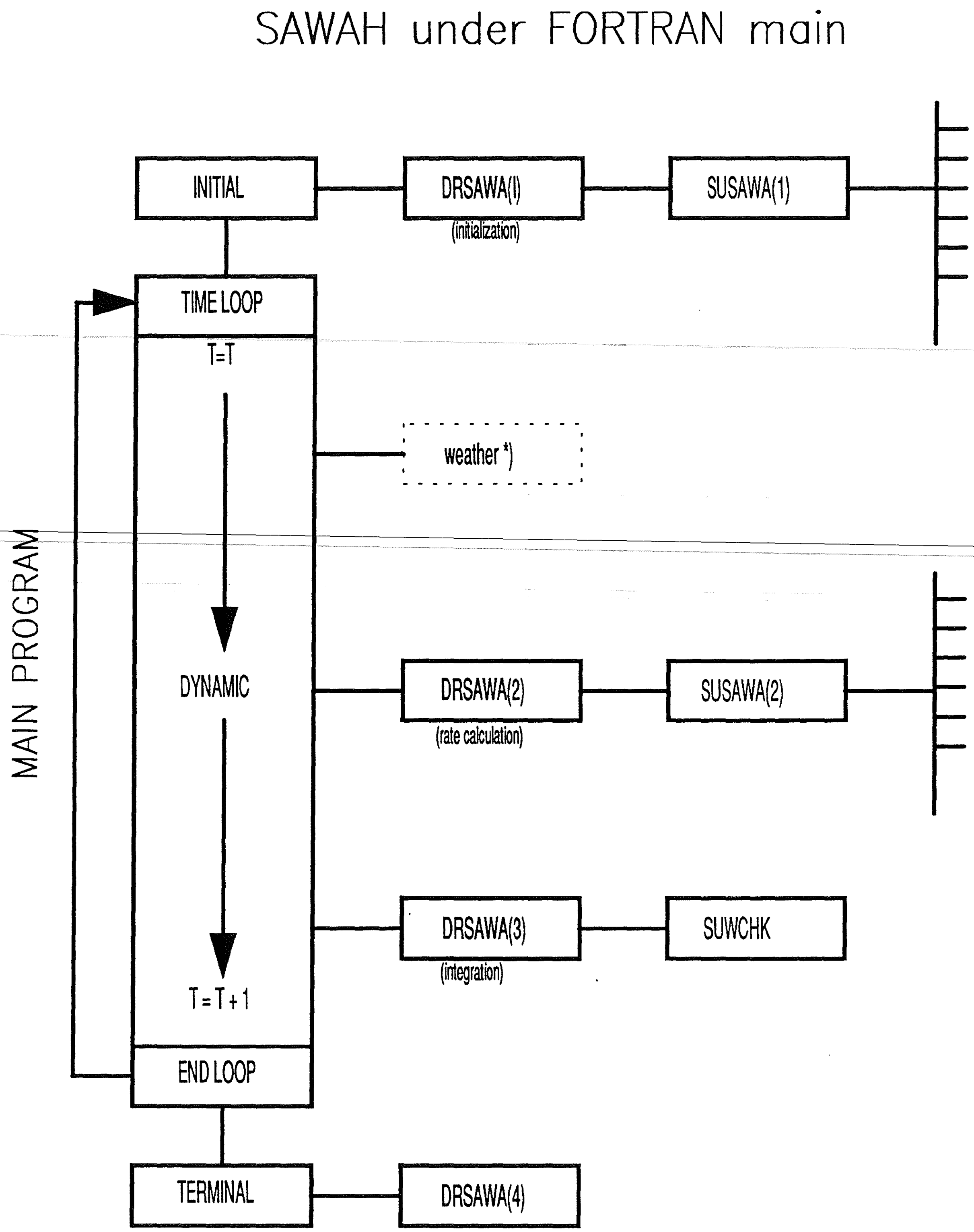


Figure 1. Interface between SAWAH-submodel and MAIN program for a MAIN written in FORTRAN.
* 'weather' refers to any group of user-defined routines that read groundwater depth and weather inputs and convert the weather data to potential soil evaporation on a daily basis.

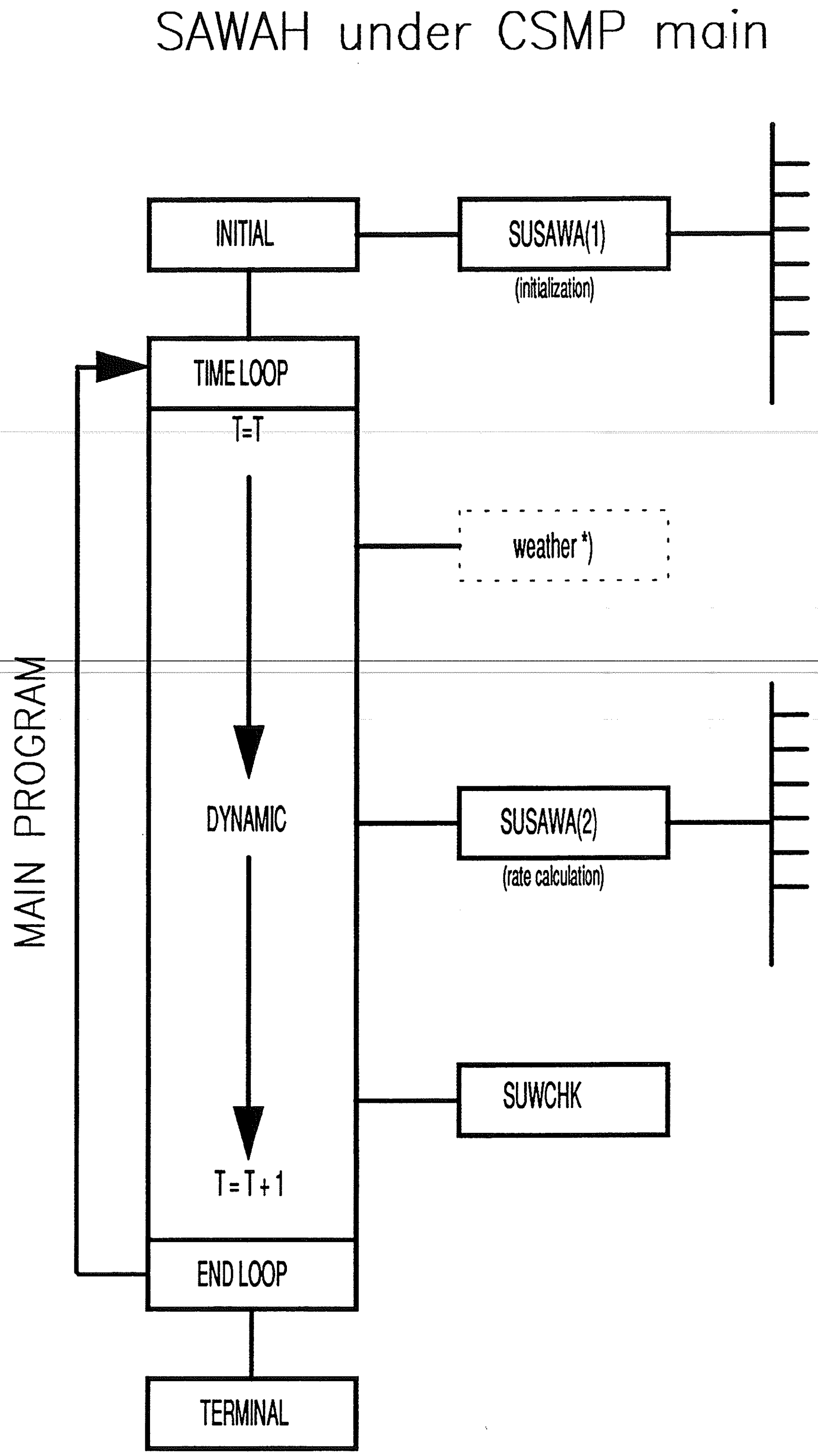


Figure 2. Interface between SAWAH submodel and MAIN program for a MAIN written in PCSMP or CSMP.
* 'weather' refers to any group of user-defined routines that read weather inputs and convert these to potential transpiration and potential soil evaporation on a daily basis.

1. *Only initialization* and *rate* CALLs are made. *Integration* and *terminal* CALLs are omitted: it is assumed that the CSMP-user prefers the specific CSMP integration and output preparation statements. Hence, these operations are performed by the MAIN program (Appendix 2).
2. A CSMP MAIN is not combined with the shell DRSAWA, but 'calls' SUSAWA directly.

A CALL to SUSAWA from a CSMP MAIN is identical to that from the FORTRAN driver DRSAWA, both at initialization and at rate-calculation (ITASK=1 and ITASK=2 in DRSAWA, respectively). *It is repeated that, below the level of DRSAWA, the submodel and its tasks and order of operations are identical under FORTRAN and CSMP MAIN programs*, apart from the SAVE statements (Section 1.5).

2.5 Inputs to the submodel

The presentation by the user of input variables transferred from the MAIN to the submodel depends on whether the MAIN is written in CSMP or in FORTRAN. The driver DRSAWA (under FORTRAN MAIN) reads most of the input variables from an input file and these, consequently, need not be entered through the argument list of CALL DRSAWA in the MAIN. Moreover, the driver performs the integration, so the state variables (integrals) need not be communicated from MAIN to submodel as under a CSMP MAIN, but from driver to submodel, instead. (They are also given as output from driver to MAIN.)

SAWAH under a CSMP MAIN uses no input file or driver; as a consequence, the presentation of input variables is different. See also Appendices 1-4.

2.5.1 Inputs under a FORTRAN main

A number of input variables are dynamic and are therefore communicated from MAIN to driver at each time step DELT; this is through the *argument list* of CALL DRSAWA. Other inputs represent *static system characteristics* and *option switches*, and are read from an *input file*. Both types will be treated in the following. More details are given in Appendices 3-4.

No COMMON blocks exist between MAIN and DRSAWA. The COMMON variables in DRSAWA are shared only with underlying subroutines (SUSAWA and lower levels).

2.5.1.1 Inputs via argument list to DRSAWA

acronym	*type*	*description*	*units*
		run control variables	
ITASK	I4	action task of subroutine (Subsection 2.4.1)	-
IUNIT	I4	unit number used for input/output file. (warning: SAWAH will use the unit number IUNIT, and IUNIT+2 till IUNIT+6)	
IUNLOG	I4	unit number used for logfile = 0, no logfile is used > 0, error messages are written to logfile	-
IOUT	I4	output control parameter, indicating which outputs are requested; integer consisting of five digits, each of them to be set to 1 for output, 0 for no output. The last digit refers to output file WATER1.OUT, the one-but-last to WATER2.OUT, etc. For definition of output files see Subsection 2.6.1	-
		Examples : 10001 --> WATER5.OUT and WATER1.OUT 00011 --> WATER2.OUT and WATER1.OUT 00110 --> WATER3.OUT and WATER2.OUT 11111 --> All 5 output files are produced When the last digit is 2, a table of soil water contents at saturation, field capacity, wilting point, and 'air dry' is added to the header of WATER1.OUT. For full output, IOUT should be assigned the value 11112	
FILIN	C*	name of input file (soil data, option switches, discretization variables, and groundwater levels)	
		time control variables	
TIME	R4	simulated time	d
DAY	R4	Julian day number	
DELT	R4	time step in MAIN program; period over which average rates are calculated	d
		dynamic external variables	
EVSC	R4	potential soil evaporation rate	mm d^{-1}
RAIN	R4	rainfall or irrigation rate	mm d^{-1}
TRWL()	R4	actual rate of water extraction by roots, per soil compartment	mm d^{-1}

(brackets in first column indicate arrays; maximum size 10 elements)

2.5.1.2 Inputs from input file to DRSAWA

Option switches, soil characteristics and initial values of state variables are read from an input file. The name of this file (stored in the variable FILIN) is specified by the user in the argument list of DRSAWA (subsection 2.5.1.1). All variables in the input file, and their units, are defined in Appendix 4.

2.5.2 Inputs under a CSMP MAIN

Under a CSMP MAIN, the inputs to the submodel are the input arguments to the subroutine SUSAWA, and the variables included in labelled COMMON blocks. The input arguments can be classified in several categories, which differ from those listed in Subsection 2.5.1.1, because *no input file* is used under CSMP.

acronym	*type*	*description*	*units*
		run control variables	
ITASK	I4	action task of subroutine (Subsection 2.4.2)	-
IUNLOG	I4	unit number for logfile	-
		= 0, no logfile is used	
		> 0, error messages are written to logfile	
		time control variables	
DELT	R4	time step in MAIN program	d
DTFX	R4	fixed time step in submodel	d
DTMX1	R4	maximum time step in submodel	d
DTMIN	R4	minimum time step in submodel (warning: See Subsection 3.6.3)	d
		spatial discretization	
NL		number of soil compartments	-
TKL()		thickness of soil compartments	m
IDRAIN	I4	index of compartment with artificial drains	-
		system properties	
CSA	R4	soil evaporation constant (Section 4.3.3; App.7)	$cm^2\ d^{-1}$
CSB	R4	soil evaporation constant (Section 4.3.3; App.7)	-
CSC2	R4	soil evaporation constant (Section 4.3.3; App.7)	$cm^2\ d^{-1}$
TYL()	R4	soil type per compartment; used in reading data from standard soil tables;	-
		the hydraulic properties of the soil are read from standard tables as a function of TYL	-
WLØMX	R4	maximum surface storage capacity	m
		dynamic external variables	
EVSC	R4	potential evaporation rate	$mm\ d^{-1}$
RAIN	R4	rainfall or irrigation rate	$mm\ d^{-1}$
TRWL()	R4	rate of water extraction by roots, per compartment	$mm\ d^{-1}$
ZW	R4	depth of groundwater table (ref. soil surface) In the initial CALL, ZW influences the equilibrium water contents.	m
		dynamic internal variables	
WCLQT()	R4	volumetric water content per compartment	-
WLØQT	R4	thickness of water layer on surface	m
ZEQT	R4	depth of evaporation front	m

Input variables in labelled COMMON blocks

acronym	*type*	*description*	*block name*	*units*
SWIT1- -SWIT9	I4	option switches (Section 3.9)	SWITCH	-
WCAD()	R4	volumetric water content at air dry	VOLWAT	-
WCFC()	R4	volumetric water content at field capacity	VOLWAT	-
WCST()	R4	volumetric water content at saturation	VOLWAT	-
WCWP()	R4	volumetric water content at wilting point	VOLWAT	-
KMSMX()	R4	parameter in Rijtema *k*(*h*) relation	HYDCON	cm
KMSA1()	R4	exponent in Rijtema *k*(*h*) relation	HYDCON	cm^{-1}
KMSA2()	R4	coefficient in Rijtema *k*(*h*) relation	HYDCON	$cm^{2.4} d^{-1}$
KST()	R4	hydraulic conductivity at saturation	HYDCON	$cm\ d^{-1}$
MSWCA()	R4	coefficient moisture characteristic	PFCURV	-

(brackets refer in first column indicate arrays; maximum size 10 elements)

2.6 Outputs from the submodel

The differences between a FORTRAN and a CSMP MAIN also affect the list of output variables, communicated between MAIN and submodel. Under CSMP, all rates are specified as output arguments from submodel to MAIN program. Under FORTRAN this is not necessary, because integration of rates is effectuated in the driver.

2.6.1 Outputs under a FORTRAN MAIN

A distinction is made between outputs returned by the driver to the MAIN in the *argument list* of CALL DRSAWA, and outputs written to *output files*.

2.6.1.1 File output from DRSAWA

Example output files are given in Appendix 5.
A maximum of five output files may be produced, depending on the value of the switch variable IOUT (Subsection 2.5.1.1). The full output, i.e. files **WATER1-5.OUT**, is generated if this variable is assigned a value 11112. If IUNLOG > 0, also a logfile will be produced containing error messages.

WATER1.OUT gives the values of the state variables:

acronym	*type*	*description*	*units*
WCLQT()	R4	volumetric water content, per compartment	-
WLØ	R4	thickness of water layer on surface	mm
ZEQT	R4	depth of evaporation front	mm

WATER1.OUT also summarizes a number of soil characteristics, relevant to crop behaviour, if the last digit in IOUT is set to 2:

acronym	*type*	*description*	*units*
TKL()	R4	compartment thickness	m
WCAD()	R4	volumetric water content at air dry	-
WCWP()	R4	volumetric water content at wilting point	-

WCFC()	R4	volumetric water content at field capacity	-
WCST()	R4	volumetric water content at saturation	-

WATER2.OUT gives the distribution of daily transpiration:

acronym	*type*	*description*	*units*
TRWL()	R4	water extraction rate by roots per compartment	mm d^{-1}

WATER3.OUT gives the daily water balance terms: the rates at which water enters (+) or leaves (-) the system at its boundaries:

acronym	*type*	*description*	*units*
RAIN	R4	rainfall or irrigation	mm d^{-1}
RUNOF	R4	runoff	mm d^{-1}
TRW	R4	transpiration	mm d^{-1}
EVSW	R4	soil evaporation	mm d^{-1}
EVSC	R4	potential soil evaporation (not a water balance term; included only for reference)	mm d^{-1}
UPRISE	R4	capillary rise rate at lower profile boundary	mm d^{-1}
DRAIQT	R4	drainage (by artificial drains)	mm d^{-1}
PROREL	R4	water release from profile	mm d^{-1}
SURREL	R4	water release from surface storage	mm d^{-1}

WATER4.OUT gives the cumulative water balance terms: quantities that entered or left the system since the start of simulation:

acronym	*type*	*description*	*units*
RAINCU	4	rainfall	mm
RNOFCU	R4	runoff	mm
TRWCU	R4	transpiration	mm
EVSWCU	R4	soil evaporation	mm
UPRICU	R4	drainage over lower profile boundary	mm
DRAICU	R4	drainage from artifical drains	mm
WCUMCO	R4	contribution of stored soil water to overall water balance	mm
WLØCO	R4	contribution of surface-stored water to overall water balance	mm

WATER5.OUT contains the day-averaged fluxes over the compartment interfaces:

acronym	*type*	*description*	*units*
FLXQT()	R4	flux over lower boundary of each compartment	mm d^{-1}

(brackets in first column indicate arrays; maximum size 10 elements except FLXQT(11))

2.6.1.2 Output arguments from DRSAWA

acronym	*type*	*description*	*units*
		spatial discretization	
INLAY	I4	number of soil compartments (from input file)	-
TKL	R4	thickness of soil compartments	m
		constants (soil characteristics)	
		(identical to WCAD, WCWP, WCFC, WCST)	

WCADX	R4	volumetric water content at 'air dry'	-
WCWPX	R4	volumetric water content at wilting point	-
WCFCX	R4	volumetric water content at field capacity; this variable is not used in SAWAH; it is calculated for possible use in the MAIN and is defined here as the volumetric water content at 100 hPa matric suction.	-
WCSTX	R4	volumetric water content at saturation	-
		dynamic internal variables	
EVSW	R4	actual soil evaporation rate	mm d^{-1}
FLXQT()	R4	flux over compartment interfaces	mm d^{-1}
WCLQT()	R4	volumetric moisture content	-
WLØQT	R4	thickness surface water layer	m
ZEQT	R4	depth of evaporation front	m
DRAICU	R4	cumulative drainage from artificial drains	mm
UPRICU	R4	cumulative flux over lower soil boundary	mm
EVSWCU	R4	cumulative soil evaporation	mm
RAINCU	R4	cumulative rainfall	mm
RNOFCU	R4	cumulative runoff	mm
TRWCU	R4	cumulative transpiration	mm
FLXCU()	R4	cumulative flux at compartment interfaces	mm
WCUMCO	R4	contribution of stored soil water to overall water balance	mm
WLØCO	R4	contribution of surface-stored water to overall water balance	mm
PRHEAD()	R4	pressure head at compartment centres	hPa
DTAV	R4	average DT (averaged over DELT)	d

(brackets in the first column indicate arrays; maximum size 10 elements except FLXQT(11) and FLXCU(11)

2.6.2 Outputs under a CSMP MAIN

No output files are produced, except a logfile if IUNLOG > 0. The following outputs are returned from the submodel to the MAIN as arguments of CALL SUSAWA:

acronym	*type*	*description*	*units*
		spatial discretization	
ZLT	R4	total depth of soil profile	m
		time control variables	
DTAV	R4	average DT (averaged over DELT)	
		dynamic internal variables	
UPRISE	R4	rate of capillary rise at lower soil profile boundary	mm d^{-1}
DRAIQT	R4	rate of water loss through artificial drains	mm d^{-1}
EVSW	R4	actual soil evaporation rate	mm d^{-1}
FLXQT()	R4	fluxes over compartment interfaces, averaged over DELT	mm d^{-1}
RUNOF	R4	rate of water loss by runoff	mm d^{-1}

WCLCH()	R4	rate of change of volumetric water content, per soil compartment	d^{-1}
WCLEQI()	R4	volumetric water content in equilibrium with groundwater depth (per soil compartment)	-
WCUMCH	R4	rate of change of total profile-stored water	mm d^{-1}
WLØCH	R4	rate of change of thickness of surface water layer	m d^{-1}
ZECH	R4	rate of change of depth of evaporation front	m d^{-1}
PRHEAD()	R4	pressure head at compartment centers	hPa

(brackets in first column indicate array; maximum size 10 elements except FLXQT(11))

2.7 Error messages

Message number	*Name subroutine*	*Condition*	*Possible cause*
~~1~~	~~SUMSKM~~	~~MS < 0. or > 1.E8~~	Wrong calculation of MS
2	SUSLIN	WCLQTI(I) < WCAD(I) or > WCST(I)	Wrong initialization of water contents; check measurements, units
3	SUWCMS	WCLQT(I) < WCAD(I) or > WCST(I)	Wrong calculation of water content in a soil compartment
4	SUWCMS	MS < 0. or > 1.E8	Wrong calculation of MS

The above messages have been included to facilitate error detection following user-made adaptations in the program source. Such adaptations are required if the user prefers to apply other $k(h)$ or $\theta(h)$ curves than those provided in SAWAH (See Subsections 4.2.3.2 and 4.2.3.3).
These messages are generated by the subroutine SUERR and are sent to both screen and logfile (if specified by input IUNLOG).

Other messages may be generated by the routine ERROR from the TTUTIL utility library (Rappoldt and Van Kraalingen, 1990). Messages by ERROR are sent to the screen and are self-explanatory. They include comments on illegal switch combinations, illegal initialization, and illegal soil type numbers.

A special set of error messages is generated by the routine SUWCHK, that performs a water balance check. These messages are sent both to screen and logfile, if specified (Section 2.8).

2.8 Water balance check

The water balance check routine SUWCHK is automatically included in the driver DRSAWA for FORTRAN users. It compares the following two variables:
CKWIN: the change in water content of the entire system since initialization, i.e. a *difference* between *states*.
CKWFL: the sum of time-integrated *fluxes* over the system boundaries.
SUWCHK sends an error message to the screen and to a logfile, if two conditions occur simultaneously:

(1) the difference between CKWIN and CKWFL exceeds 1 mm;
(2) the relative error 2(CKWIN-CKWFL)/(CKWIN+CKWFL) exceeds 1 %.

For CSMP users - using no driver - it is recommended to include a CALL to SUWCHK in the MAIN program (Appendix 2, and Figures 1 and 2).

3 Program structure of SAWAH

3.1 Principles

The guiding principles applied in designing the model structure were:

1. separate CALLs are made for *initialization*, *rate calculations*, and *integration*;
2. some functions that users might want to change, such as retention curves, conductivity curves, and matric flux potential curves are readily accessible as separate subroutines;
3. options are offered to the user through *option switches*;
4. water uptake for transpiration (the 'sink term' in the flow equation) is not calculated in the submodel, but is to be supplied as an input argument;
5. incoming and outgoing arguments to the submodel pass through a 'conversion sieve', to facilitate converting their units.

3.2 Discretization of time and space

The transport equations for water are solved numerically after linearization. This involves spatial discretization and time discretization into *finite steps*. Time steps are discussed in the Sections 2.3, 3.3, 3.6, and 3.8.

Space steps (i.e. soil compartment sizes) are defined in the array TKL. Its maximum size, i.e. the maximum number of soil compartments, is 10. The size of the compartments is defined in the input file (variable file name FILIN) when the MAIN is in FORTRAN (Appendix 4), and in the MAIN itself when the latter is in CSMP (Subsection 2.5.2.).

3.3 SUSAWA structure

Time loop: DELT and DT

The subroutine SUSAWA is the module's framework. It is 'called' from the MAIN (CSMP) or from the driver DRSAWA (FORTRAN), once per time step DELT (Chapter 2).

SUSAWA 'loops' with time steps DT, much smaller than DELT, the time step of the MAIN. It keeps track of time by accumulating the steps DT, until one DELT is completed. At each iteration, it calls the underlying subroutines. At the completion of a full time step DELT, execution is returned to the MAIN.

SUSAWA operations

Figure 3 illustrates the structure of the program. The top section (INITIAL, SUSAWA(1)) is executed once per run: at the initialization phase. The lower section (DYNAMIC, SUSAWA(2)) is executed at least once per DELT, but the shaded routines are executed many more times (once per DT). Six groups of operations can be distinguished, executed in the following order:

	Section in this report
(1) conversion of units;	3.4
(2) addition/subtraction of gain/loss terms;	3.5
*(3) *** time loop ****	
flux calculations and dummy integration;	3.6
- identification of saturated and unsaturated sections	3.6.1
- calculation of fluxes in respective sections	3.6.1
- selection of fluxes at section boundaries	3.6.2
- assessment of flux divergence	3.6.3
- dummy integration	3.6.3
**** end of loop ****	
(4) assessment of change in depth of evaporation front;	3.7
(5) conversion of units;	3.4
(6) assessment of average rates-of-change of state variables	3.8

Section 3.9 lists the option switches. An overview of all subroutines is given in Chapter 5.

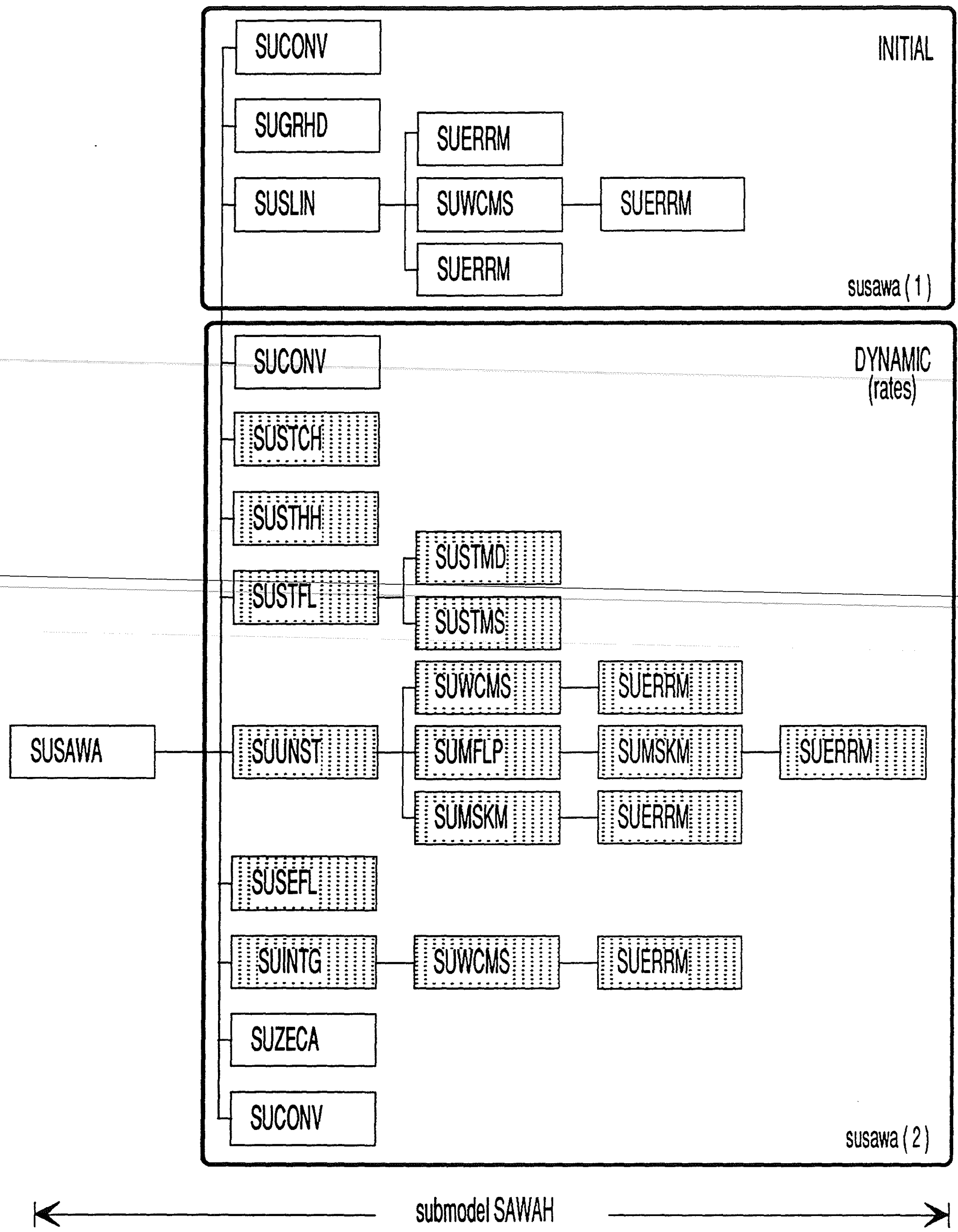


Figure 3. Structural diagram of the submodel. The two blocks 'INITIAL' and 'DYNAMIC' perform initialization of state variables and rate calculations, respectively. Integration of rates is in the driver or in the MAIN (not in this figure). Subroutines depicted in shaded rectangles are 'called' at each DT, the other subroutines are 'called' once per DELT only.
units used in the MAIN, SUCONV has to be adapted to specific needs by changing the conversion factors F1 and F2.

3.4 Conversion of units

SUCONV

The units of input arguments, as defined by the user in his MAIN, are converted to local units in the routine SUCONV. The reverse conversion is performed before leaving SUSAWA, again in SUCONV. In dependence of the

Units

In SUSAWA, and in all routines 'called' by SUSAWA, *lengths* (depths, heads, etc) are in *cm* and *time* is in *days*; this applies to all statements between the two SUCONV CALLs. The version of SUCONV published in this report applies to a MAIN with *lengths* in *m* (depth of groundwater, thickness of soil layers, depth of ponded water), *time* in *days*, and *fluxes* in *mm/d* (rain, evaporation, transpiration, etc.).

3.5 Addition/subtraction of source/sink terms: impulse functions

Impulses

In reality, all the water balance terms (rainfall, drainage, evaporation, etc.) are *flux variables* across the system boundaries. Some of them, however, can be treated as an *impulse*, the size of which equals the integral of flux over time, during a period of DELT. For example, when DELT is one day, daily rainfall can be considered a quantity, rather than a flux. Such a quantity can be instantaneously added/subtracted to/from the quantities in the system. Obviously, this is only possible for those balance terms whose DELT-cumulated value is directly available, that is, without simulating its dynamics during DELT. So, the quantity gained or lost during the coming DELT is estimated directly from current state of the system and external conditions. This applies to *evaporation*, *transpiration*, and *rainfall*; not to *drainage*, *capillary rise*, *runoff*, *infiltration*, *exfiltration* and *artificial drainage*. (see Section 4.3.)

Evaporation

The evaporative water loss from soil to atmosphere, calculated in SAWAH, is subtracted once per DELT from the quantity of water present in the top soil compartment. Its value is determined once per DELT from the current system status and external conditions, assuming that, within that period, no feedback mechanisms affect the validity of this linearization. If a layer of water is present on the surface, the evaporative loss is first subtracted from this layer, until it is depleted. The remainder of the demand is then withdrawn from the top soil compartment.

Transpiration

In the user-defined MAIN, actual transpiration should be specified per compartment and per DELT. (Usually it will be calculated in the MAIN from the status of soil and crop.) It should be provided *before* the CALL SUSAWA statement. SUSAWA then subtracts the specified quantities *instantaneously* (once per DELT) from the quantities of water in the corresponding compartments.

Rain

Rain is 'added' as a water layer on the surface once per DELT, before flux calculations start. No distinction is made between rainfall and irrigation.

3.6 Flux calculations and dummy integration

As a result of the processes described by impulse functions and of groundwater movement, water starts flowing. The *local time loop*, i.e. the loop in SUSAWA that divides the period of DELT in many small steps DT, is started: the flow rates and the rates-of-change of the state variables (*water content*, and *thickness of surface water layer*) are calculated; and the rates are integrated. These steps are elaborated below. For a review of state variables and their associated rates, see Chapter 2.

3.6.1 Identification of saturated and unsaturated sections

Profile sections

To calculate fluxes, the soil profile is divided into saturated and non-saturated sections. These sections are treated by different solution schemes. Identification and subsequent solution are repeated each DT, because the situation may change in the course of time. One or more saturated sections may exist within the profile. Each section may consist of one or more compartments. Obviously, complete absence of saturated sections is also possible. An example profile is given in Figure 5 (Subsection 4.2.2).

Labelling

In the subroutine SUSTCH the water contents of all compartments are examined to decide whether or not a compartment is saturated. If volumetric moisture content in a given compartment is within 0.01 of the saturated water content WCST, the compartment is considered saturated. Compartments are either fully saturated or unsaturated.

In SUSTCH the saturated sets and their compartments are 'labelled' : a *set number i* and *internal layer number j* is assigned to each saturated soil compartment. The indices of saturated compartments are stored in the two-dimensional array INXSAT. INXSAT(i,j) is the integer compartment index of the j^{th} layer in the i^{th} saturated section, counting both from the top downward. INXSAT is used by the routine SUSTHH to calculate the total difference in hydraulic head over each set. Subsequently, in SUSTFL and SUUNST the first estimates of the fluxes through the saturated and unsaturated sections of the profile are calculated, respectively. These may have to be adjusted by SUSEFL.

Fluxes within sections: differences saturated vs unsaturated

In both saturated and non-saturated sections, fluxes are calculated on the basis of the *Darcy* equation (Subsections 4.2.2 and 4.2.3). The numerical procedure, however, is different. In the *unsaturated* sections, fluxes over compartment interfaces are calculated by an *explicit* scheme, using the local gradient in hydraulic head between two adjacent compartments, and the local hydraulic conductivity. (In fact, the local gradient of *matric flux potential* is used; see Subsection 4.2.3). This principle can be applied in unsaturated soil, because *pressure head* follows directly from *moisture content*. For the saturated sections, in contrast, pressure head in each individual compartment cannot be assessed directly. The entire saturated set must be taken into account to calculate local pressure head, its gradients, and the resulting fluxes by an *implicit* solution (Chapter 4).

3.6.2 State conditions and fluxes at the saturated-unsaturated interfaces; choice of fluxes

Interface pressure head

In SAWAH, the pressure head *h* at any interface between a saturated and a non-saturated compartment is defined to be *zero*. Such interfaces cannot be present at locations other than compartment boundaries. Hence, due to these assumptions, the thickness of saturated sections cannot change *gradually*, but only in *discrete* steps.

Fluxes

Applying two independent solution schemes for saturated and unsaturated flow, respectively, causes a problem at the *saturated-unsaturated boundary*. At both sides of this boundary, the flux is calculated from the gradient in hydraulic head. The definition of the pressure component of this water potential, however, changes at the interface (Subsection 4.2.1). The potential at the interface is used as a boundary condition to both sections, saturated and unsaturated.

The consequence of this procedure is that at the interface two different flux values are calculated. There is no reason why they should come out equal. In reality, however, the fluxes should be identical because the interface has no capacity. Hence, a choice must be made. For example, the flux *within* the saturated section might restrict the flow *into* or out of the adjacent unsaturated section. The selection is made in the routine SUSEFL. The set of rules that is the basis for this procedure is complex. It must allow compartments to shift from the 'saturated' to the 'unsaturated' state, should the dynamic boundary conditions induce this, and *vice versa*. Saturated-unsaturated oscillations as a numerical artefact must be avoided in this procedure. For more details see Subsection 4.2.4 and Appendix 8.

3.6.3 Rates, dummy integration, time steps

Rates

From the fluxes over all compartment interfaces, rates-of-change of water content in each compartment are calculated in SUINTG, making use of the *continuity equation* (Subsection 4.2.3.6). These rates are then integrated in SUINTG.

Local time and dummy integration

In SUINTG, the rectangular *Euler integration* is performed over time steps DT. After each time step DT, SUSAWA augments the 'local time' TIMTOT and checks whether a time period DELT (the time step in the MAIN) has been completed. If this is not the case, the time loop in SUSAWA performs another iteration: again, saturated compartments are identified, fluxes and rates are calculated, and rates are integrated. This integration step is called *dummy integration* (Section 3.8).

Time step

DT can have a *variable* or a *fixed* value. The user selects either of the two options, by setting the switch SWIT5 to the appropriate value (Section 3.9). For the *variable step* option, DT is internally assessed on the basis of the system's current time constants. A maximum permitted time step, DTMX1, must then be defined by the user, as well as a minimum step, DTMIN. In general, the variable step option is preferable.

Fixed step

In the fixed step option, SUINTG uses the value of the parameter DTFX as time step. However, even if the user defines a fixed step, the actual value of DT can still be different from DTFX. DT is *smaller than or equal to* DTFX. Six conditions are examined in SUINTG, and DT is decreased to a value lower than DTFX, if:

(a) DTFX exceeds the time period needed to saturate a compartment, on the basis of current rates;
(b) DTFX exceeds the time period required to remove all the water from a compartment, down to 'air-dryness';
(c) DTFX exceeds the time period required to remove the ponded water layer on the surface;
(d) DTFX exceeds the time period required to reach equilibrium with the groundwater; this condition is applied only if upward flow through the lower profile boundary occurs. It prevents oscillations when groundwater is within the profile;
(e) DTFX is erroneously defined larger than DTMX1;
(f) DTFX exceeds the time period remaining until DELT is completed.

Variable step

If the *variable step* option is selected, DT is assessed internally. First, the maximum step size that satisfies all of the above six conditions is determined. Subsequently, a loop reduces the tentative value of DT by 50% each time, until a value is attained that allows maintaining the current rates-of-change during that period without inversion of the *hydraulic head gradient*, over *any* of the interfaces. The value then obtained is the actual DT. SAWAH ignores this condition of *no gradient inversion* when DT becomes smaller than DTMIN.

Warning: too high DTMIN values lead to oscillations. Roughly, DTMIN should be around 0.01 d for low-conductivity soils (KST $\leq$ 10 cm d^{-1}), and down to 0.0005 for highly permeable soils (KST > 100 cm d^{-1}) if compartments of about 0.1 m thickness are used. Critical examination of simulated moisture contents and, possibly, adjustment of DTMIN are recommended.

3.7 Evaporation front

Drying of the topsoil results in a steep gradient in moisture content and pressure head near the soil surface. To simulate this numerically, a fine discretization of space and time is required. That, however, carries a heavy price in terms of the permitted step size DT. Yet, reducing potential soil evaporation to actual evaporation is an important process in reality, that can not be ignored in water balance modelling. In SAWAH, therefore, soil evaporation is calculated by an analytical approximation. This avoids solving the flow equation for the near-surface region. The approximation is based on the concept of the *evaporation front*. The depth of this front is expressed in the variable ZE. Once per DELT, the new value of ZE is calculated in SUZECA. The difference with the previous value, divided by DELT, is returned to the MAIN (or driver) as a rate for integration.

3.8 Average rates of change and integration

When the *time loop* in SUSAWA has covered a period equal to DELT, the final result of the dummy integrations in SUINTG represents the new state of the system (WCL, WLO, ZE). This state is compared to its previous state, i.e. in the MAIN or driver just before the CALL SUSAWA statement (WCLQT, WLOQT, ZEQT). The difference between the two values is divided by DELT, yielding for each state the *average rate* of change, which is returned to the MAIN (WCLCH, WLOCH, ZECH). In the MAIN or in the program driver - Section 2.2 - these rates are integrated in a single step by Euler integration. Hence, integration occurs twice, but at different levels; therefore the term *dummy integration* for the lower level integrations.

3.9 Switches and option switches

SAWAH contains a number of switches indicated by SWIT1 to SWIT9. All switches are *integer* variables. Some are user-defined option switches. These are defined in the input file to DRSAWA (FORTRAN MAIN) or in the MAIN program (CSMP MAIN) and are transferred to local levels through the COMMON block /SWITCH/. Others are internally adjusted input arguments to multiple-function subroutines. See also inputs, Section 2.5.

acronym *description*

SWIT1 Input argument to SUSAWA; not user-defined
=1 initial call
=2 rate call

SWIT2 Input argument to SUCONV; not user-defined
=1 conversion from MAIN to local units
=2 conversion from local to MAIN units

SWIT3 Input (COMMON) to SUMSKM; user-defined; selects conductivity function; (Subsection 4.2.3.2)
=1 simple Rijtema expression (exponential)
=2 full Rijtema expression (two-branch)
=3 Van Genuchten expression
=4 power function
=5 user-defined function

This switch is also an input (COMMON) to SUMFLP; its value is the same as in SUMSKM, and ensures the consistency between operations in the two subroutines. In SUMFLP:
=1 analytical integral
=2 numerical integral (Gauss)
=3 numerical integral (Gauss)
=4 analytical integral
=5 numerical integral (Gauss)

SWIT4 Input argument to SUWCMS; not user-defined;
=1 calculation of suction from water content
=2 calculation of water content from suction

SWIT5 Input (COMMON) to SUINTG; user-defined; time step choice;
=1 variable time step
=2 fixed time step

SWIT6 In MAIN or DRSAWA; user-defined; selects type of initialization of moisture content;
=1 initial moisture contents in hydrostatic equilibrium with groundwater level;
=2 initial moisture contents given as measured values
=3 initial moisture contents at wilting point (pF 4.2)

SWIT7 Input (COMMON) to SUUNST; user-defined lower boundary condition;
=1 pressure head condition (calculated from groundwater level)
=2 free drainage

SWIT8 Input (COMMON) to SUWCMS; user-defined; selects moisture characteristic; (Subsection 4.2.3.3)
=1 Driessen relation
=2 Van Genuchten relation
=3 linear interpolation on log scale (user-defined table)
=4 user-defined function

SWIT9 In MAIN or DRSAWA; user-defined; soil parametrization by soil type number or by quantified physical parameters in input file
=1 by quantified parameters
=2 by soil type number only

N.B. For SWIT9=2, SWIT3 and SWIT8 can only occur in restricted combinations: SWIT3=1 and SWIT8=1
SWIT3=2 and SWIT8=1
SWIT3=3 and SWIT8=2
See also comments in input file.

4 The processes: theory and modelling

4.1 Relational diagram

The main processes modelled are depicted in Figure 4. The relations included in the figure are numbered to facilitate identification in this text. The symbols in the figure follow the convention Forrester (1961). Yet, Figure 4 is not strictly a *relational diagram* in the sense that it represents only system processes and no 'programming features'. In particular, it includes information on the interface between submodel and MAIN-plus-'physical-environment', and on the temporal resolution with which processes are simulated.

