Technical Bulletin
2001 No. 7

Upland: A Simulation Model for Water Balance in Upland Soils

M.V.R. Murty and M. Kondo

IRRI
INTERNATIONAL RICE RESEARCH INSTITUTE

UPLAND: a simulation model for water balance in upland soils

M.V.R. Murty and M. Kondo

Even though the upland rice area is 12% of the total world rice area, it contributes only 4% to the total rice production (Pandey 1996). Average grain yields are about 1 t ha^{-1} (IRRI 1997). Rice cultivation in rainfed uplands usually takes place in unbunded and bunded parcels. Puddling is not practiced and the only source of water is rain. Thus, in these lands, water is generally the main constraint. Under such situations, it is important to understand water balance processes. To understand water capture by rice roots, we must quantify, as accurately as possible, the evaporation from the bare soil between rows. In this context, we developed a simulation model for water balance in upland soils based on earlier models developed mainly for lowland rice, such as that of Wopereis et al (1996).

UPLAND, programmed under the Fortran Simulation Environment (FSE, version 2.1), was developed by van Kraalingen (1995). The FSE system has a main program with a driver and a weather utility system along with a general utility library. The FSE driver handles the simulation loop by initializing, calculating the rate, integrating, and terminating all the models via ITASK = (1or2or3or4) calls through MODELS. In addition, the driver runs a loop across each rerun, handles input and output files, and reads weather data using TTUTIL and WEATHER libraries (van Kraalingen 1995) (Fig. 1).

UPLAND, a soil water balance and nitrate leaching model, is one-dimensional. It can be used for freely drained soil as well as soils with impeded drainage. The nitrate leaching part of the model is not

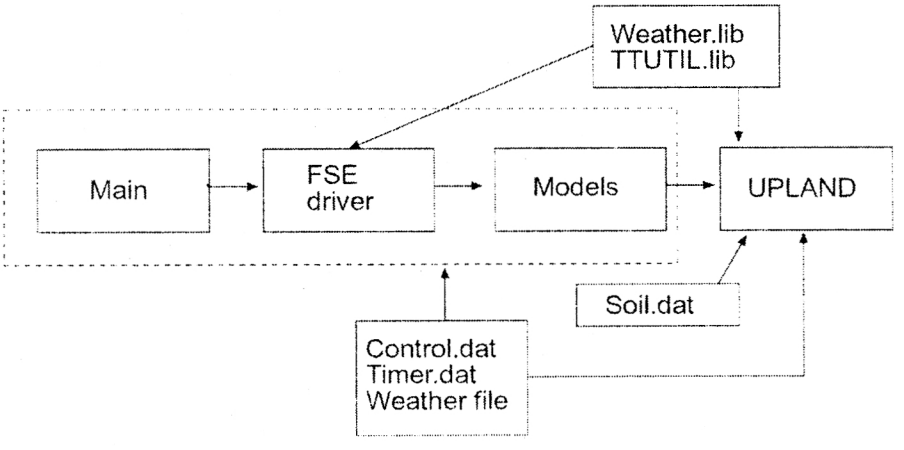

Fig. 1. Flow diagram of FSE system (adapted from van Kraalingen 1995).

yet exhaustive for inherent soil nitrogen transformations. We have not attempted to include the transformation processes; future attempts will be made to do this. The current version indicates the importance of nitrate adsorption-desorption by the solid phase in tropical uplands; later versions will attempt to simulate nitrogen transformations. In temperate soils, little or no nitrate is adsorbed on the soil surface. In the tropical soils of Asia, however, particularly highly weathered acidic soils can have a significant amount of nitrate adsorption by the solid phase (George et al 1996). Our laboratory measurements also confirm this view.

Figure 2 shows a general water balance approach for soils.

$$\Delta W = I+R+UF-DF-E-RF$$

where

ΔW = change in water storage
I = irrigation
R = rainfall
UF = upward flux
DF = downward flux
E = evaporation
RF = runoff

All the units are in mm m^{-1}. The present model follows the same principle. The downward flux from the last soil compartment is considered as deep drainage.

A complete listing of the source code as well as input files is provided in the appendices.

Input requirements

The following inputs are needed for UPLAND

NL	Number of layers	(–)
TKL	Thickness of layers	m
WCLI	Initial volumetric moisture content (layerwise)	$m^3 m^{-3}$
RIRRIT	Irrigation table	mm d^{-1}
SWITIR	Irrigation switch	(–)
SWITFD	Free drainage/impeded drainage switch	(–)
KST	Saturated hydraulic conductivity	cm d^{-1}
VGA	van Genuchten alpha parameter	(–)
VGL	van Genuchten lambda parameter	(–)
VGN	van Genuchten n parameter	(–)
VGR	van Genuchten residual water	(–)
KST	Saturated hydraulic conductivity	cm d^{-1}

For nitrate leaching (optional)

NCONAP	Nitrate applied	kg ha^{-1}
NCONTI	Initial nitrate concentration	ml cm^{-3}
BDSOIL	Bulk density of the soil (layerwise)	g cc^{-1}

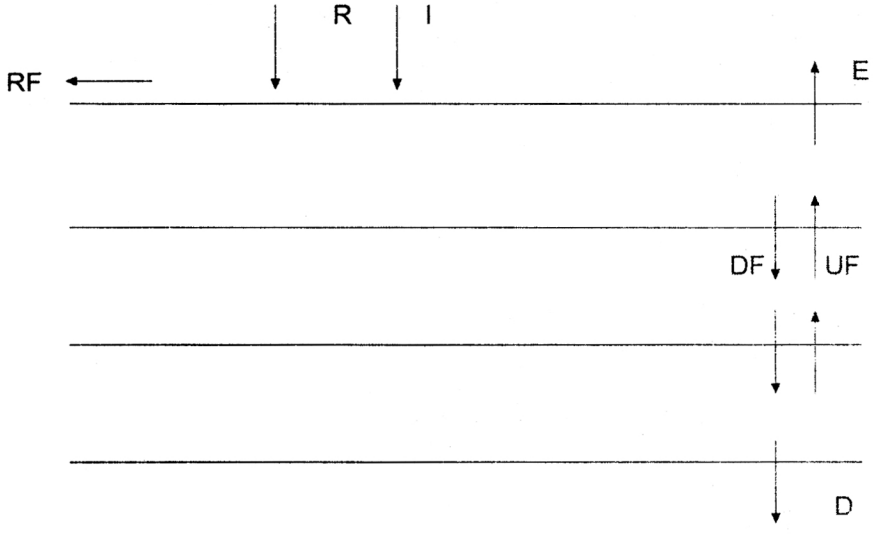

Fig. 2. Vertical profile of water balance components.

PARAA	Parameter A for the nitrate equilibrium equation	(–)
PARAB	Parameter B for the nitrate equilibrium equation	(–)
PARAC	Parameter C for the nitrate equilibrium equation	(–)
PARAD	Parameter D for the nitrate equilibrium equation	(–)
SET	Breakaway point for choosing the right line	(–)

For the nitrate equilibrium equation

NCONT	Total nitrate concentration	µmol cm^{-3}
NCONW	Nitrate concentration in water	µmol cm^{-3}
NCONS	Nitrate concentration in solids	µmol cm^{-3}
NCONCH	Change in nitrate concentration	µmol cm^{-3}

Background

Soil water balance

Water balance simulation has been attempted in the past by Jones and Kinery (1986), Wopereis et al (1996), and Penning de Vries et al (1989), among many others. Upland rice culture has been defined as rice grown in rainfed bunded or unbunded fields with naturally well drained soils and no surface water accumulation (IRRI 1984). Based on this, some simulation models were built taking into account that rainfall does not stay in the soil but drains to field capacity, usually one day at a time, as in SAHEL (Wopereis et al 1996). In some water balance models generally designed for lowlands, the contribution of capillary rise was made from the water table, as was done in PADDY (Wopereis et al 1996). However, upland soils do not usually have a shallow water table. Often, the groundwater table might be very deep, making the contribution from the water table to the surface layer negligible. Under such situations, following a rainfall or irrigation event, evaporation losses from surface soils will become a dominant factor in driving subsurface moisture up following steep potential gradients. This will be a basis for redistributing moisture within the profile. CERES-type models have a similar approach based on Darcy's principle (Gabrielle et al 1995). A similar approach has been attempted in the present model, which looks at the redistribution of moisture within the profile using Darcy's principle.

To some extent, UPLAND retains the flow structure developed in PADDY. Basically, UPLAND was developed using PADDY as an example. Some of the subroutines used in PADDY were retained with minor adaptations.

Even though uplands with sandy to loamy-type soils are generally freely drained, upland soils with high clay content are not freely drained. In such soils, a major source of rainwater loss is through surface runoff. A drainage switch, SWITFD, with 0 for impeded drainage and 1 for free drainage, was retained from PADDY. The simulation of water balance for free drainage was similar to that of PADDY under ponded water conditions except for upward flux.

At the beginning, the model calculates the downward and upward fluxes using Darcy's principle. But upward fluxes are limited to a maximum such that the layer above will not have more than field capacity. Similarly, downward fluxes are limited to a maximum from any layer such that it will not drain below field capacity if that layer has a water content higher than the field capacity.

```
I = 1
  DO WHILE (I.LE.(NL-1))
       DMS(I) = (MS(I+1)-MS(I))
       CONDCA(I) = (SQRT(KMS(I)*KMS(I+1)))
     DFLUX(I) = (CONDCA(I)*((DMS(I)+TKLM(I))/TKLM(I)))*10.

 IF ((DFLUX(I).GE.0.).AND.(MS(I).LE.300.)) THEN
 DFLUX(I) = MIN(10.*KSAT(I),MIN((WL(I)-WLFC(I)),&
 (WLST(I+1)-WL(I+1))/DELT))
 ELSE IF ((DFLUX(I).GE.0.).AND.(MS(I).GT.300.)) THEN
 DFLUX(I) = MIN(DFLUX(I),(WLST(I+1)-WL(I+1))/DELT)
ELSE
 DFLUX(I) = 0.
END IF
 UDFLUX(I) = (CONDCA(I)*((DMS(I)+TKLM(I))/TKLM(I)))*10.
```

```
 IF (UDFLUX(I).LT.0.0) THEN
   UDFLUX(I) = ABS(UDFLUX(I))
  ELSE
   UDFLUX(I) = 0.
END IF

 IF (MS(I).GE.300.) THEN
  UDFLUX(I) = MIN(ABS(UDFLUX(I)),MAX(0.0,(WLFC(I)-WL(I))/DELT))
ELSE
 UDFLUX(I) = 0.
END IF
 I = I+1
END DO
 IF (I.EQ.NL) THEN
  DFLUX(I) = MIN(10.*KSAT(I),MAX(0.0,(WL(I)-WLFC(I))/DELT))
 UDFLUX(I) = 0.0
END IF
```

As with PADDY, in UPLAND the flow of the algorithm starts with ponding water. It checks whether there is standing water (WL0). When the field has standing water, the model calculates downward fluxes using the same approach as PADDY. In this case, it assumes that there is no upward flux, as all layers will be at least up to field capacity.

When the standing water partly contributes to the evaporative demand, a similar approach is used as in PADDY. When there is no standing water, which is usual for most of the growing season for rainfed upland rice, downward fluxes (DFLUX) are used. Similarly, upward fluxes (UDFLUX) are used with the DNFL1 subroutine. DFLUX and UDFLUX are calculated at the beginning.

Actual evaporation

When there is standing water to either completely or partially fulfill the evaporative demand, soil evaporation will be at a potential rate. This potential evaporation rate is calculated by the subroutine ETPOT. In situations when there is no ponded water on the surface, a common occurrence in the uplands, a reduction factor is calculated. This reduction factor is based on the relationship between the ratio of potential to actual evaporation and the relative water content of the surface layer. Finally, to obtain the actual evaporation rate, potential evaporation with the soil background supplied by ETPOT is multiplied by the reduction factor.

```
*Calculate evaporation rate from soil surface (mm/d)

ZDD = (WCL(1)-WCAD(1))/(WCST(1)-WCAD(1))
RFDD = LINT(RFDDTB,ILRFDD,ZDD)
EVSW = EVSCS*RFDD
EVSWS = EVSW
```

Changes in soil water

After calculating the water flux rates, calculate changes in water contents (WLCH) for all the soil layers.

```
   I = 1
DO WHILE (I.LE.NL)
   IF (I.EQ.1) THEN
      WLCH(I) = WLFL(I)-WLFL(I+1)-TRWL(I)-EVSWS+UDFLUX(I)
   ELSE
     WLCH(I) = WLFL(I)-WLFL(I+1)-TRWL(I)+UDFLUX(I)-UDFLUX(I-1)
   END IF
    WCUMCH = WCUMCH+WLCH(I)
   I = I+1
END DO
```

Integration over the day is carried over for the time step of 1 day for all state variables as in PADDY.

