CEN0257
Modelagem de Sistemas Agrícolas e Ecológicos

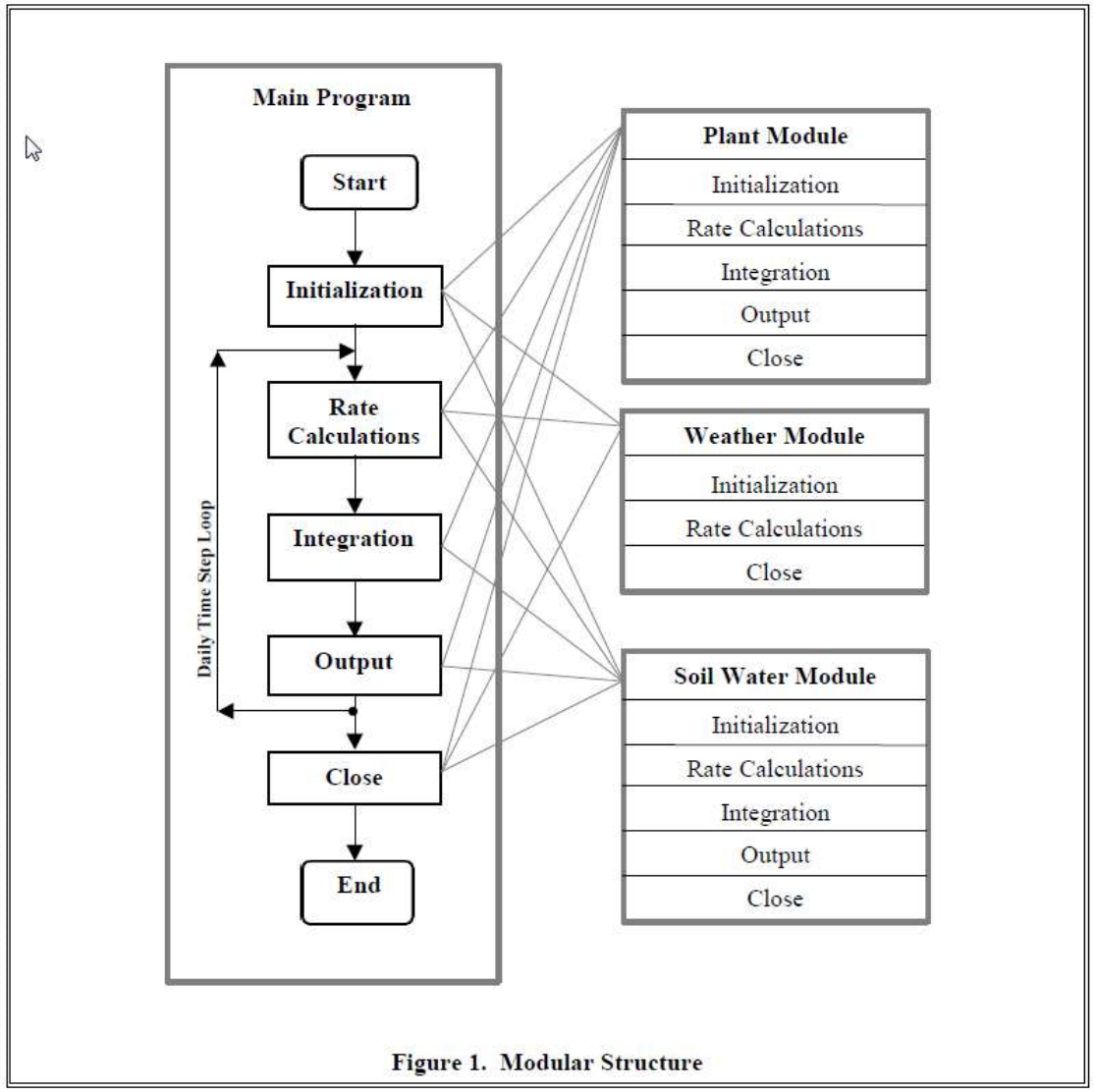
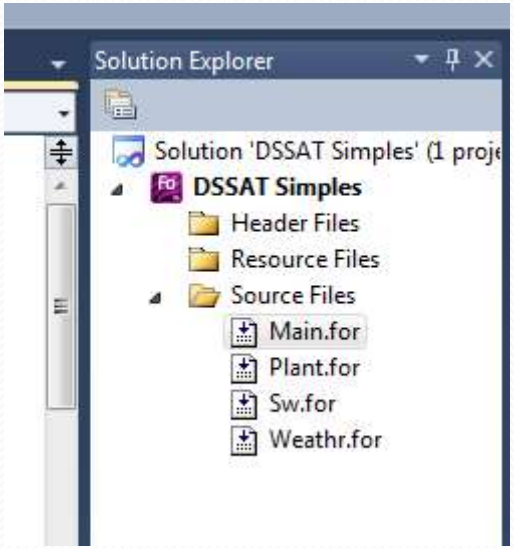


