Alternative numerical solutions for the ponded water store

Introduction

It is common in land models to solve ordinary differential equations using explicit Euler approximations based on an operator-splitting approach with imposed constraints. This solution method can be problematic for three reasons: First, the sequential calculations limit representations of the coupling across processes (i.e., the solution is not very general); second, the explicit Euler method relies on extrapolating fluxes computed at the start of the time step for the entire length of the time step (i.e., the solution is not very accurate); and third, the imposition of sharp thresholds can lead to erratic model behavior which complicates the application of many model analysis methods. In the following sections we will illustrate the current solution method using the ODE for ponded water, and we will introduce alternative solution methods that are more general, more accurate, and that avoid imposition of sharp thresholds. The ponded water store is used here as an example of issues that will arise in other components of land models.

1 Current solution methods in CLM

Consider the following ODE for ponded water

$$\frac{dS_{pond}}{dt} = \left(q_{in} - e_{pond}\right) - q_{surf} - q_d \tag{1}$$

where q_{in} is the flux at the upper boundary, e_{pond} is the evaporation from the pond, q_{surf} is the surface runoff, and q_d is the drainage at the lower boundary. The sequence of steps can be to (1) add/subtract net input; (2) compute surface runoff; and (3) compute drainage, as

$$S^{*}_{pond} = S^{n}_{pond} + \left(q_{in} - e_{pond}\right)\Delta t \tag{2}$$

$$S^{**}_{pond} = S^{*}_{pond} - q^{*}_{surf}\Delta t \tag{3}$$

$$S^{n+1}_{pond} = S^{**}_{pond} - q^{**}_{d}\Delta t \tag{4}$$

where the superscripts n and $n+1$ define the start and the end of the time step, and the superscripts (*) and (**) denote intermediate solutions. The fluxes are formulated to satisfy solution constraints, where, for example

$$q_{surf}(S) = \begin{matrix} \min\left(\kappa\left(S - S_{mt}\right), \left(S - S_{mt}\right)/\Delta t\right) & S \geq S_{mt} \\ 0 & S < S_{mt} \end{matrix} \tag{5}$$

$$q_d(S) = \min\left(k_{sat}, S/\Delta t\right) \tag{6}$$

In the surface runoff example κ (s^{-1}) is the time constant for drainage, S_{mt} (m) is the threshold for drainage (e.g., to represent micro-topography), and in the drainage example k_{sat} (m s^{-1}) is the temporally constant saturated hydraulic conductivity.

2 Alternative solution methods

For simplicity we will continue to consider the fluxes sequentially – i.e., $dS/dt = -q$. We will show how alternative numerical methods can reduce erratic model behavior and improve numerical accuracy. We will also explain how some of the numerical methods can be extended to the more general case of process coupling.

2.1 Surface runoff

The surface runoff flux in equation (5) is a linear equation and hence is fairly straightforward to solve.

2.1.1 The implicit Euler solution

The state equation $dS/dt = -q_{surf}$ can be discretized in time as

$$S^{n+1} - S^{n} = -q_{surf}^{n+1}\Delta t \tag{7}$$

where the superscript n denotes the time index. Equation (7) is an implicit equation since the flux depends on the state at the end of the time step (time index $n+1$). Since the surface runoff flux q_{surf} in equation (5) has a linear dependence on storage, the implicit flux q_{surf}^{n+1} at the end of the time step can be estimated using the first-order Taylor series expansion

$$q_{surf}^{n+1} = q_{surf}^{n} + \left(\frac{dq_{surf}}{dS} \right)^{n} \left(S^{n+1} - S^{n} \right) \tag{8}$$

Combining equations (7) and (8), then and

$$S^{n+1} = S^{n} - \frac{q_{surf}^{n} \Delta t}{1 + \left(\frac{dq_{surf}}{dS} \right)^{n} \Delta t} \tag{9}$$

Given $q_{surf} = \kappa \left(S - S_{mt} \right)$ and $dq_{surf} / dS = \kappa$, and

$$S^{n+1} = S^{n} - \frac{\kappa \left(S^{n} - S_{mt} \right) \Delta t}{1 + \kappa \Delta t} \qquad S^{n} \geq S_{mt} \tag{10}$$

Note that implicit Euler solution in equation (10) is no more expensive than the implicit Euler solution, but the solution is much more accurate and does not require use of the "min" constraint.

2.1.2 The analytical solution

If q_{surf} is handled separately (i.e., operator splitting), then it is possible to solve $dS/dt = -q_{surf}$ analytically. Combining terms that depend on storage on the left-hand-side, the state equation can be given as

$$\frac{\frac{dS}{dt}}{-\kappa \left(S - S_{mt} \right)} = 1 \tag{11}$$

and integrating both sides of equation (25) with respect to t yields

$$\int \frac{dS}{-\kappa\left(S-S_{mt}\right)} = \int 1\,dt \tag{12}$$

Calculating the indefinite integrals provides

$$-\frac{1}{\kappa}\ln\left|S-S_{mt}\right| + C_1 = t + C_2 \tag{13}$$

and rearranging

$$S-S_{mt} = \pm\exp\left(-\kappa t-\kappa\left(C_2-C_1\right)\right) \tag{14}$$

Defining an arbitrary constant

$$C = \pm\exp\left(-\kappa\left(C_2-C_1\right)\right) \tag{15}$$

then

$$S-S_{mt} = C\exp\left(-\kappa t\right) \tag{16}$$

Given $t=0$, then $C=S_0-S_{mt}$, and

$$S\left(t\right) = \left(S_0-S_{mt}\right)\exp\left(-\kappa t\right)+S_{mt} \qquad S_0 \geq S_{mt} \tag{17}$$

where S_0 is the storage at time $t=0$.

