Project 2: Finite Volume Solvers

AE 623, Computational Fluid Dynamics II, Winter 2019
Runda Ji

I. Introduction

In order to simulate the inviscid compressible flow in a channel with a bump perturbation, first and second-order finite-volume methods (FVM) were implemented on a set of grids with different resolutions. After running the post-processing, the lift, drag and pressure coefficients as well as the Mach number field were obtained. A convergence study was also performed, showing that the convergence rate for the first-order FVM is approximately 1, while the convergence rate for the second-order FVM is around 2.

II. Methods

Within this section, we will briefly introduce the setup of boundary conditions as well as the time stepping, which will be applied when implementing the first and second-order FVM. The details about the implementation of the first and second-order FVM will be discussed in section III.

1. Boundary conditions

For this specific problem, we have three types of boundary conditions: inviscid wall, subsonic inflow and subsonic outflow.

(a) Inviscid wall

The key for implementing an inviscid wall boundary condition is to determine the "wall pressure" p_b,

$$p_b = (\gamma - 1)\left[\rho E^+ - \frac{1}{2}\rho^+ |\vec{v}^b|^2\right] \tag{1}$$

The boundary flux on the wall can be expressed as,

$$\hat{\mathbf{F}}^b = [0, p^b n_x, p^b n_y, 0]^T \tag{2}$$

where $\vec{n} = n_x\hat{x} + n_y\hat{y}$. The boundary velocity $\vec{v}^b$ is given by,

$$\vec{v}^b = \vec{v}^+ - (\vec{v}^+ \cdot \vec{n})\vec{n} \tag{3}$$

(b) Subsonic inflow

The inflow boundary condition includes three quantities: the flow angle α, the stagnation temperature T_t and the stagnation pressure p_t, where,

$$\begin{aligned} T_t &= 1 + \frac{\gamma - 1}{2} M_\infty^2 \\ p_t &= T_t^{\frac{\gamma}{\gamma - 1}} \end{aligned} \tag{4}$$

To begin with, we need to calculate the Riemann invariant based on the interior state.

$$J^+ = u_n^+ + \frac{2c^+}{\gamma - 1} \tag{5}$$

where $u_n^+ = \vec{v}^+ \cdot \vec{n}$ is the wall-normal velocity component pointing outward the domain. Then, we need to obtain the inflow Mach number M_b by solving,

$$\left(\gamma R T_t d_n^2 - \frac{\gamma-1}{2}(J^+)^2\right)(M^b)^2 + \left(\frac{4\gamma R T_t d_n}{\gamma-1}\right)M^b + \frac{4\gamma R T_t}{(\gamma-1)^2} - (J^+)^2 = 0 \tag{6}$$

where, $d_n = \vec{n}_{in} \cdot \vec{n}$ and $\vec{n}_{in} = [\cos(\alpha), \sin(\alpha)]$. After obtaining M_b, the exterior state can be defined using:

$$\begin{aligned}
T^b &= \frac{T_t}{1 + 0.5(\gamma-1)(M^b)^2} \\
p^b &= p_t\left(\frac{T^b}{T_t}\right)^{\frac{\gamma}{\gamma-1}} \\
\rho^b &= \frac{p^b}{RT^b} \\
c^b &= \sqrt{\frac{\gamma p^b}{\rho^b}} \\
\vec{v}^b &= M^b c^b \vec{n}_{in} \\
(\rho E)^b &= \frac{p^b}{\gamma-1} + \frac{1}{2}\rho^b|\vec{v}^b|^2
\end{aligned} \tag{7}$$

Finally, the boundary flux can be calculated using the Euler flux,

$$\hat{\mathbf{F}}^b = \vec{\mathbf{F}}(\mathbf{u}^b) \cdot \vec{n} \tag{8}$$

(c) Subsonic outflow

In order to compute the flux on the subsonic outflow boundary, we first need to calculate the exterior density,

$$\rho^b = \left(\frac{p^b}{S^+}\right)^{1/\gamma} \tag{9}$$

Now we can compute the boundary normal velocity u_n^b using J^+ obtained from equation (5) and $c^b = \sqrt{\gamma p^b/\rho^b}$. Specifically,

$$u_n^b = J^+ - \frac{2c^b}{\gamma-1} \tag{10}$$

The boundary velocity is defined as,

$$\vec{v}^b = \vec{v}^+ - (\vec{v}^+ \cdot \vec{n})\vec{n} + (\vec{v}^b \cdot \vec{n})\vec{n} \tag{11}$$

$(\rho E)^b$ is computed as $p^b/(\gamma-1) + \frac{1}{2}\rho^b|\vec{v}^b|^2$. After the boundary state $\mathbf{u}^b$ is determined, the boundary flux can be written as,

$$\hat{\mathbf{F}}^b = \vec{\mathbf{F}}(\mathbf{u}^b) \cdot \vec{n} \tag{12}$$

2. Time stepping

(a) Forward Euler

When implementing the first-order FVM, we can use the forward Euler scheme for time stepping, which can be expressed as,

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n - \frac{\Delta t_i^n}{A_i}\mathbf{R}_i(\mathbf{u}^n) \tag{13}$$

where the local time step is given by,

$$\text{local } \Delta t_i = \frac{\text{CFL } d_i}{|\bar{s}|_i} = \frac{2A_i\text{CFL}}{\sum_{e=1}^{3}|s|_{i,e}\Delta l_{i,e}} \tag{14}$$

where $|s|_{i,e}$, the maximum wave speed on the edge e of triangle i, will be computed when evaluating the numerical flux across the edge e.

