Journal of Scientific Computing, Vol. 32, No. 1, July 2007 (© 2006)
DOI: 10.1007/s10915-006-9113-9

Spectral Difference Method for Unstructured Grids II: Extension to the Euler Equations

Z. J. Wang,[1,4] Yen Liu,[2] Georg May,[3] and Antony Jameson[3]

Received June 25, 2006; accepted (in revised form) September 20, 2006; Published online December 7, 2006

An efficient, high-order, conservative method named the spectral difference method has been developed recently for conservation laws on unstructured grids. It combines the best features of structured and unstructured grid methods to achieve high-computational efficiency and geometric flexibility; it utilizes the concept of discontinuous and high-order local representations to achieve conservation and high accuracy; and it is based on the finite-difference formulation for simplicity. The method is easy to implement since it does not involve surface or volume integrals. Universal reconstructions are obtained by distributing solution and flux points in a geometrically similar manner for simplex cells. In this paper, the method is further extended to nonlinear systems of conservation laws, the Euler equations. Accuracy studies are performed to numerically verify the order of accuracy. In order to capture both smooth feature and discontinuities, monotonicity limiters are implemented, and tested for several problems in one and two dimensions. The method is more efficient than the discontinuous Galerkin and spectral volume methods for unstructured grids.

KEY WORDS: High-order; conservation laws; unstructured grids; spectral difference; spectral collocation method; Euler equations.

1. INTRODUCTION

A new, high-order, conservative, and efficient method named the spectral difference (SD) method has been recently developed by Liu et al.

[1] Department of Aerospace Engineering, Iowa State University, 2271 Howe Hall, Ames, IA 50011, USA. E-mail: zjw@iastate.edu
[2] NASA Ames Research Center, Mail Stop T27B-1, Moffett Field, CA 94035, USA. E-mail: liu@nas.nasa.gov
[3] Department of Aeronautics and Astronautics, Stanford University, Durand Building, 496 Lomita Mall, Stanford, CA 94305-4035, USA.
[4] To whom correspondence should be addressed. E-mail: zjw@iastate.edu

[15,16] for conservation laws on unstructured grids. In the present study, the SD method is further extended to the Euler equations. The primary motivation for developing another numerical method is to seek a simpler to implement and more efficient method than the current state of the art—the discontinuous Galerkin (DG) method [2,5,6], and the spectral volume (SV) method [17,27–30], to name just a few high-order methods for conservation laws on unstructured grids. As a matter of fact, the DG, SV, and SD methods are similar in that they share the same the solution space, i.e., the space of piece-wise discontinuous polynomials, and some Riemann solvers [11,18,19] are used at the element interfaces to provide solution coupling between the discontinuous elements and appropriate numerical dissipation necessary to achieve stability. In addition, all of them are conservative locally at the element level, making them suitable for problems with discontinuities. They do differ on how solution unknowns or degrees-of-freedom (DOFs) are chosen, and how the DOFs are updated. In a DG method, the DOFs are either the expansion coefficients for a given set of polynomial basis functions or solutions at selected locations within the element. In a SV method, however, the DOFs are subcell averaged solutions, while in the SD method, the DOFs are the solutions at (usually) the Gauss quadrature points. The difference between the DG, SV, and SD methods is similar to the difference between the Galerkin finite element (FE), finite volume (FV), and finite difference (FD) methods.

In the DG method, a Galerkin finite-element method is employed to update the unknowns within each cell. This requires (usually) the inversion of a mass matrix, and the use of quadratures of roughly twice the order of accuracy of the reconstruction to evaluate the surface integrals for non-linear flux functions and additional volume integrals. In the SV method, the integral conservation law is used to update volume averages over subcells defined by a geometrically similar partition of each grid cell. As the order of accuracy increases, the partitioning for a 3D simplex cell requires the introduction of a large number of parameters, whose optimization to achieve convergence becomes increasingly more difficult. Also, the large number of interior facets, and the additional increase in the number of quadrature points for each facet increase the computational cost significantly. Because there are no volume or surface integrals in the SD method, it is easier to implement in multiple dimensions than the DG and SV methods. For the same reason, it will be shown later that the SD method is indeed more efficient than the DG and SV methods.

In the SD method, the number of DOFs in each cell is the number of nodal values required to support a reconstruction of a given order of accuracy. Their locations are chosen so that a quadrature

approximation for the volume integral exists at least to the same order of accuracy. The fluxes are calculated at a different set of nodes, whose number supports a reconstruction of one order higher, since the flux derivatives are used to update the conservative unknowns. They are located so that quadrature approximations for surface integrals over the cell boundaries exist to a required order of accuracy. In addition, the locations of the solution points and the flux points must be such that the integral conservation law is satisfied for the cell to the desired order of accuracy. If the points are distributed in a geometrically similar manner for all cells, the reconstruction and discretization become universal, and can be expressed as the same weighted sums of the products of the local metrics and fluxes. These metrics are constants for the line, triangle, and tetrahedron elements, and can be computed analytically for curved elements. We can also show that the number of flux points is less than the number of quadrature points in the SV method. Since all unknowns are decoupled, no mass matrix inversion is required.

