Project 3: Discontinuous Galerkin Solver

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, the discontinuous Galerkin (DG) finite-element method (FEM) has been implemented in this project. The solution order p varies from 0 to 2, and the impact of p on the final solution has been discussed. Curved elements with geometry order $q = 3$ were applied near the bottom wall in order to correctly match the solution order p. A convergence study has been performed using a sequence of meshes with different resolutions. The convergence rates corresponding to different solution orders have been calculated.

II. Methods

Boundary conditions implemented in this project are exactly the same as the previous project. Hence, we will not repeat the set up for boundary conditions here. Within this section, we would like to briefly introduce three "updates" comparing with the previous project: 1. FEM, 2. curved element and 3. fourth order Runge-Kutta (RK4) time-marching scheme.

A. FEM

Different with the finite volume method (FVM) in previous project, where we used the cell average and gradient to represent the sate inside an element, the FEM use basis functions to represent the variation of state inside an element. Specifically,

$$\mathbf{u}(\vec{x},t) = \sum_{m=1}^{N_e}\sum_{j=1}^{n_p} \mathbf{U}_{m,j}(t)\phi_{m,j}(\vec{x}) \tag{1}$$

where $\mathbf{u}(\vec{x},t)$ is the state vector, N_e is the total element number, n_p is the number of basis functions in each element.

$$n_p = (p+1)(p+2)/2 \tag{2}$$

$\mathbf{U}_{m,j}(t)$ in equation (1) is the coefficient for the j^{th} basis function in element m. While $\phi_{m,j}(\vec{x})$ is the j^{th} basis function of element m evaluated at location $\vec{x}$.

1. Weak Form

Recall the advection equation for 2D,

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \vec{\mathbf{F}} = 0 \tag{3}$$

Multiply the advection equation by test (basis) function $\phi_{k,i}$ and then integrate by part, we can obtain the weak from for 2D advection, i.e. equation (4.3.20) in the lecture note.

$$\int_{\Omega_k} \phi_{k,i}\frac{\partial \mathbf{u}}{\partial t} d\Omega - \int_{\Omega_k} \nabla\phi_{k,i}\cdot\vec{\mathbf{F}} + \int_{\partial\Omega_k} \phi^+_{k,i}\hat{\mathbf{F}}(\mathbf{u}^+,\mathbf{u}^-,\vec{n})dl = 0 \tag{4}$$

Substituting equation (1) into equation (4) gives,

$$\mathbf{M}\frac{d\mathbf{U}}{dt} + \mathbf{R}(\mathbf{U}) = 0 \tag{5}$$

where $\mathbf{M}$ is the mass matrix, which is a block diagonal matrix. Consider the k^{th} block corresponding to the k^{th} element in the mesh,

$$\begin{aligned}[M_k]_{i,j} &= \int_{\Omega_k} \phi_{k,i}\phi_{k,j} d\Omega \\ &= \int_{T_k} \phi_{k,i}(\xi,\eta)\phi_{k,j}(\xi,\eta) J d\xi d\eta\end{aligned} \tag{6}$$

where $J = \det(\mathbf{J})$, $\mathbf{J}$ is the Jacobin matrix of element k. Since the mass matrix is block diagonal, the update equation can be rewritten as,

$$\mathbf{M}_k \frac{d\mathbf{U}_k}{dt} + \mathbf{R}_k(\mathbf{U}_k) = 0 \tag{7}$$

which indicates that each element is being updated independently. So where is the term representing the interaction between adjacent elements? It is included the second term in equation (8),

$$\mathbf{R}_{k,i} = \underbrace{-\int_{\Omega_k} \nabla\phi_{k,i} \cdot \vec{\mathbf{F}}}_{\text{interior contribution}} \underbrace{+\int_{\partial\Omega_k} \phi^+_{k,i} \hat{\mathbf{F}}(\mathbf{u}^+, \mathbf{u}^-, \vec{n}) dl}_{\text{edge contribution}} \tag{8}$$

2. Numerical Integration - Quadrature Rules

In order to evaluate the integration in equation (6) and equation (8), we need to introduce the quadrature rules. For a 2D integration inside an element, such as equation (6), it can be numerically evaluated by,

$$[M_k]_{i,j} = \sum_{g=1}^{N_g} \phi_{k,i}(\vec{\xi_g})\phi_{k,j}(\vec{\xi_g}) J(\vec{\xi_g}) w_g \tag{9}$$

For linear elements, $J(\vec{\xi_g})$ is a constant and can be pull out of the integral. However, for curved elements $J(\vec{\xi_g})$ varies as $\vec{\xi_g}$ changes. We will discuss this in detail later on. Similarly, the first term in equation (8) can be rewritten as,

$$\mathbf{R}_{k,i} - = \sum_{g=1}^{N_g} \left.\frac{\partial\phi_{k,i}}{\partial\vec{\xi}}\right|_{\vec{\xi_g}} \cdot \mathbf{J}^{-1}\Big|_{\vec{\xi_g}} \vec{\mathbf{F}}(\vec{\xi_g}) w_g \tag{10}$$

Note that $\vec{\xi_g}$ and w_g are the coordinates of quadrature points (in reference space) and the corresponding weights, respectively.

As for the integration over an edge, such as the edge contribution term in equation (8), it can be rewritten as,

$$\begin{aligned}\mathbf{R}_{kL,i} + &= \sum_{g=1}^{N_g} \phi^+_{kL,i}(s_g) \hat{\mathbf{F}}(\mathbf{u}_{L,g}, \mathbf{u}_{R,g}, \vec{n_g}) J_{edge} w_g \\ \mathbf{R}_{kR,i} - &= \sum_{g=1}^{N_g} \phi^+_{kR,i}(s_g) \hat{\mathbf{F}}(\mathbf{u}_{L,g}, \mathbf{u}_{R,g}, \vec{n_g}) J_{edge} w_g\end{aligned} \tag{11}$$

where $\vec{n_g}$ is the normal vector at the g^{th} quadrature point, pointing from the left element towards the right element.

It is worth noting that we need to be careful while looping over the 1D quadrature points over an interior edge. As shown in figure 1, the nodes in both left and right elements are stored counter-clockwisely. In order to match the traversing order of the quadrature points represented by the dash arrow, we need to "inverse" those quadrature points belongs to the right element, while keeping the counter-clockwise order for the quadrature points belongs to the left element.