(b) TVD RK2

As for the second-order FVM, we need use the TVD RK2 scheme for stability consideration,

$$\begin{aligned} \mathbf{u}_i^{FE} &= \mathbf{u}_i^n - \frac{\Delta t_i^n}{A_i}\mathbf{R}_i(\mathbf{u}^n) \\ \mathbf{u}_i^{n+1} &= \frac{1}{2}\left[\mathbf{u}_i^n + \mathbf{u}_i^{FE} - \frac{\Delta t_i^n}{A_i}\mathbf{R}_i(\mathbf{u}^{FE})\right] \end{aligned} \tag{15}$$

Note that the local time steps in both stages are the same.

III. Questions and Tasks

Task 1. Roe flux test

The Roe flux can be written as,

$$\hat{\mathbf{F}} = \frac{1}{2}(\mathbf{F}_L + \mathbf{F}_R) - \frac{1}{2}\begin{Bmatrix} |\lambda_3|\Delta\rho & +C_1 & \\ |\lambda_3|\Delta(\rho\vec{v}) & +C_1\vec{v} & +C_2\vec{n} \\ |\lambda_3|\Delta(\rho E) & +C_1 H & +C_2 u \end{Bmatrix} \tag{16}$$

where,

$$\begin{aligned} C_1 &= \frac{G_1}{c^2}(s_1 - |\lambda_3|) + \frac{G_2}{c}s_2 \\ C_2 &= \frac{G_1}{c}s_2 + (s_1 - |\lambda_3|)G_2 \\ G_1 &= (\gamma-1)\left(\frac{q^2}{2}\Delta\rho - \vec{v}\cdot\Delta(\rho\vec{v}) + \Delta(\rho E)\right) \\ G_2 &= -u\Delta\rho + \Delta(\rho\vec{v})\cdot\vec{n} \\ s_1 &= \frac{1}{2}(|\lambda_1| + |\lambda_2|) \\ s_2 &= \frac{1}{2}(|\lambda_1| - |\lambda_2|) \\ \Delta\mathbf{u} &= \mathbf{u}_R - \mathbf{u}_L \end{aligned} \tag{17}$$

Note that $u = \vec{v}\cdot\vec{n}$, $\mathbf{F}_L = \vec{\mathbf{F}}(\mathbf{u}_L)\cdot\vec{n}$ and $\mathbf{F}_R = \vec{\mathbf{F}}(\mathbf{u}_R)\cdot\vec{n}$.

Additionally, an entropy fix is also needed when implementing the Roe flux,

$$\text{if } |\lambda|_i < \epsilon \text{ then } \lambda_i = \frac{\epsilon^2 + \lambda_i^2}{2\epsilon}, \text{ for all } i \in [1,4] \tag{18}$$

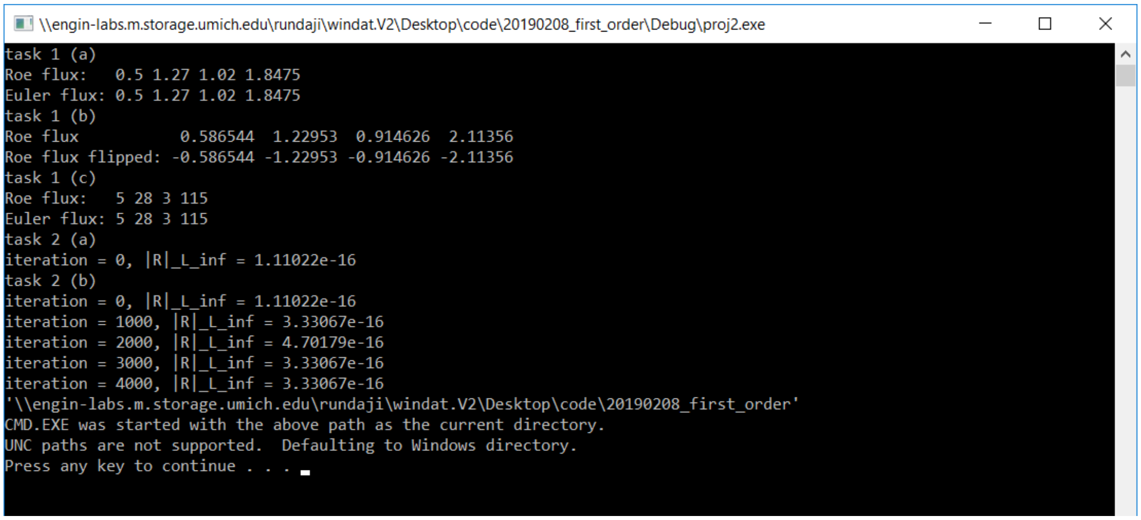
where we set $\epsilon = 0.1c$.

(a) Consistency check

Now consider an interior edge, let $\mathbf{u}_L$ and $\mathbf{u}_R$ represent the left and right-hand state across that edge. If $\mathbf{u}_L$ is identical to $\mathbf{u}_R$, then the Roe flux $\hat{\mathbf{F}}(\mathbf{u}_L, \mathbf{u}_L, \vec{n})$ should be the same as the analytical flux $\vec{\mathbf{F}}(\mathbf{u}_L)\cdot\vec{n}$. For simplicity, we set $\vec{n} = [1,1]^T$. At the same time, the state vector is defined as,

$$\mathbf{u}_L = \begin{bmatrix} 1 \\ M_L\cos(\alpha) \\ M_L\sin(\alpha) \\ \frac{1}{\gamma-1} + \frac{1}{2}M_L^2\gamma \end{bmatrix} \tag{19}$$

where we set $M_L = M_\infty = 0.5$, $\alpha = 0$. The results are shown in figure 1, where we can clearly see that the Roe flux is exactly the same as the Euler (analytical) flux when $\mathbf{u}_L = \mathbf{u}_R$.