Conventional unstructured FD [24] and FV [1,7] methods require data reconstructions based on the least-squares formulation using neighboring point or cell data. Since each unknown employs a different stencil, one must repeat the least-squares inversion for every point or cell at each time step, or store the inversion coefficients. In a high-order, 3D computation, the former would involve impractically large CPU time, while for the latter the memory requirement becomes prohibitive. In addition, the FD method does not usually satisfy the integral conservation in general. In contrast, the DG, SV, and SD methods employ a local, universal reconstruction of a given order of accuracy in each cell in terms of internally defined conservative unknowns. This is the main reason why high-order DG, SV, and SD methods are more efficient than a high-order finite volume method.

The SD formulation is similar to the pseudo-spectral or collocation spectral method [3] in that both employ nodal solutions as the DOFs and both formulations are based on the differential form of the governing equations. In fact, the multi-domain spectral method developed by Kopriva [12,13] and the SD method degenerate to a similar method in one dimension. The SD method can be viewed as an extension of the multi-domain spectral method to a simplex unstructured grid.

The paper is organized as follows. In the next section, the basic idea of the SD method is presented in the physical domain. Its efficient implementation and conservation property are discussed in Sec. 3. Discontinuity capturing and data limiters are discussed in Sec. 4. Sample numerical results including a numerical accuracy study are presented in Sec. 5. Conclusions and possible future work are outlined in Sec. 6.

2. BASIC IDEA OF THE SPECTRAL DIFFERENCE METHOD

Consider the unsteady 2D Euler equations in conservative form written as

$$\frac{\partial Q}{\partial t}+\frac{\partial f}{\partial x}+\frac{\partial g}{\partial y}=0, \tag{1a}$$

where Q is the vector of conserved variables, f and g are the inviscid fluxes given below:

$$Q=\begin{Bmatrix}\rho\\ \rho u\\ \rho v\\ E\end{Bmatrix}, \qquad f=\begin{Bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ u(E+p)\end{Bmatrix}, \qquad g=\begin{Bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ v(E+p)\end{Bmatrix}, \tag{1b}$$

where ρ is the density, u and v the velocity components in x and y-directions, p the pressure, and E is the total energy. The pressure is related to the total energy by

$$E=\frac{p}{\gamma-1}+\frac{1}{2}\rho(u^2+v^2) \tag{1c}$$

with a constant ratio of specific heats $\gamma=1.4$ for air. Define a flux vector with two components, i.e., $F=(f,g)$. Equation (1a) can be expressed in the following divergence form

$$\frac{\partial Q}{\partial t}+\nabla\bullet F=0. \tag{1d}$$

Equation (1) is to be solved on a non-overlapping simplex grid with proper initial and boundary conditions. Within each cell or element, we define two different sets of grid points, i.e., the *solution points* and *flux points*. The solution points are the locations where the nodal values of the conservative variables Q are specified (usually Gauss quadrature points). Flux points are the locations where the nodal values of fluxes F are computed. The solution unknowns or degrees of freedom in the SD method are the conservative variables at the solution points. Figure 1 displays the placements of solution and flux points for the first to third-order SD schemes [16]. These nodal sets were computed using Mathematica [32] following ideas presented in [14], and the Lebegue constant is found to be comparable to that of other nodal sets published in the literature [4, 10] (if any reader is interested in obtaining the nodal sets, please e-mail the authors). Let the position vector of the jth solution point at cell i be

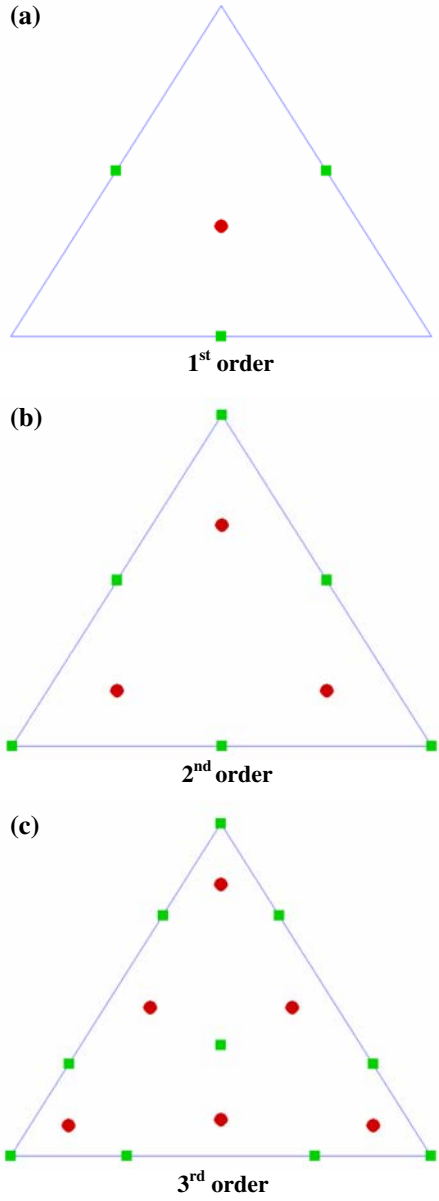


Fig. 1. Placement of solution (●) and flux (■) points for a triangular element.

