Project 6: 2D Finite-Volume Solver

AE 623, Computational Fluid Dynamics II, Winter 2020

Due: April 3, 11:55pm, electronically via Canvas

1 Problem Description

In this project you will write a first-order finite volume code to solve the two-dimensional shallow-water equations (SWE),

$$
\begin{array}{lllllll}
\frac{\partial}{\partial t}(h) & + & \frac{\partial}{\partial x}(hu) & + & \frac{\partial}{\partial y}(hv) & = 0 & \text{(continuity)} \\
\frac{\partial}{\partial t}(hu) & + & \frac{\partial}{\partial x}(hu^2 + gh^2/2) & + & \frac{\partial}{\partial y}(huv) & = 0 & (x\text{-momentum}) \\
\frac{\partial}{\partial t}(hv) & + & \frac{\partial}{\partial x}(hvu) & + & \frac{\partial}{\partial y}(hv^2 + gh^2/2) & = 0 & (y\text{-momentum})
\end{array}
$$

In these equations, h is the water height [m], $\vec{v} = u\hat{\mathbf{i}} + v\hat{\mathbf{j}}$ is the water velocity [m/s], and $g = 9.8\text{m/s}^2$ is the acceleration due to gravity. The spatial domain is the city geometry from the previous project. At time $t = 0$, the solution is initialized to

$$h^0(x, y) = 1.0 + 0.3e^{-50(x-1.5)^2 - 50(x-0.7)^2}, \quad \vec{v}^0(x, y) = 0.$$

Figure 1 shows this initial condition on the coarsest mesh.

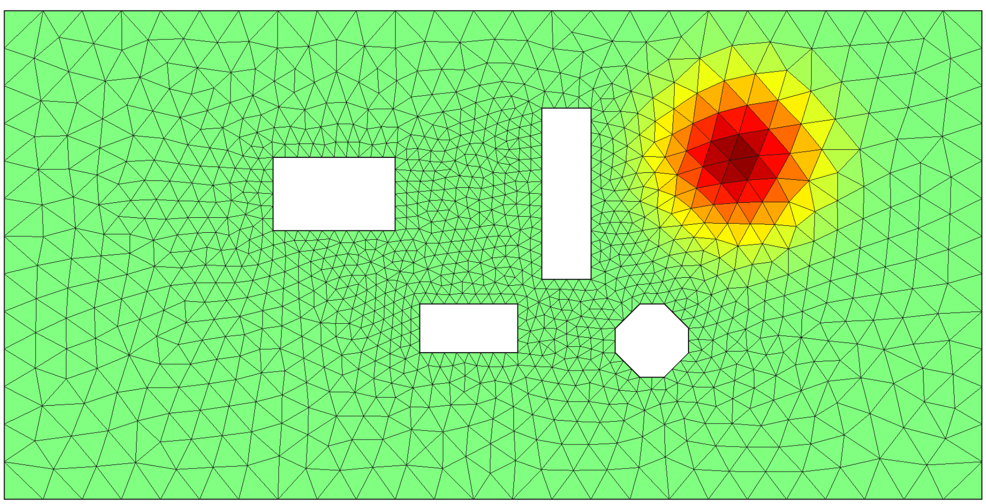

Figure 1: Initial condition, $h^0(x, y)$, on the coarsest mesh. The color scale is from 0.7 to 1.3.

2 Approach

2.1 Finite Volume Solver

Implement a first-order finite-volume solver, where the state, $\mathbf{u} = [h, hu, hv]$, inside each cell is approximated as a constant cell-average. The residual on cell i is given by

$$\mathbf{R}_i = \sum_{e=1}^{3} \hat{\mathbf{F}}(\mathbf{u}_i, \mathbf{u}_{N(i,e)}, \vec{n}_{i,e}) \Delta l_{i,e},$$

where e indexes the edges, $N(i, e)$ is the neighbor cell of i across edge e, $\vec{n}_{i,e}$ is the normal pointing out of cell i on edge e, and $\Delta l_{i,e}$ is the length of edge e of cell i. $\hat{\mathbf{F}}$ is the numerical flux: use the Roe flux as the rotated approximate Riemann solver, with an entropy fix of $\epsilon = 0.01c$. At $t = 0$ set the cell averages from the given initial condition evaluated at the centroid of each cell.

2.2 Boundary Conditions

In this project, all boundary conditions are walls. Impose this condition through the flux, which you should evaluate by using the interior state with the wall-normal velocity component zeroed out. Note that you do not need to use ghost-cells: the flux is the continuous flux dotted with the normal vector, computed with the boundary state (zero normal velocity).

2.3 Time Stepping

To advance the solution in time, use the forward Euler method with a constant time step. Compute this time step from the CFL condition and the largest wave speed present in the initial condition. The update for the cell-average in cell i is:

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n - \frac{\Delta t^n}{A_i} \mathbf{R}_i^n,$$

where A_i is the area of cell i and $\mathbf{R}_i^n$ is the residual on cell i, computed from $\mathbf{u}^n$.

2.4 Output Calculation

Of interest in this project are the forces on the buildings. In the shallow water equations, the force exerted by the water on a building is

$$\vec{F} = F_x \hat{\mathbf{i}} + F_y \hat{\mathbf{j}} = \int_{\text{building}} \frac{1}{2} \rho g h^2 \vec{n} \, dl,$$

where $\rho = 1000\text{kg/m}^3$ is the water density, and $\vec{n}$ is the normal pointing out of the computational domain (i.e. into the building). Evaluate this expression for every building by summing over the mesh boundary edges that constitute each building. Note that this integral is the same (except for the ρ multiplication) as the one appearing in the residual calculation (the momentum component). This means that you can share code between the output and residual calculations.

3 Tasks

1. **[20%]** Derive and implement the Roe flux for the 2D shallow water equations. Design and document tests of the fluxes that verify your implementation.

2. **[25%]** Write a first-order finite-volume solver and perform the following two tests on your `city0.gri` mesh. Describe your implementation and the setup/results of the tests.

   (a) *Free-stream test*: impose free-stream (constant $h = 1$) boundary conditions on all boundaries, and initialize the state on each cell to the same free-stream. You should obtain a near machine-precision zero residual norm on the first iteration.

   (b) *Free-stream preservation test*: after passing the previous test, let your code run for many, $\mathcal{O}(1000)$, time steps. The residual norm should hover around machine precision. State what time step you use in this test.

3. **[25%]** Write a code for calculating outputs, which are the time-varying forces on each of the four buildings. Using your three grids, perform simulations up to a final time of $T = 2$s, and generate plots of the building forces as functions of time. Make two plots for each building, showing the x and y force components, and overlay the forces from all three meshes on each plot. Discuss the results, in particular noting the differences between meshes.

4. **[20%]** Generate contour plots of the water height at times $t = 0, 0.05, .1, .15, .2, .25$s on your finest mesh. Use color axis limits of [0.7, 1.3] at $t = 0$ and [0.9, 1.1] for all later times. Discuss qualitatively whether the contours are consistent with your expectations.

4 Deliverables

You should turn in, electronically via Canvas,

1. A technical report as a `.pdf` file that describes your methods and results, and that answers the questions. The report should be professional, complete, and concise. **10%** of the project grade will be determined by professionalism: neatness, labeling of axes, spelling, etc.

2. Documented source files of all codes written for this project, as one `.zip` archive. You should provide all files necessary to run your program(s).

This is an individual assignment. You can discuss the project at a high level with each other, but you must turn in your own work.