Potential evaporation

Potential evaporation from free water surfaces and bare soil as well as evapotranspiration were calculated by Penman (1948). The current model uses these main concepts built into a subroutine, ETPOT, developed by Bouman and incorporated in ORYZA-_W (Wopereis et al 1996). Full documentation is available in Wopereis et al (1996). Potential evaporation means that there is no limitation to the supply of water. This is mainly driven by environmental parameters such as radiation load and wind forces. Radiation load is calculated by estimating the energy balance of the surface and this differs for open water, soil background, and crop surface.It also considers latent heat of evaporation of water. The drying power in Penman's equation takes into account vapor pressure deficits in the atmosphere along with a wind speed function. A combination of these two terms is the potential evaporation of that surface.

Nitrate leaching

Nitrogen is one of the major nutrients that limit crop growth. When crop growth depends entirely on rainfall, it is often difficult to assess whether N or water is the limiting factor in rainfed rice cultivation. It becomes essential to resort to a simulation of these factors in soils to have an idea of their availability. Nutrient losses, particularly nitrate loss through leaching below the root zone, are of major concern. In the upland soils of tropical Asia, the amount of nitrate leaching depends on the quantity and intensity of rainfall and on the nitrate adsorption behavior by the solid phase of the soil. In arid soils, this was not considered highly important (van Keulen and Seligman 1987). If the soils are able to adsorb a greater quantity of nitrate applied, it will be available to the roots for a longer time. Experiments conducted on upland soils of Asia have shown that nitrate adsorption-desorption characteristics vary greatly from one area to another.

Nitrate adsorption by the solid phase, and thereby the equilibrium between the solid and solution phase, can be simulated through adsorption isotherms represented by the equation

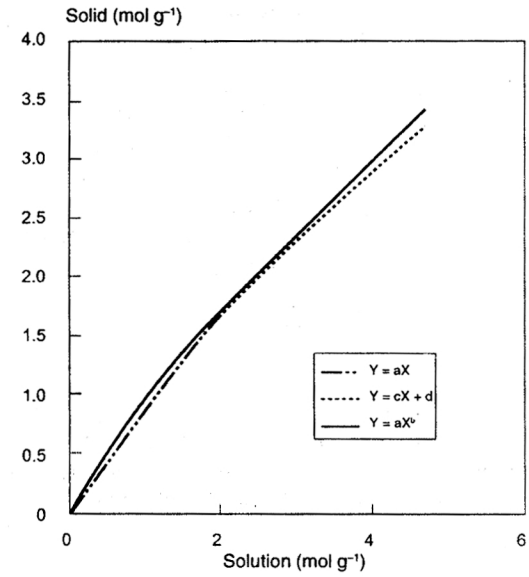

Fig. 3. Nitrate equilibrium between soil solid and liquid phases represented by a set of linear equations.

$$Y = aX^b$$

where a and b are constants.

To simplify, the curve was represented by two sets of linear equations (Fig. 3):

$$Y = aX$$

and

$$Y = cX + d$$

The first equation describes the initial part of the equilibrium and the second set describes the latter part of the equation. Saito (1990) used a similar linear approach for nitrate adsorption-desorption equilibrium.

Simulation

The model first calls all the input parameters, then it reads the nitrate application, if any, from the input file and the application is added to the top layer as nitrate concentration (μmol cm^{-3}).

```
      GNAPP = LINT(NAPPTB,INAPP,DOY)
         NCONAP = GNAPP/((TKL(1)/10.)*1.4)
       I = 1
      DO WHILE (I.LE.NL)
        IF (I.EQ.1) THEN
          NCONT(I) = NCONTI(I)+NCONAP
        ELSE
          NCONT(I) = NCONTI(I)
        END IF
       I = I+1
      END DO
```

Subsequently, nitrate distribution between soil solids and solution is simulated by using the linear equations for equilibrium. For this, the parameters for different layers (PARAA, #PARAB, #PARAC, #PARAD, and SET) are used. First, the load (NLODW) in the soil water is estimated and then nitrate concentration in water (NCONW) and nitrate concentration in solids (NCONS) is calculated:

```
NLODW(I) = (NCONT(I)-(PARAB(I)*BDSOIL(I)))/(WCL(I)+(PARAA(I)
     &                  *BDSOIL(I)))
          IF (NLODW(I).LT.SET(I)) THEN
            NLODW(I) = NLODW(I)
         ELSE
         NLODW(I) = (NCONT(I)-(PARAD(I)*BDSOIL(I)))/(WCL(I)+(PARAC(I)
     &                  *BDSOIL(I)))

         END IF
          NCONW(I) = NLODW(I)*WCL(I)
          NCONS(I) = NCONT(I)-NCONW(I)
          IF (NCONS(I).LE.TINY) THEN
         NCONS(I) = 0.0
         ELSE
          NCONS(I) = NCONS(I)
         END IF

           NLODS(I) = NCONS(I)/BDSOIL(I)
         I =I+1
         END DO
```

Then the total nitrate in the soil solution and in the soil solids is calculated:

```
I = 1
 DO WHILE (I.LE.NL)
    NTOTW(I) = (NCONW(I)*(TKL(I)/10.)*1.4)
    NTOTS(I) = (NCONS(I)*(TKL(I)/10.)*1.4)
    NTOT(I) = NTOTW(I)+NTOTS(I)
    I = I+1
    END DO
```

Rate calculations. In the rate calculation sections, the amount of nitrate coming into a compartment (NIN) as well as the amount of nitrate going out of that compartment (NOUT) is calculated. These depend on water fluxes going out of the compartment (WLFL) and upward fluxes going out of and into that compartment (UDFLUX).

```
*----Nitrate fluxes----------
      I = 1
       DO WHILE (I.LE.NL)
         IF (I.EQ.1) THEN
            NIN(I) = NLODW(I+1)*(UDFLUX(I)/10.)
            NOUT(I) = NLODW(I)*(WLFL(I+1)/10.)
         ELSE IF (I.EQ.NL) THEN
            NIN(I) = (NLODW(I-1)*(WLFL(I)/10.))
            NOUT(I) = (NLODW(I)*(WLFL(I+1)/10.))
         ELSE
            NIN(I) = (NLODW(I+1)*(UDFLUX(I)/10.))+(NLODW(I-1)*&
                      (WLFL(I)/10.))
            NOUT(I) = (NLODW(I)*(UDFLUX(I-1)/10.))+(NLODW(I)*&
                      (WLFL(I+1)/10.))
         END IF
         I = I+1
       END DO
```

Changes in state variables or in nitrate concentrations as a transient phase are then calculated as a difference between nitrate coming in and nitrate going out of a particular compartment. The nitrate application, if any, on that day is also read from the input file and added to the top compartment. Finally, the total amount of nitrate drained out of the profile is calculated as the nitrate going out of the last compartment (NDRAIN). This is also expressed in kg ha^{-1} (NDRAIKG).

```
GNAPP = LINT(NAPPTB,INAPP,DOY)
 NCONAP = GNAPP/((TKL(1)/10.)*1.4)

I = 1
DO WHILE (I.LE.NL)
 IF (I.EQ.1) THEN
   NCONCH(I) = (NIN(I)-NOUT(I))/(TKL(I)/10.)+NCONAP
 ELSE
   NCONCH(I) = (NIN(I)-NOUT(I))/(TKL(I)/10.)
```

```
            END IF
               I = I+1
            END DO
*----Estimation of NDRAIN in kg/ha
            I = NL
             DO WHILE  (I.EQ.NL)
                 NDRAIN(I) = NOUT(I)/(TKL(I)/10.)
                 NDRAKG(I) = NDRAIN(I)*(TKL(I)/10.)*1.4
            I = I+1
            END DO
```

Integration. Total nitrate concentration is integrated with the change in total nitrate concentration for each soil compartment. Then the nitrate load in the soil solution is obtained by establishing equilibrium between the solid and liquid phases. Afterwards, the new nitrate concentrations in the liquid and solid phases are calculated. The nitrate adsorbed in the solids and in the solution is calculated in kg layer^{-1}.

```
*----Nitrate equilibrium----------
          I = 1
           DO WHILE (I.LE.NL)
             NCONT(I) = INTGRL(NCONT(I),NCONCH(I),DELT)
             NCONT(I) = MAX(0.,NCONT(I))
             I = I+1
           END DO

          I = 1
           DO WHILE (I.LE.NL)

            NLODW(I) = (NCONT(I)-(PARAB(I)*BDSOIL(I)))/(WCL(I)+(PARAA(I)&
                          *BDSOIL(I)))
            IF (NLODW(I).LT.SET(I)) THEN
              NLODW(I) = NLODW(I)
            ELSE
            NLODW(I) = (NCONT(I)-(PARAD(I)*BDSOIL(I)))/(WCL(I)+(PARAC(I)&
                          *BDSOIL(I)))

              END IF
            NCONW(I) = NLODW(I)*WCL(I)
            NCONS(I) = NCONT(I)-NCONW(I)
            IF (NCONS(I).LE.TINY) THEN
            NCONS(I) = 0.0
            ELSE
            NCONS(I) = NCONS(I)
            END IF

            NLODS(I) = NCONS(I)/BDSOIL(I)
            I =I+1
            END DO
```

```
I = 1
DO WHILE (I.LE.NL)
NTOTW(I) = (NCONW(I)*(TKL(I)/10.)*1.4)
NTOTS(I) = (NCONS(I)*(TKL(I)/10.)*1.4)
NTOT(I) = NTOTW(I)+NTOTS(I)
I = I+1
END DO
```

Appendix 1 gives the source code of the model.

Model parameterization

Hydraulic conductivity and moisture characteristics

The measured water retention data from field samples were parameterized using van Genuchten equations (van Genuchten 1980):

$$S = (\theta - \theta_r) / (\theta_s - \theta_r) = (1 + |\alpha h|^n)^{-m}$$

and

$$k = k_s S^l [1 - (1 - S^{1/m})^m]^2$$

where $m = 1 - 1/n$, θ_r and θ_s are residual and saturated volumetric water contents, k_s is the saturated hydraulic conductivity, and S refers to the degree of saturation.

The parameters α, n, and l were optimized using the RECT program (van Genuchten et al 1991).

Bare soil evaporation

In some modeling approaches, actual evaporation from the bare soil is calculated by assuming that cumulative evaporation is proportional to the square root of time (Wopereis et al 1996, Penning de Vries et al 1989). In general, this approach works well for obtaining gross approximations in a simplified way. However, when tested with actual field bare soil evaporation data, it underestimated actual evaporation; therefore, we used a different approach. Partitioning of water loss into bare soil evaporation and water extraction by roots in the surface layer is very important in rainfed systems. To improve understanding of the dynamics of water capture by roots with increasing soil water potential, we need a reliable estimation of bare soil evaporation dynamics from the exposed soil surface between rows of the crop canopy. This will make it easier to estimate water capture by roots in field experiments.

Evaporation from the bare soil is driven by the radiation load and the drying power of the air. Potential evaporation rates can be estimated using Penman's approach. Evaporation from the soil surface can occur at potential rates only when soil water is not a limiting factor or soil water content is at near saturation. When the soil water content falls below this level, the evaporation rate will be governed mainly by the decreasing soil water content in the surface layer.