denoted by $\mathbf{r}_{j,i}$, and the kth flux point at cell i be denoted by $\mathbf{r}_{k,i}$. Denote $Q_{j,i}$ the solution at $\mathbf{r}_{j,i}$. Given the solutions at $\mathbf{r}_{j,i}$, an element-wise degree p polynomial can be constructed using Lagrange-type polynomial basis, i.e.,

$$Q_i(\mathbf{r}) = \sum_{j=1}^{N_p} L_{j,i}(\mathbf{r}) Q_{j,i}, \tag{2}$$

where $L_{j,i}(\mathbf{r})$ are the cardinal basis functions and N_p is the number of basis functions required to support a degree p polynomial reconstruction, and can be easily derived with Mathematica. Obviously, the locations of the solution $\mathbf{r}_{j,i}$ uniquely determine the cardinal basis functions $L_{j,i}(\mathbf{r})$. With the polynomial distribution given in (2), the solutions of Q at the flux points $\mathbf{r}_{k,i}$ can be computed easily from

$$Q(\mathbf{r}_{k,i}) = \sum_{j=1}^{N_p} L_{j,i}(\mathbf{r}_{k,i}) Q_{j,i}. \tag{3}$$

Since the solutions are element-wise polynomials, they are discontinuous across element boundaries. As a result, the fluxes at the element interfaces are not uniquely defined, for example, at the corner and face points shown in Fig. 2.

At the corner point, five solutions exist from all the cells (*A, B, C, D*, and *E*) sharing the point. At the face point between cells *C* and *D*,

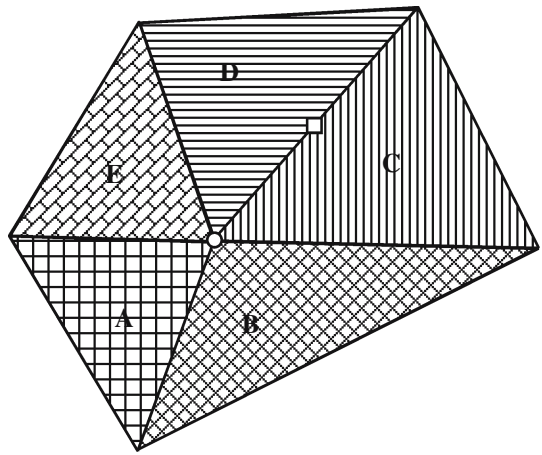


Fig. 2. Illustration of multi-dimensional Riemann problems at the corner and face point.

two solutions exist. The naïve approach is to compute an averaged solution from these multiple solutions and then determine the flux based on the averaged solution. However, it is well known that this naïve approach is equivalent to central differencing and is not stable. An alternative approach is to find the physical solution Q at the corner and face points at time $t = 0^+$ with the discontinuous solutions as the initial condition for the Euler equations at $t = 0$. This idea is of course due to Godunov [8], who pioneered the well-known Godunov-type finite volume methods, which become the standard method for conservation laws [25]. Unfortunately, this "multi-dimensional Riemann problem" shown in Fig. 2 is very difficult to solve, either analytically or numerically. Simpler approximate Riemann solvers must be found to determine these fluxes. We again turn to the FV method to look for inspirations. Obviously, in order to ensure conservation, the normal component of the flux vector on each face should be identical for the two cells sharing the face. Physically this means a mass flux going out of a cell must completely enter the neighboring cell without mass generation or loss. To ensure conservation, a one dimensional Riemann solver is employed in the face normal direction to compute the common normal flux. Consider the face flux point shown in Fig. 3, and denote the outgoing normal from cell C to cell 1 $\boldsymbol{n}_1$. For this interface point, Q_L is computed from cell C and Q_{R} is computed from cell 1. Then the common normal component of the flux can be computed with any Riemann solvers such as the Rusanov [19] or Roe [18] flux. In the case of the simpler Rusanov flux, the normal component is computed from

$$F_{\mathrm{n}} = F_{\mathrm{n}}(Q_{\mathrm{L}}, Q_{\mathrm{R}}, \mathbf{n}) = \frac{1}{2}\{[F(Q_{\mathrm{L}}) + F(Q_{\mathrm{R}})] \bullet \mathbf{n} - (\bar{v}_{\mathrm{n}} + \bar{c})(Q_{\mathrm{R}} - Q_{\mathrm{L}})\}, \quad (4)$$

where $\bar{v}_{\mathrm{n}}$ is the average normal velocity and $\bar{c}$ is the average speed of sound computed from the left and right solutions. Since the tangential component of the flux does not affect the conservation property, we have the complete freedom in determining it at the face point. In fact, it is not strictly necessary to have a unique tangential component physically at the face point (e.g. think of a contact discontinuity in which density is discontinuous). Let the unit vector in the tangential direction be $\boldsymbol{l}$. Here we offer two possibilities. One is to use a unique tangential component by averaging the two tangential components from both sides of the face, i.e.,

$$F_{\mathrm{l}} = F_{\mathrm{l}}(Q_{\mathrm{L}}, Q_{\mathrm{R}}, \mathbf{l}) = \frac{1}{2}\{[F(Q_{\mathrm{L}}) + F(Q_R)] \bullet \mathbf{l}\}. \quad (5)$$

The other possibility is to use its own tangential component from the current cell, allowing the tangential component to be discontinuous.

