Computers & Fluids 98 (2014) 111–121

Contents lists available at ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

Computation of flows with shocks using the Spectral Difference method with artificial viscosity, I: Basic formulation and application

Sachin Premasuthan [a,*], Chunlei Liang [b], Antony Jameson [a]

[a] Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, United States
[b] Department of Mechanical and Aerospace Engineering, George Washington University, Washington, DC 20052, United States

ARTICLE INFO

Article history:
Received 8 August 2013
Received in revised form 17 November 2013
Accepted 11 December 2013
Available online 27 December 2013

Keywords:
Spectral Difference
Artificial viscosity
Compressible flows
Shock capturing
Unstructured high order

ABSTRACT

The present work combines the Spectral Difference method with an artificial viscosity based approach to enable high-order computation of compressible fluid flows with discontinuities. The study uses an artificial viscosity approach similar to the high-wavenumber biased artificial viscosity approach (Cook and Cabot, 2005, 2004; Kawai and Lele, 2008) [1–3], extended to an unstructured grid setup. The model employs a bulk viscosity for treating shocks, a shear viscosity for treating turbulence, and an artificial conductivity to handle contact discontinuities. The high-wavenumber biased viscosity is found to stabilize numerical calculations and reduce oscillations near discontinuities. Promising results are demonstrated for 1D and 2D test problems.

1. 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 and computational aeroacoustics, has seen the development of higher order schemes for unstructured meshes. In particular, there has been a rise in the popularity and application of locally discontinuous formulations. Methods such as Discontinuous Galerkin (DG) method [4,5], Spectral Volume (SV) method [6,7] and Spectral Difference (SD) method [8,9], Lifting Collocation Penalty (LCP) approach [10], etc. fall under this category.

The SD method is a high-order approach based on the differential form of the conservative equations. This 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. The absence of volume or surface integrals also makes this method efficient. The origins of the SD method can be traced back to 1996, when Kopriva and Kolias [11] and Kopriva [12] introduced their formulation for the solution of the 2D compressible Euler equations on unstructured quadrilateral meshes, which they called the 'Conservative Staggered-Grid Chebyshev Multi-Domain method'. Liu et al. [8] developed a general formulation of this approach on simplex cells and applied it to wave equations on triangular grids. Wang et al. [9] extended it to 2D Euler equations on triangular grids. It was further extended to the 2D N–S equations by May and Jameson [13], and Wang et al. [14]. Sun et al. [15] further developed it for three-dimensional Navier–Stokes equations on hexahedral unstructured meshes. Recently, Jameson [16] obtained a theoretical proof that the SD method is stable for all orders of accuracy in a Sobolev norm provided that the interior flux points are located at the zeros of the corresponding Legendre polynomial. This is valid for the 1D formulation and applies to tensor-product based quadrilateral and hexahedral cells. However, the SD scheme is not stable on simplex elements. In this regard, Balan et al. [17] proposed an alternate formulation of the SD scheme, featuring a flux interpolation technique using Raviart–Thomas spaces, which exhibits linear stability for triangular elements.

One of the greatest challenges with using high-order methods 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 contaminate the solution in smooth regions of the flow often causing the simulations to go unstable. Higher order approximations are less dissipative than their low-order counterparts, and hence it is typically necessary to add explicit dissipation in order to obtain a stable solution. However this has a negative effect on accuracy in the vicinity of the discontinuity. It may also degrade the resolution of turbulent scales due to excessive damping. The development of numerical algorithms that can capture discontinuities and also resolve the scales of turbulence in compressible turbulent flows remains a significant challenge.

* Corresponding author. Tel.: +1 650 799 4703.
E-mail address: sachin.premasuthan@gmail.com (S. Premasuthan).

0045-7930/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compfluid.2013.12.013

A classical approach to shock capturing is the addition of artificial viscosity (AV), pioneered by von Neumann and Richtmeyer [18]. The concept of flexible addition of artificial viscosity/dissipation has been used very successfully by Jameson et al. [19–22], 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 [1,2]. This was followed up with work by Fiorina and Lele [23], on high-order compact difference schemes, wherein artificial diffusivity was added in addition to artificial viscosity. Kawai and Lele [3] 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 and artificial conductivity where needed. This facilitates the capturing of discontinuities by smearing the discontinuity over a numerically resolvable scale. 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 [24] 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 [25] proposed shock-capturing using a combination of higher-order PDE-based artificial viscosity and enthalpy-preserving dissipation operator. The above methods were proposed for high-order Discontinuous Galerkin (DG) discretizations. Nguyen and Peraire [26] proposed an adaptive shock-capturing approach for the hybridizable DG method. Yang and Wang [27] suggested the use of limiters with SD schemes for shock capturing but reported issues with convergence when using limiters.

The current study focuses on extending the artificial viscosity approach proposed by Cook and Cabot [1,2], and modified by Kawai and Lele [3] to computations on unstructured quadrilateral grids using the Spectral Difference scheme. It must be mentioned that the present manuscript is an extended version of the work submitted to the 2009 AIAA CFD Conference [28]. This paper will discuss the salient aspects of implementing artificial viscosity within the Spectral Difference setup. The applicability and limitations of this approach will be demonstrated with test cases in 1D and 2D. The current implementation of artificial viscosity can also be extended to the 3D Spectral Difference scheme.

In Section 2, we look at the formulation of the SD method on unstructured quadrilateral meshes. Section 3 discusses the details of the artificial viscosity method used. In Section 4, we look at the numerical results obtained from the application of the artificial viscosity method to multiple test cases. Section 5 discusses the conclusions of our study and the direction of future efforts.

2. Formulation of 2D Spectral Difference scheme on quadrilateral meshes

The formulation of the equations for the 2D SD scheme on quadrilateral meshes is similar to the formulation of Sun et al. [15] 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 flux vectors in the x and y direction respectively. F and G can be split into inviscid and viscous parts, $F=F_i+F_v$ and $G=G_i+G_v$.

The conservative variables and the inviscid components of the fluxes are given by,

$$Q=\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho E\end{pmatrix} F_i=\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ u(\rho E+p)\end{pmatrix} G_i=\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ v(\rho E+p)\end{pmatrix} \tag{2}$$

where ρ is the density of the fluid, u and v are the cartesian velocity components of the flow, p is the pressure, and E is the specific total energy.

The viscous flux vectors can be written as

$$F_v=-\begin{pmatrix}0\\ \tau_{xx}\\ \tau_{yx}\\ u\tau_{xx}+v\tau_{yx}+\kappa\frac{\partial T}{\partial x}\end{pmatrix} \tag{3}$$

$$G_v=-\begin{pmatrix}0\\ \tau_{xy}\\ \tau_{yy}\\ u\tau_{xy}+v\tau_{yy}+\kappa\frac{\partial T}{\partial y}\end{pmatrix}$$

where the τ's are components of the shear stress tensor, and κ is the thermal conductivity of the fluid. The shear stress tensor is related to the velocity gradients as given below.

$$\begin{aligned}\tau_{xx}&=2\mu\frac{\partial u}{\partial x}+\beta\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)\\ \tau_{yy}&=2\mu\frac{\partial v}{\partial y}+\beta\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)\\ \tau_{yx}&=\mu\left(\frac{\partial v}{\partial x}+\frac{\partial u}{\partial y}\right)\\ \tau_{xy}&=\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\end{aligned} \tag{4}$$

where μ is the dynamic (shear) viscosity coefficient, and β is the bulk viscosity coefficient. The latter is related to the viscous stress caused by a volume change. However, under the Stokes' hypothesis, the bulk viscosity is related to the dynamic viscosity as $\beta=-(2/3)\mu$, and the trace of the shear stress tensor vanishes.