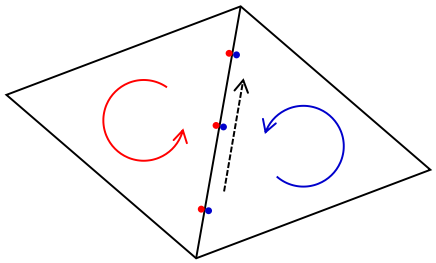

Figure 1. Quadrature points on an interior edge.

3. Full-Order Basis Function

The basis function can be computed using the provided MATLAB function "TriLagrange2D.m". The $p = 1$ basis function can be written as,

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ \xi \\ \eta \end{bmatrix} \tag{12}$$

The $p = 2$ basis function can be written as,

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \end{bmatrix} = \begin{bmatrix} 1 & -3 & 2 & -3 & 4 & 2 \\ 0 & 4 & -4 & 0 & -4 & 0 \\ 0 & -1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & -4 & -4 \\ 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & -1 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ \xi \\ \xi^2 \\ \eta \\ \xi\eta \\ \eta^2 \end{bmatrix} \tag{13}$$

We also need $q = 3$ basis function for implementing the curved element,

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \\ \phi_{10} \end{bmatrix} = \begin{bmatrix} 1 & -5.5 & 9 & -4.5 & -5.5 & 18 & -13.5 & 9 & -13.5 & -4.5 \\ 0 & 9 & -22.5 & 13.5 & 0 & -22.5 & 27 & 0 & 13.5 & 0 \\ 0 & -4.5 & 18 & -13.5 & 0 & 4.5 & -13.5 & 0 & 0 & 0 \\ 0 & 1 & -4.5 & 4.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0 & 9 & -22.5 & 13.5 & -22.5 & 27 & 13.5 \\ 0 & 0 & 0 & 0 & 0 & 27 & -27 & -0 & -27 & 0 \\ 0 & 0 & 0 & 0 & 0 & -4.5 & 13.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -4.5 & 4.5 & 0 & 18 & -13.5 & -13.5 \\ 0 & 0 & 0 & 0 & 0 & -4.5 & 0 & 0 & 13.5 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -4.5 & 0 & 4.5 \end{bmatrix} \begin{bmatrix} 1 \\ \xi \\ \xi^2 \\ \xi^3 \\ \eta \\ \xi\eta \\ \xi^2\eta \\ \eta^2 \\ \xi\eta^2 \\ \eta^3 \end{bmatrix} \tag{14}$$

B. Curved Element

1. Jacobin Matrix

For linear element ($q = 1$), the Jacobin matrix $\mathbf{J}$ is constant inside an element.

$$\mathbf{J} = \begin{bmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{bmatrix} \tag{15}$$

where $\vec{x}_i$ $(i = 1, 2, 3)$ are the coordinates of triangle vertices in the global space. However, within a curved element, the Jacobin matrix $\mathbf{J}$ is no longer a constant.

$$\begin{aligned}\mathbf{J}(\vec{\xi}) = \frac{\partial \vec{x}}{\partial \vec{\xi}} &= \sum_{i=1}^{N_q} \vec{x}_i \frac{\partial \phi_i}{\partial \vec{\xi}}(\vec{\xi}) \\ &= \begin{bmatrix} \sum x_i \frac{\partial \phi_i}{\partial \xi}(\vec{\xi}) & \sum x_i \frac{\partial \phi_i}{\partial \eta}(\vec{\xi}) \\ \sum y_i \frac{\partial \phi_i}{\partial \xi}(\vec{\xi}) & \sum y_i \frac{\partial \phi_i}{\partial \eta}(\vec{\xi}) \end{bmatrix}\end{aligned} \tag{16}$$

Note that the summation is from $i = 0$ to $i = N_q$. For $q = 3$, $N_q = (q+1)(q+2)/2 = 10$.

2. Normal Vector

In order to calculate the normal vector, we need to first consider the tangential vector,

$$\begin{aligned}\frac{d\vec{x}_{edge}}{d\sigma} &= \frac{d\vec{x}}{d\xi} \cdot \frac{d\xi}{d\sigma} + \frac{d\vec{x}}{d\eta} \cdot \frac{d\eta}{d\sigma} \\ &= \vec{t} \cdot \frac{ds}{d\sigma}\end{aligned} \tag{17}$$

where $\vec{t}$ can be consider as normalized tangential vector, $\frac{ds}{d\sigma}$ is the edge Jacobin J_{edge}.

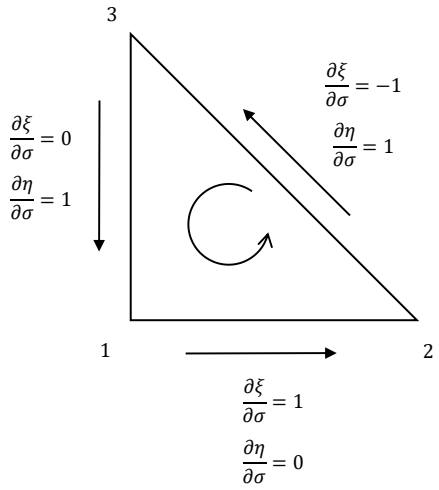

Figure 2. Evaluating $\frac{d\xi}{d\sigma}$ and $\frac{d\eta}{d\sigma}$ in reference space.