Experiments with undisturbed core samples collected from the field (12 cm height and 10 cm diameter) from three sites in the Philippines (Matalom, Siniloan, and IRRI upland farm) were thoroughly saturated and kept in IRRI's phytotron facility under constant conditions of evaporativity (30 °C, 1,200 µEinsteins radiation, and relative humidity 70%). Water loss through surface evaporation was monitored regularly by carefully weighing the samples twice a day till evaporation loss became negligible and the soil became sufficiently dry. Finally, cores were put in an oven at 105 °C for 48 h, after which the final weights were determined. Throughout the experiment, a similar container with normal water was also placed in the phytotron and water loss was monitored. The water loss from the open water surface was considered as the potential evaporation loss at that evaporative demand of the atmosphere. A relationship between the ratio of actual evaporation (AE) and potential evaporation (PE) and relative water content was developed (Fig. 4). Relative water content (RWC) is defined as follows:

$$RWC = (\theta - \theta_{AD})/(\theta_{ST} - \theta_{AD})$$

where θ is the actual soil volumetric moisture content ($cm^3\ cm^{-3}$), θ_{AD} is the volumetric moisture content at air dryness ($cm^3\ cm^{-3}$), and θ_{ST} is the volumetric moisture content at saturation ($cm^3\ cm^{-3}$).

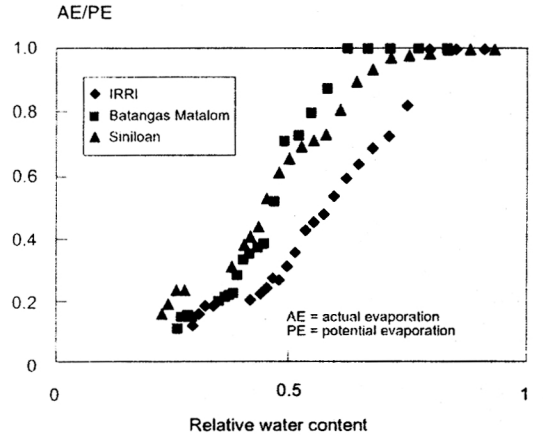

Fig. 4. Reduction of actual bare soil evaporation based on the relationship of the ratio between actual to potential evaporation and relative water content.

The amount of water loss through evaporation from the soil surface can be expressed as a fraction of the potential soil evaporation:

$$Ea = Ep * RF$$

where Ea = actual evaporation (mm d^{-1}), Ep = potential evaporation rate (mm d^{-1}), and RF = reduction factor for the influence of soil moisture content on evaporation rate from the soil surface (dimensionless factor).

Model validation

The current version of the model was tested using data from a field experiment conducted on the IRRI upland farm in the 1997 dry season. Measurements of volumetric water contents in a bare plot during a drying cycle were used. Saturated hydraulic conductivity and unsaturated hydraulic conductivity were measured on undisturbed cores collected from the field at various depths. Saturated hydraulic conductivity was measured by the constant head method (Wopereis et al 1993), while unsaturated hydraulic conductivity was measured by the wind method (Wopereis et al 1993). A soil moisture characteristic curve was constructed using undisturbed cores from the field and a pressure plate apparatus. Later, the data were parameterized using the van Genuchten equation (van Genuchten 1980). All these were used as input parameters along with actual weather data. The model was run for the period of experimentation. Results (Fig. 5) show that the model could simulate actual soil evaporation satisfactorily.

Input and output files

All the input data required for the model are given in one file. Appendix 2 contains a sample file. Appendix 3 gives a sample file that provides the appropriate selection and path of weather data and the start and finish times of the simulations.

A control file is required for the input and output control of the simulation runs. Appendix 4 shows a sample file. In this file, FILEI1 to FILEI4 are specified; the names of input data files, for example, soil.dat, need to be specified. Similarly, FILEIT is for the timer file name, FILEIR is for the specification of the reruns file, FILEON is for the specification of the name of the output file in which results will be stored, and FILEOL is for the specification of the log file.

The FSE system provides the facility of having more simulation runs with a single execution command. In the reruns file, the input variables and/or parameters or start and finish times or a combination of any of them with which we need to rerun the simulations can be specified. If we need several reruns, then we can specify each set one below the other. However, the order and number of variables in all the specified reruns should be the same. Appendix 5 shows an example of a reruns file.

An output file created by the FSE system, whose name can be defined in the control.dat file, will contain values of selected variables at desired time steps as indicated by the PRDEL option in the timer file.

Sample soil, timer, and control files are presented as Appendices 2–4.

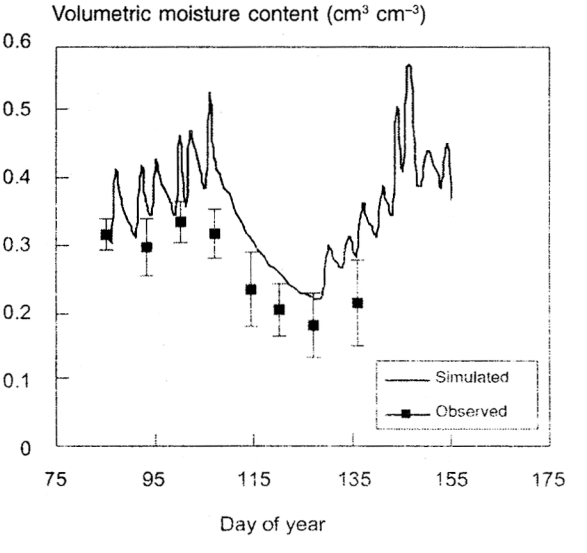

Fig. 5. Comparison of simulated and actual volumetric water contents from a bare soil.

Listing

Some of the variables in UPLAND, which are similar to those in PADDY, are not listed here.

UDFLUX	Upward flux	(mm)
DFLUX	Downward flux	(mm)
NL	Number of layers	(–)
TKL	Thickness of layers	m
WCLI	Initial volumetric moisture content (layerwise)	$m^3\ m^{-3}$
RIRRIT	Irrigation table	$mm\ d^{-1}$
SWITIR	Irrigation switch	(–)
SWITFD	Free drainage/impeded drainage switch	(–)
KST	Saturated hydraulic conductivity	$cm\ d^{-1}$
VGA	van Genuchten alpha parameter	(–)
VGL	van Genuchten lambda parameter	(–)
VGN	van Genuchten n parameter	(–)
VGR	van Genuchten residual water	(–)
KST	Saturated hydraulic conductivity	$cm\ d^{-1}$

For nitrate leaching (optional)

NCONAP	Nitrate applied	$kg\ ha^{-1}$
NCONTI	Initial nitrate concentration	$ml\ cm^{-3}$
BDSOIL	Bulk density of the soil (layerwise)	$g\ cc^{-1}$
PARAA	Parameter A for the nitrate equilibrium equation	(–)
PARAB	Parameter B for the nitrate equilibrium equation	(–)
PARAC	Parameter C for the nitrate equilibrium equation	(–)
PARAD	Parameter D for the nitrate equilibrium equation	(–)
SET	Breakaway point for choosing the right line	(–)

For the nitrate equilibrium equation

NCONT	Total nitrate concentration	$\mu mol\ cm^{-3}$
NCONW	Nitrate concentration in water	$\mu mol\ cm^{-3}$
NCONS	Nitrate concentration in solids	$\mu mol\ cm^{-3}$
NCONCH	Change in nitrate concentration	$\mu mol\ cm^{-3}$

References

Gabrielle B, Menasseri S, Houot S. 1995. Analysis and field evaluation of the CERES models water balance component. Soil Sci. Soc. Am. J. 59:1403-1412.

George T, Kirk GJD, Almendras A, Sonon L, Serohijos R, Chaitup W, Thirathon A, Khatib W, Abdullah S, Warman A, Naim T, Magbanua R. 1996. Strategic research on soil fertility management for upland rice areas in South and Southeast Asia. In: Piggin C, Courtois B, Schmit V, editors. Upland rice research in partnership. IRRI Discuss. Pap. Ser. No. 16. Manila (Philippines): IRRI. p 231-238.

IRRI (International Rice Research Institute) 1984. Terminology for rice-growing environments. Manila (Philippines): IRRI. 35 p.

IRRI (International Rice Research Institute) 1997. Rice almanac. Manila (Philippines): IRRI. 181 p.

Jones CA, Kinery JR. 1986. CERES-Maize, a simulation model of maize growth and development. College Station Tex. (USA): Texas A&M University Press.

Pandey S. 1996. Socioeconomic context and priorities for strategic research on Asian upland rice ecosystems. In: Piggin C, Courtois B, Schmit V, editors. 1996. Upland rice research in partnership. IRRI Discuss. Pap. Ser. No. 16. Manila (Philippines): IRRI. p 103-124.

Penman HL. 1948. Natural evaporation from open water, bare soil and grass. Proceedings of the Royal Society of London Series A 193. p 120-146.

Penning de Vries FWT, Jansen DM, ten Berge HFM, Bokema A. 1989. Simulation of ecophysiological processes of growth in several annual crops. Simulatiom monographs. PUDOC, Wageningen, The Netherlands. 271 p.

Ritchie JR. 1972. Model for predicting evaporation from a row crop with incomplete cover. Water Resources Res. 8:1204-1213.

Saito M. 1990. Mineralization of organic nitrogen and its movement in an andisol under field conditions-model simulation and its verification. Bull. Tohoku Natl. Agric. Exp. Stn. 82:63-76.

van Genuchten MTh. 1980. A closed form equation for predicting the hydraulic properties of unsaturated soils. Soil Sci. Soc. Am. J. 44:892-898.

van Genuchten M Th, Leij FJ, Yates SR. 1991. The RECT code for quantifying the hydraulic functions of unsaturated soils. US-EPA, Ada, Oklahoma, USA. 85 p.

van Keulen H, Seligman SG. 1987. Simulation of water use, nitrogen nutrition and growth of a spring wheat crop. PUDOC, Wageningen, The Netherlands. 310 p.

van Kraalingen DWG. 1995. The FSE system for crop simulation, version 2.1. Quantitative Approaches in Systems Analysis No. 1. 59 p.

Wopereis MCS, Bouman BAM, Tuong TP, ten Berge HFM, Kropff MJ. 1996. Oryza_W: rice growth model for irrigated and rainfed environments. SARP Research Proceedings. 159 p.

Wopereis MCS, Kropff MJ, Bouma J, Van Wijk ALM, Woodhead T. 1993. Soil physical properties: measurement and use in rice-based cropping systems. Los Baños (Philippines): International Rice Research Institute. 110 p.

