A Water Balance Model for Agent Based Modelling

FJ Cook, GD Carlin, MG Hartcher, X Su, S Asseng and P Campbell

CSIRO Land and Water Science Report 47/08
June 2008

CSIRO Land and Water Science Report series ISSN: 1834-6618

Water Balance Model for Agent Based Modelling

FJ Cook[1], GD Carlin[1], M Hartcher[1], S Xianfeng[2], S Asseng[2] and AP Campbell[3]

[1] CSIRO Land and Water

[2] CSIRO Plant Industry

[3] University of South Australia

CSIRO Land and Water Science Report 47/08

June 2008

Acknowledgements

Funding for this project was provided from CSIRO Complex Systems Initiative. Dr John Finnigan and the core group at the Complex Systems Group of CSIRO are thanked for their support throughout this project.

Executive Summary

Many Agent Based Modelling platforms like RePast, Cormas, Mason, MATLAB, NetLogo, DIAS, etc offer a framework for creating ABMs (Agent Based Models). ABMs are commonly used to create complex models by integrating well understood scientific models and at a smaller scale or within a particular scientific discipline to create a model that may combine biophysical, social and economic models. This approach is becoming increasing popular and improved methods for understanding the biophysical interactions of humans, animals and the environment is of growing concern. The model developed here is a relatively standard water balance model that has been included in a two agent-based models and is intended to become a module for inclusion in other models.

The model described here is a cascading bucket model that incorporates the force/restore method of Deardorff (1977), to improved evaporation of water from the soil. This requires that the first layer is a surface layer nested within layer 2. Layer 1, the surface layer, is required because soil evaporation is often underestimated (Deardoff, 1977). Deardoff (1977) devised a force-restore model often used in heat transport modelling for soil evaporation. This has been used here with some adaptation. There is a transfer of water to or from the surface layer, from or to layer 2 by an exchange coefficient. In Deardorff's model this exchange coefficient (C_1) is an empirical parameter. The minimum number of layers for this model is 2 and the model assumes the soil properties for layer 1 and 2 are the same.

The runoff generated by the model at this scale can then be summed over a defined area within the modelled spatial area. If we look at a catchment approach for using this model then the spatial area would be the catchment, which could be further segmented into subcatchments. Appropriate number of contour segments based on digital elevation models (DEM) are then defined for the catchment or subcatchment (Carlin et al., 2007). Within these contour segments, runoff is then summed for each of the modelled areas that lie within the contour segment. The number of contours required depends on the spatial size of the modelled areas as each contour should a have reasonable number of representative modelled areas.

Contents

1. Introduction

Many Agent Based Modelling platforms like RePast, Cormas, Mason, MATLAB, NetLogo, DIAS, etc offer a framework for creating ABM's (Agent Based Models). ABMs are commonly used to create complex models by integrating well understood scientific models and at a smaller scale or within a particular scientific discipline to create a model that may combine biophysical, social and economic models (Su et al. 2008). This approach is becoming increasing popular and improved methods for understanding the biophysical interactions of humans, animals and the environment is of growing concern (Fath, 2004;Favier et al., 2004;Teegavarapau, 2007). Complex systems models were developed to combine biophysical, social and economic components (Smagjl et al., 2007; Su et al. 2008). These models required the biophysical components in order to perform. They have paddock agents as the minimal spatial and management unit within them and as such require biophysical descriptors of these paddock agents. Here we describe the water balance agent that was developed for the paddock agents.

The model used is a cascading bucket model (Fig 1.) with modifications to allow for improved soil evaporation. This modification follows Deardorff (1977) and uses a force/restore method to better determine soil evaporative losses. This requires that the first layer is a surface layer nested within layer 2. Layer 1, the surface layer, is required because soil evaporation is often underestimated (Deardoff, 1977). Deardoff (1977) devised a force-restore model often used in heat transport modelling for soil evaporation. This has been used here with some adaptation. There is a transfer of water to or from the surface layer, from or to layer 2 by an exchange coefficient. In Deardorff's model this exchange coefficient (C_1) is an empirical parameter. The minimum number of layers for this model is 2 and the model assumes the soil properties for layer 1 and 2 are the same. The developments presented here follow those of Cook et al. (2003).

Capacity or tipping bucket models are approximations of the flow processes that are taking place in the soil which are described in a continuous manner by the Richards equation (Hillel, 1980). However, Richards equation is considered to be inappropriate when the scale is greater that about 10 m^2 ((Addiscott et al., 1995). This is due to the inability to parameterize and to test at scales larger than this.