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:

$$\begin{pmatrix}x\\ y\end{pmatrix}=\sum_{i=1}^{K}M_i(\xi,\eta)\begin{pmatrix}x_i\\ y_i\end{pmatrix} \tag{5}$$

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 $M_i(\xi,\eta)$ 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 transferred into the computational domain, and the transformed equations take the following form:

$$\frac{\partial\widetilde{Q}}{\partial t}+\frac{\partial\widetilde{F}}{\partial\xi}+\frac{\partial\widetilde{G}}{\partial\eta}=0 \tag{6}$$

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

$$\begin{pmatrix}\widetilde{F}\\ \widetilde{G}\end{pmatrix}=|J|\begin{pmatrix}\xi_x\xi_y\\ \eta_x\eta_y\end{pmatrix}\begin{pmatrix}F\\ G\end{pmatrix} \tag{7}$$

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

$$X_s = \frac{1}{2}\left[1 - cos\left(\frac{2s-1}{2N}\cdot\pi\right)\right], \quad s = 1,2,\ldots,N. \tag{8}$$

The flux points were selected to be Legendre–Gauss quadrature points plus the two end points 0 and 1, as suggested by Huynh [29]. Choosing $P_{-1}(\xi) = 0$ and $P_0(\xi) = 1$, we can determine the higher-degree Legendre polynomials as

$$P_n(\xi) = \frac{2n-1}{n}(2\xi - 1)P_{n-1}(\xi) - \frac{n-1}{n}P_{n-2}(\xi) \tag{9}$$

The locations of these Legendre–Gauss quadrature points are the roots of equation $P_{N-1}(\xi) = 0$. They are stable (in contrast to the Gauss–Lobatto flux points) and produce more accurate solutions for the SD scheme [30].

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

$$h_i(X) = \prod_{s=1,s\neq i}^{N}\left(\frac{X - X_s}{X_i - X_s}\right) \tag{10}$$

Similarly, 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:

$$l_{i+1/2}(X) = \prod_{s=0,s\neq i}^{N}\left(\frac{X - X_{s+1/2}}{X_{i+1/2} - X_{s+1/2}}\right) \tag{11}$$

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

$$Q(\xi,\eta) = \sum_{j=1}^{N}\sum_{i=1}^{N}\frac{\widetilde{Q}_{i,j}}{|J_{i,j}|}h_i(\xi)\cdot h_j(\eta) \tag{12}$$

Similarly, the reconstructed flux polynomials take the following form:

$$\widetilde{F}(\xi,\eta) = \sum_{j=1}^{N}\sum_{i=0}^{N}\widetilde{F}_{i+1/2,j}\cdot l_{i+1/2}(\xi)\cdot h_j(\eta),$$
$$\widetilde{G}(\xi,\eta) = \sum_{j=0}^{N}\sum_{i=1}^{N}\widetilde{G}_{i,j+1/2}\cdot h_i(\xi)\cdot l_{j+1/2}(\eta) \tag{13}$$

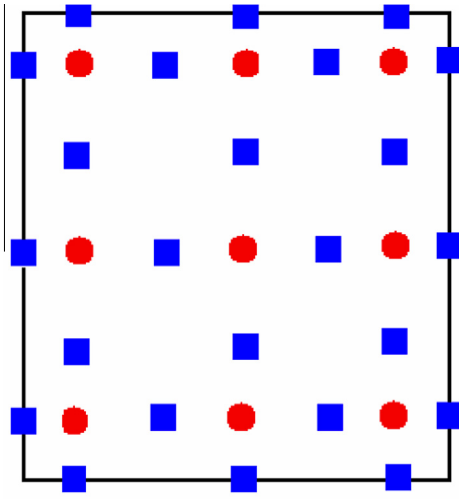

Fig. 1. Position of solution (circles) and flux (squares) points on a standard square element for 3rd order SD.

The reconstructed fluxes are only element-wise continuous, but discontinuous across cell interfaces. For the inviscid flux, a Riemann solver is employed to compute a common flux at interfaces to ensure conservation and stability. In our case, we have used the Rusanov solver [31] 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)

$$\left(\frac{\partial\widetilde{F}}{\partial\xi}\right)_{i,j} = \sum_{r=0}^{N}\widetilde{F}_{r+1/2,j}\cdot l'_{r+1/2}(\xi_i), \tag{14}$$

$$\left(\frac{\partial\widetilde{G}}{\partial\eta}\right)_{i,j} = \sum_{r=0}^{N}\widetilde{G}_{i,r+1/2}\cdot l'_{r+1/2}(\eta_j) \tag{15}$$

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 Ref. [15] 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 Eq. (12).
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 similar to Eq. (14), where $\nabla Q = \begin{Bmatrix} Q_x \\ Q_y \end{Bmatrix}$ 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 [32] to advance in time.

3. Artificial viscosity

The current work proposes the use of an artificial viscosity approach similar to the 'Local artificial viscosity and diffusivity' approach of Kawai and Lele [3]. Their method was a modification of the original high-wavenumber biased artificial viscosity approach introduced by Cook and Cabot [1], extended to anisotropic and curvilinear structured grids. The present formulation attempts to extend this approach to a SD setup on unstructured quadrilateral meshes.

This approach adds grid-dependent components to the viscosity coefficients, as proposed by Kawai and Lele,

$$\mu = \mu_f + \mu_\Delta,$$
$$\beta = \beta_f + \beta_\Delta,$$
$$\kappa = \kappa_f + \kappa_\Delta \tag{16}$$

where μ is the dynamic (shear) viscosity, β is the bulk viscosity, and κ is the thermal conductivity (see Eqs. (3) and (4)). The f and Δ subscripts denote the fluid and artificial transport coefficients respectively.

These artificial transport coefficients are defined by:

$$\mu_\Delta = C_\mu \rho \overline{|\sum_{l=1}^{3}\sum_{m=1}^{3} \Delta_l^{r+2} \left(\frac{\partial \xi_l}{\partial x_m}\right)^r \frac{\partial^r S}{\partial \xi_l^r}|}$$
$$\beta_\Delta = C_\beta \rho \overline{|\sum_{l=1}^{3}\sum_{m=1}^{3} \Delta_l^{r+2} \left(\frac{\partial \xi_l}{\partial x_m}\right)^r \frac{\partial^r (\nabla \cdot \mathbf{u})}{\partial \xi_l^r}|}$$
$$\kappa_\Delta = C_\kappa \overline{\frac{\rho c_s}{T} |\sum_{l=1}^{3}\sum_{m=1}^{3} \Delta_l^{r+1} \left(\frac{\partial \xi_l}{\partial x_m}\right)^r \frac{\partial^r e}{\partial \xi_l^r}|} \tag{17}$$

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. ρ, c_s and T are the density, speed of sound and temperature respectively. 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 ($\nabla \cdot \mathbf{u}$), and the internal energy (e) are the sensors corresponding to artificial shear viscosity, bulk viscosity and conductivity respectively. Studies on structured grids suggest that r equals 4 or higher [33]. For a given wavenumber k and sufficiently high r, the high-wavenumber bias (k^r) results in damping of wavenumbers close to the unresolved wavenumbers. The overbar denotes a filter to smooth the artificial transport coefficients. In structured grid calculations, a truncated Gaussian filter is used [1]. The filter is also meant to eliminate cusps introduced by the absolute value operator, which in turn ensures that artificial viscosities are positive.