\\engin-labs.m.storage.umich.edu\rundaji\windat.V2\Desktop\code\20190208_first_order\Debug\proj2.exe

```
task 1 (a)
Roe flux:   0.5 1.27 1.02 1.8475
Euler flux: 0.5 1.27 1.02 1.8475
task 1 (b)
Roe flux           0.586544  1.22953  0.914626  2.11356
Roe flux flipped: -0.586544 -1.22953 -0.914626 -2.11356
task 1 (c)
Roe flux:   5 28 3 115
Euler flux: 5 28 3 115
task 2 (a)
iteration = 0, |R|_L_inf = 1.11022e-16
task 2 (b)
iteration = 0, |R|_L_inf = 1.11022e-16
iteration = 1000, |R|_L_inf = 3.33067e-16
iteration = 2000, |R|_L_inf = 4.70179e-16
iteration = 3000, |R|_L_inf = 3.33067e-16
iteration = 4000, |R|_L_inf = 3.33067e-16
'\\engin-labs.m.storage.umich.edu\rundaji\windat.V2\Desktop\code\20190208_first_order'
CMD.EXE was started with the above path as the current directory.
UNC paths are not supported.  Defaulting to Windows directory.
Press any key to continue . . .
```

Figure 1. Test results for the Roe flux and first-order FVM.

(b) Flipping the direction

In order to implement this flipping test, we define $\mathbf{u}_R$ as,

$$\mathbf{u}_R = \begin{bmatrix} 1 \\ M_R \cos(\alpha) \\ M_R \sin(\alpha) \\ \frac{1}{\gamma-1} + \frac{1}{2} M_R^2 \gamma \end{bmatrix} \tag{20}$$

where $M_R = 1.0$. As shown in figure 1, when flipping $\mathbf{u}_L$ and $\mathbf{u}_R$, the Roe flux is also flipped. i.e. $\hat{\mathbf{F}}(\mathbf{u}_L, \mathbf{u}_R, \vec{n}) = -\hat{\mathbf{F}}(\mathbf{u}_R, \mathbf{u}_L, \vec{n})$

(c) State with supersonic normal velocity components

Within this test, we set $M_L = 5.0$ and $M_R = 4.0, 5.0, 6.0$. No matter what M_R we choose, the result under ”task 1 (c)” shown in figure 1 does not change. The Roe flux always returns the analytical flux from the upwind state.

Task 2. First-order finite-volume solver

For clarity, a flowchart for the first-order finite-volume solver is depicted in figure 2. Note that the key step for the finite-volume method is the evaluation of flux residual,

$$\mathbf{R}_i = \sum_{e=1}^{3} \hat{\mathbf{F}}(\mathbf{u}_i, \mathbf{u}_{N(i,e)}, \vec{n}_{i,e}) \Delta l_{i,e} \tag{21}$$

where $\hat{\mathbf{F}}$ is the numerical flux across the e^{th} edge of cell i, $\mathbf{u}_i$ and $\mathbf{u}_{N(i,e)}$ are the cell averaged states for cell i and the adjacent cell corresponding to the e^{th} edge, respectively. After obtaining the residual vector $\mathbf{R}_i$, we can update the state on each cell using equation (13). For all runs, we monitor the discrete L_∞ norm of the entire residual vector, taken over all components,

$$|\mathbf{R}|_{L_\infty} = \max |\mathbf{R}| \tag{22}$$

The solution is considered as converged when $|\mathbf{R}|_{L_\infty} < 10^{-7}$.