Therefore, the normal vector can be expressed as,

$$\vec{n} = \vec{t} \times \hat{k} = [t_y, -t_x] \tag{18}$$

C. RK4 Time-Stepping

For stability consideration, we use RK4 for time-stepping in this project. The RK4 time-marching scheme can be written as follows,

Step 1:

$$\begin{aligned}&\mathbf{u}^n \Rightarrow \mathbf{R}^0(\mathbf{u}^n) \\ &\mathbf{u}^0 = \mathbf{u}^n - \frac{1}{2}\frac{\Delta t_i^n}{A_i}\mathbf{R}^0(\mathbf{u}^n)\end{aligned} \tag{19}$$

Step 2:

$$\begin{aligned}&\mathbf{u}^0 \Rightarrow \mathbf{R}^1(\mathbf{u}^0) \\ &\mathbf{u}^1 = \mathbf{u}^n - \frac{1}{2}\frac{\Delta t_i^n}{A_i}\mathbf{R}^1(\mathbf{u}^0)\end{aligned} \tag{20}$$

Step 3:

$$\begin{aligned} \mathbf{u}^1 &\Rightarrow \mathbf{R}^2(\mathbf{u}^1) \\ \mathbf{u}^2 &= \mathbf{u}^n - \frac{\Delta t_i^n}{A_i}\mathbf{R}^2(\mathbf{u}^1) \end{aligned} \tag{21}$$

Step 4:

$$\mathbf{u}^2 \Rightarrow \mathbf{R}^3(\mathbf{u}^2) \tag{22}$$

Step 5:

$$\mathbf{u}^{n+1} = \mathbf{u}^n - \frac{\Delta t_i^n}{6}\mathbf{M}_i^{-1}(\mathbf{R}^0 + 2\mathbf{R}^1 + 2\mathbf{R}^2 + \mathbf{R}^3) \tag{23}$$

where $\mathbf{u}^n$ and $\mathbf{u}^{n+1}$ are the state vectors at the n^{th} and $(n+1)^{th}$ time step, respectively. $\mathbf{u}^0, \mathbf{u}^1$ and $\mathbf{u}^2$ are the intermediate states. $\mathbf{M}_i^{-1}$ is the inverse mass matrix for the i^{th} element. $\mathbf{R}^0, \mathbf{R}^1, \mathbf{R}^2$ and $\mathbf{R}^3$ are the residual vectors of corresponding states.

The local time step Δt_i^n can be written as,

$$\text{local } \Delta t_i = \frac{\text{CFL } d_i}{|\bar{s}|_i} = \frac{2A_i \text{ CFL}}{\sum_{e=1}^{3} |s|_{i,e} \Delta l_{i,e}} \tag{24}$$

where $|s|_{i,e}$ is the maximum wave speed on the edge e of triangle i. It will be computed when evaluating the numerical flux across the edge e. For maximum efficiency, the CFL number should be proportional to $1/(p+1)$. In our computation, we set,

$$\text{CFL} = \frac{0.85}{p+1} \tag{25}$$

III. Questions and Tasks

Task 1. Curved Meshes

For comparison, we first show the original **bump1.gri** in figure 3. Note that all elements in figure 3 are linear elements i.e. $q = 1$. A zoomed-in view showing the details near the bump is depicted in figure 4.

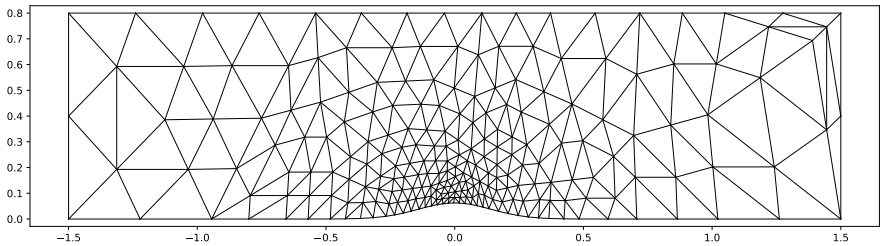

Figure 3. Mesh bump1.gri with linear elements ($q = 1$).

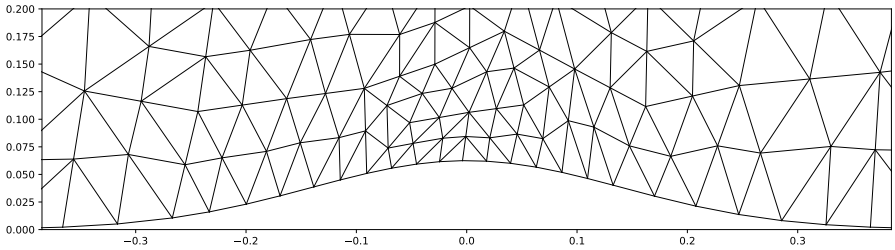

Figure 4. Mesh bump1.gri with linear elements ($q = 1$), zoomed in near the bump.

After the elements adjacent to the bottom wall were curved, we can obtain the **bump1_curved.gri** mesh. Note that the blue lines in figure 5 and figure 6 are only used for illustrating the positions of higher order nodes inside each element. In other words, those "interior" edges do not physically exist.

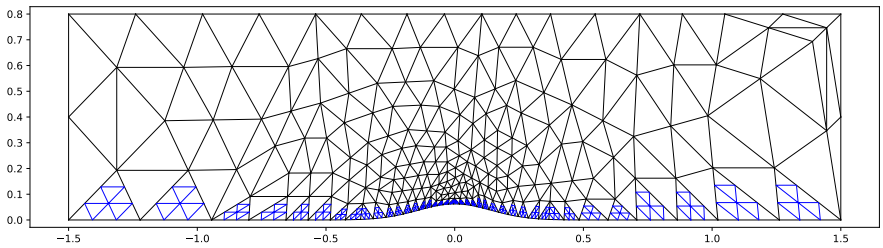

Figure 5. Mesh bump1_curved.gri with curved elements adjacent to the bottom wall ($q = 3$).

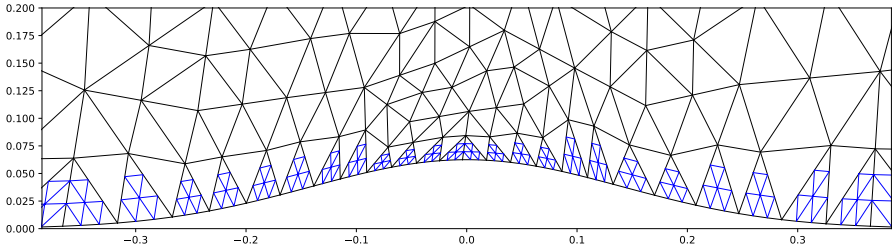

Figure 6. Mesh bump1_curved.gri with curved elements adjacent to the bottom wall ($q = 3$), zoomed in near the bump.

Task 2. DG Solver

For clarity, a flowchart for DG solver is depicted in figure 7. Note that the evaluation of residual vector in FEM is quite different with FVM. Specifically, the residual update includes two parts: contribution from element interior and contribution across the edges.