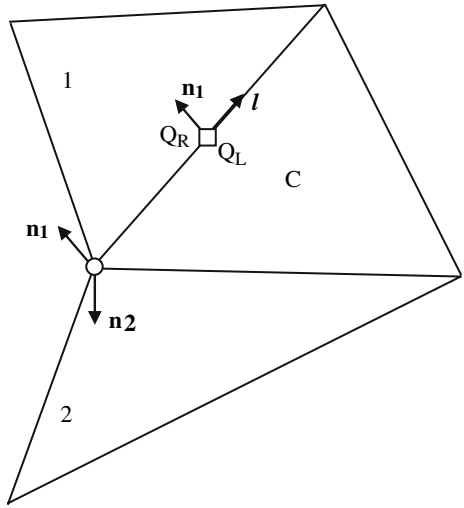


Fig. 3. Flux computation for a corner (○) and a face (□) point using one-dimensional Riemann solvers.

Therefore, the tangential component of the flux on either side is not modified. For the left cell, the tangential and normal components are $(F(Q_L) \bullet \mathbf{l}, F_n)$, and for the right cell, they become $(F(Q_R) \bullet \mathbf{l}, F_n)$.

For a corner flux point in cell C, two faces (from cell C) share the corner point, as shown in Fig. 3. Let the unit normals of the two faces be $\mathbf{n}_1$ and $\mathbf{n}_2$. Once again, the normal components of flux F_{n1} and F_{n2} in $\mathbf{n}_1$ and $\mathbf{n}_2$ directions are computed with a 1D Riemann solver in the normal directions. The full flux vector can then be uniquely determined from the two normal flux components

$$F \bullet \mathbf{n}_1 = F_{n1}, \tag{6a}$$

$$F \bullet \mathbf{n}_2 = F_{n2}. \tag{6b}$$

It is important to emphasize here that fluxes at cell corner points do not have unique values for all the cells sharing the corner. In spite of that, local conservation is guaranteed because neighboring cells do share a common normal flux at all the flux points. Once the fluxes at all the flux points are re-computed, they are used to form a degree $p+1$ polynomial, i.e.,

$$F_i(\mathbf{r}) = \sum_{k=1}^{N_{p+1}} M_{k,i}(\mathbf{r}) F_{k,i}, \tag{7}$$

where $M_{k,i}(\mathbf{r})$ are the set of cardinal basis functions defined by $\mathbf{r}_{k,i}$ and $F_{k,i} = F(\mathbf{r}_{k,i})$. Obviously, the divergence of the flux at any point within the cell can be computed using

$$\nabla \bullet F_i(\mathbf{r}) = \sum_{k=1}^{N_{p+1}} \nabla M_{k,i}(\mathbf{r}) \bullet F_{k,i}. \tag{8}$$

To update the solutions at the solution points $\mathbf{r}_{j,i}$, we need to evaluate the divergence at these points, which can be easily computed according to

$$\nabla \bullet F_i(\mathbf{r}_{j,i}) = \sum_{k=1}^{N_{p+1}} \nabla M_{k,i}(\mathbf{r}_{j,i}) \bullet F_{k,i}. \tag{9}$$

Finally the semi-discrete scheme to update the solution unknowns can be written as

$$\frac{dQ_{j,i}}{dt} + \sum_{k=1}^{N_{p+1}} \nabla M_{k,i}(\mathbf{r}_{j,i}) \bullet F_{k,i} = 0. \tag{10}$$

For time integration, high-order TVD (or SSP) Runge–Kutta schemes [21,23] are employed.

3. EFFICIENT IMPLEMENTATION AND CONSERVATION PROPERTY

In the last section, we avoided the implementation and conservation issues in order to focus on presenting the basic idea of the SD method. The reconstruction formulas for the solution and flux presented in (2), (7) may give the readers the impression that each cell has a different set of reconstruction coefficients. It will be shown that for triangles with straight edges (most of the cells except curved wall boundary cells), the reconstruction coefficients are universal for all triangles. This is true because any triangle can be transformed to a standard triangle as shown in Fig. 4 through the following linear transformation:

$$\mathbf{r} = \mathbf{r}_{0,i} + \xi(\mathbf{r}_{1,i} - \mathbf{r}_{0,i}) + \eta(\mathbf{r}_{2,i} - \mathbf{r}_{0,i}), \quad 0 \leqslant \xi, \quad \eta \leqslant 1 \text{ and } \quad \xi + \eta \leqslant 1, \tag{11}$$

where $\mathbf{r}_{0,i}$, $\mathbf{r}_{1,i}$, and $\mathbf{r}_{2,i}$ are the three vertices of cell i. If the solution points $\mathbf{r}_{j,i}$ and flux points $\mathbf{r}_{k,i}$ are distributed in a geometrically similar manner for all cells, they all have the same local position $\boldsymbol{\xi}_j$ and $\boldsymbol{\xi}_k$