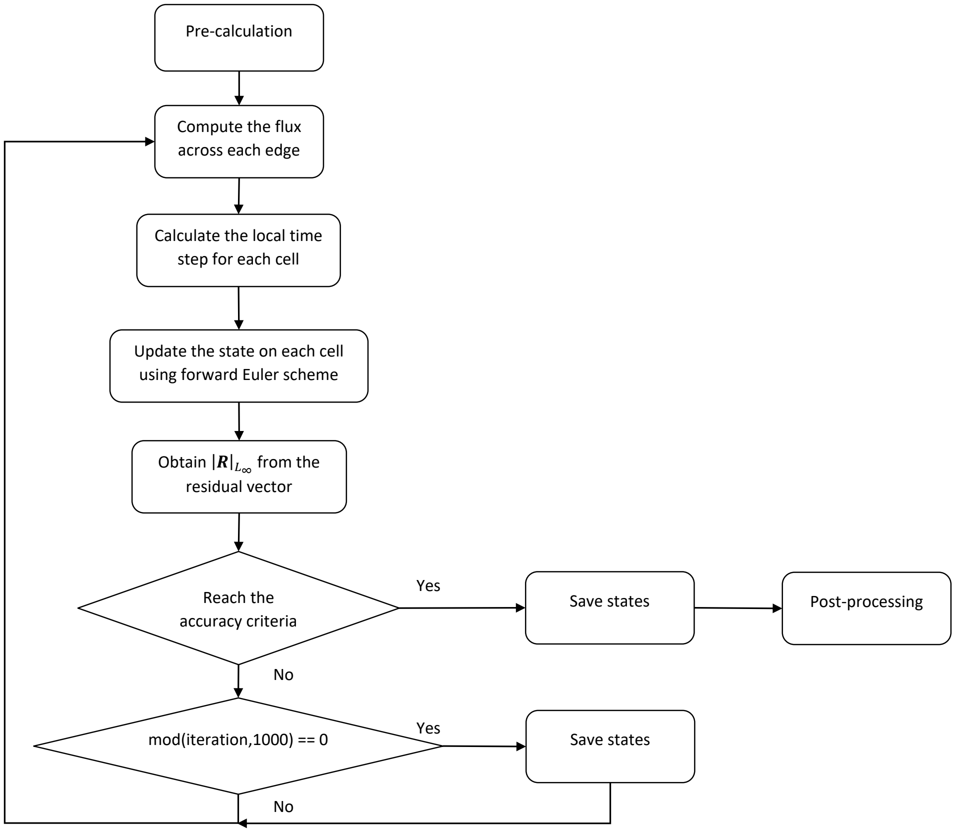

Pre-calculation
Compute the flux across each edge
Calculate the local time step for each cell
Update the state on each cell using forward Euler scheme
Obtain $|\boldsymbol{R}|_{L_\infty}$ from the residual vector
Reach the accuracy criteria
Yes
Save states
Post-processing
No
mod(iteration,1000) == 0
Yes
Save states
No

Figure 2. Flow chart of the first-order FVM.

(a) Free-stream test

When performing the free-stream test, we set all "exterior" state as $\mathbf{u}_\infty$,

$$\mathbf{u}_\infty = \begin{bmatrix} 1 \\ M_\infty \cos(\alpha) \\ M_\infty \sin(\alpha) \\ \frac{1}{\gamma-1} + \frac{1}{2} M_\infty^2 \gamma \end{bmatrix} \tag{23}$$

where $M_\infty = 0.5$, $\alpha = 0.5$ and $\gamma = 1.4$ for air. The result of free-stream test is shown in figure 1. After a single time step, the residual is in the order of 1e-16.

(b) Free-stream preservation test

After passing the free-stream test, we performed a free-steam preservation test. Where we run our code for 4000 time steps and check if the residual will stay around the machine precision. As shown in figure 1, during this free-steam preservation test, the residual norm hover around 4e-16, which can be considered as the machine precision.

(c) Actual boundary conditions

After the actual boundary conditions were implemented, we run our first-order FVM code iteratively until the accuracy criteria is satisfied. The residual norm history is plotted in figure 3. Note that we run our code on four meshes with different resolutions (bump0 through bump3). As we can see in figure 3, as the resolution increases, the iteration steps needed for convergence also increases. At the same time, the oscillation of L_∞ also becomes more significant when using a finer mesh.

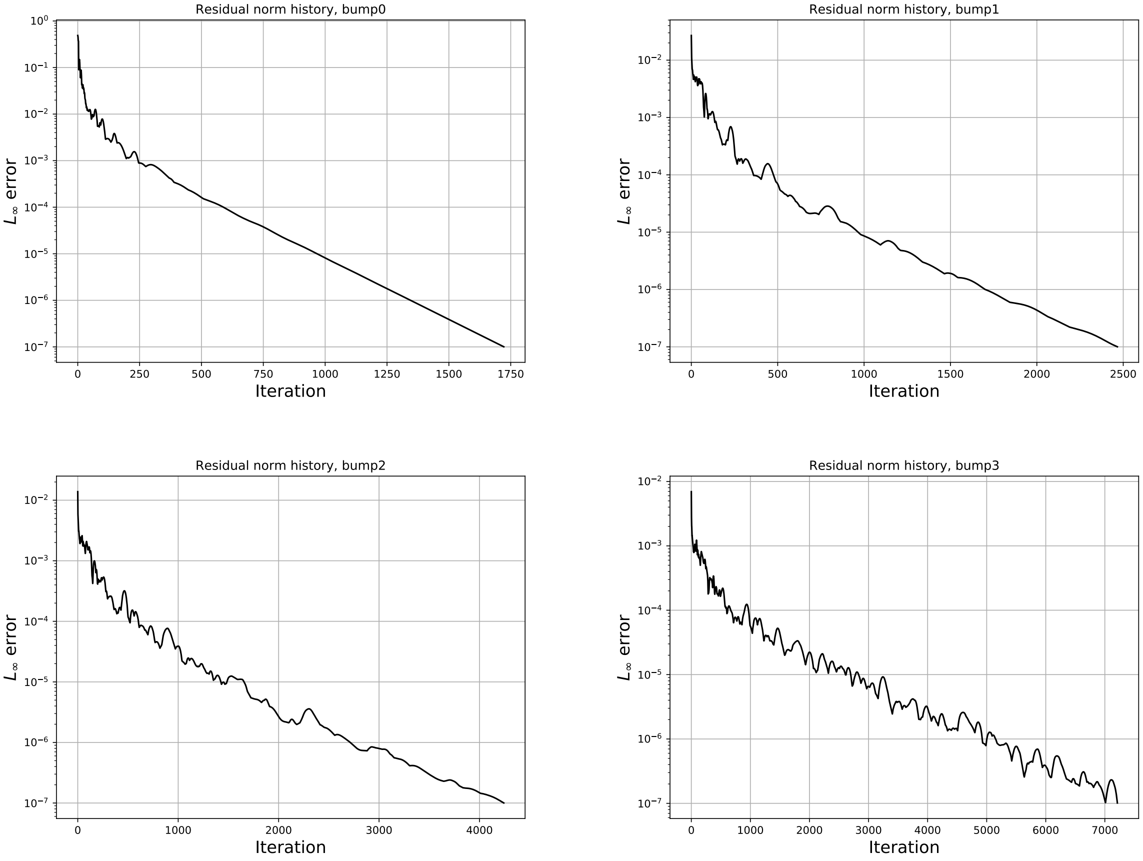


Figure 3. Residual norm history versus solver iterations, first-order FVM

Task 3. Second-order finite-volume solver

Similar to figure 2, a flowchart for the second-order finite-volume solver is depicted in figure 4. Note that the time stepping for the second-order FVM is slightly different from the first-order FVM. Specifically, within each time step, we need to evaluate the residual vector twice: the first time is finding $\mathbf{R}_i(\mathbf{u}^n)$ and the second time is finding $\mathbf{R}_i(\mathbf{u}^{FE})$. For convenient, the residual calculation is encoded in a separate function.

