48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition
4 - 7 January 2010, Orlando, Florida

AIAA 2010-1449

Computation Of Flows with Shocks Using Spectral Difference Scheme with Artificial Viscosity

Sachin Premasuthan, [*] Chunlei Liang [†] and Antony Jameson [‡]

Stanford University, Stanford, CA 94305, USA

The current work focuses on applying an artificial viscosity approach to the Spectral Difference (SD) method to enable high-order computation of compressible fluid flows with discontinuities. The study modifies the artificial viscosity approach proposed in the earlier work.[24] Studies show that a dilatation sensor for artificial viscosity, combined with a dilatation-based switch and filter for smoothing, works well for curvilinear and unstructured grids. The artificial viscosity model is found to stabilize numerical calculations and reduce oscillations near discontinuities. Promising results are demonstrated for 2-D test problems. Adaptive mesh refinement is used in conjunction with artificial viscosity to obtain improved shock profiles. A mortar element method is used to handle the non-conforming interfaces generated from mesh-refinement of quadrilateral elements.

I. Introduction

Until recently, compressible flow computations on unstructured meshes have generally been dominated by schemes restricted to second order accuracy. However, the need for highly accurate methods in applications such as large eddy simulation, direct numerical simulation, computational aeroacoustics etc., has seen the development of higher order schemes for unstructured meshes such as the Discontinuous Galerkin (DG) Method,[2,20] Spectral Volume (SV) method[19,33] and Spectral Difference (SD) Method.[18,32] The SD method is a newly developed efficient high-order approach based on differential form of the governing equation. The SD method can be viewed as an extension of the staggered-grid multi-domain method introduced by Kopriva and Kolias.[15] It was originally proposed by Liu et al.[18] and developed for wave equations in their paper on triangular grids. Wang et al.[32](2007) extended it to 2D Euler equations on triangular grids and Sun et al.[31](2007) further developed it for three-dimensional Navier-Stokes equations on hexahedral unstructured meshes. The SD method combines elements from finite-volume and finite-difference techniques, and is particularly attractive because it is conservative, has a simple formulation and straightforward implementation.

One of the greatest challenges with using high-order unstructured solvers is their inability to handle flow discontinuities. When flows involve steep gradients such as shock waves or contact surfaces, non-physical spurious oscillations arise that cause the simulations to go unstable. For higher order approximations, it is typically necessary to add explicit dissipation in order to obtain a stable solution. But this has a negative effect on accuracy, and the resolution of turbulent scales. The development of numerical algorithms that capture discontinuities and also resolve the scales of turbulence in compressible turbulent flows remains a significant challenge.

A classical approach to shock capturing is the addition of artificial viscosity, pioneered by von Neumann and Richtmeyer.[21] The addition of artificial viscosity or dissipation facilitates the capturing of discontinuities by smearing the discontinuity over a numerically resolvable scale. The concept of flexible addition of artificial viscosity/dissipation has been used notably by Jameson et al.,[10–13] thus producing non-oscillatory and sharp resolution of shocks for structured and unstructured finite volume calculations.

Cook and Cabot proposed such a method for high-order centered differencing schemes, wherein a spectral-like high-wavenumber biased artificial viscosity and diffusivity were dynamically added.[6,7] This was followed up with work by Fiorina and Lele,[8] on high-order compact difference schemes, wherein artificial diffusivity

[*] Doctoral Candidate, Department of Aeronautics and Astronautics, Stanford University, AIAA Member.

[†] Post-Doc, Department of Aeronautics and Astronautics, Stanford University, AIAA Member.

[‡] Professor, Department of Aeronautics and Astronautics, Stanford University, AIAA Fellow.

was added in addition to artificial viscosity. Kawai and Lele[14] extended the method to non-uniform and curvilinear meshes. This method involves the dynamic addition of grid-dependent localized transport coefficients such as artificial bulk viscosity, shear viscosity, artificial conductivity where needed. The application of this form of artificial viscosity (hyperviscosity) has been limited to structured grid computations.

Other forms of artificial viscosity have been applied to high-order unstructured grid calculations. Persson and Peraire[22] introduced a p-dependent artificial viscosity and demonstrated that higher-order representations and a piecewise-constant artificial viscosity can be combined to produce sub-cell shock resolution. Barter and Darmofal[3] proposed shock-capturing using a combination of higher-order PDE-based artificial viscosity and enthalpy-preserving dissipation operator. Both the above methods were proposed for high-order Discontinuous Galerkin (DG) discretizations.

Our earlier paper[24] presented the extension the artificial viscosity approach proposed by Cook et al,[6,7] and modified by Kawai et al[14] to computations on unstructured quadrilateral grids using the Spectral Difference scheme. The current work modifies the artificial viscosity formulation to add artificial bulk coefficient that is scaled as the dilatation, and is 2nd order in smooth regions of flow. It is shown that this modification is necessary when the grids are non-cartesian. A number of test-cases in 2D are included to demonstrate its applicability as well as limitations. The current implementation of artificial viscosity can also be easily extended to 3D spectral difference scheme.

It is known that the addition of artificial viscosity causes the accuracy to drop to first order in the vicinity of the shock. Thus the accuracy of shock computations can be improved only by refining the mesh in the region of shocks. Adaptive mesh refinement is one of the most commonly used tools in shock-capturing. However, adaptive mesh refinement in quadrilaterals leads to hanging nodes on the interfaces. To deal with these non-conforming interfaces, we use a mortar-element method. The mortar-element method for a SD setup was introduced by Kopriva.[16]

In section II, we look at the formulation of the Spectral Difference (SD) method on unstructured quadrilateral meshes. Section III discusses the details of the artificial viscosity method used. In section IV, we take a look at the mortar-element method required to enable computations with local mesh-refinement. In section V, we look at the numerical results obtained from the application of the current artificial viscosity model to 2D test cases. A couple of cases where artificial viscosity has been combined with local mesh-refinement are also demonstrated. Section VI discusses the conclusions of our study and the direction of future efforts.

II. Formulation of 2D Spectral Difference Scheme on quadrilateral meshes

The formulation of the equations for the 2D spectral difference scheme on quadrilateral meshes is similiar to the formulation of Sun et al[31] for unstructured hexahedral grids

Consider the unsteady compressible 2D Navier Stokes equations in conservative form

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0 \tag{1}$$

where Q is the vector of conserved variables; F and G are the total fluxes including both inviscid and viscous flux vectors.

To achieve an efficient implementation, all elements in the physical domain (x, y) are transformed into a standard square element. $0 < \xi < 1$, $0 < \eta < 1$. The transformation can be written as:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \sum_{i=1}^{K} M_i\left(\xi, \eta\right) \begin{pmatrix} x_i \\ y_i \end{pmatrix} \tag{2}$$

where K is the number of points used to define the physical element, (x_i, y_i) are the cartesian coordinates at those points, and $M_i\left(\xi, \eta\right)$ are the shape functions. The metrics and the Jacobian of the transformation can be computed for the standard element. The governing equations in the physical domain are then transfered into the computational domain, and the transformed equations take the following form:

$$\frac{\partial \tilde{Q}}{\partial t} + \frac{\partial \tilde{F}}{\partial \xi} + \frac{\partial \tilde{G}}{\partial \eta} = 0 \tag{3}$$

where $\tilde{Q} = |J| \cdot Q$ and

$$\begin{pmatrix} \tilde{F} \\ \tilde{G} \end{pmatrix} = |J| \begin{pmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{pmatrix} \begin{pmatrix} F \\ G \end{pmatrix} \tag{4}$$

In the standard element, two sets of points are defined, namely the solution points and the flux points, illustrated in figure 1. In order to construct a degree $(N-1)$ polynomial in each coordinate direction, solution at N points are required. The solution points in 1D are chosen to be the Chebyshev-Gauss points defined by:

$$X_s = \frac{1}{2}\left[1 - cos\left(\frac{2s-1}{2N} \cdot \pi\right)\right], s = 1, 2, \cdots, N. \tag{5}$$

The flux points were selected to be Legendre-Gauss quadrature points plus the two end points 0 and 1, as suggested by Huynh.[9] Choosing $P_{-1}(\xi) = 0$ and $P_0(\xi) = 1$, we can determine the higher-degree Legendre polynomials as

$$P_n(\xi) = \frac{2n-1}{n}(2\xi - 1)P_{n-1}(\xi) - \frac{n-1}{n}P_{n-2}(\xi) \tag{6}$$

The locations of these Legendre-Gauss quadrature points are the roots of equation $P_n(\xi) = 0$. They are generally found to be more stable than the Gauss-Lobatto flux points and produce more accurate solutions for high-order spectral difference schemes.

Using the solutions at N solution points, a degree $(N-1)$ polynomial can be built using the following Lagrange basis defined as:

$$h_i(X) = \prod_{s=1, s\neq i}^{N} \left(\frac{X - X_s}{X_i - X_s}\right) \tag{7}$$

Similiarly, using the fluxes at $(N+1)$ flux points, a degree N polynomial can be built for the flux using a similar Lagrange basis defined as:

$$l_{i+1/2}(X) = \prod_{s=0, s\neq i}^{N} \left(\frac{X - X_{s+1/2}}{X_{i+1/2} - X_{s+1/2}}\right) \tag{8}$$

The reconstructed solution for the conserved variables in the standard element is just the tensor products of the two one-dimensional polynomials,

$$Q(\xi, \eta) = \sum_{j=1}^{N}\sum_{i=1}^{N} \frac{\tilde{Q}_{i,j}}{|J_{i,j}|} h_i(\xi) \cdot h_j(\eta) \tag{9}$$

Similiarly, the reconstructed flux polynomials take the following form:

$$\begin{aligned} \tilde{F}(\xi, \eta) &= \sum_{j=1}^{N}\sum_{i=0}^{N} \tilde{F}_{i+1/2,j} \cdot l_{i+1/2}(\xi) \cdot h_j(\eta), \\ \tilde{G}(\xi, \eta) &= \sum_{j=0}^{N}\sum_{i=1}^{N} \tilde{G}_{i,j+1/2} \cdot h_i(\xi) \cdot l_{j+1/2}(\eta) \end{aligned} \tag{10}$$

The reconstructed fluxes are only element-wise continuous, but discontinuous across cell interfaces. For the inviscid flux, an approximate Riemann solver is employed to compute a common flux at interfaces to ensure conservation amd stability. In our case, we have used the Rusanov solver[26] or Scalar Diffusion[12] as the approximate Reimann solver to compute the interface fluxes.

In summary, the algorithm to compute the inviscid flux derivatives consists of the following steps:

1. Given the conservative variables at the solution points, the conservative variables are computed at the flux points

2. The inviscid fluxes at the interior flux points are computed using the solutions computed at Step 1

3. The inviscid fluxes at the element interfaces are computed using the Rusanov solver.

4. The derivative of the fluxes are computed at the solution points according to (equation)

$$\left(\frac{\partial \tilde{F}}{\partial \xi}\right)_{i,j} = \sum_{r=0}^{N} \tilde{F}_{r+1/2,j} \cdot l'_{r+1/2}(\xi_i), \tag{11}$$

$$\left(\frac{\partial \tilde{G}}{\partial \eta}\right)_{i,j} = \sum_{r=0}^{N} \tilde{G}_{i,r+1/2} \cdot l'_{r+1/2}(\eta_j) \tag{12}$$

The viscous flux is a function of both the conserved variables and their gradients. Therefore, the solution gradients have to be calculated at the flux points. In our solver, the average approach described in reference[31] is used to compute the viscous fluxes. The procedure to compute the viscous fluxes can be described as follows.

1. Reconstruct Q_f at the flux points from the Q at the solution points using equation 9.
2. At the element interfaces, find the average of left and right values of Q_f; $\overline{Q_f} = \frac{1}{2}(Q_f^L + Q_f^R)$. For interior flux points, $\overline{Q_f} = Q_f$. Appropriate boundary conditions are applied at flux points on boundary edges.
3. Evaluate ∇Q at the solution points from $\overline{Q_f}$ using equation 11, where $\nabla Q = \left\{ \begin{array}{c} Q_x \\ Q_y \end{array} \right\}$ and $Q_x = \frac{\partial Q}{\partial \xi}\xi_x + \frac{\partial Q}{\partial \eta}\eta_x$, etc.
4. Reconstruct ∇Q to the flux points, apply appropriate boundary conditions for boundary flux points, and average them on the element interfaces as $\overline{\nabla Q_f} = \frac{1}{2}(\nabla Q_f^L + \nabla Q_f^R)$
5. Use $\overline{Q_f}$ and $\overline{\nabla Q_f}$ in order to compute viscous flux vectors at the flux points.

It should be mentioned that all explicit time-marching calculations for steady flows have been done using a Jameson type four-stage Runge Kutta scheme (RK4), which is 2nd order accurate in time. For the unsteady problems, we have used a 4th order accurate, strong-stability-preserving five-stage Runge-Kutta scheme[29] to advance in time.

III. Artificial Viscosity

The application of spatially high-order schemes to compute flows with discontinuities such as shock or contact-discontinuities, results in non-physical spurious oscillations that make the computation unstable. One of the major concerns in simulating such flows is to ensure the removal of these non-physical oscillations while avoiding damping of the resolved scales of turbulence.