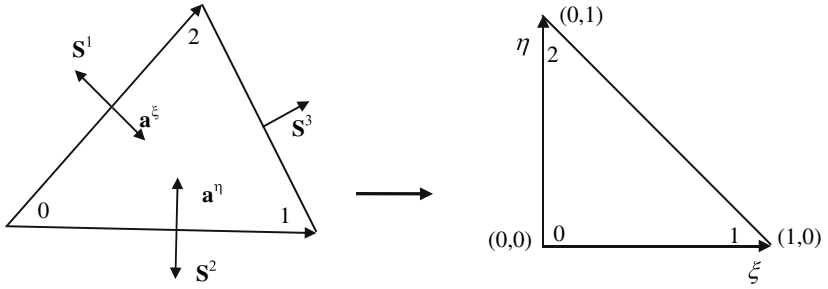


Fig. 4. Transformation from a physical element to the standard element.

with $\boldsymbol{\xi} = (\xi, \eta)$. It is shown in [16] that a universal reconstruction can be written as

$$Q_i(\boldsymbol{\xi}) = \sum_{j=1}^{N_p} L_j(\boldsymbol{\xi})\, Q_{j,i}, \tag{12}$$

where the cardinal basis functions $L_j(\boldsymbol{\xi})$ are universal for all triangles. Therefore the solution at the flux points can be computed using

$$Q_{k,i} \equiv Q_i(\boldsymbol{\xi}_k) = \sum_{j=1}^{N_p} l_{kj}\, Q_{j,i}, \tag{13}$$

where $l_{kj} = L_j(\boldsymbol{\xi}_k)$. Similarly the reconstruction polynomial for the flux can be written as

$$F_i(\boldsymbol{\xi}) = \sum_{k=1}^{N_{p+1}} M_k(\boldsymbol{\xi})\, F_{k,i}, \tag{14}$$

where $M_k(\boldsymbol{\xi})$ are the universal cardinal basis functions based on the flux points. It is then straightforward to show that the gradient of F on the computational domain (standard element) takes the following universal form for all cells

$$\nabla F_i(\boldsymbol{\xi}) = \sum_{k=1}^{N_{p+1}} \nabla M_k(\boldsymbol{\xi})\, F_{k,i}. \tag{15}$$

From (15), the gradients of the flux at the solution points can be computed according to

$$\frac{\partial F_i(\boldsymbol{\xi}_j)}{\partial \xi} = \sum_{k=1}^{N_{p+1}} m_{jk,\xi} F_{k,i}, \tag{16}$$

$$\frac{\partial F_i(\boldsymbol{\xi}_j)}{\partial \eta} = \sum_{k=1}^{N_{p+1}} m_{jk,\eta} F_{k,i}. \tag{17}$$

where $m_{jk,\xi} = \frac{\partial M_k(\boldsymbol{\xi}_j)}{\partial \xi}$ and $m_{jk,\eta} = \frac{\partial M_k(\boldsymbol{\xi}_j)}{\partial \eta}$. By applying the chain rule, we can easily relate the divergence of F in the physical domain to the gradients of the flux in the computational domain

$$\begin{aligned}\nabla \bullet F_i(\mathbf{r}) = \frac{\partial f_i(\mathbf{r})}{\partial x} + \frac{\partial g_i(\mathbf{r})}{\partial y} &= \frac{\partial f_i(\boldsymbol{\xi})}{\partial \xi}\xi_{x,i} + \frac{\partial f_i(\boldsymbol{\xi})}{\partial \eta}\eta_{x,i} + \frac{\partial g_i(\boldsymbol{\xi})}{\partial \xi}\xi_{y,i} \\ &+ \frac{\partial g_i(\boldsymbol{\xi})}{\partial \eta}\eta_{y,i}.\end{aligned} \tag{18}$$

For the linear transformation given in (11), it is easy to show that

$$\begin{bmatrix} \xi_{x,i} & \xi_{y,i} \\ \eta_{x,i} & \eta_{y,i} \end{bmatrix} = \frac{1}{2V_i}\begin{bmatrix} y_{2,i} - y_{0,i} & -x_{2,i} + x_{0,i} \\ -y_{1,i} + y_{0,i} & x_{1,i} - x_{0,i} \end{bmatrix}, \tag{19}$$

where V_i is the volume of cell i. Let $\mathbf{a}^{\xi}$ and $\mathbf{a}^{\eta}$ be the inward-pointing area vectors of face 02 and face 01 in Fig. 4. Obviously, we have

$$(\xi_{x,i}, \xi_{y,i}) = \frac{\mathbf{a}_i^{\xi}}{2V_i}, \quad (\eta_{x,i}, \eta_{y,i}) = \frac{\mathbf{a}_i^{\eta}}{2V_i}. \tag{20}$$

Equation (18) can be more concisely written as

$$\nabla \bullet F_i(\mathbf{r}) = \frac{1}{2V_i}\left[\frac{\partial F_i(\boldsymbol{\xi})}{\partial \xi} \bullet \mathbf{a}_i^{\xi} + \frac{\partial F_i(\boldsymbol{\xi})}{\partial \eta} \bullet \mathbf{a}_i^{\eta}\right]. \tag{21}$$