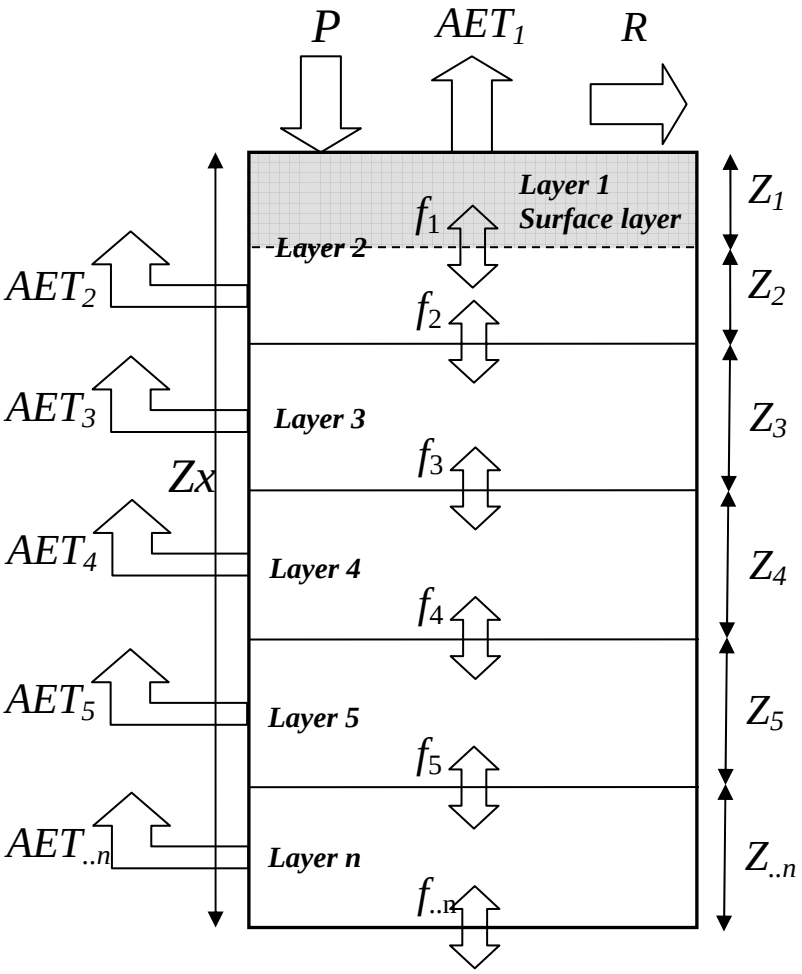

Figure 1. Schematic diagram of cascading bucket model. Block arrows represent material fluxes of water. First layer is a surface layer nested within layer 2. AET_1 is the actual evapotranspiration from the surface layer and $AET_{2..n}$ is evapotranspiration from all other layers where plant roots extend into. P is the precipitation and R is the surface runoff, $f_{1..n}$ is the flux between the layers, $z_{1..n}$ is the depth of the layers and z_x is the depth of all layers (bulk layer). $F_{1..n}$ is the flux between layers or deep drainage depending on the number of layers.

The time step used in the model is daily. This was considered the smallest practical time step for this model, as data are rarely available at time scales less than this.

2. Infiltration and Runoff

Infiltration of water into the soil can be restricted by two processes, lack of soil storage capacity due to saturation of the profile, and the rainfall rate being greater than the infiltration rate. Both processes are included in this modeling framework. When the infiltration rate is less than the rainfall rate, runoff is generated.

The model first calculates the infiltration storage capacity of the soil profile and potential infiltration rate of the soil on the day of interest. The infiltration storage capacity (Ic_j) on day j for depth zi is calculated by:

$$Ic_j = \sum_{i=1}^{n} (\theta_{si} - \theta_{i,j}) z_i \tag{1}$$

where θ_{si} is the saturated water content of layer i and $\theta_{I,j}$ is the water content of the layer on day j.

The potential infiltration rate (Im_j) is calculated as follows. The infiltration capacity is calculated using the constant diffusivity (D^*). The value of D^* can be approximated for a linear soil by (Philip, 1969):

$$D^* = \frac{\pi S^2}{4\Delta\theta^2} \tag{2}$$

where S is the measured sorptivity [M $T^{-1/2}$] (Philip, 1957), $\Delta\theta = \theta_f - \theta_n$ is the difference in the volumetric water content [L^3 L^{-3}]; θ_f is the final and θ_n is the initial water content. The use of a constant diffusivity when the soil is wet was found to be valid by Clothier and White (1981). The sorptivity of the upper layer (So_j) can be given by (Philip, 1969):

$$So_j = \sqrt{4D^* \left(\theta_{s1} - \theta_{1,j}\right)^2 / \pi} \tag{3}$$

where θ_{s1} and $\theta_{1,j}$ are the saturated volumetric water content and volumetric water content of layer 1on day j. The two term Philip infiltration (Philip, 1957) equation is valid until the gravitational time tg_j (Philip, 1987):

$$tg_j = \left(So_j / \left[\left(1 - 0.36\right) K_{s1}\right]\right)^2 \tag{4}$$

where K_{s1} is the saturated hydraulic conductivity of layer 1 [L T^{-1}], which is when gravitational forces dominate the flow process. The infiltration rate for time > tg_j can be obtained by splicing solutions together (Philip, 1987). The constant 0.36 in Eqn (4) is for a soil with Burger's equation properties which is close to what real soils exhibit (Philip, 1987). Now since the time step in the model is 1 day then (after Philip,1987):

$$\begin{aligned} I_m &= S_o + 0.36K_{s1},\ tg \le 1 \\ &= S_o\sqrt{tg} + K_{s1}\left[1 - tg\left(1 - 0.36\right)\right],\ tg > 1 \end{aligned} \tag{5}$$

where I_m is the infiltration rate capacity (m day^{-1}).