Figure 1. Modular Structure



Variáveis de estado (*State variables*) Biomassa Y (kg/ha)

Variáveis de taxa (*Rate variables*) Taxa de crescimento dY/dt (kg/ha/d)

Integração $Y_{t+1} = Y_t + dY/dt * dt$ (kg/ha)

estado taxa passo de tempo

```
!***********************************************************************
!***********************************************************************
!     INITIALIZATION AND INPUT OF DATA
!***********************************************************************
      CALL OPENF(DOYP, FROP)

      CALL WEATHR(SRAD,TMAX,TMIN,RAIN,PAR,'INITIAL    ')

      CALL SW(
     &    DOY, LAI, RAIN, SRAD, TMAX, TMIN,              !Input
     &    SWFAC1, SWFAC2,                                !Output
     &    'INITIAL    ')                                 !Control

      CALL PLANT(DOY, endsim, TMAX, TMIN,
     &    PAR, SWFAC1, SWFAC2,
     &    LAI,
     &    'INITIAL    ')

!-----------------------------------------------------------------------
!     DAILY TIME LOOP
!-----------------------------------------------------------------------
      DO 500 DOY = 0,1000
        IF (DOY .NE. 0) THEN

          CALL WEATHR(SRAD,TMAX,TMIN,RAIN,PAR,'RA


!***********************************************************************
```

```
      SUBROUTINE SW(
     &    DOY, LAI, RAIN, SRAD, TMAX, TMIN,              !Input
     &    SWFAC1, SWFAC2,                                !Output
     &    DYN)                                           !Control

!-----------------------------------------------------------------------
      IMPLICIT NONE
      SAVE

      INTEGER   DATE, DOY
      REAL      SRAD,TMAX,TMIN,RAIN,SWC,INF,IRR,ROF,ESa,EPa,DRNp
      REAL      DRN,DP,WPp,FCp,STp,WP,FC,ST,ESp,EPp,ETp,LAI
      CHARACTER*10 DYN

      REAL CN, SWFAC1, SWFAC2, POTINF
      REAL SWC_INIT, TRAIN, TIRR, TESA, TEPA, TROF, TDRN
      REAL TINF, SWC_ADJ

!***********************************************************************
!***********************************************************************
!     INITIALIZATION
!***********************************************************************
      IF (INDEX(DYN,'INITIAL') .NE. 0) THEN
!***********************************************************************
        OPEN(3,FILE='SOIL.INP',STATUS='UNKNOWN')
        OPEN(10,FILE='sw.out',  STATUS='REPLACE')
        OPEN(11,FILE='IRRIG.INP',STATUS='UNKNOWN')
```

```
!***********************************************************************
!     INITIALIZATION
!***********************************************************************
      IF (INDEX(DYN,'INITIAL') .NE. 0) THEN
!***********************************************************************
        OPEN(3,FILE='SOIL.INP',STATUS='UNKNOWN')
        OPEN(10,FILE='sw.out',  STATUS='REPLACE')
        OPEN(11,FILE='IRRIG.INP',STATUS='UNKNOWN')

        READ(3,10) WPp,FCp,STp,DP,DRNp,CN,SWC
   10   FORMAT(5X,F5.2,5X,F5.2,5X,F5.2,5X,F7.2,5X,F5.2,5X,F5.2,5X,F5.2)
        CLOSE(3)
```

SIMCTRL.INP | Soil.inp

WPp	FCp	STp	DP	DRNp	CN	SWC
0.06	0.17	0.28	145.000	0.10	55.00	246.5
(cm3/cm3)	(cm3/cm3)	(cm3/cm3)	(cm)	(frac/d)	-	(mm)

Table 2 – Input data read for soil water balance module

Variable Name	Definition	Units
CN	Runoff curve number	--
DP	Depth of soil profile	cm
DRNp	Daily drainage percentage (fraction of void space)	1/day
FCp	Soil water content at field capacity (fraction of void space)	cm^3/cm^3
STp	Soil water content saturation (fraction of void space)	cm^3/cm^3
SWC	Soil water content in the profile (value read from file represents initial soil water content)	mm
WPp	Soil water content at wilting point (fraction of void space)	cm^3/cm^3