Finally (10) becomes

$$\begin{aligned}&\frac{dQ_{j,i}}{dt} + \frac{1}{2V_i}\left[\frac{\partial F_i(\boldsymbol{\xi}_j)}{\partial \xi} \bullet \mathbf{a}_i^{\xi} + \frac{\partial F_i(\boldsymbol{\xi}_j)}{\partial \eta} \bullet \mathbf{a}_i^{\eta}\right] \\ &= \frac{dQ_{j,i}}{dt} + \frac{1}{2V_i}\left[\mathbf{a}_i^{\xi} \bullet \sum_{k=1}^{N_{p+1}} m_{jk,\xi} F_{k,i} + \mathbf{a}_i^{\eta} \bullet \sum_{k=1}^{N_{p+1}} m_{jk,\eta} F_{k,i}\right] = 0.\end{aligned} \tag{22}$$

Note that for all triangles (with straight faces), only the two vectors $\frac{\mathbf{a}_i^\xi}{2V_i}$, $\frac{\mathbf{a}_i^\eta}{2V_i}$ need to be stored. The computation of the flux gradients on the computational domain is universal for all triangles. Obviously this formulation is much simpler than that of the DG and SV methods, and is much easier to implement too.

In order to prove conservation, we need to show that the integral form of (1) is satisfied in each cell, i.e.,

$$\int_{V_i} \left(\frac{\partial Q}{\partial t} + \nabla \bullet F \right) dV = \frac{d}{dt} \int_{V_i} Q \, dV + \oint_{\partial V_i} F \bullet \mathbf{n} \, dS = 0. \tag{23}$$

The volume and surface integrals are computed using quadrature formulas based on the solutions at the solution points and fluxes at the flux points

$$\int_{V_i} Q \, dV = V_i \sum_{j=1}^{N_p} w_j Q_{j,i}, \tag{24}$$

$$\oint_{\partial V_i} F \bullet \mathbf{n} \, dS = \sum_{l=1}^{3} \left(\mathbf{n}_l \bullet \int_{S_i^l} F \, dS \right) = \sum_{l=1}^{3} \left(\mathbf{S}_i^l \bullet \sum_{k=1}^{N_{p+1}} w_k^l F_{k,i} \right), \tag{25}$$

where w_j is the volume integral quadrature weights, and w_k^l are the surface integral quadrature weights for face l, and $\mathbf{S}_i^l$ is the outward area vector of face l. Using the fact that the area vectors form a closed surface, $\sum_{l=1}^{3} \mathbf{S}_i^l = 0$, or $\mathbf{S}_i^3 = -\mathbf{S}_i^1 - \mathbf{S}_i^2$, and $\mathbf{S}_i^1 = -\mathbf{a}^\xi$, $\mathbf{S}_i^2 = -\mathbf{a}^\eta$, (25) can be further written as

$$\oint_{\partial V_i} F \bullet \mathbf{n} \, dS = -\left(\mathbf{a}_i^\xi \bullet \sum_{k=1}^{N_{p+1}} w_k^\xi F_{k,i} + \mathbf{a}_i^\eta \bullet \sum_{k=1}^{N_{p+1}} w_k^\eta F_{k,i} \right). \tag{26}$$

where $w_k^\xi = w_k^1 - w_k^3$, and $w_k^\eta = w_k^2 - w_k^3$. Obviously, only flux points at element interfaces are used in the surface integral. Therefore, for all interior flux points, the weights should be zero. Substituting (22) into (24), we obtain

$$\frac{d}{dt}\int_{V_i} Q\,dV = V_i \sum_{j=1}^{N_p} w_j \frac{dQ_{j,i}}{dt} = -\frac{1}{2}\sum_{j=1}^{N_p} w_j \sum_{k=1}^{N_{p+1}} F_{k,i} \bullet (m_{jk,\xi}\,\mathbf{a}_i^{\xi} + m_{jk,\eta}\mathbf{a}_i^{\eta})$$
$$= -\left(\mathbf{a}_i^{\xi} \bullet \frac{1}{2}\sum_{k=1}^{N_{p+1}}\sum_{j=1}^{N_p} w_j m_{jk,\xi} F_{k,i} + \mathbf{a}_i^{\eta} \bullet \frac{1}{2}\sum_{k=1}^{N_{p+1}}\sum_{j=1}^{N_p} w_j m_{jk,\eta} F_{k,i}\right). \tag{27}$$

Comparing (26) and (27), the conservation conditions are

$$w_k^{\xi} = -\frac{1}{2}\sum_{j=1}^{N_p} w_j m_{jk,\xi} \quad \text{and} \quad w_k^{\eta} = -\frac{1}{2}\sum_{j=1}^{N_p} w_j m_{jk,\eta}. \tag{28}$$

Since these equations depend only on the locations of the solution and quadrature points, they can be satisfied by properly placing the solution and flux points. In fact, these conditions are satisfied for all the placements shown in Fig. 1.