The solid line, separating SUBMODEL from MAIN PROGRAM, represents the system boundary in conceptual (not geometrical) sense: variables 'outside' this frame are inputs/outputs to/from the submodel. The variables in *shaded* symbols are calculated at each timestep DT by SAWAH. Those in *non-shaded* symbols are calculated/read only once per DELT. The numbered relations in Figure 4 represent the processes and/or program characteristics as described in this manual. See Table 3. The acronyms used in Figure 4 are listed in Table 4.

Table 3. Legend to numbered processes depicted in Figure 4 and described in this report.

relation # in Figure 4	Subsection	process/aspect
1	2.4, 2.5	input of state variables
2a	3.5, 4.3.4	water balance; crop transpiration
2b	3.5, 4.3.3	water balance; evaporation from ponded water
2c	3.5, 4.3.3	water balance; soil surface evaporation
2d	3.5, 4.3.1	water balance; rain
2e	3.5, 4.3.1	water balance; runoff
2f	4.3.5	water balance; capillary rise/drainage
3	4.2.2, 4.2.3	flux; Darcy equation
4	4.2.2, 4.2.3	flux; continuity equation
5	2.3, 2.4, 3.6.3	flux; dummy integration
6	3.8	DELT-average rate of change of (a) moisture content, and (b) depth of surface water layer
7	3.8	integration of DELT-average rates
8	3.8, 4.3.3	DELT-average rate of change of depth of evaporation front
Øa	4.2.3.2, 4.2.3.3	system parameters

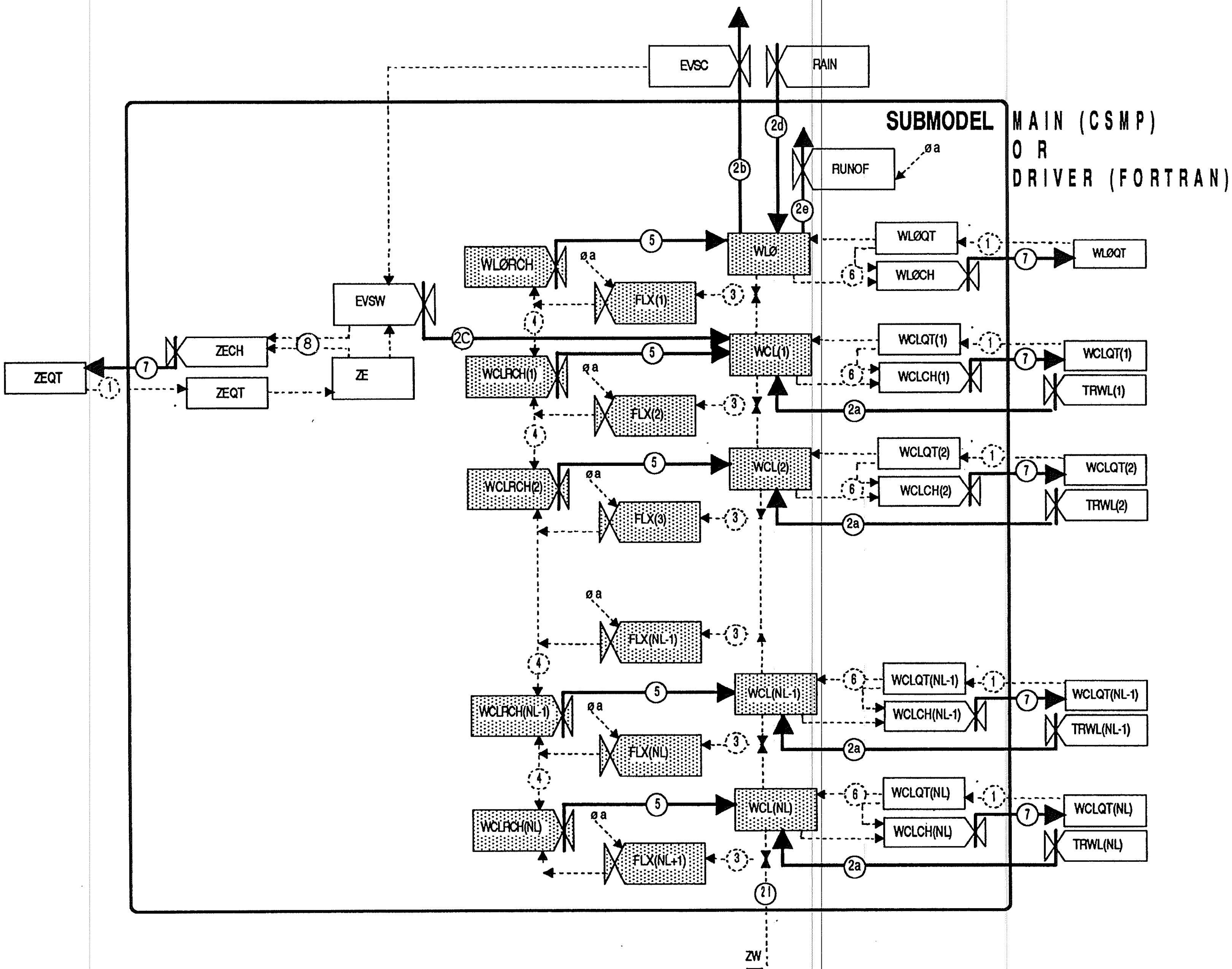


Figure 4. Relational diagram of the SAWAH soil water balance module. The diagram is explained in the text.

Table 4. List of acronyms in Figure 4.

		dimension
Øa	: some parameter (system property)	
EVSC	: potential soil evaporation rate	LT^{-1}
EVSW	: actual soil evaporation rate	LT^{-1}
FLX	: soil water flux over compartment interface	LT^{-1}
RAIN	: rainfall/irrigation rate	LT^{-1}
RUNOF	: runoff rate	LT^{-1}
TRWL	: uptake per compartment (transpiration)	LT^{-1}
WCL	: volumetric soil water content	-
WCLCH	: rate of change of water content, average over DELT	T^{-1}
WCLQT	: volumetric soil water content	-
WCLRCH	: rate of change of water content over DT	T^{-1}
WLØ	: surface water depth	L
WLØCH	: rate of change of surface water depth, averaged over DELT	LT^{-1}
WLØQT	: depth of surface water layer	L
WLØRCH	: rate of change of depth of water layer over DT	LT^{-1}
ZE	: depth of evaporation front	L
ZECH	: rate of change of depth of evaporation front over DELT	LT^{-1}
ZEQT	: depth of evaporation front	L

N.B. Units of some variables differ between MAIN and submodel. See Sections 2.5, 2.6; and Chapter 5, subroutine headers; and Appendix 6.

4.2 Flow processes

4.2.1 Water potential in saturated and unsaturated soil

Assumptions

Movement of water results from a gradient in *total water potential*. This gradient is the driving force to flow. In defining the total water potential, the following assumptions are made in SAWAH:

- osmotic gradients are absent;
- temperature gradients are absent;
- there is no external electrostatic force field;
- soil gas pressure is equal to atmospheric pressure;
- the solid soil matrix is rigid (i.e. there is no component in the pressure potential that results from the weight of overlying soil);
- water movement is one-dimensional (vertical).

Potentials

All potentials in the model are expressed in *head equivalents*, with the dimension of *length* (L). As a consequence of the above simplifications, only two components of the total water potential are considered: the *pressure* component h (pressure potential/head; the sign indicates that the value of this component decreases with increasing depth z) and the *gravitational* component $-z$ (gravitational potential/head; the sign indicates that the value of this component decreases with increasing depth z). The latter only depends on the position in the gravity force field. The sum of both components is the *hydraulic head H*, which is now equal to the total water potential in head equivalents. The water flux density ('flux', for short) at a point *p* is then determined by the gradient in total water potential, and the hydraulic conductivity, both at *p*. *All the above applies to water in both saturated and unsaturated soil. However, one important difference exists between the two, and this has a strong bearing on model formulation: the definition of the pressure potential itself.*

Unsaturated soil

In unsaturated soil, only the *local matric forces* - exerted by the solid soil particles on the water - determine the pressure potential ('local' refers to a volume large enough to include several pores, particles, *etc.*). A unique relation exists - at each point - between the *local water content* and the *local pressure potential*. This relation is the soil *moisture characteristic* or *pF-curve*. (It is assumed that hysteresis in this curve is absent.) In SAWAH, accordingly, this local character of pressure potential is applied: *h* is calculated for unsaturated soil from moisture content and the moisture characteristic. From local conductivity and local hydraulic potential gradient, the local flux is calculated (Section 4.2.3).

Saturated soil

In saturated soil, pressure potential at a given point is not determined by local moisture content, but by the weight of the overlying water mass. The profile of hydraulic conductivity in the saturated section directly affects the partitioning of this weight over a force on the solid matrix and a force on the underlying water, and therefore also affects the profile of pressure potential. As a consequence, a water mass perched on a completely impervious layer will be in a state of *hydrostatic equilibrium*, contrary to the same mass perched on a layer of low but finite permeability. In a state of hydrostatic equilibrium, the gradients in pressure and gravitational potential fully compensate each other: hence, the *gradient in total water potential (hydraulic head) is zero*, and there is no movement.

When the gravity force, in contrast, is not fully compensated by a gradient in pressure potential, a net *static* force remains, and results in flow. (The net *total* force is zero because friction compensates for the net static force, which is then borne by the solid matrix.)

Obviously, the flux - constant with depth within *a saturated section* - and the profile of pressure gradient must be calculated *simultaneously* from a set of equations, pertaining to all the soil compartments in the section. This includes defining the conductivities, and the pressure at the *borders* of the saturated section. (See Subsection 4.2.2.)

Summarizing: the flux at any point in the saturated section is, still, the product of local conductivity and local potential gradient, but this local gradient cannot be independently assessed, that is, without considering the entire saturated section: the conductivities of its composing layers (i.e. compartments), and the potentials at the section borders.

4.2.2 Flow in saturated sections of the soil profile

Flux density equation

For a one-dimensional saturated system - with flow along the vertical space coordinate *z* (L) - the flux density *q* (L/T) at any point in the system is given by the *Darcy equation* or *flux density* equation:

$$q = -k_s \frac{dH}{dz} = -k_s \frac{dh}{dz} + k_s \qquad (4.1)$$

with z positive downward, H the hydraulic head (L), h the pressure head (L), and k_s the hydraulic conductivity at saturation (L/T). The flux density equation for saturated flow is simple compared to that for unsaturated flow, because *only saturated* conductivity is included.

Mass conservation equation

The mass conservation or *continuity equation* expresses the rate of change of volumetric moisture content θ as the divergence of the flux density:

$$\frac{\partial \theta}{\partial t} = -\frac{\partial q}{\partial z} \qquad (4.2)$$

with t the time. Equation 4.2 simplifies for saturated soil to:

$$\frac{dq}{dz} = 0 \qquad (4.3)$$

The flux must be equal at each point within the saturated section, because no changes in moisture content can occur (except at the borders of the saturated section, which then would result in *expansion* or *contraction* of the saturated section).

Linearization

Equations 4.1 and 4.2 are approximated by replacing differentials by finite differences. In this report, a saturated section of the profile is called a *set*, because it consists of one, two or more adjacent soil compartments. The two *set borders* always coincide with *compartment boundaries*. The resulting set of linear equations is then solved in SAWAH under given boundary conditions and under the condition expressed in Equation 4.3.

Solution

The flux through a saturated set is calculated from:
(1) the difference in hydraulic head H between the two (upper and lower) borders of the set;
(2) the saturated hydraulic conductivities k_s of the compartments in the set.

The difference in H over a saturated set equals its thickness, if both boundaries are *free*, i.e. if both borders represent a transition to unsaturated soil. The pressure head at the borders is then zero. If, on the other hand, one of the set's borders coincides with the *soil surface* or the *lower profile boundary*, a pressure head term resulting from ponded water (surface) or groundwater (bottom) must be added to obtain the hydraulic head at the borders. The following equations are formulated for a total of n compartments i - *with known* k_i and z_i - in a set, and with known H_1 and H_{n+1} :

n flux density equations:

$$q_1 = - k_1 \frac{H_2-H_1}{z_2-z_1} \qquad (4.4)$$

$$q_2 = - k_2 \frac{H_3-H_2}{z_3-z_2}$$

$$\ldots\ldots = \ldots\ldots\ldots\ldots$$

$$q_n = - k_n \frac{H_{n+1}-H_n}{z_{n+1}-z_n}$$

and the summation

$$(H_2-H_1)+(H_3-H_2)+\ldots\ldots\ldots +(H_{n+1}-H_n) = H_{n+1} - H_1 \qquad (4.5)$$

Since we have n+1 linear equations with n+1 unknowns (the n differences ($H_{i+1}-H_i$) and the flux q, which is *constant throughout the set*), the system can be solved directly. In matrix form the equation is written as:

$$A \quad . \quad x = b \qquad (4.6)$$

or:

$$\begin{vmatrix} \frac{-k_1}{\Delta_1 z} & 0 & \ldots & 0 & -1 \\ 0 & \frac{-k_2}{\Delta_2 z} & \ldots & 0 & -1 \\ \ldots\ldots & \ldots\ldots & \ldots & \ldots\ldots & \ldots \\ 0 & 0 & \ldots & \frac{-k_n}{\Delta_n z} & -1 \\ +1 & +1 & \ldots & +1 & 0 \end{vmatrix} . \begin{vmatrix} \Delta_1 H \\ \Delta_2 H \\ \ldots\ldots \\ \Delta_n H \\ q \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \\ \ldots \\ 0 \\ H_{n+1}-H_1 \end{vmatrix}$$

(in which 'Δ' indicates a finite difference in z and H, respectively, between the two boundaries of each compartment, and the subscripts refer to the compartment index within the saturated set containing a total of n saturated compartments).

Note that the subscripts of k refer to compartment centers and those of z, H and q to interfaces, both numbered from the top downward and starting with the top border of the set as the first interface.

Solution: subroutines SUSTMS and SUSTMD

Equation 4.6 is solved by decomposing A in upper and lower triangles following standard routines as described by Press et al. (1986). Their subroutines LUDCMP and LUBKSB have been renamed to SUSTMD and SUSTMS, respectively. The solution vector x returns the differences in H over each compartment, and the flux q through the set.

Number of saturated sets

Usually there will be only *one* saturated section (a set, consisting of one or more compartments) in the profile, but more can occur, e.g. after intermittent heavy showers and/or when several layers of low permeability are present. In the model there is no limitation to the *number of saturated sets*, other than that associated with the total number of compartments: in a profile consisting of n compartments, a maximum of $n/2$ saturated sets may develop. They are separated by unsaturated layers. The solution procedure described is applied to each set independently. Figure 5 illustrates just one example of possible configurations.

Flow direction in saturated sets

The flow within a saturated section of the profile is *downward*, *except* when a positive pressure head exists at the lower end of the set. This is only the case when the lower border of the saturated set coincides with the lower boundary of the profile. It then depends on the value of pressure head at that lower boundary, whether the direction of the flow is upward or downward. If pressure head at that boundary is sufficient to counteract the effect of gravity, the flow direction is upward (or just zero). This happens when the groundwater table rises to reach into the profile.

For all other conditions the pressure head at the lower border of the set is zero. Then, the difference in pressure head between upper and lower border is zero (or positive, if the top border coincides with a ponded surface), and the difference in gravitational head between the two set borders determines the overall driving force, resulting in downward flow.

Pressure profile in saturated sets

Pressure head *within* saturated sets is generally *not zero*. On the contrary: the top part of the set usually shows a pressure head profile that, starting from zero at the upper edge, increases downward, followed by a steep decline to zero again at the lower end. This is the typical pressure profile when a flow-impeding layer creates a perched water table, as in wetland rice cultivation (e.g. Iwata et al., 1988). Of course, such profiles result 'automatically' from solving the above set of linear equations.

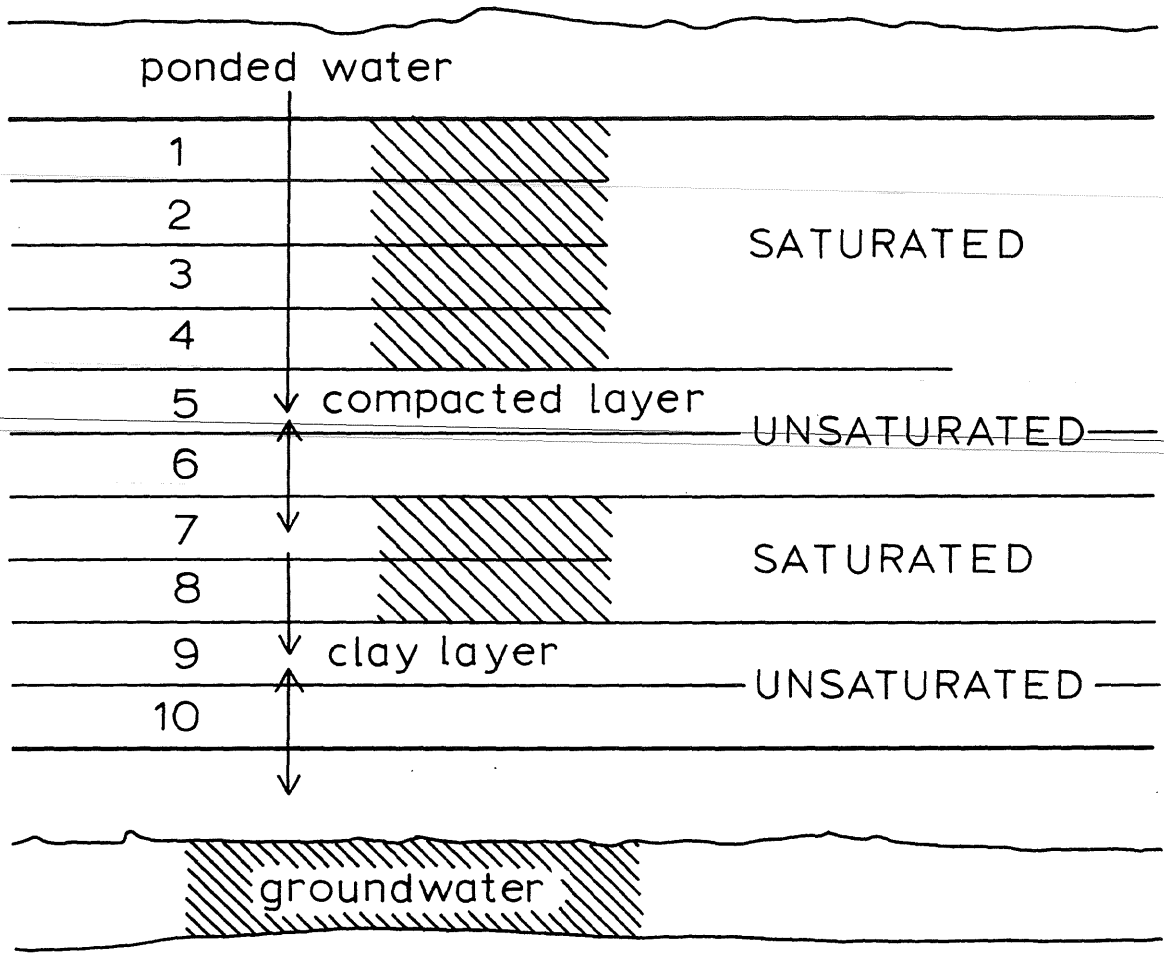


Figure 5. Example profile combining two perched water tables, with a true groundwater table below the bottom compartment of the soil profile. Double-headed arrows indicate that flux may be upward or downward. Single-headed arrows indicate that only one flow direction is possible at that particular depth in this configuration. The configuration may change within the duration of DELT.

4.2.3 Flow in unsaturated sections of the soil profile

4.2.3.1 Flux density equation and continuity equation

Flux density equation

Water flow between soil compartments is, also in this situation, governed by local hydraulic conductivity and local gradient in hydraulic head, as expressed in the *flux density equation*. The transport coefficient k, however, is now a function of the soil water content, and therefore of pressure head:

$$q = -k(h) \frac{dH}{dz} = - k(h) \frac{dh}{dz} + k(h) \qquad (4.7)$$

The $k(h)$ relation will be discussed in Subsection 4.2.3.2.

Mass conservation equation

Changes in moisture content follow from the *continuity equation* (Equation 4.2) which now shows, generally, a non-zero left hand term:

$$\frac{\partial \theta}{\partial t} = - \frac{\partial q}{\partial z} \qquad (4.2)$$

Linearization

To solve Equations 4.2 and 4.7 under given boundary conditions, the equations are linearized. At each time step DT, Equation 4.7 is applied to the *interfaces* between unsaturated compartments, and Equation 4.2 to the *compartments*. Thus, the state variable θ, its associated rate-of-change, and the space coordinate z refer to *compartment centers*, whereas the fluxes apply to the *interfaces*. The pressure potential h is also a state variable. It is directly derived from θ and therefore refers to the compartment centers, too. The moisture characteristic $h(\theta)$ will be discussed in Subsection 4.2.3.3.

In linearizing Equation 4.7, numerical artefacts arising from the averaging of the transport coefficient $k(h)$ over the region between compartment centers can be mitigated by using a *weighing procedure*. This is conveniently done by applying the concept of *matric flux potential*. This transformation will be discussed in Subsection 4.2.3.4. It will be shown that the concept can also be applied to heterogeneous (layered) soils.

4.2.3.2 The hydraulic conductivity function

SAWAH provides several options to describe the hydraulic conductivity function. The user can make a selection by setting the option switch SWIT3 to the proper value (Section 3.9):

Simple Rijtema (SWIT3 = 1)

The first option is an *exponential* $k(h)$ relation:

$$k(h) = k_s \, e^{\alpha h} \qquad (4.8)$$

where α (L^{-1}) is an empirical soil-specific coefficient, and k_s is the conductivity at saturation (LT^{-1}). Since hysteresis in the $h(\theta)$ curve is ignored, using $k(h)$ rather than $k(\theta)$ imposes no difficulties. This option is referred as 'simple Rijtema', because it represents only one branch of the full Rijtema form.

Full Rijtema (SWIT3 = 2)

The Rijtema form is a *two-branch* curve:

$$k(h) = k_s \, e^{\alpha h} \qquad \text{for} \qquad h >= h_{max} \qquad (4.9a)$$

$$k(h) = a|h|^{-1.4} \qquad \text{for} \qquad h < h_{max} \qquad (4.9b)$$

where a ($L^{2.4}T^{-1}$) and h_{max} (L) are empirical soil-specific constants.

Van Genuchten (SWIT3 = 3)

The third option is the form described by van Genuchten (1980), widely applied at present:

$$k(S) = k_s \; S^{\lambda} \; (1-(1-S^{1/m})^m)^2 \qquad (4.10)$$

where k_s is defined as above, and the dimensionless m and λ are soil-specific constants, and the *relative saturation* S at moisture content θ is defined as:

$$S = \frac{\theta - \theta_r}{\theta_s - \theta_r} \qquad (4.11)$$

where θ_s is the moisture content at saturation, and θ_r the so-called *residual moisture content*. This option is usually combined with the definition of the moisture characteristic according to Van Genuchten (Subsection 4.2.3.3, option SWIT8=2).

Power function (SWIT3 = 4)

Hydraulic conductivity can also be expressed as:

$$k = k_s \qquad \text{for } -1 < h < 0 \qquad (4.12a)$$

$$k = k_s|h|^n \qquad \text{for } h < -1 \qquad (4.12b)$$

where the exponent n, a soil specific constant, should be smaller than -1, and h in the domain definitions is expressed in cm.

User-defined function (SWIT3 = 5)

A 'blank' option is included for users who wish to define their own conductivity function by adapting the subroutine SUMSKM. Instructions are given in the listing of this subroutine.

Subroutine SUMSKM

The hydraulic conductivity pertinent to each unsaturated compartment is assessed at each time step DT in the subroutine SUMSKM. In this routine, *k is always calculated as a function of h*, also for those forms where the basic expression is one for $k(\theta)$. There is a good reason: the procedure that allows applying the *matric flux potential* concept to heterogeneous soils cannot be based on $k(\theta)$ or $D(\theta)$, the soil water diffusivity. See Subsection 4.2.3.5, *layered profiles*.

4.2.3.3 The moisture characteristic

SAWAH provides several options to describe the moisture characteristic. Similar to the hydraulic conductivity functions, a selection is made with the help an option switch: SWIT8. Any combination of expressions for hydraulic conductivity and moisture characteristic can be selected (See previous subsection and Section 3.9).

Driessen (SWIT8 = 1)

The moisture characteristic described by Driessen (1986) has a simple form, but is attractive because it contains only one parameter, in addition to the moisture content at saturation, θ_s:

$$h(\theta) = - \exp\left(\left(- \frac{1}{\gamma} \ln \frac{\theta}{\theta_s}\right)^{1/2}\right) \qquad (4.13)$$

where γ is a texture-dependent soil parameter (dimensionless). This model is a coarse approximation for most soils, especially near saturation.

Van Genuchten (SWIT8 = 2)

The 'counterpart' of Equation 4.10 is Van Genuchten's moisture characteristic:

$$h(S) = - \frac{1}{\alpha} \left(S^{-1/m} -1\right)^{1/n} \qquad (4.14)$$

where α is a scale factor (L^{-1}), and S is the relative saturation (Equation 4.11). The relation between the parameters m and n is approximated by:

$$m = 1 - 1/n \qquad (4.15)$$

Interpolation between measured data points (SWIT8 = 3)

Measured pF curves can be introduced as pairs of (θ/θ_s, pF), thus avoiding a functional relation. In this option, eleven pF-values must be entered, corresponding to θ/θ_s increments of 0.1. The order is from pF = 7 for θ/θ_s = 0 ('oven-dry') down to pF = 0 for θ/θ_s = 1. (In fact, pF < 0 for $h > -1$ cm, but the portion of the curve between h = - 1 and h = 0 is ignored). For input preparation, see input file FILIN (Subsection 2.5.1.2 and Appendix 4).

User-defined function (SWIT8 = 4)

An option is included for a user-defined moisture characteristic. Instructions are given in the listing of the subroutine SUWCMS.

Subroutine SUWCMS: h(θ) or θ(h)

Calculation of $\theta(h)$ and $h(\theta)$ for various purposes in the model, is performed in the routine SUWCMS on the basis of one of the above equations. At *initialization*, $\theta(h)$ is calculated if the initial state of the profile is assumed to be in equilibrium with groundwater depth. At most of the SUWCMS CALLs during *rate calculations*, $h(\theta)$ is assessed, but at some instances during calculation of the time step DT, $\theta(h)$ is produced as output to the calling subroutine.

4.2.3.4 Matric flux potential

Averaging problem

In crop growth modelling, a limited number of soil compartments is generally distinguished. Spatial refinement in the description of soil water processes is usually not the objective, and often not necessary. Using thick layers implies, however, that flux calculations are somewhat 'crude', if the linearized form of Equation 4.7 is applied injudiciously.

The inaccuracy results from the need to obtain, in calculating the flux over an interface i between compartments i-1 and i, an average conductivity $k_{av,i}$. Usually, $k_{av,i}$ is assessed as some average of the k values in the centres of the two adjacent compartments, k_{i-1} and k_i. Values of k can differ, however, from one compartment to the next by several orders of magnitude because of the strong non-linearity of $k(\theta)$ or $k(h)$. Hence, different averaging procedures applied to obtain k at interface i may lead to widely different values of k_{av}, with a dramatic effect on the flux resulting from multiplication with the local potential gradient; particularly because the latter is often steep.

Matric flux potential

Application of the matric flux potential concept provides a solution to the averaging problem. Essentially, it allows calculating a weighted average of k over h. An upper limit is thus imposed on the flux across an interface, because the simulated flux can never be higher than the linearized gradient of the flux potential, which has a limited range. Matric flux potential, Φ, is defined as:

$$\Phi = - \int_{h}^{0} k(h)\, dh \qquad (4.16)$$

(Klute, 1952; Gardner, 1958). Combining Equation 4.16 with either of the conductivity functions presented in Subsection 4.2.3.2 yields $\Phi(h)$. In SAWAH $k(h)$ is numerically integrated if the full *Rijtema*, *Van Genuchten*, or user-defined function are selected. As a consequence of numerical integration, computation time is relatively high for these functions. For the *simple Rijtema* and the *power* expression, the analytical integrals are available.

Integration over h: subroutine SUMFLP

SAWAH calculates the integral Φ in the subroutine SUMFLP. A switch in SUMFLP indicates which $k(h)$ curve (Subsection 4.2.3.2) is used. If numerical integration has to be applied, Φ is obtained by *Gauss* integration. Then, SUMFLP 'calls' the routine SUMSKM to assess $k(h)$ at key points. For the analytical integrals, Equation 4.16 is evaluated directly.

4.2.3.5 The flux density equation in matric flux potential form

Transformation

Equation 4.16 can be substituted into the flux density equation (Equation 4.7) to express the *matric* component of the flux as the gradient in *matric* flux potential:

$$q = -d\Phi/dz + k \qquad (4.17)$$

Now, the gravity component is still expressed in its old form, k. Because it is assumed that the flux i between the centres of adjacent compartments i-1 and i is constant with depth during each time step, the integral form of Equation 4.7 may be written as:

$$q \cdot \Delta z = -\Delta \Phi + \int_{z_{i-1}}^{z_i} k \, dz \qquad (4.18)$$

where the symbol Δ represents a finite difference. The second term on the right-hand side is the gravity term. Its value can be approximated by assuming a linear relation of h with depth between the centres of compartments i-1 and i:

$$\int_{z_{i-1}}^{z_i} k \, dz = (dz/dh) \int_{h_{i-1}}^{h_i} k \, dh \qquad (4.19)$$

Since this linearization applies only to the gravity component, it is not a gross simplification. The flux density equation may now be rewritten as:

$$q = \left(\frac{-1}{\Delta z} + \frac{1}{\Delta h}\right) * \int_{h_{i-1}}^{h_i} k \, dh \qquad (4.20)$$

which, combined with Equation 4.16, yields

$$q = \left(\frac{-1}{\Delta z} + \frac{1}{\Delta h}\right) * \Delta \Phi \qquad (4.21)$$

Within the brackets, the first term refers to the *matric* component of the flux, and the second to the *gravity* component. So, the flux potential concept applies to both terms.

Calculation of fluxes: subroutine SUUNST

The flux across each interface between two unsaturated compartments is calculated from Equation 4.21 in the subroutine SUUNST. SUUNST expresses the linearized gradient in matric flux potential as the *difference between the values at the compartment centres*, divided by the distance. At interfaces that separate *saturated* from *unsaturated* compartments, the matric flux potential gradient is linearized over the distance *between compartment center and interface*. At such interfaces, both h and Φ are equal to zero. See also Subsection 4.2.4.

Unit gradient flow; free drainage

If no pressure gradient exists between two compartments, the gravity term in Equation 4.21 is indefinite. Obviously, ΔΦ is then zero according to Equation 4.16. The flux resulting from gravity is then directly assessed in SUUNST as the conductivity $k(h)$, obtained by a CALL to SUMSKM. The same applies to the lower boundary of the soil profile, if the *free drainage*

option has been selected; in that case, drainage proceeds at the rate $k(h_n)$, where n refers to the bottom compartment.

Matric flux potential in layered profiles

A slight complication arises when *intrinsic soil properties*, such as the hydraulic conductivity function and the moisture characteristic, change with depth. Writing the flux density as the gradient of a matric flux potential *per se* is correct *only* for homogeneous soils. It has often been stated, therefore, that the matric flux potential form of the flux density equation cannot be applied to *heterogeneous* soils. This notion, however, is not correct.

The averaging procedure applied must avoid evaluating the gradient of matric flux potential across an interface that separates different soil materials. In SAWAH, this is done as follows. Consider two compartments, $i-1$ and i, of different soil types (Types 1 and 2), at pressure heads h_{i-1} and h_i, respectively. The finite difference $\Delta\Phi$ is first approximated as the integral over h as if only soil Type 1 were present in both compartments. That is repeated, assuming both compartments to be of Type 2. Subsequently, the geometric mean is taken, thus replacing the integral in Equation 4.20 by:

$$\int_{h_{i-1}}^{h_i} k \, dh = \left[\int_{h_{i-1}}^{h_i} k_1(h)dh * \int_{h_{i-1}}^{h_i} k_2(h)dh \right]^{½} \qquad (4.22)$$

where $k_1(h)$ and $k_2(h)$ are the conductivity functions for the two soil types. While this procedure is fully analogous to that applied in numerical schemes based on $k(h)$, $k(\theta)$ or $D(\theta)$ - namely, calculating an average transport coefficient between nodes or compartments as a geometric mean of the values in the individual compartments - the advantage of the matric flux potential form is retained: h-weighted averaging of k.

This explains why in SAWAH only $k(h)$ relations and not $k(\theta)$ or $D(\theta)$ are applied (Subsection 4.2.3.2), even when the basic curves are given as a function of θ. The equivalent of Equation 4.22 obtained by integration over θ, rather than h, would - erroneously - always result in a flux *directed towards the lower* θ. In reality, however, this does *not necessarily* happen because it is the gradient of hydraulic potential, not of moisture content, that drives water flow. For a more elaborate treatment of the theory and application of matric flux potential, see Klute (1952); Gardner (1958); Raats (1970); Shaykewich & Stroosnijder (1977); for measurement of $\Phi(\theta)$, see ten Berge et al. (1987)).

4.2.3.6 Continuity equation

The rate of change of θ is calculated in SAWAH from the linearized form of Equation 4.2. (Subsection 4.2.3.1). After each time step DT, the new θ in each compartment is calculated in SUINTG by subtracting the outflow from the inflow during that timestep, dividing this difference by compartment thickness, integrating the resulting rate of change over DT, and adding the change to the previous θ value. Integration over time is *rectangular (Euler)*.

For the next time step, the pressure head h corresponding with the new moisture content is assessed again for each compartment, and the whole procedure is repeated.

4.2.3.7 *Flow equation*

The combination of Equations 4.2 and 4.7 (here replaced by 4.21), the *flow equation* or *Richards* equation, is written as:

$$\frac{\partial \theta}{\partial t} = \frac{-\partial}{\partial z} \left(-k(h) \frac{\partial h}{\partial z} + k(h) \right) \qquad (4.23)$$

or, in matric flux potential form

$$\frac{\partial \theta}{\partial t} = \frac{-\partial}{\partial z} \left(- \frac{\partial \Phi}{\partial z} + k(h) \right) \qquad (4.24)$$

This equation is not applied directly in SAWAH, because it is solved in two steps: *flux* calculation, and *flux divergence* calculation. Yet, in principle it is the flow equation that is solved by all models that simulate non-steady soil water flow in unsaturated soil on the basis of the Darcy equation, under given boundary conditions.

4.2.4 Selection of fluxes at saturated-unsaturated interfaces

4.2.4.1 *The selection problem*

Fluxes between adjacent unsaturated compartments, FLXUNT, are calculated from the local gradient in matric flux potential across their interface. A flux *across* that interface is the result (Subsection 4.2.3). For an *unsaturated* compartment adjacent to a *saturated* compartment, the gradient in matric flux potential is evaluated between the *center* of the unsaturated compartment and the *interface* with the saturated soil, where h and Φ are zero. A flux *towards* or away *from* the interface results. The latter need not be equal to the flux FLXSTT in the *saturated* compartment at the other side of the interface, calculated according to the principles outlined in Subsection 4.2.2. Yet, the actual flux over an interface can have only one value, because no water can be stored there. Each time step, a careful choice must be made at each unsaturated-saturated interface, as to which of the two *'tentative fluxes'* - FLSTT or FLXUNT - is selected as the *actual flux* FLX. In short, this is referred to as the *selection problem.* It is solved in the subroutine SUSEFL. This subsection explains the decision rules applied. For brevity, a boundary between a saturated and an unsaturated soil section will be referred to as an *S-U* boundary.

Causes of the selection problem

The selection problem arises from a few basic assumptions in SAWAH:
- the solution scheme for saturated flow differs from that for unsaturated flow, as a consequence of different definitions of pressure potential (Section 4.2.1), and of the different conditions imposed on the continuity equation (Equation 4.3);
- *S-U* boundaries coincide with compartment boundaries; compartment boundaries are fixed in space, hence S-U boundaries move in discrete steps;
- at the *S-U* interface, the pressure potential has a defined, fixed value (zero).

'Continuous' alternative?

For homogeneous soils, some of these modelling problems could have been avoided by allowing the *S-U* boundary to move continuously, rather than in discrete steps. Ignoring the effect of overlying water mass on pressure potential in saturated soil, as in some other models, would also simplify the problem. In most field situations, however, soils *are layered*, i.e. heterogeneous in the vertical direction. Modelling a continuous movement of the *S-U* boundary involves then its own particular difficulties, especially in situations with more than one saturated section. After all, saturated sections usually occur *as a consequence* of the layered character of soil, so a module should be capable of handling this combination of saturation and soil profile heterogeneity.

Examples

In simulation, *extension* or *contraction* of saturated sections depends on the proper choice of fluxes at *S-U* interfaces. A realistic solution to each selection problem is one that not only yields a correct description of duration and spatial extent of saturation, but also of the associated flux values; and that prevents oscillations at the same time. No single general rule can be applied to determine FLX for all types of *S-U* boundaries, as many hydrologically different situations exist where saturated soil 'meets' unsaturated soil. Some examples:

- water in a saturated section of the profile is fully 'supported' by groundwater: *hydrostatic equilibrium*;
- groundwater is entering the profile: pressure head at the lower profile boundary exceeds the value that would just maintain hydrostatic equilibrium within the saturated zone;
- groundwater level retreats, but is still within the profile: pressure head at the lower profile boundary has decreased and is now insufficient to maintain the current saturation status;
- a perched water table is present: saturated soil is overlying unsaturated soil;
- a saturated section, resulting from a perched water table, 'desintegrates': it 'looses' compartments 'to the unsaturated state';
- the thickness of a saturated section, resulting from a perched water table increases by surface supply: it 'absorbs' (at the top end) compartments that were previously unsaturated;
- a perched saturated section moves down as a whole;
- the upper border of the saturated soil section coincides with the soil surface;
- the lower border of the saturated section coincides with the bottom of the soil profile.

Oscillations

Oscillations in the moisture status of compartments (saturated/ unsaturated), as a numerical artefact, tend to occur particularly when the piezometer level is *between* compartment interfaces rather than exactly at an interface: the total difference in hydraulic head over the saturated set is then just positive or just negative, depending on whether the compartment that 'contains' the free groundwater level is included in the set or not. Oscillations may be avoided by including *prospective calculations*, indicating whether including an additional compartment in a saturated set - or excluding one from an existing set - would invert the direction of the flow. For example, when groundwater rises into the profile, pressure head at the lower profile boundary is higher than needed to 'support' the current saturated section. So, the saturated set is extended at the top by including an additional compartment as a result of upward flow. This continues at a rate that depends on the values of FLXSTT in the set and FLXUNT at the top of the set. At some point, however, including one more compartment would revert the flow. Reaching that point is avoided by a prospective calculation (FLXSQ1, FLXSQ2).

4.2.4.2 Solution procedure

Description of acronyms

FLXSTT tentative flux according to SUSTFL in a saturated set;
FLXUNT tentative flux according to SUUNST in an unsaturated compartment bordering an *S-U* interface (flux towards/away from *S-U* interface);
FLX the actual (realized) flux at a compartment interface, selected according to the procedure in SUSEFL; FLX is calculated for all compartment interfaces, whether S-U, U-U, or S-S;
FLXSQ1 the value of FLXSTT that would result if the thickness of a saturated set would be reduced by one compartment at the top of the set;
FLXSQ2 the value of FLXSTT that would result if the thickness of a saturated set would be increased by one compartment at the top of the set;
(*a*) location index: the upper S-U interface of a saturated set;
(*b*) location index: the lower S-U interface of a saturated set;

General rules

Each of the examples in Subsection 4.2.4.1 can be characterized by analyzing the magnitude and direction of the flux at either side of each *S-U* boundary, at any given time during the simulation. All combinations of FLXSTT and FLXUNT possible at such a boundary can be grouped into so-called *situation classes*. A few general rules are given:

- FLXSTT in *perched* saturated sections is always downward (i.e. positive); at the borders of such sections, the unsaturated fluxes FLXUNT are always 'dominant' over FLXSTT, if they are directed *away from* the saturated set (upward at the top, downward at the bottom). Then, FLX equals FLXUNT.
- In contrast, when FLXUNT is directed towards the perched section, FLXSTT can impose a limit on FLXUNT, and sets the value of FLX. Note that *unsaturated* flow towards an *S-U* boundary (h=0) can only be *downward*, since upward flow is always in the direction of decreasing pressure head. So, unsaturated flow towards an S-U boundary occurs only at the top of a saturated set.
- In both *perched* and *phreatic* saturated sections - the latter characterized by a positive pressure head at the lower profile boundary - fluxes over the internal interfaces can never exceed FLXSTT; at the borders, however, they can.
- When FLXSTT is directed towards an *S-U* interface, and FLXUNT imposes a limit (FLX) on the outflow at that interface, the fluxes over the internal interfaces in the saturated set are also limited to the same value, FLX.

Analysis

Analysis starts at the *outflow end* of a set, which is either at (*a*) or at (*b*), depending on the direction of FLXSTT. If FLXSTT=0 (no *outflow end* exists), analysis starts at (*a*). The objective is to assess the fluxes FLX(*a*), FLX(*a*+1, ...,*b*-1) and FLX(*b*): the fluxes at *top*, *internal*, and bottom interfaces of a saturated set, respectively. The procedure is repeated for each saturated set, and at each time step DT.

In the following notation, the *minimum of a set of values* is written as min(...,...), the *maximum* as max(...,...). It is important to realize that flow is positive in *downward* direction, *negative in upward* direction, and that the min/max operator applies over the entire domain of fluxes- i.e. including positive, negative, and zero values. *THE WHOLE PROCEDURE IS GRAPHICALLY ILLUSTRATED IN APPENDIX 8.*

Situation classes:

Three main situation classes are distinguished, each subdivided into a number of subclasses:

1. the tentative flux in saturated section is downward (FLXSTT > 0)
2. the tentative flux in saturated section is upward (FLXSTT < 0)
3. the tentative flux in saturated section is zero (FLXSTT = 0)

The classes and subclasses are further elaborated below.

Class

1. *Definition* Downward flow in saturated section: FLXSTT > 0.

1.1 *Definition* Saturated section overlying unsaturated soil.

Description This is the standard situation of a *perched* water table: a *free* lower end at the saturated section. At outflow end (*b*), FLXUNT can only be downward or zero. At (*a*), FLXUNT may be upward or downward, depending on suction in the overlying unsaturated compartment.

Solution

FLX(*b*)=min(FLXUNT(*b*),FLXSTT)	downward
FLX(*a*)=min(FLX(*b*),FLXUNT(*a*))	upward/downward
FLX(*a*+1,...,*b*-1)=FLX(*b*)	downward

Remarks This solution implies that a perched saturated set 'desintegrates' from the top compartment downward.

When (*a*)=1, i.e. when the top of the saturated section coincides with the soil surface, *and* water is ponded on the surface, FLXUNT(*a*) is not defined;
then, the solution retains the same form, except that

FLX(*a*)=FLX(*b*)	downward

1.2 *Definition* Saturated section coinciding with the lower end of the profile.