As for the flux evaluation, similar to equation (21), we have

$$\mathbf{R}_i = \sum_{e=1}^{3} \hat{\mathbf{F}}(\mathbf{u}_i^*, \mathbf{u}_{N(i,e)}^*, \vec{n}_{i,e})\Delta l_{i,e} \tag{24}$$

It is worth noting that the states $\mathbf{u}_i^*$ and $\mathbf{u}_{N(i,e)}^*$ are the states at the edge midpoint evaluated using the cell gradient, rather than the cell averaged states used in the first-order FVM. Specifically, the state at the edge midpoint can be written as,

$$\mathbf{u}^*(\vec{x}) = \mathbf{u}_i + \nabla\mathbf{u}\big|_i \cdot (\vec{x} - \vec{x}_i) \tag{25}$$

where $\vec{x}_i$ is the cell centroid and $\vec{x}$ is the location of a desired edge midpoint.

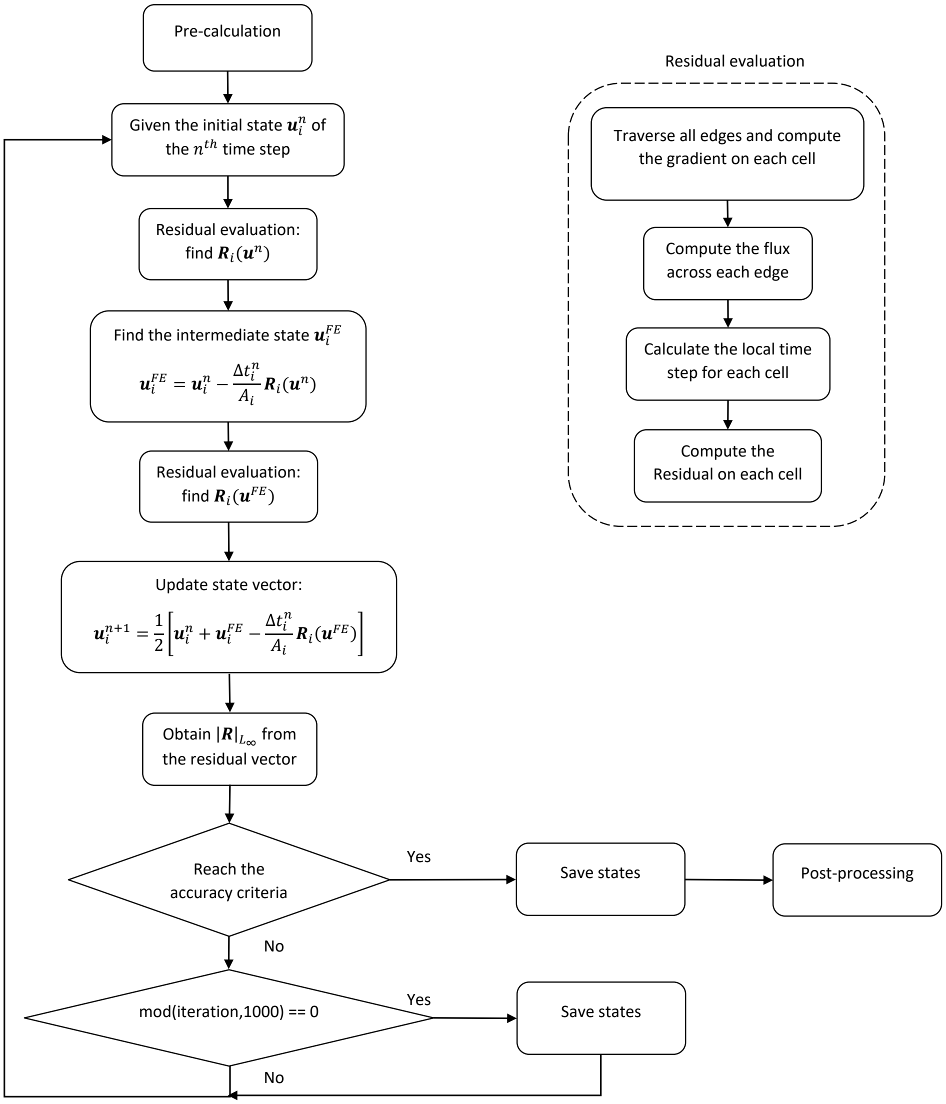
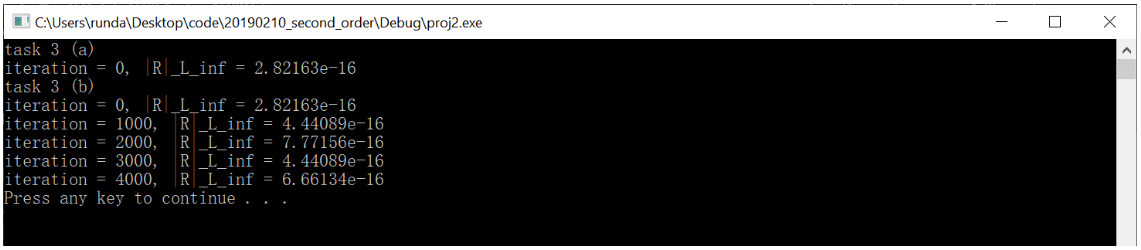