Recall equation (10), in order to determine the interior contribution, we need to loop over all quadrature points inside an element. At each quadrature point (ξ, η), we need to evaluate the gradient of basis function $\frac{\partial \phi_{k,i}}{\partial \vec{\xi}}\big|_{\vec{\xi}_g}$ as well as the inverse of Jacobin matrix $\mathbf{J}^{-1}\big|_{\vec{\xi}_g}$. At the same time, we also need to figure out the state at the current quadrature point using,

$$\mathbf{u}(\xi, \eta) = \sum_{j=1}^{n_p} \mathbf{U}_j \phi_j(\xi, \eta) \tag{26}$$

After the state at quadrature point is obtained, we can further evaluate the analytical (Euler) flux at these quadrature points,

$$\vec{\mathbf{F}}(\xi, \eta) = \vec{\mathbf{F}}(\mathbf{u}(\xi, \eta)) \tag{27}$$

Eventually, we can use equation (10) to sum over all quadrature points and figure out the element interior contribution for each cell. Similarly, we can determine the edge contribution using equation (11). Note that for interior edges, the element on both side will be updated, which means that we only need to compute the Roe flux once for each (interior) edge.

For all runs, we monitor the discrete L_∞ norm of the entire residual vector, taken over all components,

$$|\mathbf{R}|_{L_\infty} = \max |\mathbf{R}| \tag{28}$$

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

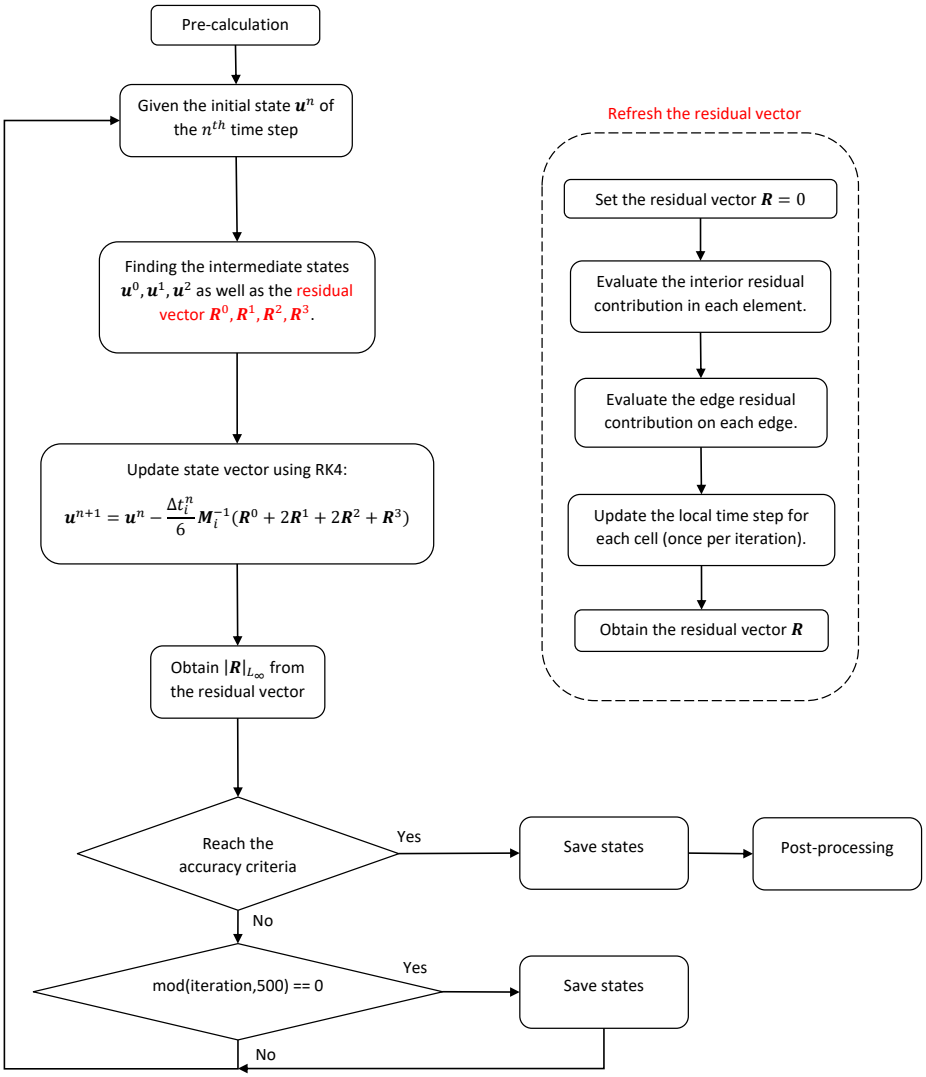

Figure 7. Flow chart of DG.

1. *Free-stream Test*

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

$$\mathbf{u}_\infty = \left[1, M_\infty \cos(\alpha), M_\infty \sin(\alpha), \frac{1}{\gamma - 1} + \frac{1}{2} M_\infty^2 \gamma\right]^T \tag{29}$$

where $M_\infty = 0.5$, $\alpha = 0.5$ and $\gamma = 1.4$ for air.

Before running the free-stream test on the **bump0** mesh, we first run the free-stream test on simple meshes shown in figure 8 for debugging purpose.

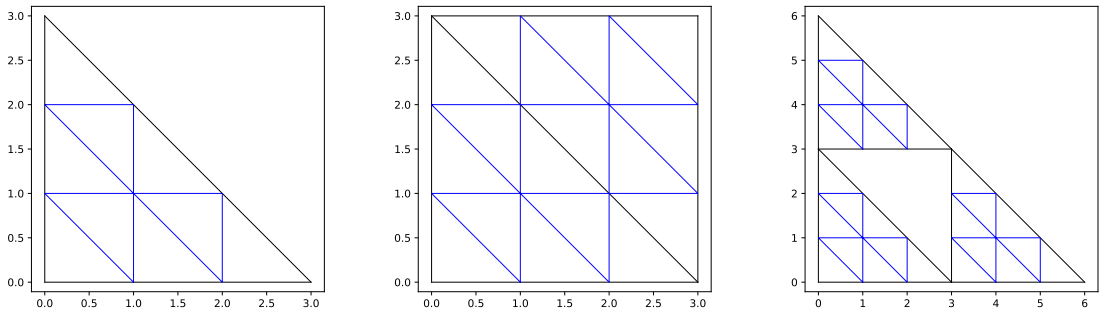

Figure 8. Simple meshes for debugging.