Description (*b*) coincides with the bottom of the profile; no limitation to outflow from set is imposed by unsaturated soil underlying the saturated set.

Solution

FLX(*b*)=FLXSTT	downward
FLX(*a*)=min(FLX(*b*),FLXUNT(*a*))	upward/downward
(or FLX(*a*)=FLX(*b*) if *a* is surface)	downward
FLX(*a*+1,...,*b*-1)=FLX(*b*)	downward

2. *Definition* Upward flow in saturated section: FLXSTT < 0.

Description This situation only occurs when at the bottom end of the profile a positive pressure head exists, large enough to compensate for the gravity potential gradient within the section. Then (*b*) coincides with the lower profile boundary. The system is developing towards a state of hydrostatic equilibrium, with groundwater in the profile; water is still entering the profile. Key questions are:

- will the (upward) direction of flow be maintained if the next overlying unsaturated compartment becomes saturated;
- what would FLXSTT be, if indeed the overlying compartment would be incorporated into the saturated section;

The answer to both depends on the pressure head at the lower profile boundary, and the thickness of the saturated set after incorporating an additional compartment. The associated value of FLXSTT in that case is stored in FLXSQ2.

2.1	*Definition*	FLXSQ2 < 0 or FLXSQ2 = 0	
	Description	Pressure head at (*b*) high enough to incorporate next unsaturated layer at (*a*) without inverting the direction of flow in the saturated set.	
	Solution	FLX(*a*)=FLXSTT	upward
		FLX(*a*+1,...,*b*-1)=FLX(*a*)	upward
		FLX(*b*)=FLX(*a*)	upward
2.2	*Definition*	FLXSQ2 > 0	
	Description	Instability (oscillation) would result if an additional overlying unsaturated compartment would be incorporated into the saturated section. Flow at (*a*) in the unsaturated compartment can be either downward towards the saturated set (FLXUNT(*a*) > 0), upward out of the saturated set (FLXUNT(*a*) < 0), or zero. FLXUNT(*a*) is not defined when (*a*)=1 (surface).	
2.2.1	*Definition*	FLXUNT(*a*) < 0 or FLXUNT(*a*) =0	
	Solution	FLX(*a*)=max(FLXUNT(*a*),FLXSTT)	upward or zero
		FLX(*a*+1,...,b-1)=FLX(*a*)	upward or zero
		FLX(*b*)=FLX(*a*)	upward or zero
	Remarks	The unsaturated compartment overlying the saturated set is near hydrostatic equilibrium; a procedure is included to accelerate attainment of the moisture content corresponding to hydrostatic equilibrium in that compartment, thus avoiding impractically small time steps; once this value is reached, a steady state (inflow = outflow) is maintained in that compartment as long as this situation 2.1.1. persists.	
2.2.2	*Definition*	FLXUNT(*a*) > 0	
	Solution	FLX(*a*)=min(FLXSQ2,FLXUNT(*a*))	downward
		FLX(*a*+1,...,*b*-1)=FLX(*a*)	downward
		FLX(*b*)=FLX(*a*)	downward
2.2.3	*Definition*	FLXUNT(*a*) = 0	
	Solution	FLX(*a*)=0	-
		FLX(*a*+1,...,*b*-1)=0	-
		FLX(*b*)=0	-

2.2.4 *Definition* a=1 (surface; FLXUNT(a) not defined)
Solution FLX(a)=FLXSTT upward
FLX(a+1,...,b-1)=FLX(a) upward
FLX(b)=FLX(a) upward

3. *Definition* FLXSTT=0.
Description hydrostatic equilibrium (zero flow) *would* exist in the saturated set, *if* flow in the unsaturated compartment overlying the S-U interface at (a) would be zero. Flow through the saturated set, however, may be non-zero due to non-zero unsaturated flow at (a). Groundwater is present in the profile. (b) should always coincide with the bottom of the profile. Depending on FLXUNT(a), the saturated set could 'loose' its upper compartment. The associated value of FLXSTT is stored in FLXSQ1. For this situation class, FLXSQ2 (see *2. Description*) is positive, FLXSQ1 is always negative (upward): starting from equilibrium (FLXSTT=0), adding a compartment increases the total difference in gravitational potential over the set, whereas the total difference in pressure head over the set remains unchanged; and opposite for 'removal' of a saturated compartment.
If, in this situation class, groundwater level coincides with the soil surface, FLXSQ2 is set to zero in SUSTFL and FLXUNT(a) is set to zero in SUUNST.

3.1 *Definition* FLXUNT(a) < 0
Description upward unsaturated flow away from interface (a). If FLXSQ1 is insufficient to 'sustain' FLXUNT(a), the top compartment of the saturated set will become unsaturated.
Solution FLX(a)=FLXUNT(a) upward
FLX(a+1,...,b-1)=max(FLXSQ1,FLX(a)) upward
FLX(b)=max(FLXSQ1,FLX(a)) upward

3.2 *Definition* FLXUNT(a) > 0
Description downward unsaturated flow towards interface (a). If FLXSQ2 is smaller than FLXUNT(a), the overlying unsaturated compartment may become saturated.
Solution FLX(a)=min(FLXSQ2,FLXUNT(a)) downward
FLX(a+1,...,b-1)=FLX(a) downward
FLX(b)=FLX(a) downward

3.3 *Definition* FLXUNT(a) = 0
Description no flow at interface (a). This case occurs when the overlying unsaturated compartment is in equilibrium with groundwater, and also when (a) coincides with the soil surface. In both cases, zero fluxes result for all interfaces of the saturated set.
Solution FLX(a)=0 zero
FLX(a+1,...,b-1)=0 zero
FLX(b)=0 zero

4.3 System boundaries, boundary conditions, and water balance

Geometry

In the absence of surface ponding, the upper boundary of the system is the soil surface. Otherwise, the water surface represents the upper boundary. The lower boundary is the bottom of the user-defined soil profile. Also the surface of roots can be considered a boundary to the soil-water system.

Water balance

The fluxes across these boundaries are the water balance terms: *rain/irrigation* (*p*), *runoff* (*r*), *soil evaporation* (*e*), *transpiration* (*t*), *drainage/capillary rise* (*d*), *and artificial drainage* from tubes (*ad*). The *water balance equation* is written as:

$$\int_{z=z_{1b}}^{0} (\theta_t - \theta_i)\, dz + (z_p - z_{pi}) = \int_{t=t_i}^{t} (p+r+e+t+d+ad)\, dt \qquad (4.25)$$

where z_p is the thickness of the surface water layer and the subscript *i* refers to the initial state; z_{1b} is the depth of the lower boundary of the soil profile. The water balance terms (right hand of equation) are *positive* for *gain* and *negative* for *loss* terms. SAWAH includes a water balance check on the basis of this equation, performed in the routine SUWCHK. See Section 2.8

Boundary conditions

The values of the fluxes at the boundaries are calculated from a combination of external conditions and system conditions/characteristics. When the latter restrict the flux to a value lower than 'external' supply (rain) or demand (evaporation), the boundary condition 'automatically' becomes a *potential*. This is always the case at the lower boundary, and sometimes at the soil surface too.

When supply rate (rain) or potential extraction rate (potential evaporation) is limiting, an upper limit is imposed on the flux that would otherwise result from maintaining a fixed potential value (zero for rain and very negative for evaporation) at the surface boundary.

4.3.1 Rainfall, infiltration, and runoff

Partitioning of rain

As a consequence of selecting the water surface as the upper boundary, the partitioning of *rain* over *surface storage*, *infiltration* and *runoff* occurs *within* the system boundaries. Only *rain* and *runoff* are *water balance terms* proper, because they represent fluxes *across* system boundaries.

Rain is treated as an impulse function (Section 3.5). The quantity received per DELT is added to the ponded surface layer and is left to infiltrate. First, the depth of the resulting water layer is independent of the surface storage capacity. Subsequently, water flow calculations start by solving the *flow equation* with small time steps DT. At the completion of DELT, water still present on the surface *in excess of the surface storage capacity*, is removed as runoff.

Boundary conditions during ponding

As long as infiltration continues, the surface matric flux potential is non-negative. The flux through the surface is obtained by linearization of the matric flux potential between the surface and the centre of the top compartment.

A complication arises because pressure head at the surface is positive (equal to the depth of the ponded water layer). Since matric flux potential, in Equation 4.18, is defined with pressure head ranging from negative values up to zero, an additional term must be included to express the effect of positive surface pressure. This term $k_s\ z_p$ is introduced as follows:

$$q = \left(\frac{-1}{\Delta z} + \frac{1}{\Delta h}\right) * \left[-\int_{h_1}^{h=0} k\ dh \ - (k_s\ z_p) \right] \qquad (4.26)$$

where k_s is the saturated conductivity, h_1 the pressure head at the centre of the first compartment, and z_p the depth of the ponded water layer.

When the top soil compartment becomes saturated, the flux through the surface is calculated according to Subsection 4.2.2. The surface water layer then adds to the surface pressure head.

Inaccuracies in infiltration

Applying a coarse space discretization to solve Equation 4.26, as in SAWAH, leads to underestimation of infiltration rate during the first stage of infiltration, because the gradient of hydraulic head - and hence matric flux potential - near the surface is steep. Accurate simulation of infiltration over the period of DELT requires fine space steps near the surface, or analytical solutions. While the former approach imposes time steps too small for practical purposes in crop simulation, analytical solutions require a homogeneous initial state. The underestimation of infiltration resulting from combining Equation 4.26 with large Δz is counteracted by two other simplifications introduced:
- the water is retained on the surface throughout the full duration of DELT, or as long as needed for all water to infiltrate;
- a positive pressure head that exceeds the actual value is maintained during that period because all rain is 'stacked' on the surface.

Alternative approaches

An alternative impulse formulation could be proposed: subtracting runoff at the beginning of DELT, thus restricting the quantity of water on the surface to what can be stored at any moment. That, however, would lead to overestimated runoff in soils with high conductivities, where infiltration capacity (on a daily basis) exceeds surface storage capacity.

Inaccuracies result from describing a continuous event in terms of an impulse function. Under high-intensity rainfall regimes, reality is probably best approximated by an impulse-type formulation, except on sloping land. Under low-intensity regimes, it may be more realistic to use the DELT-averaged rainfall intensity as a 'supply rate' active during the full DELT. The difference between the two approximations is relevant where runoff plays a major role. Rainfall intensity is then important to predict daily water intake.

Information on *rainfall intensity* and time of the day at which precipitation occurs, is usually not available. Moreover, the movement of soil water near the surface is difficult to predict: reported soil hydraulic properties often do not apply to the surface centimetres of soil; temperature (viscosity) may play a role; and discrete steps in time and space should be small. In view of these uncertainties and numerical difficulties, the current formulation is considered adequate.

4.3.2 Exfiltration

If the full profile is saturated and *artesian* water moves upward, the flux through the upper boundary is assessed by the same procedure used for saturated sets in general, as described in Subsection 4.2.2. Such upward flow continues untill a surface water layer has developed of a depth sufficient to just compensate the positive pressure head at the soil surface. Where this *equilibrium depth* exceeds the surface storage capacity (which in rice fields is usually determined by bund height), water will continue to be lost. This loss term is interpreted in the model as runoff.

4.3.3 Evaporation

Ponded surface

If water is ponded on the surface, evaporative water loss takes place at the expense of surface water. When only a very thin surface water layer is present, this is removed first and the remainder is extracted from the top soil compartment.

Dry surface: other models

In calculating the evaporative surface flux, many simulation models solve the flow equation numerically and use thin surface compartments to cope with the steep moisture content gradients near the surface. It has been shown, however, that compartment size seriously affects the predictions. Lascano & Van Bavel (1986) reported errors in cumulative evaporation of up to 20%, as top compartment size was increased from 0.5 to 1.0 cm. Computations with these compartment sizes already become very time consuming. Moreover, precise solutions of the flow equation for evaporation are only warranted when temperature is also taken into account. This is beyond the scope of the present model.

Other models use an *a priori* determined course of *actual evaporation vs time*, thus avoiding a direct relation between near-surface water content and evaporation rate. Such an approach, however, cannot properly deal with the proximity of groundwater, unless its depth is constant. Also, the parameters required in such models are not directly related to the standard soil physical parameters, and do not apply to layered profiles. In lowland rice systems, variable shallow groundwater is generally the rule and the soil profile is always layered.

To cope with these problems, SAWAH combines a numerical flow scheme for the subsoil (Section 4.2) with an analytical solution near the surface.

Evaporation front

Whereas potential evaporation is determined by non-soil factors, the actual evaporation rate of soil water may be lower than the potential rate, due to transport resistance in the soil and to the low energy status of soil water (relative humidity). As evaporation proceeds, a layer of dry soil gradually develops and the evaporation rate decreases. In a wet soil, the actual transition from liquid to vapour takes place at the surface. In a drying soil, this zone of liquid-vapour transition moves down. Evaporation then occurs only below the dry surface layer and vapour moves upward by diffusion. Often, the transition between the dry top layer and the moist subsoil is rather sharp and is called the *evaporation front* and its depth is indicated here as z_E. It is considered a plane, i.e. with no capacity.

Evaporation rate

The dry surface layer is treated as a diffusion resistance. Vapour diffusion through dry soil is of the same order of magnitude as diffusion through free air. The 'obstruction' of molecules by soil solid particles is partly compensated - and sometimes more than that - by interaction of vapour molecules with 'liquid bridges' in the dry soil (e.g. Philip and De Vries, 1957).

At any given time, actual evaporation rate e is written as:

$$e = c_2 / z_E \qquad (4.27)$$

where c_2 is a constant representing the resistance of dry soil to vapour movement. c_2 is considered as a soil-specific constant, though its value depends on air humidity, wind speed and soil temperature.

The integral of e over the period of DELT is subtracted as an impulse function from the amount of water contained in the first soil compartment. Hence, the flux over the top boundary is not incorporated in the flow equation.

Movement of z_E

The front depth z_E is a state variable. The corresponding rate variable, i.e. the rate at which z_E moves into the soil as drying proceeds, is the net result of two principal processes: upward *vapour removal* through the dry soil into the atmosphere, and *liquid supply* from the subsoil to the front at z_E. If the two transport processes proceed at comparable rates, the position of z_E will hardly change. However, z_E will increase if removal of vapour dominates, while alternatively, when the liquid supply to the front dominates, the front will move towards the surface. This may be the situation with rising groundwater.

To compute the supply of liquid water to the front, some assumptions made by Parlange in describing infiltration can be applied (Appendix 7). The evaporation front moves downward into the soil at a rate which depends on the depth itself and on the soil conditions:

$$dz_E/dt = c_3/z_E \qquad (4.28)$$

where c_3 is defined as:

$$c_3 = c_2 / ((\theta_i - \theta_{ad}) + c_1 / c_2) \qquad (4.29)$$

The variable c_1 is a function of initial moisture content θ_i, final moisture content θ_{ad} (air dry) of the top layer and the (liquid) soil water diffusivity $D(\theta)$:

$$c_1 = \int_{\theta_{ad}}^{\theta_i} D(\theta)\ (\theta_i - \theta)\ d\theta \qquad (4.30)$$

('initial', in this context, refers to the water content at the start of each DELT, and not to the value at the start of the simulation).

From Equation 4.29 it can be seen that the depth z_E increases rapidly, if c_1 is small compared to c_2, i.e. when vapour diffusion is rapid compared to liquid supply. It can be seen from Equations 4.29 and 4.30 that low θ_i results in high c_3 values and thus in rapidly increasing z_E.

The value of z_E is reset to zero when the topsoil is saturated by rain or rising groundwater.

Calculation of c1

Because it is not convenient to perform the integration of Equation 4.30 for every DELT again, the integral is approximated in terms of initial moisture content and some constants:

$$c_1 = A\,(e^{B(\theta_i/\theta_s - c_4)} - 1) \qquad (4.31)$$

The parameters A and B are soil-specific constants. c_4 is about 0.5 (dimensionless) for all soils, and is assumed constant in SAWAH; θ_s is the water content at saturation. Equation 4.31 was obtained by curve fitting on numerically determined integrals for many different soil types. As an alternative, an expression for $D(\theta)$ could be used such that the RHS of Equation 4.30 can be solved analytically. Generally, however, expressions for $D(\theta)$ also contain two parameters.

The above set of equations was developed for an 'ideal' soil, i.e. a deep soil with uniform initial moisture content. It can also be used as an approximation for a layered soil. The top compartment should be relatively thick (0.05-0.10 m), compared to maximum z_E levels. It can be shown that, for homogeneous profiles, the current formulation simplifies to a relation of the type:

$$e = S_d\, t^{1/2} \qquad (4.32)$$

where S_d is the *desorptivity*, in analogy with the *sorptivity* that is used to characterize infiltration (Appendix 7).

Subroutine SUZECA; parameter values

The calculations based on Equations 4.28 - 4.30 are performed in the subroutine SUZECA. Indicative values for the relevant parameters are 0.1 $cm^2\ d^{-1}$ for c_2, 0.005-0.5 $cm^2\ d^{-1}$ for A, and 5-15 (dimension-less) for B. Realistic combinations of A and B should yield c_1 values ranging from 5.0 to 50.0 $cm^2\ d^{-1}$ with exceptions up to 400 $cm^2\ d^{-1}$. Coarse soils tend to have high B values combined with low A values; low B and intermediate A values hold for silty soils, and low A and low B values for very fine-textured soils. c_1 and c_3 are assumed constant during each DELT (e.g., day).

4.3.4 Transpiration

The quantity of water used for transpiration is introduced as an input to SAWAH, per DELT and per compartment. It is recommended that transpiration be calculated in a separate subroutine, called by the MAIN *before* the water balance calculations. No associated theory is discussed here because transpiration is not calculated in SAWAH.

Actual and potential transpiration

Actual transpiration is usually calculated from potential transpiration, and soil and root conditions. Potential transpiration depends on atmospheric demand (radiation, wind, vapour pressure deficit) and leaf area. Soil and root conditions are used to reduce potential transpiration to actual transpiration. Various approaches may be used to characterize the root system: rooted depth, total root length, root length density, etc. The distribution of the roots over the various compartments, their activity, and the distribution of moisture determine the distribution of uptake over depth.

Relation with soil water

The availability of soil water may be expressed on the basis of volumetric soil moisture content θ, or soil water potential h. The latter seems more sound theoretically, since individual roots experience a potential rather than a moisture content, and the relation *uptake vs h* should be more universal than relations between *uptake* and θ. If, on the other hand, uptake is transport-limited, water content could play an important role again, because of its direct effect on hydraulic conductivity; hence it is difficult to decide which is to be preferred as a measure of water availability, h or θ. The formulation of uptake offered by De Willigen and Van Noordwijk (1991) avoids this choice because it takes into account root pressure, root conductivity and soil conductivity, root length density, and moisture content and pressure head in the bulk soil.

Impulse and alternatives

Whereas uptake is a gradual process in reality, uptake for transpiration in SAWAH is instantaneously subtracted from the amount of stored moisture in each compartment. Thus, the model treats uptake as an *impulse*, activated at the start of each DELT. Again, this formulation is a simplification. Two alternatives may seem more appropriate:

(1) converting the amounts extracted per DELT to DELT-averaged extraction rates, and introducing these rates into the flow equation executed at each time interval DT;

(2) allowing a continuous interaction between root water uptake and soil water flow, i.e. not only considering uptake as a sink term in the flow equation, but also adjusting its rate after each DT on the basis of crop-soil interaction.

Formulation (2) generally offers no practical alternative, because the *diurnal course* of extraction rate is usually unknown. Moreover, introducing crop variables into soil modules interferes with the modular approach. Alternative (1) represents an extreme that is the opposite of the current impulse formulation. It is no more realistic, however, in view of the diurnal pattern of extraction. Moreover, it may lead to another problem: as a consequence of *uncoupling* calculation of the extraction rate from the flow calculations, the situation could arise where the predefined extraction rate cannot be maintained throughout the duration of DELT. This would happen when, in the course of DELT, so much water is being removed from a given compartment as a result of flow, that no water is left for root extraction. At any given time during a simulation run it cannot be assessed *a priori* whether this complication will arise.

The current formulation avoids this because it subtracts the total amount corresponding to DELT from the amount of water in each compartment only to the extent available at the start of each DELT.

4.3.5 Groundwater, perched water, capillary rise, and drainage.

Groundwater depth

The type of boundary condition at z_{lb}, the depth of the lower profile boundary, can be selected by the switch SWIT7 (Chapter 3). When a *pressure head lower boundary condition* is chosen, groundwater depth with respect to the surface (z_w) is introduced as a dynamic external variable, a forcing function. z_w is called the *true groundwater* depth (as opposed to perched groundwater), and is obtained from observation wells as the actual piezometer reading with respect to the surface. One single value of z_w applies per period DELT. SAWAH interpolates on a linear scale between pairs of *time*-z_w specified in the input file FILIN (Section 2.5). Preferably *daily* values of z_w are specified.

Pressure head at z_{lb}

Groundwater level is not used directly in flow calculations. Pressure head at z_{lb}, the depth of user-defined lower boundary of the soil profile, is used as the actual boundary condition to the system. It is calculated from z_w by assuming hydrostatic equilibrium between z_{lb} and z_w: i.e., h at z_{lb} is equal to z_{lb}-z_w, irrespective of the position of z_w relative to z_{lb}.

As a consequence of the above assumption, z_{lb} is preferably equal to the installation depth of the observation well that provides the input data. The water within a piezometer is, by definition, in hydrostatic equilibrium. The height of the water column in the piezometer above the bottom of the tube is equal to the pressure head at its bottom. It is also equal to the pressure head in the soil at the tube's end. One cannot be sure, however, whether hydrostatic equilibrium exists within the soil around the tube. Thus, equating z_{lb}-z_w to h at z_{lb} introduces errors if the bottom of the tube is situated much deeper or shallower than z_{lb}. (Note that this problem is not specific for SAWAH; it is implied in any soil water model that takes groundwater into account.)

Groundwater in profile

When z_w is *smaller* than z_{lb}, groundwater is said to be *within* the profile. Essentially there is no difference with groundwater *below* the profile. It should be realized, however, that the use of a pressure head boundary condition has a peculiar consequence: the depth of the upper boundary of the deepest saturated section of the profile does not necessarily coincide with z_w. Equality between the two is attained after equilibrium has been established between water in the soil and z_w (i.e. water in the observation tube), which may take some time. Exact matching of the two is only possible if z_w coincides with a compartment boundary.

Groundwater as model output?

It would be attractive to calculate groundwater depth *as a result* of simulated flow processes, i.e. as an output rather than input. Unfortunately, the regional geometry of the system is often too complex. Especially in ricelands, varying relative field elevations, irregular valley shapes and variations in subsoil properties prohibit any such calculations as the lateral flow component cannot be estimated with any accuracy. Values of the phreatic level obtained from observation wells are the only realistic way of incorporating groundwater.

Air entry

The model does not include provisions for negative pressure heads in saturated soil. For situations where the air entry value of the soil is high, e.g. exceeding 10 cm, this assumption leads to errors. In structured soils, high air entry values will seldom be encountered. In puddled rice soils, on the other hand, they are not hypothetical. Further investigation is needed to assess the relevance of such details to field water balance simulation. A possible approach could be to lower z_w by an amount equal to the air entry value; other complications, however, would arise in that case.

Perched and true groundwater; choice of z_{lb}

If a saturated section of the soil profile overlies an unsaturated section, it is referred to as a *perched water table*. The only difference between *perched* and *true* groundwater is that in the *perched* situation, the lower boundary of the saturated soil section is *within* the user-defined soil profile. As a consequence, the pressure head at this lower boundary is zero for perched water, whereas it equals z_{lb}-z_w with *true* groundwater within the profile. If the bottom of the deepest saturated section coincides with the lower profile boundary, then the pressure head at this boundary may be positive, zero or negative, depending only on z_w.

All groundwater reservoirs, shallow and deep, result from accumulation of water above less permeable layers (or, above liquids of higher density, e.g. saline water). Afterall, when considering the real world without making attempts to measure, model or schematize, there is no difference between *perched* and *true* groundwater. It is clear, therefore, that the terminology used here hinges on the choice of the lower system boundary. Observation wells, used for recording the input z_w should be deeper than the chosen z_{lb}. Obviously, however, most users of the model will define z_{lb} depending on available z_w readings. Problems can be expected when the soil near the lower end of the tube is very impermeable and z_{lb} deviates substantially from the depth of the tube.

Observations from additional wells shallower than z_{lb} are suitable for validation purposes: to verify the existence of perched water tables predicted.

Free drainage

When the free drainage option is selected (Chapter 3), the lower boundary condition is a *constant force* (gravity) condition: water will drain at a rate equal to $k(h)$. If groundwater enters the profile, SAWAH automatically switches to the *pressure head* boundary condition.

Fluxes: drainage and capillary rise

Quantification of the supply of water from a water table to the root zone is important in evaluating upland crop production potentials, and lowland potentials in the dry season. The flux over the lower boundary is also relevant to solute transport.

In contrast to the other terms of the water balance, the flux over the lower profile boundary (*drainage/capillary rise*) is calculated at each DT, applying the flow equation, either for saturated or unsaturated conditions, as outlined earlier (Sections 4.2.2 and 4.2.3). Its calculation is an integral part of the flow calculations in SUUNST, SUSTFL and SUSEFL. Except for the direction of flow, the governing processes are the same for drainage and capillary rise. If the *pressure head boundary* option is selected and the bottom compartment of the profile is unsaturated, the flux over the lower boundary is calculated as the linearized gradient of matric flux potential. This gradient is evaluated from matric flux potential at the boundary (with h derived from z_w) and at the centre of the bottom compartment (with h derived from θ). If the bottom compartment is saturated, the flux over the lower boundary is computed by the procedure explained in Section 4.2.2. Again, h at the boundary is then derived from z_w.

4.3.6 Artificial drainage

Drainage by artificial systems buried in the soil is modelled in a simple manner. The soil compartment containing the tubes or mole holes (drained layer) is identified by the variable IDRAIN: for IDRAIN=n, drains are assumed to be positioned at the centre of the n^{th} soil compartment. At each DT, the *excess water* is removed from the drained layer by the subroutine *SUSAWA*. This excess is defined as a fixed fraction of saturated water content of that particular layer. In the current version of SAWAH this fraction is 0.1.

Delay

The current formulation assumes that the drainage system is sufficiently 'dense' to prevent saturation of the drained layer. In reality, outflow rate may be limited because of drain spacing, as a function of soil properties, z_W, etc. This phenomenon (delay) can be included in a later version of the model.

Other models

Other models of artificial drainage are usually based on modification of groundwater depth, resulting from drainage. It should be kept in mind, however, that a model applicable to lowland ricelands should also be able to handle drainage of water entering from above (perched water). Although drainage of ricelands seems an irrelevant topic, it is currently becoming increasingly important. One of the reasons is the increase in world acreage under rice-wheat rotation where waterlogging may be a constraint to wheat productivity; another is the recognition that percolation rate in lowland rice has a strong bearing on the functioning of roots (nutrient uptake).