The limitation on infiltration (I_l) is determined by:

$$I_l = \min(I_c, I_m) \tag{6}$$

Any runoff (*Ro*) that may be generated on this paddock and infiltration (*I*) on this day are calculated by:

$$\begin{aligned} I &= R, \quad Ro = 0, \qquad R \le I_l \\ I &= I_l, \quad Ro = R - I_l, \quad R > I_l \end{aligned} \tag{7}$$

where R is the rainfall rate (m day^{-1}).

3. Transfer between Layers

There are two processes that are used in this model to transfer water between the layers. The first is a downward transport of water due to gravity when the water content of the layer is greater than field capacity. The second occurs due to the force-restore process in the upper layer (layers 1 and 2).

For the first layer the water balance model is based on that of Deardorff (1977) and uses a force/restore method to better determine soil evaporative losses. This model consists of two layers, a surface and a bulk layer (Fig. 1). The surface layer is nested inside and is part of layer 2. There is a transfer of water to or from the surface layer from or to layer 2 by an exchange coefficient. In Deardorff's model this exchange coefficient (C_1) is an empirical parameter. Here it will be defined based on the soil water diffusivity and is approximated by:

$$f_1 = D^* \frac{2(\theta_2 - \theta_1)}{(z_2 - z_1)} - k \tag{8}$$

where θ_1 is the volumetric water content of the surface layer [L^3 L^{-3}] and θ_2 is the volumetric water content of layer 2 [L^3 L^{-3}], z_2 is the depth of soil layer 2 [L], z_1 is the depth of the surface soil layer [L], k is the hydraulic conductivity [L T^{-1}] and D* is the constant soil water diffusivity [L^2 T^{-1}] (Philip, 1969). The value of D* can be approximated for a linear soil by (Philip, 1969):

$$D^* = \frac{\pi S^2}{4\Delta\theta^2} \tag{9}$$

where S is the sorptivity [M T$^{-1/2}$] (Philip, 1957), $\Delta\theta = \theta_f - \theta_i$ is the difference in the volumetric water content [L^3 L^{-3}], θ_f is the final and θ_i is the initial water content. The use of a constant diffusivity when the soil is wet was found to be valid by Clothier and White (1981).

$$
\begin{aligned}
f_1 &= \min\left[D^* \frac{2(\theta_2 - \theta_1)}{(z_2 - z_1)}, (\theta_1 - \theta_{PWP1}).z_1 \right], \qquad \theta_1 < \theta_{FC1} \\
&= \max\left[K_{s1}.(\theta_1 - \theta_{FC1})/(\theta_1 - \theta_{FC1}), (\theta_1 - \theta_{FC1}).(z_2 - z_1)\right], \quad \theta_1 \geq \theta_{FC1}
\end{aligned}
\tag{10}
$$

Where θ_{PWP1} is the permanent wilting point water content for layers (1 & 2)

The transfer between layers occurs in the following sequence. The gravitational flow of water only occurs when the soil is wetter than field capacity (θ_{FC}). A value of the transfer rate is needed and we use a scheme as outlined below.

It can also be shown for a linear soil that (Philip, 1969, Eqn (127)):

$$
\frac{dk}{d\theta} = const = k^* \tag{11}
$$

From Eqn (8), assuming that at field capacity (θ_{FC}) that $k(\theta_{FC}) \approx 0$ the value of k^* is given by:

$$
k^* = K_s / (\theta_s - \theta_{FC}) \tag{12}
$$

where K_s is the saturated hydraulic conductivity [L T^{-1}] and θ_s is the saturated water content [L^3 L^{-3}]. The value of k is then calculated using:

$$
\begin{aligned}
k(\theta) &= K_s \frac{(\theta - \theta_{FC})}{(\theta_s - \theta_{FC})}, \qquad \theta > \theta_{FC} \\
&= 0, \qquad \theta \leq \theta_{FC}
\end{aligned}
\tag{13}
$$

The flux between the layers is calculated by Eqn (10) multiplied by the time step. If the units of K_s are m day^{-1} then as the time step is 1 day Eqn (10) becomes the transfer rate. We can obviously not transfer more water of the layer than is available (θ_1 - θ_{FCi}) to be transferred, so the flux ($f_{i,i+1}$) is determined by:

$$
\begin{aligned}
f_{i,i+1} &= \min\left[K_{si}.(\theta_i - \theta_{FCi})/(\theta_i - \theta_{FCi}), (\theta_i - \theta_{FCi}).(z_i - z_{i-1}) \right] \\
&= \min\left[2D_i(\theta_{i+1} - \theta_i)/(z_{i+1} - z_{i-1}), (\theta_{i\_1} - \theta_{FCi+1}).(z_{i+1} - z_i) \right], WT > z_{i+1}, i < n \\
&= 0, \qquad WT > z_i
\end{aligned}
\tag{14}
$$

where θ_i, θ_{FCi} is the water content and field capacity for the *i*th layer, K_{si} is the saturated hydraulic conductivity for the *i*th layer, z_i and z_{i+1} are the depth of the *i*th and *i*+1 layers respectively and *WT* is the water table depth. The middle condition allows water transfer upward from the water table if a water table occurs in the layer below and restricts the transfer to only reducing the water content in this layer to the field capacity value for this layer. Justification for the form of this middle condition is given below. The latter condition in Eqn (11) does not allow downward transfer of water to occur if the water table is above the bottom of layer. If the water table depth is above the base of the layer then water accumulates within the layer.