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. [21], in schemes like JST (Jameson–Schmidt–Turkel) and SLIP (Symmetric Limited Positive), where the artificial dissipation is 3rd order in smooth regions of the flow and first order when there is a discontinuity.

An advantage of such an artificial viscosity scheme is that it eliminates the need for limiters and switches to turn off the artificial bulk viscosity in regions of expansion and isentropic compression. Also, the artificial shear viscosity is automatically zero in regions of uniform shear. The artificial viscosity/conductivity is calculated at each flux point. This is computationally expensive, but it ensures smooth variation of artificial transport coefficients. A smooth representation of artificial viscosity within mesh elements is advantageous over piecewise-constant artificial viscosity formulations, as element-to-element variations can lead to oscillations in state gradients due to disparate artificial fluxes across neighboring elements near the shock location.

The steps involved in the implementation of the artificial viscosity method to the SD solver can be explained as follows. Consider the calculation of artificial shear viscosity.

1. The sensor S (strain rate) is computed at the solution points.
2. It is then extrapolated to the flux points, and averaged (left and right cells values) at the interface flux points.
3. $\frac{\partial S}{\partial \xi_l}$ is computed at the solution points, then reconstructed to the flux points, and averaged at the interface flux points.
4. Another differentiation operation, gives $\frac{\partial^2 S}{\partial \xi_l^2}$ at the solution points.
5. The above steps can be repeated required number of times to obtain $\frac{\partial^r S}{\partial \xi_l^r}$ at the solution points.
6. Similarly, the partial derivatives of the other sensors can also be computed.
7. The artificial transport coefficients are computed at the solution points using Eq. (17).
8. The coefficients are then filtered and reconstructed to the flux points.

Since the approach requires the computation of higher order derivatives, there is an implied limitation on the order of the method. For instance, when using $r = 2$, we would like the solution accuracy to be 3rd order or higher. To explain this, consider the 3rd order computation. Step 1 begins with the velocity values at 4 flux points to compute the strain rate S at the 3 solution points. This S, which is a degree 2 polynomial, is differentiated twice (with interface averaging at each step) to get the artificial viscosity. It is clear that any further differentiation would only add error into the system. This also implies that for 3rd and 4th order computations, using $r = 4$ may be undesirable.

3.1. 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 [1]. 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 and Schmidt [34] 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 the implementation of the R-P filter can be described in 1D as follows,

1. Consider a 4th order SD element. The artificial viscosity terms have been computed at the 4 solution points (Fig. 2(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) (Fig. 2(b)). 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 (Fig. 2(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) (Fig. 2(d)).

To illustrate the effect of the R-P filter, we use Fig. 3(a) which 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 artificial coefficients were computed using $r = 2$. The figure shows that prior to filtering the artificial conductivity field is noisy and may have oscillatory behavior.

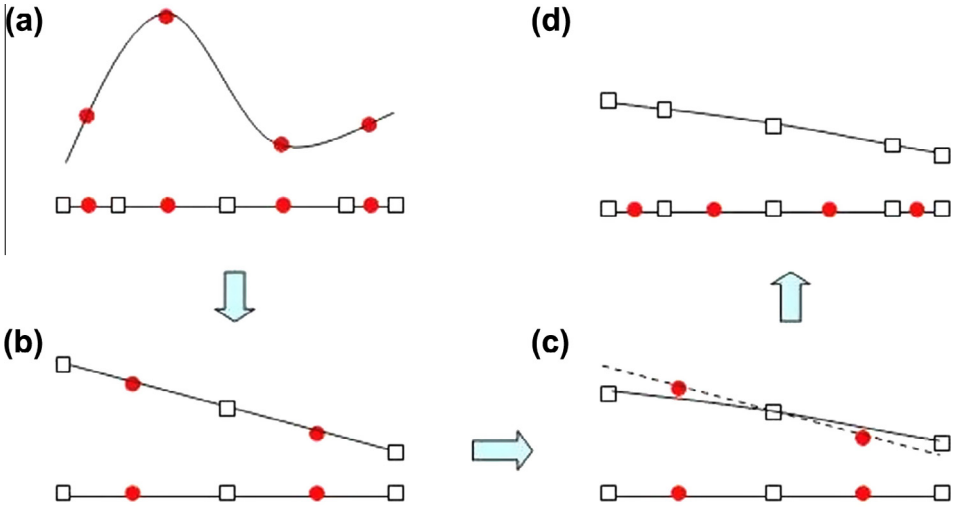

Fig. 2. Steps involved in the implementation of the filter for 4th order SD. Circles represent the solution points and squares the flux points.

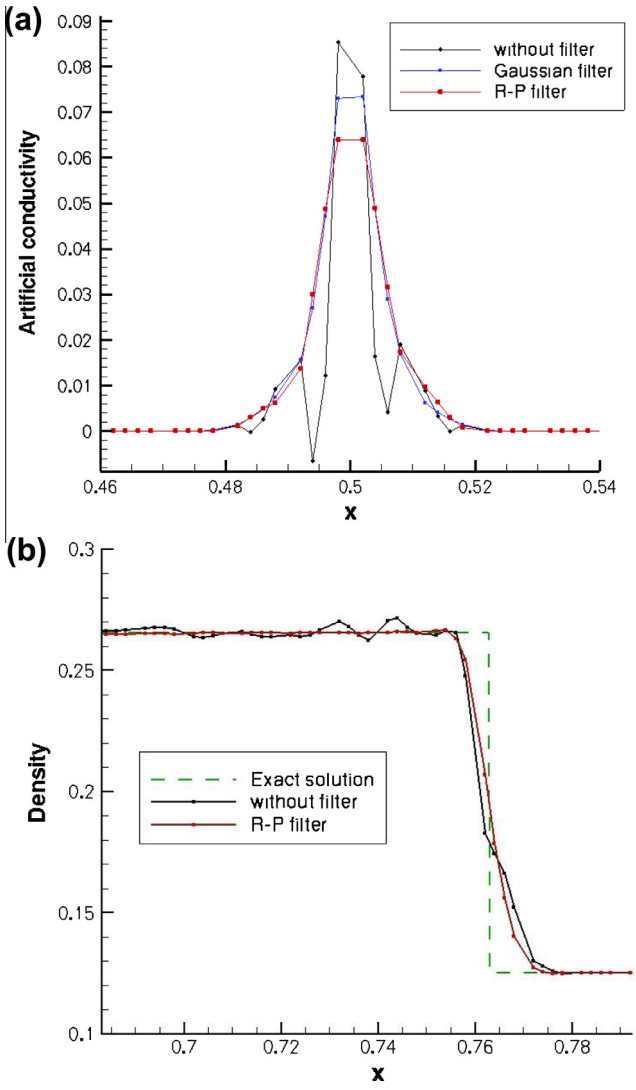

Fig. 3. (a) Effect of filtering on the smoothness of artificial viscosity/conductivity coefficients (at $\tau = 0$). (b) Effect of filtering on density profile for shock (at $\tau = 0.15$).

The filtered coefficient is smoother and results in a better solution. Fig. 3(b) shows the density jump for the advancing shock front at time $\tau = 0.15$. It is clear that non-smoothed artificial viscosity profile is not as effective in eliminating the non-spurious oscillations. Also, in 1D, this filter performs comparably (in the qualitative sense) to the Gaussian filter applied on flux points. It must be mentioned that the effect of the filter is more significant for higher values of r. Since the solution representation in multiple dimensions is just a tensor product of 1D polynomials, the extension of this filter to 2D and 3D is straight-forward.

