NOTES ON DEVELOPING A RASTER 2D FLOW ROUTING MODEL FROM CONSERVATION LAWS AND A D8 FLOW GRID

SCOTT D. PECKHAM
FEBRUARY 23, 2017

1. 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 $R(x, y)$ is known as *runoff*, *effective rainrate* or *excess rainrate* and gives the net volume of water per unit area per unit time that is added or subtracted vertically at the point (x, y). This forcing can be decomposed into five parts — rainfall, P, snowmelt, S, baseflow seepage from subsurface, B, evaporation, E, and infiltration, I — so that $R = (P + S + B) - (E + I)$. 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 DEM, then for a grid cell with dimensions Δx and Δy, the time derivative of the volume of water within the cell is given by

$$\frac{\partial V}{\partial t} = R_i\,(\Delta x\, \Delta y) - Q_i + \sum_{k \in N} Q_k. \tag{2}$$

where V is the volume of water, $R(i, t)$ is the *excess rainrate* or *runoff* in cell i at time t, $Q(i, t)$ is the volume flow rate of water *leaving* the grid cell, and the sum is the total volume flow rate of water *entering* the grid cell from its D8 neighbor cells. If this volume of water were distributed uniformly over the grid cell, we would have

$$V = d_c\, \Delta x\, \Delta y, \tag{3}$$

where d_c would be the uniform water depth over a grid cell. However, if all of the water is contained within a prismatic channel of length ds that has a trapezoidal cross-section, then water volume can be expressed as

$$V = A_w \, ds = \left[d \, w_b + d^2 \, \tan(\theta) \right] \Delta s, \tag{4}$$

where w_b is the bottom width of the trapezoid, A_w is the wetted cross-sectional area of the channel, d is the depth of water in the channel, Δs is the channel segment length (in the grid cell) and θ is the *flare angle* of the bank (relative to the vertical). Note that once we have updated V with (2), we can solve the quadratic (4) to get the new flow depth, d, as

$$d = \frac{-w_b + \sqrt{w_b^2 + 4 \, \tan(\theta) \, V / \Delta s}}{2 \, \tan(\theta)}. \tag{5}$$

Computing the time derivative of (4) and simplifying, we also find that

$$\frac{\partial V}{\partial t} = \left(\frac{\partial d}{\partial t} \right) \left[w_b + 2 \, d \tan(\theta) \right] \, ds = \left(\frac{\partial d}{\partial t} \right) w_{top} \, ds = \left(\frac{\partial d}{\partial t} \right) A_{top}. \tag{6}$$

Here, w_{top} is the top width of the wetted cross-section and A_{top} is the top area of the prismatic channel. Equating (2) and (6) for $\partial V / \partial t$, we obtain

$$\frac{\partial d}{\partial t} \approx \frac{\Delta d}{\Delta t} = \left(\frac{R \, \Delta x \Delta y - Q_i + \sum_k Q_k}{A_{top}} \right). \tag{7}$$

TopoFlow solves (2) for $V(i,t)$ with simple, explicit time-stepping, and then computes channel flow depth for every grid cell using (5).

2. 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

$$\frac{\partial}{\partial t} \int_\Omega \rho \langle \underline{u} \rangle \, d \, dA = \int_\Omega - \rho g d \nabla h \, dA + \int_\Omega \underline{\tau_b} \, (1 + \nabla b \cdot \nabla b)^{1/2} \, dA - \int_{\partial\Omega} d \, \langle \underline{u} \rangle \, (\hat{n} \cdot \rho \langle \underline{u} \rangle) \, dw, \tag{8}$$

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. The factor $(1 + \nabla b \cdot \nabla b)^{1/2} \, dA$

that appears in the momentum-loss term is just the differential surface area of the bed, b, which can be appreciably greater than dA near the banks of the channel.

Assuming incompressible flow, we can divide through by the water density, ρ. With the channel segment in grid cell i of the DEM as the control volume, Ω, equation (8) then becomes

$$\frac{\partial}{\partial t}(u_i V_i) = g\, S_i\, V_i - f_i\, u_i^2\, P_w\, \Delta s - u_i\, Q_i + \sum_{k \in N} u_k\, Q_k. \tag{9}$$

Here, S_i is the free-surface slope between pixel i and its D8 parent pixel (downstream), P_w is the wetted perimeter of the trapezoidal cross-section, given by

$$P_w = w_b + \frac{2\, d}{\cos(\theta)}, \tag{10}$$

and f_i is half the Fanning friction factor, which for Manning flow resistance is given by

$$f_i = \frac{\tau_i}{\rho\, u_i^2} = \frac{g\, n^2}{R_h^{1/3}} \approx \frac{g\, n^2}{d^{1/3}}. \tag{11}$$

(Note: When we use Manning's formula and the depth-slope product formula for total shear stress — both of which strictly only hold for uniform, nonaccelerating flow — we can't really decouple the gravity and friction terms as we can when using the Logarithmic Law of the Wall. So there is some circularity here, but this seems to be standard practice.) Note that the last two terms in (9) represent the flux of momentum leaving a DEM grid cell and the flux of momentum that enters the grid cell from its D8 neighbors. Letting $M_i = u_i\, V_i$, (total channel momentum divided by ρ), we have

$$\Delta M_i = \Delta t \left[g\, S_i\, V_i - f_i\, u_i^2\, P_w\, \Delta s - u_i\, Q_i + \sum_{k \in N} u_k\, Q_k \right]. \tag{12}$$

In TopoFlow, for the *dynamic wave method*, M_i is updated with (9) after each time step, and then the flow velocity is updated as: $u_i = M_i / V_i$. Note, however, that with this simple, explicit time-stepping method, Δt may need to be very small to achieve numerical stability. Instability results in the friction term becoming much larger than the gravity term, which leads to large, negative momentum in some grid cells. (TopoFlow currently disallows this by setting the momentum to zero in those cells.) The problem is made worse by running TopoFlow with an initial flow depth of zero in all grid cells. This is the default, which is not a problem for the *kinematic wave* and *diffusive wave* methods of flow routing. In view of these facts, the dynamic wave method should be used with caution. Performing a series of model runs with successively smaller time steps and comparing peak values can help to determine whether true numerical stability has been achieved. For stable runs, the frictional loss term typically increases smoothly to approach the value of the gravitational acceleration term from below, and the momentum influx and outflux are also approximately equal. A reversal of flow direction is currently not allowed and would require dividing the backflow between multiple contributing D8 neighbors.

In the TopoFlow source code, we first update runoff, R, then discharge, Q, then flow volume, V, then flow depth, d, then hydraulic radius, Rh, then free surface slope, S, then friction factor, f, then $M = u\,V$ and finally, the mean flow velocity, u.