When the last layer *n* is reached then this becomes:

$$\begin{aligned} f_n &= \min\left[K_{sn}.(\theta_n - \theta_{FCn})/(\theta_n - \theta_{FCn}), (\theta_n - \theta_{FCn}).(z_n - z_{n-1})\right],\ \theta_n > \theta_{FCn},\ (WT - z_n) > 3\lambda \\ &= \min\left[2D_i(\theta_{sn} - \theta_n)/(WT + (z_n - z_{n-1})/2), (\theta_{sn} - \theta_n).(z_n - z_{n-1})\right],\ \theta_n < \theta_{sn}(WT + (z_n - z_{n-1}) \\ &= 0, \theta_n \le \theta_{FCn}, (WT - z_n) > 3\lambda \end{aligned} \tag{15}$$

where λ is the macroscopic capillary length scale (White and Sully, 1987) [L] and can be calculated by:

$$\lambda = D_n^*(\theta_{sn} - \theta_{PWPn})/K_{sn} \tag{16}$$

for the properties already described here.

4. Evapotranspiration

Water is lost from the soil by evapotranspiration, drainage and runoff via the surface layer and further water loss from all other soil layers can occur via evapotranspiration, and drainage. In the original model of Deardorff's no surface runoff or deep drainage components were included as the model was only run for short periods when drying by evaporation (from a bare soil surface) was the dominant process. The modifications made here to include surface runoff and deep drainage in Deardorff's model where necessary as the model described here operates for a longer time period. Similar modifications to allow the model to run for longer time periods have also been made by Shao *et al*. (1996).

The loss or gain of water from the soil is via soil layers 2.to n as well as through the surface layer (layer 1). Evapotranspiration (*ET*) is first lost from layer 1 and then progressively from the next layer down until the potential evapotranspiration is reached or all other soil layers have provide what water they can at this time. The apportioning of the water loss is done as a two-part process. The total *ET* is calculated from the reference evapotranspiration (ET_o) and a crop factor (*Kc*). ET_o is calculated using the

Penman-Monteith equation for a well watered grass surface. This method is the current FAO standard (Smith *et* al., 1996). Further discussion on this method for sugarcane can be found in (Attard *et al*., 2003). The maximum value of *Kc* has been found to be 1.25 in a number of studies on sugar cane (Attard *et al*., 2003). *Kc* for other crops can be found in Smith *et al*. (1996).

The increase in the crop factor and maximum rooting depth increase with crop growth as follows (Monteith, 1986):

$$\begin{aligned} & t_j > (p+10),\ p > 0 \\ & Kc_j = \min\left[Km, Kc_{j-1} + A_c\left(t_j - (p+\phi)\right)\right], \\ & RD = \min\left[Rm, 0.15 + A_r\left(t_j - (p+\phi)\right)\right] \end{aligned} \tag{17}$$

where t_j is the day number for this year, p is the day no. on which the crop was planted and equals zero if no crop planted, ϕ is the time for the crop to emerge, Kc_j and Kc_{j-1} are the crop factor on day j and day I+1 respectively, A_c is the rate of crop factor expansion, which will vary with the crop, A_r is the rate of root expansion this will vary with crop and climate Monteith (1986) suggest an average rate of 35 mm day^{-1}, Km is the maximum crop factor and Rm is the maximum rooting depth for this crop, Kc_j is crop factor at time t_j.

The calculation of evapotranspiration (ET) then proceeds as follows. First, the potential evapotranspiration is calculated for day j.

$$PET_j = ET_{oj}.Kc_j \tag{18}$$

Then the available storage of each layer is calculated by:

$$T_{i,j} = \theta_{i,j} - \theta_{PWPi} \tag{19}$$

The evapotranspiration is then calculated sequentially as follows:

$$\begin{aligned} Et_{1,j} &= \min\left[PET_j, T_{1,j}.z_1 \right], && T_{1,j} > AWC_1/2 \\ &= \min\left[0, PET_j\left(2T_{1,j}/AWC_1\right)\right], && T_{1,j} \le AWC_1/2 \\ AET_{1,j} &= Et_{1,j} \end{aligned} \tag{20}$$

This process for layer 1 is always executed. Evapotranspiration will occur from the next layer depending on whether there are any roots there or not. Soil evaporation from layer 2 occurs due to the restore sequence for layer 1 outlined above.