4. Results

It must be mentioned that the SD scheme has been implemented in 1D and 2D solvers which have been tested, validated and found to exhibit formal order accuracy [35,36]. In this section, the results obtained from the application of SD scheme with artificial viscosity to problems with shocks, are discussed. It must be mentioned that all the test-cases correspond to inviscid flows and involve the solution of the Euler equations along with viscous terms corresponding to the artificial viscosity terms. Firstly, three 1D cases are demonstrated, namely the Sod shock-tube case, the Shu–Osher shock-entropy wave interaction case and the stationary shock case. Then the application of artificial viscosity for shock-capturing in 2D is demonstrated with the reflecting oblique shock wave test case and the supersonic flow past bump test case. In the artificial viscosity formulation, the value of parameter r used was 2 for all the test cases presented here. As explained previously, using $r = 4$ for 3rd and 4th order computations only results in additional computational effort and possibly higher error from high-order derivative calculations. To obtain the user-defined coefficients for the 1D problems, we started with a high starting value ($\mathcal{O}(1)$) to obtain a starting solution. The value was then reduced until we got to the smallest value that enabled a stable and sharp capture of the discontinuity with small overshoot ($\leqslant 1\%$).

4.1. Sod Shock-tube problem

The first 1D test case is the shock-tube problem introduced by Sod [37]. 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 $x > 0.5$, where ρ, u and p

correspond to the flow density, velocity and pressure respectively. Simulations are performed on a uniformly spaced grid in the region $0 \leqslant x \leqslant 1$. 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 Eq. (17)), corresponding to second order derivatives of the sensor quantities. C_μ was set to zero. 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.

Figs. 4(a), (b) and 5(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. Fig. 5(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. Fig. 6(a) shows the artificial bulk viscosity coefficient. It is seen to be maximum in the vicinity of the shock. Fig. 6(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.

4.2. Shu–Osher problem

The second 1D test case is the shock-entropy wave interaction introduced by Shu and Osher [38]. 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 $x \geqslant -4$. Simulations are performed on a uniformly spaced grid in the region $-5 \leqslant x \leqslant 5$. $r = 2$ was used with the user-defined coefficients same as those used in the Sod case.

Fig. 7(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 obtained using 400 cells shows excellent

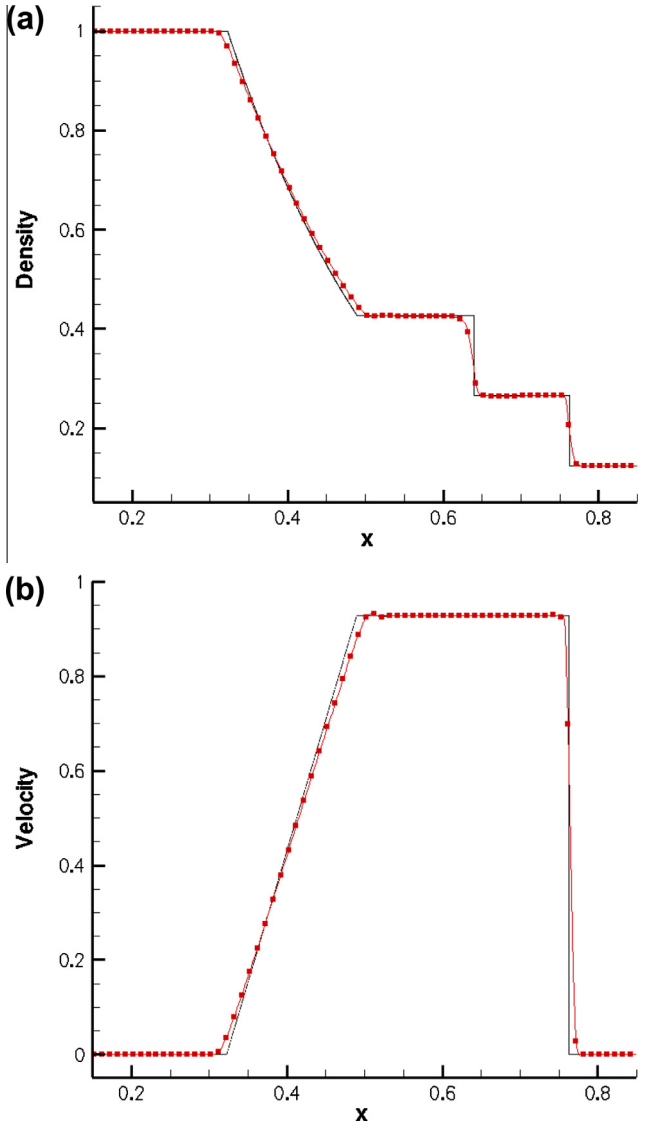

Fig. 4. Sod shock tube case, black line – exact solution, red line – 4th order SD with 100 cells at $\tau = 0.15$ (a) density vs. x; (b) velocity vs. x. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

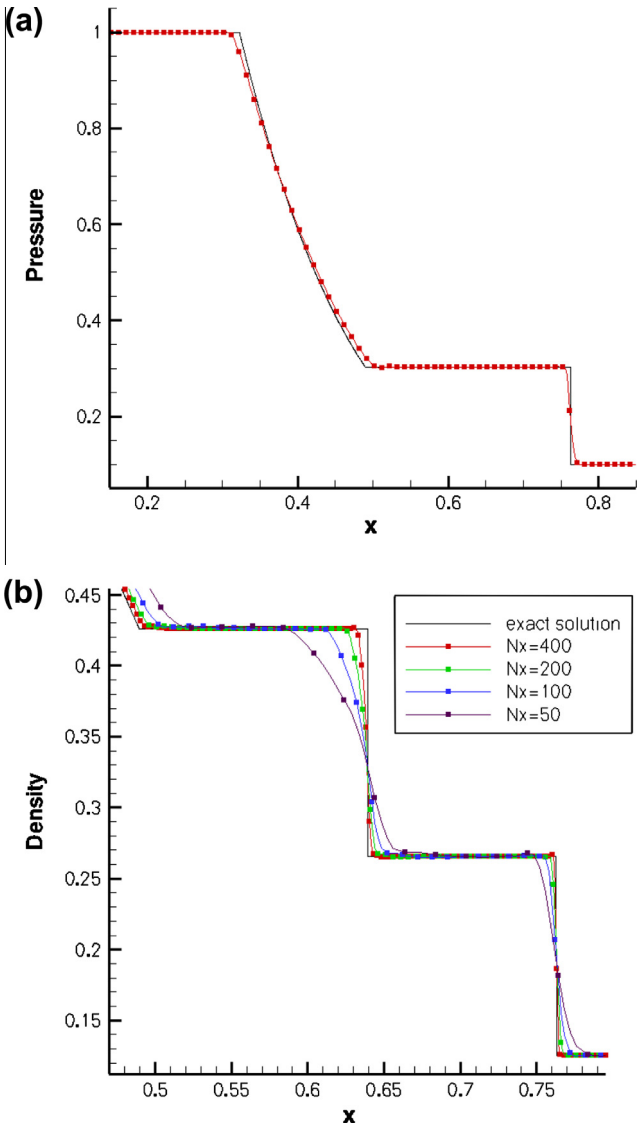

Fig. 5. Sod shock tube case, (c) pressure vs. x; (d) effect of grid-refinement on density (Nx = number of cells).