2.2 Ponded water drainage

The original parameterization for ponded water drainage in equation (6) imposes a non-negative storage constraint to avoid "over-draining" the ponded water store. This sharp threshold can create erratic model behavior, since similar simulations can be on different sides of the threshold. The sharp threshold also creates difficulties in using derivative-based numerical methods (such as the implicit Euler method), since the "min" function makes it difficult to parameterize the dependence of the flux on storage.

Kavetski and Kuczera (2007) suggest use of a smoothing kernel to provide an alternative formulation of equation (6), where

$$q_d(S) = k_{sat} f_s(S) \tag{18}$$

where $f_s(S)$ is a dimensionless smoothing kernel. The smoothing kernel is selected such that in general $f_s \approx 1$, but asymptotically $f_s \to 0$ as $S \to 0$.

A conventient exponential smoothing kernel is

$$f_s(S) = 1 - \exp\left(-\frac{S - S_x}{m_s}\right) \tag{19}$$

where S_x is the lower solution bound (here $S_x = 0$) and m_s defines the strength of the smoothing. The transition $f_s \to 0$ becomes "sharper" $m_s \to 0$.

The modified parameterization in equation (18) behaves like the original parameterization in equation (6) whenever $f_s \approx 1$, but produces a smooth asymptotic transition as $S \to 0$. The smoothing in equation (18) opens up the possibilities to use more general numerical solution methods.

2.2.1 The implicit Euler solution

The state equation $dS/dt = -q_d$ can be discretized in time as

$$S^{n+1} - S^n = -q_d^{n+1} \Delta t \tag{20}$$

where the superscript n denotes the time index. As in the surface runoff case, equation (20) is an implicit equation since the flux depends on the state at the end of the time step (time index $n+1$).

In contrast to equation (5), equation (18) has a non-linear dependence on storage and an iterative solution is necessary (e.g., using the Newton-Raphson method). The implicit flux $q_{surf}^{n+1,m+1}$ after an iteration can be estimated using the first-order Taylor series expansion

$$q_d^{n+1,m+1} = q_d^{n+1,m} + \left(\frac{dq_d}{dS}\right)^{n+1,m} \Delta S^{n+1,m} \tag{21}$$

where $\Delta S^{n+1,m}$ defines the m^{th} iteration increment.

The state equation for drainage can then be formulated as

$$\left(S^{n+1,m} - S^n\right) + \Delta S^{n+1,m} = -k_{sat} f_s^{n+1,m} - k_{sat}\left(\frac{df_s}{dS}\right)^{n+1,m} \Delta S^{n+1,m} \tag{22}$$

where $S^{n+1,m} - S^n$ defines the state increment from the start of the time step to the start of an iteration. Here f_s and df_s / dS are from equation (19) assuming $S_x = 0$, i.e., $f_s = 1 - \exp\left(-S/m_s\right)$ and $df_s/dS = \exp\left(-S/m_s\right)/m_s$. Equation (22) can then be solved for the iteration increment $\Delta S^{n+1,m}$ as

$$\Delta S^{n+1,m} = \frac{S^n - k_{sat} f_s^{n+1,m} - S^{n+1,m}}{1 + k_{sat}\left(\frac{df_s}{dS}\right)^{n+1,m}} \tag{23}$$

and the state updated as

$$S^{n+1,m+1} = S^{n+1,m} + \Delta S^{n+1,m} \tag{24}$$

Equations (23) and (24) are applied repeatedly until $\Delta S^{n+1,m}$ is below a user-prescribed error tolerance.

2.2.2 The analytical solution

As with surface runoff, if q_d is handled separately (i.e., operator splitting), then it is possible to solve $dS/dt = -q_d$ analytically. Combining terms that depend on storage on the left-hand-side, the state equation can be given as

$$\frac{\frac{dS}{dt}}{k_{sat} f(S)} = 1 \tag{25}$$

and integrating both sides of equation (25) with respect to t yields

$$\int \frac{dS}{k_{sat} f(S)} = \int 1 dt \tag{26}$$

Given $u = 1 - \exp(-S/m_s)$, then $du/dS = (1-u)/m_s$, hence

$$\int \frac{m_s}{-k_{sat} u(1-u)} du = \int 1 dt \tag{27}$$

calculating the indefinite integrals provides

$$m_s \left(\ln|u-1| - \ln|u| \right) / k_{sat} + C_1 = t + C_2 \tag{28}$$

and, rearranging terms

$$\frac{u-1}{u} = \pm \exp\left(k_{sat} t / m_s + k_{sat} (C_2 - C_1) / m_s \right) \tag{29}$$

Defining an arbitrary constant

$$C = \pm \exp\left(k_{sat} (C_2 - C_1) / m_s \right) \tag{30}$$

then

$$u = -\frac{1}{C \exp(k_{sat} t / m_s) - 1} \tag{31}$$

Given $t = 0$

$$C = 1 - 1/u_0 \tag{32}$$

and S can be obtained from equation (19) as

$$S(u) = -\ln(1-u)m_s \tag{33}$$

The analytical solution hence requires implementing equations (31), (32), and (33).