The general formulation for artificial viscosity approach used here is similar to the 'Local artificial viscosity and diffusivity' approach of Kawai and Lele.[14] This method is a modification of the original high-wavenumber biased artificial viscosity approach introduced by Cook and Cabot,[7] extended to anisotropic and curvilinear structured grids. The present artificial viscosity approach adds grid-dependent components to the viscosity coefficients, as proposed by Kawai and Lele,

$$\begin{aligned} \mu &= \mu_f + \mu_\Delta, \\ \beta &= \beta_f + \beta_\Delta, \\ \kappa &= \kappa_f + \kappa_\Delta \end{aligned} \tag{13}$$

where μ is the dynamic (shear) viscosity, β is the bulk viscosity, and κ is the thermal conductivity. The f and Δ subscripts denote the fluid and artificial transport coefficients respectively.

These artificial transport coefficients are defined by:

$$\mu_\Delta = C_\mu \rho \, \overline{| \sum_{l=1}^{3} \sum_{m=1}^{3} \Delta_l^{r+2} \left(\frac{\partial \xi_l}{\partial x_m}\right)^r \frac{\partial^r S}{\partial \xi_l^r} |}$$

$$\beta_\Delta = C_\beta \rho \overline{\mid \sum_{l=1}^{3} \sum_{m=1}^{3} \Delta_l^{r+2} \left(\frac{\partial \xi_l}{\partial x_m} \right)^r \frac{\partial^r \left(\nabla . \mathbf{u} \right)}{\partial \xi_l^r} \mid}$$

$$\kappa_\Delta = C_\kappa \frac{\rho c_s}{T} \overline{\mid \sum_{l=1}^{3} \sum_{m=1}^{3} \Delta_l^{r+2} \left(\frac{\partial \xi_l}{\partial x_m} \right)^r \frac{\partial^r e}{\partial \xi_l^r} \mid} \tag{14}$$

where C_μ, C_β and C_κ are user-specified constants. ξ_l refers to the computational coordinates and x_m refer to the physical coordinates. Δ_l is the physical grid spacing along a grid line in the ξ_l direction. The magnitude of the strain rate tensor (S), the dilatation $(\nabla.\mathbf{u})$, and the internal energy (e) are the sensors corresponding to artificial shear viscosity, bulk viscosity and conductivity respectively. Experiments using high-order structured grid calculations suggest that r equals 4 or higher. For sufficiently high r, the high-wavenumber bias (k^r) results in damping of wavenumbers close to the unresolved wavenumbers.

It must be noted that such a formulation results in the addition of artificial viscosity terms that are $O(\Delta^{r+2})$ in smooth regions of the flow and $O(\Delta)$ in the vicinity of the shock. This can in fact be compared to the blended diffusion used by Jameson et al,[12] in schemes like JST (Jameson-Schmidt-Turkel) and SLIP (Symmetric Limited Positive), where the artificial dissipation is third order in smooth regions of the flow and first order when there is a discontinuity.

Our earlier experiments with this form of viscosity suggested using $r = 2$, which essentially meant adding a $O(\Delta^4)$ term to the viscosity coefficients in smooth of the flows. It also involved computing the Laplacian of the sensor quantities. An advantage of such an artificial viscosity scheme is that it eliminates the need for limiters/switches to turn off the artificial bulk viscosity in regions of expansion and isentropic compression. However, the use of $r = 2$ was found to have a detrimental effect when using curvilinear meshes, or fully unstructured meshes. This could be attributed to the inaccurate calculation of the laplacian of sensor quantities using the SD setup, when we have non-cartesian grids.

The current study proposes the use of $r = 0$ for artificial viscosity (AV) computations. With r=0 the formulation reduces to

$$\mu_\Delta = C_\mu \overline{\rho \mid \Delta^2 S \mid}$$

$$\beta_\Delta = C_\beta \overline{\rho S_\beta \mid \Delta^2 \left(\nabla . \mathbf{u} \right) \mid} \tag{15}$$

where Δ is the grid spacing, and S_β is a dilation-based switch. This form of artificial viscosity was found to give much better results for flows with shocks on curvilinear and fully unstructured meshes. This is clearly demonstrated in figure 5. Figure 5(a) shows the computed artificial viscosity (with r=2) when a shock is located at x=0.0, and the discontinuity is aligned with the grid lines. However in figure 5(b), the discontinuity is at x=0.5 and not aligned with the mesh lines. Here we see non-smooth AV contours and non-physical transverse variations in AV. However, while using r=0 for the AV computations, we see that the AV profile is much smoother, and there are no significant transverse variations, as shown in figure 5(c). The superior performance of using r=0 is also seen when using a fully unstructured grid as shown in figure 6. The AV contours obtained with r=0 computation is found to be much smoother and gave much more accurate results, as compared to the r=2 case (see figures 7 and 8).

However reducing r to 0 comes with certain disadvantages. Firstly the artificial viscosity term becomes 2nd order $O(\Delta^2)$ in smooth region, which is obviously less accurate than the 4th order term obtained with r=2. Secondly, the formula for the artificial conductivity becomes invalid, and hence we have to find an alternative for problems with contact discontinuities. Thirdly, there is now a need for a switch to turn off artificial viscosity in regions of smooth flow (which is explained in section III.B). The one advantage though is that the computational cost for calculating the artificial viscosity becomes much lesser.

The overbar in equation 15 denotes a filter to smooth the artificial transport coefficients. In structured grid calculations, a truncated Gaussian filter is used.[7] The filter is also meant to eliminate cusps introduced by the absolute value operator, which in turn ensures that artificial viscosities are positive. Here we have designed a restriction-prolongation filter suitable to the SD setup. There is also a need for a switch (S_β in the formulation), to ensure that AV is added only in the region of shocks, and is zero in smooth regions of the flow.

The artificial viscosity/conductivity is calculated at each flux point. This is computationally extensive, but it ensures smooth variation of artificial transport coefficients. A smooth representation of artificial viscosity within mesh elements is considered beneficial as compared to the piecewise-constant artificial viscosity formulations, as element-to-element variations can lead to oscillations in state gradients and disparate equilibrium shock-jump conditions in neighboring elements.

III.A. Filter for unstructured SD setup

The filter plays an important role in artificial viscosity computations as it ensures smooth variation of artificial transport coefficients within the domain. For calculations using artificial viscosity on structured grids, a truncated Gaussian filter is used. A 7-point or 9-point stencil is generally used for this purpose.[7] For calculations in 2D, the Gaussian filter is applied along each grid-line separately. However, for unstructured grids it is not reasonable to implement the Gaussian filter in its existing form, as obtaining a stencil for each solution/flux point can be tedious. The stencil would lie across cells, and the non-uniform spacing would have to be taken into account, thus making it cumbersome to implement. This motivated the development of a filter that would be suited to the current SD setup.

In the current study we use an element-wise restriction-prolongation filter (we will refer to it as the R-P filter). The concept is similar to the one used by Blackburn et al[5] for spectral element filtering. It involves the projection of the quantity in concern to a lower-order basis (restriction), smoothing at this level, and then extrapolation back to original basis (prolongation). The basic steps in implementation of the R-P filter can be described in 1-D as follows,

1. Consider a 4th order SD element. The artificial viscosity terms have been computed at the 4 solution points (Figure 3(a)).

2. The function (represented by a cubic polynomial through the 4 solution points) is restricted to 2 solution points (corresponding to 2nd order SD) (Figure 3(a)). The polynomial fit through the interpolated function is reduced to linear. The function is now extrapolated to the 3 flux points corresponding to second order solution.

3. The function values are averaged at the interface for all element interfaces (Figure 3(c)). This is equivalent to smoothing of the function at the lowest level. A quadratic polynomial is fitted through this smoothed function through the 3 flux points.

4. It is then extrapolated to the flux points at the highest level (4th order)(Figure 3(d)).

It was found that in 1-D, this filter performs comparably to the Gaussian filter applied on flux points. Since the solution representation in multiple dimensions is just a tensor product of 1-D polynomials, the extension of this filter to 2-D and 3-D is straight-forward.

Figure 4 corresponds to the initial condition of the SOD shock tube case with a density discontinuity at x=0.5. The artificial conductivity is non-zero in the vicinity of x=0.5. The figure shows that prior to filtering the artificial conductivity field is noisy and may have oscillatory behavior. The filtered coefficient is smoother and results in a better solution. Also, smoothing effect of the present R-P filter compares quite well with the Gaussian filter, even though the peaks do not match.

III.B. Switch in artificial viscosity formulation

The switch used here is one proposed by Bhagatwala and Lele[4] to restrict the addition of AV only to regions of strong shocks. The formulation of the switch is as shown

$$S_{beta} = 0.5 * (1 - tanh(C1 + C2 * \frac{\Delta}{c} \nabla . \mathbf{u})) \tag{16}$$

The constants C1 and C2 were chosen to be 2 and 20 respectively for the present cases. This switch is designed so as to add AV only in regions of strong negative dilatation, i.e, corresponding to shocks.

IV. Mortar element method for mesh refinement

Local mesh refinement for quadrilateral meshes results in hanging nodes on cell faces (figure 2(a)). As a result, the flux points on the interface do not coincide with each other and the approximate Reimann solver cannot be applied directly. The treatment of such sub-domain refinement using mortar elements had been illustrated by Kopriva.[16] The current work uses a similar mortar method to deal with the non-conforming meshes obtained from local mesh refinement.

When a sub-domain is refined as shown in figure 2(b), two mortars are introduced corresponding to each of the short faces. This ensures that the outflow condition is satisfied. The outflow condition requires that the projection of face values from sub-domains Ω^2 and Ω^3 onto a mortar, and the subsequent projection back onto the faces returns the original polynomial functions.

We can define the solution approximations along the faces as

$$\begin{aligned} U^1(\xi) &= \sum_{j=1}^{N^1} U_j^1 h_j^1(\xi) \in \mathbf{P}^{N^1-1} \\ U^2(\xi) &= \sum_{j=1}^{N^2} U_j^2 h_j^2(\xi) \in \mathbf{P}^{N^2-1} \\ U^3(\xi) &= \sum_{j=1}^{N^3} U_j^3 h_j^3(\xi) \in \mathbf{P}^{N^3-1} \end{aligned} \tag{17}$$

where $\xi \in [0,1]$ is the local sub-domain coordinate. We also define four mortar functions

$$\begin{aligned} \phi^{1,(L,R)}(z) &= \sum_{j=1}^{J^1} \phi_j^{1,(L,R)} h_j^{\Xi^1}(z) \in \mathbf{P}^{J^1-1} \\ \phi^{2,(L,R)}(z) &= \sum_{j=1}^{J^2} \phi_j^{2,(L,R)} h_j^{\Xi^2}(z) \in \mathbf{P}^{J^2-1} \end{aligned} \tag{18}$$

which are functions of the local mortar coordinate $z \in [0,1]$. The superscripts L and R correspond to values on left and right of the mortar. We also define variables o^k and s^k to be the offset and the scale of a mortar with respect to the sub-domain Ω^k that contributes to it. Thus for $z \in [0,1]$, $\xi^k = o^k + s^k z$.

The orders of the mortar polynomials must be chosen sufficiently high so that the outflow condition is satisfied. This means the mortar must be at least as large as the largest sub-domain order of all contributing sub-domains. Thus, J^1=max(N^1, N^2) and J^2=max(N^1, N^3). It must be mentioned that in the present study, the polynomial order in all the sub-domains are the same.

The computation of fluxes on the sub-domain faces using mortar elements involves three main steps -

1. Project the solution from sub-domain to mortar
2. Compute the fluxes on the mortar
3. Project the flux from mortar back to sub-domain

IV.A. Sub-domain $\longrightarrow$ Mortar Projections

To compute the sub-domain to mortar projection matrices, we use an un-weighted L^2 projection. Thus, we seek polynomials on the two mortars that best approximate the polynomial along the contributing face. For each mortar Ξ and each subdomain contributor Ω we require,

$$\int_0^1 (\phi(z) - U(\xi)) h_m^{\Xi}(z) dz = 0, \qquad m = 1, 2, \cdots, J \tag{19}$$

Then the vector of the solution values along the mortar can be computed by

$$\Phi = P^{\Omega \rightarrow \Xi} U = M^{-1} S U \tag{20}$$

where,

$$\begin{aligned} S_{mj} &= \int_0^1 h_m^{\Xi}(z) h_j^{\Omega}(o^k + s^k z) dz, & m = 1, \cdots, M, j = 1, \cdots, J \qquad (21) \\ M_{mj} &= \int_0^1 h_m^{\Xi}(z) h_j^{\Xi}(z) dz, & m, j = 1, \cdots, J \qquad (22) \end{aligned}$$

IV.B. Flux calculation on the mortar elements

The fluxes on the mortar are computed as earlier because the flux points on either side of the mortar coincide with each other. An approximate Reimann solver is used to compute the fluxes on the mortar. For viscous flow computations, the solution values are averaged on the mortar. We will denote the fluxes on the mortar by the variable Ψ.