Fortran FORMAT descriptors

Purpose		*Edit Descriptors*	
Reading/writing **INTEGER**s		**I***w*	**I***w.m*
Reading/writing **REAL**s	Decimal form	**F***w.d*	
	Exponential form	**E***w.d*	**E***w.d***E***e*
	Scientific form	**ES***w.d*	**ES***w.d***E***e*
	Engineering form	**EN***w.d*	**EN***w.d***E***e*
Reading/writing **LOGICAL**s		**L***w*	
Reading/writing **CHARACTER**s		**A**	**A***w*
Positioning	Horizontal	***n*X**	
	Tabbing	**T***c*	**TL***c* and **TR***c*
	Vertical	/	
Others	Grouping	*r*(....)	
	Format Scanning Control	:	
	Sign Control	**S**, **SP** and **SS**	
	Blank Control	**BN** and **BZ**	

- *w*: the number of positions to be used
- *m*: the minimum number of positions to be used
- *d*: the number of digits to the right of the decimal point
- *e*: the number of digits in the exponent part

```
!***********************************************************************
!     INITIALIZATION
!***********************************************************************
      IF (INDEX(DYN,'INITIAL') .NE. 0) THEN
!***********************************************************************
        OPEN(3,FILE='SOIL.INP',STATUS='UNKNOWN')
        OPEN(10,FILE='sw.out',  STATUS='REPLACE')
        OPEN(11,FILE='IRRIG.INP',STATUS='UNKNOWN')

        READ(3,10) WPp,FCp,STp,DP,DRNp,CN,SWC
   10   FORMAT(5X,F5.2,5X,F5.2,5X,F5.2,5X,F7.2,5X,F5.2,5X,F5.2,5X,F5.2)
        CLOSE(3)
```

F77	F90	Math
.EQ.	==	=
.NE.	<>	≠
.LT.	<	<
.LE.	<=	≤
.GT.	>	>
.GE.	>	≥

```
IF (logical-expression-1) THEN
   statements-1
ELSE IF (logical-expression-2) THEN
   statements-2
ELSE IF (logical-expression-3) THEN
   statement-3
ELSE IF (.....) THEN
   ...........
ELSE
   statements-ELSE
END IF
```

```
IF (x > 0) THEN
   WRITE(*,*)  '+'
ELSE IF (x == 0) THEN
   WRITE(*,*)  '0'
ELSE
   WRITE(*,*)  '-'
END IF
```

```
IF
   (...)
ELSEIF
   (...)
ELSEIF
   (...)
ELSE
   (...)
ENDIF
```

```
     &     LAI,                                                    !Output
     &     'INITIAL   ')                                           !Control

!-----------------------------------------------------------------------
!     DAILY TIME LOOP
!-----------------------------------------------------------------------
      DO 500 DOY = 0,1000
        IF (DOY .NE. 0) THEN

          CALL WEATHR(SRAD,TMAX,TMIN,RAIN,PAR,'RATE      ')


!***********************************************************************
!***********************************************************************
!     RATE CALCULATIONS
!***********************************************************************
          CALL SW(
     &      DOY, LAI, RAIN, SRAD, TMAX, TMIN,                    !Input
     &      SWFAC1, SWFAC2,                                      !Output
     &      'RATE      ')                                        !Control
```

