Notes on Developing a Raster 2D Flow Routing Model from Conservation Laws and a Flow Grid

Scott D. Peckham
April 23, 1997

Conservation of Mass

The integral form of mass balance for surface flows can be written as

$$\frac{\partial}{\partial t}\int_{\Omega}\rho\, d\, dA = \int_{\Omega}\rho\, R\, dA - \int_{\partial\Omega}(\hat{n}\cdot\rho\langle\underline{u}\rangle d)\, dw. \tag{1}$$

This equation simply states that the mass in an arbitrary control volume (extending from the bed to the free water surface) can change with time in only two ways; namely it can (1) be added or subtracted from the top or bottom of the control volume, or (2) be convected across the (vertical) boundary of the control volume. In equation (1), $\langle u\rangle$ denotes vertically-averaged downstream velocity, and d is the flow depth. The function $R(x, y)$ will be called the effective rainrate and gives the net volume of water per unit area per unit time that is added or subtracted at the point (x, y). This forcing can be decomposed into four parts — precipitation, P, seepage from subsurface, S, evaporation, E, and infiltration, I — so that $R = (P + S) - (E + I)$. In general, it is clear that the variation of R in space and time may be quite complicated for a real land surface. We will be treating R as a function of space and time that is given to us as a sequence of 2D arrays, and will be trying to predict the basin response to this "forcing."

If we apply a discretized version of equation (1) to pixel i in a RiverTools flow grid, we get

$$\frac{\Delta d(i,t)}{\Delta t}(\Delta x\,\Delta y) = R(i,t)(\Delta x\,\Delta y) + \sum_{k\in N} Q(k,t) - Q(i,t) \tag{2}$$

where Δx and Δy are the pixel dimensions, Δt is the time step, N is the set of pixels that have pixel i as their parent,

$$Q(i,t) = u(i,t)\, d(i,t)\, \Delta w(i), \tag{3}$$

is the discharge from pixel i to its "parent" pixel just downstream, and $\Delta w(i)$ is the "differential" width of flow away from pixel i, which may be approximated with Δx or Δy. (I'm not quite sure about this yet.) Formally solving for Δd yields

$$\Delta d(i,t) = \Delta t\,\left\{\left[\frac{\sum_{k\in N} Q(k,t) - Q(i,t)}{\Delta x\,\Delta y}\right] + R(i,t)\right\}. \tag{4}$$

Conservation of Momentum

Equations for the conservation of vertically-integrated horizontal momentum can be obtained in a similar way. For surface flows, momentum balance simply states that the total (horizontal) momentum of the fluid within an arbitrary control volume (extending from the bed to the water surface) can change with time in three different ways, namely the water can (1) be accelerated down the free-surface gradient by gravity, (2) be decelerated through frictional processes, or (3) be convected across the boundary of the control volume. The loss of momentum through friction is a complex process — the no-slip boundary condition applied to the roughness elements on the bed results in a velocity gradient normal to the bed which acts to diffuse momentum from the interior of the flow to the bed. Once there, it is either transferred to the mobile bed elements, or dissipated as heat. For a vertically-integrated hydrostatic flow, momentum balance can be written

$$\frac{\partial}{\partial t}\int_{\Omega} \rho\langle \underline{u}\rangle \, d \, dA = \int_{\Omega} -\rho g d \nabla h \, dA + \int_{\Omega} \underline{\tau_b} \, (1+\nabla b \cdot \nabla b)^{1/2} \, dA - \int_{\partial\Omega} d \, \langle \underline{u}\rangle \, (\hat{n}\cdot \rho\langle \underline{u}\rangle) \, dw, \quad (5)$$

where $\langle \underline{u}\rangle = (u, v)$ is the vertically-integrated horizontal velocity, d is the depth, b is the height of the bed above an arbitrary datum, and $h = (b + d)$ is the free-surface height. The quantity τ_b appearing in the third term is the horizontal component of the (total) shear stress at the bed, which we are taking to include all of the momentum loss mechanisms in the problem, including skin friction due to grain roughness, and form (or pressure) drag due to bedforms, bars, and any other topographic elements. It has been argued previously by Nelson and Smith [1989] and others that this approach is justified since ultimately all of these momentum-loss effects are a consequence of processes that act at or near the bed. The factor $(1 + \nabla b \cdot \nabla b)^{1/2} \, dA$ that appears in the momentum-loss term of (5) is just the differential surface area of the bed, b, which can be appreciably greater than dA near the banks of the channel.