5. SAWAH subroutines

```
*----------------------------------------------------------------------*
*  SUBROUTINE  SUCONV                                                  *
*                                                                      *
*  Authors: Hein ten Berge and Don Jansen                              *
*  Date   : 1-MRT-1992                                                 *
*  Version: 2.0                                                        *
*                                                                      *
*  Purpose: SUCONV converts units (main to submodel and vv.).          *
*           In the main program, length is in mm or m, time in d.      *
*           In the submodel (SUSAWA plus underlying subroutines), length *
*           (distance, water layer and head) is in cm, time in d.      *
*                                                                      *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)     *
*  name   type meaning                                   units  class *
*  ----   ---- -------                                   -----  ----- *
*  SWIT2   I4  switch for conversion                       -     C,I  *
*            =1 from main to local                                     *
*            =2 from local to main                                     *
*                                                                      *
*             Note: Units of argument variables are changed            *
*                   by SUCONV. The direction of change depends         *
*                   on the value of SWIT2:                             *
*             a: Input at SWIT2 = 1, Output at SWIT2 = 2               *
*             b: Output at SWIT2 = 1, Input at SWIT2 = 2               *
*                                                                      *
*  TKL     R4  thickness of soil compartment              a:  m    I,O  *
*                                                         b: cm    I,O  *
*  ZL      R4  depth of compartment interface             a:  m    I,O  *
*                                                         b: cm    I,O  *
*  ZLT     R4  total depth of profile                     a:  m    I,O  *
*                                                         b: cm    I,O  *
*  RAIN    R4  rainfall/irrigation rate                   a: mm/d  I,O  *
*                                                         b: cm/d  I,O  *
*  ZW      R4  depth of free groundwater level below soil a:  m    I,O  *
*              surface                                    b: cm    I,O  *
*  ZE      R4  depth of evaporation front below soil      a:  m    I,O  *
*              surface                                    b: cm    I,O  *
*  WL0     R4  depth of water layer on field              a:  m    I,O  *
*                                                         b: cm    I,O  *
*  WL0MX   R4  maximum surface water storage              a:  m    I,O  *
*                                                         b: cm    I,O  *
*  RUNOF   R4  runoff rate                                a: mm/d  I,O  *
*                                                         b: cm/d  I,O  *
*  TRWL    R4  uptake per compartment for                 a: mm/d  I,O  *
*              transpiration                              b: cm/d  I,O  *
*  EVSW    R4  actual soil evaporation rate               a: mm/d  I,O  *
*                                                         b: cm/d  I,O  *
*  EVSC    R4  potential soil evaporation rate            a: mm/d  I,O  *
*                                                         b: cm/d  I,O  *
*  UPRISE  R4  water flux over lower profile boundary     a: mm/d  I,O  *
*                                                         b: cm/d  I,O  *
*  DRAIQT  R4  water loss rate by artificial drainage     a: mm/d  I,O  *
*                                                         b: cm/d  I,O  *
*  FLXQT   R4  DELT-averaged flux over compartment        a: mm/d  I,O  *
*              interface                                  b: cm/d  I,O  *
*  NL      I4  number of soil compartments                  -     I    *
*                                                                      *
*  WARNINGS : none                                                     *
*                                                                      *
*  OPTIONS:    see input SWIT2                                         *
*                                                                      *
*  SUBROUTINES called : none                                           *
*                                                                      *
*  FUNCTIONS called : none                                             *
*                                                                      *
*  FILE usage : none                                                   *
*----------------------------------------------------------------------*

      SUBROUTINE SUCONV (SWIT2,TKL,ZL,ZLT,RAIN,ZW,ZE,WL0,WL0MX,
     $                   RUNOF,TRWL,EVSW,EVSC,UPRISE,DRAIQT,FLXQT,NL)

      IMPLICIT  REAL (A-Z)
      INTEGER   I, NL, SWIT2
      DIMENSION FLXQT(11), TKL(10), ZL(10), TRWL(10)

*-----variables retain their values between subsequent calls
*     of this subroutine
      SAVE

*-----conversion factors
      F1 = 0.01
      F2 = 10.0
      IF (SWIT2.EQ.1) F1 = 1. / F1
      IF (SWIT2.EQ.1) F2 = 1. / F2

*-----conversion
      DO 10 I=1,NL
         TKL(I)   =  TKL(I) * F1
         ZL(I)    =   ZL(I) * F1
         TRWL(I)  = TRWL(I) * F2
         FLXQT(I) = FLXQT(I) * F2
10    CONTINUE
      FLXQT(NL+1) = FLXQT(NL+1) * F2
      ZE     = ZE     * F1
      ZLT    = ZLT    * F1
      ZW     = ZW     * F1
      WL0    = WL0    * F1
      WL0MX  = WL0MX  * F1
      RAIN   = RAIN   * F2
      EVSW   = EVSW   * F2
      EVSC   = EVSC   * F2
      RUNOF  = RUNOF  * F2
      UPRISE = UPRISE * F2
      DRAIQT = DRAIQT * F2

      RETURN
      END
```

```
*----------------------------------------------------------------------*
*  SUBROUTINE SUERR                                                    *
*                                                                      *
*  Author : Don Jansen and Hein ten Berge                              *
*  Date   : 1-MRT-1992                                                 *
*  Version: 2.0                                                        *
*                                                                      *
*  Purpose: SUERR checks whether value of variable X is within         *
*           pre-specified domain                                       *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)     *
*  name   type meaning                                   units  class *
*  ----   ---- -------                                   -----  ----- *
*  MNR     R4  message number                           variable   I  *
*  X       R4  value of variable to be checked          variable   I  *
*  XMIN    R4  minimum allowable value of X             variable   I  *
*  XMAX    R4  maximum allowable value of X             variable   I  *
*                                                                      *
*  WARNINGS:                                                           *
*                                                                      *
*     X < XMIN * 0.99 and XMIN .NE. -99 then expert message is produced *
*     X > XMAX * 1.01 and XMAX .NE. -99 then expert message is produced *
*                                                                      *
*  SUBROUTINES called : none                                           *
*                                                                      *
*  FUNCTIONS called :                                                  *
*                                                                      *
*  FILE usage : none                                                   *
*----------------------------------------------------------------------*

      SUBROUTINE SUERR (IMNR,X,XMIN,XMAX)

      IMPLICIT REAL (A-Z)
      INTEGER  IUNLOG,IMNR
      CHARACTER*1 DUMMY
      CHARACTER*38 ERRM(5)

*-----common block
      COMMON /UNITNR/ IUNLOG

*-----variables retain their values between subsequent calls
*     of this subroutine
      SAVE

      DATA ERRM/' MATRIC SUCTION OUT OF RANGE IN SUMSKM',
     $          ' WATER CNT OUT OF RANGE IN SUSLIN     ',
     $          ' WATER CNT OUT OF RANGE IN SUWCMS     ',
     $          ' MATRIC SUCTION OUT OF RANGE IN SUWCMS',
     $          ' ONE OR MORE TRWL(I) OUT OF RANGE     '/

      IF ((X.LT.XMIN*0.99).AND.(XMIN.NE.-99.)) GOTO 10
      IF ((X.GT.XMAX*1.01).AND.(XMAX.NE.-99.)) GOTO 10
      RETURN

10    CONTINUE
      WRITE (*,20) IMNR, X, XMIN, XMAX
      WRITE (*,30) ERRM(IMNR)
      IF (IUNLOG.GT.0) THEN
         WRITE (IUNLOG,20) IMNR, X, XMIN, XMAX
         WRITE (IUNLOG,30) ERRM(IMNR)
      ENDIF
      READ (*,'(A)') DUMMY
      STOP
20    FORMAT (//,' ***fatal error in variable or parameter value ***',
     $        /,'    message number, value, minimum and maximum: ',/,
```

```
     $        10X,I2,3(3X,E10.3))
 30   FORMAT(A)
      END

*----------------------------------------------------------------------*
* SUBROUTINE SUGRHD                                                    *
*                                                                      *
* Authors: Hein ten Berge and Don Jansen                               *
* Date    : 1-MRT-1992                                                 *
* Version: 2.0                                                         *
*                                                                      *
* Purpose: SUGRHD calculates the gravitational component of hydraulic  *
*          head at compartment interfaces, with respect to the surface *
*          where this component is zero.                               *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name   type meaning                                    unit   class  *
* ----   ---- -------                                    -----  -----  *
* TKL     R4  thickness of soil compartments              cm      I    *
* NL      R4  number of soil compartments                  -      I    *
* HGT     R4  gravitational head at top of compartment    cm      O    *
* HGB     R4  gravitational head at bottom of compartment cm      O    *
*                                                                      *
* SUBROUTINES called : none                                            *
*                                                                      *
* FUNCTIONS called : none                                              *
*                                                                      *
* FILE usage : none                                                    *
*----------------------------------------------------------------------*

      SUBROUTINE SUGRHD (TKL,NL,HGT,HGB)

      DIMENSION  TKL(10), HGT(10), HGB(10)

*-----variables retain their values between subsequent calls
*     of this subroutine
      SAVE

*-----distance from surface to top/bottom of compartment I
      HGT(1) = 0.
      HGB(1) = -TKL(1)
      DO 10 I=2,NL
         HGT(I) = HGT(I-1) - TKL(I-1)
         HGB(I) = HGB(I-1) - TKL(I)
 10   CONTINUE
      RETURN
      END

*----------------------------------------------------------------------*
* SUBROUTINE SUINTG                                                    *
*                                                                      *
* Authors: Hein ten Berge and Don Jansen                               *
* Date    : 1-MRT-1992                                                 *
* Version: 2.0                                                         *
*                                                                      *
* Purpose: SUINTG calculates the internal timestep DT, and             *
*          performs dummy integration.                                 *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name   type meaning                                    units  class  *
* ----   ---- -------                                    -----  -----  *
* HPBP    R4  pressure head at bottom of profile          cm      I    *
* FLX     R4  soil water flux over compartment interface  cm/d    I    *
* TKL     R4  thickness of soil compartment               cm      I    *
* WCL     R4  volumetric soil water content                -     I,O   *
* MS      R4  matric suction                              cm      I    *
* DTMIN   R4  relative minimum timestep                    d      I    *
* DTMX1   R4  absolute maximum timestep                    d      I    *
* DTMX2   R4  relative maximum timestep                    d      I    *
* DTFX    R4  fixed time step size                         d      I    *
* INXSAT  I4  index of saturated compartment (top=1)       -      I    *
* DHH     R4  hydraulic head difference over                           *
*             saturated section                           cm      I    *
* JTOT    I4  index of saturated set (top=1)               -      I    *
* FLXSQ2  R4  prospective water flux: resulting after                  *
*             including extra overlying compartment in                 *
*             saturated set                               cm/d    I    *
* WL0     R4  depth of surface water layer                cm     I,O   *
* WL0MX   R4  maximum depth of surface water layer        cm      I    *
* NL      I4  number of soil compartments                  -      I    *
* IDRAIN  I4  index of tube-/mole- drained compartment     -      I    *
* ZW      R4  depth of groundwater                        cm      I    *
* FLUXDT  R4  flux at compartment interface integrated over            *
*             time                                        cm     I,O   *
* DRAIDT  R4  tube-/mole- drainage integrated over time   cm     I,O   *
* DELT    R4  time step in main                            d      I    *
* DT      R4  time step for internal (=dummy) integration  d      O    *
*                                                                      *
* OPTIONS: SWIT5=1   variable time step                                *
*          SWIT5=2   fixed time step                                   *
*                                                                      *
* SUBROUTINES called :                                                 *
*                                                                      *
*    - SUWCMS                                                          *
*                                                                      *
* FUNCTIONS called : none                                              *
*                                                                      *
* FILE usage : none                                                    *
*----------------------------------------------------------------------*

      SUBROUTINE SUINTG (HPBP,FLX,TKL,WCL,MS,DTMIN,DTMX1,DTMX2,DTFX,
     $                  INXSAT,DHH,JTOT,JJTOT,FLXSQ1,FLXSQ2,WL0,NL,
     $                  IDRAIN,WL0MX,ZW,FLUXDT,DRAIDT,DELT,DT)

      IMPLICIT REAL (A-Z)
      INTEGER  NL, I, IX, ITEL1, ITEL2, INXSAT, IDRAIN, JTOT, JJTOT
      INTEGER  LABELA, LABELB

      DIMENSION WCL(10), INXSAT(10,10), FLUXDT(11), JJTOT(10)
      DIMENSION WCLRCH(10), FLX(11), TKL(10), MS(10), DHH(10)

*-----common blocks
*     only swit5 used in SUINTG
      INTEGER SWIT3, SWIT5, SWIT7, SWIT8, SWIT9, SWIT10
      COMMON /SWITCH/ SWIT3,SWIT5,SWIT7,SWIT8,SWIT9,SWIT10
*     WCFC and WCWP dummy in SUINTG
      COMMON /VOLWAT/ WCAD(10),  WCFC(10),  WCST(10),  WCWP(10)

*-----variables retain their values between subsequent calls
*     of this subroutine
      SAVE

      DATA TINY1,TINY2,TINY3/1.0E-5, 0.001, 0.01/

*-----dummy values wix
      WIX1=0.
      WIX2=0.

*-----timestep permitted to just remove surface water
      HPTP = AMIN1(WL0,WL0MX)
      IF(FLX(1).GT.TINY2) THEN
         IF(HPTP.LE.TINY2) THEN
            HPTP=0.
            DTSRF = DELT
         ELSE
            DTSRF = AMAX1 (TINY1, HPTP/FLX(1))
         ENDIF
      ELSEIF(FLX(1).LT.-TINY2) THEN
            DIFLEV=AMAX1(0.,(-ZW-HPTP))
            DTSRF=-DIFLEV/FLX(1)
      ELSE
         DTSRF=DELT
      ENDIF
      DT = AMAX1(0.,AMIN1 (DELT,DTSRF,DTMX1,DTMX2))

*-----timestep allowed to just reach saturation, or to just reach
*     air-dry condition of any compartment
      DO 10 I=1,NL
         WCLRCH(I) = (FLX(I)-FLX(I+1)) / TKL(I)
         IF (WCLRCH(I).LT.-TINY1)
     $      SATTIM =-(WCL(I)-WCAD(I))/WCLRCH(I)
         IF (WCLRCH(I).GT. TINY1)
     $      SATTIM = (WCST(I)-WCL(I))/WCLRCH(I)
         IF (ABS(WCLRCH(I)).LT.TINY1) THEN
            SATTIM = DELT
            WCLRCH(I)=0.
         ENDIF
         DT = AMIN1(SATTIM,DT)
10    CONTINUE


      LABELA=0
      LABELB=0

      IF(JTOT.EQ.0) THEN


         IF(ABS(WCLRCH(NL)).LT.TINY1) THEN
            CONTINUE
         ELSEIF(HPBP.GT.0.) THEN
            IF(HPBP.GE.TKL(NL)) THEN
               CONTINUE
```

```
         ELSE
*-----      groundwater level in last compt; set dt such that equilibrium
*           is reached within one DT;
            MSAL=TKL(NL)/2.
            MSACT=MS(NL)
            CALL SUWCMS(NL,2,WIX2,MSAL)
            CALL SUWCMS(NL,2,WIX1,MSACT)
            SIGN=(WIX2-WIX1)*WCLRCH(NL)
            IF(SIGN.GT.TINY1) DT1=(WIX2-WIX1)/WCLRCH(NL)
            IF(SIGN.LE.TINY1) DT1=DTMIN
            DT=AMIN1(DT,DT1)
            LABELA=NL
            LABELB=NL+1
         ENDIF
      ENDIF


   ELSEIF(INXSAT(JTOT,JJTOT(JTOT)).NE.NL)  THEN

*-----  deepest compartment not saturated; check situation in NL
      IF(ABS(WCLRCH(NL)).LT.TINY1) THEN
         CONTINUE
      ELSEIF(HPBP.GT.0.) THEN
         IF(HPBP.GE.TKL(NL)) THEN
            CONTINUE
         ELSE
*-----      groundwater level in last compt; set dt such that equilibrium
*           is reached within one DT;
            MSAL=TKL(NL)/2.
            MSACT=MS(NL)
            CALL SUWCMS(NL,2,WIX2,MSAL)
            CALL SUWCMS(NL,2,WIX1,MSACT)
            SIGN=(WIX2-WIX1)*WCLRCH(NL)
            IF(SIGN.GT.TINY1) DT1=(WIX2-WIX1)/WCLRCH(NL)
            IF(SIGN.LE.TINY1) DT1=DTMIN
            DT=AMIN1(DT,DT1)
            LABELA=NL
            LABELB=NL+1
         ENDIF
      ENDIF

   ELSE

*-----  cmptmnt NL is part of sat set; if upper boundary of deepest
*       saturated set is within TKL from equilibrium
*       groundwater level, force quick equilibrium to avoid oscillations
        IF(DHH(JTOT).LT.-TINY3) THEN
*-----      note that jtot is always the deepest saturated set!
*           no equilibrium case; downward flow in saturated set
           IF(INXSAT(JTOT,1).EQ.NL) THEN
              IF(ABS(WCLRCH(NL)).LT.TINY1) THEN
                 CONTINUE
              ELSEIF(HPBP.GT.0.) THEN
                 IF(HPBP.LT.TKL(NL)) THEN
                   MSAL=TKL(NL)/2.
                   MSACT=MS(NL)
                   CALL SUWCMS(NL,2,WIX2,MSAL)
                   CALL SUWCMS(NL,2,WIX1,MSACT)
                   SIGN=(WIX2-WIX1)*WCLRCH(NL)
                   IF(SIGN.GT.TINY1) DT1=(WIX2-WIX1)/WCLRCH(NL)
                   IF(SIGN.LE.TINY1) DT1=DTMIN
                   DT=AMIN1(DT,DT1)
                   LABELA=NL
                   LABELB=NL+1
                 ENDIF
              ENDIF
           ELSE
              IX=INXSAT(JTOT,1)
              IF(ABS(WCLRCH(IX)).LT.TINY1) THEN
                 CONTINUE
              ELSEIF(FLXSQ1.LE.-TINY1) THEN
                 MSAL=TKL(IX)/2.
                 MSACT=MS(IX)
                 CALL SUWCMS(IX,2,WIX2,MSAL)
                 CALL SUWCMS(IX,2,WIX1,MSACT)
                 SIGN=(WIX2-WIX1)*WCLRCH(IX)
                 IF(SIGN.GT.TINY1) DT1=(WIX2-WIX1)/WCLRCH(IX)
                 IF(SIGN.LE.TINY1) DT1=DTMIN
                 DT=AMIN1(DT,DT1)
                 LABELA=IX
                 LABELB=IX+1
              ENDIF
           ENDIF

        ELSE
```

```
              IX=INXSAT(JTOT,1)-1
              IF(IX.LT.1) THEN
*----            shallowest compt of deepest saturated set is surface compt
                 CONTINUE
              ELSEIF(ABS(WCLRCH(IX)).LT.TINY1) THEN
                 CONTINUE
              ELSEIF(FLX(IX+1).GT.-0.1) THEN
                 CONTINUE
              ELSEIF(FLXSQ2.LE.TINY2) THEN
*                including extra compartment in sat set would not lead
*                to inversion of flow direction
                 CONTINUE
              ELSE
*                including extra compartment cannot be allowed
*                determine time step required to just reach equilibrium
                 MSAL=TKL(IX)/2.
                 MSACT=MS(IX)
                 CALL SUWCMS(IX,2,WIX2,MSAL)
                 CALL SUWCMS(IX,2,WIX1,MSACT)
                 SIGN=(WIX2-WIX1)*WCLRCH(IX)
                 IF(SIGN.GT.TINY1) DT1=(WIX2-WIX1)/WCLRCH(IX)
                 IF(SIGN.LE.TINY1) DT1=DTMIN
                 DT=AMIN1(DT,DT1)
                 LABELA=IX
                 LABELB=IX+1
              ENDIF
           ENDIF
        ENDIF


*-----time step option
      IF (SWIT5.EQ.2) THEN
*-----   fixed time step option
         DT=AMIN1(DT,DTFX)
         GOTO 100
      ELSE
*-----   variable time step option
         CONTINUE
      ENDIF

****************** DT reduction loop ***********************
*-----variable timestep as limited by intermediate compartments
*     flow I over interface between compartments I-1 (upper) and
*     I (lower)
      DO 70 I=2,NL
*-----   loop while DT too large
         GOTO 60
 50      DT = DT/2.
 60         ITEL1 = 0
            ITEL2 = 0
            IF(I.EQ.LABELA) GOTO 70
            IF(I.EQ.LABELB) GOTO 70
            IF (DT.LT.DTMIN) GOTO 100
            IF (ABS(WCLRCH(I-1)).LT.TINY1.AND.ABS(WCLRCH(I)).LT.TINY1)
     $         GOTO 70
*-----         no changes
            IF (ABS(WCLRCH(I-1)).LT.TINY1) THEN
*-----         no changes in upper compartment
               IF (ABS(WCL(I-1)-WCST(I-1)).LT.TINY3) THEN
*-----            upper compartment saturated
                  DELZ  = 0.5*TKL(I)
                  MST1  = 0.
                  ITEL2 = 1
               ELSE
*-----            upper compartment unsaturated
                  DELZ  = 0.5*(TKL(I-1)+TKL(I))
                  MST1  = MS(I-1)
               ENDIF
            ELSE
*-----         changes in upper compartment
               DELZ  = 0.5*(TKL(I-1)+TKL(I))
               WCT1  = WCL(I-1)+DT*WCLRCH(I-1)
               CALL SUWCMS (I-1,1,WCT1,MST1)
            ENDIF

            IF (ABS(WCLRCH(I)).LT.TINY1) THEN
*-----         no changes in lower compartment
               IF (ABS(WCL(I)-WCST(I)).LT.TINY3) THEN
*-----            lower compartment saturated
                  DELZ = 0.5*TKL(I-1)
                  MST2 = 0.
               ELSE
*-----            lower compartment unsaturated
                  DELZ = 0.5*(TKL(I-1)+TKL(I))
                  MST2 = MS(I)
                  IF(ITEL2.EQ.1) DELZ=0.5*TKL(I)
```

```
            ENDIF
            IF (FLXSQ2.LE.TINY2)  ITEL1 = 1
*-----      enough pressure to capture upper next compartment
*           into saturation
         ELSE
*-----      changes in lower compartment
            DELZ = 0.5 * (TKL(I-1) + TKL(I))
            IF (ITEL2.EQ.1) DELZ = 0.5 * TKL(I)
            WCT2 = WCL(I) + DT * WCLRCH(I)
            CALL SUWCMS (I,1,WCT2,MST2)
         ENDIF
         DH = -MST2+MST1-DELZ
         CH = -DH*FLX(I)
         IF (ITEL1.EQ.1) CH = +1.
*-----   in this case flow is forced irrespective of time constant
      IF (CH.LT.-TINY1) GOTO 50
70    CONTINUE
      GOTO 90

******************** third DT reducytion loop *********************
*-----variable timestep as limited by lowest compartment
*     loop while DT too large
 80  DT = DT / 2.
      IF (DT.LT.DTMIN) GOTO 100
 90   IF (ABS(WCLRCH(NL)).LT.TINY1) GOTO 100
      IF(NL.EQ.LABELA) GOTO 100
      ITEL1 = 0
      ITEL2 = 0
      WCT1  = WCL (NL) + DT * WCLRCH(NL)
      CALL SUWCMS (NL,1,WCT1,MST1)
      DELZ  = 0.5 * TKL(NL)
      IF (HPBP.GE.TKL(NL)) ITEL1  =1
      MST2=-HPBP
      DH = -MST2 + MST1 - DELZ
      CH = -DH * FLX(NL+1)
      IF (ITEL1.EQ.1) CH = +1.
      IF (CH.LT.-TINY1) GOTO 80
     CONTINUE
******************* DT reduction loops completed ******************

*-----rectangular integration; evaporation  was already subtracted
*     from WL0 and/or WCL at start of susawa; transpiration as well
 100 WL0   =WL0-DT*FLX(1)
     IF (WL0.LT.0.) WL0=0.
     DO 110 I=1,NL
        WCL(I) = WCL(I) + DT * WCLRCH(I)
 110 CONTINUE



*-----integration for averaging of fluxes
     DO 120 I=1,NL+1
        FLUXDT(I) = FLUXDT(I) + DT * FLX(I)
 120 CONTINUE
*-----water lost through artificial drains
     IF (IDRAIN.NE.0) THEN
        IF (WCL(IDRAIN).GT.0.9*WCST(IDRAIN)) THEN
           DRAIN = -(WCL(IDRAIN) -
     $              0.9 * WCST(IDRAIN)) * TKL(IDRAIN)
           WCL(IDRAIN) = WCL(IDRAIN) + DRAIN / TKL(IDRAIN)
        ELSE
           DRAIN = 0.
        END IF
     ELSE
        DRAIN = 0.
     ENDIF
     DRAIDT = DRAIDT + DRAIN
     RETURN
     END

*----------------------------------------------------------------------*
* SUBROUTINE SUMFLP                                                    *
*                                                                      *
* Authors: Hein ten Berge and Don Jansen                               *
* Date   : 1-MRT-1992                                                  *
* Version: 2.0                                                         *
*                                                                      *
* Purpose: SUMFLP calculates the matric flux potential for             *
*          compartment I at given matric suction.                      *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name   type meaning                                    units  class *
* ----   ---- -------                                    -----  ----- *
* I       I4  compartment index                            -      I   *
* MS      R4  matric suction                              cm      I   *
* MFLP    R4  matric flux potential                      cm2/d    O   *
*                                                                      *
* OPTIONS: SWIT3=1  analytical integral                                *
*          SWIT3=2  analytical/numerical integral                      *
*          SWIT3=3  numerical integral                                 *
*          SWIT3=4  analytical integral                                *
*          SWIT3=5  numerical integral                                 *
*                                                                      *
* SUBROUTINES called :                                                 *
*                                                                      *
*     - SUMSKM                                                         *
*                                                                      *
* FUNCTIONS called : none                                              *
*                                                                      *
* FILE usage : none                                                    *
*----------------------------------------------------------------------*

      SUBROUTINE SUMFLP (I,MS,MFLP)

      IMPLICIT REAL (A-Z)
      INTEGER  I, IG, IX

      DIMENSION MSI(8), XGAUS(3), WGAUS(3)

*-----common blocks
*     only swit3 used in SUMFLP
      INTEGER SWIT3, SWIT5, SWIT7, SWIT8, SWIT9, SWIT10
      COMMON /SWITCH/ SWIT3,SWIT5,SWIT7,SWIT8,SWIT9,SWIT10
      COMMON /POWER / PN(10)
      COMMON /HYDCON/ KMSMX(10), KMSA1(10), KMSA2(10), KST(10)

*-----variables retain their values between subsequent calls
*     of this subroutine
      SAVE

      DATA MSI   /0., 10., 50., 250., 750., 1500., 5000., 10000./
      DATA XGAUS /0.112702, 0.5, 0.887298/
      DATA WGAUS /0.277778, 0.444444, 0.277778/
      DATA TINY  /1.E-10/

      MFLP = 0.
      IF (MS.GT.TINY) THEN
         IF (SWIT3.EQ.1) THEN
*-----      simple Rijtema conductivity
            MFLP = (KST(I)/KMSA1(I))*(EXP(-KMSA1(I)*MS)-1.)
         ELSEIF(SWIT3.EQ.4) THEN
*-----      power function conductivity
            MFLP = (KST(I)/(PN(I)+1.))*(1.-MS**(PN(I)+1.))-KST(I)
         ELSEIF(SWIT3.EQ.2.AND.MS.LE.KMSMX(I)) THEN
*-----      extended Rijtema option, first branch
            MFLP = (KST(I)/KMSA1(I))*(EXP(-KMSA1(I)*MS)-1.)
         ELSE
*-----      numerical integration by Gauss procedure;
*           used for all conductivity options except simple Rijtema
*           and power function
            DO 30 IX=2,7
               DMFLP = 0.
               IF (MS.GT.MSI(IX-1)) THEN
                  MSX = AMIN1 (MSI(IX),MS)
                  IF (IX.EQ.2) THEN
                     DO 10 IG=1,3
*-----                  logarithmic three-point Gauss integration
*                       for MS below 10 cm
                        X = MSX * XGAUS(IG)
                        CALL SUMSKM (I,X,KMSX)
                        DMFLP = DMFLP + KMSX * WGAUS(IG)
 10                  CONTINUE
                     MFLP = MFLP - DMFLP * (MSX-MSI(IX-1))
                  ELSE
                     DO 20 IG=1,3
*-----                  logarithmic three-point Gauss integration
*                       for MS over 10 cm
                        X = MSX * (MSI(IX-1)/MSX) **XGAUS(IG)
                        CALL SUMSKM (I,X,KMSX)
                        DMFLP = DMFLP + X * KMSX * WGAUS(IG)
 20                  CONTINUE
                     IF (DMFLP.LE.0.0) GOTO 40
                     MFLP = MFLP - DMFLP * ALOG (MSX/MSI(IX-1))
                  ENDIF
               ENDIF
 30         CONTINUE
         ENDIF
      ENDIF
 40   CONTINUE
      RETURN
      END
```

```
*----------------------------------------------------------------------*
* SUBROUTINE SUMSKM                                                    *
*                                                                      *
* Authors: Hein ten Berge and Don Jansen                               *
* Date   : 1-MRT-1992                                                  *
* Version: 2.0                                                         *
*                                                                      *
* Purpose: SUMSKM calculates the hydraulic conductivity at             *
*          given suction for compartment I on the basis of chosen      *
*          option (SWIT3). For explanation of option switch see input  *
*          file or SAWAH manual.                                       *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name   type meaning                                   units  class *
* ----   ---- -------                                   -----  ----- *
* I       I4  compartment index                           -      I   *
* MS      R4  matric suction                              cm     I   *
* KMS     R4  hydraulic conductivity                      cm/d   O   *
*                                                                      *
* OPTIONS: SWIT3=1 simple Rijtema form                                 *
*          SWIT3=2 extended Rijtema form                               *
*          SWIT3=3 Van Genuchten form                                  *
*          SWIT3=4 power form                                          *
*          SWIT3=5 user-defined form                                   *
*                                                                      *
* SUBROUTINES called :                                                 *
*                                                                      *
*     - SUERR, SUWCMS                                                  *
*                                                                      *
* FUNCTIONS called :  none                                             *
*                                                                      *
* FILE usage : none                                                    *
*----------------------------------------------------------------------*

      SUBROUTINE SUMSKM (I,MS,KMS)

      IMPLICIT REAL (A-Z)
      INTEGER  I

*-----common blocks
*     only swit3 used in SUMSKM
      INTEGER SWIT3,  SWIT5, SWIT7, SWIT8, SWIT9, SWIT10
      COMMON /SWITCH/ SWIT3,SWIT5,SWIT7,SWIT8,SWIT9,SWIT10
      COMMON /NUCHT / VGN(10),   VGA(10),    VGR(10), VGL(10)
      COMMON /HYDCON/ KMSMX(10), KMSA1(10), KMSA2(10), KST(10)
      COMMON /VOLWAT/ WCAD(10),  WCFC(10),  WCST(10), WCWP(10)
      COMMON /POWER / PN(10)

*-----variables retain their values between subsequent calls
*     of this subroutine
      SAVE

      DATA TINY /1.E-10/
      DATA MSAD /1.E7/

*-----check input value MS
      IF(MS.LT.-TINY.OR.MS.GT.1.E8) THEN
         CALL SUERR (1,MS,0.,1.E8)
      ENDIF

      IF (MS.GE.MSAD-TINY) THEN
*-----   air dry
         KMS = 0.
      ELSE
*-----   calculate conductivity
         IF(SWIT3.EQ.1) THEN
*-----      simple Rijtema option
            KMS = KST(I) * EXP(-KMSA1(I) * MS)
         ELSEIF(SWIT3.EQ.2) THEN
*-----      full Rijtema option
            IF(MS.LE.KMSMX(I)) THEN
               KMS = KST(I) * EXP(-KMSA1(I) * MS)
            ELSE
               KMS = KMSA2(I) * (MS**(-1.4))
            ENDIF
         ELSEIF(SWIT3.EQ.3) THEN
*-----      Van Genuchten conductivity
            WCL=0.
*           dummy value; wcl is returned by suwcms!
            CALL SUWCMS (I,2,WCL,MS)
            VGM  = 1.0 - 1.0/VGN(I)
            WREL = (WCL-VGR(I)) / (WCST(I)-VGR(I))
            HLP1 = WREL**VGL(I)
            HLP2 = 1.0 - WREL**(1./VGM)
            HLP3 = 1.0 - HLP2**VGM
            KMS  = KST(I) * HLP1 * HLP3 * HLP3
         ELSEIF(SWIT3.EQ.4) THEN
*-----      power function conductivity
            IF(MS.LE.1.) KMS=KST(I)
            IF(MS.GT.1.) KMS=KST(I)*(MS**PN(I))
         ELSEIF(SWIT3.EQ.5) THEN
*           user can here specify preferred conductivity function;
*           the following two lines should be removed:
            WRITE(*,10)
            STOP
         ENDIF
         IF(KMS.LT.TINY) KMS = 0.
      ENDIF
10    FORMAT(///,' *** fatal error; option SWIT3=5 requires ',/,
     $         ' specification of conductivity function')
      RETURN
      END

*----------------------------------------------------------------------*
* SUBROUTINE SUSAWA                                                    *
*                                                                      *
* Authors: Hein ten Berge and Don Jansen                               *
* Date   : 1-MRT-1992                                                  *
* Version: 2.0                                                         *
*                                                                      *
* Purpose: SUSAWA serves as a framework for various operations related *
*          to the soil water balance:                                  *
*         -initialisation of main state variables                      *
*         -calculation of rates-of-change of main state variables      *
*          WCLQT, ZEQT, WL0QT (i.e. solving Darcy and Continuity       *
*          equations)                                                  *
*         -instantaneous and cumulative flux calculations              *
*         -solving the water balance equation                          *
*                                                                      *
* Note1 :  Units indicated for input and output variables refer to     *
*          values at input/output to/from SUSAWA, respectively.        *
*          Units may be different between the two SUCONV calls.        *
*                                                                      *
* Note2:   The main state variables WCLQT, WL0QT, ZEQT are renamed to  *
*          WCL, WL0, ZE, respectively. This is to avoid confusion      *
*          arising from integration over time at two levels.           *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name   type meaning                                   units class *
* ----   ---- -------                                   ----- ----- *
* SWIT1   I4  switch for task definition                  -     C,I *
*             =1 initialization                                      *
*             =2 rate calculation                                    *
* IUNLOG  I4  unit number for log file                    -      I  *
* WCLQT   R4  volumetric moisture content                 -      I  *
* WL0QT   R4  depth of surface water layer                m      I  *
* NL      I4  number of soil compartments                 -      I  *
* TRWL    R4  uptake rate from each compartment due to    mm/d   I  *
*             transpiration                                         *
* EVSC    R4  potential soil evaporation rate             mm/d   I  *
* RAIN    R4  rainfall rate                               mm/d   I  *
* ZW      R4  depth of  groundwater level below soil             I  *
*             surface                                     m      I  *
* TKL     R4  thickness of soil compartment               m      I  *
* TYL     R4  type of soil per compartment                -      I  *
* DELT    R4  integration interval in MAIN                d      I  *
* DTMIN   R4  relative minimum time step                  d      I  *
* DTMX1   R4  absolute maximum time step                  d      I  *
* DTFX    R4  fixed time step size                        d      I  *
* IDRAIN  I4  index of tube- or mole-drained soil                I  *
*             compartment                                 -      I  *
* WL0MX   R4  maximum surface water storage               m      I  *
* ZEQT    R4  depth of evaporation front                  m      I  *
* CSA     R4  soil evaporation parameter                  cm2/d  I  *
* CSB     R4  soil evaporation parameter                  -      I  *
* CSC2    R4  effective soil vapour diffusivity           cm2/d  I  *
* WCLCH   R4  rate of change of water content, averaged             *
*             over DELT                                   1/d    O  *
* WL0CH   R4  rate of change of surface water depth,                *
*             WL0, averaged over DELT                     m/d    O  *
* WCLEQI  R4  water content at hydrostatic equilibrium with         *
*             groundwater level                           -      O  *
* EVSW    R4  actual soil evaporation rate                mm/d   O  *
* RUNOF   R4  runoff rate                                 mm/d   O  *
* UPRISE  R4  water flux over lower profile boundary      mm/d   O  *
* DRAIQT  R4  DELT-averaged rate of tube-/mole- drainage  mm/d   O  *
* WCUMCH  R4  rate of change in amount of stored soil               *
*             water, averaged over DELT                   mm/d   O  *
* ZECH    R4  rate of change of evaporation front, averaged         *
*             over DELT                                   m/d    O  *
* ZLT     R4  total depth of profile                      m      O  *
* FLXQT   R4  DELT-averaged value of waterflux at compart- mm/d  O  *
```

```
*            ment interface                              mm/d    O *
*  PRHEAD()R4  pressure head at compartment center        hPa     O *
*  DTAV    R4  DELT-averaged DT value                      d      O *
*                                                                   *
*  OPTIONS: see input SWIT1                                         *
*                                                                   *
*  SUBROUTINES called :                                             *
*                                                                   *
*      - SUCONV                                                     *
*      - SUGRHD                                                     *
*      - SUSLIN                                                     *
*      - SUSTCH                                                     *
*      - SUSTHH                                                     *
*      - SUSTFL                                                     *
*      - SUUNST                                                     *
*      - SUSEFL                                                     *
*      - SUINTG                                                     *
*      - SUZECA                                                     *
*                                                                   *
*  FUNCTIONS called : none                                          *
*                                                                   *
*  FILE usage : none                                                *
*                                                                   *
*      - * (screen)                                                 *
*                                                                   *
*-------------------------------------------------------------------*

      SUBROUTINE SUSAWA (SWIT1,IUNLOG,WCLQT,WL0QT,NL,TRWL,EVSC,RAIN,ZW,
     $           TKL,TYL,DELT,DTMIN,DTMX1,DTFX,IDRAIN,WL0MX,ZEQT,
     $           CSA,CSB,CSC2,WCLCH,WL0CH,WCLEQI,EVSW,RUNOF,UPRISE,
     $           DRAIQT,WCUMCH,ZECH,ZLT,FLXQT,PRHEAD,DTAV)

      IMPLICIT REAL (A-Z)
      INTEGER  SWIT1, IUNLOG, IULOG, ICOUNT
      INTEGER  NL, I, INXSAT, JJTOT, JTOT, J, JJ, IDRAIN

      DIMENSION INXSAT(10,10), JJTOT(10), WCL(10), TKL(10), MS(10)
      DIMENSION FLXSTT(11), FLXUNT(11), FLX(11), FLUXDT(11), FLXQT(11)
      DIMENSION HGT(10), HGB(10), DHH(10), TRWL(10), TYL(10), ZL(10)
      DIMENSION WCLQT(10), WCLCH(10), WCLEQI(10), PRHEAD(10), DELH(10)

*-----common blocks
      COMMON /UNITNR/ IULOG
      COMMON /VOLWAT/ WCAD(10),  WCFC(10),  WCST(10),  WCWP(10)

*-----variables retain their values between subsequent calls
*     of this subroutine
      SAVE
      DATA CONV /1000./
      DATA TINY /1.E-10/
      DATA TINY1  /1.E-3/
      DATA TINY2  /1.E-4/
      DATA TINYJP /5.E-2/

*-----unit number renamed for common
      IULOG=IUNLOG

*-----variables marked "x"QT are integrated with DELT steps in main
*     program, but "x" is dummy-integrated in SUSAWA with small steps DT
      DO 10 I=1,NL
         IF (SWIT1.EQ.1) ZL(I) = 0.
         WCLCH(I) = 0.
         WCL(I) = WCLQT(I)
 10   CONTINUE
      WL0CH = 0.
      WL0  = WL0QT
      ZECH = 0.
      ZE   = ZEQT

*-----cumulatives over DELT; set to zero
      EVSW   = 0.
      UPRISE = 0.
      RUNOF  = 0.
      DRAIDT = 0.
      DO 15 I = 1, NL+1
         FLUXDT(I) = 0.
15    CONTINUE

*-----conversion of units
      CALL SUCONV (1,TKL,ZL,ZLT,RAIN,ZW,ZE,WL0,WL0MX,RUNOF,TRWL,
     $     EVSW,EVSC,UPRISE,DRAIQT,FLXQT,NL)

*-----initialization/rate calculation
      IF (SWIT1.EQ.1) THEN
*-----   susawa called from initial section of main
*-----   gravitational component of hydraulic head at interfaces
         CALL SUGRHD (TKL,NL,HGT,HGB)
*-----   equilibrium moisture contents; depth axis
         CALL SUSLIN (TYL,NL,TKL,ZW,ZL,ZLT,WCL)
         DO 20 I = 1, NL
            WCLEQI(I) = WCL(I)
            JJTOT(I)=0
20       CONTINUE
         JTOT=0
         CALL SUCONV (2,TKL,ZL,ZLT,RAIN,ZW,ZE,WL0,WL0MX,RUNOF,TRWL,
     $                EVSW,EVSC,UPRISE,DRAIQT,FLXQT,NL)
      ELSE
*-----   susawa called from dynamic section of main

*-----   pressure head at lower profile boundary; later inactivated
*        for free drainage option (SUUNST)
         HPBP  = ZLT - ZW

*-----   check whether input transpiration is within bounds
         DO 25 I=1,NL
            AVAI=MAX(0.,(WCL(I)-WCAD(I))*TKL(I))
            IF(-TRWL(I).LT.-TINY.OR.-TRWL(I).GT.AVAI) THEN
               CALL SUERR(5,TRWL(I),-AVAI,0.)
            ENDIF
25       CONTINUE

*-----   resetting of state variables by activating impulse functions
*        of water balance equation
         DO 30 I= 1, NL
            WCL(I) = WCL(I) + DELT * TRWL(I) / TKL(I)
 30      CONTINUE
         WL0   = WL0 + RAIN * DELT

*-----   extraction of water evaporated from soil and/or surface water
*-----   warning: evsw and evsc are negative since they are loss terms;
         IF (-WL0.LE.EVSC*DELT) THEN
*-----      more water on surface than will be lost by evaporation
            EVSW  = EVSC
            EVSWX = 0.
            WL0   = WL0 - (-EVSC * DELT)
         ELSEIF (WL0.GT.0.) THEN
*-----      some water on surface
            EVSW1  = WL0 / DELT
            EVSW1B = CSC2/(ZE+TINY1)
            EVSW2  = AMIN1 (-EVSC-EVSW1,EVSW1B,
     $                      (WCL(1)-WCAD(1))*TKL(1)/DELT)
            EVSW   = -(EVSW1 + EVSW2)
            EVSWX  = EVSW2
            WCL(1) = WCL(1) - EVSW2 * DELT / TKL(1)
            WL0    = 0.
         ELSE
*-----      no water on surface
            EVSW1  = -EVSC
            EVSW2  = (WCL(1)-WCAD(1))*TKL(1)/DELT
            EVSW3  = CSC2 / (ZE+TINY1)
            EVSW   = -AMIN1 (EVSW1,EVSW2,EVSW3)
            EVSWX  = -EVSW
            WCL(1) = WCL(1) + EVSW * DELT / TKL(1)
         ENDIF

*----    time loop for dummy integration
         TIMTOT = 0.
         DTMX2 = 1.
         DTAV=0.
         ICOUNT=0
************************** time loop *********************************
40       IF (DTMX2.GT.TINY2) THEN
*-----      loop while local time less than DELT
            ICOUNT=ICOUNT+1

*-----      identification of saturated compartments
            JTOT = 0
            DO 50 J=1,NL
               JJTOT(J) = 0
 50         CONTINUE
            CALL SUSTCH (WCL,NL,INXSAT,JTOT,JJTOT)

*-----      calculation of prospective saturated fluxes FLXSTT
            IF (JTOT.NE.0) THEN
*-----         one or more compartments are saturated;
               CALL SUSTHH (INXSAT,JTOT,JJTOT,NL,HGT,HGB,WL0,WL0MX,HPBP,
     $               DHH)
               CALL SUSTFL (NL,INXSAT,JTOT,JJTOT,TKL,DHH,FLXSTT,FLXSQ1,
     $               FLXSQ2,DELH)
            ENDIF

*-----      calculation of tentative unsaturated fluxes FLXUNT at all
```

```
*          interfaces bordering unsaturated soil
           IF (JTOT.EQ.1.AND.JJTOT(1).EQ.NL) THEN
*-----        full profile saturated; only surface value needs to be
*             explicitly set to zero.
              FLXUNT(1) = 0.
           ELSE
              CALL SUUNST (WCL,TKL,NL,WL0,WL0MX,HPBP,
     $             MS,FLXUNT)
           ENDIF
*-----     selection of fluxes at sat-unsat interfaces
           CALL SUSEFL (INXSAT,NL,JTOT,JJTOT,FLXSTT,FLXUNT,HPBP,
     $          WL0,WL0MX,FLXSQ1,FLXSQ2,MS,TKL,FLX)

*-----     selection of max abs flux value
           FLXMX=0.
           DO 60 I=1,NL+1
             IF(ABS(FLX(I)).GT.FLXMX) THEN
               FLXMX=ABS(FLX(I))
             ENDIF
60         CONTINUE

           IF(FLXMX.LT.TINYJP) THEN
*----         freeze movement; jump to end of time loop
              DTMX2=0.
           ELSE
*-----        timestep DT choice and dummy integration
              CALL SUINTG (HPBP,FLX,TKL,WCL,MS,DTMIN,DTMX1,DTMX2,
     $             DTFX,INXSAT,DHH,JTOT,JJTOT,FLXSQ1,FLXSQ2,WL0,NL,
     $             IDRAIN,WL0MX,ZW,FLUXDT,DRAIDT,DELT,DT)
              TIMTOT = TIMTOT + DT
              DTAV   = TIMTOT/ICOUNT
              DTMX2  = DELT - TIMTOT
*-----        dtmx2 is remaining time; if large enough, loop again
           ENDIF
```

```
*                      top compt of sat set
                       PRHEAD(I)=0.+(DELH(I)+TKL(I))/2.
                     ELSE
*                      compt within a sat set
                       PRHEAD(I)=PRHEAD(I-1)+(DELH(I-1)+
     $                           DELH(I)+TKL(I-1)+TKL(I))/2.
                     ENDIF
                  ENDIF
               ENDIF
72          CONTINUE
74       CONTINUE
      ENDIF
70    CONTINUE
*-----end of switl=2 section

*-----conversion of units to main program standard
      CALL SUCONV (2,TKL,ZL,ZLT,RAIN,ZW,ZE,WL0,WL0MX,RUNOF,TRWL,
     $     EVSW,EVSC,UPRISE,DRAIQT,FLXQT,NL)

*-----rates of change of main state variables, averaged over DELT
      ZECH   = (ZE - ZEQT) / DELT
      WL0CH  = (WL0 - WL0QT) / DELT
      WCUMCH = 0.
      DO 90 I=1,NL
         WCLCH(I) = (WCL(I) - WCLQT(I)) / DELT
         WCUMCH   = WCUMCH + WCLCH(I) * TKL(I) * CONV
90    CONTINUE

      ENDIF
      RETURN
      END
```

```
*********************** end time loop *******************************
*-----  see whether timestep DELT not yet full
        GO TO 40
        END IF

        type *, ' dtav = ', dtav

*-----  runoff calculation
        IF (WL0.GT.WL0MX) THEN
           RUNOF =-(WL0-WL0MX)/DELT
           WL0   =WL0MX
        ENDIF

*-----  fluxes averaged over DELT; positive downward
        DO 80 I= 1, NL+1
           FLXQT(I) = FLUXDT(I) / DELT
80      CONTINUE
        UPRISE = -FLXQT(NL+1)
        DRAIQT = DRAIDT / DELT

*-----  change in depth of evaporation front
        CALL SUZECA (WCLQT(1),EVSC,RAIN,-EVSWX,FLXQT(2),
     $       WL0,DELT,ZE,CSA,CSB,CSC2)


*-----  pressure heads only for output to MAIN; is terminal after DELT
*       completion
        JTOT = 0
        DO 73 J=1,NL
           JJTOT(J) = 0
73      CONTINUE
        CALL SUSTCH (WCL,NL,INXSAT,JTOT,JJTOT)
        IF (JTOT.NE.0) THEN
           CALL SUSTHH (INXSAT,JTOT,JJTOT,NL,HGT,HGB,WL0,WL0MX,HPBP,
     $                  DHH)
           CALL SUSTFL (NL,INXSAT,JTOT,JJTOT,TKL,DHH,FLXSTT,FLXSQ1,
     $                  FLXSQ2,DELH)
        ENDIF
        DO 70 I=1,NL
           CALL SUWCMS(I,1,WCL(I),MS(I))
           PRHEAD(I)=-MS(I)
           IF(JTOT.GT.0) THEN
*             saturated compartments
              DO 74 J=1,JTOT
                 DO 72 JJ=1,JJTOT(J)
                    IF(I.EQ.INXSAT(J,JJ)) THEN
                       IF(I.EQ.1) THEN
*                         surface compartment
                          PRHEAD(1)=WL0+(DELH(1)+TKL(1))/2.
                       ELSE
                          IF(JJ.EQ.1) THEN
```

```
*----------------------------------------------------------------------*
* SUBROUTINE SUSEFL                                                    *
*                                                                      *
* Authors: Hein ten Berge and Don Jansen                               *
* Date   : 1-MRT-1992                                                  *
* Version: 2.0                                                         *
*                                                                      *
* Purpose: SUSEFL assigns values to fluxes at saturated-               *
*          unsaturated interfaces. A choice is made between tentative  *
*          values for saturated and unsaturated fluxes supplied by     *
*          SUSTFL and SUUNST, respectively. Downward fluxes are        *
*          positive. The full procedure is outlined in Appendix 8 of   *
*          the manual.                                                 *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name   type meaning                                    units  class *
* ----   ---- -------                                    -----  ----- *
* INXSAT R4   index of saturated compartment               -      I   *
* NL     I4   number of soil compartments                  -      I   *
* JTOT   R4   index of saturated set (top=1)               -      I   *
* JJTOT  R4   number of compartments in saturated set      -      I   *
* FLXSTT R4   tentative soil water flux through saturated cm/d    I   *
*             set                                                      *
* FLXUNT R4   unsaturated soil water flux at compartment  cm/d    I   *
*             interface                                                *
* HPBP   R4   pressure head at bottom of profile           cm     I   *
* WL0    R4   depth of water layer on surface              cm     I   *
* WL0MX  R4   maximum value of WL0                         cm     I   *
* FLXSQ1 R4   prospective water flux: resulting from                   *
*             excluding top compartment from saturated set cm/d   I   *
* FLXSQ2 R4   prospective water flux: resulting from                   *
*             including extra compartment at top of                    *
*             saturated set                               cm/d    I   *
* MS     R4   matric suction                               cm     I   *
* TKL    R4   compartment thickness                        cm     I   *
* FLX    R4   soil water flux over compartment interface  cm/d    O   *
*                                                                      *
* SUBROUTINES called : none                                            *
*                                                                      *
* FUNCTIONS called : none                                              *
*                                                                      *
* FILE usage : none                                                    *
*----------------------------------------------------------------------*

      SUBROUTINE SUSEFL (INXSAT,NL,JTOT,JJTOT,FLXSTT,FLXUNT,
     $           HPBP,WL0,WL0MX,FLXSQ1,FLXSQ2,MS,TKL,FLX)

      IMPLICIT REAL (A-Z)
      INTEGER  INXSAT, JJTOT, JTOT, J, JJ, I, IIN, IOUT, NL

      DIMENSION INXSAT(10,10), JJTOT(10), FLXSTT(11), FLXUNT(11)
      DIMENSION FLX(11), MS(10), TKL(10)
```

```
*-----variables retain their values between subsequent calls
*     of this subroutine