IV.C. Mortar $\longrightarrow$ Sub-domain projection

The computed fluxes on the mortar elements have to be projected back to the sub-domain faces. The current study involves computations where the polynomial order is same in all the cells. This means that the polynomial order on the mortar matches those on the short faces. Thus it is clear that the projection matrix for projection from mortar to the short faces is just the identity matrix, i.e, the computed fluxes are just copied from the mortar to the short faces. However, the projection from the mortars to the long face is a little more complicated. The piecewise polynomial that represents the fluxes along the mortars, possibly discontinuous, must be used to compute the continuous polynomial along the face. As before, we seek the best polynomial on the face that approximates the mortar solutions in the least squares sense. To obtain this least squares projection, we seek flux that satisfies.

$$\sum_{k=1}^{N^{\Xi}} \int_{o^k}^{o^k+s^k} (F^1(\xi) - \Psi^{\Xi^k}(\xi)) h_m^1(\xi) d\xi = 0 \tag{23}$$

The vector of values on the long sub-domain face can be computed by the projection

$$F^1 = \sum_{k=1}^{N^{\Xi}} P^k \Psi^{\Xi^k}, P^k = M^{-1} S^k \tag{24}$$

where the matrices M and S^k are defined as

$$\begin{aligned} M_{mj} &= \int_0^1 h_m^1(\xi) h_j^1(\xi) d\xi, & m, j = 1, \cdots, N^1 \\ S_{mj}^k &= s^k \int_0^1 h_m^1(o^k + s^k z) h_j^{\Xi^k}(z) dz, & m = 1, \cdots, N^1, j = 1, \cdots, J^k \end{aligned} \tag{25}$$

V. Results

It must be mentioned that the Spectral Difference scheme has been implemented in 1D, 2D and 3D solvers which have been tested, validated and found to exhibit formal order accuracy.[17,23] In the first part of this section, the results obtained from the application of AV with r=2 to problems with shocks in 1D, are discussed. Secondly, 2-D cases using r=0 are demonstrated, namely the supersonic flow past bump, Mach 3 flow past cylinder, and transonic flow past an airfoil.

V.A. SOD Shock-tube problem

The first 1-D test case is the shock-tube problem introduced by SOD.[28] The initial left and right-side conditions are $\rho_l = 1.0$, $u_l = 0.0$ and $p_l = 1.0$ for $x \leq 0.5$, and $\rho_r = 0.125$, $u_r = 0.0$ and $p_r = 0.1$ for $x > 0.5$. Simulations are performed on a uniformly spaced grid in the region $0 \leq x \leq 1$. Artificial bulk viscosity and conductivity are used. The coefficients used were $C_\beta = 0.06$ and $C_\kappa = 0.01$, along with r=2 (in equation 14), corresponding to second order derivatives of the sensor quantities. C_μ was set to zero.

Figure 9 (a),(b) and 10 (a) shows the comparison between density, velocity and pressure for the exact solution and 4th order SD computation with 100 cells at time $\tau = 0.15$. The shock and the contact discontinuity are captured well without significant spurious oscillations, and show reasonable agreement with the exact solution. Figure 10 (b) shows the variation of density profile with grid-refinement. It is observed that as the grid is refined, the solution converges closer to the exact solution. It should also be noted that for all grid spacings, the shock is spread over two cells and the contact discontinuity is spread over 3 cells. Figure 11 (a) shows the artificial bulk viscosity coefficient. It is seen to be maximum in the vicinity of the shock. Figure 11 (b) shows the artificial conductivity. There are two peaks corresponding to the shock and the contact discontinuity. In the SOD problem, artificial conductivity plays an important role because the artificial viscosity sensor does not sense the contact discontinuity.

The results described above were obtained using $r = 2$ in the artificial transport coefficient calculations. Computations were also conducted using $r = 4$, but the results obtained were very similar to those obtained for $r = 2$. For $r = 4$, $C_\beta = 0.001$ and $C_\kappa = 0.0001$ were used. For both 1-D cases it was observed that $r = 2$ and $r = 4$ gave very similar results. For all cases considered in this study, $r = 2$ has been used.

V.B. Shu-Osher problem

The second 1-D test case is the shock-entropy wave interaction introduced by Shu and Osher.[27] Because the entropy waves are sensitive to the numerical dissipation, excessive numerical dissipation damps the entropy waves. Initial left and right side conditions are given by: $\rho_l = 3.857143$, $u_l = 2.629369$ and $p_l = 10.33333$ for $x < -4$, and $\rho_r = 1 + 0.2 * sin(5x)$, $u_r = 0.0$ and $p_r = 1.0$ for $x \geq -4$. Simulations are performed on a uniformly spaced grid in the region $-5 \leq x \leq 5$. The coefficients used are the same as for the SOD case.

Figure 12(a) shows the comparison between the reference solution and 4th order SD simulations with 100, 200 and 400 cells. The reference solution is obtained using 5th order WENO on 2000 grid points. The density profile using 400 cells shows excellent agreement with the reference solution. Figure 12(b) shows a close up of the density plot in the region of the entropy waves. It is found that solution with 200 cells also shows reasonable agreement with reference solution. Figures 13(a) and (b) show the velocity and pressure profiles behind the shock.

V.C. Inviscid Supersonic flow past bump

This test-case consists of inviscid supersonic flow in a channel with a 4% thick circular bump on the bottom. The length of the channel is 3 units and its height 1 unit. The inlet Mach number is 1.4. This test case has been used by Ripley et al.[25] in computations using adaptive unstructured mesh refinement. Third and fourth order SD computations were conducted on two meshes. The coarse computational mesh has 1200 elements, and 20 nodes to resolve the bump, as depicted in figure 14. The fine mesh has 4800 cells, and has twice the number of nodes in the x and y directions. The surface of the bump is represented as a quadratic and cubic boundary for third and fourth order calculations respectively.

The pressure contours obtained using the 3rd order SD scheme with artificial viscosity on the coarse mesh is shown in figure 15, and compares well with those obtained using adaptive unstructured mesh refinement.[25] The pressure contours obtained for the fine mesh are shown in figure 16. It is observed that on the finer mesh, the shock profiles are sharper, and smoother contours are obtained. Figure 17 shows the drop in global residual, indicating a stable, convergent solution for both 3rd and 4th order cases. It must be mentioned that in the absence of artificial viscosity, the solution develops spurious oscillations and the simulation becomes unstable.

Figure 18 gives a plot of the artificial viscosity for third order computation on the coarse mesh. Figure 19 shows the variation of artificial bulk viscosity on the fine mesh. We see that artificial viscosity is added only in regions with sharp gradients of dilatation, corresponding to shocks. Figure 20 shows the pressure contours obtained using 4th order SD. It is observed that the shock profiles are slightly sharper than in the case of the third order computation. Also, 4th order on finer mesh gives sharper shock resolution and more accurate contours in comparison to the coarse mesh (see Figure 21).

The above test-cases were computed using r=0 and $C_\beta = 0.3$. Also, the artificial shear viscosity and artificial conductivity were set to zero. This is because there are no large shear gradients, and no contact discontinuities, and hence artificial bulk viscosity is sufficient to stabilize the calculations.

V.D. Inviscid Supersonic flow past circular cylinder

The next test-case is the $M_\infty = 3.0$ flow past a circular cylinder. The computational grid with 1200 cells is shown in figure 24(a). Supersonic inflow and outflow BC's are used. The cylinder wall is treated as inviscid curved wall. In the artificial viscosity model, r=0 was used with $C_\beta = 1.0$ for 3rd order computation and $C_\beta = 0.5$ for 4th order computation. Pressure and artificial bulk viscosity for the 3rd order simulation and 4th order simulation are shown in figure 24 and figure 25 respectively. It is observed that the shock profile is thinner for the 4th order case as compared to the 3rd order case.

V.E. Inviscid Transonic flow past airfoil

This test-case involves the transonic flow past a NACA0012 airfoil. The freestream mach number is 0.8 and the airfoil is at an angle of attack of 1.25 degrees. Freestream conditions are prescribed at the outer boundaries. On the airfoil surface, inviscid wall boundary condition is used. The curved airfoil surface is represented by cubic splines. A weak shock is formed on the upper surface, and an even weaker shock is formed on the lower surface. The current method is able to capture the shocks accurately for both 3rd and 4th order computations. Computations are done on a 80×16 mesh and on a 160×32 mesh.

Figures 26 and 27 show the pressure and AV contours for the 3rd order computations on the coarse and fine mesh. Figures 28 and 29 show the pressure and AV contours for the 4th order computations on the coarse and fine mesh. A look at the AV plot clearly indicates that AV is added only in the region of the shocks. The switch ensures that AV is zero even when dilatation is non-zero, such as in the region near the leading edge. The C_p plots obtained are compared with the results from FLO-82 on a 320x64 grid, and are shown in figure 30. The C_p plots obtained shows good comparison with those obtained using FLO-82 with a finer (320x64) grid. It is observed that the shock profile obtained with FLO-82 is much sharper. This is expected as the SD computations were done on coarser grids (even though the number of DOF's are comparable), and the current AV model tends to smear the shock over one to two cells.

V.F. Adaptive mesh-refinement for shock capturing

The mortar element method described above has been used to enable adaptive mesh-refinement in computations with shocks. The criterion for mesh refinement used here is non-zero artificial viscosity. This is a sufficient criterion since we are looking to refine the mesh only in the region of shocks. Firstly, adaptive mesh refinement is applied to the $M_\infty = 3.0$ flow past cylinder case. Two levels of mesh-refinement are used. Figure 31 shows the mesh after successive levels of mesh refinement. The pressure contours and AV contours obtained after local mesh-refinement are shown in figure 32 and figure 33 respectively. It is observed that as the mesh is refined, the shock structures become thinner, the amount of artificial viscosity added is lesser, and the pressure contours become more accurate.

Adaptive mesh refinement is also applied to the supersonic bump case described earlier. Once again the criterion for mesh refinement is non-zero AV. Figures 34 and 35 show the computational mesh after one and two levels of refinement respectively. Figures 36- 38 show the pressure contours obtained for successive levels of mesh refinement, for a 4th order computation. Figures 39- 41 show the artificial viscosity contours obtained for the corresponding cases.

VI. Conclusions and Future work

A dilatation-sensor based artificial viscosity scheme has been implemented in order to enable high-order computation of flows with discontinuities. The application of artificial viscosity with the Spectral Difference method for the computation of flows with shocks is demonstrated with test cases in 1D and 2D. Promising results have been obtained for these cases, with the method being able to produce a stable solution with sharp resolution of shocks, and no significant spurious oscillations. An element-based restriction-prolongation filter has been developed. The artificial viscosity has been combined with adaptive mesh refinement to get improved results. The mortar element method is used to enable adaptive refinement of the quad meshes. Further efforts will be directed towards the testing and validation of the proposed method, using a variety of shock-related problems in 2D and 3D. Efforts will also be directed towards the study of the effect of using irregular meshes, and the effect of adding artificial viscosity on the global accuracy of the Spectral Difference scheme. The scheme will then be used to tackle problems with shocks and shock-turbulence interaction in three dimensions.

VII. Acknowledgements

The authors would like to thank the grant support from NSF monitored by Dr Leland Jameson (with award number 0708071), and AFOSR monitored by Dr Fariba Fahroo (with award number FA9550-07-1-0195).

References

[1] F. Bassi & S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations, *Journal of Computational Physics*, Vol. 131, pp.267-279, 1997

[2] F.Bassi and S. Rebay, High-order accurate discontinuous finite element solution of the 2D euler equations, *Journal of Computational Physics*, Vol. 138, pp. 251-285, 1997

[3] G.E. Barter & D.L. Darmofal, Shock Capturing with Higher-Order, PDE-Based Artificial Viscosity, *AIAA paper*, AIAA-2007-3823, 2007.

[4] A. Bhagatwala & S.K. Lele, A modified artificial viscosity approach for compressible turbulence simulations, *Journal of Computational Physics*, Vol. 228, pp. 4965-4969, 2009

[5] H.M.Blackburn & S. Schmidt, Spectral Element filtering tecniques for large eddy simulation with dynamic estimation, *Journal of Computational Physics*, Vol. 186, pp.610-629, 2003

[6] A. W. Cook & W. H. Cabot, A high-wavenumber viscosity for high-resolution numerical methods *Journal of Computational Physics*, Vol. 195, pp.594-601, 2004.

[7] A. W. Cook & W. H. Cabot, Hyperviscosity for shock-turbulence interactions *Journal of Computational Physics*, Vol. 203, pp.379-385, 2005.

[8] B. Fiorina & S.K. Lele, An artificial nonlinear diffusivity method for supersonic reacting flows with shocks, *Journal of Computational Physics*, Vol. 222, pp. 246-264, 2007.

[9] H.T. Huynh, A Flux Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin Methods, *AIAA paper*, AIAA-2007-4079, 2007.

[10] A. Jameson, Transonic Potential Flow Calculations Using Conservation Form, *Proceedings of Second AIAA Computational Fluid Dynamics Conference, Hartford*, pp. 148-161, June 1975.

[11] A. Jameson, W. Schmidt, and E. Turkel, Numerical simulation of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes, *AIAA paper*, AIAA-81-1259, 1981.

[12] A. Jameson, Analysis and Design of Numerical Schemes for Gas Dynamics, 1: Artificial Diffusion, Upwind Biasing, Limiters and Their Effect on Accuracy and Multigrid Convergence, *International Journal of Computational Fluid Dynamics*, Vol. 4, pp.171-218, 1994.