DO
(...)
END DO

```
      DO 10 N = 1, 100
         ...
10    END DO
```

Example 2: A `DO` loop without statement number:

```
      DO N = 1, 100
      ...
      END DO
```

```
!***********************************************************************
!     RATE CALCULATIONS
!***********************************************************************
      ELSEIF (INDEX(DYN,'RATE') .NE. 0) THEN
!***********************************************************************
        READ(11,25) DATE, IRR
   25   FORMAT(I5,2X,F4.1)

        TIRR = TIRR + IRR
        POTINF = RAIN + IRR
        TRAIN  = TRAIN + RAIN
        CALL DRAINE(SWC, FC, DRNp, DRN)

        IF (POTINF. GT. 0.0) THEN
          CALL RUNOFF(POTINF, CN, ROF, 'RATE      ')
          INF = POTINF - ROF
        ELSE
          ROF = 0.0
          INF = 0.0
        ENDIF

!     Potential evapotranspiration (ETp), soil evaporation (ESp) and
!       plant transpiration (EPp)
        CALL ETpS(SRAD,TMAX,TMIN,LAI,ETp)
        ESp = ETp * EXP(-0.7 * LAI)
        EPp = ETp * (1 - EXP(-0.7 * LAI))
```

```
*     Subroutine ETpS
*     Calculates the daily potential evapotranspiration.
!-----------------------------------------------------------------------
*     Input:  LAI, TMAX, TMIN, SRAD
*     Output: ETp
!***********************************************************************
C
*     Local Variables
*     ALB   =  ALBEDO OF CROP-SOIL SURFACE
*     EEQ   =  EQUILIBRIUM EVAPOTRANSPIRATION (mm)
*     Tmed  =  ESTIMATED AVERAGE DAILY TEMPERATURE (C)
*     f     =

!-----------------------------------------------------------
      SUBROUTINE ETpS(SRAD,TMAX,TMIN,LAI,ETp)

!-----------------------------------------------------------
      IMPLICIT NONE
      SAVE
      REAL    ALB,EEQ,f,Tmed,LAI
      REAL TMAX, TMIN, SRAD, ETP

!-----------------------------------------------------------------------
      ALB =  0.1 * EXP(-0.7 * LAI) + 0.2 * (1 - EXP(-0.7 * LAI))
      Tmed = 0.6 * TMAX + 0.4 * TMIN
      EEQ = SRAD * (4.88E-03 - 4.37E-03 * ALB) * (Tmed + 29)

      IF (TMAX .LT. 5) THEN
        f = 0.01 * EXP(0.18 *(TMAX + 20))
      ELSEIF (TMAX .GT. 35) THEN
        f = 1.1 + 0.05 * (TMAX - 35)
      ELSE
        f = 1.1
      ENDIF

      ETp = f * EEQ
!-----------------------------------------------------------------------
      RETURN
      END SUBROUTINE ETPS
!***********************************************************************
```

Subroutine ETpS calculates the daily potential evapotranspiration rate (ETp) based on the Priestly-Taylor method. The surface albedo (ALB) is estimated as a weighted average (based on LAI) of the albedo of the soil (0.1) and crop (0.2).

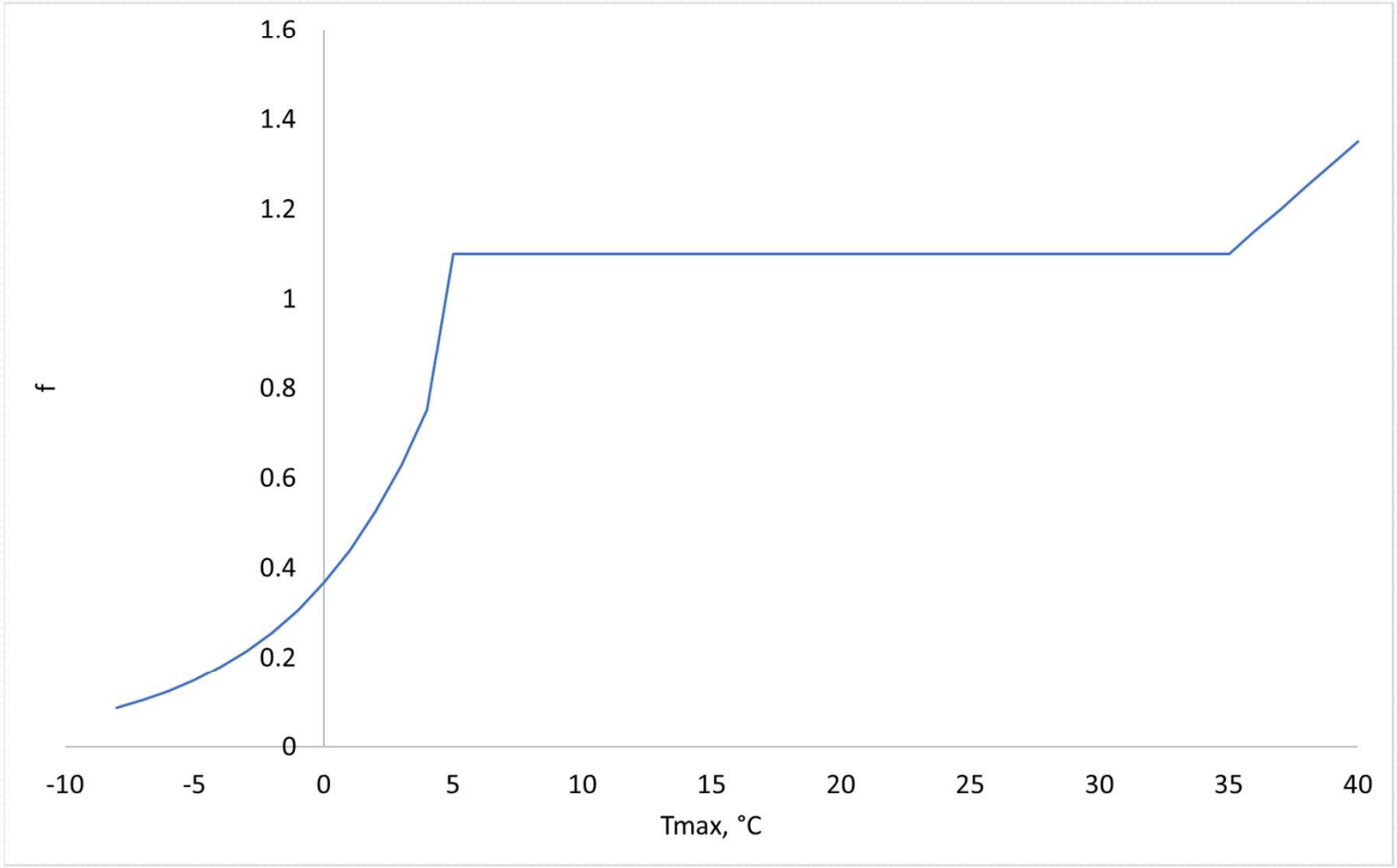
```
ALB =  0.1 * EXP(-0.7 * LAI) + 0.2 * (1 - EXP(-0.7 * LAI))
```