      SAVE

      DATA TINY /0.001/

      HPTP = AMIN1(WL0,WL0MX)
*-----first all fluxes are set equal to the unsaturated fluxes
      DO 10 I=1,NL+1
         FLX(I) = FLXUNT(I)
 10   CONTINUE
      IF(JTOT.EQ.0) GOTO 100

*-----limiting fluxes determine actual fluxes across interfaces
*     do for all saturated sets
      DO 70 J=1,JTOT
         IF (FLXSTT(J).GT.TINY) THEN
*-----      situation class 1; downward flow in saturated set
            IOUT = INXSAT(J,JJTOT(J))+1
            IIN  = INXSAT(J,1)

*-----      situation class 1.1
*           FLX(IOUT) = AMIN1(FLXSTT(J),AMAX1(0.,FLXUNT(IOUT)))
            FLX(IOUT) = FLXSTT(J)
*-----      situation class 1.2
            IF (IOUT.EQ.NL+1) FLX(IOUT) = FLXSTT(J)
*-----      situation classes 1.1 and 1.2
            IF (JJTOT(J).GT.1) THEN
               DO 20 JJ=2,JJTOT(J)
                  FLX(INXSAT(J,JJ)) = FLX(IOUT)
 20            CONTINUE
            ENDIF
            FLX(IIN) = AMIN1(FLX(IOUT),FLXUNT(IIN))
            IF(IIN.EQ.1.AND.HPTP.GT.TINY) FLX(IIN) = FLX(IOUT)

         ELSEIF (FLXSTT(J).LT.-TINY) THEN
*-----      situation class 2; upward flow in saturated set
            IOUT = INXSAT(J,1)
            IIN  = INXSAT(J,JJTOT(J))+1

*-----      first for interface IOUT
            IF(IOUT.EQ.1) THEN
*-----         class 2.2.4
               FLX(IOUT)=FLXSTT(J)
            ELSEIF (FLXSQ2.LE.TINY) THEN
*-----         class 2.1: upward flux would be sustained if next upper
*              compartment were included in sat set
               FLX(IOUT) = FLXSTT(J)
            ELSE
*-----         class 2.2: not enough pressure to sustain upward flux
*              after including next upper compartment
               MSAL=TKL(IOUT-1)/2.
               CALL SUWCMS(IOUT-1,2,WIX2,MSAL)
               MSACT=MS(IOUT-1)
               CALL SUWCMS(IOUT-1,2,WIX1,MSACT)
               IF(ABS(WIX2-WIX1).LT.0.01) THEN
*------           compartment above iout is in hydrostat equil;
*                 maintain this situation
                  FLX(IOUT)=FLX(IOUT-1)
               ELSEIF(FLXUNT(IOUT).LT.-TINY) THEN
*-----            class 2.2.1; compt above iout is dryer than hydr.equil
                  FLX(IOUT) = AMAX1(FLXUNT(IOUT),FLXSTT(J))
               ELSEIF (FLXUNT(IOUT).GT.TINY) THEN
*-----            class 2.2.2; compt above iout is wetter than hyd.equil
                  FLX(IOUT) = AMAX1(0.,AMIN1(FLXSQ2,FLX(IOUT-1)))
               ELSE
*-----            class 2.2.3
                  FLX(IOUT) = 0.
               ENDIF
            ENDIF

*-----      then for other interfaces in this sat set, if present
            IF (JJTOT(J).GT.1) THEN
               DO 30 JJ=2,JJTOT(J)
                  FLX(INXSAT(J,JJ)) = FLX(IOUT)
 30            CONTINUE
            ENDIF
*-----      and for bottom end
            FLX(IIN) = FLX(IOUT)

         ELSE
*-----      flxstt between -tiny and + tiny; implies either hydrostatic
*           equilibrium because groundwater in profile, or perched water
*           on extremely impervious layer. Top boundary of satset desig-
*           nated as iin, bottom as iout
            IIN  = INXSAT(J,1)
            IOUT = INXSAT(J,JJTOT(J))+1

            IF(IOUT.NE.NL+1) THEN
*------         perched water on extremely impervious layer
               IF (FLXUNT(IIN).LE.-TINY) THEN
                  FLX(IIN)=FLXUNT(IIN)
*------            top compt of sat set will become unsat
               ELSE
                  FLX(IIN)=0.
               ENDIF
               IF (JJTOT(J).GT.1) THEN
                  DO 35 JJ=2,JJTOT(J)
                     FLX(INXSAT(J,JJ)) = 0.
 35               CONTINUE
               ENDIF
               FLX(IOUT) = 0.

            ELSE
*-----       situation class 3; hydrostatic equilibrium in sat set;
*            groundwater in profile; flow only due to supply or removal
*            at top of sat set


               IF (FLXUNT(IIN).LE.-TINY) THEN
*-----            class 3.1; upward removal at top sat set due to suction
*                 of overlaying unsaturated soil
                  FLX(IIN) = FLXUNT(IIN)
                  IF (JJTOT(J).GT.1) THEN
                     DO 40 JJ=2,JJTOT(J)
                        FLX(INXSAT(J,JJ)) = AMAX1(FLXSQ1,FLX(IIN))
 40                  CONTINUE
                  ENDIF
                  FLX(IOUT) = AMAX1(FLXSQ1,FLX(IIN))

               ELSEIF (FLXUNT(IIN).GE.TINY) THEN
*-----            class 3.2; supply at top of sat set from overlying almos
*                 saturated soil
                  FLX(IIN) = AMIN1(FLXUNT(IIN),FLXSQ2)
                  IF (JJTOT(J).GT.1) THEN
                     DO 50 JJ=2,JJTOT(J)
                        FLX(INXSAT(J,JJ)) = FLX(IIN)
 50                  CONTINUE
                  ENDIF
                  FLX(IOUT) = FLX(IIN)

               ELSE
*-----            class 3.3; no supply or removal at top boundary;
*                 zero fluxes
                  FLX(IIN) = 0.
                  IF (JJTOT(J).GT.1) THEN
                     DO 60 JJ=2,JJTOT(J)
                        FLX(INXSAT(J,JJ)) = 0.
 60                  CONTINUE
                  ENDIF
                  FLX(IOUT) = 0.
               ENDIF
            ENDIF
         ENDIF
 70   CONTINUE

      IF(INXSAT(JTOT,JJTOT(JTOT)).EQ.NL) GOTO 500
*     last compartment saturated; necessary arrangements made above

 100   IF(HPBP.GT.0.) THEN
       IF(HPBP.LT.TKL(NL)) THEN
            MSAL=TKL(NL)/2.
            CALL SUWCMS(NL,2,WIX2,MSAL)
            MSACT=MS(NL)
            CALL SUWCMS(NL,2,WIX1,MSACT)
            IF(ABS(WIX2-WIX1).LT.0.01) THEN
               FLX(NL+1)=FLX(NL)
            ELSEIF(WIX2.LT.WIX1) THEN
*              too wet, equil level in lower compnt;
*              reach equil soon as possible
               IF(FLX(NL).LT.0.)  FLX(NL+1)=0.
               IF(FLX(NL).GT.0.)  FLX(NL+1)=10.*FLX(NL)
            ELSE
               IF(FLX(NL).GT.0.)  FLX(NL+1)=0.
               IF(FLX(NL).LT.0.)  FLX(NL+1)=10.*FLX(NL)
            ENDIF
         ENDIF
*      ENDIF
      ENDIF

 500  DO 80 I=1,NL+1
```

```
      FLXUNT(I) = 0.
      FLXSTT(I) = 0.
80  CONTINUE

    RETURN
    END

*------------------------------------------------------------------------*
* SUBROUTINE SUSLIN                                                      *
*                                                                        *
* Authors: Hein ten Berge and Don Jansen                                 *
*                                                                        *
* Date    : 1-MRT-1992                                                   *
* Version: 2.0                                                           *
*                                                                        *
* Purpose: SUSLIN calculates the depth of the top of each soil           *
*          compartment; and water contents in equilibrium with           *
*          groundwater. It also writes soil water contents at key        *
*          suction values to an output file (only under CSMP).           *
*                                                                        *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)        *
* name   type meaning                                      units  class  *
* ----   ---- -------                                      -----  -----  *
* TYL     I4  type of soil in corresponding compartment      -      I    *
* NL      I4  number of soil compartments                    -      I    *
* TKL     R4  thickness of soil compartment                  cm     I    *
* ZW      R4  depth of  ground water level below             cm     I    *
*             soil surface                                               *
* ZL()    R4  depth of top of soil compartments              cm     O    *
* ZLT     R4  total depth of profile                         cm     O    *
* WCL     R4  volumetric soil water content                  -      O    *
*                                                                        *
* SUBROUTINES called :                                                   *
*                                                                        *
*     - SUERR                                                            *
*     - SUWCMS                                                           *
*                                                                        *
* FUNCTIONS called :                                                     *
*                                                                        *
* FILE usage :                                                           *
*                                                                        *
*     - * (screen)                                                       *
*------------------------------------------------------------------------*

      SUBROUTINE SUSLIN (TYL,NL,TKL,ZW,ZL,ZLT,WCL)

      IMPLICIT REAL (A-Z)
      INTEGER  I, NL, IWRIT

      DIMENSION TYL(10),TKL(10),WCL(10),ZL(10)

*-----common blocks
      COMMON /VOLWAT/ WCAD(10),  WCFC(10),  WCST(10),  WCWP(10)
      COMMON /WRITS/ IWRIT

*-----variables retain their values between subsequent calls
*     of this subroutine
      SAVE

      IF(IWRIT.EQ.0) WRITE (6,20)
      DO 10 I=1,NL
*        write hydraulic soil properties derived from basic data
*        only for SAWAH under CSMP main; under FORTRAN, these
*        properties are written by DRSAWA
         IF(IWRIT.EQ.0) THEN
*------     sawah was called without driver
            WRITE (6,30) I,TYL(I),TKL(I)*.01,WCAD(I),
     $      WCWP(I),WCFC(I),WCST(I)
         ENDIF

*------  depths of compartment tops
         IF (I.EQ.1) THEN
            ZL(I) = 0.
         ELSE
            ZL(I) = ZL(I-1) + TKL(I-1)
         ENDIF
         MS = AMAX1 (0.,ZW-ZL(I)-0.5*TKL(I))
         CALL SUWCMS (I,2,WCL(I),MS)
         IF(WCL(I).LT.WCAD(I).OR.WCL(I).GT.WCST(I)) THEN
            CALL SUERR (2,WCL(I),WCAD(I),WCST(I))
         ENDIF
10    CONTINUE
      ZLT = ZL(NL) + TKL(NL)

20    FORMAT (/,' SOIL CHARACTERISTICS PER COMPARTMENT: ',/,
     $          ' COMPARTMENT TYPE NR     TKL     WCAD     WCWP',
     $          '   WCFC     WCST')
30    FORMAT (3X,I4,8X,F5.1,3X,F5.3,3X,4(F5.4,3X))
      RETURN
      END

*------------------------------------------------------------------------*
* SUBROUTINE SUSTCH                                                      *
*                                                                        *
* Authors: Hein ten Berge and Don Jansen                                 *
* Date    : 1-MRT-1992                                                   *
* Version: 2.0                                                           *
*                                                                        *
* Purpose: SUSTCH checks for presence of saturated compartments.         *
*          These compartments are marked and organized into              *
*          JTOT saturated sets, each with JJTOT(J) compartments.         *
*          The indexes of these compartments are stored in INXSAT.       *
*                                                                        *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)        *
* name    type meaning                                     units  class  *
* ----    ---- -------                                     -----  -----  *
* WCL      R4  volumetric soil water content                 -      I    *
* NL       I4  number of soil compartments                   -      I    *
* INXSAT   I4  index of saturated compartment                -      O    *
* JTOT     I4  index of saturated set (top=1)                -      O    *
* JJTOT    I4  number of compartments in one saturated set   -      O    *
*                                                                        *
* SUBROUTINES called : none                                              *
*                                                                        *
* FUNCTIONS called : none                                                *
*                                                                        *
* FILE usage : none                                                      *
*------------------------------------------------------------------------*

      SUBROUTINE SUSTCH (WCL,NL,INXSAT,JTOT,JJTOT)

      IMPLICIT REAL (A-Z)
      INTEGER  NL, INXSAT, I, J, JTOT, JJTOT, JJ, II, III

      DIMENSION WCL(10), INXSAT(10,10), JJTOT(10)

*-----common block
      COMMON /VOLWAT/ WCAD(10),  WCFC(10),  WCST(10),  WCWP(10)

*-----variables retain their values between subsequent calls
*     of this subroutine
      SAVE

      DATA TINY /0.01/

      DO 8 II=1,10
         DO 5 III=1,10
            INXSAT(II,III)=0
5        CONTINUE
8     CONTINUE

      J = 0
      JJ = 0
*-----do for all compartments
      DO 10 I=1,NL
         IF (ABS(WCL(I)-WCST(I)).LT.TINY) THEN
*-----      compartment is saturated; JJ counts compartments per set
            JJ = JJ + 1
            IF (JJ.NE.1) THEN
*-----         J counts sets
               J = J + 0
            ELSE
               J = J + 1
               JTOT = J
            ENDIF
            INXSAT(J,JJ) = I
            JJTOT(J) = JJ
         ELSE
            JJ = 0
         ENDIF
10    CONTINUE
      RETURN
      END

*------------------------------------------------------------------------*
* SUBROUTINE SUSTFL                                                      *
*                                                                        *
* Authors: Hein ten Berge and Don Jansen                                 *
* Date    : 1-MRT-1992                                                   *
* Version: 2.0                                                           *
*                                                                        *
* Purpose: SUSTFL calculates the tentative flux through a saturated      *
```

```
*          section. The flux is obtained by solving a set of linear     *
*          equations. This tentative flux is later compared with        *
*          possible fluxes in the unsaturated soil at the edges of      *
*          the saturated section, and a final value is determined.      *
*          Within a saturated section, all fluxes are equal.            *
*                                                                       *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)       *
* name   type meaning                                      units  class *
* ----   ---- -------                                      -----  ----- *
* NL     I    number of compartments                         -      I   *
* INXSAT I4   index of saturated compartment                 -      I   *
* JTOT   I4   index of saturated set (top-1)                 -      I   *
* JJTOT  I4   number of compartments in one saturated set    -      I   *
* TKL    R4   thickness of soil compartment                  cm     I   *
* DHH    R4   hydraulic head difference over saturated                  *
*             section                                        cm     I   *
* FLXSTT R4   tentative soil water flux through saturated   cm/d    O   *
*             section                                                   *
* FLXSQ1 R4   value of FLXSTT that would result were one    cm/d    O   *
*             compartment lost at the top of the saturated              *
*             set.                                                      *
* FLXSQ2 R4   value of FLXSTT that would result were one            O   *
*             extra compartment included at the top of the cm/d         *
*             saturated set                                             *
* DELH() R4   difference in hydraulic head over saturated               *
*             compartment                                    hPa    O   *
*                                                                       *
* SUBROUTINES called :                                                  *
*                                                                       *
*     - SUSTMD                                                          *
*     - SUSTMS                                                          *
*                                                                       *
* FUNCTIONS called : none                                               *
*                                                                       *
* FILE usage : none                                                     *
*-----------------------------------------------------------------------*

      SUBROUTINE SUSTFL (NL,INXSAT,JTOT,JJTOT,TKL,DHH,
     $                   FLXSTT,FLXSQ1,FLXSQ2,DELH)

      IMPLICIT REAL (A-Z)
      INTEGER  I, J, N, NL, JTOT, JJTOT, INXSAT, INDX, IA, JA
      INTEGER  JIK,JIKK

      DIMENSION A(11,11), TKL(10), INXSAT(10,10), JJTOT(10), B(11)
      DIMENSION DHH(10),  FLXSTT(11), DIS(10), INDX(11), KS(10)
      DIMENSION DELH(10)

*-----common block
*     KMSMX dummy in SUSTFL
      COMMON /HYDCON/ KMSMX(10), KMSA1(10), KMSA2(10), KST(10)

*-----variables retain their values between subsequent calls
*     of this subroutine
      SAVE

      DATA TINY   /1.0E-10/
      DATA LARGE  /-100./
      DATA DUMMY2 /0./

      DO 5 I=1,10
         DELH(I)=0.
5     CONTINUE
*-----do for every saturated set
      DO 190 J=1,JTOT

*-----   clear the matrix A and vector B
         DO 20 IA=1,NL+1
            DO 10 JA=1,NL+1
               A(IA,JA) = 0.
               B(JA) = 0.
 10         CONTINUE
 20      CONTINUE

*-----   calculation of tentative flux FLXSTT
         IF (ABS(DHH(J)).LT.TINY) THEN
*-----      no hydraulic head gradient over sat set
            FLXSTT(J) = 0.
            DO 25 JIK=1,JJTOT(J)
               JIKK=INXSAT(J,1)+JIK-1
               DELH(JIKK)=0.
25          CONTINUE
         ELSE
*-----      calculate flux through sat set and hydraulic heads at
*           interfaces. do for all compartments in sat set
            DO 30 IA=1,JJTOT(J)
               I       = INXSAT(J,1) + IA - 1
               KS(IA)  = KST(I)
               DIS(IA) = TKL(I)
 30         CONTINUE
*-----      assign rightmost column and diagonal elements of A
            DO 40 IA=1,JJTOT(J)
               A(IA,JJTOT(J)+1) = -1.
               JA = IA
               A(IA,JA) = -KS(IA) / DIS(IA)
 40         CONTINUE
*-----      assign lower row of A
            DO 50 JA=1,JJTOT(J)
               A(JJTOT(J)+1,JA) = +1.
 50         CONTINUE
            A(JJTOT(J)+1,JJTOT(J)+1) = 0.
*-----      define vector B
            DO 60 JA=1,JJTOT(J)
               B(JA) = 0.
 60         CONTINUE
*-----      lower element of B is total hydraulic head
*           difference over sat set
            B(JJTOT(J)+1) = DHH(J)
            N = JJTOT(J)+1
*-----      solve set of linear eqs Ax=B
*           decompose matrix
            CALL SUSTMD (A,N,INDX,D)
*-----      solve matrix equation
            CALL SUSTMS (A,N,INDX,B)
            FLXSTT(J) = B(N)
*-----      pressure head differences over sat sets for output
*           to MAIN, not for flow calc
            DO 65 JIK=1,JJTOT(J)
               JIKK=INXSAT(J,1)+JIK-1
               DELH(JIKK)=B(JIK)
65          CONTINUE
         ENDIF
*-----   calculation of tentative flux through set J completed

*-----   prospective fluxes to avoid oscillations when groundwater
*        is in profile
         IF (DHH(J).GT.-TINY) THEN
*-----      groundwater in profile; calculations for
*           sat set JTOT; clear matrix A and vector B
            DO 80 IA=1,NL+1
               DO 70 JA=1,NL+1
                  A(IA,JA) = 0.
                  B(JA) = 0.
 70            CONTINUE
 80         CONTINUE
*-----      calculation of FLXSQ1: the resulting sat flux if top
*           compartment of sat set would become unsaturated
            IF (JJTOT(J).EQ.1) THEN
*-----      only one compartment in sat set
               FLXSQ1 = LARGE
            ELSE
*-----      more compartments in sat set
               DO 90 IA=1,JJTOT(J)-1
                  I       = INXSAT(J,2) + IA - 1
                  KS(IA)  = KST(I)
                  DIS(IA) = TKL(I)
 90            CONTINUE
               DO 100 IA=1,JJTOT(J)-1
                  A(IA,JJTOT(J)) = -1.
                  JA = IA
                  A(IA,JA) = -KS(IA) / DIS(IA)
 100           CONTINUE
               DO 110 JA=1,JJTOT(J)-1
                  A(JJTOT(J),JA) = +1.
 110           CONTINUE
               A(JJTOT(J),JJTOT(J)) = 0.
               DO 120 JA=1,JJTOT(J)-1
                  B(JA) = 0.
 120           CONTINUE
               B(JJTOT(J)) = DHH(J)+TKL(INXSAT(J,1))
               N = JJTOT(J)
               CALL SUSTMD (A,N,INDX,D)
               CALL SUSTMS (A,N,INDX,B)
               FLXSQ1 = B(N)
            ENDIF

*-----      groundwater in profile; calculations for sat set JTOT.
*           clear matrix A and vector B
            DO 140 IA=1,NL+1
               DO 130 JA=1,NL+1
                  A(IA,JA) = 0.
                  B(JA) = 0.
```

```
 130          CONTINUE
 140       CONTINUE

*-----     calculation of FLXSQ2: resulting flux if next upper
*          compartment would be incorporated into sat set
           IF (INXSAT(J,1).EQ.1) THEN
*-----        groundwater reaches to surface
              FLXSQ2 = DUMMY2
           ELSE
*-----        top compartment unsaturated
              DO 150 IA=1,JJTOT(J)+1
                 I = (INXSAT(J,1)-1) + IA - 1
                 KS(IA)  = KST(I)
                 DIS(IA) = TKL(I)
 150          CONTINUE
              DO 160 IA=1,JJTOT(J)+1
                 A(IA,JJTOT(J)+2) = -1.
                 JA = IA
                 A(IA,JA) = -KS(IA) / DIS(IA)
 160          CONTINUE
              DO 170 JA=1,JJTOT(J)+1
                 A(JJTOT(J)+2,JA) = +1.
 170          CONTINUE
              A(JJTOT(J)+2,JJTOT(J)+2) = 0.
              DO 180 JA=1,JJTOT(J)+1
                 B(JA) = 0.
 180          CONTINUE
*-----        overpressure if overlying compartment would
*             be incorporated into sat set
              B(JJTOT(J)+2) = DHH(J) - TKL(INXSAT(J,1)-1)
              N = JJTOT(J) + 2
              CALL SUSTMD (A,N,INDX,D)
              CALL SUSTMS (A,N,INDX,B)
              FLXSQ2 = B(N)
           ENDIF
        ENDIF
 190  CONTINUE
      RETURN
      END

*----------------------------------------------------------------------*
* SUBROUTINE SUSTHH                                                    *
*                                                                      *
* Authors: Hein ten Berge and Don Jansen                               *
* Date   : 1-MRT-1992                                                  *
* Version: 2.0                                                         *
*                                                                      *
* Purpose: SUSTHH determines the hydraulic head difference across      *
*          each saturated set.                                         *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name   type meaning                                    units  class *
* ----   ---- -------                                    -----  ----- *
* INXSAT I4   index of saturated compartment               -      I   *
* JTOT   I4   index of saturated set (top=1)               -      I   *
* JJTOT  I4   number of compartments in one saturated set  -      I   *
* NL     I4   number of soil compartments                  -      I   *
* HGT    R4   gravitational head at top of compartment     cm     I   *
* HGB    R4   gravitational head at bottom of compartment  cm     I   *
* WL0    R4   depth of water layer on soil surface         cm     I   *
* WL0MX  R4   maximum depth of surface water layer         cm     I   *
* HPBP   R4   pressure  head at bottom of profile          cm     I   *
* DHH    R4   hydraulic head difference over saturated                *
*             section                                      cm     O   *
*                                                                      *
* SUBROUTINES called : none                                            *
*                                                                      *
* FUNCTIONS called : none                                              *
*                                                                      *
* FILE usage : none                                                    *
*----------------------------------------------------------------------*

      SUBROUTINE SUSTHH(INXSAT,JTOT,JJTOT,NL,HGT,HGB,
     $                  WL0,WL0MX,HPBP,DHH)

      IMPLICIT REAL (A-Z)
      INTEGER NL, INXSAT, J, JTOT, JJTOT

      DIMENSION INXSAT(10,10), JJTOT(10), HHT(10), HHB(10), HGT(10)
      DIMENSION HGB(10), DHH(10)

*-----variables retain their values between subsequent calls
*     of this subroutine
      SAVE

      DATA TINY /1.E-5/

*---- pressure head on the soil surface; although WL0 may exceed
*     WL0MX for rain impulse formulation (manual), it may not exceed
*     WL0MX for pressure head calculation
      HPTP  = AMIN1 (WL0,WL0MX)

      DO 10 J=1,10
         DHH(J)=0.
10    CONTINUE

*-----do for all saturated sets
      DO 20 J=1,JTOT
*-----   pressure head due to ponded surface water
         EXTRAT = 0.
         IF (INXSAT(J,1).EQ.1) EXTRAT = HPTP
*-----   pressure head at lower end of profile
         EXTRAB = 0.
         IF (INXSAT(J,JJTOT(J)).EQ.NL) EXTRAB = HPBP
*-----   hydraulic head difference over saturated set J
         HHT(J) = EXTRAT+HGT(INXSAT(J,1))
         HHB(J) = EXTRAB+HGB(INXSAT(J,JJTOT(J)))
         DHH(J) = HHB(J)-HHT(J)
         IF (ABS(DHH(J)).LT.TINY) DHH(J) = 0.
 20   CONTINUE
      RETURN
      END

*----------------------------------------------------------------------*
* SUBROUTINE SUSTMD                                                    *
*                                                                      *
* Authors: W.H. Press, B.P. Flannery, S.A. Teukolsky, and              *
*          W.T. Vetterling. In: Numerical Recipes, pp 35-36, by the    *
*          same authors. Cambridge University Press, 1986.             *
*                                                                      *
* Purpose: SUSTMD decomposes the matrix A into lower and               *
*          upper triangle. Solution is subsequently by SUSTMS.         *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name   type meaning                                    units  class *
* ----   ---- -------                                    -----  ----- *
* A      R4   See Press et al.                             -     I,O  *
* N      I4   See Press et al.                             -     I,O  *
* INDX   I4   See Press et al.                             -     I,O  *
* D      R4   See Press et al.                             -     I,O  *
*                                                                      *
* SUBROUTINES called : none                                            *
*                                                                      *
* FUNCTIONS called : none                                              *
*                                                                      *
* FILE usage : none                                                    *
*----------------------------------------------------------------------*

      SUBROUTINE SUSTMD (A,N,INDX,D)

      IMPLICIT  REAL (A-Z)
      INTEGER   I, J, K, N, IMAX, INDX

      DIMENSION A(11,11), INDX(11), VV(11)

*-----variables retain their values between subsequent calls
*     of this subroutine
      SAVE

      DATA TINY /1.0E-10/

      D = 1.
      DO 20 I=1,N
         AAMAX = 0.
         DO 10 J=1,N
            IF (ABS(A(I,J)).GT.AAMAX) AAMAX = ABS(A(I,J))
10       CONTINUE
         VV(I) = 1. / AAMAX
20    CONTINUE

      DO 90 J=1,N
         IF (J.GT.1) THEN
            DO 40 I=1,J-1
               SUM = A(I,J)
               IF (I.GT.1) THEN
                  DO 30 K=1,I-1
                     SUM = SUM - A(I,K) * A(K,J)
30                CONTINUE
                  A(I,J)=SUM
               ENDIF
40          CONTINUE
         ENDIF
```

```
         AAMAX = 0.
         DO 60 I=J,N
            SUM = A(I,J)
            IF (J.GT.1.) THEN
               DO 50 K=1,J-1
                  SUM = SUM - A(I,K) * A(K,J)
   50          CONTINUE
               A(I,J) = SUM
            ENDIF
            DUM = VV(I) * ABS(SUM)
            IF (DUM.GE.AAMAX) THEN
               IMAX  = I
               AAMAX = DUM
            ENDIF
   60    CONTINUE
         IF (J.NE.IMAX) THEN
            DO 70 K=1,N
               DUM = A(IMAX,K)
               A(IMAX,K) = A(J,K)
               A(J,K) = DUM
   70       CONTINUE
            D = - D
            VV(IMAX) = VV(J)
         ENDIF
         INDX(J) = IMAX
         IF (J.NE.N) THEN
            IF (ABS(A(J,J)).LT.TINY) A(J,J) = TINY
            DUM = 1. / A(J,J)
            DO 80 I=J+1,N
               A(I,J) = A(I,J) * DUM
   80       CONTINUE
         ENDIF
   90 CONTINUE
      IF (ABS(A(N,N)).LT.TINY) A(N,N) = TINY
      RETURN
      END

*-----------------------------------------------------------------------*
*  SUBROUTINE SUSTMS                                                    *
*                                                                       *
*  Authors: W.H. Press, B.P. Flannery, S.A. Teukolsky, and              *
*           W.T. Vetterling. In: Numerical Recipes, pp 35-36, by the    *
*           same authors. Cambridge University Press, 1986.             *
*                                                                       *
*  Purpose: SUSTMS solves a set of linear equations if                  *
*           combined with SUSTMD.                                       *
*                                                                       *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
*  name   type meaning                                     units  class *
*  ----   ---- -------                                     -----  ----- *
*  A       R4   See Press et al., 1986                       -      I   *
*  N       I4   See Press et al., 1986                       -      I   *
*  INDX    I4   See Press et al., 1986                       -      I   *
*  B       R4   See Press et al., 1986                       -      O   *
*                                                                       *
*  SUBROUTINES called : none                                            *
*                                                                       *
*  FUNCTIONS called : none                                              *
*                                                                       *
*  FILE usage : none                                                    *
*-----------------------------------------------------------------------*

      SUBROUTINE SUSTMS (A,N,INDX,B)

      IMPLICIT  REAL (A-Z)
      INTEGER   I, J, II, LL, N, INDX
      DIMENSION A(11,11), INDX(11), B(11)

*-----variables retain their values between subsequent calls
*     of this subroutine
      SAVE

      II = 0
      DO 20 I=1,N
         LL = INDX(I)
         SUM = B(LL)
         B(LL) = B(I)
         IF (II.NE.0) THEN
            DO 10 J=II,I-1
               SUM = SUM - A(I,J) * B(J)
   10       CONTINUE
         ELSE IF (SUM.NE.0.) THEN
            II = I
         ENDIF
         B(I) = SUM
   20 CONTINUE
      DO 40 I=N,1,-1
         SUM = B(I)
         IF (I.LT.N) THEN
            DO 30 J=I+1,N
               SUM = SUM - A(I,J) * B(J)
   30       CONTINUE
         ENDIF
         B(I) = SUM / A(I,I)
   40 CONTINUE
      RETURN
      END

*-----------------------------------------------------------------------*
*  SUBROUTINE SUUNST                                                    *
*                                                                       *
*  Authors: Don Jansen and Hein ten Berge                               *
*  Date   : 1-MRT-1992                                                  *
*  Version: 2.0                                                         *
*                                                                       *
*  Purpose: SUUNST calculates fluxes across the interface between       *
*           unsaturated compartments, or, if the next compartment is    *
*           saturated, between the centre of the unsaturated compartment *
*           and the common interface. At the sat-unsat boundary,        *
*           pressure head is assumed to be zero.                        *
*                                                                       *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
*  name   type meaning                                     units  class *
*  ----   ---- -------                                     -----  ----- *
*  WCL     R4   volumetric soil water content                -      I   *
*  TKL     R4   thickness of soil compartments              cm      I   *
*  NL      R4   number of compartments                       -      I   *
*  WL0     R4   depth of water layer on surface             cm      I   *
*  WL0MX   R4   maximum value of WL0                        cm      I   *
*  HPBP    R4   pressure  head at bottom of profile         cm      I   *
*  MS      R4   matric suction                              cm      O   *
*  FLXUNT  R4   unsaturated soil water flux at compartment  cm/d    O   *
*               interface                                               *
*                                                                       *
*  OPTIONS: SWIT7=1  pressure head lower boundary condition             *
*           SWIT7=2  free drainage lower boundary condition             *
*                    (if groundwater enters profile, program switches   *
*                    to pressure head condition)                        *
*                                                                       *
*  SUBROUTINES called : none                                            *
*                                                                       *
*  FUNCTIONS called : none                                              *
*                                                                       *
*  FILE usage : none                                                    *
*-----------------------------------------------------------------------*

      SUBROUTINE SUUNST (WCL,TKL,NL,WL0,WL0MX,HPBP,MS,FLXUNT)

      IMPLICIT REAL (A-Z)
      INTEGER  NL, I

      DIMENSION WCL(10), TKL(10), DIS(11), FLXUNT(11), MS(10)
      DIMENSION KMS(10), MFLP(10), MFLPQT(10)

*-----common blocks
      INTEGER SWIT3, SWIT5, SWIT7, SWIT8, SWIT9, SWIT10
      COMMON /SWITCH/ SWIT3,SWIT5,SWIT7,SWIT8,SWIT9,SWIT10
      COMMON /HYDCON/ KMSMX(10), KMSA1(10), KMSA2(10), KST(10)
      COMMON /VOLWAT/ WCAD(10),  WCFC(10),  WCST(10),  WCWP(10)

*-----variables retain their values between subsequent calls
*     of this subroutine
      SAVE

      DATA TINY1 /0.001/
      DATA TINY2 /0.01/

*---- pressure head at soil surface if ponded
      HPTP = AMIN1 (WL0,WL0MX)

*---- set matric suction to zero for all compartments
      DO 5 I=1,NL
         MS(I)=0.
    5 CONTINUE

*-----calculate fluxes FLXUNT over interfaces between unsaturated
*     compartments; and towards/away from interfaces between a saturated
*     and an unsaturated compartment, on the unsaturated side of the
*     interface, if present.
      IF (WCST(1)-WCL(1).GT.TINY2) THEN
*-----   top compartment unsaturated
         CALL SUWCMS (1,1,WCL(1),MS(1))
```

```
      CALL SUMFLP (1,MS(1),MFLP(1))
      DIS(1) = 0.5 * TKL(1)
      IF (HPTP.GT.TINY1) THEN
*-----   water ponded at surface
         DZDH = DIS(1) / (-MS(1)-HPTP)
         DMFLP = MFLP(1)-0.-KST(1) * HPTP
         FLXUNT(1) = (DZDH-1.) * DMFLP / DIS(1)
      ELSE
*-----   no water on surface; evapotranspiration
*        was already subtracted
         FLXUNT(1) = 0.
      ENDIF
   ELSE
*----- top compartment saturated; evapotranspiration
*      already subtracted.
      FLXUNT(1) = 0.
   ENDIF

*-----flux between centres of unsat compartments, or between centre of
*     unsaturated and interface with adjacent saturated compartment
*     do for all interfaces
   DO 10 I=2,NL
      IF (ABS(WCL(I)-WCST(I)).LT.TINY2) THEN
*-----   lower compartment saturated
         IF (ABS(WCL(I-1)-WCST(I-1)).GT.TINY2) THEN
*-----      upper compartment unsaturated
            DIS(I) = 0.5 * TKL(I-1)
            DZDH   = DIS(I) / (0.+MS(I-1))
            DMFLP  = 0. - MFLP(I-1)
            FLXUNT(I) = (DZDH-1.) * DMFLP / DIS(I)
         ELSE
*-----      upper compartment also saturated
            CONTINUE
         ENDIF
      ELSE
*-----lower compartment unsaturated
         CALL SUWCMS (I,1,WCL(I),MS(I))
         CALL SUMFLP (I,MS(I),MFLP(I))

         IF (ABS(WCL(I-1)-WCST(I-1)).LT.TINY2) THEN
*-----      upper compartment saturated
            DIS(I) = 0.5 * TKL(I)
            DZDH   = DIS(I) / (-MS(I)-0.)
            DMFLP  = MFLP(I)-0.
            FLXUNT(I) = (DZDH-1.) * DMFLP / DIS(I)
         ELSE
*-----      upper compartment also unsaturated
            CALL SUMFLP (I-1,MS(I),MFLPQT(I))
            CALL SUMFLP (I,MS(I-1),MFLPQT(I-1))
            DIS(I) = 0.5 * (TKL(I)+TKL(I-1))
            DMFLP1 = MFLP(I)-MFLPQT(I-1)
            IF((MS(I)-MS(I-1))*(MFLP(I)-MFLPQT(I-1)).GT.-TINY1) THEN
*-----         prevents flow errors resulting from rounding errors
*              in Gauss intgrl
               DMFLP1 = 0.
            ENDIF
            DMFLP2 = MFLPQT(I)-MFLP(I-1)
            IF((MS(I)-MS(I-1))*(MFLPQT(I)-MFLP(I-1)).GT.-TINY1) THEN
*-----         prevents flow errors resulting from rounding errors
*              in Gauss intgrl
               DMFLP2 = 0.
            ENDIF
            IF (ABS(DMFLP1).LT.TINY2.OR.ABS(DMFLP2).LT.TINY2) THEN
               DMFLP = 0.
            ELSE
               SIGN  = DMFLP1 / ABS(DMFLP1)
               DMFLP = SIGN * SQRT(DMFLP1*DMFLP2)
            ENDIF

            IF (MS(I).NE.MS(I-1)) THEN
*-----         suction gradient
               DZDH  = DIS(I) / (-MS(I)+MS(I-1))
               FLXUNT(I) = (DZDH-1.) * DMFLP / DIS(I)
            ELSE
*-----         no suction gradient; only gravity flow
               CALL SUMSKM (I,MS(I),KMS(I))
               CALL SUMSKM (I-1,MS(I-1),KMS(I-1))
               KAV = SQRT(KMS(I) * KMS(I-1))
               FLXUNT(I) = + KAV
            ENDIF
         ENDIF
      ENDIF
10 CONTINUE

*-----flux over bottom of profile at zlb
   IF (ABS(WCL(NL)-WCST(NL)).LT.TINY2) THEN
*-----   bottom compartment saturated; boundary flux was
*        already calculated by SUSTFL
         CONTINUE
      ELSE
*-----   bottom compartment unsaturated
         DIS(NL+1) = 0.5 * TKL(NL)
         CALL SUMFLP (NL,MS(NL),MFLP(NL))
         IF (HPBP.GT.TINY1) THEN
*-----      groundwater in profile
            DZDH  = DIS(NL+1) / (HPBP+MS(NL))
            DMFLP = 0. - MFLP(NL) + KST(NL) * (HPBP-0.)
            FLXUNT(NL+1) = (DZDH - 1.) * DMFLP / DIS(NL+1)
         ELSE
*-----      no groundwater in profile
            MSB = -HPBP
            CALL SUMFLP (NL,MSB,MFLPB)
            IF (HPBP.NE.-MS(NL).AND.SWIT7.EQ.1) THEN
*-----         suction gradient in last compartment; no free drainage
               DZDH  = DIS(NL+1) / (HPBP+MS(NL))
               DMFLP = MFLPB - MFLP(NL)
               IF ((MSB-MS(NL)) * (MFLPB-MFLP(NL)).GT.-TINY1) THEN
*-----            prevents flow errors resulting from rounding
*                 errors in Gauss intgrl
                  DMFLP =0.
               ENDIF
               FLXUNT(NL+1) = (DZDH-1.) * DMFLP / DIS(NL+1)
            ELSE
*-----         no suction gradient in last compartment; unit gradient
*              flow
               CALL SUMSKM (NL,MS(NL),KMS(NL))
               FLXUNT(NL+1) = +KMS(NL)
            ENDIF
         ENDIF
      ENDIF
      RETURN
      END

*----------------------------------------------------------------------*
*  SUBROUTINE SUWCHK                                                   *
*                                                                      *
*  Authors: Don Jansen and Hein ten Berge                              *
*  Date   : 1-MRT-1992                                                 *
*  Version: 2.0                                                        *
*                                                                      *
*  Purpose: SUWCHK checks the soil water balance by comparing          *
*           time-integrated boundary fluxes versus change in           *
*           total amount of water contained in the system.             *
*                                                                      *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)     *
*  name   type meaning                                  units  class *
*  ----   ---- -------                                  -----  ----- *
*  CKWFL   -   sum of time-integrated boundary fluxes     mm      I   *
*  CKWIN   -   change in water storage since start        mm      I   *
*  TIME    -   time                                        d      I   *
*                                                                      *
*  SUBROUTINES called : none                                           *
*                                                                      *
*  FUNCTIONS called : none                                             *
*                                                                      *
*  FILE usage :                                                        *
*                                                                      *
*     - * (screen), unit IUNLOG                                        *
*----------------------------------------------------------------------*

      SUBROUTINE SUWCHK (CKWFL,CKWIN,TIME)

      IMPLICIT REAL (A-Z)
      INTEGER IUNLOG

*-----common
      COMMON /UNITNR/ IUNLOG

*-----variables retain their values between subsequent calls
*     of this subroutine
      SAVE

      FUWCHK = 2.0 * (CKWIN-CKWFL) / (CKWIN+CKWFL+1.E-10)
      XDIF = ABS(CKWIN-CKWFL)
      IF (ABS(FUWCHK).GT.0.01.AND.XDIF.GT.1.0) THEN
*-----   absolute error in water balance exceeds 1 mm
*        and relative error exceeds 1%.
         WRITE (*,10) FUWCHK,CKWIN,CKWFL,TIME
         IF (IUNLOG.GT.0) WRITE (IUNLOG,10) FUWCHK,CKWIN,CKWFL,TIME
      END IF
10    FORMAT (/'* * * error in water balance, please check * * *',/,
     $         ' CKWRD  =',F6.3,' CKWIN=',F8.2,' CKWFL=',F8.2,
```

```
      $            ' AT TIME = ',F6.1)
       RETURN
       END

*----------------------------------------------------------------------*
*  SUBROUTINE SUWCMS                                                   *
*                                                                      *
*  Authors: Hein ten Berge and Don Jansen                              *
*  Date   : 1-MRT-1992                                                 *
*  Version: 2.0                                                        *
*                                                                      *
*  Purpose: SUWCMS calculates volumetric soil water content from       *
*           soil water suction, and vice versa. Various options are    *
*           offered. See SWIT8 in input file or SAWAH manual.          *
*                                                                      *
*  FORMAL PARAMETERS: (I=input, O=output, C=control, IN=init, T=time)  *
*  name   type meaning                               units   class     *
*  ----   ---- -------                               -----   -----     *
*  I      I4   compartment index                      -        I       *
*  SWIT4  I4   switch to set request MS(WCL) or WCL(MS) -      I       *
*  WCL    R4   volumetric soil water content          -       I/O      *
*  MS     R4   soil water suction                     cm      I/O      *
*                                                                      *
*  OPTIONS: SWIT8=1  Driessen moisture characteristic                  *
*           SWIT8=2  Van Genuchten moisture characteristic             *
*           SWIT8=3  linear interpolation on log-scale                 *
*           SWIT8=4  user-defined function                             *
*                                                                      *
*  SUBROUTINES called:                                                 *
*                                                                      *
*   -   SUERR                                                          *
*                                                                      *
*  FUNCTION called:                                                    *
*                                                                      *
*   -   none                                                           *
*                                                                      *
*  FILE usage:                                                         *
*                                                                      *
*   -   none                                                           *
*                                                                      *
************************************************************************

      SUBROUTINE SUWCMS (I,SWIT4,WCL,MS)

      IMPLICIT REAL (A-Z)
      INTEGER I, SWIT4, I2, J
      DIMENSION PF(10,11)

*-----common blocks
*     only SWIT8 is used in SUWCMS
      INTEGER SWIT3, SWIT5, SWIT7, SWIT8, SWIT9, SWIT10
      COMMON /SWITCH/  SWIT3, SWIT5, SWIT7, SWIT8, SWIT9, SWIT10
*     WCFC and WCWP dummy in SUWCMS
      COMMON /VOLWAT/ WCAD(10),  WCFC(10),  WCST(10),  WCWP(10)
      COMMON /PFCURV/ MSWCA(10) , PFWC00(10), PFWC01(10), PFWC02(10),
     $                PFWC03(10), PFWC04(10), PFWC05(10), PFWC06(10),
     $                PFWC07(10), PFWC08(10), PFWC09(10), PFWC10(10)
      COMMON /NUCHT/  VGN(10),VGA(10),VGR(10),VGL(10)

*-----variables retain their values between subsequent calls
*     of this subroutine
      SAVE

      DATA TINY /0.001/

*     data points for interpolation option
      PF(I,1) = PFWC00(I)
      PF(I,2) = PFWC01(I)
      PF(I,3) = PFWC02(I)
      PF(I,4) = PFWC03(I)
      PF(I,5) = PFWC04(I)
      PF(I,6) = PFWC05(I)
      PF(I,7) = PFWC06(I)
      PF(I,8) = PFWC07(I)
      PF(I,9) = PFWC08(I)
      PF(I,10)= PFWC09(I)
      PF(I,11)= PFWC10(I)

      IF (SWIT4.EQ.1) THEN
*        suction calculated from water content
         IF(WCL.LT.WCAD(I).OR.WCL.GT.WCST(I)) THEN
            CALL SUERR (3, WCL, WCAD(I), WCST(I))
         ENDIF
         IF(SWIT8.EQ.1) THEN
*           Driessen option
            HLP1 =  AMAX1 (WCAD(I), WCL)
            HLP2 = -ALOG (HLP1 / WCST(I))
            HLP3 =  SQRT (HLP2 / MSWCA(I))
            MS   =  EXP (HLP3) - 1.
         ELSEIF(SWIT8.EQ.2) THEN
*           Van Genuchten option
            HLP1=AMAX1(WCAD(I),WCL)
            WREL=(WCL-VGR(I))/(WCST(I)-VGR(I))
            VGM=1.-1./VGN(I)
            HLP2=1./VGA(I)
            HLP3=-1./VGM
            HLP4=1./VGN(I)
            MS=HLP2*(WREL**HLP3-1.)**HLP4
         ELSEIF(SWIT8.EQ.3) THEN
*           option: interpolation on pF curve
            WCLR=AMAX1(0.,AMIN1(WCL/WCST(I),1.0))
            HI2=WCLR/0.1
            I2=NINT(HI2-0.5) +1
            IF(I2.EQ.11) THEN
               MS=0.
            ELSE
               WCLRD=WCLR-0.1*(I2-1)
               IF(WCLRD.LE.TINY) WCLRD=0.
               PFF=PF(I,I2)-(WCLRD/0.1)*(PF(I,I2)-PF(I,I2+1))
               MS=10.**PFF
               MS=AMIN1(MS,1.E7)
               MS=AMAX1(MS,0.)
               GOTO 30
            ENDIF
         ELSEIF(SWIT8.EQ.4) THEN
*           user can specify own function here
*           remove the following 2 lines:
            WRITE(*,20)
            STOP
         ENDIF

      ELSEIF(SWIT4.EQ.2) THEN

*        water content calculated from suction
         IF(MS.LT.-TINY.OR.MS.GT.1.E8) THEN
            CALL SUERR (4, MS, 0., 1.E8)
         ENDIF
         IF(SWIT8.EQ.1) THEN
*           Driessen option
            HLP1 =  ALOG(MS+1.)
            HLP2 = -MSWCA(I) * (HLP1**2)
            HLP3 =  WCST(I) * EXP(HLP2)
            WCL  =  AMAX1(TINY,HLP3)
         ELSEIF(SWIT8.EQ.2) THEN
*           van Genuchten option
            VGM=1.-1./VGN(I)
            HLP1=(MS*VGA(I))**VGN(I)
            WREL=(1.+HLP1)**(-VGM)
            WCL=WREL*(WCST(I)-VGR(I))+VGR(I)
         ELSEIF(SWIT8.EQ.3) THEN
*           option: interpolation on pF curve
            IF(MS.LE.1.) THEN
               WCL=WCST(I)
            ELSEIF(MS.GE.1.E7) THEN
               WCL=0.01
            ELSE
               PFF=ALOG10(MS)
               DO 10 J=2,11
                IF(PFF.GE.PF(I,J)) THEN
                  I2=J-1
                  WCLR=0.1*((I2-1)+(PF(I,I2)-PFF)/(PF(I,I2)-PF(I,I2+1)))
                  IF(WCLR.LE.TINY) WCLR=0.
                  WCL=WCLR*WCST(I)
                  IF(WCL.LT.0.01) WCL=0.01
                  GOTO 30
                ENDIF
10             CONTINUE
            ENDIF
         ELSEIF(SWIT8.EQ.4) THEN
*           user can specify own function here
*           remove the following 2 lines:
            WRITE(*,20)
            STOP
         ENDIF
      ENDIF

20    FORMAT(//,'*** fatal error. At option SWIT8=4, the user',/,
     $          'must write his/her own moisture characteristic',/,
     $          'function in routine SUWCMS ***' )

30    RETURN
      END
```

```
*------------------------------------------------------------------------*
*  SUBROUTINE SUZECA                                                     *
*                                                                        *
*  Authors: Hein ten Berge and Don Jansen                                *
*  Date   : 1-MRT-1992                                                   *
*  Version: 2.0                                                          *
*                                                                        *
*  Purpose: SUZECA calculates the depth of the evaporation front;        *
*           below the front only liquid transport, above the             *
*           front only vapor transport occurs.                           *
*                                                                        *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)       *
*  name   type meaning                                   units  class *
*  ----   ---- -------                                   -----  ----- *
*  WCLQT1 R4   volumetric soil water content               -      I   *
*  EVSC   R4   potential soil evaporation rate            cm/d    I   *
*  RAIN   R4   rainfall/irrigation rate                   cm/d    I   *
*  FLX1   R4   DELT-averaged surface flux                 cm/d    I   *
*  FLX2   R4   DELT-averaged flux at bottom of top        cm/d    I   *
*              compartment                                            *
*  WL0    R4   depth of water layer on surface             cm     I   *
*  DELT   R4   integration interval                        d      I   *
*  ZE     R4   depth of evaporation front below soil surface  cm  I,O  *
*  AEXP   R4   soil evaporation parameter                 cm2/d   I   *
*  BEXP   R4   soil evaporation parameter                   -     I   *
*  C2     R4   effective soil vapour diffusivity          cm2/d   I   *
*                                                                        *
*  SUBROUTINES called : none                                             *
*                                                                        *
*  FUNCTIONS called : none                                               *
*                                                                        *
*  FILE usage : none                                                     *
*------------------------------------------------------------------------*

      SUBROUTINE SUZECA (WCLQT1,EVSC,RAIN,FLX1,FLX2,WL0,
     $                   DELT,ZE,AEXP,BEXP,C2)

      IMPLICIT REAL (A-Z)

*-----common block
*     WCFC and WCWP dummy in SUZECA
      COMMON /VOLWAT/ WCAD(10),  WCFC(10),  WCST(10),  WCWP(10)

*-----variables retain their values between subsequent calls
*     of this subroutine
      SAVE

      DATA TINY1 /1.E-3/
      DATA LARGE /10./

      IF (RAIN.GT.-EVSC) THEN
*-----   more rain than potential evaporation; reset ZE
         ZE = 0.
      ELSEIF (FLX2.LT.-TINY1.AND.FLX2.LT.FLX1) THEN
*-----   more supply from below than removed at surface
         ZE = ZE - (FLX1-FLX2) / (0.5*WCST(1))
         IF (ZE.LE.TINY1) ZE = 0.
      ELSEIF (RAIN.GT.0.05) THEN
*-----   all rain evaporated but no change in ZE
         CONTINUE
      ELSE
*-----   standard drying sequence
         WI = WCLQT1 / WCST(1)
         WTH = WI - WCAD(1) / WCST(1)
         IF (WTH.LE.TINY1) THEN
            ZE = LARGE
         ELSEIF (WL0.GT.TINY1.OR.ABS(WCLQT1-WCST(1)).LT.TINY1) THEN
            ZE = 0.
         ELSE
            HLP1 = (WI-0.5) * BEXP
            HLP2 = AMAX1 (0.,HLP1)
            HLP3 = EXP(HLP2)
            C1   = AEXP * (HLP3-1.)
            C3   = C2 / (WTH+C1/C2)
            ZE   = SQRT (ZE * ZE + 2. * C3 * DELT)
         ENDIF
      ENDIF
      RETURN
      END
```