By slightly displacing those higher-order nodes, we can easily check if our implementation of the curved elements is correct. After passing the free stream test on those simple meshes, we can further run our code on mesh **bump0**. The residual $|\mathbf{R}|_{L_\infty}$ is in the order of 1e-15 after a single time step, which can be reasonably considered as machine precision.

2. Free-stream Preservation Test

After passing the free-stream test, we performed a free-steam preservation test. Where we run our code for 1000 time steps and check if the residual will stay around the machine precision. As shown in figure 9, during this free-steam preservation test, the residual norm hover around 3e-16, indicating that our code can pass this preservation test successfully.

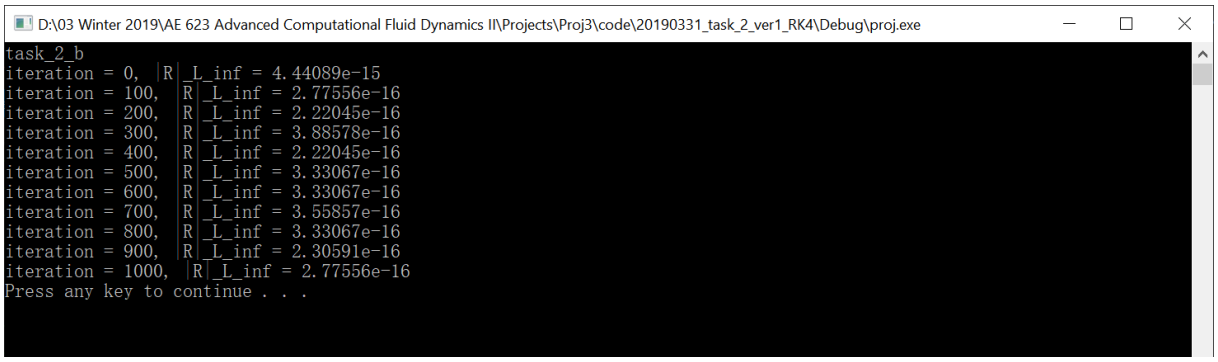

Figure 9. Free-stream preservation test on the bump0 mesh with curved elements ($p = 2, q = 3$).

3. Actual Boundary Conditions

After the actual boundary conditions were implemented, we run our DG code iteratively until the accuracy criteria is satisfied. The $|\mathbf{R}|_{L_\infty}$ history versus the solver iteration is plotted in figure 10. If we compare the residual history for different solution order p, we can see that DG with higher p takes more iteration steps to reach a convergence solution. At the same time, the oscillation of $|\mathbf{R}|_{L_\infty}$ also becomes more significant for DG with higher solution order.

So how should we explain the oscillatory behavior for $p = 2$ solution? Since the FEM with $p = 2$ is more accurate on propagating waves, the error waves propagating within the domain can last for quite a long time when $p = 2$, rather than being quickly damped out when $p = 1$ or $p = 0$. Those error waves bouncing inside the computational domain cause the oscillation of $|\mathbf{R}|_{L_\infty}$ as well as the numerical instability.

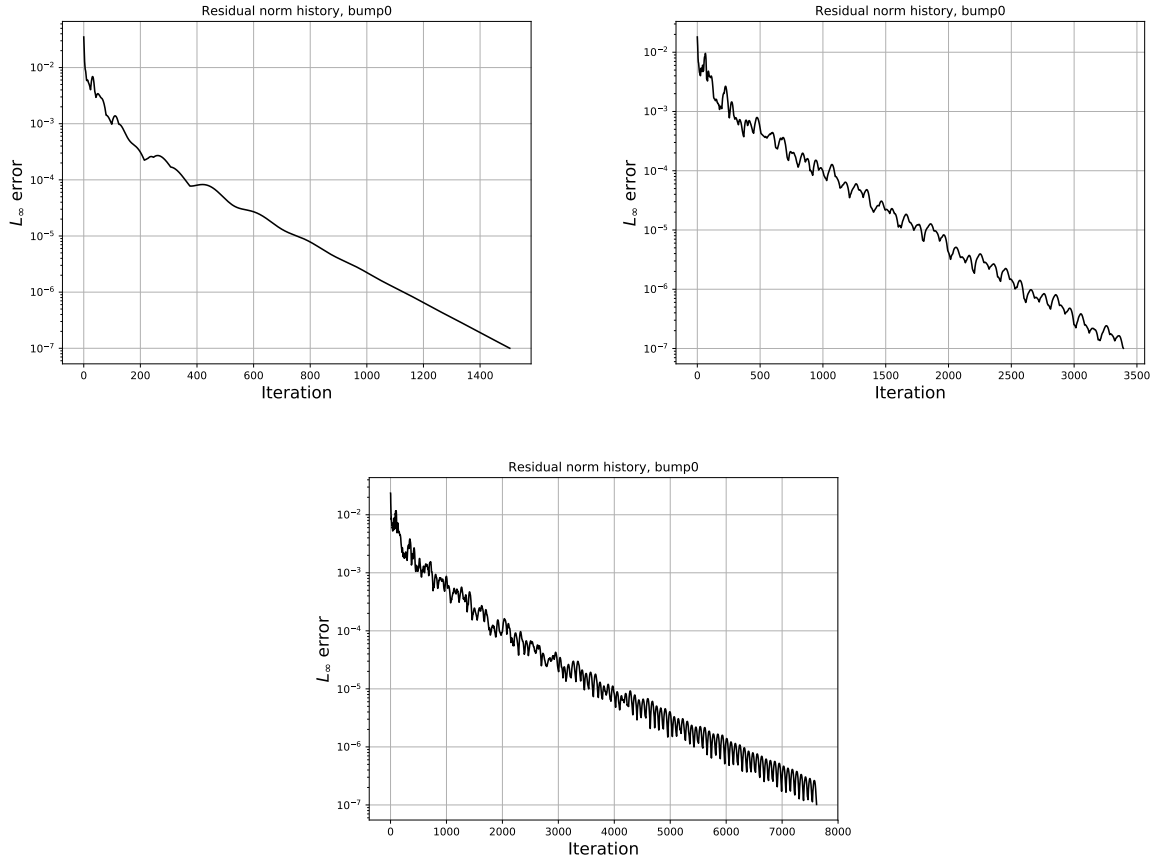

Figure 10. Residual norm history versus solver iterations. For curved bump0 with solution order $p = 0, 1, 2$

Task 3. Post-processing

4. Convergence Study

In order to implement the convergence study, we run our code on our uniform-refinement mesh sequence: **bump0**, **bump1** and **bump2**. Similar to previous project, the lift coefficient c_l, drag coefficient c_d and entropy error E_s are calculated and compared with the "exact" solution in order to evaluate the error of our numerical solution.