Appendix 1. Source code of UPLAND (v1.0).

```
*
*                              UPLAND (v1.0)
*
* A model for water balance and nitrate leaching for upland rice soils
*
*
* programmed by
* M.V.R. Murty and M. Kondo
*   February 2000
*
* Version: FSE-2.1
* Date: October 1994
*
* Partly based on PADDY from
* ORYZA_W: Rice growth model for irrigated and rainfed
*          environments
* M.C.S. Wopereis, B.A.M. Bouman, T.P. Tuong, H.F.M. ten Berge, and M.J. Kropf
*
*
* International Rice Research Institute (IRRI), MCPO Box 3127,
* Makati City 1271, Philippines
*
* Department of Theoretical Production Ecology (TPE-WAU),
* Wageningen Agricultural University, P.O. Box 430, 6700 AK Wageningen,
* The Netherlands
*
* Research Institute for Agrobiology and Soil Fertility (AB-DLO),
* Agricultural Research Department, P.O. Box 14, 6700 AA Wageningen,
* The Netherlands
*
* This model is based on the following models:
*
**************************************************************************
      PROGRAM MAIN
      CALL FSE
      END

*-------------------------------------*
* SUBROUTINE MODELS
* Authors: Daniel van Kraalingen
* Date   : 5 July 1995
* Purpose: This subroutine is the interface routine between the FSE-
*          driver and the simulation models. This routine is called
*          by the FSE-driver at each new task at each time step. It
*          can be used to specify calls to the different models
*           that have to be simulated.
*
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)
* name   type meaning (unit)                                      class
* -----  ---- --------------                                      ------
* ITASK   I4  Task that subroutine should perform (-)               I
* IUNITD  I4  Unit that can be used for input files (-)             I
* IUNITO  I4  Unit number of output file (-)                        I
```

```
* IUNITL  I4  Unit number for log file messages (-)                  I
* FILEIT  C*  Name of timer input file (-)                           I
* FILEI1  C*  Name of input file no. 1 (-)                           I
* FILEI2  C*  Name of input file no. 2 (-)                           I
* FILEI3  C*  Name of input file no. 3 (-)                           I
* FILEI4  C*  Name of input file no. 4 (-)                           I
* FILEI5  C*  Name of input file no. 5 (-)                           I
* OUTPUT  L4  Flag to indicate if output should be done (-)          I
* TERMNL  L4  Flag to indicate if simulation is to stop (-)         I/O
* DOY     R4  Day number (January 1 = 1) (-)                        I
* IDOY    I4  Day number within year of simulation (INTEGER) (d)    I
* YEAR    R4  Year of simulation (REAL) (y)                         I
* IYEAR   I4  Year of simulation (INTEGER) (y)                      I
* TIME    R4  Time of simulation (d)                                I
* STTIME  R4  Start time of simulation (=day number) (d)            I
* FINTIM  R4  Finish time of simulation (=day number) (d)           I
* DELT    R4  Time step of integration (d)                          I
* LAT     R4  Latitude of site (dec.degr.)                          I
* LONG    R4  Longitude of site (dec.degr.)                         I
* ELEV    R4  Elevation of site (m)                                 I
* WSTAT   C*  Status code from weather system (-)                   I
* WTRTER  L4  Flag whether weather can be used by model (-)         O
* RDD     R4  Daily shortwave radiation (J m-2 d-1)                 I
* TMMN    R4  Daily minimum temperature (degrees C)                 I
* TMMX    R4  Daily maximum temperature (degrees C)                 I
* VP      R4  Early morning vapour pressure (kPa)                   I
* WN      R4  Daily average windspeed (m s-1)                       I
* RAIN    R4  Daily amount of rainfall (mm d-1)                     I
*
* Fatal error checks: none
* Warnings          : none
* Subprograms called: models as specified by the user
* File usage        : none
*-------------------------------------*

      SUBROUTINE MODELS(ITASK,IUNITD,IUNITO,IUNITL,FILEIT,FILEI1,FILEI2,
     &                  FILEI3,FILEI4,FILEI5,OUTPUT,TERMNL,DOY,IDOY,
     &                  YEAR,IYEAR,TIME,STTIME,FINTIM,DELT,LAT,LONG,
     &                  ELEV,WSTAT,WTRTER,RDD,TMMN,TMMX,VP,WN,RAIN)
      IMPLICIT REAL(A-Z)

*     Formal parameters
      INTEGER ITASK,IUNITD,IUNITO,IUNITL,IDOY,IYEAR
      CHARACTER FILEIT*(*),FILEI1*(*),FILEI2*(*),FILEI3*(*)
      CHARACTER FILEI4*(*),FILEI5*(*)
      LOGICAL OUTPUT,TERMNL,WTRTER
      CHARACTER WSTAT*6
      CHARACTER WUSED*6

*     Local variables
      INTEGER SWIWLP
*      INTEGER SWITPF
      INTEGER IDOYH,ISTAT2
      INTEGER IWVAR

*---Standard local declarations
      INTEGER INL,I
      PARAMETER (INL=10)

*---Declarations for water-limited production
```

```
      REAL WCWP(INL),WCFC(INL),WCST(INL)
      REAL TRWL(INL)

      INTEGER NL
      SAVE

*     code for the use of RDD, TMMN, TMMX, VP, WN, RAIN  (in that order)
*     a letter 'U' indicates that the variable is used in calculations
      DATA WUSED/'UUUUUU'/

*     Check weather data availability
      IF (ITASK.EQ.1.OR.ITASK.EQ.2.OR.ITASK.EQ.4) THEN
         DO IWVAR = 1,6
*           is there an error in the IWVAR-th weather variable ?
            IF (WUSED(IWVAR:IWVAR).EQ.'U'.AND.WSTAT(IWVAR:IWVAR).EQ.'4')
     &           THEN
               WTRTER = .TRUE.
               TERMNL = .TRUE.
               RETURN
            END IF
10       END DO
      END IF

*-------------------------------------
*        Initialization section
*-------------------------------------
      IF (ITASK.EQ.1) THEN

*---Read values from TIMER file
         CALL RDINIT(IUNITD,IUNITL,FILEIT)
         CALL RDSINT('SWIWLP',SWIWLP)
         CALL RDSREA('MULTIP',MULTIP)
         CLOSE (IUNITD)


         RAINN = 0.

      END IF

*--To run soil water balance; to get rain of next day
      IDOYH = MIN(IDOY+1,365)
      CALL WEATHR(IDOYH,ISTAT2,RDDN,TMMNN,TMMXN,VPN,WNN,RAINN)


           CALL UPLAND(ITASK,IUNITD,IUNITO,IUNITL,FILEI2,OUTPUT,TERMNL,
     &                WSTAT,WTRTER,DOY,DELT,TIME,RAIN,EVSCS,EVSCWL,
     &                  INL,NL,WCWP,WCFC,WCST,WCL,LAT,RDD,TMMN,
     &                   TMMX,VP,WN,FILEIT,
     &                  WL0)


      RETURN
      END
```

```
*-------------------------------------------*
*    SUBROUTINE UPLAND
*   Authors: M.V.R. Murty and M. Kondo
*
*    Documented in SARP Research Proceedings (1994)
*    ORYZA_W: rice growth model for fully irrigated and
*    water-limited conditions
*    M.C.S. Wopereis, B.A.M. Bouman, T.P. Tuong, H.F.M ten Berge, and M.J. Kropff
*
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)
* name   type meaning (unit)                                          class
* -----  ---- ----------------                                       ------
* ITASK   I4  Task that subroutine should perform (-)                  C
* IUNITD  I4  Unit that can be used for input files (-)                C/IN
* IUNITO  I4  Unit number of output file (-)                           C/IN
* IUNITL  I4  Unit number for log file messages (-)                    C/IN
* FILEI2  C*  Name of input file no. 2 (-)                             C/IN
* OUTPUT  R4  Flag to indicate if output should be done (-)            C/I
* TERMNL  R4  Flag to indicate if simulation is to stop (-)            C/I/O
* WSTAT   C*  Status code from weather system (-)                      C
* WTRTER  L4  Flag whether weather can be used by model (-)            C
* DOY     R4  Day number (January 1 = 1) (-)                          I
* DELT    R4  Time step of integration (d)                            T
* TIME    R4  Time of simulation (d)                                  T
* ITIM    I4  Time of simulation (d)                                  T
* RAIN    R4  Daily amount of rainfall (mm d-1)                       I
* EVSC    R4  Potential soil evaporation rate (mm d-1)                I
* TRWL    R4  Array of actual transpiration rate/layer (mm d-1)       I
* INL     ??  Number of soil compartments (-)                         I
* NL      I4  Number of soil layers (-)                               I
* WCWP    R4  Array of water content at wilting point/layer (cm3
*             cm-3)                                                   O
* WCFC    R4  Array of water content field capacity/layer (cm3
*             cm-3)                                                   O
* WCST    R4  Array of water content saturation/layer (cm3 cm-3)      O
* WCL     R4  Array of actual water content/layer (cm3 cm-3)          O
* WL0     R4  Amount of ponded water (mm)                             O
*-------------------------------------------*

      SUBROUTINE UPLAND(ITASK,IUNITD,IUNITO,IUNITL,FILEI2,OUTPUT,TERMNL,
     &                  WSTAT,WTRTER,DOY,DELT,TIME,RAIN,EVSCS,EVSCWL,
     &                    INL,NL,WCWP,WCFC,WCST,WCL,LAT,RDD,TMMN,
     &                     TMMX,VP,WN,FILEIT,
     &                    WL0)


        IMPLICIT REAL(A-Z)

*-- Formal parameters
      INTEGER ITASK,IUNITD,IUNITO,IUNITL,INL,NL

      LOGICAL OUTPUT,TERMNL,WTRTER
      CHARACTER FILEI2*80,WSTAT*6
      REAL DOY,DELT,TIME,RAIN,EVSC,TKLT,WL0,EVSCS,EVSCWL
      REAL WCWP(INL),WCFC(INL),WCST(INL),WCL(INL)
      REAL LINT
       CHARACTER*(*) FILEIT
      REAL YEAR
      REAL LAT,RDD,TMMN,TMMX,VP,WN
      Local variables
```

```
*— Water balance parameters
      INTEGER SWITIR,SWITKH
      INTEGER SWITFD
      INTEGER SWITCR
      INTEGER MNL
      PARAMETER (MNL=10)
      REAL WCLI(MNL),TKL(MNL),WLFL(MNL+1),TKLM(MNL)
      REAL WLAD(MNL),WLFC(MNL),WLST(MNL),WL(MNL),MS(MNL)
      REAL KMS(MNL),DFLUX(MNL)

      REAL CONDCA(MNL),DMS(MNL)
      REAL CONAU(MNL),UDMS(MNL),UDFLUX(MNL+1)
      REAL WLCH(MNL)
      REAL WL0FCU
*— N content————————
      REAL NCONTI(MNL)
      REAL NOUT(MNL),NIN(MNL),NCONCH(MNL)
      REAL NLODS(MNL),NLODW(MNL),NCONW(MNL),NCONS(MNL),NCONT(MNL)
      REAL PARAA(MNL),PARAB(MNL),PARAC(MNL),PARAD(MNL),BDSOIL(MNL)
      REAL SET(MNL),NDRAIN(MNL),NDRAKG(MNL)
      REAL NTOT(MNL),NTOTW(MNL),NTOTS(MNL)
      REAL TRWL(MNL)
*     AFGEN functions
      INTEGER IMRFDD,ILRFDD
      PARAMETER (IMRFDD=40)
      DIMENSION RFDDTB(IMRFDD)
*—For LAI reading
      INTEGER IMLAI,ILLAI
      PARAMETER (IMLAI=40)
      DIMENSION LAITB(IMLAI)

      LOGICAL FREEDR
      INTEGER I,IRIRR
      INTEGER INAPP
      REAL RIRRIT(300)
      REAL NAPPTB(100)

      COMMON /NUCHT / VGN(10),VGA(10),VGR(10),VGL(10)
      COMMON /HYDCON/ KST(10),WCAD(10),WCSTRP(10)
      COMMON /POWER / PN(10)
      COMMON /SWIT  / SWITKH
      REAL KSAT(10)

      SAVE
*— Standard local variables
      INTEGER ITOLD

      TINY = 1.0E-5

      DATA ITOLD/4/

*     The task that the subroutine should do (ITASK) against the task
*     that was done during the previous call (ITOLD) is checked. Only
*     certain combinations are allowed. These are
*
*           New task:              Old task:
*           initialization         terminal
*           integration            rate calculation
*           rate calculation       initialization, integration
*           terminal               <any old task>
```

```
*      Note: there is one combination that is correct but will not cause
*      calculations to be done, i.e., if integration is required
*      immediately after initialization.

       IF (ITASK.EQ.2) THEN
          IF (WSTAT(2:2).EQ.'4'.OR.WSTAT(3:3).EQ.'4'.OR.WSTAT(4:4)
     &         .EQ.'4') THEN
             WTRTER = .TRUE.
             TERMNL = .TRUE.
             ITOLD = ITASK
             RETURN
          END IF
       END IF

       IF (ITASK.EQ.1) THEN
*-------------------------*
*-- Initialization
*-------------------------*

          FREEDR = .FALSE.
          IR = 0.
          GNAPP = 0.
          NLODAP = 0.

          CALL RDINIT(IUNITD,IUNITL,FILEIT)
          CALL RDSREA('ANGA',ANGA)
          CALL RDSREA('ANGB',ANGB)
          CLOSE (IUNITD)
*-- Read input from soil data file
          CALL RDINIT(IUNITD,IUNITL,FILEI2)
*-- Free draining/impeded drainage switch
          CALL RDSINT('SWITFD',SWITFD)
          IF (SWITFD.EQ.1) THEN
             FREEDR = .TRUE.
          ELSE IF (SWITFD.EQ.0) THEN
             FREEDR = .FALSE.
          END IF
          IF ((SWITFD.NE.0).AND.(SWITFD.NE.1))
     &         STOP 'PLEASE CHECK VALUE SWITFD IN SOIL DATA FILE'


*--- Irrigation switch
          CALL RDSINT('SWITIR',SWITIR)