List of references

Angus, J.F., and H.G. Zandstra, 1980. Climatic factors and the modelling of rice growth and yield, pp 189-190 in Proceedings of the WMO-IRRI symposium on Agrometeorology of the rice crop, Los Banos, Philippines.

Belmans, C., J.G. Wesseling, and R.A. Feddes, 1983. Simulation model of the water balance of a cropped soil: SWATRE. Journal of Hydrology 63: 271-286.

Berge, H.F.M. ten, K. Metselaar and L. Stroosnijder, 1987. Measurement of matric flux potential: a simple procedure for the hydraulic characterization of soils. Netherlands Journal of Agricultural Science 35, 371-384.

Berge, H.F.M. ten, 1990. Heat and water transfer in bare topsoil and the lower atmosphere. Simulation Monographs 33, Pudoc, Wageningen.

Boesten, J.J.T.I., and L. Stroosnijder, 1986. Simple model for daily evaporation from fallow tilled soil under spring conditions in a temperate climate. Netherlands Journal of Agricultural Science 34: 75-90.

Bolton, F.R. and H.G. Zandstra, 1981. A soil moisture based yield model of wetland rainfed rice. IRRI Research Paper 62. IRRI, Los Banos, Philippines.

Diepen, C.A. van, C.Rappoldt, J.Wolf and H. van Keulen, 1988. CWFS crop growth simulation model WOFOST, Documentation version 4.1, Center for World Food Studies, P.O.Box 14, 6700 AA Wageningen, The Netherlands

Driessen, P.M., 1986. The water balance of the soil. In: H. van Keulen and J. Wolf Eds., Modelling of Agricultural Production: weather, soils and crops. Simulation Monographs, Pudoc, Wageningen. pp. 76-116.

Forrester, J.W., 1961. Industrial dynamics. Massachusets Institute of Technology Press, Cambridge, Mass., USA.

Gardner, W.R., 1958. Some steady state solutions of the unsaturated moisture flow equation with application to evaporation from a water table. Soil Science 85: 228-232.

Genuchten, M.Th. Van, 1980. A closed form equation for predicting the hydraulic conductivity of unsaturated Soils. Soil Science Society of America Journal 44: 892-898.

Giraldez, J.V., and G. Sposito, 1985. Infiltration in swelling Soils. Water Resources Research 21: 33-44.

IBM, 1975. Continuous System Modelling Program III. General system information manual (GH19-7000) and users manual (SH19-7001-2). IBM Data Processing Div., White Plains, N.Y.

Iwata, S. and T. Tabuchi with B.P. Warkentin, 1988. Soil-Water Interactions, mechanisms and applications. Marcel Dekker, N.Y. 380 pp.

Jansen, D.M., R.T. Dierkx, H.H. van Laar and M.J. Alagos, 1988. PCSMP on IBM PC-AT's or PC-XT's and compatibles. Simulation Reports CABO-TT nr. 15. CABO, P.O. Box 14, 6700 AA Wageningen.

Klute, A., 1952. A numerical method for solving the flow equation for water in unsaturated materials. Soil Science 73: 105-117.

Kraalingen, D.W.G. van, and C. Rappoldt, 1989. Subprograms in simulation models. Simulation Reports CABO-TT nr. 18, CABO, P.O. Box 14, 6700 AA Wageningen.

Landeros Sanchez, 1987. First stage evaporation from bare soils: numerical treatment of available soil physical data. MSc thesis, Wageningen Agricultural University.

Lascano, R.J., and C.H.M. van Bavel, 1986. Simulation and measurement of evaporation from a bare soil. Soil Science Society of America Journal 50: 1127-1133

Mc Mennamy, J.A., and J.C. O'Toole, 1983. RICEMOD: a physiologically based rice growth and yield model. IRRI Research Paper 87. IRRI, Los Banos, Philippines.

Parlange, J.-Y., 1971. Theory of water movement in soils: 2. One-dimensional infiltration. Soil Science 111: 170-174.

Penning de Vries, F.W.T., D.M. Jansen, H.F.M. ten Berge, and A. Bakema. Simulation of ecophysiological processes of growth in several annual crops. Simulation Monographs 29, Pudoc, Wageningen. 271 pp.

Philip, J.R., and D.A. De Vries, 1957. Moisture movement in porous materials under temperature gradients. Transactions of the American Geophysical Union 38: 222-231.

Press, W.H., B.P.Flannery, S.A. Teukolsky, and W.T. Vetterling, 1986. Numerical Recipes, The art of scientific computing. Cambridge University Press N.Y. ISBN 0 521 30811 9. 818 pp.

Raats, P.A.C., 1970. Steady infiltration from line sources and furrows. Soil Science Society of America Proceedings 34: 709-714.

Rappoldt, C., and D.W.G. Van Kraalingen. Reference manual of the FORTRAN utility library TTUTIL with applications. Simulation Reports CABO-TT nr. 20. CABO, P.O. Box 14, 6700 AA Wageningen.

Rijtema, 1969. Soil moisture forecasting. Bulletin 513, Institute of Land and Water Management Research (ICW), Wageningen, 17 pp.

Shaykewich, C.F. and L. Stroosnijder, 1977. The concept of matric flux potential applied to simulation of evaporation from soil. Netherlands Journal of Agricultural Science 25: 63-82.

Smith, R.E., and J.-Y. Parlange, 1978. A parameter efficient hydrologic infiltration model. Water Resources Research 14: 533-538.

Stroosnijder, L., 1982. Simulation of the soil water balance. In: Simulation of plant growth and crop production, F.W.T. Penning de Vries and H.H. van Laar Eds. Simulation Monographs, Pudoc, Wageningen. pp. 175-209.

Willingen, P. de, and M. van Noordwijk, 1991. Modelling nutrient uptake: from single roots to complete root Systems. In: F.W.T. Penning de Vries, H.H. van Laar and M.J. Kropff Eds. Simulation and systems analysis for rice production. Pudoc, Wageningen. pp. 277-295.

Appendix 1. Listing FORTRAN MAIN

```
*      PROGRAM MAIN (FORTRAN EXAMPLE)

*-----declarations for use of DRSAWA
*     control variables
      INTEGER ITASK,IUNIT,IOUT,IUNLOG
*     timing
      REAL TIME,DAY,DELT,DTAV
*     number of soil layers
      INTEGER INLAY
*     layer thicknesses
      REAL TKL(10)
*     volumetric water contents
      REAL WCAD(10),WCWP(10),WCFC(10),WCST(10)
*     rates of change (per day)
      REAL EVSC,TRWL(10),EVSW,FLXQT(11)
*     status variable
      REAL WCLQT(10),PRHEAD(10)
*     cumulative amounts
      REAL FLXCU(11)
      REAL DRAICU, DRSLCU, EVSWCU, RAINCU, RNOFCU
      REAL TRWCU, WCUMCO, WL0CO

*     weather related variables
      INTEGER ISTAT, IDAY, IYEAR
      REAL LONG, LAT, ALT, A,B
      REAL RAD,TMIN,TMAX,VAPOUR,WIND,RAIN

*-----declaration other help variables
      INTEGER I
      REAL FINTIM

*-----commons
      COMMON/UNITNR/IUNLOG

      SAVE

*     INITIAL SECTION
*     ===============

*     log files
      IUNLOG=80
      CALL FOPEN(IUNLOG,'MODEL.LOG','NEW','DEL')

*     timer
      DELT=1.
      FINTIM=240.
      TIME=225.
      DAY  = 1.0 + MOD (TIME-1.0,365.0)
      IDAY = NINT(DAY)
      IYEAR= 1986

*     weather initialisation
      CALL STINFO(0000,'WEATHER_DATA:',' ','PHIL',
     $            2,IYEAR,ISTAT,LONG,LAT,ALT,A,B)

*     initialization of soil water system
      ITASK = 1
      IUNIT = 70
      IOUT  = 11112
      CALL DRSAWA (ITASK,IUNIT,IUNLOG,IOUT,'VBINPUT.DAT',
     $             TIME,DAY,DELT,EVSC,RAIN,TRC,TRWL,
     $             INLAY,TKL,WCAD,WCWP,WCFC,WCST,
     $             EVSW,FLXQT,WCLQT,WL0QT,ZEQT,
     $             DRAICU,DRSLCU,EVSWCU,RAINCU,RNOFCU,TRWCU,
     $             FLXCU,WCUMCO,WL0CO,PRHEAD,DTAV)


*     DYNAMIC SECTION
*     ===============

*     time loop
10    IF (TIME.LE.FINTIM) THEN

*--------integration (first call after initialization is dummy)
         ITASK = 3
         CALL DRSAWA (ITASK,IUNIT,IUNLOG,IOUT,'VBINPUT.DAT',
     $                TIME,DAY,DELT,EVSC,RAIN,TRC,TRWL,
     $                INLAY,TKL,WCAD,WCWP,WCFC,WCST,
     $                EVSW,FLXQT,WCLQT,WL0QT,ZEQT,
     $                DRAICU,DRSLCU,EVSWCU,RAINCU,RNOFCU,TRWCU,
     $                FLXCU,WCUMCO,WL0CO,PRHEAD,DTAV)

*--------rate calculation
*        Reading of dynamic boundary conditions, and rate
*        calculations by other submodels are to be inserted here.
*        For the present example only weather data are entered,
*        making use of the CABO-TPE weather routines and library.
*        (CABO/TPE internal communication: User manual of CABO/TPE
*        weather system. June 1991.) Example:
*        CALL WEATHR(IDAY,ISTAT,RAD,TMIN,TMAX,VAPOUR,WIND,RAIN)
*        IF(ISTAT.NE.0) THEN
*           WRITE(*,'(A,I4)') ' ISTAT <> ZERO, ISTAT=',ISTAT
*        ENDIF
*        current test purpose: rain zero, start with water layer on field
         RAIN=0.
*        Potential transpiration in this example is fixed at 8.0 mm/d.
*        Usually it will be determined from weather and crop variables.
*        NOTE: TRC is not used by the submodel SAWAH or its driver DRSAWA
*        except for printing in output file; user has to define TRWL
*        externally on the basis of soil water status and crop variables.
         TRC = -8.
         DO 15 I=1,4
            TRWL(I)=AMAX1(-1.8,
     $                 -1000.*TKL(I)*AMAX1(0.,(WCLQT(I)-WCWP(I))))
15       CONTINUE
         DO 20 I=5,10
            TRWL(I)=0.
20       CONTINUE

*        Potential soil evaporation
         EVSC=-4.0

         ITASK = 2
         CALL DRSAWA (ITASK,IUNIT,IUNLOG,IOUT,'VBINPUT.DAT',
     $                TIME,DAY,DELT,EVSC,RAIN,TRC,TRWL,
     $                INLAY,TKL,WCAD,WCWP,WCFC,WCST,
     $                EVSW,FLXQT,WCLQT,WL0QT,ZEQT,
     $                DRAICU,DRSLCU,EVSWCU,RAINCU,RNOFCU,TRWCU,
     $                FLXCU,WCUMCO,WL0CO,PRHEAD,DTAV)

*--------timer
         TIME = TIME + DELT
         DAY  = 1.0 + MOD (TIME-1.0,365.0)
         IDAY = NINT(DAY)

      GOTO 10
      END IF

*     TERMINAL
*     ========
      ITASK = 4
      CALL DRSAWA (ITASK,IUNIT,IUNLOG,IOUT,'VBINPUT.DAT',
     $             TIME,DAY,DELT,EVSC,RAIN,TRC,TRWL,
     $             INLAY,TKL,WCAD,WCWP,WCFC,WCST,
     $             EVSW,FLXQT,WCLQT,WL0QT,ZEQT,
     $             DRAICU,DRSLCU,EVSWCU,RAINCU,RNOFCU,TRWCU,
     $             FLXCU,WCUMCO,WL0CO,PRHEAD,DTAV)
      END
```

Appendix 2. Listing CSMP MAIN

```
FIXED I, ITYL, IUNLOG, NL
FIXED SWIT3, SWIT5, SWIT6, SWIT7, SWIT8, SWIT9
FIXED S3,S5,S6,S7,S8,S9
STORAGE TRWL(10),WCLEQI(10),WCLQTM(10),KMSA1T(20),KMSA2T(20)
STORAGE KMSMXT(20),KSTT(20),MSWCAT(20),WCSTT(20)
STORAGE PRHEAD(10), FLXQT(11), TYL(10), TKL(10)

/      COMMON /VOLWAT/ WCAD(10),WCFC(10),WCST(10),WCWP(10)
/      COMMON /HYDCON/ KMSMX(10),KMSA1(10),KMSA2(10),KST(10)
/      COMMON /PFCURV/ MSWCA(10)
/      COMMON /SWITCH/ S3,S5,S6,S7,S8,S9


INITIAL

* initialization of states
INCON ZEQTI =0.
INCON WLOQTI=0.
ZWI   =AFGEN(ZWTB,DATEB)
PROCEDURE WCLQTI,WCUMI=PRWCLI(WCLEQI,WCLQTM)
   IF(SWIT6.EQ.1.OR.ZWI.LT.ZLT) THEN
      DO 1 I=1,NL
        WCLQTI(I)=WCLEQI(I)
 1    CONTINUE
   ELSEIF(SWIT6.EQ.2) THEN
      DO 2 I=1,NL
        WCLQTI(I)=WCLQTM(I)
 2    CONTINUE
   ENDIF
   WCUMI  =0.
   DO 4 I=1,NL
      WCUMI    =WCUMI+WCLQTI(I)*TKL(I)*1000.
 4 CONTINUE
ENDPROCEDURE

PROCEDURE WCLEQI,TKLT=PRWCLE(NL,TKL,TYL,ZWI)
   CALL SUSWS(SWIT3,SWIT5,SWIT6,SWIT7,SWIT8,SWIT9,...
             S3,   S5,   S6,   S7,   S8,   S9)
   DO 5 I=1,NL
     ITYL    =TYL(I)
     KST(I)  =KSTT  (ITYL)
     KMSMX(I)=KMSMXT(ITYL)
     KMSA1(I)=KMSA1T(ITYL)
     KMSA2(I)=KMSA2T(ITYL)
     MSWCA(I)=MSWCAT(ITYL)
     WCST(I) =WCSTT (ITYL)
     WCFC(I) =FUWCMS(I,100.0)
     WCWP(I) =FUWCMS(I,1.6E4)
     WCAD(I) =FUWCMS(I,1.0E7)
 5 CONTINUE
   WCLCH,WLOCH,WCLEQI,EVSW,RUNOF,UPRISE,DRAIQT,WCUMCH,ZECH,...
   ZLT,FLXQT,PRHEAD,DTAV...
   =SUSAWA(1,IUNLOG,WCLQT,WLOQT,NL,TRWL,EVSC,RAIN,ZWI,TKL,...
   TYL,DELT,DTMIN,DTMX1,DTFX,IDRAIN,WLOMX,ZEQT,CSA,CSB,CSC2)
ENDPROCEDURE


DYNAMIC

*state variables
WCLQT  =INTGRL(WCLQTI,WCLCH,10)
WLOQT  =INTGRL(WLOQTI,WLOCH)
WCUM   =INTGRL(WCUMI ,WCUMCH)
ZEQT   =INTGRL(ZEQTI ,ZECH)

* cumulative water balance
DRAICU = INTGRL(0.,DRAIQT)
UPRICU = INTGRL(0.,UPRISE)
EVSWCU = INTGRL(0.,EVSW)
RAINCU = INTGRL(0.,RAIN)
RNOFCU = INTGRL(0.,RUNOF)
TRWCU  = INTGRL(0.,TRW)
WLOCO  = INTGRL(0.,SURREL)
WCUMCO = INTGRL(0.,PROREL)

* groundwater
ZW    =AFGEN(ZWTB,DATE)

* rate call to SUSAWA
WCLCH,WLOCH,WCLEQI,EVSW,RUNOF,UPRISE,WCUMCH,ZECH,ZLT,...
FLXQT,PRHEAD,DTAV...
=SUSAWA(2,IUNLOG,WCLQT,WLOQT,NL,TRWL,EVSC,RAIN,ZW,TKL,TYL,...
DELT,DTMIN,DTMX1,DTFX,IDRAIN,WLOMX,ZEQT,CSA,CSB,CSC2)

* transpiration example
PROCEDURE TRWL=PRTRAN(TRC,WCLQT,TKL)
   DO 6 I=1,4
     TRWL(I)=AMIN1(TKL(I)*1000.*(WCLQT(I)-0.10),TRC/4.)
 6 CONTINUE
   DO 7 I=5,10
     TRWL(I)=0.
 7 CONTINUE
ENDPROCEDURE


* water balance check
 PROREL = - WCUMCH
 SURREL = -1000.*WLOCH
 CKWIN  = -(WCUMCO+WLOCO)
 CKWFL  = RAINCU+RNOFCU+EVSWCU+TRWCU+UPRICU+DRAICU
 CKWRD  = SUWCHK(CKWFL,CKWIN,TIME)

* run control and output
TIMER TIME=0., FINTIM=15., PRDEL=1., DELT=1.
PARAMETER DTMIN =0.001, DTMX1 =0.1, DTFX =0.03
PRINT WCLQT(1-10), ZEQT, WLOQT, FLXQT(1-11)
METHOD RECT
DATEB=10.

* option switches
PARAM SWIT3=1, SWIT5=1, SWIT6=1
PARAM SWIT7=1, SWIT8=1, SWIT9=2


* soil characteristics
PARAMETER CSC2 =0.1, CSA =0.15, CSB =10.
PARAMETER WLOMX =0.02
TABLE TYL(1-10)=10*13.

* other parameters
PARAMETER NL =10
TABLE TKL(1-10)=10*0.10
PARAM IUNLOG=2
PARAM TRC=8.

* measured initial soil water content
TABLE WCLQTM(1-10)=10*0.36

* groundwater table
FUNCTION ZWTB  =0.,.50, 366.,1.0

* characteristics of Soil Types 1-20
TABLE KMSA1T(1-20)=...
.1960,.1385,.0821,.0500,.0269,.0562,.0378,.0395,.0750,.0490,...
.0240,.0200,.0231,.0353,.0237,.0248,.0274,.0480,.0380,.1045
TABLE KMSA2T(1-20)=...
  .08,  .63, 3.30,10.90,15.00, 5.26, 2.10,16.40,  .24,22.60,...
26.50,47.30,14.40,33.60, 3.60, 1.69, 2.77,28.20, 4.86, 6.82
TABLE KMSMXT(1-20)=...
 80.0, 90.0,125.0,175.0,165.0,100.0,135.0,200.0,150.0,130.0,...
300.0,300.0,300.0,200.0,300.0,300.0,300.0, 50.0, 80.0, 50.0
TABLE KSTT(1-20)  =...
1120.00,300.00,110.00, 50.00,  1.00,  2.30,   .36, 26.50,...
  16.50, 14.50, 12.00,  6.50,  5.00, 23.50,  1.50,   .98,...
   3.50,  1.30,   .22,  5.30
TABLE MSWCAT(1-20)=...
.0853,.0450,.0366,.0255,.0135,.0153,.0243,.0299,.0251,.0156,...
.0186,.0165,.0164,.0101,.0108,.0051,.0085,.0059,.0043,.0108
TABLE WCSTT(1-20) =...
.3950,.3650,.3500,.3640,.4700,.3940,.3010,.4390,.4650,.4550,...
.5040,.5090,.5030,.4320,.4750,.4450,.4530,.5070,.5400,.8630

END
STOP

      FUNCTION FUWCMS(I,MS)
      REAL MS
      CALL SUWCMS(I,2,WCL,MS)
      FUWCMS=WCL
      RETURN
      END

      SUBROUTINE SUSWS(SWIT3,SWIT5,SWIT6,SWIT7,SWIT8,SWIT9,
     $                 S3,    S5,    S6,    S7,    S8,    S9)

      INTEGER SWIT3,SWIT5,SWIT6,SWIT7,SWIT8,SWIT9
      INTEGER S3,    S5,    S6,    S7,    S8,    S9
      S3=SWIT3
      S5=SWIT5
      S6=SWIT6
      S7=SWIT7
      S8=SWIT8
      S9=SWIT9
      RETURN
      END

ENDJOB
```

Appendix 3. Listing DRSAWA

```
*----------------------------------------------------------------------*
*  SUBROUTINE DRSAWA                                                   *
*                                                                      *
*  Authors: Kees Rappoldt and Willem Stol                              *
*  Date   : March 1, 1992                                              *
*  Version: 3.0                                                        *
*                                                                      *
*  NOTE: This version runs only in combination with the RD* routines   *
*        from November 1990 as documented in the TTUTIL manual of      *
*        December 1990 referenced below.                               *
*        In 1992 a new version of TTUTIL will require adaptation of this *
*        driver. The CALL's to RDDATA will be changed into CALL's to   *
*        a new subroutine RDFREA.                                      *
*                                                                      *
*  Purpose: DRSAWA is the interface between the SAWAH module and the   *
*           user-defined FORTRAN MAIN. It performs the tasks of        *
*           - initialization                                           *
*           - rate calculation                                         *
*           - integration                                              *
*           - terminal treatments                                      *
*           The use of this subroutine is described in the SAWAH user's *
*           manual. Background information on the used program structure *
*           and documentation on the called utilities can be found in: *
*           D.W.G.van Kraalingen and C.Rappoldt,                       *
*             Subprograms in simulation models,                        *
*             Simulation Report CABO-TT no 18, 1989.                   *
*           C.Rappoldt and D.W.G.van Kraalingen,                       *
*             Reference manual of the FORTRAN utility library TTUTIL,  *
*             Simulation Report CABO-TT no 20, 1990.                   *
*                                                                      *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)     *
*  name   type meaning                                    units  class *
*  ----   ---- -------                                    -----  ----- *
*  control                                                             *
*  ITASK   I4  determines action of routine                -      C,I  *
*  IUNIT   I4  unit number to be used, see file usage below -   IN,C,I *
*  IUNLOG  I4  unit number in use for LOG FILE             -    IN,C,I *
*              = 0, no log file is used or assumed to exist            *
*              > 0, error messages are written to log file             *
*  IOUT    I4  output control with five digit integer number -  IN,C,I *
*              last digit refers to first file, examples               *
*              10001 --> WATER5.OUT and WATER1.OUT                     *
*                 11 --> WATER2.OUT and WATER1.OUT                     *
*                110 --> WATER3.OUT and WATER2.OUT                     *
*              11111 --> All 5 output files are produced               *
*              When the last  figure is 2 instead of 1, then           *
*              a table of soil characteristic water contents           *
*              is added to the header of WATER1.OUT. So for            *
*              full output enter IOUT=11112                            *
*  FILIN   C*  name of file with soil data                 -    IN,C,I *
*                                                                      *
*  time variables                                                      *
*  TIME    R4  simulation time                             d      T,I  *
*  DAY     R4  calendar day number (groundwater,output)    -       I   *
*  DELT    R4  time step                                   d      T,I  *
*                                                                      *
*  dynamic input                                                       *
*  EVSC    R4  potential evaporation rate                  mm/d    I   *
*  RAIN    R4  rainfall / irrigation rate                  mm/d    I   *
*  TRWL    R4  actual transpiration rate per layer         mm/d    I   *
*  TRC     R4  potential transpiration rate of crop        mm/d    I
*                                                                      *
*  soil description (available after initial call)                    *
*  INLAY   I4  number of layers specified in input file    -     IN,O  *
*  TKL     R4  thickness of soil compartments              m     IN,O  *
*  WCADX   R4  volumetric water content airdry             -     IN,O  *
*  WCWPX   R4  volumetric water content at wilting point   -     IN,O  *
*  WCFCX   R4  volumetric water content at field capacity  -     IN,O  *
*  WCSTX   R4  volumetric water content at saturation      -     IN,O  *
*                                                                      *
*  dynamic output                                                      *
*  EVSW    R4  actual (realized) evaporation rate          mm/d    O   *
*  FLXQT   R4  layer boundary fluxes (rates)               mm/d    O   *
*  WCLQT   R4  volumetric soil water content per layer     -       O   *
*  WL0QT   R4  depth of surface water layer                m       O   *
*  ZEQT    R4  depth of evaporation front                  m       O   *
*                                                                      *
*  cumulated, derived and help variables                               *
*  DRAICU  R4  cumulative drainage by drains               mm      O   *
*  UPRICU  R4  cumulative capp. rise over lower soil boundary mm   O   *
*  EVSWCU  R4  cumulative evaporation                      mm      O   *
*  RAINCU  R4  cumulative rainfall                         mm      O   *
*  RNOFCU  R4  cumulative runoff                           mm      O   *
*  TRWCU   R4  cumulative transpiration                    mm      O   *
*  FLXCU   R4  cumulative flux for each layer boundary     mm      O   *
*  WCUMCO  R4  contribution of soil storage term to                    *
*              overall water balance                       mm      O   *
*  WL0CO   R4  contribution of surface storage term to                 *
*              overall water balance                       mm      O   *
*  PRHEAD  R4  pressure head at compartment center         hPa     O   *
*  DTAV    R4  DELT-averaged DT value                      d       O   *
*                                                                      *
*  SUBPROGRAMS called :                                                *
*   - from library  SAWAH: SUCONV, SUERR , SUGRHD, SUINTG, SUMFLP      *
*                          SUMSKM, SUSAWA, SUSEFL, SUSLIN, SUSTCH      *
*                          SUSTFL, SUSTHH, SUSTMD, SUSTMS, SUUNST      *
*                          SUWCH , SUWCMS, SUZECA                      *
*   - from library TTUTIL: DECCHK, DECINT, DECREA,  ERROR, EXTENS      *
*                          FOPENG, IFINDC,  ILEN,   LINT, RDDATA       *
*                          RDAREA, RDINDX, RDINIT, RDSINT, RDSREA      *
*                          UPPERC                                      *
*                                                                      *
*  FILE usage : - Soil definition file opened and closed for ITASK=1   *
*                 unit numbers used are IUNIT and IUNIT+1              *
*               - For IOUT>0 output files are generated using the unit *
*                 numbers IUNIT+2,IUNIT+3,IUNIT+4,IUNIT+5 and IUNIT+6  *
*----------------------------------------------------------------------*

      SUBROUTINE DRSAWA (ITASK,IUNIT,IUNLOG,IOUT,FILIN,
     $                   TIME,DAY,DELT,EVSC,RAIN,TRC,TRWL,
     $                   INLAY,TKL,WCADX,WCWPX,WCFCX,WCSTX,
     $                   EVSW,FLXQT,WCLQT,WL0QT,ZEQT,
     $                   DRAICU,UPRICU,EVSWCU,RAINCU,RNOFCU,TRWCU,
     $                   FLXCU,WCUMCO,WL0CO,PRHEAD,DTAV)

*-----type declarations of all symbols are given, nevertheless
      IMPLICIT REAL (A-Z)

*-----subroutine arguments
      INTEGER ITASK,  IUNIT, IUNLOG,   IOUT,  INLAY
*     simple real variables
      REAL      TIME,    DAY,   DELT,   EVSC,   RAIN, EVSW
      REAL     WL0QT,   ZEQT, DRAICU, UPRICU, EVSWCU, RAINCU
      REAL    RNOFCU,  TRWCU, WCUMCO,  WL0CO,   DTAV, TRC
*     arrays (length declared below)
      REAL      TRWL,    TKL,  WCADX,  WCWPX,  WCFCX,  WCSTX
      REAL     FLXQT,  WCLQT,  FLXCU, PRHEAD
*     other type
      CHARACTER*(*)    FILIN

*-----local and water model common variables ; integers
      INTEGER     I, IDRAIN,    ILF,  ILOUT, ILZMAX,  ITOLD
      INTEGER  ITYL,   IRUN,  IZWTB, ILUNIT,    IUN,   IERR
      INTEGER    IS,     NL, NRDTYP, NVGTYP,  IWRIT,  SWIT3
      INTEGER SWIT5,  SWIT6,  SWIT7,  SWIT8,  SWIT9, SWIT10

*     simple real variables
      REAL     AIRDR,  CKWFL,  CKWIN,    CSA,    CSB,   CSC2
      REAL    DRAIQT, UPRISE,   DTFX,  DTMIN,  DTMX1,  FIELD
      REAL     LTIME, PROREL,  RUNOF, SURREL,   TINY,    TRW
      REAL      WCUM, WCUMCH,  WCUMI,  WILTP,  WL0CH,  WL0MX
      REAL    WL0QTI,   ZECH,  ZEQTI,    ZLT,     ZW,    ZWI
*     arrays (length declared below)
      REAL    KMSA1, KMSA1T,  KMSA2, KMSA2T,  KMSMX, KMSMXT
      REAL      KST,   KSTT,  MSWCA, MSWCAT, PFWC00, PFWC01
      REAL   PFWC02, PFWC03, PFWC04, PFWC05, PFWC06, PFWC07
      REAL   PFWC08, PFWC09, PFWC10,     PN,    TYL,    VGA
      REAL     VGAT,    VGL,   VGLT,    VGN,   VGNT,    VGR
      REAL    VGWRT,  VGWST,  VGKST,   WCAD,   WCFC,  WCLCH
      REAL   WCLEQI, WCLQTI, WCLQTM,   WCST,  WCSTT,   WCWP
      REAL     ZWTB

*     other types
      CHARACTER*6     STRWCL, STRTRW, STRFLX
      CHARACTER*10    FILNAM
      LOGICAL OKINIT, OPENED, USEFIL, EXIST

*-----functions called
      INTEGER ILEN
      REAL LINT

*-----Soil characteristics according to Rijtema/Driessen system
*     The number of soil types defined is NRDTYP
      PARAMETER (NRDTYP=20)
      DIMENSION KMSA1T(NRDTYP), KMSMXT(NRDTYP),  KSTT(NRDTYP)
      DIMENSION KMSA2T(NRDTYP), MSWCAT(NRDTYP), WCSTT(NRDTYP)

*-----Soil characteristics according to Van Genuchten system
*     The number of soil types defined is NVGTYP
      PARAMETER (NVGTYP=2)
      DIMENSION VGWRT(NVGTYP), VGWST(NVGTYP), VGKST(NVGTYP)
      DIMENSION  VGAT(NVGTYP),  VGNT(NVGTYP),  VGLT(NVGTYP)
```

```
*-----length:10 (max. nr of layers)
      DIMENSION  TRWL(10), WCLEQI(10), WCLQTM(10)
      DIMENSION WCLQT(10), WCLQTI(10),  WCLCH(10)
      DIMENSION   TKL(10),    TYL(10), PRHEAD(10)

*-----sawah common blocks (cannot contain arguments)
      COMMON /VOLWAT/  WCAD(10) ,  WCFC(10),  WCST(10),  WCWP(10)
      COMMON /HYDCON/  KMSMX(10), KMSA1(10), KMSA2(10),   KST(10)
      COMMON /POWER /  PN(10)
      COMMON /NUCHT /  VGN(10)   ,   VGA(10),   VGR(10),   VGL(10)
      COMMON /PFCURV/  MSWCA(10),PFWC00(10),PFWC01(10),PFWC02(10),
     $                PFWC03(10),PFWC04(10),PFWC05(10),PFWC06(10),
     $                PFWC07(10),PFWC08(10),PFWC09(10),PFWC10(10)
      COMMON /SWITCH/  SWIT3, SWIT5, SWIT7, SWIT8, SWIT9, SWIT10
      COMMON /WRITS /  IWRIT

*-----parameter used in error checking pF-function
      PARAMETER (TINY=0.0001)

*-----copies of volumetric water contents (subroutine arguments)
      DIMENSION WCADX(10), WCWPX(10), WCFCX(10), WCSTX(10)

*-----array with time-integrated values of FLX
*     and DELT averaged value of FLX
      DIMENSION FLXCU(11), FLXQT(11)

*-----groundwater table
      PARAMETER (ILZMAX=400)
      DIMENSION ZWTB(ILZMAX)

*-----file header strings
      DIMENSION STRWCL(10),STRTRW(10),STRFLX(11)
*     output control
      DIMENSION OPENED(5), USEFIL(5), IUN(5)

*-----variables retain their values between subsequent calls
      SAVE


*-----Soil type properties according to the Rijtema/Driessen combination.
*     Data from these tables will be used at the simultaneous occurrence
*     of the following switch values:
*           SWIT9=2  and (SWIT3=1 or 2) and (SWIT8=1)
*     The 6 soil properties are given in data tables for NRDTYP soils.
*     (for declarations, see above).  By defining the soil type TYL(I)
*     for each layer I, a consistent combination of soil properties
*     is selected. Data refer to the twenty standard soil types according
*     to Rijtema (as described by Driessen, 1986).
*
*      1. Coarse sand                11. Fine sandy loam
*      2. Medium coarse sand (mcs)   12. Silt loam
*      3. Medium fine sand           13. Loam
*      4. Fine sand                  14. Sandy clay loam
*      5. Humous loamy mcs           15. Silty clay loam
*      6. Light loamy mcs            16. Clay loam
*      7. Loamy mcs                  17. Light clay
*      8. loamy fine sand            18. Silty clay
*      9. Sandy loam                 19. Heavy clay
*     10. Loess loam                 20. Peat

*     ALPHA in cm-1 (Rijtema/Driessen, simple and extended)
      DATA KMSA1T  /.1960, .1385, .0821, .0500, .0269, .0562, .0378,
     $              .0395, .0750, .0490, .0240, .0200, .0231, .0353,
     $              .0237, .0248, .0274, .0480, .0380, .1045/

*     a in cm**2.4/d (Rijtema/Driessen, extended)
      DATA KMSA2T    /.08,   .63,  3.30, 10.90, 15.00,  5.26,  2.10,
     $              16.40,   .24, 22.60, 26.50, 47.30, 14.40, 33.60,
     $               3.60,  1.69,  2.77, 28.20,  4.86,  6.82/

*     transition suction in cm (Rijtema/Driessen, extended)
      DATA KMSMXT    /80.0,  90.0, 125.0, 175.0, 165.0, 100.0, 135.0,
     $              200.0, 150.0, 130.0, 300.0, 300.0, 300.0, 200.0,
     $              300.0, 300.0, 300.0,  50.0,  80.0,  50.0/

*     saturated conductivity in cm/d (Rijtema/Driessen)
      DATA KSTT   /1120.0, 300.0, 110.0,  50.0,  1.00,   2.3,   .36,
     $              26.50, 16.50, 14.50,  12.0,  6.50,  5.00, 23.50,
     $               1.50,   .98,  3.50,  1.30,   .22,  5.30/

*     gamma, dimensionless (Rijtema/Driessen)
      DATA MSWCAT  /.0853, .0450, .0366, .0255, .0135, .0153, .0243,
     $              .0299, .0251, .0156, .0186, .0165, .0164, .0101,
     $              .0108, .0051, .0085, .0059, .0043, .0108/
```

```
*     saturated soil moisture content, dimensionless (Rijtema/Driessen)
      DATA WCSTT   /.3950, .3650, .3500, .3640, .4700, .3940, .3010,
     $              .4390, .4650, .4550, .5040, .5090, .5030, .4320,
     $              .4750, .4450, .4530, .5070, .5400, .8630/

*-----Soil type properties according to the van Genuchten system.
*     The 6 soil properties are given in data tables for NVGTYP soils.
*     (for declarations, see above).  By defining the soil type TYL(I)
*     for each layer I, a consistent combination of soil properties
*     is selected. In the case presented here, data refer to
*     a few soil types of the Lovinkhoeve (Peter de Willigen)
*     To be extended by user.
*
*      1. Lovinkhoeve 12b
*      2. Lovinkhoeve 16a

*     Van Genuchten form:
*     TETA-r, dimensionless
      DATA VGWRT /  .0448,    .0000/
*     TETA-s, dimensionless
      DATA VGWST /  .4012,    .4505/
*     K-s in cm/d
      DATA VGKST /  .7985, 25.134 /
*     ALPHA in cm-1
      DATA VGAT  /  .0036,    .0067/
*     N, dimensionless
      DATA VGNT  / 1.5007,  1.2318/
*     L, dimensionless
      DATA VGLT  /-2.418 ,   .0001/
*     end van Genuchten form

*-----initialize some other variables
      DATA ITOLD/4/, IRUN/0/, OKINIT/.FALSE./, IS/0/
      DATA OPENED /.FALSE.,.FALSE.,.FALSE.,.FALSE.,.FALSE./
*     character strings used for output file headers
      DATA STRWCL/'  WCL1',' WCL2',' WCL3',' WCL4',' WCL5',
     $            ' WCL6',' WCL7',' WCL8',' WCL9',' WCL10'/
      DATA STRTRW/' TRWL1',' TRWL2',' TRWL3',' TRWL4',' TRWL5',
     $            ' TRWL6',' TRWL7',' TRWL8',' TRWL9','TRWL10'/
      DATA STRFLX/'FLXQT1','    ..2','    ..3','    ..4','    ..5',
     $   '    ..6','    ..7','    ..8','    ..9','   ..10','   ..11'/


*-----initial section
*-------------------
      IF (ITASK.EQ.1) THEN
*        error check
         IF (ITOLD.NE.4) CALL ERROR ('DRSAWA',
     $    'cannot re-initialize model without terminal call')

*        SUSAWA model run number
         IRUN = IRUN + 1
*        local copy of unit number
         IF (IRUN.EQ.1) ILUNIT = IUNIT

*--------indicate to suslin that call is from FORTRAN main
         IWRIT = 1

*        **********************************
*        reading data from input file FILIN
*        **********************************
         CALL RDINIT (ILUNIT,IUNLOG,FILIN)
         WRITE (*,'(2A)') ' DRSAWA reads data from ',FILIN

*        reading of option switches
         CALL RDSINT ('SWIT3 ',SWIT3 )
         CALL RDSINT ('SWIT5 ',SWIT5 )
         CALL RDSINT ('SWIT6 ',SWIT6 )
         CALL RDSINT ('SWIT7 ',SWIT7 )
         CALL RDSINT ('SWIT8 ',SWIT8 )
         CALL RDSINT ('SWIT9 ',SWIT9 )
*        switch consistency check
         IF (SWIT9.EQ.2 .AND.
     $     ((SWIT3.EQ.1.AND.SWIT8.NE.1) .OR.
     $      (SWIT3.EQ.2.AND.SWIT8.NE.1) .OR.
     $      (SWIT3.EQ.3.AND.SWIT8.NE.2) .OR.
     $      (SWIT3.NE.1.AND.SWIT3.NE.2.AND.SWIT3.NE.3)))
     $    CALL ERROR ('DRSAWA','switch inconsistency SWIT3/8/9')

*--------time discretization
         CALL RDSREA ('DTMIN',DTMIN)
         CALL RDSREA ('DTMX1',DTMX1)
         IF (SWIT5.EQ.2) CALL RDSREA ('DTFX',DTFX )

*--------space discretization
*        compartment number + copy to output ; layer thickness
```

```
      CALL RDSINT ('NL',NL)
      INLAY = NL
*     exactly NL array elements required:
      CALL RDDATA (4,0,0,'x',IS,'TKL',TKL,10,NL)

*--------initial values of some state variables
*     moisture content; only for SWIT6=2
      IF (SWIT6.EQ.2) CALL RDDATA
     $ (4,0,0,'x',IS,'WCLQTM',WCLQTM,10,NL)
*     surface water depth
      CALL RDSREA ('WL0QTI',WL0QTI)
*     evaporation front depth
      CALL RDSREA ('ZEQTI ',ZEQTI )

*--------soil hydraulic conductivity and moisture characteristic
      IF (SWIT9.EQ.1) THEN
*        properties not automatically assigned by soil type but
*        quantified by user, per layer and per property
*        conductivity:
         DO 8 I=1,INLAY
            TYL(I) = 0.0
8        CONTINUE
         IF (SWIT3.EQ.1) THEN
            CALL RDDATA (4,0,0,'x',IS,'KST'   ,KST   ,10,NL)
            CALL RDDATA (4,0,0,'x',IS,'KMSA1',KMSA1,10,NL)
         ELSE IF (SWIT3.EQ.2) THEN
            CALL RDDATA (4,0,0,'x',IS,'KST'   ,KST   ,10,NL)
            CALL RDDATA (4,0,0,'x',IS,'KMSA1',KMSA1,10,NL)
            CALL RDDATA (4,0,0,'x',IS,'KMSA2',KMSA2,10,NL)
            CALL RDDATA (4,0,0,'x',IS,'KMSMX',KMSMX,10,NL)
         ELSE IF (SWIT3.EQ.3) THEN
            CALL RDDATA (4,0,0,'x',IS,'KST'   ,KST   ,10,NL)
            CALL RDDATA (4,0,0,'x',IS,'VGN'   ,VGN   ,10,NL)
         ELSE IF (SWIT3.EQ.4) THEN
            CALL RDDATA (4,0,0,'x',IS,'KST'   ,KST   ,10,NL)
            CALL RDDATA (4,0,0,'x',IS,'PN'    ,PN    ,10,NL)
         ELSE IF (SWIT3.EQ.5) THEN
*           user must specify conductivity parameters to be read
*           and include error check
            WRITE (*,'(2(/,A))')
     $      ' Reading of parameters for user-specified',
     $      ' conductivity function not properly adapted in DRSAWA'
            CALL ERROR ('DRSAWA','cannot read conductivity function')
         END IF

*-----------moisture characteristic
         IF (SWIT8.EQ.1) THEN
            CALL RDDATA (4,0,0,'x',IS,'WCST' ,WCST ,10,NL)
            CALL RDDATA (4,0,0,'x',IS,'MSWCA',MSWCA,10,NL)
         ELSE IF (SWIT8.EQ.2) THEN
            CALL RDDATA (4,0,0,'x',IS,'WCST',WCST,10,NL)
            CALL RDDATA (4,0,0,'x',IS,'VGA' ,VGA ,10,NL)
            CALL RDDATA (4,0,0,'x',IS,'VGL' ,VGL ,10,NL)
            CALL RDDATA (4,0,0,'x',IS,'VGR' ,VGR ,10,NL)
            CALL RDDATA (4,0,0,'x',IS,'VGN' ,VGN ,10,NL)
         ELSE IF (SWIT8.EQ.3) THEN
            CALL RDDATA (4,0,0,'x',IS,'WCST',WCST,10,NL)
*           read pF values
            CALL RDDATA (4,0,0,'x',IS,'PFWC00',PFWC00,10,NL)
            CALL RDDATA (4,0,0,'x',IS,'PFWC01',PFWC01,10,NL)
            CALL RDDATA (4,0,0,'x',IS,'PFWC02',PFWC02,10,NL)
            CALL RDDATA (4,0,0,'x',IS,'PFWC03',PFWC03,10,NL)
            CALL RDDATA (4,0,0,'x',IS,'PFWC04',PFWC04,10,NL)
            CALL RDDATA (4,0,0,'x',IS,'PFWC05',PFWC05,10,NL)
            CALL RDDATA (4,0,0,'x',IS,'PFWC06',PFWC06,10,NL)
            CALL RDDATA (4,0,0,'x',IS,'PFWC07',PFWC07,10,NL)
            CALL RDDATA (4,0,0,'x',IS,'PFWC08',PFWC08,10,NL)
            CALL RDDATA (4,0,0,'x',IS,'PFWC09',PFWC09,10,NL)
            CALL RDDATA (4,0,0,'x',IS,'PFWC10',PFWC10,10,NL)

*--------------check inputs for pF curve interpolation
            IERR = 0
            DO 10 I=1,NL
*              in descending order ?
               IF (PFWC01(I).GE.PFWC00(I)) IERR=IERR+1
               IF (PFWC02(I).GE.PFWC01(I)) IERR=IERR+1
               IF (PFWC03(I).GE.PFWC02(I)) IERR=IERR+1
               IF (PFWC04(I).GE.PFWC03(I)) IERR=IERR+1
               IF (PFWC05(I).GE.PFWC04(I)) IERR=IERR+1
               IF (PFWC06(I).GE.PFWC05(I)) IERR=IERR+1
               IF (PFWC07(I).GE.PFWC06(I)) IERR=IERR+1
               IF (PFWC08(I).GE.PFWC07(I)) IERR=IERR+1
               IF (PFWC09(I).GE.PFWC08(I)) IERR=IERR+1
               IF (PFWC10(I).GE.PFWC09(I)) IERR=IERR+1
*              start and end points
               IF (ABS(PFWC00(I)-7.).GT.TINY) IERR=IERR+1
               IF (ABS(PFWC10(I)   ).GT.TINY) IERR=IERR+1
10          CONTINUE
*           error occurred ?
            IF (IERR.NE.0) THEN
               WRITE(*,'(2(/,A),/,1X,I3,A)')
     $         ' Input pF-values for interpolation not in',
     $         ' descending order or invalid start/end points:',
     $         IERR,' ERROR(S) in pF-data points'
               CALL ERROR ('DRSAWA','Execution terminated')
            END IF

         ELSE IF (SWIT8.EQ.4) THEN
*           user must specify pf-curve parameters to be read
*           and include error check
            WRITE (*,'(2(/,A))')
     $      ' Reading of parameters for user-specified',
     $      ' pF-function not properly adapted in DRSAWA'
            CALL ERROR ('DRSAWA','cannot read pF-function')
         END IF

      ELSE IF (SWIT9.EQ.2) THEN
*        physical properties from soil type number
         CALL RDDATA (4,0,0,'x',IS,'TYL',TYL,10,NL)
      ELSE
         CALL ERROR ('DRSAWA','SWIT9 wrong value ; should be 1 or 2')
      END IF


*--------other soil parameters
*     soil evaporation properties
      CALL RDSREA ('CSC2  ',CSC2 )
      CALL RDSREA ('CSA   ',CSA  )
      CALL RDSREA ('CSB   ',CSB  )
*     maximum surface water storage
      CALL RDSREA ('WL0MX ',WL0MX )
*     index of tube/mole drained compartment
      CALL RDSINT ('IDRAIN',IDRAIN)
*     groundwater interpolation table and table length
      CALL RDAREA ('ZWTB',ZWTB,ILZMAX,IZWTB)

*     delete temporary file used by the input routines
      CLOSE (ILUNIT,STATUS='DELETE')
*     *******************************************
*     reading data from input file FILIN completed
*     *******************************************

*--------initialize local time, used for checking purposes
      LTIME = TIME

*--------initialize cumulative amounts and cumulative differences
      DRAICU = 0.0
      UPRICU = 0.0
      EVSWCU = 0.0
      RAINCU = 0.0
      RNOFCU = 0.0
      TRWCU  = 0.0
      WCUMCO = 0.0
      WCUMI  = 0.0
      WL0CO  = 0.0
      DO 20 I=1,NL+1
         FLXCU(I) = 0.
20    CONTINUE

*--------physical soil definition for type option
      IF (SWIT9.EQ.2) THEN
      DO 30 I=1,NL
*        soil type for this layer
         ITYL = NINT(TYL(I))
         IF (ITYL.LE.0) CALL ERROR ('DRSAWA','soil type <= 0')
         IF (SWIT3.EQ.1 .OR. SWIT3.EQ.2) THEN
*           soil parameters from Rijtema-Driessen tables
            IF (ITYL.GT.NRDTYP) CALL ERROR ('DRSAWA',
     $      'Rijtema-Driessen soil type number undefined')
            KMSA1(I) = KMSA1T(ITYL)
            KMSA2(I) = KMSA2T(ITYL)
            KMSMX(I) = KMSMXT(ITYL)
            KST(I)   = KSTT(ITYL)
            MSWCA(I) = MSWCAT(ITYL)
            WCST(I)  = WCSTT(ITYL)
         ELSE IF (SWIT3.EQ.3) THEN
*           soil parameters from Van Genuchten tables
            IF (ITYL.GT.NVGTYP) CALL ERROR ('DRSAWA',
     $      'Van Genuchten soil type number undefined')
            VGR(I)  = VGWRT(ITYL)
            WCST(I) = VGWST(ITYL)
            KST(I)  = VGKST(ITYL)
```

```
            VGA(I)   = VGAT(ITYL)
            VGN(I)   = VGNT(ITYL)
            VGL(I)   = VGLT(ITYL)
         END IF
30    CONTINUE
      END IF

      DO 40 I=1,NL
*        water contents at selected suction values
         FIELD = 1.0E2
         WILTP = 1.6E4
         AIRDR = 1.0E7
         CALL SUWCMS (I,2,WCFC(I),FIELD)
         CALL SUWCMS (I,2,WCWP(I),WILTP)
         CALL SUWCMS (I,2,WCAD(I),AIRDR)
*        copies of common variables to calling program
         WCADX(I) = WCAD(I)
         WCWPX(I) = WCWP(I)
         WCFCX(I) = WCFC(I)
         WCSTX(I) = WCST(I)
40    CONTINUE

*     depth of free groundwater level below soil surface on DAY
*     is initialized by lineair interpolation in table ZWTB
      ZWI   = LINT (ZWTB,IZWTB,DAY)

*     depth of water layer on surface and evaporation front
*     are initialised with initial values
      WL0QT = WL0QTI
      ZEQT  = ZEQTI

*--------initial call to water transport model
      CALL SUSAWA (ITASK,IUNLOG,WCLQT,WL0QT,NL,TRWL,EVSC,RAIN,ZWI,
     $             TKL,TYL,DELT,DTMIN,DTMX1,DTFX,IDRAIN,WL0MX,ZEQT,
     $             CSA,CSB,CSC2,WCLCH,WL0CH,WCLEQI,EVSW,RUNOF,UPRISE,
     $             DRAIQT,WCUMCH,ZECH,ZLT,FLXQT,PRHEAD,DTAV)

*--------initial water content
      IF (SWIT6.EQ.1) THEN
*        initialization at equilibrium moisture content
         DO 50 I=1,NL
            WCLQTI(I) = WCLEQI(I)
50       CONTINUE

      ELSE IF (SWIT6.EQ.2) THEN
*        initialization at observed initial moisture content
         DO 60 I=1,NL
            WCLQTI(I) = WCLQTM(I)
60       CONTINUE

      ELSE IF (SWIT6.EQ.3) THEN
*        initialization at wilting point
         DO 70 I=1,NL
            WCLQTI(I)=WCWP(I)
70       CONTINUE
      END IF

*--------initialize status
      DO 80 I=1,NL
*        total water content
         WCUMI    = WCUMI + WCLQTI(I) * TKL(I) * 1000.
*        initialization layer water content
         WCLQT(I) = WCLQTI(I)
80    CONTINUE
      WCUM = WCUMI

*--------open non-existing files ; set logicals for file use
      ILOUT = IOUT
      DO 90 I=1,5
         USEFIL(I) = MOD(ILOUT,10).GT.0
         IF (USEFIL(I).AND..NOT.OPENED(I)) THEN
*           file has yet to be opened
            IUN(I) = ILUNIT + I + 1
            WRITE (FILNAM,'(A,I1,A)') 'WATER',I,'.OUT'
            CALL FOPENG (IUN(I),FILNAM,'NEW','SF',0,'UNK')
            OPENED(I) = .TRUE.
         END IF
         ILOUT = ILOUT / 10
90    CONTINUE

      IF (USEFIL(1)) THEN
*        write header to water1.out
         ILF = ILEN (FILIN)
         WRITE (IUN(1),'(A,I3,A,/,A,/,2A)')
     $     ' SUSAWA RUN',IRUN,' ; SOIL STATUS',
     $     ' ========================',
     $     ' SOIL DEFINITION READ FROM FILE ',FILIN(1:ILF)
         IF (MOD(IOUT,10).EQ.2) THEN
*           write soil characteristics to this file
            WRITE (IUN(1),'(A,//,2A)')
     $       ' SOIL CHARACTERISTICS PER COMPARTMENT:',
     $       ' COMPARTMENT TYPE NR   TKL(M)   WCAD    WCWP',
     $       '    WCFC    WCST'
            DO 100 I=1,NL
               WRITE (IUN(1),'(3X,I4,8X,F4.0,4X,F5.3,3X,4(F5.4,3X))')
     $           I,TYL(I),TKL(I),WCAD(I),WCWP(I),WCFC(I),WCST(I)
100         CONTINUE
         END IF
         WRITE (IUN(1),'(/,A,10A6)')
     $     ' DAY   WL0    ZE   ',(STRWCL(I),I=1,NL)
         WRITE (IUN(1),'(A)') '          mm    mm'
      END IF

      IF (USEFIL(2)) THEN
*        write header to water2.out
         WRITE (IUN(2),'(A,I3,A,/,A,/,A,10A7)')
     $     ' SUSAWA RUN',IRUN,' ; TRANSPIRATION/COMPTMNT (MM/DAY)',
     $     ' ================================================',
     $     ' DAY  ',(STRTRW(I),I=1,NL)
         WRITE (IUN(2),'(1X)')
      END IF

      IF (USEFIL(3)) THEN
*        write header to water3.out
         WRITE (IUN(3),'(A,I3,A,/,A,/,A,//,2A,/)')
     $     ' SUSAWA RUN',IRUN,' ; DAILY WATER IN/OUT (MM/DAY)',
     $     ' ==============================================',
     $     ' Note: systems GAINS are POSITIVE, LOSSES NEGATIVE ',
     $     ' DAY    RAIN  RUNOF     TRW  (TRC)     EVSW (EVSC)   ',
     $     ' UPRISE DRAIQT  PROREL  SURREL'
      END IF

      IF (USEFIL(4)) THEN
*        write header to water4.out
         WRITE (IUN(4),'(A,I3,A,/,A,/,A,//,2A,/)')
     $     ' SUSAWA RUN',IRUN,' ; CUMULATIVE AMOUNTS (MM)',
     $     ' ==================================',
     $     ' Note:  system GAINS are POSITIVE, LOSSES NEGATIVE ',
     $     ' DAY     RAINCU   RNOFCU     TRWCU   EVSWCU   UPRICU ',
     $     '     DRAICU     WCUMCO      WL0CO '
      END IF

      IF (USEFIL(5)) THEN
*        write header to water5.out
         WRITE (IUN(5),'(A,I3,A,/,2A,/,A,11A6)')
     $     ' SUSAWA RUN',IRUN,
     $     ' ; FLUX THROUGH COMPARTMNT INTRFACES (MM/DAY)',
     $     ' ============',
     $     '==========================================',
     $     ' DAY  ',(STRFLX(I),I=1,NL+1)
         WRITE (IUN(5),'(1X)')
      END IF

*     initialization done
      OKINIT = .TRUE.


*-----rate calculation
*--------===============
      ELSE IF (ITASK.EQ.2) THEN
*        error check
         IF (.NOT.OKINIT) CALL ERROR ('DRSAWA','model not initialized')
*        check time against local time
         IF (LTIME.NE.TIME) CALL ERROR ('DRSAWA','inconsistent time')

*        groundwater level for current day
         ZW = LINT (ZWTB,IZWTB,DAY)

*--------rate call to water transport model
         CALL SUSAWA (ITASK,IUNLOG,WCLQT,WL0QT,NL,TRWL,EVSC,RAIN,ZW,
     $                TKL,TYL,DELT,DTMIN,DTMX1,DTFX,IDRAIN,WL0MX,ZEQT,
     $                CSA,CSB,CSC2,WCLCH,WL0CH,WCLEQI,EVSW,RUNOF,UPRISE,
     $                DRAIQT,WCUMCH,ZECH,ZLT,FLXQT,PRHEAD,DTAV)

*--------transpiration rate summed over all layers
         TRW = 0
         DO 110 I=1,NL
            TRW = TRW + TRWL(I)
110      CONTINUE

*--------dynamic output
         IF (USEFIL(1)) WRITE (IUN(1),'(F5.0,2F6.0,2X,10F6.3)')
```

```
     $     DAY,WL0QT*1000.0,ZEQT*1000.0,(WCLQT(I),I=1,NL)
         IF (USEFIL(2)) WRITE (IUN(2),'(F5.0,1X,10F7.1)')
     $     DAY,(TRWL(I),I=1,NL)
         IF (USEFIL(3)) WRITE
     $     (IUN(3),'(F5.0,2F7.1,2(F7.1,A,F5.1,A),F8.1,F7.1,2F8.1)')
     $     DAY,RAIN,RUNOF,TRW,' (',TRC,')',EVSW,' (',EVSC,')',
     $     UPRISE,DRAIQT,PROREL,SURREL
         IF (USEFIL(4)) WRITE (IUN(4),'(F5.0,6F9.1,1X,2F9.1)')
     $     DAY,RAINCU,RNOFCU,TRWCU,EVSWCU,UPRICU,DRAICU,WCUMCO,WL0CO
         IF (USEFIL(5)) WRITE (IUN(5),'(F5.0,1X,11F6.0)')
     $     DAY,(FLXQT(I),I=1,NL+1)


*-----integration of (dummy) rates
*-----=============================
      ELSE IF (ITASK.EQ.3) THEN
*        error checks
         IF (.NOT.OKINIT) CALL ERROR ('DRSAWA','model not initialized')
         IF (ITOLD.EQ.1) THEN
*           model was initialized in previous call,
*           integration call allowed as a dummy ; nothing done
            CONTINUE
         ELSE IF (ITOLD.EQ.2) THEN
*           in previous call, rates were calculated
*           integration of water status is allowed now

*           increase of local time
            LTIME = LTIME + DELT

*           water status variables ; surface water level
            WL0QT  = WL0QT  + WL0CH  * DELT
*           depth of evaporation front
            ZEQT   = ZEQT   + ZECH   * DELT
            DO 120 I=1,NL
*              water content per layer
               WCLQT(I) = WCLQT(I) + WCLCH(I) * DELT
120         CONTINUE

*           cumulative fluxes at NL+1 depths
            DO 130 I=1,NL+1
               FLXCU(I) = FLXCU(I) + FLXQT(I) * DELT
130         CONTINUE

*           cumulative amounts; gains positive, losses negative
            DRAICU = DRAICU + DRAIQT * DELT
            UPRICU = UPRICU + UPRISE * DELT
            EVSWCU = EVSWCU + EVSW   * DELT
            RAINCU = RAINCU + RAIN   * DELT
            RNOFCU = RNOFCU + RUNOF  * DELT
            TRWCU  = TRWCU  + TRW    * DELT

*           total water stored in profile
            WCUM   = WCUM   + WCUMCH * DELT
*           contribution of profile to water balance; since start
            PROREL = -WCUMCH
            WCUMCO = WCUMCO + PROREL * DELT
*           contribution of surface water to water balance; since start
            SURREL = -1000.* WL0CH
            WL0CO  = WL0CO + SURREL * DELT

*-----------total change in system water content
            CKWIN = -(WCUMCO + WL0CO)
*           total of external contributions to system water content
            CKWFL = RAINCU + RNOFCU + EVSWCU + TRWCU + UPRICU + DRAICU
*           check this
            CALL SUWCHK (CKWFL,CKWIN,LTIME)

         ELSE
*           error
            CALL ERROR ('DRSAWA','integration without valid rates')
         END IF


*-----terminal calculations
*-----=====================
      ELSE IF (ITASK.EQ.4) THEN
*        terminal output
         DO 140 I=1,5
            IF (USEFIL(I)) WRITE
     $        (IUN(I),'(A,///)') ' SIMULATION COMPLETED'
140      CONTINUE
*        model requires initialization from now on:
         OKINIT = .FALSE.

      ELSE
*        illegal switch
         CALL ERROR ('DRSAWA','1 < ITASK > 4')
      END IF

*     save task as previous task
      ITOLD = ITASK
      RETURN
      END
```