The lift coefficient on the bottom wall is defined as,

$$c_l = \frac{\int_{\text{bottom}} (p - p_\infty) n_y dl}{\frac{\gamma}{2} p_\infty M_\infty^2 h} \tag{30}$$

where $h = 0.0625$ is the height of bump. At the same time, the drag coefficient is given by,

$$c_d = \frac{\int_{\text{bottom}} (p - p_\infty) n_x dl}{\frac{\gamma}{2} p_\infty M_\infty^2 h} \tag{31}$$

The integral of the entropy error over the domain Ω can be expressed as,

$$E_s = \sqrt{\frac{\int_\Omega (s/s_t - 1)^2 d\Omega}{\int_\Omega d\Omega}} \tag{32}$$

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 $\rho_t = p_t/(RT_t)$. The reference values for these quantities 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 11, the red, green and blue lines correspond the error of c_l, c_d and E_s, respectively. The $p = 0$ solution is represented by those solid lines with triangle markers, while the $p = 1$ solutions are shown using the dash lines with cross markers. As for the $p = 2$ solution, they are plotted using dash-dotted lines with square markers.

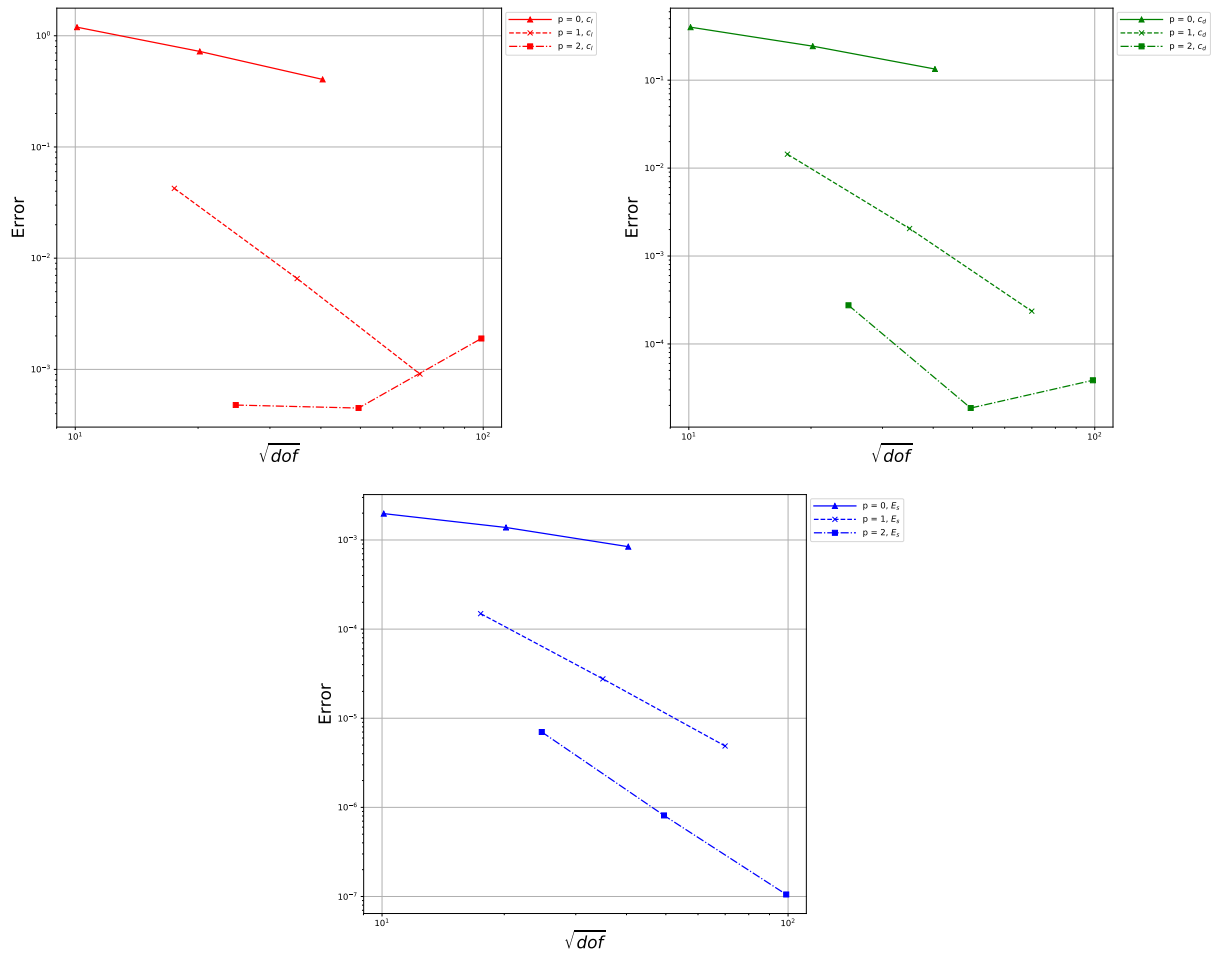

Figure 11. Convergence study for DG with $p = 0, 1, 2$.

As we can see in figure 11, the convergence rate for E_s (i.e. the slope of blue line) increases as the solution order p increases, which is in accordance with our anticipation. Meanwhile, the convergence for c_l and c_d also looks reasonable when $p = 0$ and 1.

Unfortunately, when considering $p = 2$ solutions, the convergence for c_l and c_d (represented by the red and green dash-dotted lines) are not quite satisfactory. One possible reason is that reference values for cl and c_d are not such accurate. Or, perhaps the geometry or boundary conditions are slightly different in our runs comparing with the reference runs.