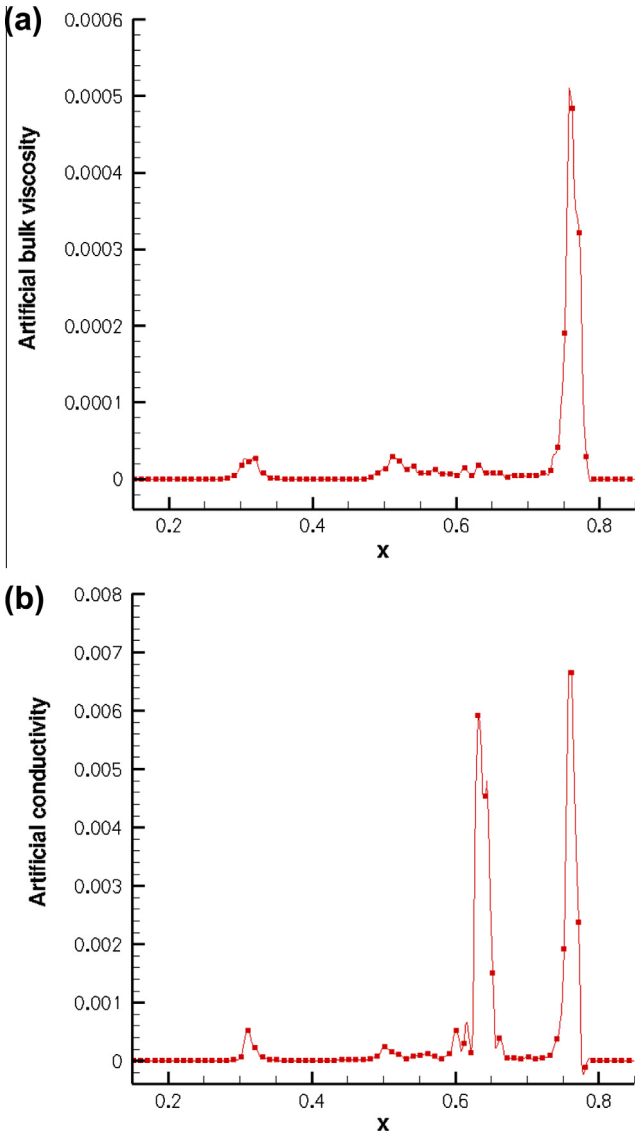

Fig. 6. Sod shock tube case (a) Artificial bulk viscosity; (b) artificial conductivity at $\tau = 0.15$.

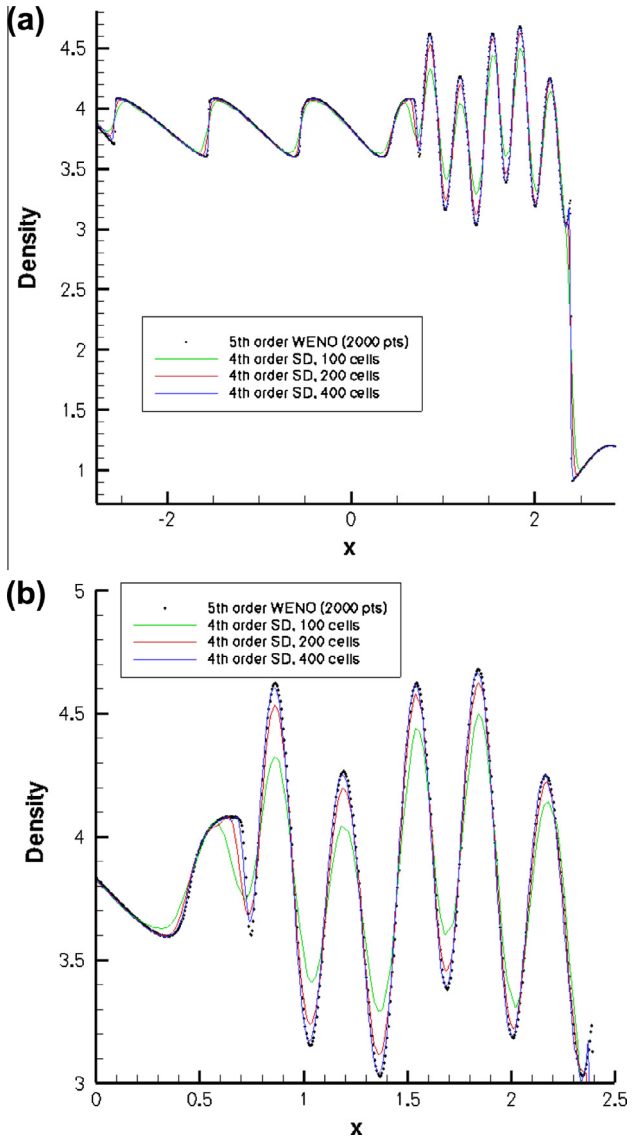

Fig. 7. Shu–Osher shock turbulence interaction. Density is presented at $\tau = 1.8$.

agreement with the reference solution. Fig. 7(b) shows a close up of the density plot in the region of the entropy waves. It is clear that the solution with 200 cells also qualitatively shows reasonable agreement. The velocity and pressure profiles (Fig. 8(a) and (b)) highlight the favorable effect of mesh refinement on solution quality in the close vicinity of the sharp discontinuity (near $x = 2.4$).

4.3. Stationary shock in 1D test case

This 1D test case corresponds to stationary normal shock at Mach number 3.0. The initial conditions for the flow are prescribed using the Rankine–Hugoniot relations for a stationary shock. The computational domain extends between $0 \leqslant x \leqslant 1$ and the shock is located at $x = 0.5$. For this case, we used $r = 2$ with $C_\beta = 0.01$. 3rd order computations were done with 50, 100, 200 and 400 cells. The stationary shock is captured well without significant oscillations or overshoot (see Fig. 9(a)). As the mesh is refined, the shock profile becomes sharper and the result converges to the exact solution. We also see that the shock profiles have a very small overshoot. Fig. 9(b) plots the percentage overshoot (normalized by the shock pressure jump) versus the grid spacing. It is clear that as the grid spacing decreases, the percentage overshoot decreases.

4.4. Two-dimensional oblique shock-reflection

The first 2D shock test case is the oblique shock reflection on an inviscid wall. The free-stream Mach number is 3 and the oblique shock angle is 33°. The computational domain extends from $x = -1.5$ to $x = 1.5$ and $y = 0$ to $y = 1$. Supersonic inlet and outlet boundary conditions are used. The exact shock jump conditions corresponding to oblique shock at $M = 3$, $\alpha = 33^\circ$ at $x = -1$, are imposed on the upper boundary. The lower boundary is an inviscid wall.3rd order SD simulations were run. Fig. 10 shows the pressure contours obtained using a 60×20 mesh (1200 cells). The next figure (Fig. 11) gives a comparison of pressure contours obtained using the coarse mesh and a finer 120×40 mesh (4800 cells). As expected the finer mesh gives a sharper shock profile, and smoother pressure contours. Pressure profiles along the $y = 0.18$ line for both meshes are shown in Fig. 12. The pressure profiles compare well with the pressure profile obtained using artificial viscosity with 6th order Compact Difference on a structured 301×101 mesh [3].