          IF (SWITIR.NE.0.AND.SWITIR.NE.1.AND.SWITIR.NE.2)
     &         STOP 'PLEASE CHECK VALUE SWITIR IN SOIL DATA FILE'

          IF (SWITIR.EQ.1) CALL RDAREA('RIRRIT',RIRRIT,300,IRIRR)
           IF (SWITIR.EQ.2) CALL RDSREA('IRRI',IRRI)
*-- Reduction factor for evaporation
          CALL RDAREA('RFDDTB',RFDDTB,IMRFDD,ILRFDD)
*-- LAITB called, NAPP also called
          CALL RDAREA('LAITB',LAITB,IMLAI,ILLAI)
          CALL RDAREA('NAPPTB',NAPPTB,100,INAPP)
          CALL RDSINT('NL',NL)
          IF (NL.GT.INL) CALL ERROR('PADDY','too many layers')

          CALL RDFREA('TKL',TKL,MNL,NL)
          CALL RDFREA('WCLI',WCLI,MNL,NL)
          CALL RDFREA('NCONTI',NCONTI,MNL,NL)
          CALL RDFREA('BDSOIL',BDSOIL,MNL,NL)
```

```
          CALL RDFREA('PARAA',PARAA,MNL,NL)
          CALL RDFREA('PARAB',PARAB,MNL,NL)
          CALL RDFREA('PARAC',PARAC,MNL,NL)
          CALL RDFREA('PARAD',PARAD,MNL,NL)
          CALL RDFREA('SET',SET,MNL,NL)

          CALL RDFREA('TRWL',TRWL,MNL,NL)
*---- pf defined in terms of van Genuchten
             CALL RDFREA('VGA',VGA,MNL,NL)
             CALL RDFREA('VGL',VGL,MNL,NL)
             CALL RDFREA('VGN',VGN,MNL,NL)
             CALL RDFREA('VGR',VGR,MNL,NL)
             CALL RDFREA('WCST',WCST,INL,NL)

*--- Bund height
          IF (.NOT.FREEDR) THEN
             CALL RDSREA('WL0MX',WL0MX)
          ELSE
             WL0MX = 0.
          END IF
*--- Minimum ponded water depth if fully irrigated
          IF (.NOT.FREEDR) THEN
             CALL RDSREA('WL0MIN',WL0MIN)
          ELSE
             WL0MIN = 0.
          END IF
*--- Initial ponded water depth
          IF (.NOT.FREEDR) THEN
             CALL RDSREA('WL0I',WL0I)
          ELSE
             WL0I = 0.
          END IF
*--- Saturated hyraulic conductivity
          IF (.NOT.FREEDR) CALL RDFREA('KST',KST,10,NL)
*---- kh switch
          CALL RDSINT('SWITKH',SWITKH)
          IF (SWITKH.NE.0.AND.SWITKH.NE.1.AND.SWITKH.NE.2)
     &       STOP 'PLEASE CHECK VALUE SWITKH IN SOIL DATA FILE'
          IF (SWITKH.EQ.1) THEN
*--- kh defined in terms of van Genuchten parameters
             CALL RDFREA('KST',KST,10,NL)
             CALL RDFREA('VGA',VGA,10,NL)
             CALL RDFREA('VGL',VGL,10,NL)
             CALL RDFREA('VGN',VGN,10,NL)
             CALL RDFREA('VGR',VGR,10,NL)
          ELSE IF (SWITKH.EQ.2) THEN
*--- kh defined in terms of power function
             CALL RDFREA('KST',KST,10,NL)
             CALL RDFREA('PN',PN,10,NL)
          END IF
*---- Reading of soil data completed
          CLOSE (IUNITD)

          I = 1
          DO WHILE (I.LE.NL)
             IF (FREEDR) KST(I) = 1000.
             KSAT(I) = KST(I)
           WCSTRP(I) = WCST(I)
             I = I+1
          END DO
```

```
            I = 1
            DO WHILE (I.LE.NL)
               CALL SUWCMS2(I,2,WCST(I),WCFC(I),300.)
               CALL SUWCMS2(I,2,WCST(I),WCWP(I),1.6E4)
               CALL SUWCMS2(I,2,WCST(I),WCAD(I),1.0E7)
               I = I+1
            END DO

         I = 1
         DO WHILE (I.LE.NL)
            IF (WCLI(I).LT.WCAD(I).OR.WCLI(I).GT.WCST(I))
     &           CALL SUERR(3,WCLI(I),WCAD(I),WCST(I))
            I = I+1
         END DO

         TKLT = 0.

*-- Convert TKL from m into mm; calculate water contents in mm
         I = 1
         DO WHILE (I.LE.NL)
            TKLM(I) = 100*TKL(I)
            TKL(I) = 1000*TKL(I)
            WLFC(I) = WCFC(I)*TKL(I)
            WLAD(I) = WCAD(I)*TKL(I)
            WLST(I) = WCST(I)*TKL(I)
            WL(I) = WCLI(I)*TKL(I)
            I = I+1
         END DO
*--- Initial (total) water content in soil profile (mm)

         I = 1
         DO WHILE (I.LE.NL)
            WCL(I) = WCLI(I)
            WCUMI = WCUMI+WL(I)
            I = I+1
         END DO

         WCUM = WCUMI

*-Nitrate application------
           GNAPP = LINT(NAPPTB,INAPP,DOY)
           NCONAP = GNAPP/((TKL(1)/10.)*1.4)

         I = 1
         DO WHILE (I.LE.NL)
          IF (I.EQ.1) THEN
            NCONT(I) = NCONTI(I)+NCONAP
          ELSE
            NCONT(I) = NCONTI(I)
          END IF
         I = I+1
         END DO

         I = 1
         DO WHILE (I.LE.NL)

          NLODW(I) = (NCONT(I)-(PARAB(I)*BDSOIL(I)))/(WCL(I)+(PARAA(I)
     &                    *BDSOIL(I)))
            IF (NLODW(I).LT.SET(I)) THEN
```

```
            NLODW(I) = NLODW(I)
          ELSE
        NLODW(I) = (NCONT(I)-(PARAD(I)*BDSOIL(I)))/(WCL(I)+(PARAC(I)
     &                  *BDSOIL(I)))

          END IF
         NCONW(I) = NLODW(I)*WCL(I)
         NCONS(I) = NCONT(I)-NCONW(I)
         IF (NCONS(I).LE.TINY) THEN
         NCONS(I) = 0.0
         ELSE
         NCONS(I) = NCONS(I)
         END IF

         NLODS(I) = NCONS(I)/BDSOIL(I)
         I =I+1
         END DO

       I = 1
       DO WHILE (I.LE.NL)
       NTOTW(I) = (NCONW(I)*(TKL(I)/10.)*1.4)
       NTOTS(I) = (NCONS(I)*(TKL(I)/10.)*1.4)
       NTOT(I) = NTOTW(I)+NTOTS(I)
       I = I+1
       END DO

*-- Depth of top of compartments
*--  Initialization of state variables

*---Initial ponded water depth (mm)
        WL0 = WL0I


*-- Reset days since last ponded water
        DSPW = 1.

*---- Reset cumulative amounts
        DRAICU = 0.
        UPRICU = 0.
        GWCU = 0.
        EVSWCU = 0.
        RAINCU = 0.
        RNOFCU = 0.
        TRWCU = 0.
        WCUMCO = 0.
        WL0CO = 0.
        WL0FCU = 0.
        NDRACU = 0.
     ELSE IF (ITASK.EQ.2) THEN

        WL0CH = 0.
        WCUMCH = 0.
        RUNOF = 0.
        EVSW = 0.
        EVSWS = 0.
        IR = 0.
        DRAIN = 0.
        GNAPP = 0.
        NCONAP = 0.
        ECSCS = 0.
        EVSCWL = 0.
```

```
*—— Reset rates to 0
        I = 1
        DO WHILE (I.LE.NL)
           WLFL(I) = 0.
           WLCH(I) = 0.
           MS(I) = 0.
           KMS(I) = 0.
            CONDCA(I) = 0.
            DMS(I) = 0.
           DFLUX(I) = 0.
           CONAU(I) = 0.
           UDMS(I) = 0.
           UDFLUX(I) = 0.
           NOUT(I) = 0.
           NIN(I) = 0.
           NCONCH(I) = 0.
           I = I+1
        END DO
        WLFL(I) = 0.
        UDFLUX(I) = 0.
*—— Transpiration summed over all layers (mm/d)

        TRW = 0
        I = 1
        DO WHILE (I.LE.NL)
           TRW = TRW+TRWL(I)
           I = I+1
        END DO

*— If irrigated, supply constant irrigation (mm/d) if ponded water
*—— Level is below minimum
        IF (SWITIR.EQ.1) IR = LINT(RIRRIT,IRIRR,DOY)

        IF (WL0.LE.WL0MIN.AND.SWITIR.EQ.2) IR = IRRI

        IF (SWITIR.EQ.0) IR = 0.

            IF (SWITKH.NE.0) THEN
            I = 1
            DO WHILE (I.LE.NL)
                  CALL SUWCMS2(I,1,WCST(I),WCL(I),MS(I))
                  CALL SUMSKM2(I,MS(I),WCST(I),KMS(I))
               I = I+1
            END DO
            END IF

          I = 1
         DO WHILE (I.LE.(NL-1))

              DMS(I) = (MS(I+1)-MS(I))
              CONDCA(I) = (SQRT(KMS(I)*KMS(I+1)))
            DFLUX(I) = (CONDCA(I)*((DMS(I)+TKLM(I))/TKLM(I)))*10.
      IF ((DFLUX(I).GE.0.).AND.(MS(I).LE.300.)) THEN
        DFLUX(I) = MIN(10.*KSAT(I),MIN((WL(I)-WLFC(I)),
     &                        (WLST(I+1)-WL(I+1))/DELT))
      ELSE IF ((DFLUX(I).GE.0.).AND.(MS(I).GT.300.)) THEN
        DFLUX(I) = MIN(DFLUX(I),(WLST(I+1)-WL(I+1))/DELT)
      ELSE
        DFLUX(I) = 0.
```

```
        END IF
              UDFLUX(I) = (CONDCA(I)*((DMS(I)+TKLM(I))/TKLM(I)))*10.

          IF (UDFLUX(I).LT.0.0) THEN
            UDFLUX(I) = ABS(UDFLUX(I))
            ELSE
            UDFLUX(I) = 0.
          END IF

        IF (MS(I).GE.300.) THEN
          UDFLUX(I) = MIN(ABS(UDFLUX(I)),MAX(0.0,(WLFC(I)-WL(I))/DELT))
        ELSE
         UDFLUX(I) = 0.
        END IF
          I = I+1
          END DO
        IF (I.EQ.NL) THEN
          DFLUX(I) = MIN(10.*KSAT(I),MAX(0.0,(WL(I)-WLFC(I))/DELT))
          UDFLUX(I) = 0.0
        END IF

          RDAS = 1.
          DTR = RDD
*--Average temperature
          TAV = (TMMN+TMMX)/2.
             LAI = LINT(LAITB,ILLAI,DOY)


      CALL ASTRO(DOY,LAT,SC,DS0,SINLD,COSLD,DAYL,DSINB,DSINBE)

          CALL ETPOT(ANGA,ANGB,DTR,DS0,TAV,VP,WN,LAI,
     &               WCL,WCST,EVSCS,EVSCWL)

*----  Ponded water on field
          IF (WL0.GE.TINY) THEN

*------- Reset number of days after ponded water
              DSPW = 1.

*------- Ponded water can sustain evaporation and transpiration
              IF (WL0/DELT+RAIN+IR.GE.EVSCWL+TRW) THEN

*-------- Calculate change in ponded water depth (mm/d)
                 WL0CH = RAIN+IR-EVSCWL-TRW

*-------- Reset transpiration losses per soil compartment to zero
*-------- as transpiration is taken from ponded water
                 I = 1
                 DO WHILE (I.LE.NL)
                    TRWL(I) = 0
                    I = I+1
                 END DO

*------- For water balance check
                 EVSW = EVSCWL
                 EVSWS = 0.

*------- Calculate flow through boundaries of soil compartments
               WLFL(1) = RAIN+IR+WL0-EVSW
                WL0CH = RAIN+IR-WLFL(1)-EVSW
```

```
                I = 1
               DO WHILE (I.LE.(NL+1))
                 UDFLUX(I) = 0.
                 I = I+1
               END DO

              I = 1
              DO WHILE (I.LE.NL)
                CALL DNFL(I,0,KSAT(I),WLFL(I),TRWL(I),EVSWS,WL(I),
     &                          DFLUX(I),WLFC(I),DELT,WLFL(I+1))

               I = I+1
              END DO
                   IF (.NOT.FREEDR) THEN
                      I = NL
                      DO WHILE (I.GE.1)
             CALL BACKFL(I,NL,0,WL(I),WLFL(I),WLFL(I+1),UDFLUX(I),
     &                    EVSWS,TRWL(I),WLST(I),DELT,FLNEW,REST)
                         WLFL(I) = FLNEW
                         I = I-1
                      END DO


                      WL0CH = WL0CH+(MAX(0.,(REST-WLST(1))/DELT))

                      IF (WL0+WL0CH*DELT.GE.WL0MX) THEN
                         RUNOF = (WL0+WL0CH*DELT-WL0MX)/DELT
                         WL0CH = WL0CH-RUNOF
                      END IF
                   END IF

*—Nitrate fluxes
              I = 1
              DO WHILE (I.LE.NL)
                IF (I.EQ.1) THEN
                   NIN(I) = NLODW(I+1)*(UDFLUX(I)/10.)
                   NOUT(I) = NLODW(I)*(WLFL(I+1)/10.)
                ELSE IF (I.EQ.NL) THEN
                   NIN(I) = (NLODW(I-1)*(WLFL(I)/10.))
                   NOUT(I) = (NLODW(I)*(WLFL(I+1)/10.))
                 ELSE
                   NIN(I) = (NLODW(I+1)*(UDFLUX(I)/10.))+(NLODW(I-1)*
     &                       (WLFL(I)/10.))
                   NOUT(I) = (NLODW(I)*(UDFLUX(I-1)/10.))+(NLODW(I)*
     &                       (WLFL(I+1)/10.))
                END IF
                I = I+1
              END DO


*——— Ponded water can sustain part of evaporation only
              ELSE IF (WL0/DELT+RAIN+IR.LT.EVSCWL) THEN

*——— Calculate change in ponded water depth (mm/d)
                 WL0CH = -WL0/DELT
                 WLFL(1) = RAIN+IR
                 I = 2
                 DO WHILE (I.LE.NL+1)
                    WLFL(I) = 0.
                    I = I+1
                 END DO
```

```
*------- Calculate contribution of first soil compartment to
*------- evaporation
              EVSW = MIN(EVSCWL+WL0CH,WL(1)/DELT-WLAD(1)/DELT+RAIN+IR)

              EVSWS = EVSW
*------- For water balance check
              EVSW = WL0/DELT+EVSWS
               I = 1
             DO WHILE (I.LE.(NL+1))
               UDFLUX(I) = 0.
               I = I+1
             END DO

*- Nitrate fluxes
            I = 1
            DO WHILE (I.LE.NL)
              IF (I.EQ.1) THEN
                 NIN(I) = NLODW(I+1)*(UDFLUX(I)/10.)
                 NOUT(I) = NLODW(I)*(WLFL(I+1)/10.)
              ELSE IF (I.EQ.NL) THEN
                 NIN(I) = (NLODW(I-1)*(WLFL(I)/10.))
                 NOUT(I) = (NLODW(I)*(WLFL(I+1)/10.))
              ELSE
                 NIN(I) = (NLODW(I+1)*(UDFLUX(I)/10.))+(NLODW(I-1)*
     &                        (WLFL(I)/10.))
                 NOUT(I) = (NLODW(I)*(UDFLUX(I-1)/10.))+(NLODW(I)*
     &                        (WLFL(I+1)/10.))
              END IF
              I = I+1
            END DO

            END IF

          ELSE



*------- No ponded water on surface
*------- Calculate evaporation rate from soil surface (mm/d)
            ZDD = (WCL(1)-WCAD(1))/(WCST(1)-WCAD(1))
            RFDD = LINT(RFDDTB,ILRFDD,ZDD)
            LAI = LINT(LAITB,ILLAI,DOY)
            EVSW = EVSCS*RFDD

            EVSWS = EVSW.
            DSPW = DSPW+1.