[13] A. Jameson, Analysis and Design of Numerical Schemes for Gas Dynamics, 2: Artificial Diffusion and Discrete Shock Structure, *International Journal of Computational Fluid Dynamics*,Vol. 5, pp.138, 1995.

[14] S. Kawai & S.K. Lele, Localized Artificial Diffusivity Scheme for Discontinuity Capturing on Curvilinear Meshes,*Journal of Computational Physics*, Vol. 227, pp.9498-9526, 2008.

[15] D.A. Kopriva and J.H. Kolias, A Conservative Staggered-Grid Chebyshev Multidomain Method for Compressible Flows, *Journal of Computational Physics*, Vol. 125, pp.244261, 1996.

[16] D.A. Kopriva, A Conservative Staggered-Grid Chebyshev Multidomain Method for Compressible Flows. II. A Semi-Structured Method, *Journal of Computational Physics*, Vol. 128, pp.475488, 1996.

[17] C. Liang, S. Premasuthan, A. Jameson & Z. Wang, Large Eddy Simulation of Compressible Turbulent Channel Flow with Spectral Difference method, *AIAA paper*, 2009-402, 2009

[18] Y. Liu, M. Vinokur, & Z. J. Wang, Spectral difference method for unstructured grids I: Basic formulation, *J. of Comput. Phys.*, Vol. 216, pp. 780-801, 2006.

[19] Y. Liu, M. Vinokur, & Z. J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids V: Extension to three-dimensional systems, *Journal of Computational Physics*, Vol. 212, pp 454-472, 2006.

[20] H. Luo, J. D. Baum and R. Löhner, A p-multigrid discontinuous Galerkin method for the Euler equations on unstructured grids, *Journal of Computational Physics*, Vol. 211, pp 767-783, 2006.

[21] J. von Neumann, & R. Richtmyer, A method for the numerical calculation of hydrodynamic shocks, *Journal of Applied Physics*, Vol. 21, pp.232237, 1950.

[22] P.-O. Persson & J. Peraire, Sub-Cell Shock Capturing for Discontinuous Galerkin Methods, *AIAA paper*, AIAA-2006-112, 2006.

[23] S. Premasuthan, C. Liang, A. Jameson & Z. Wang, p-Multigrid Spectral Difference Method For Viscous Compressible Flow Using 2D Quadrilateral Meshes, *AIAA paper*, 2009-950, 2009

[24] S. Premasuthan, C. Liang & A. Jameson, A Spectral Difference Method for Viscous Compressible Flows With Shocks, *AIAA paper*, 2009-3785, 2009

[25] R.C. Ripley, F.S. Lien & M.M. Yoyanivich, Adaptive Unstructured Mesh Refinement of Supersonic Channel Flows, *International Journal of Computational Fluid Dynamics*, Vol. 18 (2), pp.189198, 2004

[26] V.V. Rusanov, Calculation of interaction of non-steady shock waves with obstacles, *Journal of Computational and Mathematical Physics USSR*, Vol. 1, pp. 267279, 1961.

[27] C.W.Shu & S.J.Osher, Efficient implementation of essentially nonoscillatory shock capturing schemes II, *Journal of Computational Physics*, Vol. 83, pp.32-78, 1981

[28] G.A.Sod, A survey of several finite difference methods for sytems on non-linear hyperbolic conservation laws, *Journal of Computational Physics*, Vol. 27, pp.1-31, 1981

[29] R. J. Spiteri, S. J. Ruuth, A new class of optimal high-order strong-stability-preserving time discretization methods, *SIAM J. Numer. Anal.*, Vol 40, pp 469-491, 2002

[30] Y. Sun, Z. J. Wang, & Y. Liu, Efficient Implicit LU-SGS Algorithm for high-order spectral difference method on unstructured hexahedral grids, *AIAA paper*, AIAA-2007-313, 2007.

[31] Y. Sun, Z. J. Wang, & Y. Liu, High-order multidomain spectral difference method for the Navier-Stokes equations on unstructured hexahedral grids, *Communication in Computational Physics*, Vol. 2, pp. 310-333, 2007.

[32] Z.J. Wang, Y. Liu, G. May and A. Jameson, Spectral Difference Method for Unstructured Grids II: Extension to the Euler Equations, *Journal of Scientific Computing*, Vol. 32, pp. 45-71, 2007.

[33] Z. J. Wang & Y. Liu, Extension of the spectral volume method to high-order boundary representation, *Journal of Computational Physics*, Vol. 211, pp. 154-178, 2006.

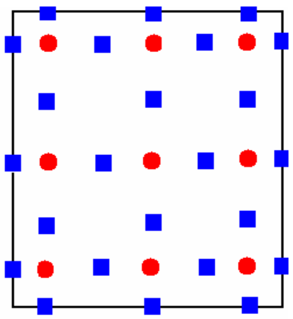
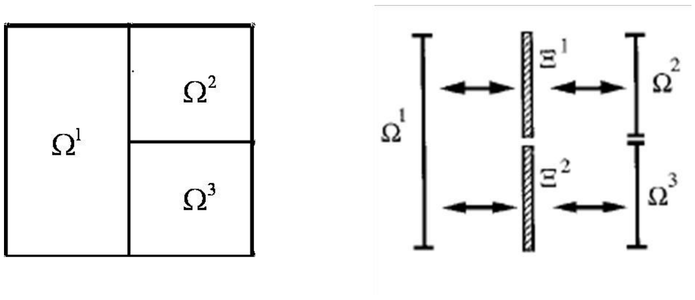


Figure 1. Position of solution (circles) and flux (squares) points on standard square element for 3rd order SD



(a) (b)

Figure 2. (a) Hanging node generated from mesh refinement (b) Mortar element for each child face on interface

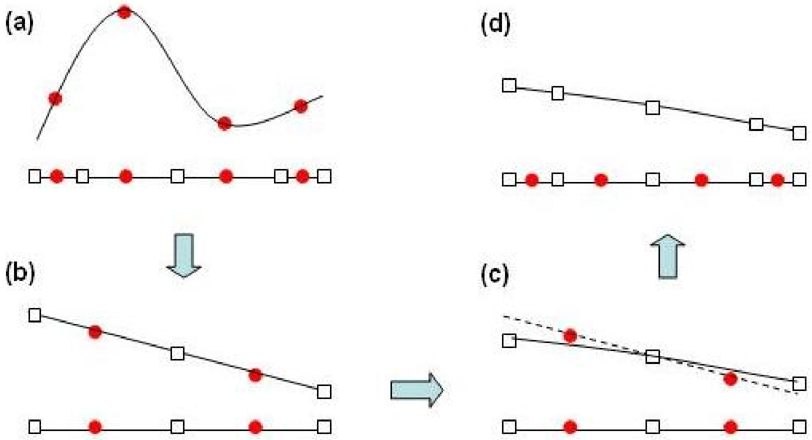


Figure 3. Steps involved in implementation of filter for 4th order SD

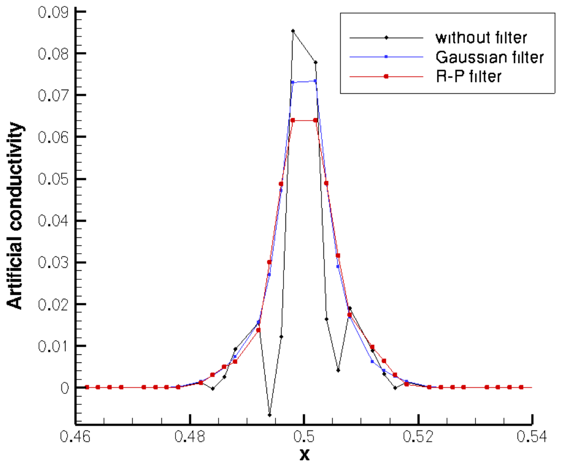


Figure 4. Effect of filtering on the smoothness of artificial viscosity/conductivity coefficients

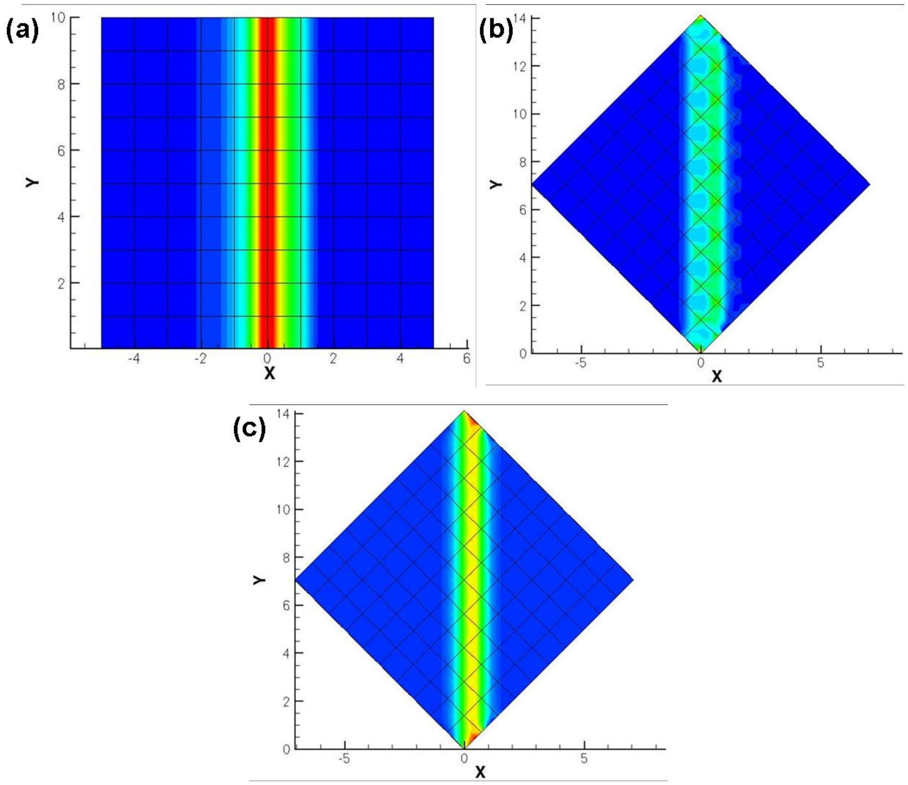


Figure 5. Artificial viscosity contours using using (a) r=2 when discontinuity is aligned with grid lines (b) r=2 when discontinuity is not aligned with grid-lines (c) r=0 when discontinuity is not aligned with grid-lines

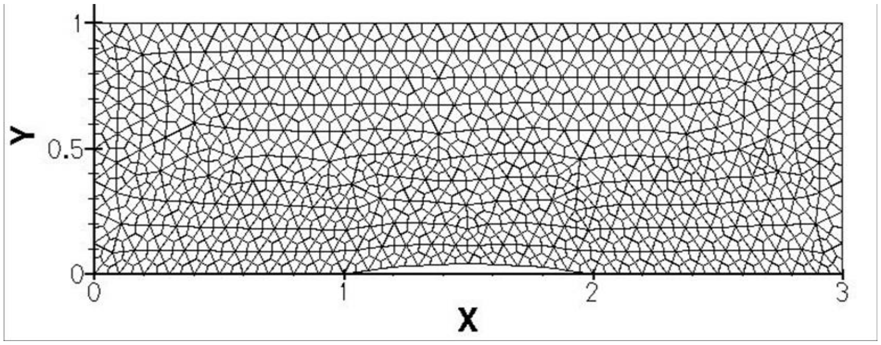


Figure 6. Unstructured grid used for inviscid supersonic flow past bump test-case

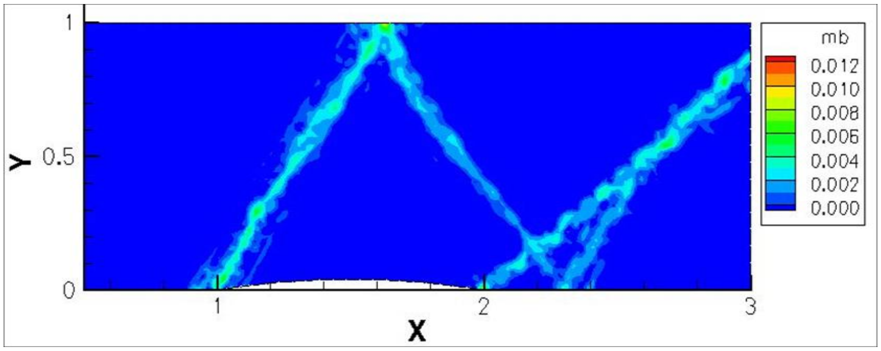


Figure 7. Artificial viscosity contours using r=2 on unstructured supersonic bump mesh

