2-D Shallow-Water FV WENO ADER-DT Model

May 25, 2019

1 Shallow-Water Equations

The Shallow-Water equations are as follows:

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} = \mathbf{s}$$

$$\frac{\partial}{\partial t}\begin{bmatrix} h \\ hu \\ hv \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix} hv \\ hvu \\ hv^2 + \frac{1}{2}gh^2 \end{bmatrix} = \begin{bmatrix} 0 \\ -gh\frac{\partial h_S}{\partial x} \\ -gh\frac{\partial h_S}{\partial y} \end{bmatrix}$$

where h is the fluid thickness, u and v are the velocities in the x- and y-directions, respectively, $g = 9.8\text{m s}^{-2}$ is acceleration due to gravity, h_S is the height of the bottom topography, and S_{fx} and S_{fy} are the Manning's coefficients in the x- and y-directions, respectively.

We could transform the bottom topography terms to include a portion of them in the conservative flux. However, this would end up making the gravity wave speed $\sqrt{g\left(h + h_s\right)}$ in the flux Jacobian diagonalization. Since h_s could become arbitrarily large, this would seemingly limite the time step to unnecessarily small values. If the equations remain as they are above, then the gravity wave speed becomes $\sqrt{gh}$, which is guaranteed to be smaller (thus permitting a larger stable time step).

1.1 Differential Transforms

We perform DTs only in the temporal dimension, leaving space continuous:

$$\mathbf{q}\left(k_t + 1\right) = -\frac{1}{k_t + 1}\frac{\partial \mathbf{f}}{\partial x}\left(k_t\right) - \frac{1}{k_t + 1}\frac{\partial \mathbf{g}}{\partial y}\left(k_t\right) + \frac{1}{k_t + 1}\mathbf{s}\left(k_t\right)$$

The spatial derivatives of the flux will be computed with a matrix-vector product. Thus, we only need the temporal DTs of the flux vectors themselves after

computing the state:

$$\mathbf{F}(k_t) = \begin{bmatrix} Q_2(k_t) \\ \Phi_{22}(k_t) + \frac{1}{2}gP(k_t) \\ \Phi_{23}(k_t) \end{bmatrix} \quad ; \quad \mathbf{G}(k_t) = \begin{bmatrix} Q_3(k_t) \\ \Phi_{23}(k_t) \\ \Phi_{33}(k_t) + \frac{1}{2}gP(k_t) \end{bmatrix} \quad ; \quad \mathbf{S}(k_t) = \begin{bmatrix} 0 \\ -g\frac{\partial h_S}{\partial x}Q_1(k_t) \\ -g\frac{\partial h_S}{\partial y}Q_1(k_t) \end{bmatrix}$$

$$\Phi_{ij}(k_t) = \frac{1}{Q_1(0)} \sum_{r_t=0}^{k_t} \left[Q_i(r_t) Q_j(k_t - r_t) - Q_1(r_t) \Phi_{ij}(k_t - r_t) \right]; \quad \forall i \in \{2,3\}, j \in \{2,3\}$$

$$P(k_t) = \sum_{r_t=0}^{k_t} Q_1(r_t) Q_1(k_t - r_t)$$

2 Balancing Attempts

2.1 Subtract a balanced state

In a fluid with zero momentum $hu = hv = 0$, there is a balance between the gradient of the balanced fluid depth, h_B, and the gradient of the terrain, h_S:

$$\frac{\partial h_B}{\partial x} = -\frac{\partial h_S}{\partial x} \implies h_B = h_0 - h_S \implies \frac{\partial g h_B^2/2}{\partial x} = -g h_B \frac{\partial h_S}{\partial x}$$

To respect this balance in a realistic flow, I use the same technique used in handling hydrostatic balance in a stratified fluid. I subtract the balanced state from the momentum equations as:

$$\frac{\partial}{\partial t}\begin{bmatrix} h \\ hu \\ hv \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 - \frac{1}{2}gh_B^2 \\ huv \end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix} hv \\ hvu \\ hv^2 + \frac{1}{2}gh^2 - \frac{1}{2}gh_B^2 \end{bmatrix} = \begin{bmatrix} 0 \\ -g(h - h_B)\frac{\partial h_S}{\partial x} \\ -g(h - h_B)\frac{\partial h_S}{\partial y} \end{bmatrix}$$

In doing this, I should be essentially adding zero to the momentum equations since the subtractions are, by definition, balanced. In numerical experiments, this would be cheating for a true fluid at rest, but when I add a fluid perturbation to the free surface height with terrain, it gets rid of the imprinting due to imbalance between the momentum fluxes and source terms.

2.1.1 How close does h_0 need to be?

It turns out that h_B needs to be very close to h to make the imbalance effects sufficiently small. But if we use a separate h_0 for every cell, then the fluxes aren't compatible in the Riemann solver. Therefore, we need a way to correct for the differences in h_0 between different cells. Luckily, we already have a stencil in order to do this.

2.1.2 Getting h_0 as close to h as possible

To subtract out as much of the source term as possible via the balance subtraction method, each cell will use its own local h_0, which is the cell-average of h. To keep the fluxes consistent in the Riemann solver, the locally computed balance corrections are saved and the fluxes are left alone during reconstruction. The balance corrections are applied in the tendency calculation instead. By doing this, even with $\mathcal{C}^0$ discontinuous terrain, the terrain is not visible during the simulation.