Appendix 4. Listing input file to DRSAWA

```
*****************************************************************
* User defined OPTION switches; all integers; dimensionless
*****************************************************************
* NB !!!!  Switches SWIT1, SWIT2, SWIT4 are not user-defined;
*          there are assigned values at lower level CALLs

* SWIT3   : conductivity function option switch
* NB !      at SWIT9=2, not all combinations of SWIT3 and SWIT8
*           are allowed; see comments at SWIT9
* SWIT3=1: simple Rijtema
* SWIT3=2: extended Rijtema
* SWIT3=3: Van Genuchten
* SWIT3=4: Power function
* SWIT3=5: user-defined; user must adapt program source
SWIT3=4   !!!!!!!!!!!!!!!!!!!!!

* SWIT5   : time step option switch
* SWIT5=1: Variable time step
* SWIT5=2: Fixed time step; DTFX must be specified
SWIT5 = 1  !!!!!!!!!!!!!!!!!!!!!

* SWIT6   : soil moisture initialization option switch
* SWIT6=1: initial soil moisture in hydrostatic equilibrium
*           with groundwater
* SWIT6=2: initial soil moisture entered as specified values;
*           WCLQTM must be specified
* SWIT6=3: initial soil moisture at wilting point
SWIT6 = 2  !!!!!!!!!!!!!!!!!!!!

* SWIT7   : lower boundary condition option switch:
* SWIT7=1: pressure head at lower boundary derived from
*           groundwater level (specified in ZWTB)
* SWIT7=2: free drainage
SWIT7 = 1  !!!!!!!!!!!!!!!!!!!!

* SWIT8   : moisture characteristic option switch; active at
*           both values of SWIT9
* NB !      at SWIT9=2, not all combinations of SWIT3 and SWIT8
*           are allowed; see comments at SWIT9
* SWIT8=1: Driessen
* SWIT8=2: Van Genuchten
* SWIT8=3: linear interpolation on log scale
* SWIT8=4: user-defined; user must adapt program source;
SWIT8=3     !!!!!!!!!!!!!!!!!!!

* SWIT9   : soil characterization option switch
* SWIT9=1: conductivity and moisture characteristic specified
*           by user through parameter values in this file;
*           all combinations of SWIT3 and SWIT8 are allowed
* SWIT9=2: conductivity and moisture characteristic derived by
*           SAWAH from soil type number only; soil type number
*           to be specified in TYL.
*           Only certain combinations of SWIT3 and SWIT8 are
*           allowed at this value of SWIT9:
*         SWIT3=1 and SWIT8=1 (simple Rijtema conductivity
*           combined with Driessen moisture characteristic)
*         SWIT3=2 and SWIT8=1 (extended Rijtema conductivity
*           combined with Driessen moisture characteristic)
*         SWIT3=3 and SWIT8=2 (Van Genuchten conductivity
*           combined with Van Genuchten moisture characterisic)
* NB1 !     for SWIT9=2, SWIT3=4/5 cannot be used
* NB2 !     for SWIT9=2, SWIT8=3/4 cannot be used
SWIT9=1     !!!!!!!!!!!!!!!!!!!
*****************************************************************


****************** time discretization **********************
* fixed time step; DTFX is only active when SWIT5=2:
  DTFX = 0.1                                    ! units: d
* relative minimum; active in all options:
  DTMIN = 0.0005                                ! units: d
* absolute maximum; active in all options:
  DTMX1 = 0.1                                   ! units: d
* Warning: too large DTFX (at SWIT5=2) and too large
*          DTMIN (SWIT5=1) will lead to oscillations
*          in simulated water contents. Both can be
*          0.01 d or larger for 'slow' soils (KST<10
*          cm/d), but should be decreased to 0.001 d
*          or lower for highly permeable soil (KST>
*          100 cm/d). (Values are based on TKL=0.1 m
*          for all compartments; for smaller TKL,
*          smaller DTFX and DTMIN are needed.
*          Check temporal behaviour of output water
*          contents.
*****************************************************************
```

```
***************** space discretization **********************
* number of compartments; maximum value 10;
* active in all options:
NL=10     !!!!    !!!!!!     !!!!!!!     !!!!!!!!    !!!!!!!!
* Thickness of soil compartments; the dimension of TKL in this
* data file should be equal to NL; active in all options:
TKL = 10*0.1                                        ! units: m
*****************************************************************


***************** initialization of state *******************
* initial depth of surface water layer; active in all options
WL0QTI = 0.20                                       ! units: m
* initial depth of evaporation front; active in all options
ZEQTI = 0.0                                         ! units: m
* initial moisture content; active only at SWIT6=2;
* NB ! values should not exceed WCST as specified in this file
* (for SWIT9=1) or in data base (for SWIT9=2); the dimension
* of WCLQTM should be equal to NL
WCLQTM = 10*0.5                                     ! units: -
*****************************************************************


***********  this section can be ignored if SWIT9=2  *********
*************** soil hydraulic conductivity  ****************
********* 'parameter' section: active only for SWIT9=1 *******
* NB ! The dimension of all parameters in this section should
*      be equal to NL !!!!!!!!!!
* active for all options SWIT3=1/2/3/4/5:
* saturated hydraulic conductivity
KST = 10., 0.10, 8*10.                           ! units: cm/d
* active for options SWIT3=1/2: parameter for simple
* and extended Rijtema
KMSA1 = 10*0.03                                  ! units: 1/cm
* active only for option SWIT3=2: parameters for extended
* Rijtema
KMSA2 = 10*3.6                             ! units: (cm**2.4)/d
KMSMX = 10*300.                                   ! units: cm
* active only for option SWIT3=3: parameter for Van Genuchten
VGN = 10*1.1
* active only for option SWIT3=4: parameter for power function
PN  = 10*-2.                                       ! units: -
********** end of conductivity 'parameter' section ***********
*****************************************************************


************* this section can be ignored if SWIT9=2  ********
**************** soil moisture characteristic ***************
****** 'parameter' section: active only for SWIT9=1 **********
* active for all options SWIT8=1/2/3/4:
* water content at saturation; volume fraction
WCST= 10*0.60                                       ! units -
* active only for option SWIT8=1: parameter for Driessen
MSWCA = 10*0.01                                     ! units -
* active only for option SWIT8=2: parameters for Van Genuchten
VGR = 10*0.01                                     ! units:  -
VGA = 10*0.1                                      ! units 1/cm
VGL = 10*-3.                                 ! units:  -
* active only for option SWIT8=3: parameters for
* interpolation. To be specified: the pF values corresponding
* to eleven given values of relative moisture content:
* PFWC00 for relative moisture content 0.0
* PFWC01 for relative moisture content 0.1, .... etc
* PFWC10 for relative moisture content 1.0
* NB1 ! Each line contains NL positions, one for each
*       compartment
* NB2 ! All values in the first line (PFWC00) should be 7.0
*       All values in the last line (PFWC10) should be 0.0
*       For any given position (=compartment), all values
*       should decrease monotonically from PFWC00 down to
*       PFWC10
PFWC00=10*7.
PFWC01=6*6.3,  2*6.4, 2*6.5
PFWC02=6*5.5,  2*5.8, 2*6.0
PFWC03=6*4.8,  2*5.3, 2*5.5
PFWC04=6*4.1,  2*4.8, 2*5.0
PFWC05=6*3.3,  2*4.0, 2*4.4
PFWC06=6*2.48, 2*3.90, 2*3.9
PFWC07=6*1.70, 2*3.33, 2*3.4
PFWC08=6*1.27, 2*2.76, 2*2.82
PFWC09=6*0.76, 2*1.48, 2*2.05
PFWC10=10*0.0
***** end of moisture characteristic 'parameter' section *****
*****************************************************************
```

```
********** this section can be ignored if SWIT9=1  ***********
************ hydraulic soil properties by soil type **********
******** 'soil type' section: active only for SWIT9=2  *******
* If SWIT9=2, physical soil characteristics are assigned to
* each compartment by only defining the soil type in each
* compartment. SAWAH makes use of a small data base,
* containing consistent sets of parameter values, tabulated in
* DRSAWA. See above comments at SWIT3, SWIT8 and SWIT9 in this
* file. For available types, see listing DRSAWA in manual .
* The dimension of TYL in this data file should be equal to NL
TYL = 1, 1, 2, 2, 6*2                          ! units: -
***************** end  of 'soil type' section ****************
**************************************************************


****************** other soil parameters ********************
*****************  active in all options ********************
* effective soil vapor diffusivity:
CSC2 = 0.1                                   ! units: cm2/d
* soil evaporation parameters:
CSA = 0.005                                  ! units: cm2/d
CSB = 1.0                                       ! ùnits:  -
* maximum layer of surface water WL0
WL0MX = 0.20                                    !  units: m
* index of compartment containing drain tubes/moles
* NB ! If no artificial drainage, set to zero !!!
IDRAIN = 0                                      !  units: -
**************************************************************


********************Groundwater depth ***********************
*************** active in all options ***********************
* Groundwater will be ignored only at the simultaneous
* occurrence of two conditions:
*  (1)  SWIT7=2
*  (2)  groundwater depth as specified here is deeper than
*       the lower profile boundary (=sum of TKL)
* If SWIT7=2, it is possible that groundwater is ignored only
* part of the simulated time due to part-time fulfillment of
* condition (2).
* The groundwater level is prescribed by means of a table
* containing datapoints in the form (DAY,LEVEL)
* The maximum number of datapairs is 365. If not all
* daynumbers are entered, SAWAH will interpolate between data
* points - linear.

ZWTB = 208., 0.815,
225.,.8,
226.,-.1,
227.,.006,
228.,0.,
229.,.30,
230.,0.6,
300.,10.8

*THAT IS ALL
```

Appendix 5. Listings of output files

WATER1.OUT

SUSAWA RUN 1 ; SOIL STATUS

SOIL DEFINITION READ FROM FILE VBINPUT.DAT
SOIL CHARACTERISTICS PER COMPARTMENT:

COMPARTMENT	TYPE NR	TKL(M)	WCAD	WCWP	WCFC	WCST
1	0.	0.100	.0100	.2311	.3969	.6000
2	0.	0.100	.0100	.2311	.3969	.6000
3	0.	0.100	.0100	.2311	.3969	.6000
4	0.	0.100	.0100	.2311	.3969	.6000
5	0.	0.100	.0100	.2311	.3969	.6000
6	0.	0.100	.0100	.2311	.3969	.6000
7	0.	0.100	.0100	.2847	.5156	.6000
8	0.	0.100	.0100	.2847	.5156	.6000
9	0.	0.100	.0100	.3235	.5415	.6000
10	0.	0.100	.0100	.3235	.5415	.6000

DAY	WL0 mm	ZE mm	WCL1	WCL2	WCL3	WCL4	WCL5	WCL6	WCL7	WCL8	WCL9	WCL10
225.	200.	0.	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500
226.	169.	0.	0.595	0.600	0.511	0.485	0.498	0.497	0.519	0.570	0.600	0.600
227.	159.	0.	0.600	0.600	0.600	0.600	0.600	0.600	0.600	0.593	0.600	0.600
228.	150.	0.	0.600	0.600	0.600	0.593	0.600	0.600	0.600	0.593	0.600	0.600
229.	141.	0.	0.600	0.600	0.600	0.600	0.600	0.600	0.600	0.593	0.600	0.600
230.	130.	0.	0.600	0.600	0.548	0.600	0.600	0.600	0.600	0.593	0.600	0.600
231.	120.	0.	0.600	0.600	0.536	0.537	0.539	0.548	0.600	0.593	0.600	0.600
232.	109.	0.	0.600	0.600	0.532	0.521	0.528	0.535	0.570	0.569	0.600	0.600
233.	98.	0.	0.600	0.600	0.529	0.510	0.519	0.529	0.566	0.569	0.572	0.600
234.	89.	0.	0.591	0.600	0.527	0.500	0.512	0.523	0.564	0.567	0.575	0.576
235.	78.	0.	0.600	0.600	0.525	0.492	0.506	0.517	0.562	0.563	0.574	0.573
236.	67.	0.	0.600	0.600	0.522	0.483	0.500	0.513	0.561	0.562	0.573	0.570
237.	57.	0.	0.600	0.600	0.520	0.475	0.495	0.508	0.559	0.561	0.572	0.568
238.	47.	0.	0.600	0.600	0.517	0.466	0.491	0.504	0.558	0.559	0.571	0.566
239.	37.	0.	0.600	0.600	0.514	0.458	0.487	0.501	0.557	0.558	0.570	0.565
240.	27.	0.	0.600	0.600	0.510	0.449	0.483	0.497	0.556	0.557	0.569	0.563

SIMULATION COMPLETED

WATER2.OUT

SUSAWA RUN 1 ; TRANSPIRATION/COMPTMNT (MM/DAY)

DAY	TRWL1	TRWL2	TRWL3	TRWL4	TRWL5	TRWL6	TRWL7	TRWL8	TRWL9	TRWL10
225.	-1.8	-1.8	-1.8	-1.8	0.0	0.0	0.0	0.0	0.0	0.0
226.	-1.8	-1.8	-1.8	-1.8	0.0	0.0	0.0	0.0	0.0	0.0
227.	-1.8	-1.8	-1.8	-1.8	0.0	0.0	0.0	0.0	0.0	0.0
228.	-1.8	-1.8	-1.8	-1.8	0.0	0.0	0.0	0.0	0.0	0.0
229.	-1.8	-1.8	-1.8	-1.8	0.0	0.0	0.0	0.0	0.0	0.0
230.	-1.8	-1.8	-1.8	-1.8	0.0	0.0	0.0	0.0	0.0	0.0
231.	-1.8	-1.8	-1.8	-1.8	0.0	0.0	0.0	0.0	0.0	0.0
232.	-1.8	-1.8	-1.8	-1.8	0.0	0.0	0.0	0.0	0.0	0.0
233.	-1.8	-1.8	-1.8	-1.8	0.0	0.0	0.0	0.0	0.0	0.0
234.	-1.8	-1.8	-1.8	-1.8	0.0	0.0	0.0	0.0	0.0	0.0
235.	-1.8	-1.8	-1.8	-1.8	0.0	0.0	0.0	0.0	0.0	0.0
236.	-1.8	-1.8	-1.8	-1.8	0.0	0.0	0.0	0.0	0.0	0.0
237.	-1.8	-1.8	-1.8	-1.8	0.0	0.0	0.0	0.0	0.0	0.0
238.	-1.8	-1.8	-1.8	-1.8	0.0	0.0	0.0	0.0	0.0	0.0
239.	-1.8	-1.8	-1.8	-1.8	0.0	0.0	0.0	0.0	0.0	0.0
240.	-1.8	-1.8	-1.8	-1.8	0.0	0.0	0.0	0.0	0.0	0.0

SIMULATION COMPLETED

WATER3.OUT

SUSAWA RUN 1 ; DAILY WATER IN/OUT (MM/DAY)

Note: systems GAINS are POSITIVE, LOSSES NEGATIVE

DAY	RAIN	RUNOF	TRW	(TRC)	EVSW	(EVSC)	UPRISE	DRAIQT	PROREL	SURREL
225.	0.0	0.0	-7.2	(-8.0)	-4.0	(-4.0)	28.2	0.0	0.0	0.0
226.	0.0	0.0	-7.2	(-8.0)	-4.0	(-4.0)	52.9	0.0	-47.5	30.5
227.	0.0	0.0	-7.2	(-8.0)	-4.0	(-4.0)	1.4	0.0	-51.8	10.2
228.	0.0	0.0	-7.2	(-8.0)	-4.0	(-4.0)	2.8	0.0	0.7	9.1
229.	0.0	0.0	-7.2	(-8.0)	-4.0	(-4.0)	-4.9	0.0	-0.7	9.0
230.	0.0	0.0	-7.2	(-8.0)	-4.0	(-4.0)	-18.4	0.0	5.2	10.9
231.	0.0	0.0	-7.2	(-8.0)	-4.0	(-4.0)	-9.1	0.0	18.8	10.8
232.	0.0	0.0	-7.2	(-8.0)	-4.0	(-4.0)	-5.5	0.0	9.6	10.6
233.	0.0	0.0	-7.2	(-8.0)	-4.0	(-4.0)	-4.3	0.0	6.2	10.5
234.	0.0	0.0	-7.2	(-8.0)	-4.0	(-4.0)	-2.3	0.0	6.0	9.5
235.	0.0	0.0	-7.2	(-8.0)	-4.0	(-4.0)	-1.9	0.0	2.2	11.3
236.	0.0	0.0	-7.2	(-8.0)	-4.0	(-4.0)	-1.5	0.0	2.8	10.2
237.	0.0	0.0	-7.2	(-8.0)	-4.0	(-4.0)	-1.3	0.0	2.6	10.1
238.	0.0	0.0	-7.2	(-8.0)	-4.0	(-4.0)	-1.2	0.0	2.5	10.0
239.	0.0	0.0	-7.2	(-8.0)	-4.0	(-4.0)	-1.0	0.0	2.4	9.9
240.	0.0	0.0	-7.2	(-8.0)	-4.0	(-4.0)	-0.9	0.0	2.4	9.8

SIMULATION COMPLETED

WATER4.OUT

SUSAWA RUN 1 ; CUMULATIVE AMOUNTS (MM)

Note: system GAINS are POSITIVE, LOSSES NEGATIVE

DAY	RAINCU	RNOFCU	TRWCU	EVSWCU	UPRICU	DRAICU	WCUMCO	WLOCO
225.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
226.	0.0	0.0	-7.2	-4.0	28.2	0.0	-47.5	30.5
227.	0.0	0.0	-14.4	-8.0	81.0	0.0	-99.3	40.7
228.	0.0	0.0	-21.6	-12.0	82.4	0.0	-98.6	49.8
229.	0.0	0.0	-28.8	-16.0	85.2	0.0	-99.3	58.8
230.	0.0	0.0	-36.0	-20.0	80.4	0.0	-94.1	69.7
231.	0.0	0.0	-43.2	-24.0	62.0	0.0	-75.3	80.5
232.	0.0	0.0	-50.4	-28.0	52.9	0.0	-65.6	91.1
233.	0.0	0.0	-57.6	-32.0	47.4	0.0	-59.5	101.6
234.	0.0	0.0	-64.8	-36.0	43.1	0.0	-53.5	111.2
235.	0.0	0.0	-72.0	-40.0	40.8	0.0	-51.3	122.4
236.	0.0	0.0	-79.2	-44.0	39.0	0.0	-48.4	132.7
237.	0.0	0.0	-86.4	-48.0	37.4	0.0	-45.8	142.8
238.	0.0	0.0	-93.6	-52.0	36.1	0.0	-43.3	152.8
239.	0.0	0.0	-100.8	-56.0	34.9	0.0	-40.9	162.8
240.	0.0	0.0	-108.0	-60.0	33.9	0.0	-38.5	172.6

SIMULATION COMPLETED

WATER5.OUT

SUSAWA RUN 1 ; FLUX THROUGH COMPARTMNT INTRFACES (MM/DAY)

DAY	FLXQT1	..2	..3	..4	..5	..6	..7	..8	..9	..10	..11
225.	27.	15.	3.	1.	0.	0.	1.	-1.	-8.	-18.	-28.
226.	6.	4.	2.	-9.	-22.	-32.	-42.	-51.	-53.	-53.	-53.
227.	5.	3.	2.	0.	-1.	-1.	-1.	-1.	-1.	-1.	-1.
228.	5.	3.	1.	0.	-3.	-3.	-3.	-3.	-3.	-3.	-3.
229.	7.	5.	3.	7.	5.	5.	5.	5.	5.	5.	5.
230.	7.	5.	3.	3.	7.	13.	18.	18.	18.	18.	18.
231.	7.	5.	3.	2.	1.	2.	4.	7.	9.	9.	9.
232.	7.	5.	3.	1.	1.	2.	2.	3.	3.	6.	6.
233.	6.	5.	3.	1.	0.	1.	2.	2.	2.	2.	4.
234.	7.	5.	3.	1.	0.	1.	1.	2.	2.	2.	2.
235.	6.	4.	3.	1.	0.	1.	1.	1.	1.	2.	2.
236.	6.	4.	3.	1.	0.	1.	1.	1.	1.	1.	2.
237.	6.	4.	2.	1.	0.	0.	1.	1.	1.	1.	1.
238.	6.	4.	2.	1.	0.	0.	1.	1.	1.	1.	1.
239.	6.	4.	2.	1.	0.	0.	1.	1.	1.	1.	1.
240.	6.	4.	2.	1.	0.	0.	1.	1.	1.	1.	1.

SIMULATION COMPLETED

Appendix 6. List of variables in alphabetic order

variable	*description*	*unit*
(* NB1	'a' refers to units before and after conversion by SUCONV, 'b' to units between the two SUCONV calls)	
(* NB2	only subroutine arguments are given	
A	See Press et al., 1986	-
AEXP	soil evaporation parameter	cm^2/d
BEXP	soil evaporation parameter	-
C2	effective soil vapour diffusivity	cm^2/d
CSC2	effective soil vapour diffusivity	cm^2/d
CKWFL	sum of time-integrated boundary fluxes	mm
CKWIN	change in water storage since start	mm
CSA	soil evaporation parameter	cm^2/d
CSB	soil evaporation parameter	-
D	See Press et al., 1986	-
DAY	calendar day number	-
DELH()	difference in hydraulic head over saturated compartment	hPa
DELT	time step (integration interval) in MAIN	d
DHH()	hydraulic head difference over saturated section	cm
DRAICU	cumulative drainage by drains	mm
DRAIDT	tube-/mole- drainage integrated over time	cm
DRAIQT	DELT-averaged rate of tube-/mole- drainage	a*: mm/d b*: cm/d
DT	time step for internal (dummy) integration	d
DTAV	DELT-averaged DT value	d
DTFX	fixed time step size	d
DTMIN	relative minimum time step	d
DTMX1	absolute maximum time step	d
DTMX2	relative maximum time step	d
EVSC	potential soil evaporation rate	a: mm/d b: cm/d
EVSW	actual soil evaporation rate	a: mm/d b: cm/d
EVSWCU	cumulative evaporation	mm
FILIN	name of file with soil data	-
FLUXDT()	flux at compartment interface integrated over time	cm
FLX()	soil water flux over compartment interface	cm/d
FLX1	DELT-averaged surface flux	cm/d
FLX2	DELT-averaged flux at bottom of top compartment	cm/d
FLXCU()	cumulative flux over compartment interfaces	mm
FLXQT()	DELT-averaged flux over compartment interface	a: mm/d b: cm/d
FLXSQ1	prospective water flux: resulting from excluding top compartment from saturated set	cm/d
FLXSQ2	prospective water flux: resulting from including extra compartment at top of saturated set	cm/d
FLXSTT()	tentative soil water flux through saturated set	cm/d
FLXUNT()	unsaturated soil water flux at compartment interface	cm/d
HGB()	gravitational head at bottom of compartment	cm
HGT()	gravitational head at top of compartment	cm
HPBP	pressure head at bottom of profile	cm
I	compartment index	-
IDRAIN	index of tube-/mole- drained compartment	-

INDX	See Press et al., 1986	-
INLAY	number of layers specified in input file	-
INXSAT()	index of saturated compartment (top=1)	-
IOUT	output control with five-digit integer number	-
ITASK	determines action of routine	-
IUNIT	unit number for output files	-
IUNLOG	unit number for log file	-
JJTOT()	number of compartments in saturated set	-
JTOT	index of saturated set (top=1)	-
KMS()	hydraulic conductivity	cm/d
MFLP()	matric flux potential	cm^2/d
MNR	message number	-
MS()	matric suction	cm
N	See Press et al., 1986	-
NL	number of soil compartments	-
PRHEAD()	pressure head at compartment center	hPa
PROREL	contribution of profile storage to water balance (positive when storage decreases)	mm/d
RAIN	rainfall/irrigation rate	a: mm/d b: cm/d
RAINCU	cumulative rainfall	mm
RUNOF	runoff rate	a: mm/d b: cm/d
RNOFCU	cumulative runoff	mm
SURREL	contribution of stored surface water to water balance (positive when storage decreases)	mm/d
SWIT1-9	see Section 3.9 of this manual	-
TIME	time	d
TKL()	thickness of soil compartment	a: m b: cm
TRWCU	cumulative transpiration	mm
TRWL()	uptake per compartment due to transpiration	a: mm/d b: cm/d
TYL()	type of soil per compartment	-
UPRICU	cumulative capillary rise over lower soil boundary	mm
UPRISE	water flux over lower profile boundary	a: mm/d b: cm/d
WCADX()	volumetric water content 'air-dry'	-
WCFX()	volumetric water content at field capacity	-
WCL()	volumetric soil water content	-
WCLCH()	rate of change of water content, averaged over DELT	1/d
WCLEQI()	water content at hydrostatic equilibrium with groundwater level	-
WCLQT()	volumetric soil water content	-
WCLQT1	volumetric water content compartment 1	-
WCSTX()	volumetric water content at saturation	-
WCUMCH	rate of change in amount of stored soil water, averaged over DELT	mm/d
WCUMCO	contribution of soil storage term to overall water balance	mm
WCWPX()	volumetric water content at wilting point	-
WLØ	depth of water layer on surface	a: m b: cm
WLØCO	contribution of surface storage term to overall water balance	mm
WLØMX	maximum depth of surface water	a: m b: cm
WLØQT	depth of surface water layer	m

WLØCH	rate of change of surface water depth, WLO, averaged over DELT	m/d
X	value of variable to be checked	variable
XMAX	maximum allowable value of X	variable
XMIN	minimum allowable value of X	variable
ZE	depth of evaporation front below soil surface	a: m b: cm
ZECH	rate of change of evaporation front, averaged over DELT	m/d
ZEQT	depth of evaporation front	m
ZL()	depth of top of soil compartment	a: m b: cm
ZLT	total depth of profile	a: m b: cm
ZW	depth of free groundwater level below soil surface	a: m b: cm

Appendix 7. Soil evaporation: derivation of main equations

A7.1 Derivation of basic relations used in SAWAH

In subsection 4.3.3 the concept of evaporation front has been introduced. The depth of this front, z_E, plays a key role in the formulation of soil evaporation as incorporated in the SAWAH model. The main equations are repeated here:

(A1) $e = c_2/z_E$

(A2) $dz_E/dt = c_3/z_E$

(A3) $c_3 \equiv c_2/((\theta_i-\theta_{ad})+c_1/c_2)$

(A4) $c_1 \equiv A\,(e^{B(\theta_i/\theta_s-c_4)} - 1)$

where the symbols are defined as		*dimension*
e	actual soil evaporation rate	L/T
z_E	depth of evaporation front	L
θ	volumetric moisture content	-
t	time	T
c_2	constant to express vapour diffusivity	L^2/T
c_4	empirical constant	-
A	empirical constant (scale)	L^2/T
B	empirical constant (shape)	-
i	subscript, referring to initial condition	
ad	subscript, referring to air-dry condition	

The variable c_1 is a measure of liquid supply to the evaporation front, high values being associated with ample supply from the underlying soil. From Eqs A2-A3 it appears that z_E increases rapidly when c_1/c_2 is small, i.e. when vapour removal through the dry topsoil is high as compared to liquid supply to the evaporation front. A rapid increase *in* z_E, in turn, implies a fast decrease in actual evaporation in time, according to Eq A1. *This appendix shows the derivation of the above set of equations.*

Let a semi-infinite soil column at initial moisture content θ_i be subjected to a potential evaporation rate e_{pot} at the surface. The semi-infinite character implies a free drainage condition at high values of z. Flow is confined to one-dimensional liquid flow in the vertical. The integral form of the mass conservation equation may then be written in terms of actual evaporation rate, hydraulic conductivity k (L/T) and initial moisture content θ_i,

(A5) $$\int_{t=0}^{t} e\, dt + k_i\, t = \int_{z=0}^{\infty} (\theta_i-\theta)\, dz$$

with z the space coordinate (L), positive downward, and k_i the value of k at $\theta=\theta_i$. So, at time t, the total amount of water lost due to evaporation (first LHS term) and deep drainage (second LHS term) equals the change in volumetric moisture content, integrated over depth (Fig. A1, shaded area).