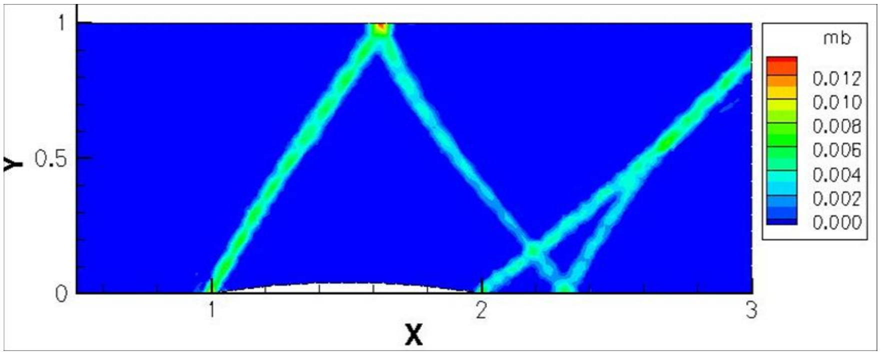
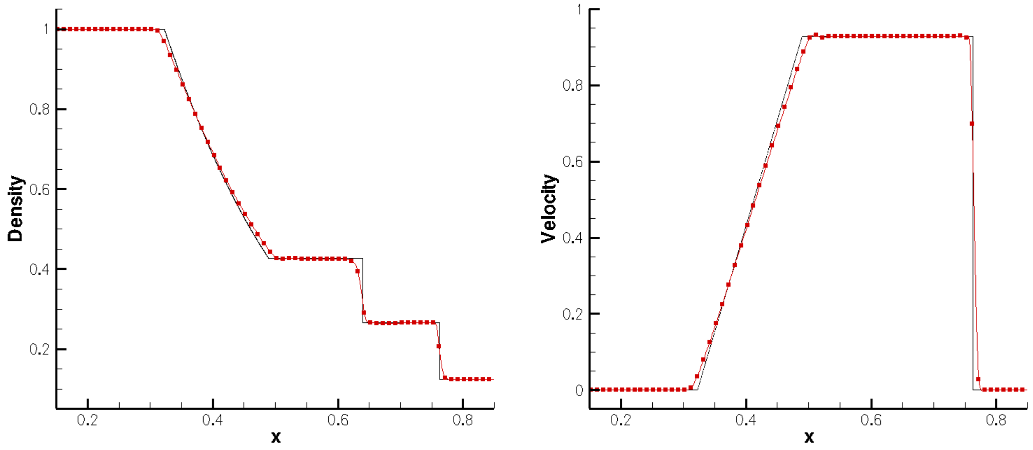
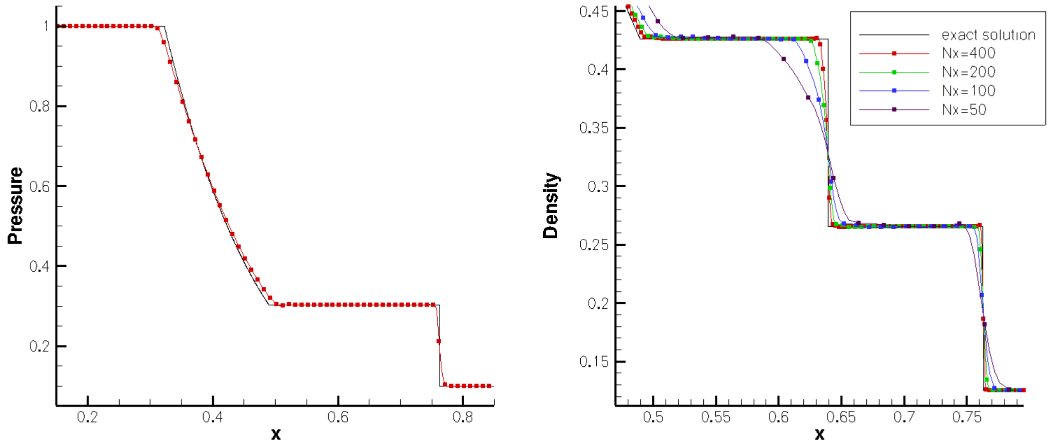


Figure 8. Artificial viscosity contours using r=0 on unstructured supersonic bump mesh



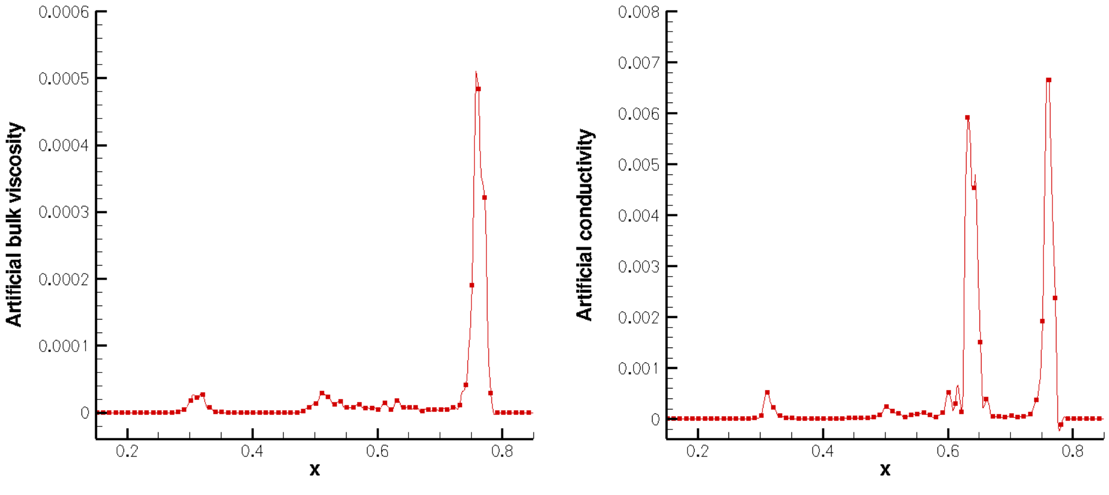
(a) (b)

Figure 9. SOD shock tube case, black line - exact solution, red line - 4th order SD with 100 cells at $\tau = 0.15$ (a) density vs. x ; (b) velocity vs. x.



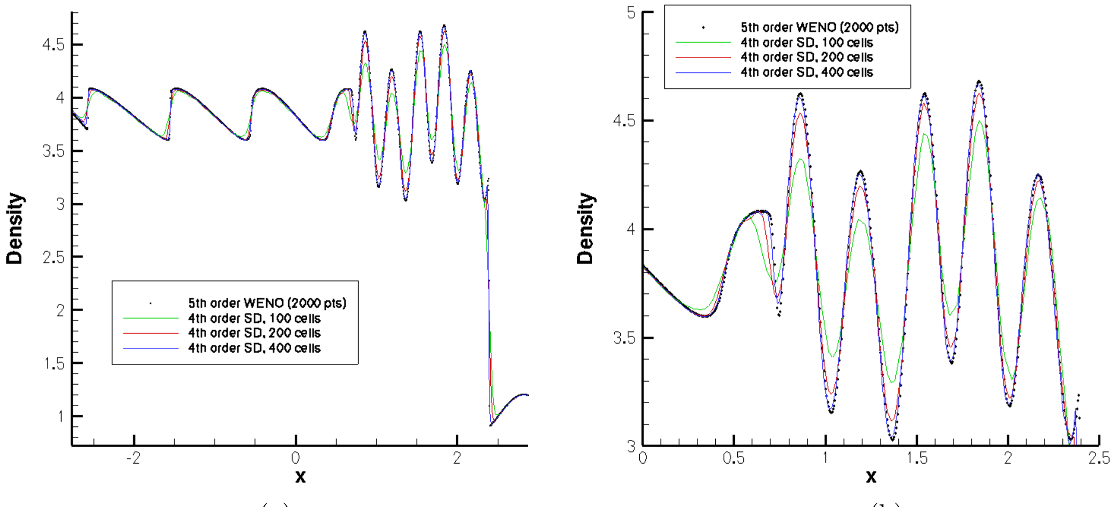
(a) (b)

Figure 10. SOD shock tube case, (c) pressure vs. x ; (d) effect of grid-refinement on density (Nx=number of cells).



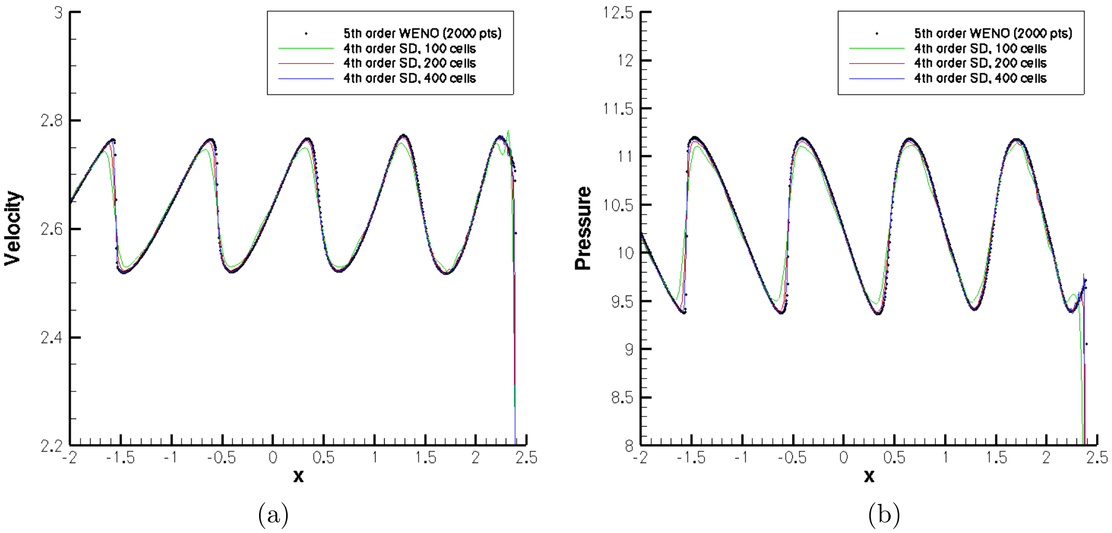
(a) (b)

Figure 11. SOD shock tube case (a) Artificial bulk viscosity; (b) Artificial conductivity at $\tau = 0.15$



(a) (b)

Figure 12. Shu-Osher Shock turbulence interaction. Density is presented at $\tau = 1.8$.



(a)

(b)

Figure 13. Shu-Osher Shock turbulence interaction. Plot of (a)Velocity, (b)Pressure, at $\tau = 1.8$.

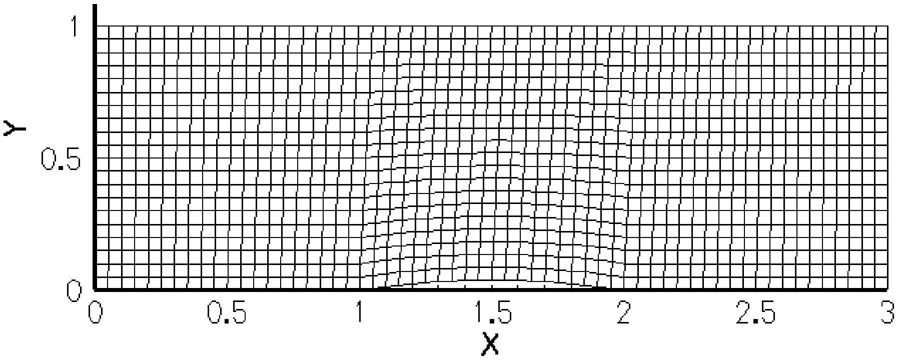


Figure 14. Computational grid for supersonic flow past bump (thickness=4%)

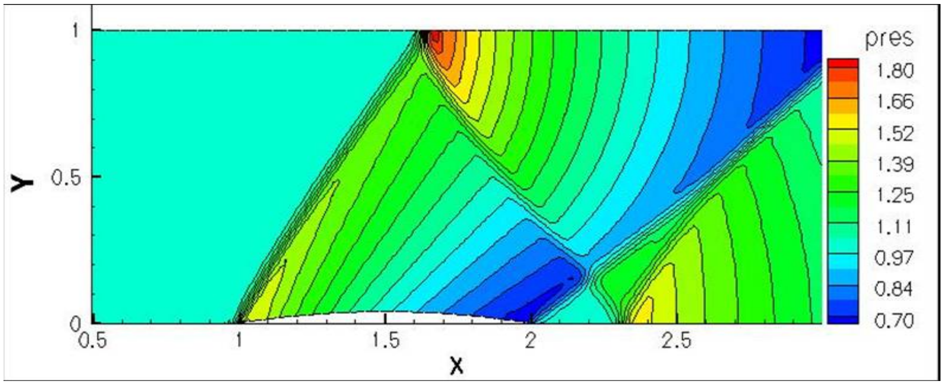


Figure 15. Non-dimensional pressure contours obtained using artificial viscosity with 3rd order SD on the 60×20 mesh.

