SOILTEMPERATURE

*General description*

Simulates soil temperature given minimal input information using a numerical scheme.

*Usage*

Soiltemp requires water content information from a water module, air temperature from the input module, and where available will use soil water evaporation and incident net radiation as part of the upper boundary condition. It should be placed after the water module in the module control list.

*Theory*

The node/element scheme for the numerical simulation is shown in Figure 1 where the circles denote the nodes and the zigzag’s show the elements. All heat storage is assumed to occur at the nodes and all resistance to heat transfer is assumed to take place within the elements.



Figure 1. A diagrammatic representation of the node structure of the numerical simulation. T is temperature, Z is depth, K is thermal conductance, S is heat storage, nz is the number of nodes in the simulation, BL stands for boundary layer, Ta is the air temperature, and Tave is the annual average soil temperature.

Webegin by defining the heat balance at a node i, where i ≠ 1 or nz. The decrease in storage of heat at the node over a particular timestep is equal to the amount of heat leaving the node minus the amount of heat coming in all divided by the timestep so that,

, [1]

where Δ stands for a change and *t* is time (s). Storage of heat is calculated from the change in temperature with time and the volumetric specific heat of the soil so that,

, [2]

where *i* stands for a particular node, *j* refers to the current timestep and j+1 to the next timestep, *T* is temperature (°C), *C* is the volumetric specific heat of the soil (J K-1 m-3), *A* is the cross-sectional area and is normally taken as 1 m2, and *z* is depth (m). Because all heat storage is assumed to occur at the nodes, the appropriate depth interval is half the distance to the node on either side of *i*. *C* can be calculated from the specific heats of the soil constituents so that,

, [3]

where ρ is the soil bulk density (Mg m-3), ρs is the particle density and is usually taken as 2.65 Mg m-3, θ is the volumetric water content (m3 m-3), the subscripts on *C* indicate the substance. *Cair* is usually assumed to be negligible while appropriate values of C for clay minerals and water are 2.39 and 4.18 MJ m-3 K-1. Where more precise values are required, the equation can be expanded to account for organic matter and other minerals. See Campbell (1985) or Jury *et al.* (1991) for details on how to calculate C in more detail.

Heat transfer is calculated from the standard Fickian equation where the rate of transfer is proportional to the temperature difference divided by the distance,

, [4]

where λ is the thermal conductivity (J s-1 m-1 K-1) and the overbar indicates that an appropriate average in time of temperature should be used. λ can either be calculated from measurements of temperature changes in the soil (see Jury *et al.*, 1991 for an example) or can be approximated from regression equations. Below is given the scheme from Campbell (1985). First four parameters dependent on density, water content, and the amount of clay are calculated. These are,

, [5]

where *fclay* is the fraction of clay in the soil. These parameters are combined into the equation for λ so that,

. [6]

We are now able to put the elements of the equation together.

. [7]

Now, in order to make some notational simplifications, define

 , [8]

and

, [9]

and multiply by -1, so that

. [10]

The average temperature is defined as

, [11]

where η is an operator controlling the forward/backward averaging of temperature. Expanding the heat balance equation,

, [12]

and collecting on the terms of T,

, [13]

we can now put the unknown, or ‘future’, terms of T on the left hand side,

, [14]

the equation can now be expressed in matrix format.

 [15]

If the three elements of the left-most matrix are expressed as ai, bi, and ci, and the right hand side as di, the equation can be expanded as an example for a simulation of nz nodes,

. [16]

We now have a tri-diagonal matrix which can be solved using the Thomas algorithm (Carnahan *et al.*, 1969).

The lower boundary condition is taken as a zero flux condition, ie. constant temperature, in which Tnz+1 equals a constant which is usually taken as the average annual temperature.

 [17]

The upper boundary condition is more complex. Usually the known temperature is the air temperature at some height above the soil and d1 is expressed as,

, [18]

where KBL is the boundary layer conductance (J s-1 m-2 K-1), Tair is the temperature at the top of the boundary layer, Rn is the net radiative input (J s-1 m-2), and Esoil is the evaporative loss of energy from the soil surface (J s-1 m-2).

*Implementation*

Soiltemp is designed to independent of the Apsim timestep. To allow for the numerical solution, the equations above are solved 48 times within each Apsim timestep. This allows for a half-hourly internal time step when Apsim is running on its customary daily step, which should be more than sufficient for numerical stability. When the Apsim timestep is 24 hours Soiltemp estimates the changes in air temperature, the upper boundary condition, in the following manner.