$$
\begin{aligned}
&Et_{i,j} = 0, \qquad RD \le z_{i-1} \\
&Et_{i,j} = \min\left[(PET_j - AET_{i-1,j}), T_{i,j}.(RD_j - z_{i-1})\right], \qquad T_{i,j} > AWC_i/2, \qquad z_{i-1} < RD_j < z_i \\
&Et_{i,j} = \min\left[(PET_j - AET_{i-1,j}).2T_{i,j}/AWC_i, T_{i,j}.(RD_j - z_{i-1})\right], T_i < AWC_i/2, \; z_{i-1} < RD_j < z_i \\
&Et_{i,j} = \min\left[(PET_j - AET_{i-1,j}), T_{i,j}.(z_i - z_{i-1})\right], T_{i,j} > AWC_i/2, \qquad RD_j \ge z_i \\
&Et_{i,j} = \min\left[(PET_j - AET_{i-1,j}).2T_{i,j}/AWC_i, T_{i,j}.(z_i - z_{i-1})\right], \quad T_{i,j} < AWC_i/2, \qquad RD_j \ge z_i \\
&AET_{i,j} = AET_{i-1,j-1} + Et_{i,j}
\end{aligned}
\tag{21}
$$

Equation (21) states that if there are no roots in layer *i* then no ET occurs from this layer (*i* th) on this day (*j* th), if the roots only occur to part of the depth of the layer then this limits the maximum we can remove from this layer, and if the amount of water in the layer is less than half of AWC_i then the rate of evapotranspiration is reduced as a linear function. Actual evaporation is accumulated with progress down the layers and with time. The ratio of *AET* accumulated with depth is used to reduce the amount of potential evapotranspiration possible from the next layer.

5. Updating the Water Balance

The water balance for each layer is then updated and becomes the water content value for the next time step. This has one subtlety due to the surface layer.

We calculate the new storage value for the layer 1 on day *j* by:

$$
\begin{aligned}
&Sm_1 = (\theta_{s1} - \theta_{PWP1}).z_1 \\
&S_{1,j+1} = \min\left(\max\left[T_{1,j}.z_1 + I - Et_{1,j} - f_1, Sm_1\right], 0\right)
\end{aligned}
\tag{22}
$$

This will give either the minimum, maximum value or a value in between. Having updated the storage the new water content is also calculated.

$$
\theta_{j+1,1} = S_{1,j+1}/z_1 + \theta_{PWP1} \tag{23}
$$

For layer 2, as it contains layer 1, the procedure is:

$$
\begin{aligned}
&Sm_2 = (\theta_{s2} - \theta_{PWP2}).z_2 \\
&S_{2,j+1} = \min\left(\max\left[T_{2,j}.z_2 + I - Et_{2,j} - Et_{1,j} - f_{2,j}, Sm_2\right], 0\right) \\
&In_j = \max\left(I_j - \left[Sm_{2,j} - (\theta_{2,j} - \theta_{PWP2}).z_2\right], 0\right) \\
&\theta_{2,j+1} = S_{2,j+1}/z_2 + \theta_{PWP2}
\end{aligned}
\tag{24}
$$

Now we proceed to the lower layers and follows

$$
\begin{aligned}
Sm_i &= \left(\theta_{si} - \theta_{PWPi}\right).(z_i - z_{i-1}) \\
S_i &= \min\left(\max\left[T_i.(z_i - z_{i-1}) + In + f_{i-1} - \sum_{k=1}^{i} Et_k - f_i, Sm_i \right], 0 \right) \\
In &= \max\left(In - \left[Sm_i - (\theta_{j,i} - \theta_{PWPi}).z_i\right], 0\right) \\
\theta_{j+1,i} &= S_i / (z_i - z_{i-1}) + \theta_{PWPi}
\end{aligned}
\tag{25}
$$

There is only one last thing to do and that is to accumulate the actual and potential ET over the growing season.

$$
\begin{aligned}
&p > 0 \\
&SPET_{j+1} = SPET_j + PET_j \\
&SAET_{j+1} = SAET_j + AET_j
\end{aligned}
\tag{26}
$$

6. Discussion

The model presented here is the structure and mathematics for a water balance model. This model couples together some of the physics of water transport in the soils from authors more concerned with atmosphereic processes (Deardoff, 1977), with authors more concerned with soil processes (Philip, 1954; 1969; 1980). Here we have merely documented the approach that we have taken in construction of the model. This model has been used so far by Cook et al. (2003) in a two layered modelled to study the biological oxygen demand in runoff waters from sugar cane paddocks.

The model has also been used as constructed here in two recent applications associated with agent-based models. The applications are for the Bowen-Broken catchment in Queensland (Smajgl et al. 2007) where it was used in a study of grazing systems and in the Katanning catchment in Western Australian where it is being used to assess the longterm viability of the region. Although these agent-based models were developed to understand the system behaviour due to human activities the physical processes such as hydrology are important as they are a fundamental driver of the system productivity. Hence the water balance model is important.

Implementation of the water balance used has been written in MatLab and the code for this is presented in Appendix I. Checks with the mass balance for this code show that it is within rounding errors of the computer with a mass difference of 10^{-12}% after 10 years of simulation for climate data from Wivenhoe, South East Queensland.

7. Conclusions

The model presented here was been developed for a agent-based models that have been devised to look at other aspects of systems behaviour such as grazing systems and regional viability. In a 2-layer version it has been tested by Cook et al. (2003) and found to work well. Shao et al. (1996) have also successfully used a similar model for predicting soil surface water content. This model provides an improved scheme for predicting the soil water content near the surface.

References

Addiscott TM, Smith J and Bradbury N. 1995. Critical evaluation of models and their parameters. *Journal of Environmental Quality* **24**: 803-807.