Pre-calculation
Given the initial state $\boldsymbol{u}_i^n$ of the n^{th} time step
Residual evaluation: find $\boldsymbol{R}_i(\boldsymbol{u}^n)$
Find the intermediate state $\boldsymbol{u}_i^{FE}$
$\boldsymbol{u}_i^{FE} = \boldsymbol{u}_i^n - \frac{\Delta t_i^n}{A_i}\boldsymbol{R}_i(\boldsymbol{u}^n)$
Residual evaluation: find $\boldsymbol{R}_i(\boldsymbol{u}^{FE})$
Update state vector:
$\boldsymbol{u}_i^{n+1} = \frac{1}{2}\left[\boldsymbol{u}_i^n + \boldsymbol{u}_i^{FE} - \frac{\Delta t_i^n}{A_i}\boldsymbol{R}_i(\boldsymbol{u}^{FE})\right]$
Obtain $|\boldsymbol{R}|_{L_\infty}$ from the residual vector
Reach the accuracy criteria
Yes
Save states
Post-processing
No
mod(iteration,1000) == 0
Yes
Save states
No
Residual evaluation
Traverse all edges and compute the gradient on each cell
Compute the flux across each edge
Calculate the local time step for each cell
Compute the Residual on each cell

Figure 4. Flow chart of the second-order FVM.

(a) Free-stream test

Similar to the free-stream test for the first-order FVM, we also performed a free-stream test to check our second-order FVM code. As shown in figure 5, the residual after one time step is around 3e-16, which can be reasonably considered as the machine precision.


C:\Users\runda\Desktop\code\20190210_second_order\Debug\proj2.exe

```
task 3 (a)
iteration = 0,  |R|_L_inf = 2.82163e-16
task 3 (b)
iteration = 0,  |R|_L_inf = 2.82163e-16
iteration = 1000,  |R|_L_inf = 4.44089e-16
iteration = 2000,  |R|_L_inf = 7.77156e-16
iteration = 3000,  |R|_L_inf = 4.44089e-16
iteration = 4000,  |R|_L_inf = 6.66134e-16
Press any key to continue . . .
```

Figure 5. Test results for the second-order FVM.

(b) Free-stream preservation test

After our second-order FVM code pass the free-stream test, we run it for 4000 time steps and check if the residual norm will hover around the machine precision. It shows that the residual norm is always in the order of 10^{-16} indicating that our second-order FVM code has pass the free-stream preservation test.

(c) Actual boundary conditions

When running our simulation with the actual boundary conditions, we plotted the residual norm history versus the solver iterations in figure 6. Comparing with figure 3, we can find that the second-order FVM takes less iterations to reach a convergence state when using a coarse grid (bump0). However, when using finer grids, the second-order FVM needs more iteration steps to converge.

Additionally, the second-order FVM is more oscillatory comparing with the first-order FVM. Specifically, if we focus on the residual norm history of bump0, we can clearly see that L_∞ converges as a straight line within figure 3 after 250 iterations. However in figure 6, L_∞ keep oscillating as it converges to 10^{-7}.

So why is the second-order FVM more oscillatory than the first-order FVM? A possible explanation is that the second-order FVM is more accurate on propagating waves and it is not as dissipative as the first-order method. Consider the error waves propagating within the domain, rather than being damped out, these waves can last for quite a long time when using the second-order FVM. Those waves bouncing inside the computational domain are likely to cause the oscillation of the residual vector $\mathbf{R}$ and residual norm $|\mathbf{R}|_{L_\infty}$.