The above computations were computed using $r = 2$ and $C_\beta = 0.01$. Also, the artificial shear viscosity and artificial conductivity are set to zero. This is because there are no large shear

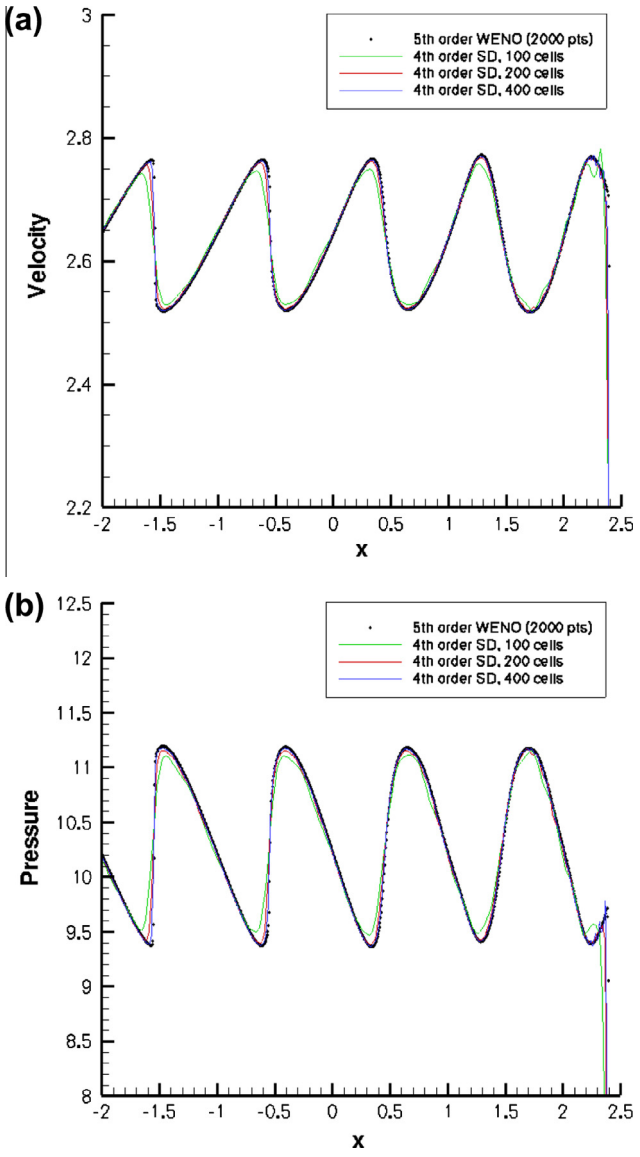

Fig. 8. Shu–Osher shock turbulence interaction. Plot of (a) velocity, (b) pressure, at $\tau = 1.8$.

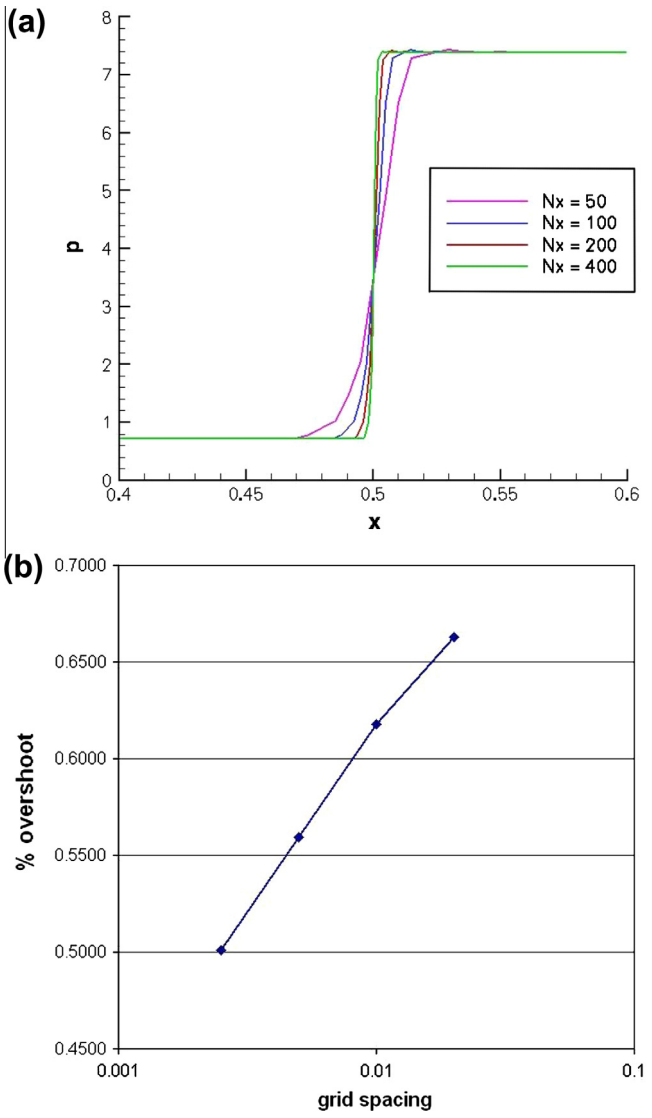

Fig. 9. Stationary shock test case. (a) Pressure profiles for shock discontinuity. (b) Variation of percentage overshoot with grid spacing.

gradients, and no contact discontinuities, and hence artificial bulk viscosity is sufficient to stabilize the calculations.

4.5. 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. [39] in computations using adaptive unstructured mesh refinement. 3rd and 4th 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 Fig. 13. 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 3rd and 4th order calculations respectively. The above computations were computed using $r = 2$ and $C_\beta = 0.008$.

The pressure contours obtained using the 3rd order SD scheme with artificial viscosity on the coarse mesh is shown in Fig. 14(a), and compares well with those obtained using adaptive unstructured mesh refinement [39]. The pressure contours obtained for the fine mesh are shown in Fig. 14(b). It is observed that on the finer mesh, the shock profiles are sharper, and smoother contours are obtained. Fig. 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.

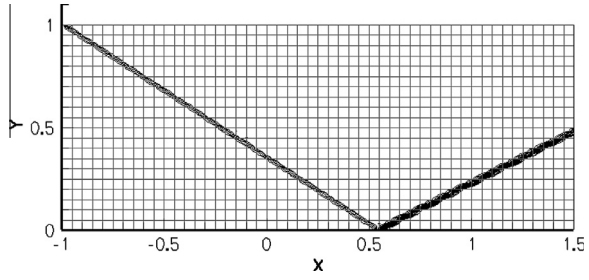

Fig. 10. Computational mesh and non-dimensional pressure contours for reflecting oblique shock test case. 20 equally spaced contours between 1 and 7.2.

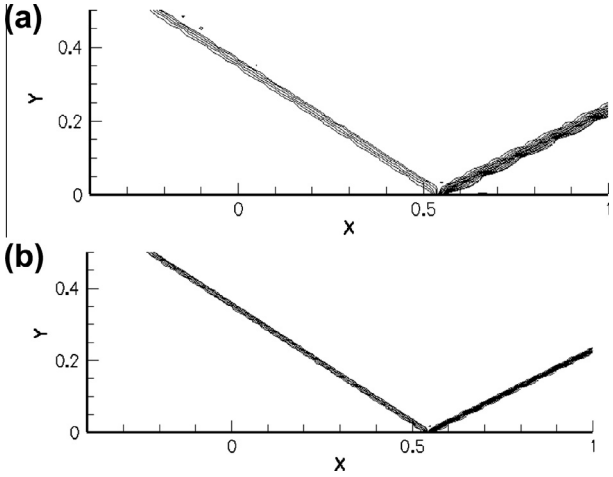

Fig. 11. Comparison of pressure contours for (a) 60×20 mesh, (b) 120×40 mesh.

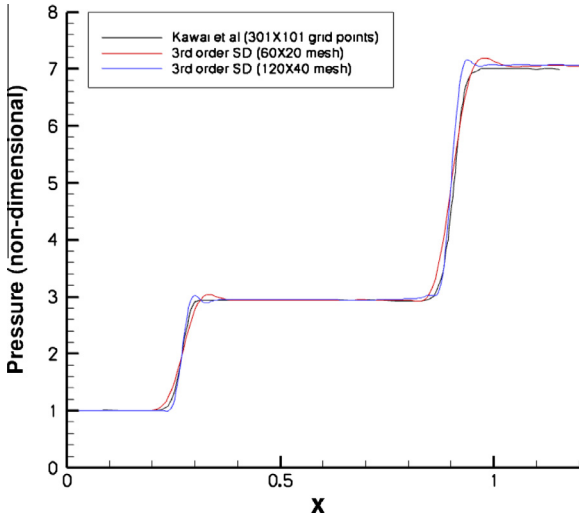

Fig. 12. Non-dimensional pressure along $y = 0.18$. Comparison with 6th order Compact Difference with artificial viscosity [3].