            WLFL(1) = MIN(RAIN+IR,(WLST(1)-WL(1)+EVSWS))

            I = 1
            DO WHILE (I.LE.NL)
           CALL DNFL(I,1,KSAT(I),WLFL(I),TRWL(I),EVSWS,WL(I),
     &                 DFLUX(I),WLFC(I),DELT,WLFL(I+1))

              I = I+1
            END DO
```

```
            IF (.NOT.FREEDR) THEN
               I = NL
               DO WHILE (I.GE.1)
             CALL BACKFL(I,NL,1,WL(I),WLFL(I),WLFL(I+1),UDFLUX(I+1),
     &                     EVSWS,TRWL(I),WLST(I),DELT,FLNEW,REST)
                  WLFL(I) = FLNEW
                  I = I-1
               END DO

                WL0CH = RAIN+IR-WLFL(1)
               IF (WL0+WL0CH*DELT.GE.WL0MX) THEN
                  RUNOF = (WL0+WL0CH*DELT-WL0MX)/DELT
                  WL0CH = WL0CH-RUNOF
               END IF
              END IF

*--Nitrate fluxes--------
            I = 1
            DO WHILE (I.LE.NL)
              IF (I.EQ.1) THEN
                 NIN(I) = NLODW(I+1)*(UDFLUX(I)/10.)
                 NOUT(I) = NLODW(I)*(WLFL(I+1)/10.)
              ELSE IF (I.EQ.NL) THEN
                 NIN(I) = (NLODW(I-1)*(WLFL(I)/10.))
                 NOUT(I) = (NLODW(I)*(WLFL(I+1)/10.))
              ELSE
                 NIN(I) = (NLODW(I+1)*(UDFLUX(I)/10.))+(NLODW(I-1)*
     &                     (WLFL(I)/10.))
                 NOUT(I) = (NLODW(I)*(UDFLUX(I-1)/10.))+(NLODW(I)*
     &                     (WLFL(I+1)/10.))
              END IF
              I = I+1
            END DO

            END IF


          I = 1
          DO WHILE (I.LE.NL)
             IF (I.EQ.1) THEN
                WLCH(I) = WLFL(I)-WLFL(I+1)-TRWL(I)-EVSWS+UDFLUX(I)
             ELSE
              WLCH(I) = WLFL(I)-WLFL(I+1)-TRWL(I)+UDFLUX(I)-UDFLUX(I-1)
             END IF
             WCUMCH = WCUMCH+WLCH(I)
             I = I+1
          END DO
*-Nitrate application content-------
            GNAPP = LINT(NAPPTB,INAPP,DOY)
            NCONAP = GNAPP/((TKL(1)/10.)*1.4)

          I = 1
          DO WHILE (I.LE.NL)
           IF (I.EQ.1) THEN
             NCONCH(I) = (NIN(I)-NOUT(I))/(TKL(I)/10.)+NCONAP
           ELSE
             NCONCH(I) = (NIN(I)-NOUT(I))/(TKL(I)/10.)
           END IF
            I = I+1
          END DO
```

```
*—Estimation of NDRAIN in kg/ha
         I = NL
         DO WHILE  (I.EQ.NL)
           NDRAIN(I) = NOUT(I)/(TKL(I)/10.)
           NDRAKG(I) = NDRAIN(I)*(TKL(I)/10.)*1.4
         I = I+1
         END DO

         IF (OUTPUT) THEN
*            CALL OUTARR('WLCH',WLCH,1,5)
             CALL OUTDAT(2,0,'WL0',WL0)
             CALL OUTDAT(2,0,'RAIN',RAIN)
*            CALL OUTDAT(2,0,'IR',IR)
*            CALL OUTARR('TRWL',TRWL,1,6)
             CALL OUTARR('WCL',WCL,1,5)
*            CALL OUTARR('WLFL',WLFL,1,6)
*            CALL OUTARR('DFLUX',DFLUX,1,10)
*            CALL OUTARR('MS',MS,1,5)
*            CALL OUTARR('WL',WL,1,5)
*            CALL OUTARR('KMS',KMS,1,5)
*            CALL OUTARR('UDFLUX',UDFLUX,1,5)
*            CALL OUTARR('NLODT',NLODT,1,NL)
*            CALL OUTARR('NLODS',NLODS,1,NL)
*            CALL OUTARR('NLODW',NLODW,1,NL)
*            CALL OUTARR('NCONS',NCONS,1,NL)
*            CALL OUTARR('NCONW',NCONW,1,NL)
*            CALL OUTARR('NCONT',NCONT,1,NL)
*            CALL OUTARR('NIN',NIN,1,NL)
*            CALL OUTARR('NOUT',NOUT,1,NL)
*            CALL OUTARR('NTOTW',NTOTW,1,NL)
*            CALL OUTARR('NTOTS',NTOTS,1,NL)
*            CALL OUTARR('NTOT',NTOT,1,NL)
*            CALL OUTDAT(2,0,'ZW',ZW)
*            CALL OUTDAT(2,0,'EVSC',EVSC)
*            CALL OUTDAT(2,0,'NCONAP',NCONAP)
*            CALL OUTDAT(2,0,'EVSW',EVSW)
*            CALL OUTDAT(2,0,'EVSWS',EVSWS)
*            CALL OUTDAT(2,0,'DRAICU',DRAICU)
*            CALL OUTDAT(2,0,'UPRICU',UPRICU)
*            CALL OUTDAT(2,0,'EVSWCU',EVSWCU)
*            CALL OUTDAT(2,0,'RAINCU',RAINCU)
*            CALL OUTDAT(2,0,'RNOFCU',RNOFCU)
*            CALL OUTDAT(2,0,'TRWCU',TRWCU)
*            CALL OUTDAT(2,0,'WCUMCO',WCUMCO)
*            CALL OUTDAT(2,0,'WL0CO',WL0CO)
*            CALL OUTDAT(2,0,'NDRACU',NDRACU)

         END IF


      ELSE IF (ITASK.EQ.3) THEN

*— Integration of state variables
         WL0 = INTGRL(WL0,WL0CH,DELT)

         I = 1
         DO WHILE (I.LE.NL)
            WL(I) = INTGRL(WL(I),WLCH(I),DELT)
            I = I+1
         END DO
```

```
          I = 1
          DO WHILE (I.LE.NL)
             WCL(I) = WL(I)/TKL(I)
             I = I+1
          END DO

           I = 1
           DO WHILE (I.LE.NL)
             NCONT(I) = INTGRL(NCONT(I),NCONCH(I),DELT)
             NCONT(I) = MAX(0.,NCONT(I))
             I = I+1
           END DO

          I = 1
          DO WHILE (I.LE.NL)

           NLODW(I) = (NCONT(I)-(PARAB(I)*BDSOIL(I)))/(WCL(I)+(PARAA(I)
     &                      *BDSOIL(I)))
             IF (NLODW(I).LT.SET(I)) THEN
               NLODW(I) = NLODW(I)
             ELSE
           NLODW(I) = (NCONT(I)-(PARAD(I)*BDSOIL(I)))/(WCL(I)+(PARAC(I)
     &                      *BDSOIL(I)))

              END IF
            NCONW(I) = NLODW(I)*WCL(I)
            NCONS(I) = NCONT(I)-NCONW(I)
            IF (NCONS(I).LE.TINY) THEN
            NCONS(I) = 0.0
            ELSE
            NCONS(I) = NCONS(I)
            END IF

            NLODS(I) = NCONS(I)/BDSOIL(I)
            I =I+1
            END DO

          I = 1
         DO WHILE (I.LE.NL)
         NTOTW(I) = (NCONW(I)*(TKL(I)/10.)*1.4)
         NTOTS(I) = (NCONS(I)*(TKL(I)/10.)*1.4)
         NTOT(I) = NTOTW(I)+NTOTS(I)
         I = I+1
         END DO
*--- Cumulative amounts
          DRAICU = DRAICU-WLFL(NL+1)*DELT
          EVSWCU = EVSWCU-EVSW*DELT
          RAINCU = RAINCU+(RAIN+IR)*DELT
          RNOFCU = RNOFCU-RUNOF*DELT
          TRWCU = TRWCU-TRW*DELT
          NDRACU = NDRACU-NDRAKG(NL)*DELT

*--- Water balance check
          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 = 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+DRAICU+UPRICU
*-- Check this
        CALL SUWCHK(CKWFL,CKWIN,TIME)

      ELSE IF (ITASK.EQ.4) THEN
      END IF
      ITOLD = ITASK
      RETURN

      END


*-----------------------------------------*
* SUBROUTINE SUMSKM2                                                  *
*                                                                     *
* Purpose: SUMSKM2 calculates the hydraulic conductivity at           *
*          given suction for compartment I on the basis of chosen     *
*          option                                                     *
*                                                                     *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)     *
* name   type meaning (unit)                                   class  *
* ----   ---- -------                                          ---- *
* I       I4  Compartment index (-)                             I    *
* MS      R4  Soil water suction (cm)                           I    *
* WCST    R4  Array of water content saturation/layer (cm3 cm-3)  O  *
* KMS     R4  Hydraulic conductivity (cm d-1)                   O    *
*                                                                     *
* SUBROUTINES called :                                                *
*                                                                     *
*     - SUERR, SUWCMS2                                                *
*                                                                     *
* FUNCTIONS called :  none                                            *
*                                                                     *
* FILE usage : none                                                   *
*-----------------------------------------*

      SUBROUTINE SUMSKM2(I,MS,WCST,KMS)

      IMPLICIT REAL(A-Z)
      INTEGER I

*-- Common blocks
      INTEGER SWITKH
      COMMON /NUCHT / VGN(10),VGA(10),VGR(10),VGL(10)
      COMMON /HYDCON/ KST(10),WCAD(10),WCSTRP(10)
      COMMON /POWER / PN(10)
      COMMON /SWIT  / SWITKH
*-- 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) CALL SUERR(1,MS,0.,1.E8)

      IF (MS.GE.MSAD-TINY) THEN
*-- Air dry
       KMS = 0.
      ELSE
*-- Calculate conductivity
         IF (SWITKH.EQ.1) THEN
*-- van Genuchten conductivity
            WCL = 0.
*           dummy value; wcl is returned by suwcms2!
            CALL SUWCMS2(I,2,WCST,WCL,MS)
            VGM = 1.0-1.0/VGN(I)
            WREL = (WCL-VGR(I))/(WCSTRP(I)-VGR(I))
            HLP1 = WREL**VGL(I)
            HLP2 = 1.0-WREL**(1./VGM)
            HLP3 = 1.0-HLP2**VGM
            KMS = KST(I)*HLP1*HLP3*HLP3
         ELSE IF (SWITKH.EQ.2) THEN
*-- Power function conductivity
            IF (MS.LE.1.) KMS = KST(I)
            IF (MS.GT.1.) KMS = KST(I)*(MS**PN(I))
         ELSE IF (SWITKH.EQ.5) THEN
*           User can here specify preferred conductivity function
*           The following two lines should be removed:
            WRITE (*,10)
            STOP
         END IF
         IF (KMS.LT.TINY) KMS = 0.
      END IF
10    FORMAT (///,' *** fatal error; option SWIT3=5 requires ',/,
     &        ' specification of conductivity function')
      RETURN
      END

*-----------------------------------------*
*  SUBROUTINE SUWCMS2                                                  *
*                                                                      *
*  Purpose: SUWCMS2 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 (unit)                                    class *
* ----   -- -------                                          -- *
* I       I4  Compartment index (-)                              I  *
* SWIT4   I4  Switch to set request MS(WCL) or WCL(MS) (-)       I  *
* WCST    R4  Array of water content saturation / layer (cm3 cm-3)  I  *
* WCL     R4  Array of actual water content / layer (cm3 cm-3)   I/O *
* MS      R4  Soil water suction (cm)                            I/O *
*                                                                      *
*  SUBROUTINES called:                                                 *
*   -   SUERR                                                          *
*                                                                      *
*  FUNCTION called:                                                    *
*   -   none                                                           *
*                                                                      *
```

```
*  FILE usage:                                                          *
*   -   none                                                            *
*                                                                       *
************************************************************************

      SUBROUTINE SUWCMS2(I,SWIT4,WCST,WCL,MS)

      IMPLICIT REAL(A-Z)
      INTEGER I,SWIT4

*-- Common blocks
      COMMON /NUCHT / VGN(10),VGA(10),VGR(10),VGL(10)
      COMMON /HYDCON/ KST(10),WCAD(10),WCSTRP(10)
*-- Variables retain their values between subsequent calls
*     of this subroutine
      SAVE

      DATA TINY/0.001/

      IF (SWIT4.EQ.1) THEN
*        Suction calculated from water content
         IF (WCL.LT.WCAD(I))
     &         CALL SUERR(3,WCL,WCAD(I),WCST)
        IF (WCL.GT.WCSTRP(I).OR.(WCL.GT.WCST)) THEN
*           It is assumed that MS remains zero during shrinkage
            MS = 0.
         ELSE
*           van Genuchten option
            HLP1 = AMAX1(WCAD(I),WCL)
            WREL = (WCL-VGR(I))/(WCSTRP(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
         END IF
      ELSE IF (SWIT4.EQ.2) THEN
*        Water content calculated from suction
         IF (MS.LT.-TINY.OR.MS.GT.1.E8) CALL SUERR(4,MS,0.,1.E8)
*        van Genuchten option
         VGM = 1.-1./VGN(I)
         HLP1 = (MS*VGA(I))**VGN(I)
         WREL = (1.+HLP1)**(-VGM)
         WCL = WREL*(WCSTRP(I)-VGR(I))+VGR(I)
      END IF

      RETURN
      END

*-----------------------------------------*
* SUBROUTINE SUERR                                                      *
*                                                                       *
* Purpose: SUERR checks whether value of variable X is within           *
*          prespecified domain                                          *
*                                                                       *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)       *
* name   type meaning (unit)                                     class *
* ---    --- --------                                          --- *