Attard SJ, Inman-Bamber NG and Engelke J, 2003. Irrigation scheduling in sugarcane based on atmospheric evaporative demand. *Proceedings of the Australian Society of Sugar cane Technol*. **25**: (CD)12p.

Carlin, G. D., Cook, F. J., Cropp, R. A., Hartcher, M., Smajgl, A., Yu, B., and Heckbert, S. (2007). Using a catchment contour approach for simulation of ground and surface water behaviour within agent based modelling platforms. In `MODSIM 2007 International congress on Modelling and Simulation.' (Eds Oxley, L. and Kulasiri, D.) 60-66. (Modelling and Simulation society of Australia and New Zealand: Christchurch, NZ.)

Clothier B.E. and White I., 1981. Measurement of sorptivity and soil water diffusivity in the field. *Soil Sci. Soc. Am. J*., **45**:241-245.

Cook F.J., Pankhurst C.E., Bohl H.P., D'Amato C., Fanning D., Rayment G.E.and Carlin G.D., 2003. Fate of spilt sugars from cane harvest: Investigation and modelling of the phenomena. CSIRO Land and Water Technical Report 48/03, 26p
.

Deardorff JW, 1977. A parameterization of ground-surface moisture content for use in atmospheric prediction models. *Journal of Applied Meteorology* **16**:1182-1185.

Fath, BD, Distributed control in ecolgical networks. *Ecological Modelling,* **179**: 235-245.

Favier C, Chave J, Fabing A, Schwartz D and Dubois MA, 2004. Modelling forest-savanna mosaic dynamics in man-influenced environments: effects of fire, climate and soil heterogeneity. *Ecological Modelling,* **171**: 85-102

Hillel, D, 1980. Fundamentals of Soil Physics. Academic Press, Inc, 413p.

Monteith, JL, 1986. How do crops manipulate water supply and demand. *Philisophical Transactions of the Royal Society of London*, **A 316**: 245-259.

Philip JR, 1957. The theory of infiltration: 4. Sorptivity and algebraic infiltration equations. *Soil Sci*. **84**:257-264
.

Philip, JR, 1969. Theory of infiltration. *Advances in Hydroscience* **5**:215-296.

Philip JR. 1987. The infiltration joining problem. *Water Resources Research* **23**: 2239-2245.

Shao Y, Raupach MR and Leys JF, 1996. A model for predicting aeolian sand drift and dust emission on scales from paddock to region. *Aust. J. Soil Res*. **34**:309-342.

Smith M, Allen R and Pereira L, 1996. Revised FAO methodology for crop water requirements. *In*: Camp, C.R., Sadler, E.J. and Yoder, R.E. (Eds) *Evapotranspiration and Irrigation Scheduling. Proceedings of the International Conference*, November 3-6, 1996, San Antonio. ASAE, St., Joseph, Michigan, USA, 115-123.

Smajgl, A., S. Heckbert, I. Bohnet, G. Carlin, M. Hartcher and J. McIvor (2007), Simulating the grazing system in the bowen broken: An agent-based modelling approach CSIRO: WfHCF Canberra., 53.

Su X, Asseng, S, Cook F, Campbell P. and Carlin G, 2008. Initial implementation of an agent based model of the socio-economic and bio-physical factors driving the dynamics of an agricultural region in Western Australia. In Prep.

Teegavarapu RSV, 2007. Use of universal function approximation in variance-dependent surface interpolation method: An application in hydrology. *Journal of Hydrology*, **332**: 16-29.

White I and Sully, MJ, 1987. Macroscopic and microscopic capillary length and time scales from field infiltration. *WaterResources Res.,* **23**:1514-1522.