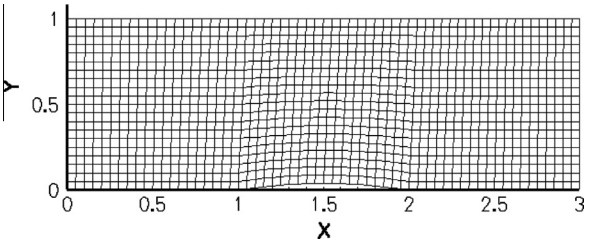

Fig. 13. Computational grid (60×20 cells) for supersonic flow past bump (thickness=4%).

Fig. 15 shows the pressure along the horizontal line corresponding to $y = 0.4$ units. It is observed that the shocks are resolved sharply, without any significant spurious oscillations. Also note that the first two shocks are captured within two cells for both the coarse and fine mesh. The last pressure jump lies close to the joining point of two oblique shocks and hence is not resolved on the coarse mesh. However, on the finer mesh, two distinct shock pressure jumps are obtained. Fig. 16 gives a plot of the sensor, which is the dilatation in this case. The dilatation is highly negative in the region of shocks. Fig. 14(c) shows the variation of artificial

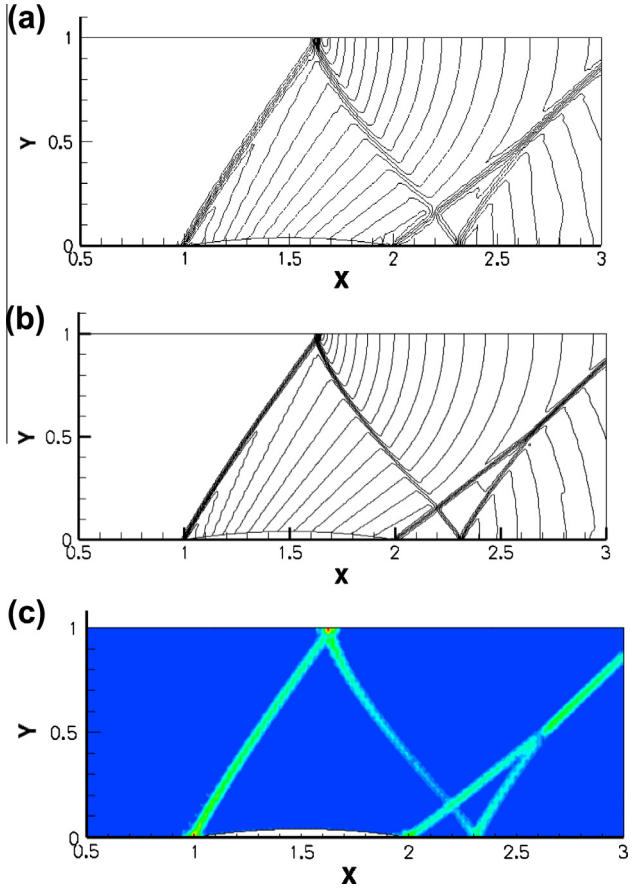

Fig. 14. 3rd order SD computation with artificial viscosity. Non-dimensional pressure contours for (a) 60×20 mesh (b) 120×40 mesh. 20 equally spaced contours from 0.6 to 2.00 (c) Artificial bulk viscosity for 120×40 mesh.

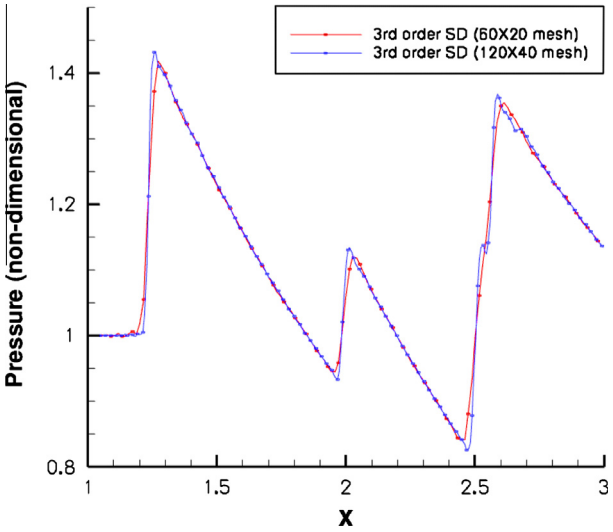

Fig. 15. Non-dimensional pressure along $y = 0.4$ line using 3rd order SD.

bulk viscosity. We see that artificial viscosity is added only in regions with sharp gradients of dilatation, corresponding to shocks.

Fig. 18(a) 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 3rd order computation. Also, 4th order on finer mesh gives sharper shock resolution and more accurate contours in comparison to the coarse mesh (see Fig. 18(b)).

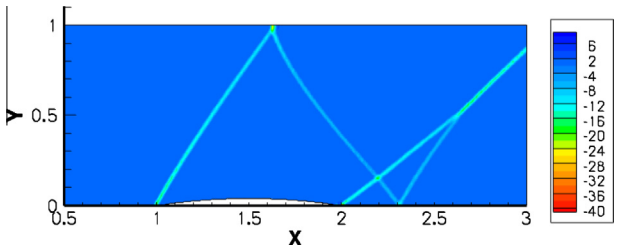

Fig. 16. Plot of the dilatation sensor for 3rd order SD computation on 120×40 mesh.

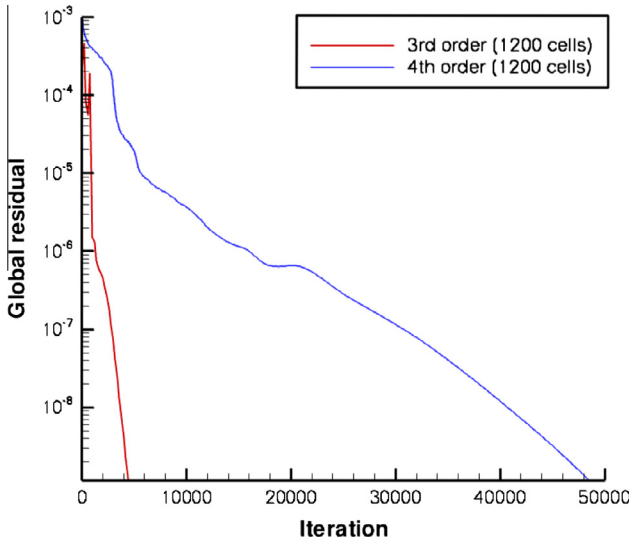

Fig. 17. Convergence plot for supersonic bump flow case using 3rd and 4th order SD with artificial viscosity.