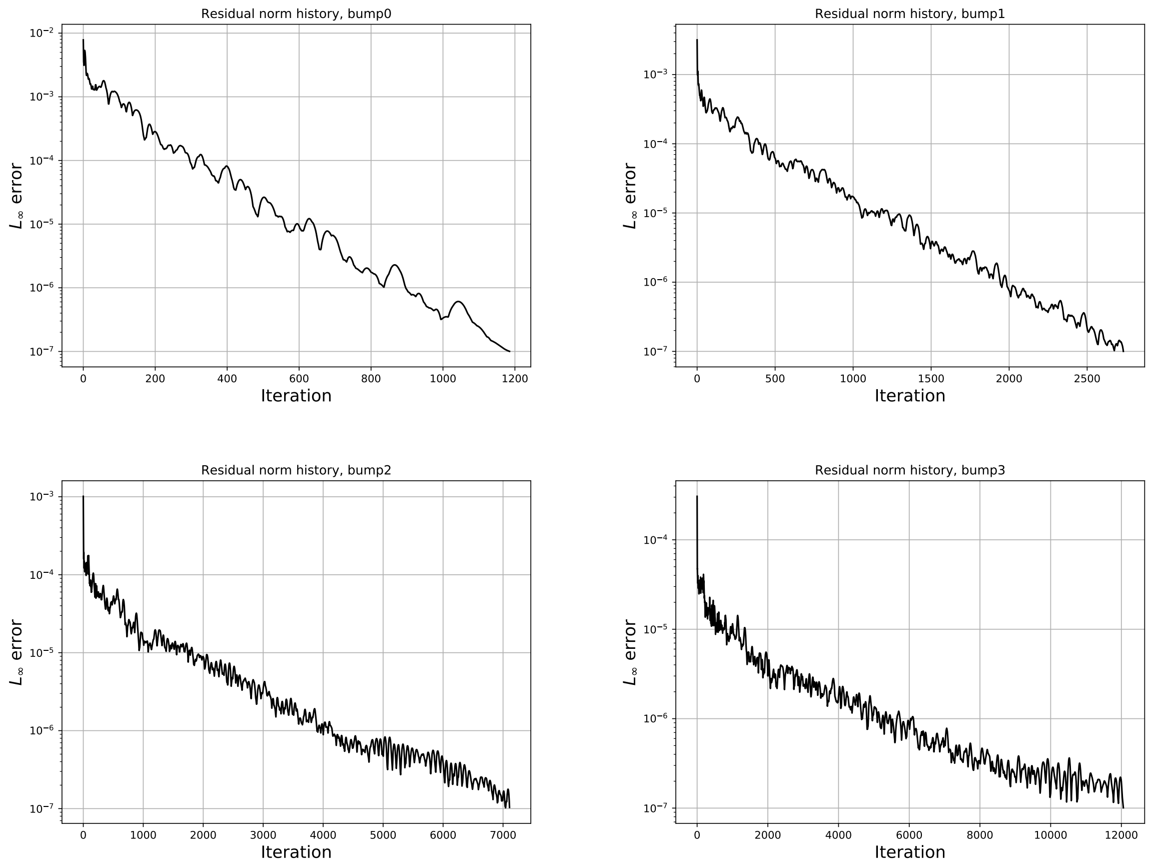


Figure 6. Residual norm history versus solver iterations, second-order FVM.

Task 4. Post-processing

(a) Convergence study

As aforementioned, we run our simulation on four grids with different resolutions: bump0, bump1, bump2 and bump3. The following three quantities are calculated for each mesh and compared with the "exact" solution in order to evaluate the error of the first and second-order FVM.

1. **Lift coefficient**
The lift coefficient on the bottom wall can be computed as,

$$c_l = \frac{\int_{\text{bottom}}(p - p_\infty)n_y dl}{\frac{\gamma}{2}p_\infty M_\infty^2 h} \tag{26}$$

where $h = 0.0625$ is the height of bump.

2. **Drag coefficient**
The drag coefficient is defined as,

$$c_d = \frac{\int_{\text{bottom}}(p - p_\infty)n_x dl}{\frac{\gamma}{2}p_\infty M_\infty^2 h} \tag{27}$$

3. **Entropy error**
The integral of the entropy error over the domain Ω is given by,

$$E_s = \sqrt{\frac{\int_\Omega (s/s_t - 1)^2 d\Omega}{\int_\Omega d\Omega}} \tag{28}$$

where the entropy is $s = p/\rho^\gamma$, and $s_t = p_t/\rho_t^\gamma$ is the stagnation entropy at the inflow. Note that $\rho_t = p_t/(RT_t)$.

Note that the "exact" values are $c_{l,\text{exact}} = 1.537095$, $c_{d,\text{exact}} = 2.94278 \times 10^{-6}$ and $E_{s,\text{exact}} = 0$. As shown in figure 7, the red, green and blues represent the error of c_l, c_d and E_s. At the same time, the solid and dash lines corresponds to the first and second-order FVM, respectively.

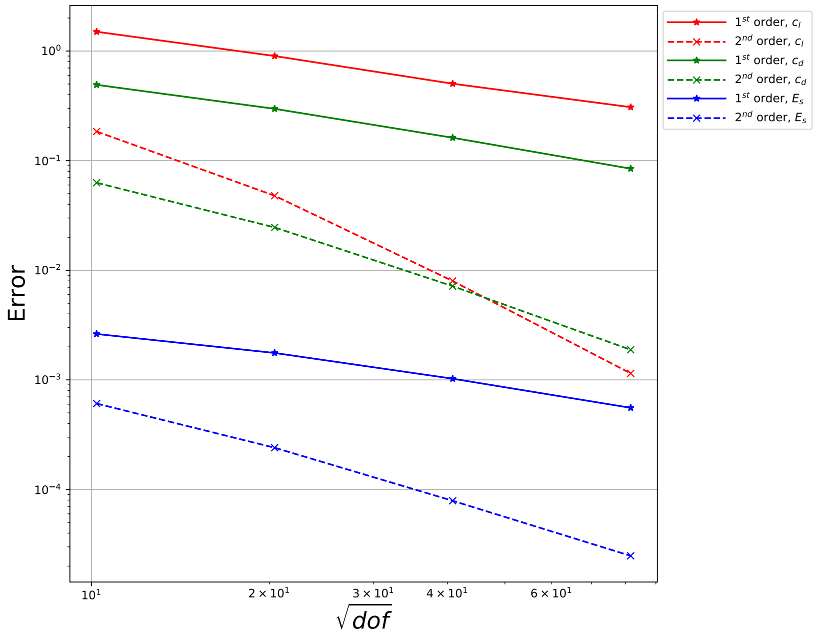


Figure 7. Convergence study for the first and second-order FVM.

The convergence rates calculated using the data points from bump3 and bump2 are shown in table 1, where we can roughly conclude that the convergence rate for the first-order FVM is nearly 1 and the rate for the second-order FVM is around 2. It is worth noting that other than the leading term, which is $\mathcal{O}(\Delta x)$ for first-order FVM and $\mathcal{O}(\Delta x^2)$ for the second-order FVM, there are also high order terms which may impact the convergence rate. It is a possible reason explaining why the convergence rates are not exactly what we anticipated.

Table 1. Convergence rate of c_l, c_d and E_s.