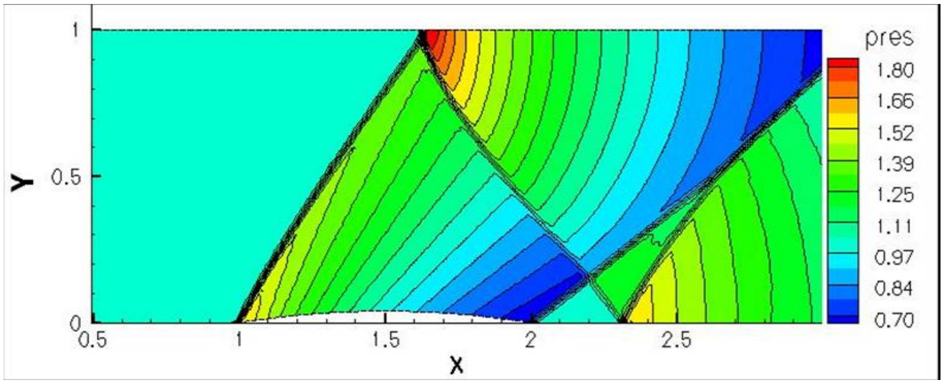


Figure 16. Non-dimensional pressure contours obtained using artificial viscosity with 3rd order SD on the 120×40 **mesh.**

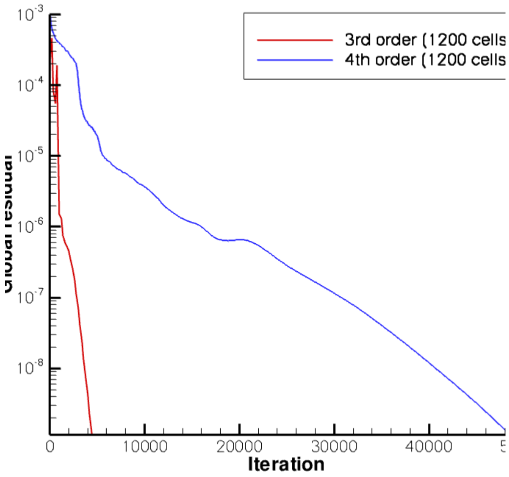


Figure 17. Convergence plot for supersonic bump flow case 3rd and 4th order SD with artificial viscosity

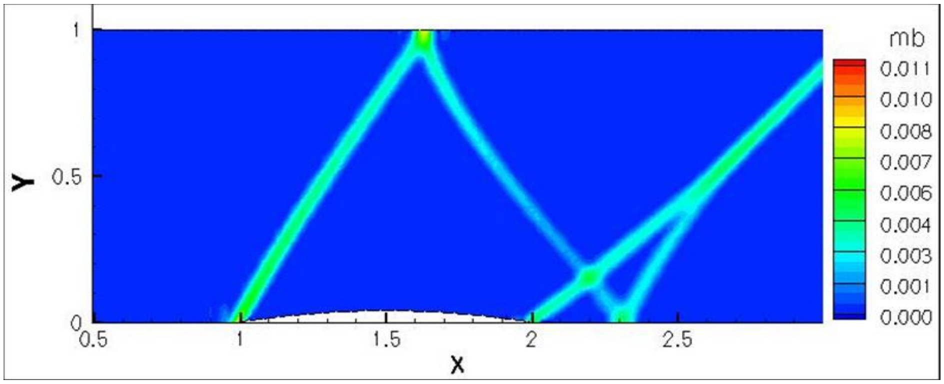


Figure 18. Plot of the artificial bulk viscosity for 3rd order SD computation on 60×20 mesh

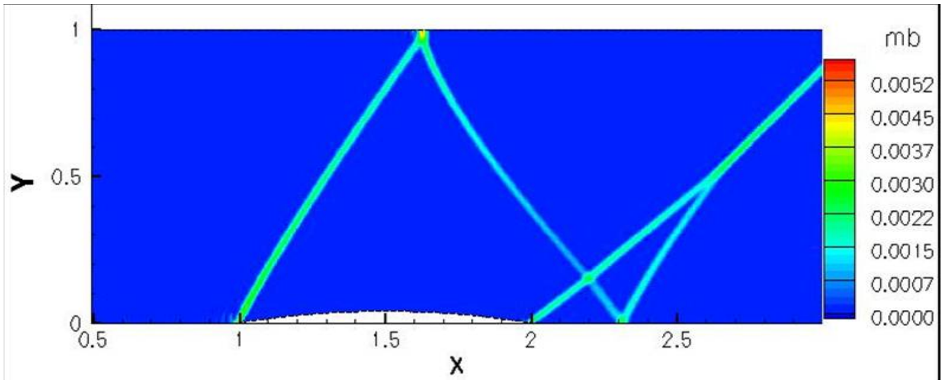


Figure 19. Plot of the artificial bulk viscosity for 3rd order SD computation on 120×40 mesh

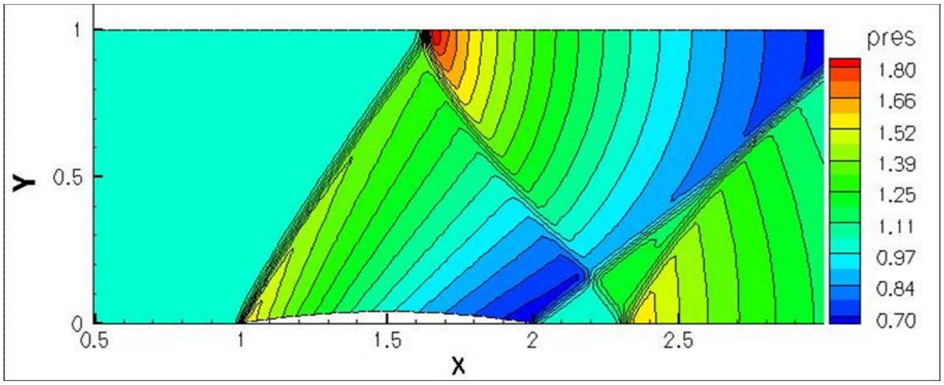


Figure 20. Non-dimensional pressure contours obtained using artificial viscosity with 4th order SD on the 60×20 mesh.

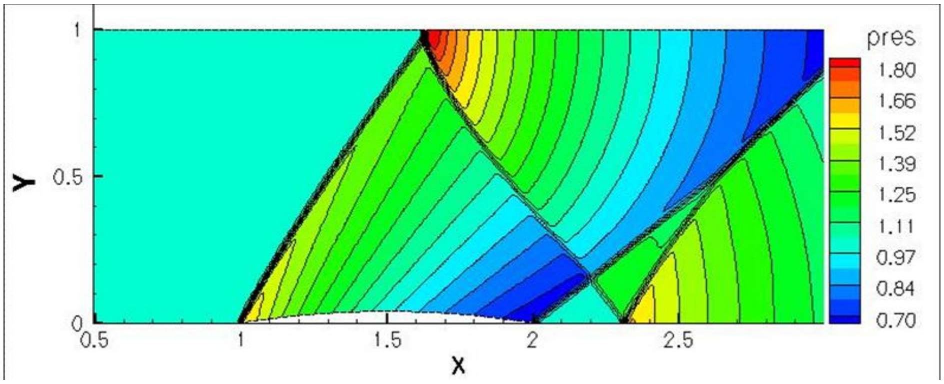


Figure 21. Non-dimensional pressure contours obtained using artificial viscosity with 4th order SD on the 120×40 **mesh.**

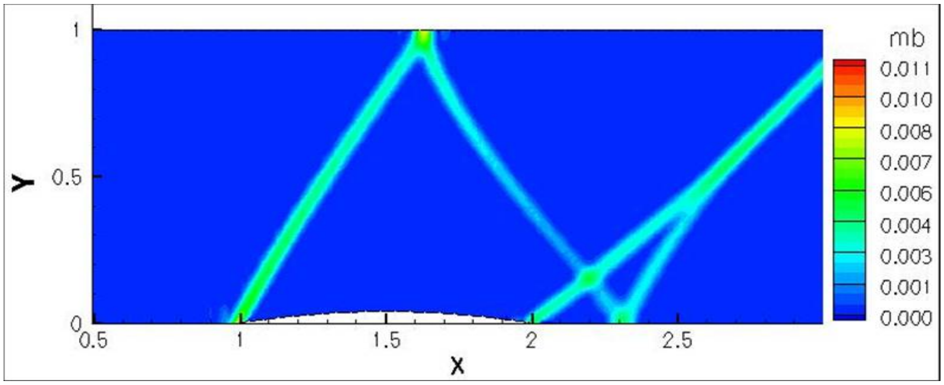


Figure 22. Plot of the artificial bulk viscosity for 4th order SD computation on 60×20 **mesh**

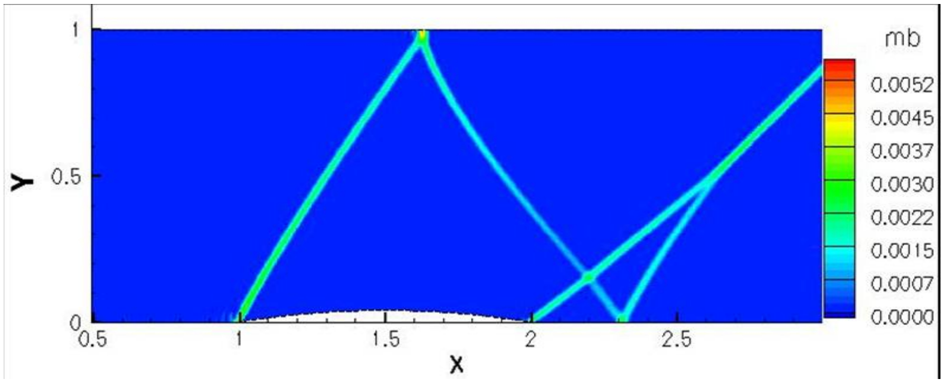
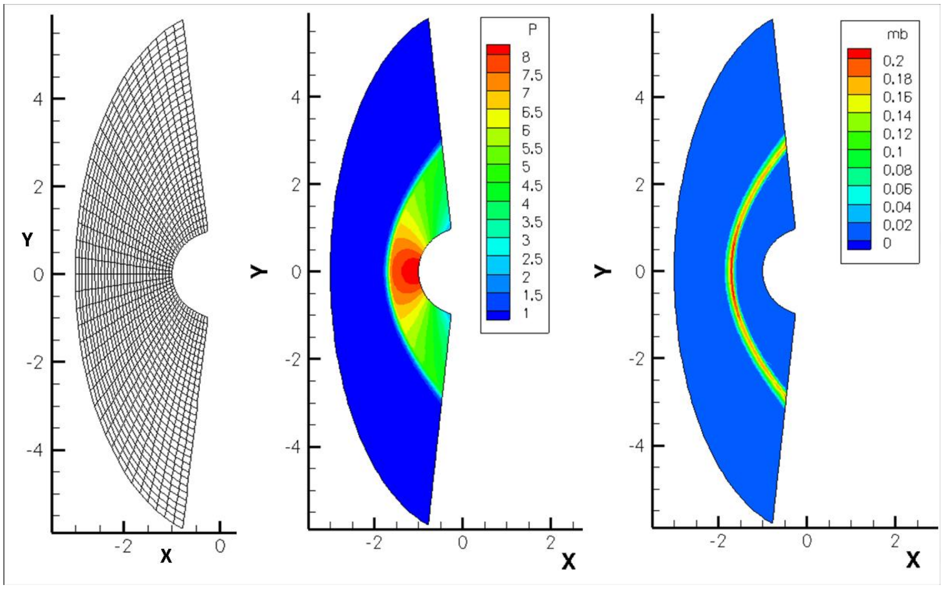
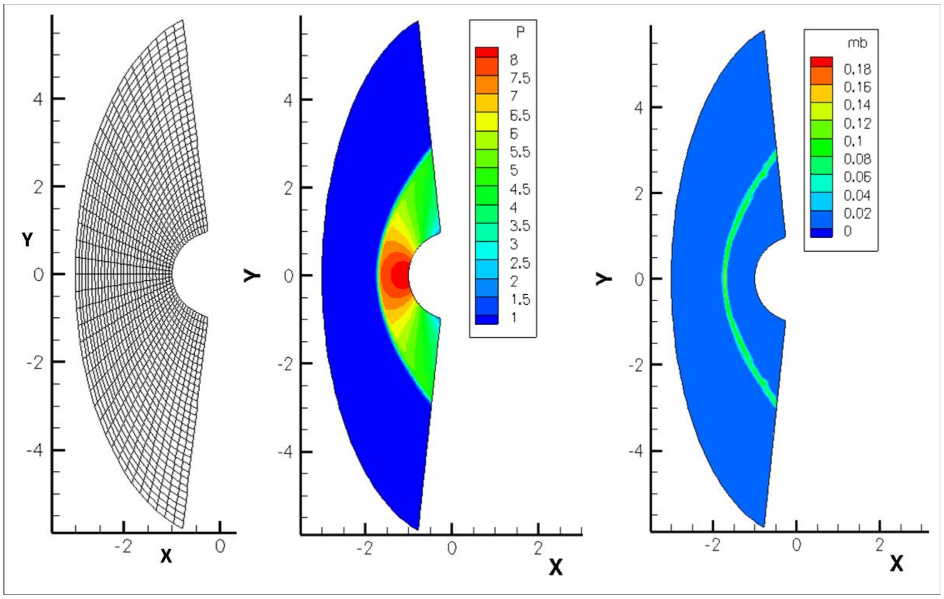


Figure 23. Plot of the artificial bulk viscosity for 4th order SD computation on 120×40 **mesh**