The convergence rates can be evaluated quantitatively using,

$$\text{rate} = \log_2\left(\frac{e_2}{e_1}\right) \tag{33}$$

where e_2 and e_1 are the error of mesh **bump2** and **bump1** respectively. The geometric interpretation of equation (33) is that the convergence rate is identical to slope between the right-most two points.

† However, the convergence rate of c_d when $p = 2$ is calculated using e_1 and e_0, i.e. the error corresponding to **bump1** and **bump0**. Because in this case, if we use e_2 and e_1 to compute the slope, we will obtain an obvious wrong answer.

The results of convergence rates are summarized in table 1. From table 1, we can see that the convergence rate for $p = 0$ solution is slightly lower than 1.0. As for $p = 1$ solutions, the convergence rate we obtained is always larger than 2.0. When $p = 2$, the convergence rate is around 3.0. Roughly speaking, the numerical convergence rates are in accordance with the anticipated rates.

There are several possible reasons explaining why the convergence rates are not exactly what we anticipated. One potential reason is that other than the leading term, there are also higher order terms within the differential approximation, which may have a impact on our convergence study. At the same time, the implementation of curved elements may also introduce some errors, which will eventually lead to inaccuracy in the convergence rate.

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

Quantity	$p = 0$	$p = 1$	$p = 2$
c_l	0.84	2.85	X
c_d	0.86	3.12	$3.88^{\dagger}$
E_s	0.71	2.50	2.95
Anticipation	1.0	2.0	3.0

† calculated using **bump1** and **bump0**

5. Pressure Coefficient

The pressure coefficient is defined as,

$$c_p = \frac{p - p_\infty}{\frac{\gamma}{2} p_\infty M_\infty^2} \tag{34}$$

The pressure coefficient distribution for **bump2** with $p = 0, 1, 2$ are depicted in figure 12 figure 13 and figure 14, respectively. Note that by aerodynamic convention, what we are actually plotting is $-c_p$ versus x over the bottom wall.

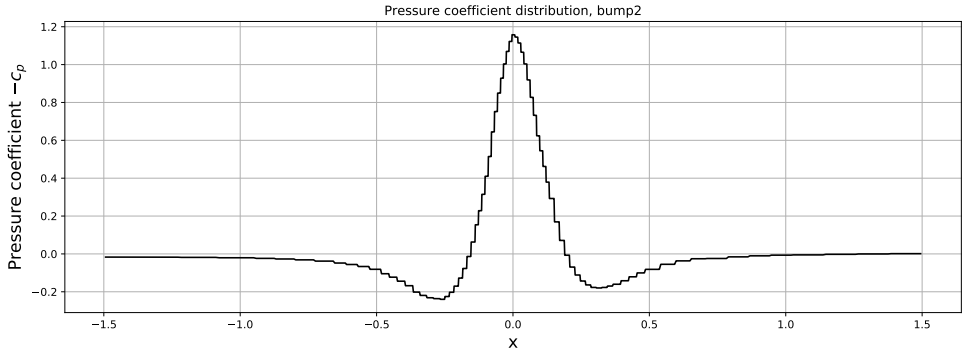

Figure 12. Pressure coefficient distribution on the bottom wall, bump2, $p = 0$.

As we can see in figure 12, the aliasing is quite obvious for the $p = 0$ (i.e. FVM) solution. Since for $p = 0$ solution, we only store the cell-averaged state within each element, there will be sharp discontinuities when crossing the interior edges. Consequently, when plotting the pressure coefficient distribution, we will get a step function jumping from one cell-average to another.

As shown in figure 13 and figure 14, if we increase the order of solution p to $p = 1$ and $p = 2$, the pressure coefficient distribution looks much smoother. These two figures are visually the same, which indicates that the $p = 1$ solution is already quite accurate if we only focus on the pressure coefficient c_p.

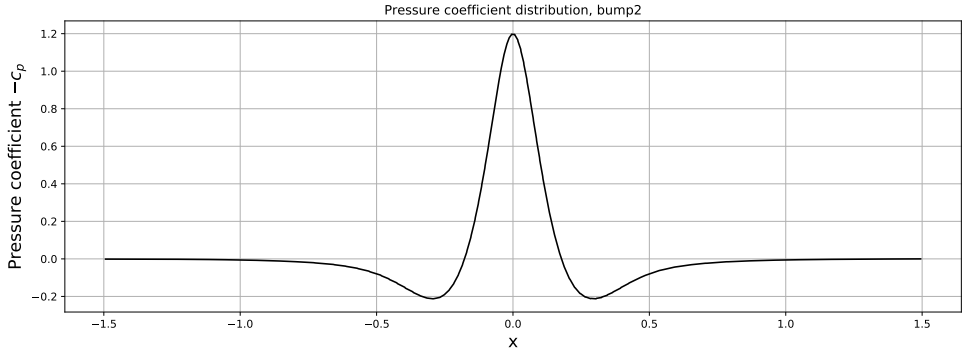

Figure 13. Pressure coefficient distribution on the bottom wall, bump2, $p = 1$.

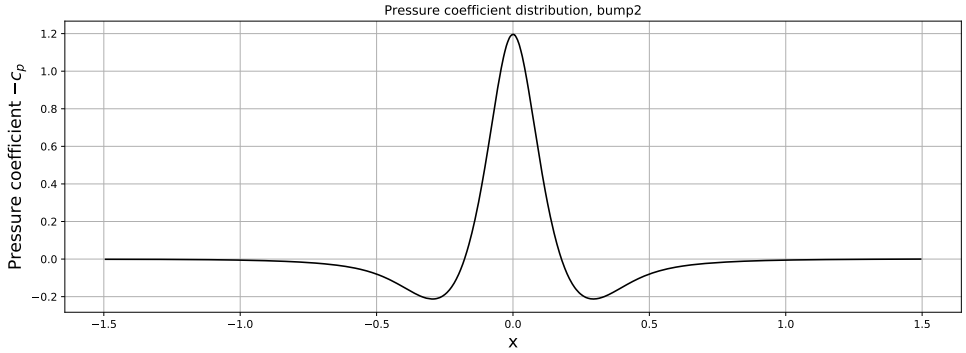

Figure 14. Pressure coefficient distribution on the bottom wall, bump2, $p = 2$.