4. DISCONTINUITY CAPTURING AND DATA LIMITING

Hyperbolic conservation laws admit solutions that develop discontinuities in finite time, even if the initial data is smooth. Enabling the computation of numerical solutions without spurious oscillations at discontinuities is essential in the construction of high-order schemes for these types of equations. It is well known that monotone linear schemes are at most first order accurate. A monotone higher order scheme must necessarily be nonlinear, which usually takes the form of limiter functions that depend on the current solution. For many high-order schemes, such as the DG method [6] and the SV method [28–30] limiters based on the minmod function have proved quite successful, and have led to convergence proofs in the total-variation-diminishing (TVD) [9] or total-variation-bounded (TVB) [20] framework. We use a similar general approach, which is outlined in the following for one and two space dimensions. Consider the reconstructed solution $Q_i(x)$ inside a cell i, denote its volume average by $\overline{Q_i}$, and define

$$\begin{aligned} \delta Q_{i+1/2} &= Q(x_{i+1/2}) - \overline{Q_i}, \\ \delta Q_{i-1/2} &= \overline{Q_i} - Q(x_{i-1/2}), \\ \overline{\Delta Q_{i+1/2}} &= \overline{Q_{i+1}} - \overline{Q_i}, \end{aligned} \tag{29}$$

where the cell interfaces are denoted by subscripts $i+1/2$ and $i-1/2$. In order to compute the fluxes at these interfaces one may construct the limited states as

$$\begin{aligned} Q_{i+1/2} &= \overline{Q_i} + \text{minmod}\left(\delta Q_{i+1/2}, \overline{\Delta Q_{i+1/2}}, \overline{\Delta Q_{i-1/2}}\right), \\ Q_{i-1/2} &= \overline{Q_i} - \text{minmod}\left(\delta Q_{i-1/2}, \overline{\Delta Q_{i+1/2}}, \overline{\Delta Q_{i-1/2}}\right), \end{aligned} \tag{30}$$

where the minmod function is given by

$$\text{min mod}\,(y_1, y_2, y_3) = \begin{cases} s \cdot \min(|y_1|, |y_2|, |y_3|), & \text{if } \operatorname{sign}(y_1) = \operatorname{sign}(y_2) \\ & = \operatorname{sign}(y_3) = s, \\ 0, & \text{otherwise.} \end{cases} \tag{31}$$

If the limiter is active, the remaining reconstruction can be restricted to a lower order approximation, requiring consistency with the volume average and the limited left and right states, which can be accomplished by a second order polynomial, or a linear slope, if only the more restrictive state at the boundary is matched by the new reconstruction. A rigorous TVD theory exists for numerous schemes combined with this limiting methodology. To avoid reduction of the accuracy at smooth extrema, inherent to the TVD limiting procedure, the minmod function is modified to be inactive at smooth extrema. This leads to TVB schemes, which retain the nominal accuracy for all smooth solutions:

$$m\,(y_1, y_2, y_3) = \begin{cases} y_1, & if\ y_1 < Mh^2, \\ \text{min mod}\,(y_1, y_2, y_3), & \text{otherwise.} \end{cases} \tag{32}$$

We deviate slightly from the standard approach in order to exploit the resolution available by the nodal data representation within each cell. If we wish to compute the flux at flux node k, we limit the differences

$$\begin{aligned} \delta Q_{k,l} &= Q(x_{f,k}) - Q(x_{u,j}), \\ \delta Q_{k,r} &= Q(x_{u,j+1}) - Q(x_{f,k}), \end{aligned} \tag{33}$$

where $Q(x_{u,j})$ and $Q(x_{u,j+1})$ are the neighboring solution collocation nodes to the left and right, respectively. Note that the index j is a generic global index for the solution nodes. For interior flux nodes k, the left and right neighbors will be solution nodes in the same cell, otherwise the next neighbors in the adjacent cells are chosen. We have

$$\begin{aligned} Q_{k,l} &= Q(x_{u,j}) + \text{m}\left(\delta Q_{k,l}, \Delta Q_{j+1}, \Delta Q_j\right), \\ Q_{k,r} &= Q(x_{u,j+1}) - \text{m}\left(\delta Q_{k,r}, \Delta Q_{j+2}, \Delta Q_{j+1}\right), \end{aligned} \tag{34}$$

where now $\Delta Q_j = Q(x_{u,j}) - Q(x_{u,j-1})$. This approach has certain advantages. Most importantly we have sub-cell resolution in the sense that it is possible to limit the reconstruction at each flux node individually as opposed to using a limited state at cell boundaries only, which necessitates the explicit reduction of the order of reconstruction for some of the cells. Dissipation can be introduced at interior nodes in case of very steep gradients or discontinuities inside the cell, while the cell boundaries may be smooth. A disadvantage is that there is no TVD or TVB theory available at this point for this approach.

The 2D TVD limiter follows a similar idea. The limiter is implemented in the following steps:

1. Compute the cell averaged state variables for each cell $\{\bar{Q}_i\}$.
2. Compute local minimum and maximum mean variables for cell i, $\bar{Q}_{i,\min}$, and $\bar{Q}_{i,\max}$ using a local stencil of cells sharing nodes.
3. Compute Q at the flux points $Q_{k,i}$.
4. If any of the solution $Q_{k,i}$ is outside $(\bar{Q}_{i,\min}, \bar{Q}_{i,\max})$, cell i is limited. Q is assumed linear and takes the following form

$$Q_i(\mathbf{r}) = \bar{Q}_i + \phi \nabla Q_i \bullet (\mathbf{r} - \mathbf{r}_i), \tag{35}$$

where ∇Q_i is the original gradient computed at the cell centroid, and $\mathbf{r}_i$ is the location of the cell centroid and ϕ isa limiter function in [0, 1] which makes the solution bounded by $[\bar{Q}_{i,\min}, \bar{Q}_{i,\max}]$, similar to that in [16].