Figure 2. Diagram showing the interpolation of air temperature within a day based on minimum and maximum temperature.

Air temperatures occurring between midnight and mint\_time are linearly interpolated from the air temperature at midnight, calculated at the end of the previous Apsim timestep, and mint. There is a linear rise in temperture from the day’s minimum to maximum. After maxt\_time until midnight, air temperature is calculated as decreasing at the same rate at which it rose.

If the Apsim timestep is less than 24 hours it is assumed that the user is supplying enough detail of the diurnal changes in air temperature that such interpolation is not required. In that case air temperture is taken as the average of mint and maxt for each Apsim timestep.

*Initialisation*

There are two sections required for initialisation of the Soiltemp module; constants and parameters. Where a variable name is followed by “[nz]” the variable is an array and the appropriate number of values must be supplied.

Constants

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Unit | Description | Value |
| nu | - | forward/backward differencing | 0.6 |
| vol\_spec\_heat\_om | J m-3 K-1 | volumetric specific heat of organic matter | 5.00e6 |
| vol\_spec\_heat\_water | J m-3 K-1 | volumetric specific heat of water | 4.18e6 |
| vol\_spec\_heat\_clay | J m-3 K-1 | volumetric specific heat of clay minerals | 2.39e6 |

Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Unit | Description | Range |
| clay[nz] | - | proportion of clay | 0.0 - 1.0 |
| bound\_layer\_cond | J s-1 m-2 K-1 | boundary layer conductance | 0.0 - 100.0 |

The higher the value of bound\_layer\_cond the greater the difference between air and soil surface temperature. If its value is unknown, Campbell (1986) suggests that a value of 20 J s-1 m-2 K-1 is an appropriate initial estimate.

A further, optional, parameter is,

|  |  |  |  |
| --- | --- | --- | --- |
| soil\_temp[nz] | °C | initial soil temperature | -100.0 - 100.0 |

which used to initialise soil temperature. If it is not supplied the soil temperature array is initialised to the average annual temperature. Simulations will eventually ‘forget’ the effect of poor initial guesses of soil temperature, but this may take some time. Testing of this module showed that it took approximately 40 days for the temperature at 1.5 m deep to converge to within 0.5 °C of the analytical solution when the initial temperature difference was 7 °C. The discrepancy will be greatest deeper in the soil profile, and where *C* is high or λ is low. In general, where soil temperature is only important in the soil surface layers the convergence will occur within the first 10 days or so.

Where the time taken to ‘forget’ the initial conditions might cause significant error, there are two strategies for overcoming this problem. The first is to run a dummy simulation prior to the start of the real simulation to estimate the starting soil temperature. The second option is to estimate the initial soil temperatures from an analytical solution. A solution for the heat flow equation assuming a sinusoidal upper boundary condition. which might for example be the annual cycle in air temperature, is (Carslaw and Jaeger, 1959),

, [19]

where

, [20]

Tave is the average annual temperature (°C), Tamp is the annual amplitude in temperature (°C), and ω is the angular frequency (radians). Thermal conductivity and heat capacity can be estimated, using soil profile averages, from equations 3, 5, and 6.

*Time step inputs from other modules*

Soiltemp must be accompanied by the input module and a soil water module in order that other inputs are supplied. These inputs are.

|  |  |  |
| --- | --- | --- |
| Name | Unit | Description |
| Variables from the Input module | | |
| temp\_average\_annual | °C | Average annual temperature, used to set the initial soil temperature if soil\_temp is not supplied. Also determines the temperature at the lower boundary. |
| timestep | min | Simulation timestep, converted to seconds internally. |
| mint\_time | hours | Specifies the hour of the day when the minimum air temperature occurs. |
| maxt\_time | hours | Specifies the hour of the day when the maximum air temperature occurs. |
| mint | °C | Minimum air temperature. |
| maxt | °C | Maximum air temperature. |
| Variables from the soil water module | | |
| dlayer[nz] | mm | Array of layer depths used to specify the nodes, converted to m internally. |
| sw[nz] | m3 m-3 | Volumetric soil water content. |
| bd[nz] | Mg m-3 | Soil bulk density. |
| eo | mm | Potential soil water evaporation, or the water-depth equivalent of the net radiation reaching the soil surface, converted to J s-1 m-2 K-1 internally. |
| es | mm | Actual soil water evaporation, or the water-depth equivalent of the evaporation from the soil surface, converted to J s-1 m-2 K-1 internally. |

To allow compatibility with modules where changes in the soil occur with time, dlayer and bd are requested at every Apsim time step. If eo and es are not available they are assumed equal to zero.