6. Mach Number Fields

In order to accurately plot the Mach number field, we first need to generate a set of sampling points within the reference element. Then, we need to evaluate the state at those sampling points and calculate the Mach number. Finally, by mapping the result from reference space to global space, we can obtain the Mach number distribution. The sampling points within the reference triangle are shown as follows,

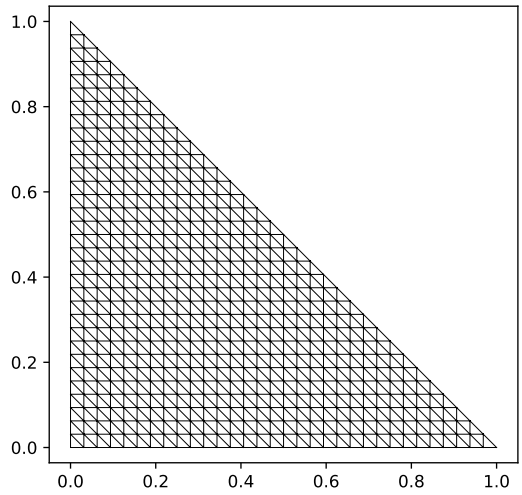

Figure 15. Sampling points within a reference triangle.

The Mach number fields from solution with order $p = 0, 1, 2$ are depicted in figure 16, figure 17 and figure 18, respectively.

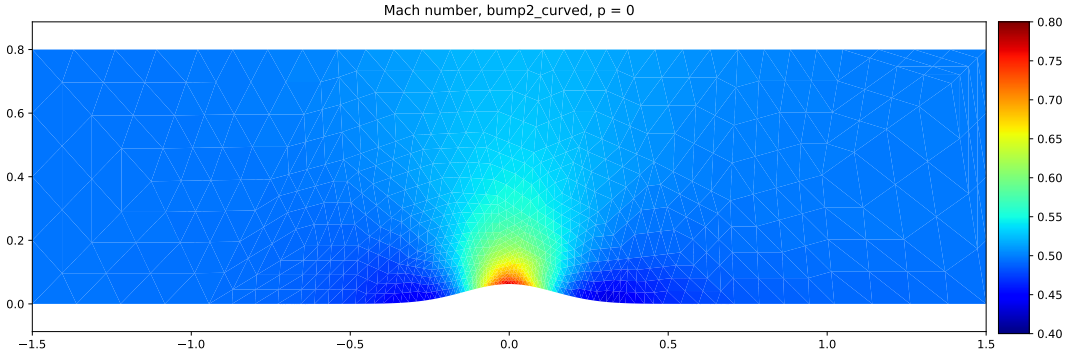

Figure 16. Mach number fields, bump2, $p = 0$.

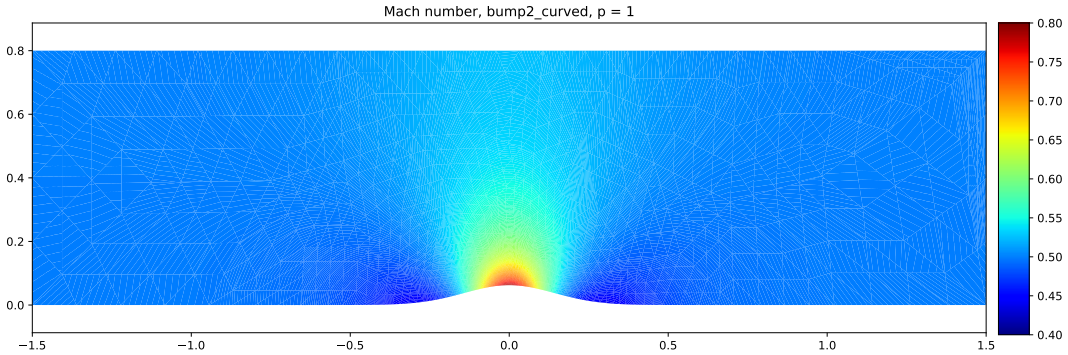

Figure 17. Mach number fields, bump2, $p = 1$.

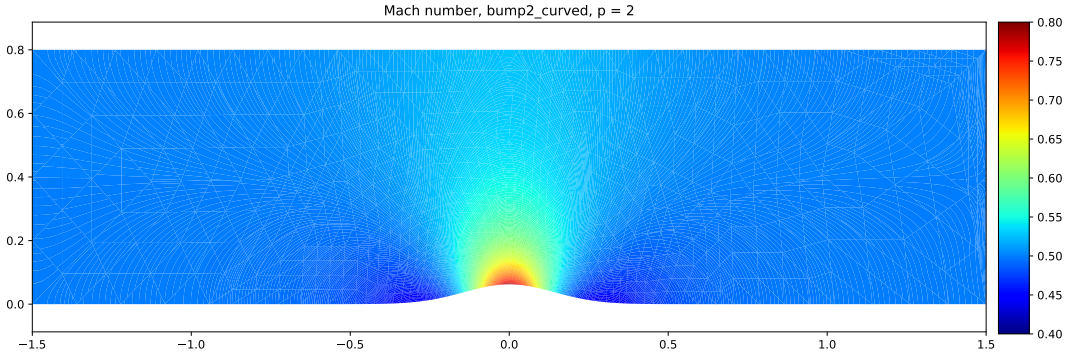

Figure 18. Mach number fields, bump2, $p = 2$.

As for the FVM solution, i.e. $p = 0$, we only store the cell-averaged state for each element. Hence, the Mach number within each cell is a constant as shown in figure 16. As we increase the solution order to $p = 1$, the state inside each triangle is no longer constant. The state varies linearly inside each element, and defined by the three states at the triangle vertices. If we scrutinize figure 17, we can find straight lines inside each element. The Mach number distribution inside each element can be considered as a tilted plane, hence the contour for this tilted plane has to be a set of straight lines. If we further increase the solution order to $p = 2$, we can see that those contour lines inside each element are no longer straight lines, but curves. Since the Mach number is now represented by a 2^{nd} order curved surface inside each element, the contour of this curved surface is no longer a set of straight line, but a set of curves.