Middle East Technical University
Department of Mechanical Engineering
ME 485-CFD using FVM
Fall 2024
Homework 3

Your task is to integrate limiters and upwinding methods into the *mefvm* code. You are going to solve the following level set advection equation given as

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0$$

where $\mathbf{u}$ is the known advection velocity and ϕ is the scalar level set function. The level-set method is a popular computational technique for tracking moving interface problems. The method relies on an implicit representation of the interface as the zero set of an auxiliary function i.e. level set function as given below

$$\phi(x,t) > 0 \text{ for } x \in \text{fluid one}$$
$$\phi(x,t) < 0 \text{ for } x \in \text{fluid two}$$
$$\phi(x,t) = 0 \text{ for } x \in \text{interface}$$

If the velocity field is conservative, satisfying the $\nabla \cdot \mathbf{v} = 0$, the level set equation can be written

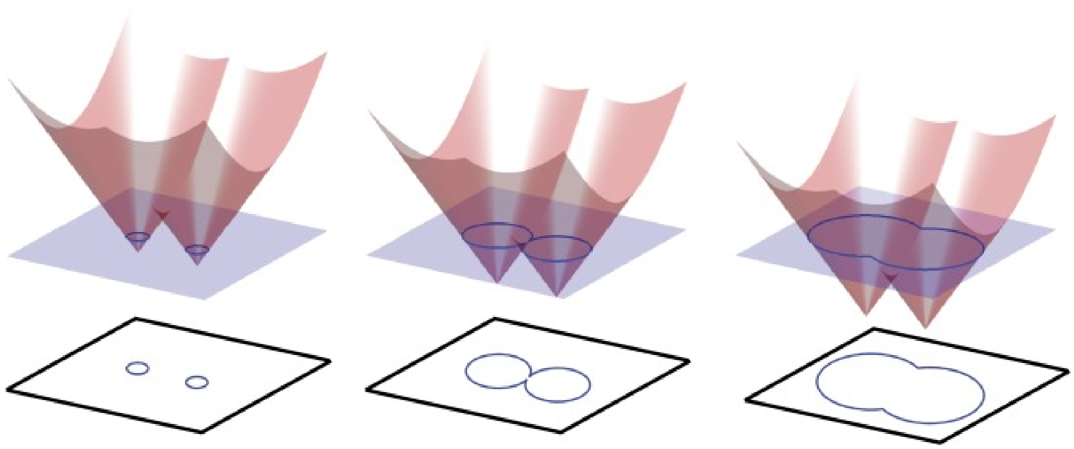

Figure 1: Level set function for interface problems

in the conservative form,

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{v}\phi) = 0$$

Steps to Complete the Homework

1. **Retrieve Code:**
   - Download the initial code version from the GitHub repository: GitHub Link.
   - Ensure you work within the branch **HW3**.

2. **Complete Missing Implementation of Barth-Jespersen limiter:**

   Unlike the lecture notes, you will implement the Barth-Jespersen limiter based on the maximum and minimum cell-centered values using the stencil given in Figure 2. If the maximum

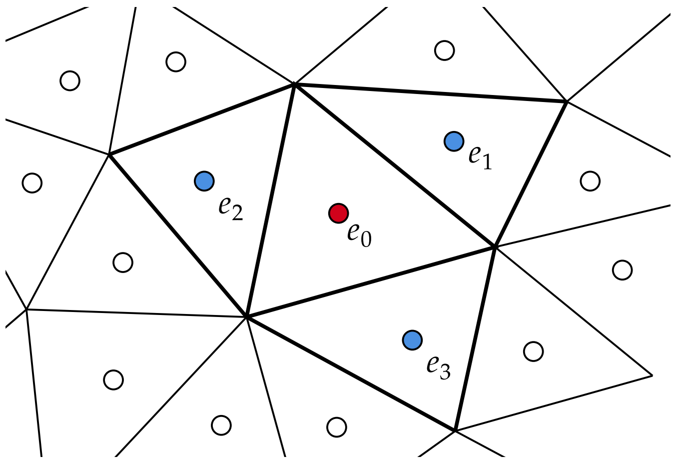

Figure 2: Stencil for Barth-Jespersen limiter implementation for $N = 4$

   and minimum of the stencil of element number N are defined as

$$\phi_{\max} = \max\left(\phi_i\right), \forall i \in [0, N-1]$$
$$\phi_{\min} = \min\left(\phi_i\right), \forall i \in [0, N-1]$$

   and the change of unlimited constructed values for every face the element $(\phi_f - \phi_0)$ is given as

$$\Delta_f = \left(\nabla q_0 \cdot \vec{r}_{0,f}\right),$$

   the Barth-Jespersen limiter can be written as

$$\Psi_0 = \min_f \begin{cases} \min\left(1, \frac{\phi_{\max}-\phi_0}{\Delta_f}\right) & \text{if } \Delta_f > 0 \\ \min\left(1, \frac{\phi_{\min}-\phi_0}{\Delta_f}\right) & \text{if } \Delta_f < 0 \\ 1 & \text{if } \Delta_f = 0 \end{cases}$$

   - Navigate to the *solvers/advection* folder.
   - Implement the following in the *AdvectionElements* class (located in *elements.py*):
     - *_make_compute_norm*: Compute the area (or volume in $3D$) of the shape i.e. region satisfying $\phi <= 0$ condition.
     - *_make_compute_fpts*: Assign cell center values to element faces.

- *_make_barth_jespersen*: Compute the limiter and fill the array(*lim* which holds Ψ values given above. Note that in the function maximum and minimum of face neighbors i.e. *fext* array is known.

- Implement the following in the *AdvectionIntInters* and *AdvectionBCInters* classes (located in *inters.py*):
  - *_make_minmax*: Compute the maximum and minimum of cell-centered values connected by the face.

3. **Implement the numerical flux functions:**
   - Implement the missing *_make_flux* function in the *AdvectionIntInters* and *AdvectionBCInters* classes (located in *inters.py*).
   - Implement the upwind and Local-Lax-Friedrich (Rusanov) flux functions in the *fsolver.py*.

4. **Run and Test the Code:**
   - Check the provided *advection.ini* file (in the *unittests/advection* folder) and the sample scripts *advection_test_mesh.py* to generate sample test meshes.
   - Execute the advection solver using the driver script *advection_test.py*.
   - Perform tests across:
     - Mesh types: Triangles, quadrilaterals, structured, unstructured.
     - Different test cases using various level set functions and velocity profiles.
     - Different numerical flux functions
     - Different time step sizes

5. **Error Computation and Reporting:**
   - Compute the area of the shapes represented by the level function. You can provide any level set benchmark problems to measure the area conservation.
   - Document your implementation process, results, and findings in a detailed report.
   - Ensure the report includes:
     - Implementation details.
     - Observations from tests on various meshes, problem definitions, and discretization methods.

6. **Submit Your Work:**
   - Provide the **complete code** (all folders) to allow replication of results.
   - Submit a **PDF report** (other formats are not accepted).

Good luck!