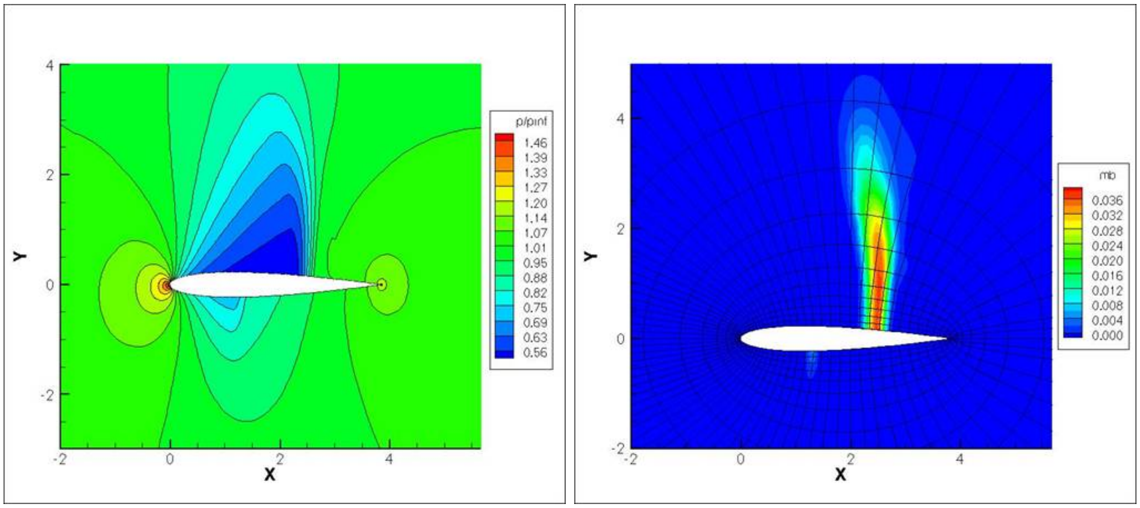
(a) (b) (c)

Figure 24. 3rd order computation of $M_\infty = 3.0$ flow past cylinder (a)40x30 mesh (b)Non-dimensional pressure contours (c) Artificial bulk viscosity contours



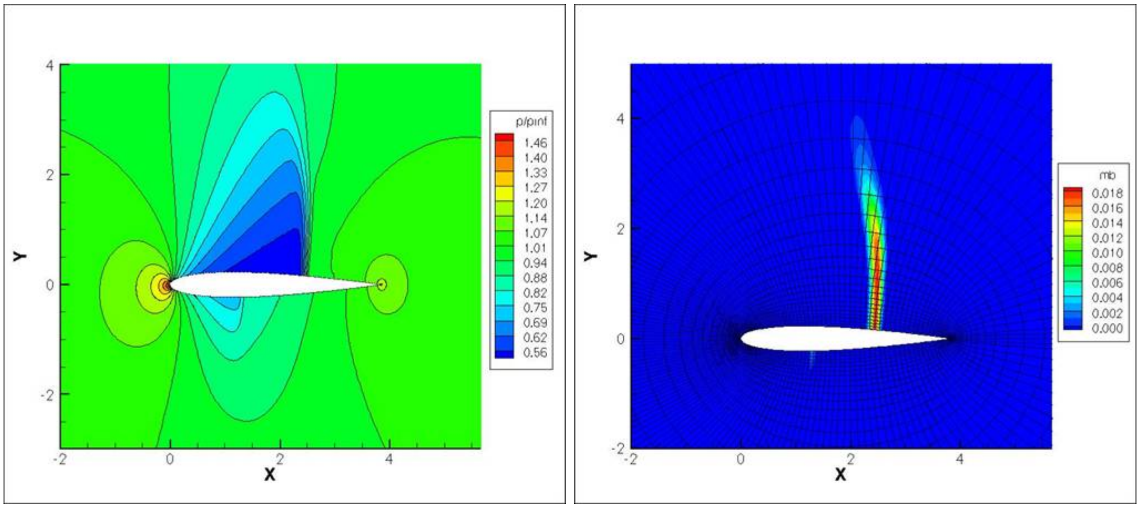
(a) (b) (c)

Figure 25. 4th order computation of $M_\infty = 3.0$ flow past cylinder (a)40x30 mesh (b)Non-dimensional pressure contours (c) Artificial bulk viscosity contours



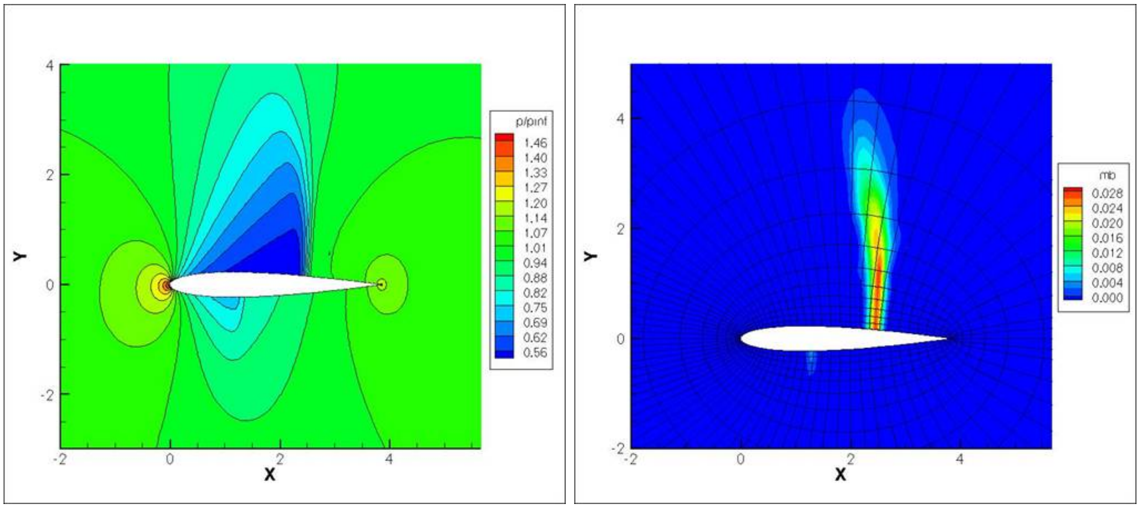
(a) (b)

Figure 26. Transonic flow past NACA0012 airfoil with M_∞=0.8 and $\alpha = 1.25$ degrees. 3rd order computation on coarse (80x16) mesh. (a)Pressure contours (b)Artificial bulk viscosity contours



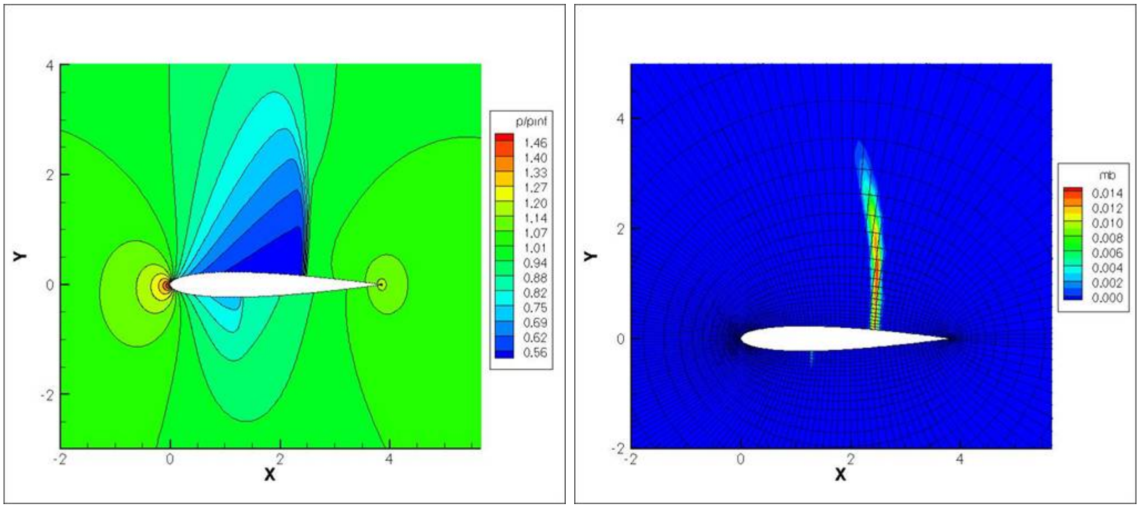
(a) (b)

Figure 27. Transonic flow past NACA0012 airfoil with M_∞=0.8 and $\alpha = 1.25$ degrees. 3rd order computation on fine (160x32) mesh.(a)Pressure contours (b)Artificial bulk viscosity contours



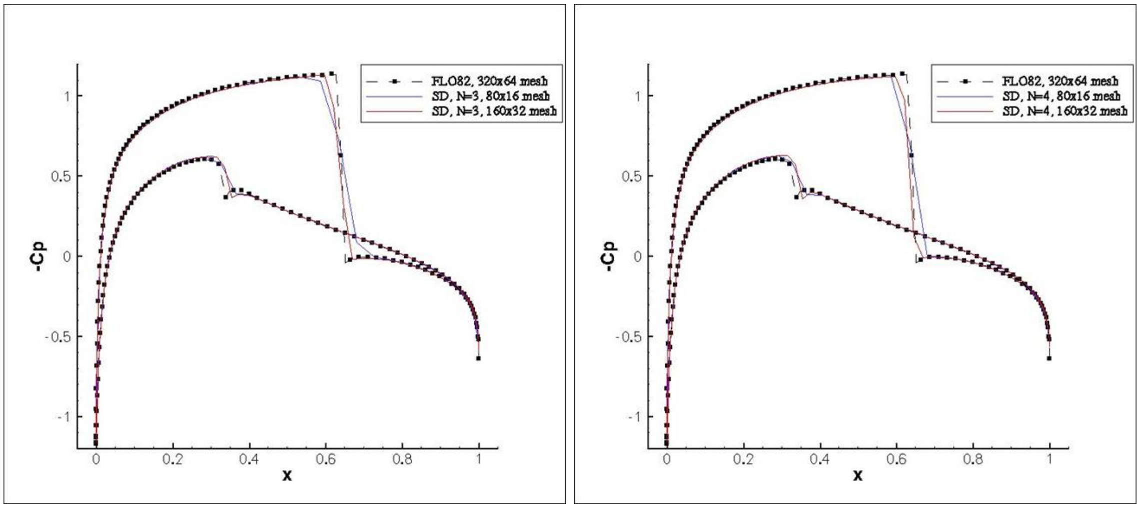
(a) (b)

Figure 28. Transonic flow past NACA0012 airfoil with M_∞=0.8 and $\alpha = 1.25$ degrees. 4th order computation on coarse (80x16) mesh. (a)Pressure contours (b)Artificial bulk viscosity contours



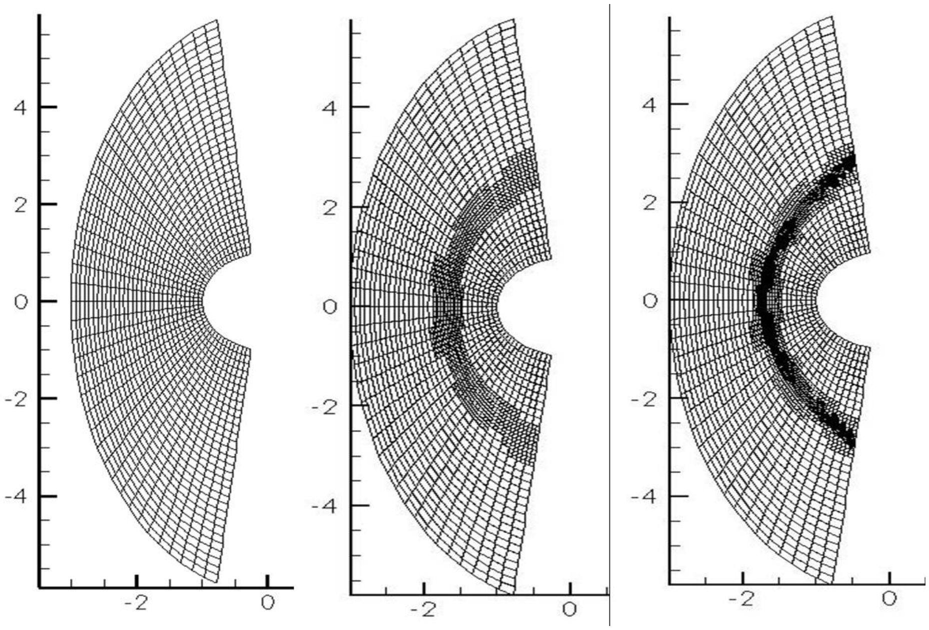
(a) (b)

Figure 29. Transonic flow past NACA0012 airfoil with M_∞=0.8 and $\alpha = 1.25$ degrees. 4th order computation on fine (160x32) mesh.(a)Pressure contours (b)Artificial bulk viscosity contours