We now wish to transform A5 into a relation expressing *e* as a function of inherent soil properties, initial conditions and potential evaporation. If we make use of the evaporation front which after some time will separate air-dry topsoil (with exclusively vapour transport) from moist subsoil (with exclusively liquid transport), the RHS of A5 becomes

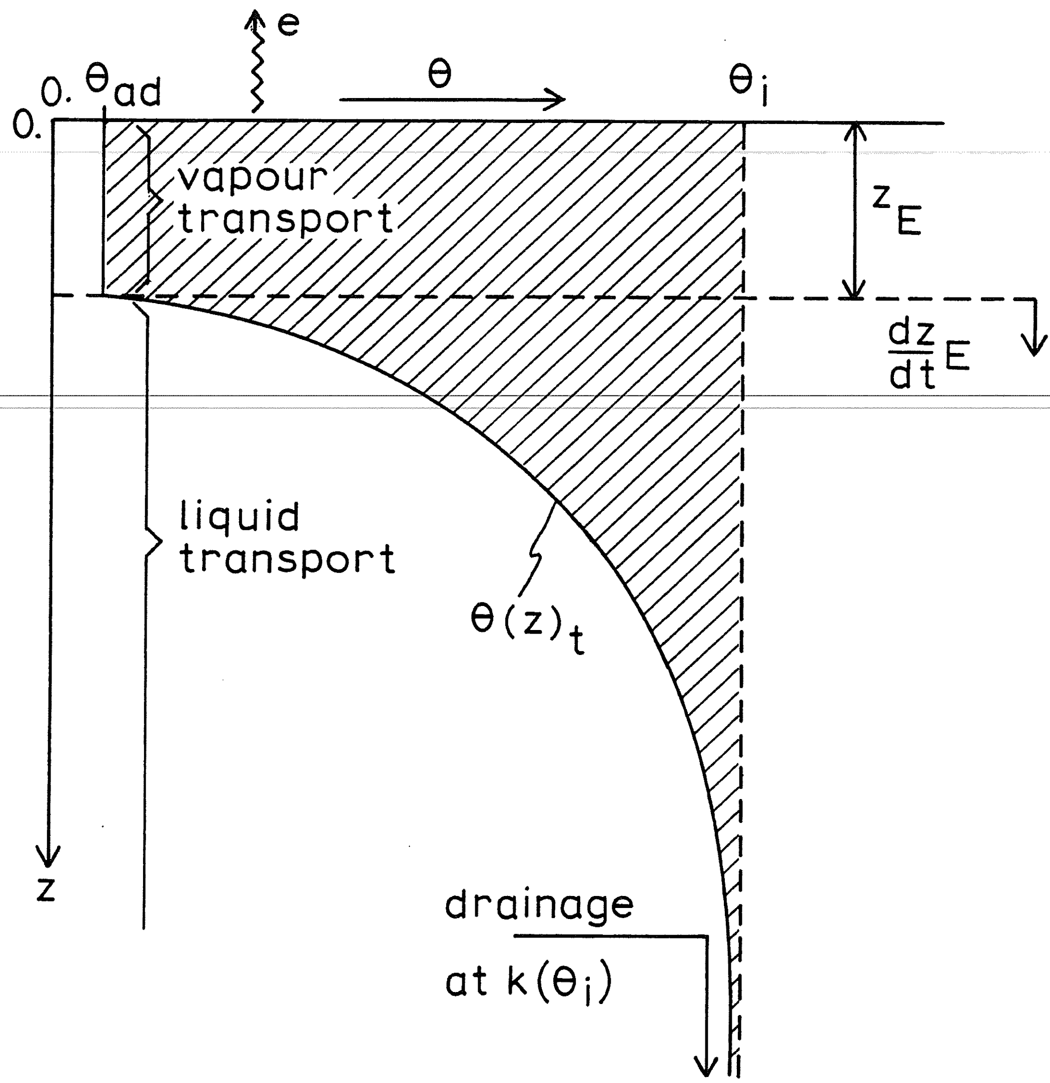
$$\text{(A6)} \qquad \int_{z=0}^{\infty} (\theta_i - \theta)\, dz = z_E\, (\theta_i - \theta_{ad}) + \int_{z=z_E}^{\infty} (\theta_i - \theta)\, dz$$

Now an expression in θ is sought for the differential d*z*. Giraldez (1986, pers. comm.) suggested that such a relation may be derived from infiltration theory (Parlange, 1971; applications by Smith and Parlange, 1978; Giraldez and Sposito, 1985). The approach is as follows.

The general flow equation for liquid in soil can be written in 'diffusivity form' as

$$\text{(A7)} \qquad \frac{\partial \theta}{\partial t} = \frac{-\partial}{\partial z} \left(-D(\theta) \frac{\partial \theta}{\partial z} + k(\theta) \right)$$

with $D(\theta)$ the soil water diffusivity (L^2/T). For notation sake, $D(\theta)$ and $k(\theta)$ will further be written as *D* and *k*, for short. In order to obtain an approximation of the slope of the moisture profile below the evaporation front, $d\theta/dz$, Eq A7 is simplified to



Figuur A1. Schematization of topsoil drying and development of evaporation front

(A8) $$\frac{d}{dz}\left(-D \frac{d\theta}{dz} + k\right) = 0$$

thus ignoring the transient term $\partial\theta/\partial t$. In other words, it is assumed that in the region below the evaporation front, where a moisture content gradient is established, the rate $(\partial\theta/\partial t)_z$ is small for all z. Eq A8 can then be integrated to yield the diffusivity form of the Darcy equation:

(A9) $$-D \frac{d\theta}{dz} + k = c$$

where the integration constant c is recognized as the flux at the evaporation front z_E for a steady state situation:

(A10) $$c = -e$$

So it follows that

(A11) $$dz = \frac{D\, d\theta}{e+k}$$

This approximation should be equally valid for surface evaporation (z_E=0) as for subsurface evaporation (z_E> 0). Since for ($z =< z_E$) the moisture content $\theta=\theta_{ad}$, and for ($z \rightarrow \infty$) $\theta=\theta_i$, combination of Eqs A5, A6 and A11 gives

(A12) $$k_i\, t + \int_{t=0}^{t} e\, dt = z_E\, (\theta_i - \theta_{ad}) + \int_{\theta=\theta_{ad}}^{\theta_i} \frac{(\theta_i - \theta)\, D}{e + k}\, d\theta$$

The effect of gravity on the shape of the moisture profile in a drying soil will be small just below the evaporation front, in the drier region of the soil. There, suction gradients dominate the flow. Omission of k in the second RHS terms is therefore warranted (see also section A7.3).

The vapour flux density q_v in soil above the front z_E is expressed by Fick's law as

(A13) $$q_v = -D_v \frac{dc}{dz}$$

where D_v is the effective vapour diffusivity in air dry soil and c is the vapour concentration. For an inert 'air-dry' soil layer (no condensation/evaporation above z_E), Eq A13 simplifies upon integration to

(A14) $$q_v = -D_v\, (c(z_E) - c(atm)) / z_E$$

with $c(z_E)$ the vapour concentration at depth z_E, and c(atm) the concentration of water vapour above the soil surface. It has been implied here that D_v is constant with depth in the dry surface layer. Although $c(z_E)$ depends on soil temperature and c(atm) on atmospheric humidity, we simplify A14 for practical purposes to

(A15) $e = c_2/z_E$

which is Eq A1. The vapour flux density q_v (positive downward) of Eq A14 has been renamed e, the actual soil evaporation (positive for upward vapour flux).

It must be noted that A15 is only valid as soon as the soil becomes limiting to evaporation, i.e. as soon as $z_E > c_2/e_{pot}$. In reality, the soil drying process passes three stages: a first stage when actual evaporation is limited by potential evaporation (evaporative demand); then a short transition stage when water supply by the soil becomes limiting but no visible permanent dry layer is present; and a third stage when evaporation takes place by vapour diffusion through the dry surface layer. In this appendix and in SAWAH, the first two stages are formally skipped: drying starts immediately with the development of a thin dry surface layer.

Further elaboration is based on A15, which causes some inconsistency when the resulting set of equations A1-A4 is applied to the whole drying sequence. In Section A7.3 it will be argued that the error associated with this inconsistency is expected to be small.

After deleting the gravity term on the RHS of A12 for reasons indicated above, we obtain from a combination of Eqs A12 and A15

(A16) $$E = - k_i\ t + z_E\ (\theta_i - \theta_{ad}) + \frac{z_E}{c_2} \int_{\theta_{ad}}^{\theta_i} (\theta_i - \theta)\ D\ d\theta$$

or

(A17) $$E = - k_i\ t + z_E\ ((\theta_i - \theta_{ad}) + c_1/c_2)$$

with E the cumulative evaporation as

(A18) $$E \equiv \int_{t=0}^{t} e(t)\ dt$$

and

(A19) $$c_1 \equiv \int_{\theta_{ad}}^{\theta_i} (\theta_i - \theta)\ D\ d\theta$$

Intermezzo

So far, the theory was based on the assumption that real soils can be represented as 'semi-infinite', homogeneous, free draining columns:
- the 'semi-infinite' character implies that processes at one boundary (evaporation at the top end) do not affect the state of the system at the other boundary (lower profile boundary)
- homogeneity implies that both the initial state and the inherent properties of the soil do not depend on the depth z;
- free drainage implies that at the lower end of the soil column, water is lost at a rate k_i.

Furthermore, it was assumed that no other mechanism of water loss from soil, such as uptake by roots, is present.

Real field soils are different and usually do not fully satisfy any of these conditions. In layered soils, in the presence of shallow groundwater, and when plants extract water, assumptions are violated. The water content near the surface is then affected by other processes than evaporation and free drainage. SAWAH copes with the problem by three further simplifications:

(1) the theory is applied to the top soil compartment only, *not* to the entire profile, thus partly solving the heterogeneity problem.

(2) in the above formulations, the initial water content θ_i is replaced by the actual water content θ_1 of the top compartment; θ_1 is viewed as θ_i during the period of a time step DELT; it is then updated by integration as described in Chapter 4.

(3) the drainage term $k_i t$ is not fixed but is replaced by the water flux q over the first compartment's lower boundary, calculated numerically as explained in Section 4.2.

The consequence of combining (1) and (2) is that soil compartments must be chosen relatively thick, so as to ensure that the semi-infinite character of the first compartment is valid during the time period DELT. On the other hand, too thick compartments affect the spatial resolution of predictions. In many cases, a top compartment of 0.1 m depth is a good tradeoff.

Updating soil moisture content as in (2) would not be necessary if water content were affected by evaporation only *and* the first compartment would be semi-infinite during the whole of the simulated period. Then, the value of θ_1 at the start of simulation could be used throughout. Since this is not the case, however, a correction is made by resetting moisture content after each DELT, but retaining the value of z_E already attained. This does indeed correct evaporation in the right direction, but the accuracy of this approximation will depend on several factors. A detailed simulation study of the evaporation process could be helpful.

Step (3) connects the local description of the top soil compartment to the numerical solution of the flow equation for the rest of the profile.

End intermezzo

After elimination of the term $k_i t$, the cumulative evaporation E becomes a simple linear function of z_E, the proportionality constant being a function of initial moisture content and of liquid and vapour diffusivities. Differentiating Eq A17 with respect to time and invoking the differential form of A18 gives

$$\text{(A20)} \qquad e = \frac{dz_E}{dt} \left((\theta_i - \theta_{ad}) + c_1/c_2\right)$$

Finally, introducing Eq A15 into A20 gives

$$\text{(A21)} \qquad \frac{c_2}{z_E} = \frac{dz_E}{dt} \left((\theta_i - \theta_{ad}) + c_1/c_2\right)$$

which is then rearranged to the simple form of Eq A2, with c_3 defined as in Eq A3. The development of z_E in time then follows directly as

$$\text{(A22)} \qquad (z_E)^2 = 2\ c_3\ t + (z_{E,i})^2$$

with $z_{E,i}$ the initial value of z_E.

The subroutine SUZECA calculates for every tine step DELT the increase in z_E from Eq A22:

$$\text{(A23)} \qquad z_E(t+\text{DELT}) = \sqrt{2\ c_3\ \text{DELT} + z_E^2(t)}$$

thereby using the actual moisture content of the first compartment instead of θ_i, in deriving c_3 according to Eq A3. With θ_i changing in time, the coefficients c_1 and c_3 need to be updated at every DELT in a simulation run. An analytical relation is used to express c_1 in terms of θ_i. Landeros Sanchez (1987) calculated this integral for soils varying over a wide range of textures and showed that the expression given in Eq A4 closely approximates the curve.

A7.2 Square-root-of-time relations vs Z_E formulation

When substituted in A1, Eq A22 with $z_{E,i}$=0 supports the familiar inverse proportionality of e vs $\sqrt{t}$. It follows also directly from A17 and A22 that the 'desorptivity' S_d in S_d=$E/\sqrt{t}$ is then equal to

$$\text{(A24)} \qquad S_d = \sqrt{2\ (c_2(\theta_i - \theta_{ad}) + c_1)}$$

So, the 'desorptivity' S_d appears to be a function of liquid and vapour diffusivities, and initial moisture content. This is not surprising, and similar functions for sorptivity and desorptivity have been given before. For homogeneous deep soil profiles at constant initial moisture content in an uninterrupted drying sequence, a simple $\sqrt{t}$ relation could thus be used, with S_d as the proportionality factor. Indeed, formulations A1-A4 as employed in SAWAH should result in a linear E-$\sqrt{t}$ relation when applied to continued drying of homogeneous soils.

One might ask 'why not discard the z_E concept and use simply S_d - evaluated on a daily basis- in a direct calculation of E ?' (as an alternative to the daily integration of dz_E/dt in SAWAH). Then it should be remembered that the initial value of z_E was set to zero to derive Eq A24. An attempt to replace the z_E formulation of e by an S_d formulation would lead to a series of terms in t, each with its own c_3 coefficient. So, instead of this series, the state variable z_E serves a memory function to the system. From day to day the conditions may change. The 'history' of evaporation is 'stored' in the form of the value of z_E. This can not be achieved by simply keeping an account of cumulative evaporation, time, or cumulative potential evaporation (e.g. Boesten and Stroosnijder, 1986); all these approaches assume an *a priori* fixed relation between actual evaporation on the one hand, and time or cumulated potential evaporation on the other. In the present approach this relation is variable in time.

Moreover it can be expected that in situations of alternate wetting and drying, a formulation based on z_E lends itself better to field validation.

Summarizing, it can be said that the proposed description of evaporation:

(1) saves computation time as compared to precise numerical calculations, which demand thin surface compartments;

(2) has a much wider range of application than purely analytical descriptions, because interactions with the subsoil are taken into account numerically;

(3) simplifies to the well known $E=S_d\sqrt{t}$ relation for uniform initial conditions and continued drying; and then relates S_d to basic soil properties.

A7.3 First stage and transition stage

In Section A7.1, several aspects have been ignored. As mentioned before, SAWAH disregards 'first stage drying' in the sense that z_E immediately starts growing after the surface floodwater has disappeared. On the basis of Eq A1 only, evaporation would attain very high values at the start of drying. SAWAH then truncates the value of e to e_{pot} as happens in reality. In this sense, SAWAH also recognizes a first stage of drying. In deriving the rate dz_E/dt, however, this truncation is not maintained and therefore the development rate of z_E is overestimated. Two arguments can be put forward to justify the use of Eqs A2-A3, irrespective of this overestimation.

First, the gravity term k was ignored in the transition A12-A16. Although suction gradients are dominant over gravity near the drying front, ignoring the latter force leads to an overestimation of c_1 and therefore to underestimation of dz_E/dt. So two ignored factors tend to compensate each other.

The second argument applies only to free draining soils. It concerns the **duration** of the first stage of drying: for most soils subject to natural evaporation this stage appears to be short. This duration is directly obtained from Eq A12:

$$\text{(A25)} \quad t_c = \frac{1}{k_i + e_{pot}} \int_{\theta=\theta_{ad}}^{\theta_i} \frac{(\theta_i - \theta)\, D(\theta)}{e + k(\theta)}\, d\theta$$

where t_c is the critical time marking the end of stage 1 (Giraldez, 1986). Light and heavy-textured soils under free drainage and conducive evaporative conditions (5 mm/d) show a t_c of less than a day (based on Eq A25). Only for medium-textured soils, and in general under a crop cover, a duration of several days may be found. During this time lapse, dz_E/dt would be overestimated and therefore e would decrease too soon. At low values of e_{pot}, as under a crop canopy, the evaporative loss term itself is small so that underestimation of e due to too high dz_E/dt would not be important in absolute terms. If the user considers this approximation too rough, it suffices to replace c_2 by $(z_E \cdot e_{pot})$ for $z_E < c_2/e_{pot}$, in the subroutine SUZECA.

Appendix 8. Selection procedure for fluxes at saturated-unsaturated interfaces: graphical representation

The figures in this appendix illustrate the operations by SUSEFL as described in Subsection 4.2.4.2. In reading the figures, the following should be noted:

1. Each *situation class* is represented by one graphic block. The identification of the situation class is given in the upper right hand corner of the block and refers to the classification system described in Subsection 4.2.4.2 of this manual.

2. Each picture block consists of a left hand side and a right hand side. The left hand side presents the *situation class*; it gives the combinations of FLXSTT, FLXUNT and FLXSQ that may possibly occur within that class. Of these, one or two are taken as example realizations (heavy arrows). If *two* example realizations are depicted, they are numbered 1 and 2. The right hand side of each picture block shows the outcome of the selection procedure by SUSEFL, that applies to the example realization given on the left. If two solutions are given, they are numbered in consistence with the left hand side: solution 1 on the right refers to realization 1 on the left.

3. The left hand side fluxes are calculated in SUUNST and SUSTFL, the right hand side fluxes in SUSEFL.

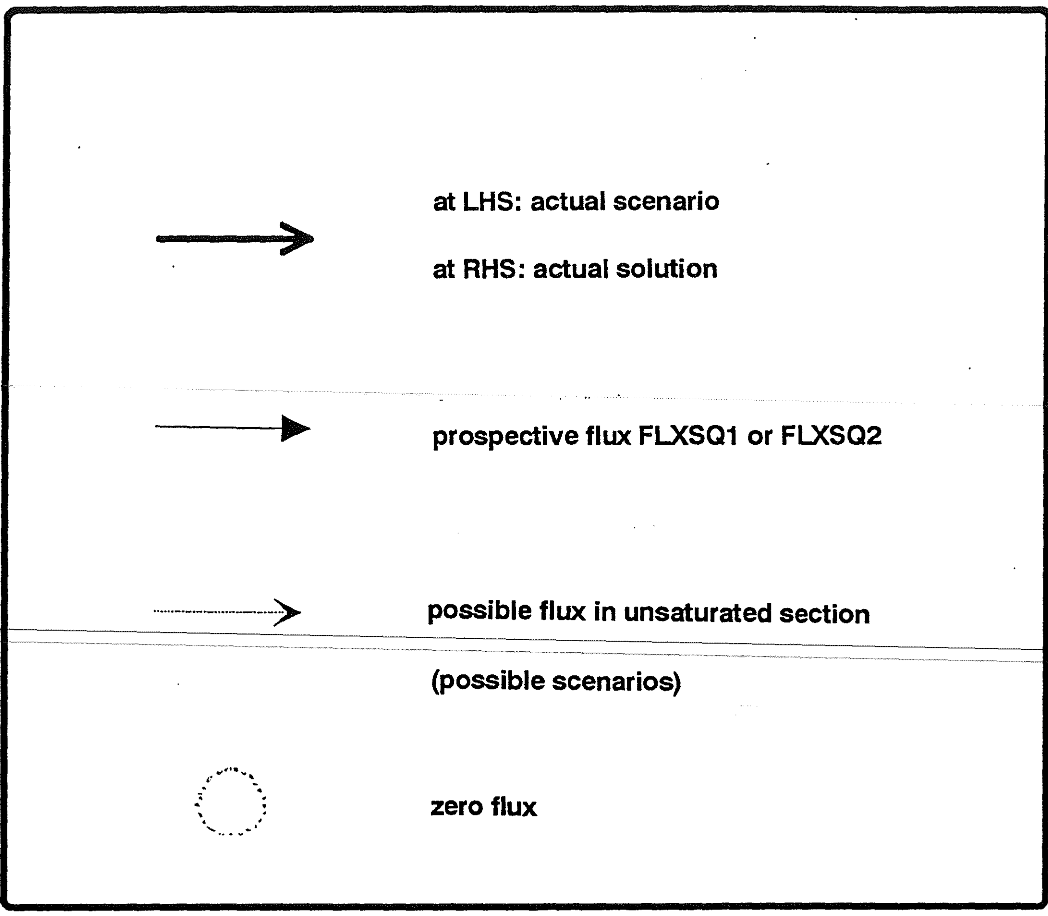
4. Arrow types are explained in the first picture block.

5. Relative arrow lengths represent relative flux sizes. Only three lengths of arrows are used in the pictures. This is the minimum required to show all combinations of 'larger than', 'smaller than', and 'equal to'. In reality and in the model, however, fluxes may attain any value on a continuous scale.

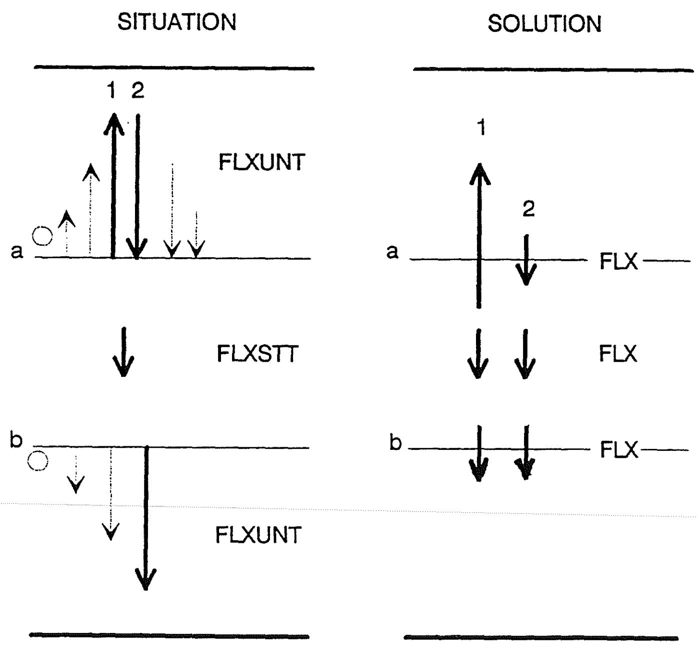
6. Heavy horizontal lines indicate profile boundaries. Thin horizontal lines indicate boundaries of saturated sections: a (top) and b (bottom). Compartment interfaces *within* saturated and unsaturated sections have been omitted from the pictures. They are not relevant to the analysis.

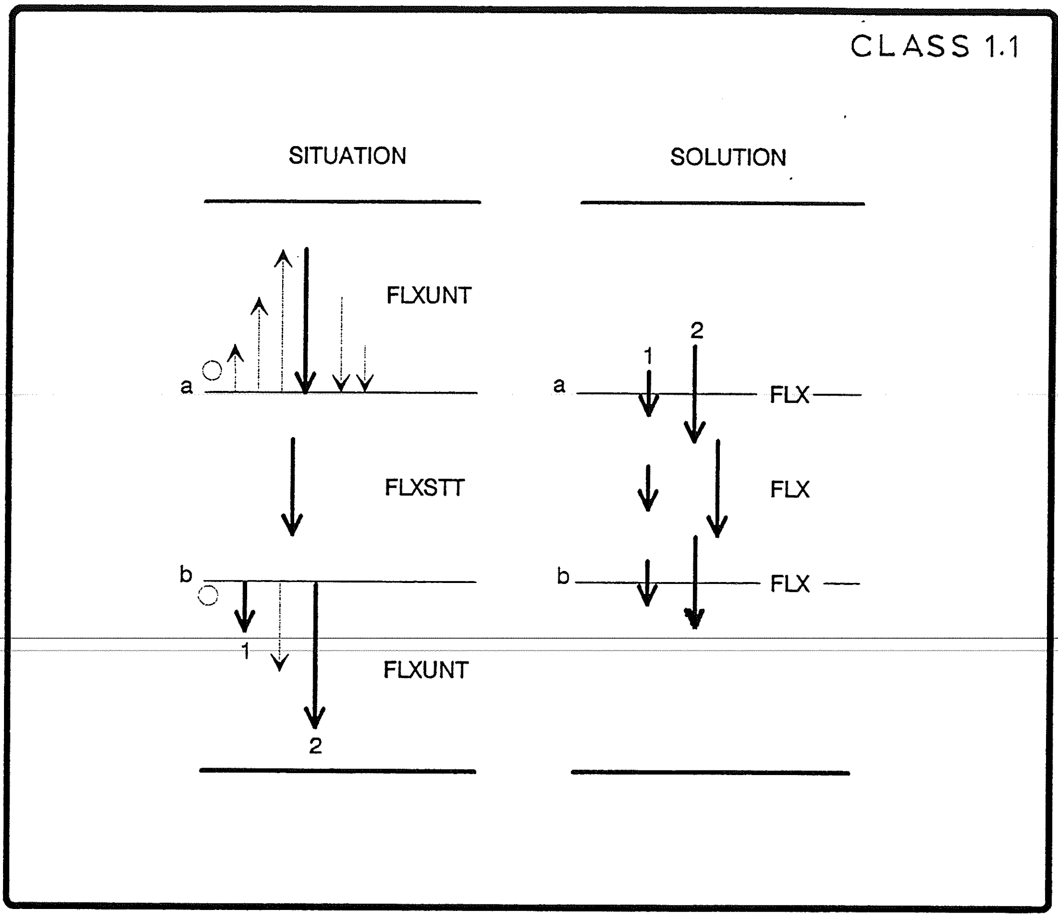
7. The arrows FLXUNT refer to the tentative fluxes *at* the saturated-unsaturated interface, not to the compartment inter-faces within the unsaturated section. The arrows FLXSTT refer to the tentative fluxes *at* the saturated-unsaturated interface *and* to the *internal* compartment interfaces within the saturated section. Also the fluxes FLX refer to both types of interfaces.

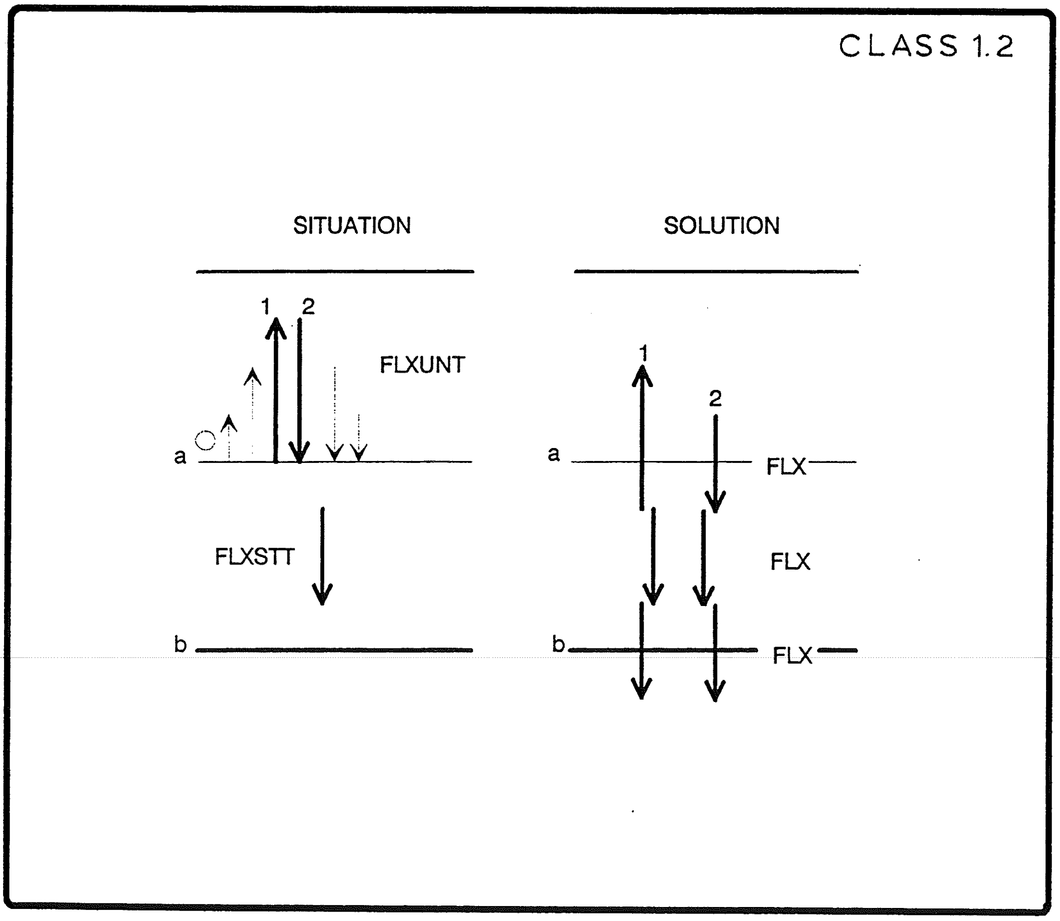
8. The prospective fluxes required for cases with groundwater within the profile are indicated by a special type of arrow. For explanation see Subsection 4.2.4.2.


at LHS: actual scenario
at RHS: actual solution
prospective flux FLXSQ1 or FLXSQ2
possible flux in unsaturated section
(possible scenarios)
zero flux

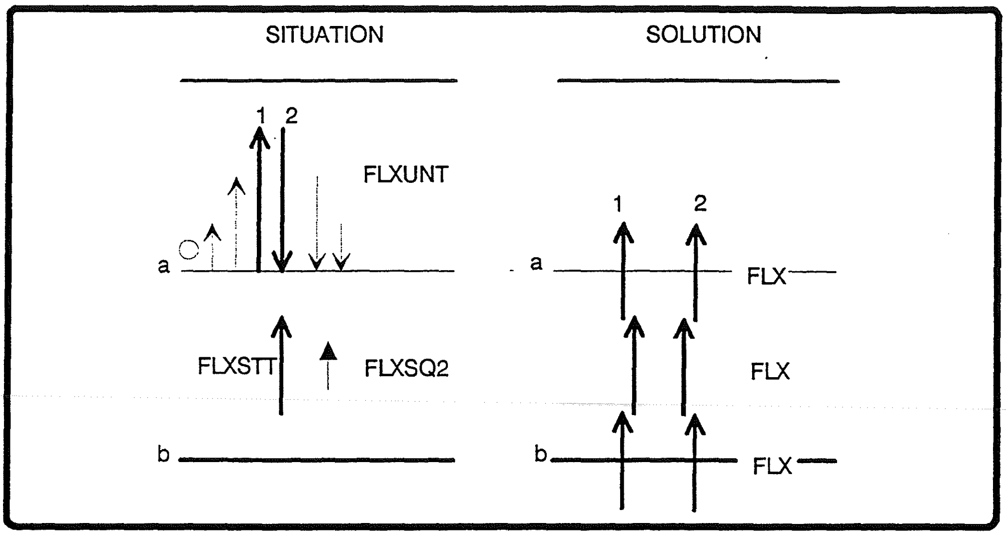
CLASS 1.1


SITUATION
SOLUTION
1 2
FLXUNT
a
FLXSTT
b
FLXUNT
1
2
a
FLX
FLX
b
FLX

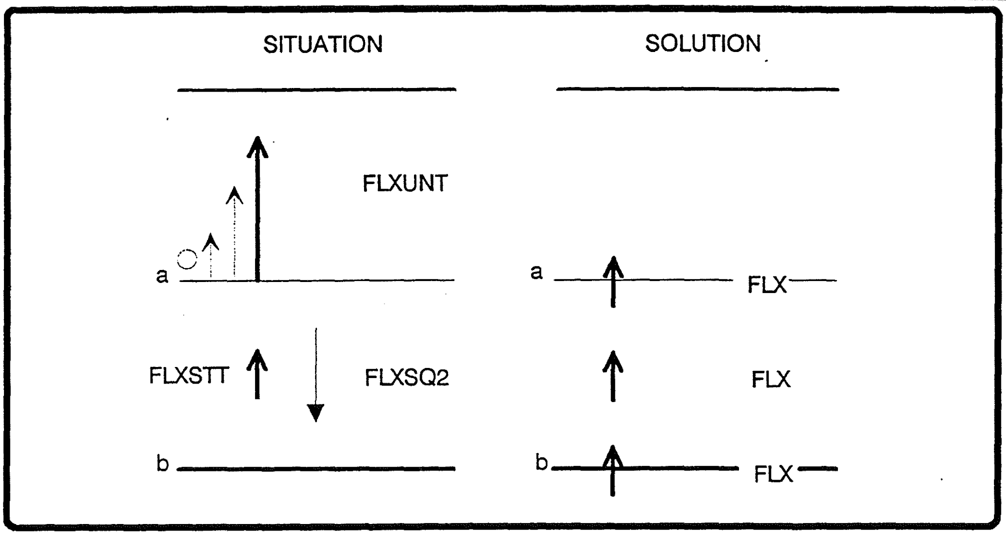

CLASS 1.1
SITUATION
SOLUTION
FLXUNT
a
FLXSTT
b
1
FLXUNT
2
2
1
a
FLX
FLX
b
FLX


CLASS 1.2
SITUATION
SOLUTION
1
2
FLXUNT
a
FLXSTT
b
1
2
a
FLX
FLX
b
FLX

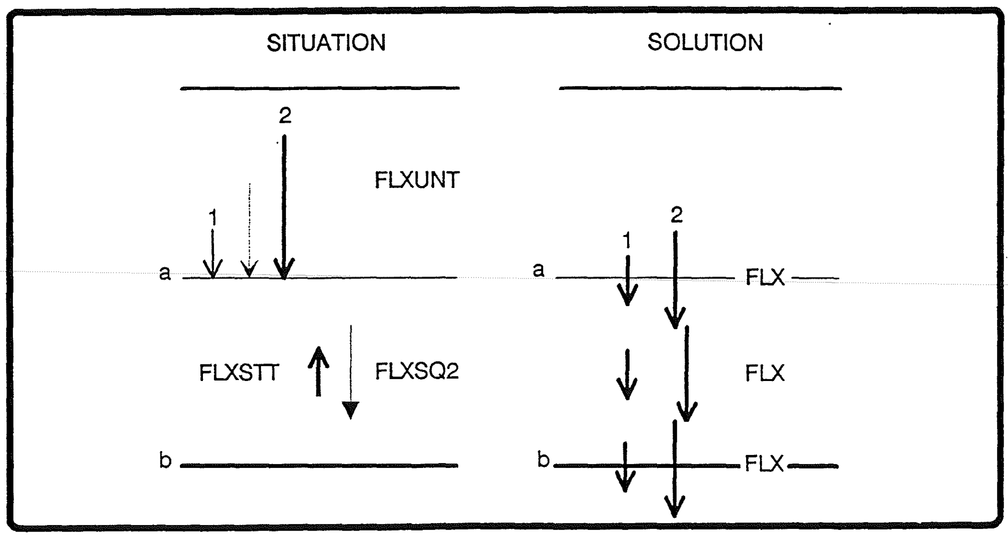
CLASS 2.1



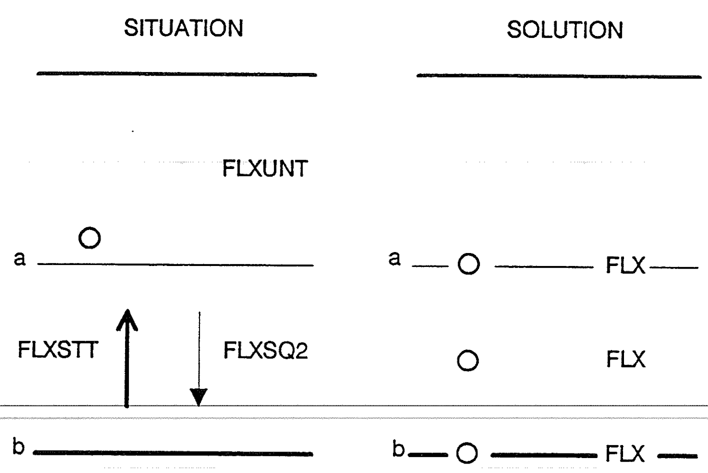
CLASS 2.2.1



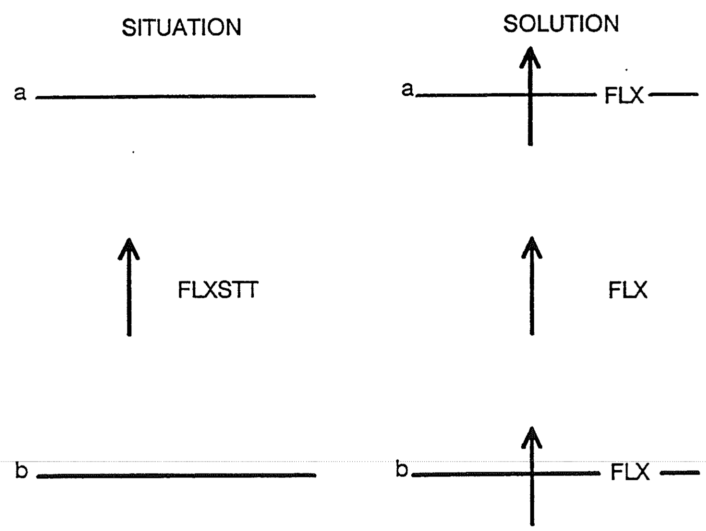
CLASS 2.2.2



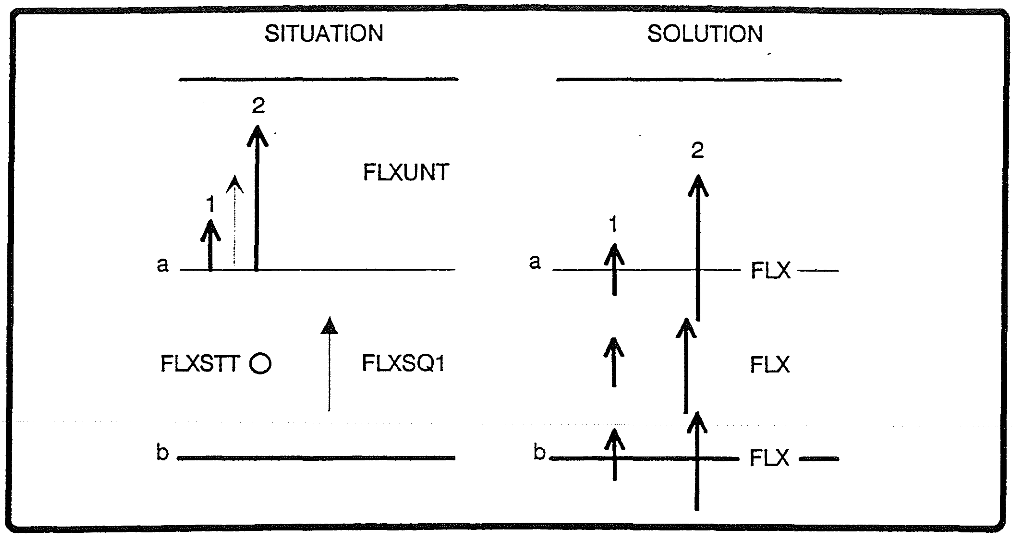
CLASS 2.2.3



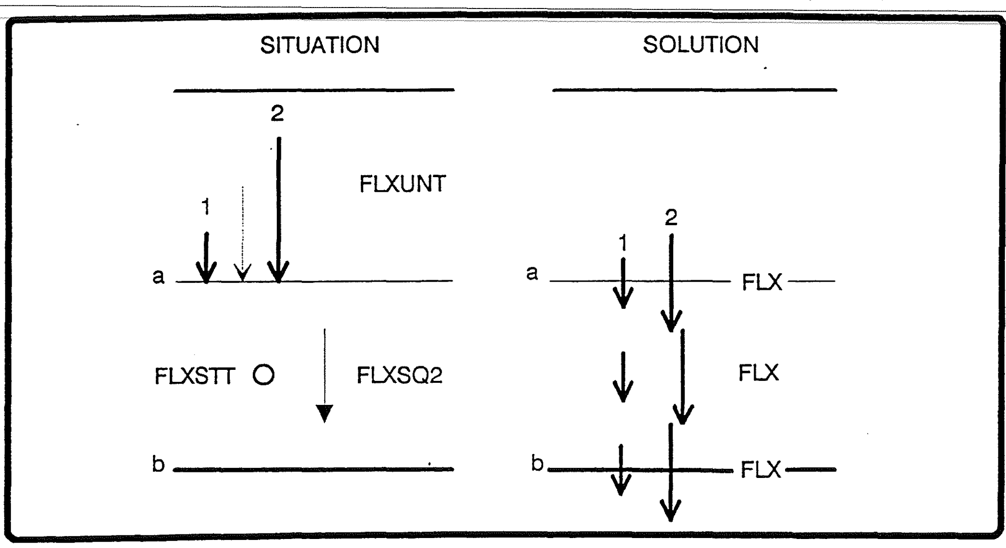
CLASS 2.2.4



CLASS 3.1



CLASS 3.2



CLASS 3.3