If we apply a discretized (and channel-fitted) version of equation (5) to pixel i in a RiverTools flow grid, we get

$$\left[\frac{\Delta u(i,t)}{\Delta t}\, d(i,t) + \frac{\Delta d(i,t)}{\Delta t}\, u(i,t)\right] (\Delta x \, \Delta y)$$

$$= \sum_{k\in N} u(k,t)\, Q(k,t) - u(i,t)\, Q(i,t) + g\, d(i,t)\, S(i,t)\, (\Delta x\, \Delta y) - f(i,t)\, u^2(i,t)\, (\Delta x\, \Delta y) \quad (6)$$

Here, $S(i,t)$ is strictly the free-surface slope between pixel i and its parent pixel. Notice that the time derivative term in (5) has been expanded using the product rule, and we have used the law of the wall as a closure for the boundary shear stress. In the latter closure, the drag coefficient, f, is given by

$$f = \left[\frac{\kappa}{\ln\,(a\, d/z_0)}\right]^2, \quad (7)$$

where z_0 is the roughness parameter, $(z_0/a\,d)$ is the relative roughness, $\kappa \approx 0.408$ is von Karman's constant, and a is an integration constant which is either 0.368 or 0.476 depending on the assumed vertical velocity profile. (See Appendix C of our DSS paper.)

Formally solving (6) for $\Delta u(i,t)$ we have

$$\Delta u(i,t) = -\left[\frac{u(i,t)}{d(i,t)}\right]\Delta d(i,t)$$

$$+\left(\frac{\Delta t}{d(i,t)}\right)\left\{\left[\frac{\sum_{k\in N} u(k,t)\,Q(k,t) - u(i,t)\,Q(i,t)}{\Delta x\,\Delta y}\right] + g\,d(i,t)\,S(i,t) - f(i,t)\,u^2(i,t)\right\} \tag{8}$$

Using our previous equation (4) for $\Delta d(i,t)$, this simplifies to

$$\Delta u(i,t) = \left(\frac{\Delta t}{d(i,t)}\right)\left\{\frac{\sum_{k\in N}\left[u(k,t) - u(i,t)\right]Q(k,t)}{\Delta x\,\Delta y}\right\}$$

$$+\left(\frac{\Delta t}{d(i,t)}\right)\left[-R(i,t)\,u(i,t) + g\,d(i,t)\,S(i,t) - f(i,t)\,u^2(i,t)\right]. \tag{9}$$

Here, as in (4), $Q(i,t)$ can be eliminated with equation (3).

The result is that equations (4) and (9) tell us how the depth and downstream velocity at each pixel will change in a time increment, Δt, as a function of conditions existing at time t. That is we have

$$u(i,t+\Delta t) = u(i,t) + \Delta u(i,t) \tag{$*$}$$

$$d(i,t+\Delta t) = d(i,t) + \Delta d(i,t). \tag{$*$}$$

Keep in mind, however, that $S(i,t)$ and $f(i,t)$ both depend on $d(i,t)$ and must be updated at each time step.

Another interesting feature of this approach is that it doesn't require the computation of any second order space derivatives, and so we don't have to worry about couplings between time steps and "space steps," as we would if we were working with the heat equation, for example. No spatial derivatives of velocity occur in our approach, because we have not employed the divergence theorem to convert integrals over the boundary of our control volume (a pixel) to integrals over its interior.