Quantity	1^{st}-order FVM	2^{nd}-order FVM
c_l	0.71079	2.79961
c_d	0.93711	1.92758
E_s	0.88266	1.66819

(b) Pressure coefficient

The pressure coefficient is defined as,

$$c_p = \frac{p - p_\infty}{\frac{\gamma}{2} p_\infty M_\infty^2} \tag{29}$$

As shown in figure 8 and figure 9, we plot the pressure-coefficient distribution, i.e. $-c_p$ versus x, over the bottom wall.

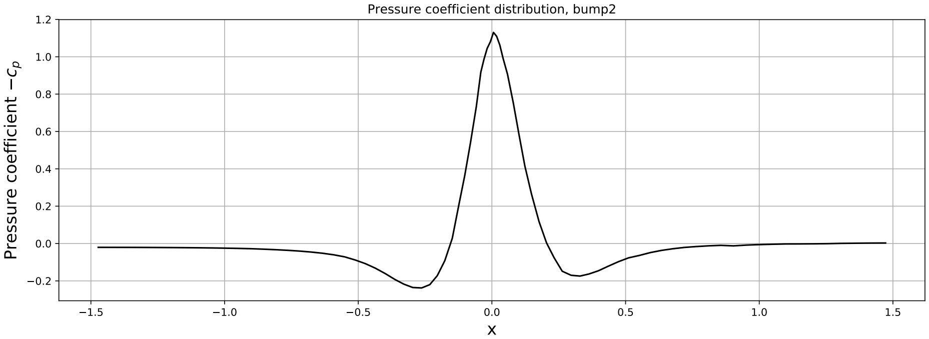


Figure 8. Pressure coefficient distribution, first-order FVM.

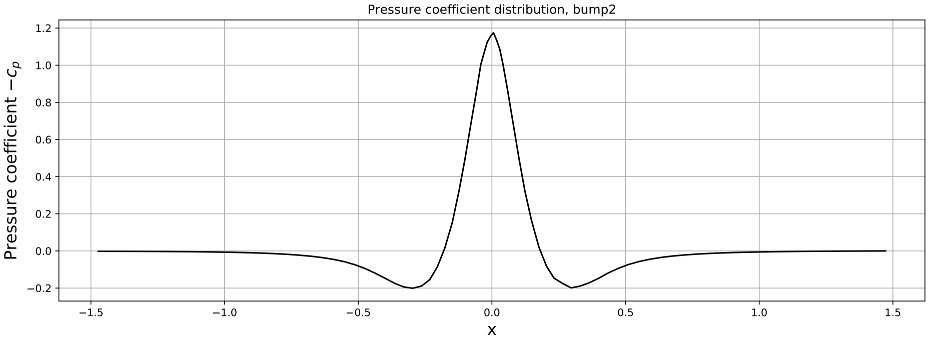


Figure 9. Pressure coefficient distribution, second-order FVM.

(c) Mach number fields

The Mach number fields obtained using first-order FVM and second-order FVM are depicted in figure 10 and figure 11, respectively.

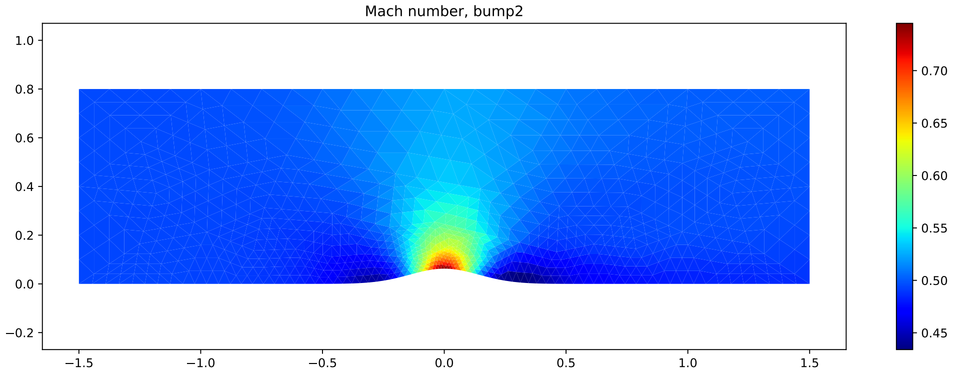


Figure 10. Mach number fields, first-order FVM.

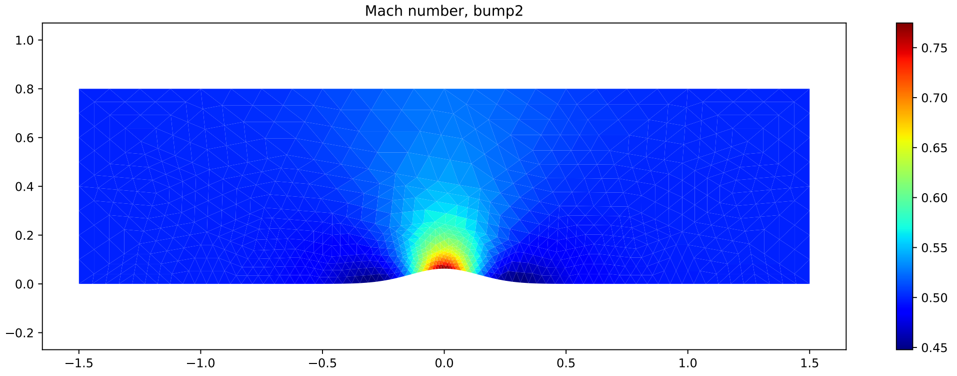


Figure 11. Mach number fields, second-order FVM.