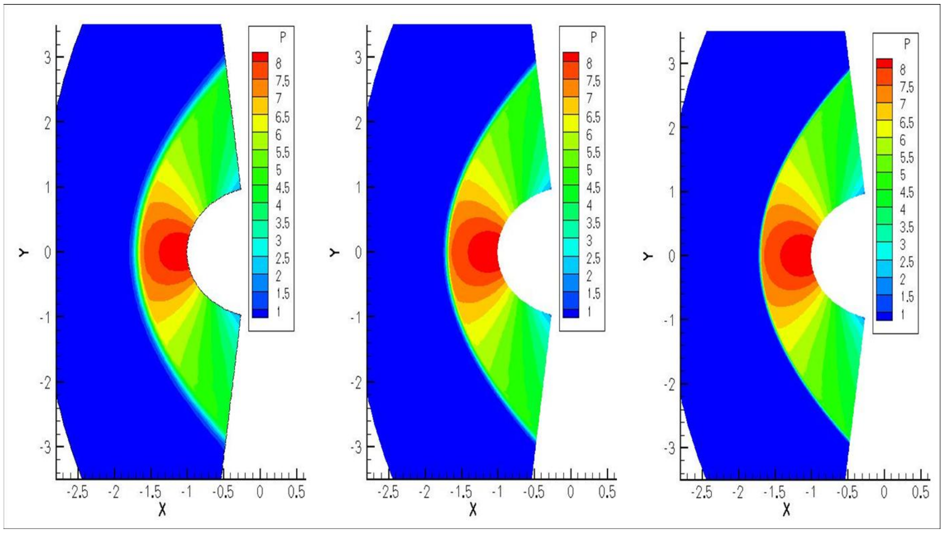
(a) (b)

Figure 30. Transonic flow past NACA0012 airfoil with M_∞=0.8 and $\alpha = 1.25$ degrees. Comparison of Cp plots obtained using present method with those from FLO82 on a 320x64 mesh.(a)3rd order computation (b)4th order computation



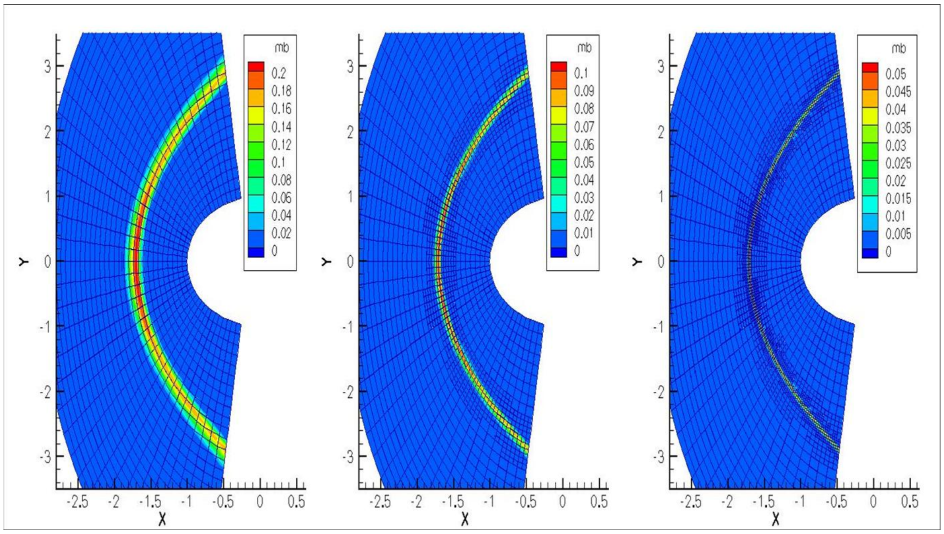
(a) (b) (c)

Figure 31. Successive levels of mesh refinement for $M_\infty = 3.0$ flow past cylinder test-case (a)40x30 initial mesh (b)One level mesh refinement (c) Two levels of mesh refinement



(a) (b) (c)

Figure 32. Pressure contours for 3rd order computation of $M_\infty = 3.0$ flow past cylinder (a)40x30 initial mesh (b)One level mesh refinement (c) Two levels of mesh refinement



(a) (b) (c)

Figure 33. Artificial bulk viscosity contours for 3rd order computation of $M_\infty = 3.0$ flow past cylinder (a)40x30 initial mesh (b)One level mesh refinement (c) Two levels of mesh refinement

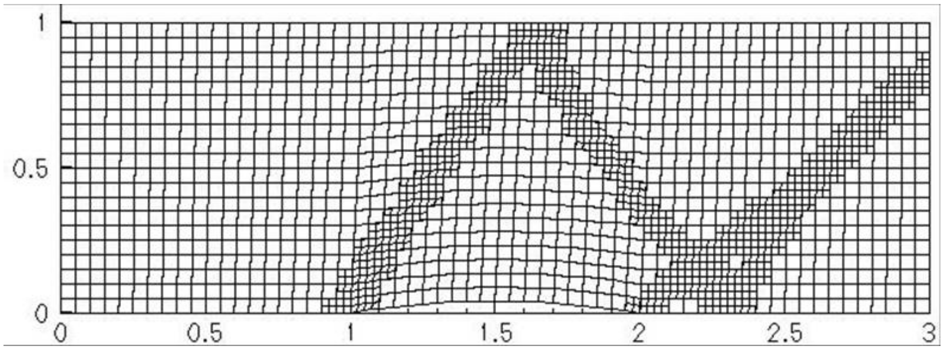


Figure 34. Computational grid for supersonic flow past bump after one level mesh refinement

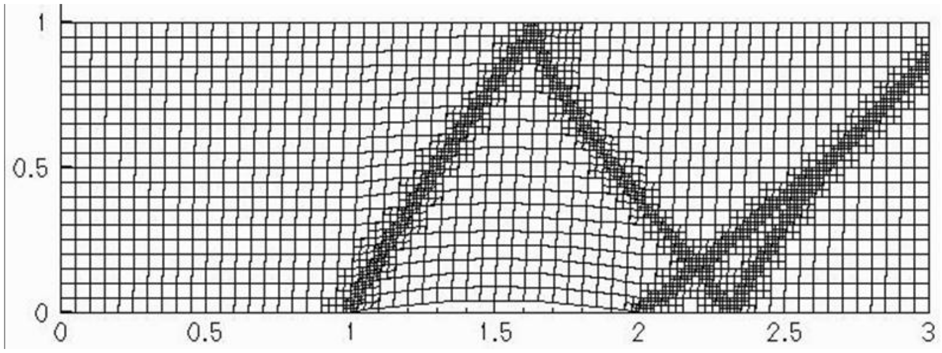


Figure 35. Computational grid for supersonic flow past bump after two levels mesh refinement

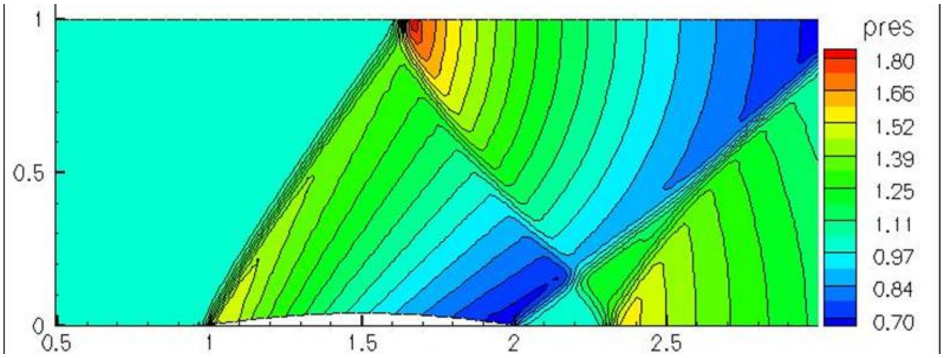


Figure 36. Non-dimensional pressure contours obtained using 4th order SD on initial mesh.

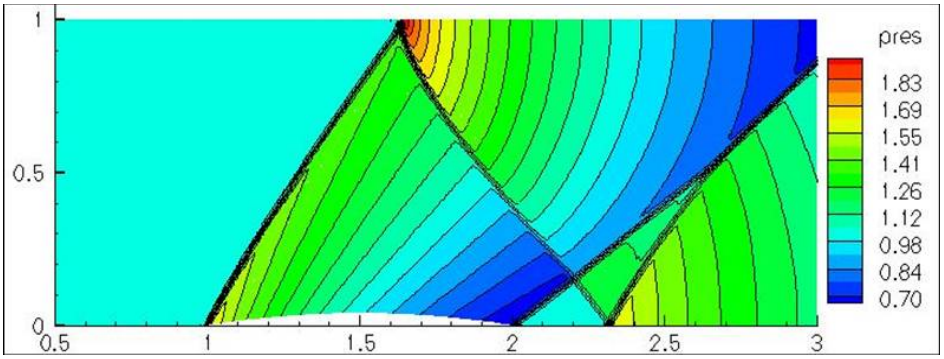


Figure 37. Non-dimensional pressure contours obtained after one level mesh refinement.

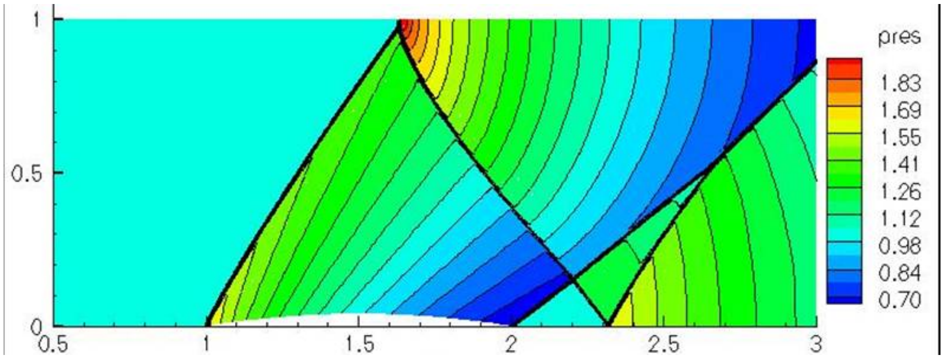


Figure 38. Non-dimensional pressure contours obtained after 2 levels mesh refinement.

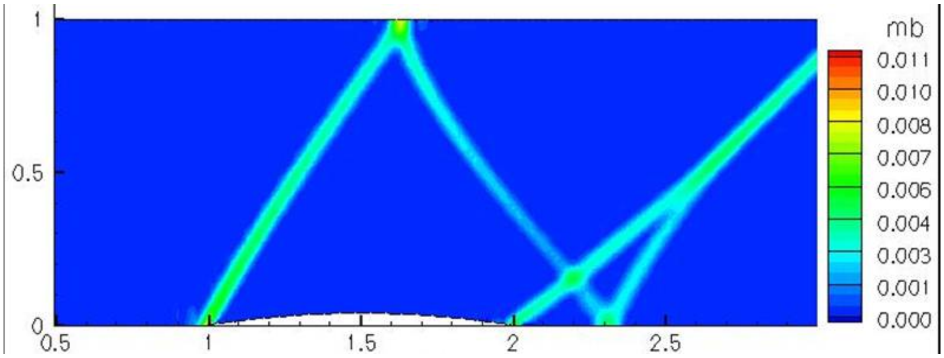


Figure 39. Artificial bulk viscosity contours obtained using 4th order SD on initial mesh.

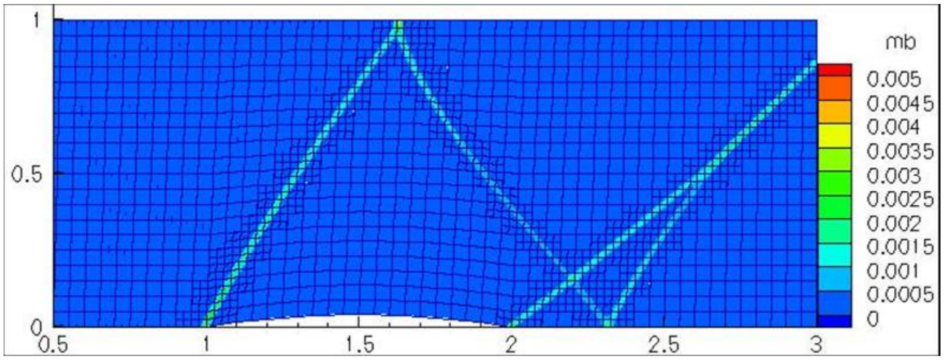


Figure 40. Artificial bulk viscosity contours obtained after one level mesh refinement.

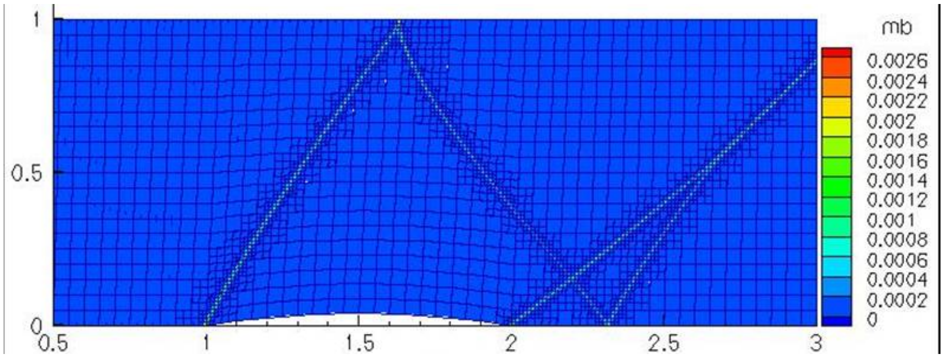


Figure 41. Artificial bulk viscosity contours obtained after 2 levels mesh refinement.