Appendix I

The % sign indicates that anything after this is text and not executed

```
function WB_ROe_n3(KC)
% Rewritten FC
Climate=xlsread('wivenhoe_wagga');% day no., Ep, Precip
KC = 1 % LN - added to the function call
N = size(Climate);
Soildata = xlsread('WivenhoeWaggaSoil2a');% z,thetaS, ThetaFC, Theta PWP,
Ks, D*, RD for each layer
%Initialise

Z(1) = Soildata(1,1);%LN - Now Za is Z1, Z1 is Z2 and so on
size_Z=size(Soildata);
n_layers=size_Z(1);%LN -=ABS((1+B2^2)*(G17+B2/A17)/(1-
H17*A17*B2))^(G17/(F17-G17)) get number of layers
for i=2:n_layers
    Z(i) = Soildata(i,1);
end
Z(n_layers+1) = Z(n_layers);% need for k loop
%LN - get parameters with names instead of using Soildata - easy to read
AWC = Soildata(:,3)- Soildata(:,4);
PWP=Soildata(:,4);
FC=Soildata(:,3);
ThetaS=Soildata(:,2);
Ks=Soildata(:,5);
Ks(n_layers+1) = Ks(n_layers);% need for k layer loop for infiltration and
drainage
D=Soildata(:,6);

MassW(1)= Z(1)*FC(1);% Initial mass in each bucket
MassW(2)= (Z(2)-Z(1))*FC(2)+MassW(1);

%LN - calculates depth of bottom layer, easyer to use when comparing to
%Root Depth RD
Depth(1)=0;
Depth(2)=Z(2);
for i=3:size_Z(1)
    MassW(i) = Z(i)*Soildata(i,3);
    Depth(i)=Depth(i-1)+Z(i);
end
MassW(n_layers+1) = MassW(n_layers);% needed for k loop
Dn = N(1,1);% no. of days
```

```
%D = Soildata(1,6);
RD = Soildata(1,7);

%LN - Initialize matricces and pre-allocate memory
PET= zeros(Dn,1);
RO = zeros(Dn,1);
CumI = zeros(Dn,1);
Drain = zeros(Dn,1);
Rain = zeros(Dn,1);
AET = zeros(Dn,1);
WC = zeros(n_layers+1,Dn);
I =zeros(n_layers+1);
Dr50 = zeros(Dn,1);
Dr90 = zeros(Dn,1);

for i = 2:Dn+1
%Evaporation process first
   % Surface layer
   I = zeros(1,n_layers+1);

   Ep = Climate(i-1,2)*KC;
   PET(i) = PET(i-1)+Ep; % Potential Et
   WC(1,i) = MassW(1)/Z(1);
   WC(2,i) = MassW(2)/Z(2);
   %layer 1(A)
   T = max(WC(1,i) - PWP(1),0);
   % available water
   Se = T*Z(1);
   if (Se - AWC(1)*Z(1)/2) >= 0
      E(1) = min(Ep,Se);
      E(1) = min(E(1),(WC(2,i) - PWP(2))*Z(2));
   elseif Se > 0
     E(1) = min(2*Ep*T/AWC(1),(WC(2,i) - PWP(2))*Z(2)) ;
   else
     E(1) =0;
   end
   MassW(1) = MassW(1)- E(1);
   WC(1,i) = MassW(1)/Z(1);
   AE = E(1);
   MassW(2) = MassW(2) - E(1);
   WC(2,i) = MassW(2)/Z(2);
   if RD > Z(2)
      F(1) = 1;
   elseif RD > Z(1)
      F(1) = (RD-Z(1))/(Z(2)-Z(1));
   else
```

```
    F(1) = 0;
end
%Layer 1(B)
T = max(WC(2,i) - PWP(2),0);
Se = T*Z(2);
E(2) = max((Ep -E(1))*F(1),0);
if E(2) > 0
    if (Se - AWC(2)*Z(2)/2) > 0
        E(2) = min(E(2),Se);
    elseif Se > 0
        E(2) = min(2*E(2)*T/AWC(2), Se);
    else
        E(2) = 0;
    end
end
MassW(2) = MassW(2) - E(2);
WC(2,i) = MassW(2)/Z(2);
AE = AE + E(2);
%k layers
WC(3,i) = MassW(3)/Z(3);%needed to initiate
for k=3:n_layers
    T = max(WC(k,i) - PWP(k),0);
    Se = T*Z(k);
    if RD > Depth(k);%(Z(2)+ sum(Z(3:k)))
        F(k-1) = 1;
    elseif RD > Depth(k-1);%Z(2)+ sum(Z(3:k-1))
        F(k-1) = (RD-Depth(k-1))/Z(k);%(RD-Z(2)-sum(Z(3:k-1)))/Z(k)
    else
        F(k-1) =0;
    end
    E(k) = max((Ep-sum(E(1:k-1)))*F(k-1), 0);
    if E(k) > 0
        if (Se - AWC(k)*Z(k)/2) >= 0
            E(k) = min(E(k),Se);
        elseif Se > 0
            E(k) = min(2*E(k)*T/AWC(k),Se) ;
        else
            E(k) = 0;
        end
    end
    MassW(k) = MassW(k) - E(k);
    WC(k,i) = MassW(k)/Z(k);
    AE = AE + E(k);
end
AET(i) = AET(i-1) + AE;
```

```
%Infiltration
   if Climate(i-1,3) > 0
      Ic = 0;
      for k=2:n_layers
         Ic = Ic + (ThetaS(k) -WC(k,i))*Z(k);
      end
      Sorp = sqrt(D(2)*4*(ThetaS(2) - WC(2,i))^2/pi);
      tg = (Sorp/((1-0.36)*Ks(1)))^2;
      if tg < 1
         Ir = Sorp*tg + Ks(1)*(1-tg*(1-0.36));
      else
         Ir = Sorp +0.36*Ks(1);
      end
      Imax = min(Ir,Ic);
      if Climate(i-1,3) > Imax
         Infil = Imax;
         ROf = Climate(i-1,3)-Imax;
      else
         Infil = Climate(i-1,3);