* IMNR     I4  Message number                                     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)


*(JJ) IUNLOG is an obsolete variable: it is NEVER assigned a value, but
*     still it is 'used' in determining whether to write something to
*     the log file it is supposed to be defined by.
*     Note that this variable is used in various subroutines
*     throughout WBAL8.

      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 SUMSKM2',
     &     'WATER CNT OUT OF RANGE IN SUSLIN     ',
     &     'WATER CNT OUT OF RANGE IN SUWCMS2     ',
     &     'MATRIC SUCTION OUT OF RANGE IN SUWCMS2',
     &     '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)
      END IF
      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 SUWCHK                                                    *
*                                                                      *
* 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 (unit)                                    class  *
* ----   -- --------                                          -- *
* CKWFL    R4  Sum of time-integrated boundary fluxes (mm)          I  *
* CKWIN    R4  Change in water storage since start (mm)             I  *
* TIME     R4  Time of simulation (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 DOWNFL                                                    *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name   type meaning (unit)                                    class  *
* ----   -- --------                                          -- *
* I        I4  Compartment index (-)                              I  *
* KSAT     R4  Saturated hydraulic conductivity (cm d-1)          I  *
* FLIN     R4  Flux into soil compartment (mm d-1)                I  *
* TRWL     R4  Array of actual transpiration rate/layer (mm d-1)  I  *
```

```
* EVSWS    R4  Actual evaporation rate soil compartment 1 (m d-1)    I  *
* WL       R4  Actual water content (mm)                             I  *
* WLFC     R4  Array amount of water per soil compartment at 'field     *
*              capacity' (mm)                                        I  *
* DELT     R4  Time step of integration (d)                          T  *
* FLOUT    R4  Flux out of soil compartment (mm d-1)                 O  *
*-------------------------------------------*

      SUBROUTINE DNFL(I,SWIC,KSAT,FLIN,TRWL,EVSWS,WL,DFLUX,WLFC,
     &                     DELT,FLOUT)

      IMPLICIT REAL(A-Z)
      INTEGER I,SWIC
      SAVE
      IF (SWIC.EQ.0) THEN
      IF (I.EQ.1) THEN
         FLOUT = MIN(10.*KSAT,MAX(0.,FLIN-EVSWS-TRWL+(WL-WLFC)/DELT))
       ELSE
         FLOUT = MIN(10.*KSAT,MAX(0.,FLIN-TRWL+(WL-WLFC)/DELT))
      END IF
       ELSE
         FLOUT = DFLUX
       END IF
      RETURN
      END


*-------------------------------------------*
* SUBROUTINE BACKFL                                                    *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name   type meaning (unit)                                    class  *
* ----   -- -------                                          -- *
* I        I4  Compartment index (-)                           I  *
* WL       R4  Actual water content (mm)                       I  *
* FLIN     R4  Flux into soil compartment (mm d-1)             I  *
* FLOUT    R4  Flux out of soil compartment (mm d-1)           I  *
* EVSWS    R4  Actual evaporation rate soil compartment 1 (m d-1)  I  *
* TRWL     R4  Array of actual transpiration rate/layer (mm d-1)   I  *
* WLST     R4  Array amount of water per soil compartment at       *
*              saturation (mm)                                     I  *
* DELT     R4  Time step of integration (d)                        T  *
* FLNEW    R4  Boundary flow between soil compartments recalculated *
*              via subroutine BACKFL (mm d-1)                      O  *
* HLP      R4  Help variable (mm)                                  O  *
*-------------------------------------------*

      SUBROUTINE BACKFL(I,NL,SWIC,WL,FLIN,FLOUT,UDFLUX,EVSWS,TRWL,
     &                     WLST,DELT,FLNEW,HLP)

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

      REAL UDFLUX(NL+1)
      SAVE

      HLP = 0.
      IF (SWIC.EQ.0) THEN
      IF (I.EQ.1) THEN
         HLP = WL+(FLIN-FLOUT-EVSWS-TRWL)*DELT
```

```
      ELSE
         HLP = WL+(FLIN-FLOUT-TRWL)*DELT
      END IF

      IF (HLP.GT.WLST) THEN
         FLNEW = FLIN-(HLP-WLST)/DELT
      ELSE
         FLNEW = FLIN
      END IF
      ELSE

      IF (I.EQ.1) THEN
         HLP = WL+(FLIN-FLOUT-EVSWS-TRWL+UDFLUX(I))*DELT
      ELSE
         HLP = WL+(UDFLUX(I)+FLIN-FLOUT-TRWL-UDFLUX(I-1))*DELT
      END IF

      IF (HLP.GT.WLST) THEN
         FLNEW = FLIN-(HLP-WLST)/DELT
      ELSE
         FLNEW = FLIN
      END IF
      END IF
      RETURN
      END

*-----------------------------------------*
* SUBROUTINE ETPOT-1                                                  *
* Author: B.A.M. Bouman                                               *
* Version: 2.0                                                        *
* Date: November 1993                                                 *
*  Slightly modified by M.V.R. Murty                                  *
*   May 2000                                                          *
* Purpose: Calculation of Penman reference value for potential evapo- *
*          transpiration of a reference crop (mostly from formulation *
*          as given in van Laar et al 1992).                          *
*          Calculation of potential transpiration of a rice crop (with *
*          a soil or a water layer background), and of potential      *
*          evaporation of soil surfaces and of open water.            *
* Reference: van Laar HH, Goudriaan J, van Keulen H, editors. 1992.   *
*          Simulation of crop growth for potential and water-limited  *
*          production situations (as applied to spring wheat),        *
*          CABO-DLO report 27. Wageningen (The Netherlands): CABO-DLO. *
*                    *
*                                                                     *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)     *
* name   type meaning (unit)                                  class *
* ----   ---- --------                                      --- *
* SWIWLP  I4  Switch to select production environment (-)       C  *
* ITIM    I4  Time of simulation (d)                            T  *
* ITRT    I4  Time of transplanting (d)                         T  *
* ANGA    R4  Constant A in Angstrom formulae (-)               I  *
* ANGB    R4  Constant B in Angstrom formulae (-)               I  *
* RDT     R4  Daily solar radiation (J m-2 d-1)                 I  *
* DSO     R4  Daily extraterrestrial radiation (J m-2 d-1)      I  *
* TAV     R4  Average daily temperature (°C)                    I  *
* VP      R4  Early morning vapor pressure (kPa)                I  *
* WN      R4  Daily average windspeed (m s-1)                   I  *
* LAI     R4  Apparent leaf area index (incl. stem area) (ha ha-1)  I  *
* WCLQT   R4  Array of actual soil water contents/layer (cm3 cm-3)  I  *
```

```
* WCST     R4  Array of water content saturation/layer (cm3 cm-3)    I  *
* WL0      R4  Amount of ponded water (mm)                           I  *
* TRC      R4  Potential transpiration rate (mm d-1)                 O  *
* EVSC     R4  Potential soil evaporation rate (mm d-1)              O  *
*                                                                       *
*  FATAL ERROR CHECKS: none                                             *
*  FILE usage : none                                                    *
*-----------------------------------------*
      SUBROUTINE ETPOT(ANGA,ANGB,RDT,DS0,TAV,VP,WN,LAI,
     &                 WCL,WCST,EVSCS,EVSCWL)


      IMPLICIT REAL(A-Z)
      REAL WCL(1),WCST(1)

      SAVE

*---Conversion from kpa -> mbar
      VAPOR = VP*10.
      WIND = WN

      LHVAP = 2.4E6
      PSYCH = 0.67
      BOLTZM = 5.668E-8
      ALBDS = 0.25
      ALBOW = 0.05
      ALBC = 0.25
      WCUP = WCL(1)
      WCSTUP = WCST(1)
*=========================================================*
*---Calculation of Penman terms for evapotranspiration   *
*=========================================================*

      SVP = 6.11*EXP(17.4*TAV/(TAV+239.))
      SLOPE = 4158.6*SVP/(TAV+239.)**2

      ALBS = ALBDS*(1.-0.5*WCUP/WCSTUP)
      ALB = ALBS*EXP(-0.5*LAI)+ALBC*(1.-EXP(-0.5*LAI))
      ALBWL = ALBOW*EXP(-0.5*LAI)+ALBC*(1.-EXP(-0.5*LAI))

      CLEAR = LIMIT(0.,1.,((RDT/DS0)-ANGA)/ANGB)
      FCLEAR = 0.1+0.9*CLEAR
      FVAP = 0.56-0.079*SQRT(VAPOR)
      BBRAD = BOLTZM*(TAV+273.)**4
      RLWN = BBRAD*FVAP*FCLEAR*86400.
      NRAD = (1.-ALB)*RDT-RLWN
      NRADWL = (1.-ALBWL)*RDT-RLWN
      NRADOW = (1.-ALBOW)*RDT-RLWN

      WDF = 0.263*(1.0+0.54*WIND)
      WDFOW = 0.263*(0.5+0.54*WIND)
      DRYP = (SVP-VAPOR)*WDF
      DRYPOW = (SVP-VAPOR)*WDFOW

*---Calculation of EVR and EVD
      EVD = DRYP*PSYCH/(SLOPE+PSYCH)
      EVDOW = DRYPOW*PSYCH/(SLOPE+PSYCH)
      EVR = (1./LHVAP)*(SLOPE/(SLOPE+PSYCH))*NRAD
      EVRWL = (1./LHVAP)*(SLOPE/(SLOPE+PSYCH))*NRADWL
      EVROW = (1./LHVAP)*(SLOPE/(SLOPE+PSYCH))*NRADOW
```

```
*=============================================================*
*---Calculation of transpiration and evaporation of crop *
*=============================================================*

*---Crop transpiration with water layer
      TRCWL = EVRWL*(1.-EXP(-0.5*LAI))+EVD*(MIN(2.5,LAI))
*---Crop transpiration with soil background
      TRCS = EVR*(1.-EXP(-0.5*LAI))+EVD*(MIN(2.5,LAI))

*---Soil evaporation with water layer
      EVSCWL = EXP(-0.5*LAI)*(EVRWL+EVD)
*---Soil evaporation with soil background
      EVSCS = EXP(-0.5*LAI)*(EVR+EVD)

*---Open water evaporation
      EVSCOW = EVROW+EVDOW

      RETURN
      END
*-------------------------------------------*
*  SUBROUTINE ASTRO                                                  *
*  Purpose: This subroutine calculates astronomic daylength,         *
*           diurnal radiation characteristics, such as the daily      *
*          integral of sine of solar elevation, and solar constant.  *
*                                                                    *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)    *
* name   type meaning (unit)                                  class *
* ----   -- -------                                          -- *
* DOY     R4  Daynumber (January 1 = 1) (-)                    I  *
* LAT     R4  Latitude of site (dec. degr.)                    I  *
* SC      R4  Solar constant (J m-2 s-1)                       O  *
* DS0     R4  Daily extraterrestrial radiation (J m-2 d-1)     O  *
* SINLD   R4  Seasonal offset of sine of solar height (-)      O  *
* COSLD   R4  Amplitude of sine of solar height (-)            O  *
* DAYL    R4  Astronomic daylength (base = 0 degrees) (h)      O  *
* DSINB   R4  Daily total of sine of solar height (s)          O  *
* DSINBE  R4  Daily total of effective solar height (s)        O  *
*                                                                 *
*  FATAL ERROR CHECKS (execution terminated, message)             *
*  condition: LAT > 67, LAT < -67                                 *
*                                                                 *
*  FILE usage : none                                              *
*-------------------------------------------*
      SUBROUTINE ASTRO(DOY,LAT,SC,DS0,SINLD,COSLD,DAYL,DSINB,DSINBE)
      IMPLICIT REAL(A-Z)
      SAVE

*-- PI and conversion factor from degrees to radians
       PI = 3.141592654
       RAD = PI/180.

*-- Check on input range of parameters
       IF (LAT.GT.67.) STOP 'ERROR IN ASTRO: LAT> 67'
       IF (LAT.LT.-67.) STOP 'ERROR IN ASTRO: LAT>-67'

*-- Declination of the sun as function of daynumber (DOY)
       DEC = -ASIN(SIN(23.45*RAD)*COS(2.*PI*(DOY+10.)/365.))
```

```
*-- SINLD, COSLD and AOB are intermediate variables

      SINLD = SIN(RAD*LAT)*SIN(DEC)
      COSLD = COS(RAD*LAT)*COS(DEC)
      AOB = SINLD/COSLD

*-- Daylength (DAYL)
      DAYL = 12.0*(1.+2.*ASIN(AOB)/PI)

      DSINB = 3600.*(DAYL*SINLD+24.*COSLD*SQRT(1.-AOB*AOB)/PI)
      DSINBE = 3600.*(DAYL*(SINLD+0.4*(SINLD*SINLD+COSLD*COSLD*0.5))
     &         +12.0*COSLD*(2.0+3.0*0.4*SINLD)*SQRT(1.-AOB*AOB)/PI)

*-- Solar constant (SC) and daily extraterrestrial radiation (DS0)
      SC = 1370.*(1.+0.033*COS(2.*PI*DOY/365.))
      DS0 = SC*DSINB

      RETURN
      END
```