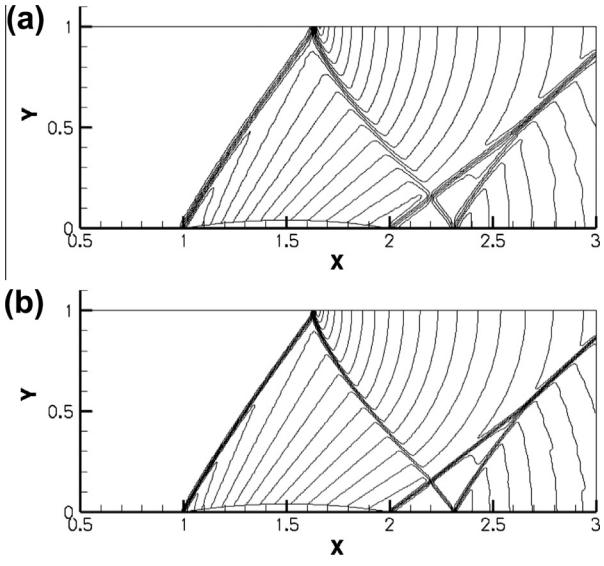

Fig. 18. 4th order SD computation with artificial viscosity. Non-dimensional pressure contours for (a) 60×20 mesh (b) 120×40 mesh. 20 equally spaced contours from 0.6 to 2.

5. Conclusions

A high wave-number biased artificial viscosity scheme has been implemented in order to enable high-order computation of flows with discontinuities on unstructured grids. 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. Further efforts will be directed towards the testing and validation of the proposed method, using a variety of shock-related problems in 2D. 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 SD scheme. The present artificial viscosity formulation can be extended to the 3D SD Scheme on hexahedral mesh elements.

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] Cook AW, Cabot WH. Hyperviscosity for shock-turbulence interactions. J Comput Phys 2005;203:379–85.
[2] Cook AW, Cabot WH. A high-wavenumber viscosity for high-resolution numerical methods. J Comput Phys 2004;195:594–601.
[3] Kawai S, Lele S. Localized artificial diffusivity scheme for discontinuity capturing on curvilinear meshes. J Comput Phys 2008;227:9498–526.
[4] Bassi F, Rebay S. A high-order discontinuous finite element method solution of the 2D Euler equations. J Comput Phys 1997;138:251–85.
[5] Luo H, Baum JD, Lohner R. A p-multigrid discontinuous Galerkin method for the Euler equations on unstructured grids. J Comput Phys 2006;211:767–83.
[6] Liu Y, Vinokur M, Wang ZJ. Spectral (Finite) volume method for conservation laws on unstructured grids V: extension to three-dimensional systems. J Comput Phys 2006;212:454–72.
[7] Wang ZJ, Liu Y. Extension of the spectral volume method to high-order boundary representation. J Comput Phys 2006;211:154–78.
[8] Liu Y, Vinokur M, Wang ZJ. Spectral difference method for unstructured grids I: basic formulation. J Comput Phys 2006;216:780–801.
[9] Wang ZJ, Liu Y, May G, Jameson A. Spectral difference method for unstructured grids II: extension to the Euler equations. J Sci Comput 2007;32:45–71.
[10] Wang ZJ, Gao H. A unifying lifting collocation penalty formulation including the discontinuous Galerkin, spectral volume/difference methods for conservation laws on mixed grids. J Comput Phys 2009;228:8161–86.
[11] Kopriva DA, Kolias JH. A conservative staggered-grid Chebyshev multidomain method for compressible flows. J Comput Phys 1996;125:244–61.
[12] Kopriva DA. A conservative staggered-grid Chebyshev multidomain method for compressible flows II: semi-structured method. J Comput Phys 1996;128:475–88.
[13] May G, Jameson A. A spectral difference method for the Euler and Navier–Stokes equations. AIAA paper 2006-304.
[14] Wang ZJ, Sun Y, Liang C, Liu Y. Extension of the SD method to viscous flow on unstructured grids. In: Proceedings of the fourth international conference on computational fluid dynamics, ICCFD, Ghent, Belgium, July 2006. p. 119–124.
[15] Sun Y, Wang ZJ, Liu Y. High-order multidomain spectral difference method for the Navier–Stokes equations on unstructured hexahedral grids. Commun Comput Phys 2007;2:310–33.
[16] Jameson A. A proof of the stability of the spectral difference method for all orders of accuracy. J Sci Comput 2010;45:348–58.
[17] Balan A, May G, Schoberl J. A stable high-order spectral difference method for hyperbolic conservation laws on triangular elements. J Comput Phys 2012;231:2359–75.
[18] von Neumann J, Richtmyer R. A method for the numerical calculation of hydrodynamic shocks. J Appl Phys 1950;21:232–7.
[19] Jameson A. Transonic potential flow calculations using conservation form. In: Proceedings of second AIAA computational fluid dynamics conference, Hartford. p. 148–61.
[20] Jameson A, Schmidt W, Turkel E. Numerical simulation of the Euler equations by finite volume methods using Runge–Kutta time stepping schemes. AIAA paper 81-125.
[21] Jameson A. Analysis and design of numerical schemes for gas dynamics, 1: artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence. Int J Comput Fluid Dynam 1994;4:171–218.
[22] Jameson A. Analysis and design of numerical schemes for gas dynamics, 2: artificial diffusion and discrete shock structure. Int J Comput Fluid Dynam 1995;5:1–38.

[23] Fiorina B, Lele S. An artificial nonlinear diffusivity method for supersonic reacting flows with shocks. J Comput Phys 2007;222:246–64.
[24] Persson PO, Peraire J. Sub-cell shock capturing for discontinuous Galerkin methods. AIAA paper 2006-112; 2006.
[25] Barter G, Darmofal D. Shock capturing with higher-order. PDE-based artificial viscosity. AIAA paper 2007-3823.
[26] Nguyen N, Peraire J. An Adaptive shock-capturing HDG method for compressible flows. AIAA Paper 2011-3060.
[27] Yang M, Wang Z. A parameter-free generalized moment limiter for high-order methods on unstructured grids. AIAA paper 2009-605.
[28] Premasuthan S, Liang C, Jameson A. A spectral difference method for viscous compressible flows with shocks. AIAA paper 2009-3785.
[29] Huynh HT. A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods. AIAA Paper 2007-4079.
[30] van den Abeele K, Lacor C, Wang ZJ. On the stability and accuracy of the spectral difference method. J Sci Comput 2008;37:162–88.
[31] Rusanov V. Calculation of interaction of non-steady shock waves with obstacles. J Comput Math Phys USSR 1961;1:267–79.
[32] Spiteri RJ, Ruuth SJ. A new class of optimal high-order strong-stability-preserving time discretization methods. SIAM J Numer Anal 2002;40:469–91.
[33] Mani A, Larsson J, Moin P. Suitability of artificial bulk viscosity for large-eddy simulation of turbulent flows with shocks. J Comput Phys 2009;228:7368–74.
[34] Blackburn HM, Schmidt S. Spectral Element filtering techniques for large eddy simulation with dynamic estimation. J Comput Phys 2003;186:610–29.
[35] Premasuthan S, Liang C, Jameson A, Wang ZJ. A p-multigrid spectral difference method for viscous compressible flow using 2D quadrilateral meshes. AIAA Paper 2009-0950.
[36] Liang C, Premasuthan S, Jameson A, Wang ZJ. Large eddy simulation of compressible turbulent channel flow with spectral difference method. AIAA Paper 2009-402.
[37] Sod GA. A survey of several finite difference methods for systems on non-linear hyperbolic conservation laws. J Comput Phys 1981;27:1–31.
[38] Shu CW, Osher SS. Efficient implementation of essentially non-oscillatory shock-capturing schemes. J Comput Phys 1988;77:439–71.
[39] Ripley R, Lien F, Yoyanivich M. Adaptive unstructured mesh refinement of supersonic channel flows. Int J Comput Fluid Dynam 2004;18:189–98.