         ROf = 0;
      end
   else
      ROf = 0;
      Infil = 0;
      I(2) = 0;
   end
   RO(i) = RO(i-1)+ROf;
   CumI(i)= CumI(i-1) + Infil;
   Rain(i) = Rain(i-1) + Climate(i-1,3);

   % Infiltration into layer 1 bulk
   if Infil <= (ThetaS(2)-WC(2,i))*Z(2)
      MassW(2) = MassW(2) + Infil;
      I(2) = 0;
      TD(2) = 1;
   else
      MassW(2) = ThetaS(2)*Z(2);%Layer 1(B) becomes saturated
      I(2) = Infil - (ThetaS(2) - WC(2,i))*Z(2);
      TD(2) = 1 - min(1,I(2)/Ks(3));
      if I(2) > Ks(3); %Then all water cannot pass to next layer in 1 day
         ROx = I(2) - Ks(3);
         RO(i) = RO(i) + ROx;% Can't fit infiltration into soil so ROf increase
         CumI(i) = CumI(i) - ROx;
         I(2) = Ks(3);
      end
   end
```

```
    WC(2,i) = MassW(2)/Z(2);

    if TD(2) == 0
        Dr(2) = 0;
    elseif TD(2) < 1
        Dr(2) = min(Ks(2)*TD(2),(WC(2,i)-FC(2))*Z(2));
    elseif (TD(2) == 1) & (WC(2,i) > FC(2))
        Dr(2) = min(Ks(2)*(WC(2,i)- FC(2))/(ThetaS(2) - FC(2)),(WC(2,i)-
FC(2))*Z(2));
    else
        Dr(2) = 0;
    end
    if Dr(2)+I(2) > Ks(3)
        Dr(2) = Ks(3)-I(2);
    end
    MassW(2) = MassW(2) - Dr(2);
    WC(2,i) = MassW(2)/Z(2);
    MassW(3) = MassW(3)+Dr(2);
    WC(3,i) = MassW(3)/Z(3);

    %Infiltration into remaining layers

    for k=3:n_layers
        if I(k-1) <= (ThetaS(k)-WC(k,i) )*Z(k)
            MassW(k) = MassW(k) + I(k-1);
            I(k) = 0;
            TD(k) = 1;
        else
            MassW(k) = ThetaS(k)*Z(k);%saturation of k layer
            I(k) = I(k-1) - (ThetaS(k)-WC(k,i) )*Z(k);
            TD(k) = 1 - min(1,I(k)/Ks(k+1));
            if I(k) > Ks(k+1)
                ROx = I(k) - Ks(k+1);
                RO(i) = RO(i) + ROx;
                CumI(i) = CumI(i) - ROx;
                I(k) = Ks(k+1);
            end
        end
        WC(k,i) = MassW(k)/Z(k);
        if TD(k) == 0
            Dr(k) = 0;
        elseif TD(k) < 1
            Dr(k) = min(Ks(k)*TD(k),(WC(k,i)-FC(k))*Z(k));
        elseif (TD(k) == 1) & (WC(k,i) > FC(k))
            Dr(k) = min(Ks(k)*(WC(k,i)- FC(k))/(ThetaS(k) - FC(k)),(WC(k,i)-
FC(k))*Z(k));
```

```
        else
            Dr(k) = 0;
        end
        if Dr(k)+I(k) > Ks(k+1)
            Dr(k) = Ks(k+1)-I(k);
        end
        MassW(k) = MassW(k) - Dr(k);
        WC(k,i) = MassW(k)/Z(k);
        MassW(k+1) = MassW(k+1) + Dr(k);
        WC(k+1,i) = MassW(k+1)/Z(k+1);
    end
    Dr50(i) = Dr(6)+I(7)+Dr50(i-1);
    Dr90(i) = Dr(10)+I(11)+Dr90(i-1);
%Refilling layer 1(A)
    if WC(1,i) < FC(1)
        if WC(1,i) < WC(2,i)
            if WC(2,i) > PWP(2)
                SF1 = min(2*D(1)*(WC(2,i) - WC(1,i))/Z(2),(WC(2,i)-PWP(2))*Z(2)) ;
            else
                SF1 = 0;
            end
            SF2 = (FC(1) - WC(1,i))*Z(1);% maximum fill potential
            SFlux = min(SF2,SF1);
        else
            SFlux = 0;
        end
    else
        SFlux = 0;
    end
    MassW(1) = MassW(1)+SFlux;
    WC(1,i)=MassW(1)/Z(1);
    Drain(i) = Drain(i-1)+Dr(n_layers)+ I(n_layers);
    TM(i) =0;
    for k = 2:n_layers
        TM(i)=TM(i) + MassW(k);
    end
% Output
    WivOut(i-1,1) = Climate(i-1,1);
    WivOut(i-1,2) = PET(i);
    WivOut(i-1,3) = AET(i);
    WivOut(i-1,4) = Rain(i);
    WivOut(i-1,5) = CumI(i);
    WivOut(i-1,6) = RO(i);
    WivOut(i-1,7) = Drain(i);
    WivOut(i-1,8) = Dr50(i);
    WivOut(i-1,9) = Dr90(i);
```

```
    WivOut(i-1,10) = TM(i);
    massOut(:,i)=MassW(:);
end
xlswrite('Wagga2b.xls',WivOut);
csvwrite('WCWivenhoe1.xls',WC');
```

Contact Us
Phone: 1300 363 400
+61 3 9545 2176
Email: enquiries@csiro.au
Web: www.csiro.au

Your CSIRO

Australia is founding its future on science and innovation. Its national science agency, CSIRO, is a powerhouse of ideas, technologies and skills for building prosperity, growth, health and sustainability. It serves governments, industries, business and communities across the nation.