Appendix 2. Example of input file for UPLAND model by Murty and Kondo.

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

* Switches:
* Drainage switch: free draining (1); impeded drainage (0)
SWITFD = 1
* Irrigation switch: no irrigation (0); irrigation read from table (1);
* irrigation if ponded water depth drops below minimum value (2)
SWITIR = 1
* Conductivity switch: van Genuchten parameters (1);
* Power function (2)
SWITKH = 1

* Number of soil layers (maximum is 10)
NL = 5
TRWL = 5*0.0

* From the phytotron (sat> content)New
RFDDTB = 0.1,0.05,0.25,0.1,0.45,0.4,0.65,0.9,1.0,1.0,1.1,1.0

*Initial nitrate content of soil
NCONTI = 5*0.001
* Bulk density layerwise
BDSOIL = 1.1,1.3,1.2,1.1,1.1
* Parameters for nitrate equilibrium
PARAA = 5*0.0
PARAB = 5*0.
PARAC = 5*0.
PARAD = 5*0.0
SET   = 5*1.0
* Leaf area as a forcing function
LAITB = 1.0,0.0,365.,0.0
* Nitrate application rate (kg/ha)
NAPPTB = 0.0,0.0,95.0,0.0,96.0,20.0,97.0,0.0,365.0,0.0
* Thickness of soil compartments (m)
TKL = 5*0.20

* Irrigation table, amount of irrigation (y in mm) for a given calendar
* day (x), used if SWITIR = 1
RIRRIT = 0., 20.,366., 20.

* Irrigation parameter, used if SWITIR = 2, i.e., amount of irrigation
* If ponded water depth drops below WL0MIN (mm)
IRRI = 90.

* Saturated hydraulic conductivity
KST = 43.5,7.7,7.7,74.,200.
* van Genuchten parameters
VGA = 0.04056,0.0322,0.02119,0.1827,0.77218
VGL = 0.5,-1.5,0.5,-7.0,0.5
VGN = 1.15095,1.1559,1.12044,1.1478,1.13307
VGR = 5*0.01
```

```
* Power function parameter (needed if SWITKH = 2)
PN = 3*-2.5, 3*-2.5, 2*-2.5, -2.5

* Saturated volumetric water content
WCST = 0.57,0.65,0.68,0.67,0.60
* Initial volumetric water content
WCLI = 0.3202, 0.4765, 0.5037, 0.5835,0.5835

* Ponded water depth (mm)
WL0MX = 100.
* Minimum ponded water depth (mm)
WL0MIN =50.
* Initial ponded water depth (mm)
WL0I =0.
```

Appendix 3. Timer file generated by FST translator version 1.15.

```
*
*
* contains:
* - The used DRIVER and TRACE in case of GENERAL translation
* - The TIMER variables used in both translation modes
* - Additional TIMER variables in case of GENERAL translation
* - The WEATHER control variables if weather data are used
* - Miscellaneous FSE variables in case of FSE translation
*
* TIMER variables used in GENERAL and FSE translation modes
* ---------------------------------
STTIME = 85.                    ! start time
FINTIM = 180.                   ! finish time
DELT   = 1.                     ! time step (for Runge-Kutta first guess)
PRDEL  = 1.                     ! output time step
IPFORM = 4                      ! code for output table format:
                                ! 4 = spaces between columns
                                ! 5 = TABs between columns (spreadsheet output)
                                ! 6 = two-column output
  MULTIP = 1.
  ANGA = 0.29
  ANGB = 0.42

           ! The string array PRSEL contains the output variables for which
           ! formatted tables have to be made. One or more times there is a
           ! series of variable names terminated by the word <TABLE>.
           ! The translator writes the variables in each PRINT statement to
* PRSEL =  ! a separate table

 COPINF = 'N'                   ! Switch variable whether to copy the input files
                                ! to the output file ('N' = do not copy,
                                ! 'Y' = copy)
 DELTMP = 'N'                   ! Switch variable what should be done with the
                                ! temporary output file ('N' = do not delete,
                                ! 'Y' = delete)
 IFLAG  = 1100                  ! Indicates where weather error and warnings
                                ! go (1101 means errors and warnings to log
                                ! file, errors to screen, see FSE manual)
*IOBSD = 1991,182               ! List of observation data for which output is
                                ! required. The list should consist of pairs
                                ! <year>,<day> in combination
* WEATHER control variables
* ------------------
 WTRDIR = ' '
 CNTR   = 'phil'                ! Country code
 ISTN   = 2                     ! Station code
 IYEAR  = 1997                  ! Year

 SWIWLP = 3
```

Appendix 4. File names to be used by FSE 2.0.

```
*-----------------------------------------*
*                                         *
* The input files (except FILEIR) may be used in reruns.  *
* Up to five input data files may be used (FILEI1-5).     *
*                                         *
*-----------------------------------------*

   FILEON = 'RESULTS.OUT'    ! Normal output file
   FILEOL = 'MODEL.LOG'      ! Log file
   FILEIR = 'RERUNS.DAT'     ! Reruns file
   FILEIT = 'TIMER.DAT'      ! File with timer data
*  FILEI1 = ' '              ! First input data file (not used)
   FILEI2 = 'soil.dat '      ! Second input data file (soil)
*  FILEI3 = ' '              ! Third input data file (not used)
*  FILEI4 = ' '              ! Fourth input data file (not used)
*  FILEI5 = ' '              ! Fifth input data file (not used)
```