Prof. Fabien Evrard & Fabian Dettenrieder
AE 410/CSE 461 – Computational Aerodynamics
Spring 2024

Homework #4

Due date: April 11, 2024

In this homework, you will solve the inviscid Burgers equation using finite volumes and the Godunov method. The inviscid Burgers equation is given as

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(f(u)\right) = 0 \ , \quad \text{with } f(u) = \frac{u^2}{2} \ . \tag{1}$$

Problem 1 · Suppose an initial condition $u(x, t = 0) = u_0(x)$ that is smooth, i.e., differentiable everywhere.

Q1 → Show that the exact solution to Burgers' equation with this initial condition is given implicitly as

$$u(x, t) = u_0(x - tu(x, t)) \ . \tag{2}$$

Note: Since you will need to use the differential form of Burgers' equation to get to this result, this expression is only valid until the solution forms its first shockwave (i.e., until it is not differentiable everywhere anymore).

Problem 2 · To solve Burgers' equation using finite volumes, you will employ Godunov's method. This requires knowing the solutions to the Riemann problem corresponding to the initial conditions

$$u(x, 0) = \begin{cases} u_L & \text{if } x \leq 0 \\ u_R & \text{if } x > 0 \end{cases} \tag{3}$$

Q2.1 → Derived the five fundamental solutions to this Riemann problem and express them in terms of u_L, u_R, and the similarity variable $\xi = x/t$.

Q2.2 → Develop a computational *Riemann solver* that takes u_L, u_R, and ξ as inputs and returns the corresponding solution to the Riemann problem for the similarity variable ξ. Verify that each case is correctly implemented by testing relevant combinations of u_L and u_R.

Problem 3 · When using finite volumes, the discrete variables under consideration are the cell averages of the solution $u(x, t)$. To be rigorous, initializing our simulation thus requires to calculate the cell averages of the initial condition $u(x, 0)$.

Q3.1 → Develop a computational function that returns the average of any function $\mathcal{F}(x)$ on an interval $[a, b]$ using the trapezoidal integration rule:

$$\frac{1}{b-a}\int_a^b \mathcal{F}(x)\ \mathrm{d}x \simeq \frac{1}{2K}\sum_{k=0}^{K-1}\left[\mathcal{F}\left(a + (b-a)\frac{k}{K}\right) + \mathcal{F}\left(a + (b-a)\frac{k+1}{K}\right)\right] \ , \tag{4}$$

The inputs of this function should be:

- The bounds a and b.
- The function $\mathcal{F}$.
- The number of integration points K.

Its only output should be:

- The (approximated) average of $\mathcal{F}(x)$ on $[a, b]$.

Q3.2 → Verify your code by computing the cell-averages of the function

$$\mathcal{F}(x) = \sin(10x) \tag{5}$$

on the computational domain $[0, 1]$ discretized into $N_x = 10$ cells (i.e., the first cell spans $[0, 1/N_x]$, the second cell spans $[1/N_x, 2/N_x]$, etc.).

Try using $K = 1$, $K = 2$, $K = 3$, or $K = 5$ integration points per cell. Quantitatively comment on the accuracy of your numerical integration against the exact analytical cell-averages of the function.

Q3.3 → **`This question is only required for those taking AE 410/CSE 461 for 4 credit hours`**. Develop a computational function that returns the average of any function $\mathcal{F}(x)$ on an interval $[a, b]$ using the Gauss-Legendre integration rule:

$$\frac{1}{b-a}\int_a^b \mathcal{F}(x)\ \mathrm{d}x \simeq \frac{1}{2}\sum_{k=0}^{K-1}\left[w_k\, \mathcal{F}\left(a + \frac{(b-a)}{2}(x_k+1)\right)\right] \ , \tag{6}$$

where the abscissae and weights $(x_k, w_k), k \in \{0, \ldots, K-1\}$, are given by the Gauss-Legendre quadrature rule of order K.

Try using $K = 1$, $K = 2$, $K = 3$, or $K = 5$ quadrature points per cell. Quantitatively comment on the accuracy of your numerical integration against the exact analytical cell-averages of the function.

Hints: In `Python`, you can compute the abscissae and weights corresponding to the Gauss-Legendre quadrature rule of order K using the `Numpy` function:

```
x, w = np.polynomial.legendre.leggauss(K)
```

Alternatively, exact Gauss-Legendre abscissae and weights can be found at:

`https://mathworld.wolfram.com/Legendre-GaussQuadrature.html`.

Problem 4 · Write your own finite-volume solver for Burgers' equation using the space- and time-integrated form of the governing equation in each computational cell,

$$U_n^{m+1} - U_n^m + \frac{\Delta t}{\Delta x}\left(F_{n+1/2} - F_{n-1/2}\right) = 0 \ , \tag{7}$$

where all variables have the meanings discussed in class and given in `AE410-Notes-6.pdf`. According to Godunov's method, compute the time averaged fluxes from the solution of localized Riemann problems at the cell faces.

Boundary conditions can be imposed via the introduction of "ghost" cells. At an outflow boundary, the ghost cell value can be extrapolated from the interior of the domain. For instance, at a right outflow boundary, you can use

$$U_{N_x} = U_{N_x-1} \qquad \text{or} \qquad U_{N_x} = 2U_{N_x-1} - U_{N_x-2} \ . \tag{8}$$

At an inflow boundary, since we do not provide Dirichlet boundary conditions, the ghost cell value can be chosen so as to impose the gradient of u at the boundary to be zero. For instance, at a left inflow boundary, you can use

$$U_{-1} = U_0 \ . \tag{9}$$

Q4.1 → Initialize your simulation by computing the cell averages of

$$u(x,0) = \frac{1}{10} + \exp\left(\frac{-\left(x-\frac{1}{4}\right)^2}{2\sigma^2}\right) \ , \quad \sigma = \frac{3}{40} \ , \tag{10}$$

on the domain $[0,1]$, discretized into $N_x = 500$ cells.

Q4.2 → With this initial condition, plot the solution to Burgers' equation at times $t \in \{0, 0.1, 0.2, \ldots, 1.5\}$. Make sure that the CFL number is always smaller than 1/2.

Q4.3 → **`This question is optional. If successfully answered, it will grant you bonus points`**. On the same figure as in Q4.2, plot the analytical solution for those times, until a shock forms.

Hint: You can make use of your result from Problem 1.

Submission guidelines · Instructions on how to prepare and submit your report are available on the course's Canvas page at `https://canvas.illinois.edu/courses/43781/assignments/syllabus`