The average temperature during the day (T) and the equilibrium evaporation (EEQ) are calculated.

```
Tmed = 0.6 * TMAX + 0.4 * TMIN
EEQ = SRAD * (4.88E-03 - 4.37E-03 * ALB) * (Tmed + 29)
```

The equilibrium evaporation rate is adjusted by a coefficient (f) resulting in the final value of ETp.

$f(T_{max})$


1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
f
-10
-5
0
5
10
15
20
25
30
35
40
Tmax, °C

```
!     Potential evapotranspiration (ETp), soil evaporation (ESp) and
!       plant transpiration (EPp)
        CALL ETpS(SRAD,TMAX,TMIN,LAI,ETp)
        ESp = ETp * EXP(-0.7 * LAI)
        EPp = ETp * (1 - EXP(-0.7 * LAI))

!     Actual soil evaporation (ESa), plant transpiration (EPa)
        CALL ESaS(ESp,SWC,FC,WP,ESa)
        EPa = EPp * MIN(SWFAC1, SWFAC2)

!***********************************************************************
!***********************************************************************
!     INTEGRATION
!***********************************************************************
      ELSEIF (INDEX(DYN,'INTEG') .NE. 0) THEN
!***********************************************************************
```

```
!***********************************************************************
*     Subroutine ESaS
*     Calculates the actual daily soil evaporation.
!-----------------------------------------------------------------------
*     Input:  SWC, WP, FC, ESp
*     Output: ESa
!***********************************************************************

      SUBROUTINE ESaS(ESp,SWC,FC,WP,ESa)

!-----------------------------------------------------------------------
      IMPLICIT NONE
      SAVE
      REAL a, SWC, WP, FC, ESa, ESp
!-----------------------------------------------------------------------
      IF (SWC .LT. WP) THEN
        a = 0
      ELSEIF (SWC .GT. FC) THEN
        a = 1
      ELSE
        a = (SWC - WP)/(FC - WP)
      ENDIF

      ESa = ESp * a

!-----------------------------------------------------------------------
      RETURN
      END SUBROUTINE ESAS
!***********************************************************************
```

```
!     Potential evapotranspiration (ETp), soil evaporation (ESp) and
!       plant transpiration (EPp)
        CALL ETpS(SRAD,TMAX,TMIN,LAI,ETp)
        ESp = ETp * EXP(-0.7 * LAI)
        EPp = ETp * (1 - EXP(-0.7 * LAI))

!     Actual soil evaporation (ESa), plant transpiration (EPa)
        CALL ESaS(ESp,SWC,FC,WP,ESa)
        EPa = EPp * MIN(SWFAC1, SWFAC2)

!***********************************************************************
!***********************************************************************
!     INTEGRATION
!***********************************************************************
      ELSEIF (INDEX(DYN,'INTEG') .NE. 0) THEN
!***********************************************************************
```

Como a transpiração real (EPa) é calculada?

Descubra como são calculadas as duas varáveis SWFAC1 e SWFAC2