*Time step outputs*

|  |  |  |
| --- | --- | --- |
| Name | Unit | Description |
| final\_soil\_temp[nz] | °C | Soil temperature at the end of the final internal Soiltemp timestep. |
| soil\_temp[nz] | °C | Average soil temperature during the Apsim timestep. |
| mint\_soil[nz] | °C | Minimum soil temperature found in each layer during the Apsim timestep. |
| maxt\_soil[nz] | °C | Maximum soil temperature found in each layer during the Apsim timestep. |
| therm\_cond[nz] | J s-1 m-1 K-1 | Thermal conductivity for each layer. |
| heat\_store[nz] | J m-3 K-1 | Volumetric specific heat for each layer. |
| airt | °C | Average air temperature during the Apsim timestep. |

*References*

Campbell G.S., 1985, *Soil Physics with Basic*, Elsevier, Amsterdam.

Carnahan B., Luther H.A., Wilkes J.O., 1969, *Applied Numerical Methods*, 1st Ed. Wiley, New York.

Carslaw H.S., Jaeger J.C., 1959, *Conduction of Heat in Solids*, Oxford University Press, London.

Jury W.A., Gardner W.R., Gardner W.H., 1991, *Soil Physics*, 5th Ed. Wiley, New York.

*Appendix*

Fortran code for calculating the coefficients for the Thomas algorithm, the apgorithm, and computing the updated temperatures.

!the boundary layer conductance, therm(0) does not need divide z

do i = 1,nz

!rate of heat accumulation

heat(i) = heat\_store(i) \* 0.5 \* (z(i+1) - z(i-1)) / dt

!convert to thermal conductivity to thermal conductance

therm(i) = therm\_cond(i) / (z(i+1)-z(i))

enddo

!top node, net\_rad\_soil is net radiation input to the soil surface

!evap\_soil is latent heat loss from the soil, both in J/s/m2

a(1) = 0.0

b(1) = nu \* therm(1)

: + nu \* therm(0)

: + heat(1)

c(1) = -nu \* therm(1)

d(1) = t(0) \* (1-nu) \* therm(0)

: - t(1) \* (1-nu) \* therm(1)

: - t(1) \* (1-nu) \* therm(0)

: + t(1) \* heat(1)

: + t(2) \* (1-nu) \* therm(1)

: + therm(0)\* tn(0) \* nu

: - net\_rad\_soil

: + evap\_soil

!the main loop

do i=2,nz-1

a(i) = -nu \* therm(i-1)

b(i) = nu \* therm(i)

: + nu \* therm(i-1)

: + heat(i)

c(i) = -nu \* therm(i)

d(i) = t(i-1) \* (1-nu) \* therm(i-1)

: - t(i) \* (1-nu) \* therm(i)

: - t(i) \* (1-nu) \* therm(i-1)

: + t(i) \* heat(i)

: + t(i+1) \* (1-nu) \* therm(i)

enddo

!lower node

a(nz) = -nu \* therm(nz-1)

b(nz) = nu \* therm(nz)

: + nu \* therm(nz-1)

: + heat(nz)

c(nz) = 0.0

d(nz) = t(nz-1) \* (1-nu) \* therm(nz-1)

: - t(nz) \* (1-nu) \* therm(nz)

: - t(nz) \* (1-nu) \* therm(nz-1)

: + t(nz) \* heat(nz)

: + t(nz+1) \* (1-nu) \* therm(nz)

: + therm(nz) \* nu \* tn(nz+1)

! the Thomas algorithm

do i=1,nz-1

c(i)=c(i)/b(i)

d(i)=d(i)/b(i)

b(i+1)=b(i+1)-a(i+1)\*c(i)

d(i+1)=d(i+1)-a(i+1)\*d(i)

enddo

! Calculate the new temperatures

tn(nz)=d(nz)/b(nz)

do i = nz-1,1,-1

tn(i)=d(i)-c(i)\*tn(i+1)

enddo