5. NUMERICAL RESULTS

5.1. Quasi-1D Nozzle Flow

The quasi-1D Euler equations can be written

$$\frac{\partial Q}{\partial t} + \frac{\partial f}{\partial x} + S = 0,$$

where

$$Q = \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}, \qquad f = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(E+p) \end{pmatrix}, \qquad S = \frac{1}{A}\frac{dA}{dx}\begin{pmatrix} \rho u \\ \rho u^2 \\ u(E+p) \end{pmatrix}$$

and A is the cross-section area. For all the test cases considered below the flow conditions at the inflow and outflow are subsonic (for transonic cases the flow is decelerated by a normal shock before leaving the computational domain). At the inflow we impose boundary conditions by extrapolating

Table I. Isentropic Nozzle Flow: Accuracy in Mesh Refinement for the Third-Order Scheme with CUSP Flux

No. of Cells	L_∞	Order L_∞	L_2	Order L_2
6	$1.13E-05$	–	$6.53E-06$	–
10	$2.90E-06$	2.66	$1.48E-06$	2.91
16	$7.08E-07$	3.00	$3.47E-07$	3.08
24	$2.02E-07$	3.09	$9.62E-08$	3.16
30	$1.02E-07$	3.07	$4.75E-08$	3.16
40	$4.19E-08$	3.09	$1.92E-08$	3.14
50	$2.11E-08$	3.08	$9.60E-09$	3.12
60	$1.20E-08$	3.08	$5.46E-09$	3.10

the outgoing Riemann invariant and fixing the total enthalpy and entropy, while at the outflow the entropy and total enthalpy are extrapolated and the pressure is fixed. All computations use the CUSP flux [11] at cell interfaces.

A.1 Smooth Flow

We first consider smooth subsonic flow in a symmetric nozzle, where $A(x)$ is defined on the interval [0,1] by the relation

$$A(x) = 1 - (1 - d_t)\frac{\sin(\xi) + 1}{2}$$

and $\xi = \pi/2(4x - 1)$. The nozzle throat area is given by $d_t = 0.875$. A nozzle exit Mach number of $M = 0.3$ is chosen, which leads to entirely subsonic, isentropic flow, so that one may choose the steady-state entropy as a measure of error. Tables I and II show the results of a mesh refinement study using the third and fourth order schemes. It can be seen that the entropy error decreases at the nominal rate in both L_2 and L_∞ norms.

A.2 Transonic Shocked Flow

We now turn to an asymmetric nozzle configuration with the aim of computing transonic flow with a shock in the diverging part of the nozzle. We define the area on the interval $[0, x_{\text{out}} = 10]$ by the relation

$$A(x) = \begin{cases} d_{in} + \xi(x)^2(3 - 2\xi(x))(d_t - d_{\text{in}}), & x \leqslant x_t, \\ d_t + \eta(x)^2(3 - 2\eta(x))(d_{\text{out}} - d_t), & x > x_t, \end{cases}$$

Table II. Isentropic Nozzle Flow: Accuracy in Mesh Refinement for the Fourth-Order Scheme with CUSP Flux

No.of Cells	L_∞	Order L_∞	L_2	Order L_2
6	$1.55E-06$	–	$7.97E-07$	–
10	$1.71E-07$	4.32	$8.21E-08$	4.45
16	$2.56E-08$	4.03	$1.20E-08$	4.09
24	$5.14E-09$	3.96	$2.36E-09$	4.02
30	$2.09E-09$	4.03	$9.65E-10$	4.00
40	$7.84E-10$	3.40	$3.11E-10$	3.94

where $\xi(x) = x/x_t$, and $\eta(x) = (x - x_t)/(x_{\text{out}} - x_t)$. The coordinate $x_t = 3.75$ is the location of the nozzle throat, while the nozzle area at the inlet, outlet and throat are given by $d_{\text{in}} = 1$, $d_{\text{out}} = 1.25$, and $d_t = 0.875$, respectively.

Figure 5 shows the solution in terms of the density for the exit Mach number $M = 0.6$, using the third-order Spectral Difference scheme. The figure shows a shock standing in the diverging part of the nozzle, and clearly demonstrates the excellent shock capturing capabilities of the scheme with the TVB limiter. We also consider the entropy distribution for this test case. The entropy behind the shock is different from that ahead of the shock, caused by the shock jump conditions. Since the shock position cannot be captured exactly, there is a first-order error proportional to the mesh size. Therefore the entropy error behind the shock is not expected to decrease with order-refinement (or p-refinement). The results for schemes of order 3 through 6, shown in Fig. 6, confirm that good shock capturing is combined with high accuracy in smooth regions. A mesh of 40 spectral cells has been used for this study.

A.3 Polynomial Refinement Study

The natural limit of the SD scheme, when operating on only one cell in one dimension, is a nodal spectral method on a staggered grid (there are no cell interfaces in this case). Since the scheme is primarily intended for use on higher-dimensional unstructured meshes, using local approximations of medium order, we shall not explore this particular limit here. However, we demonstrate in this section that spectral convergence can be obtained for a fixed mesh and increasing local polynomial accuracy.

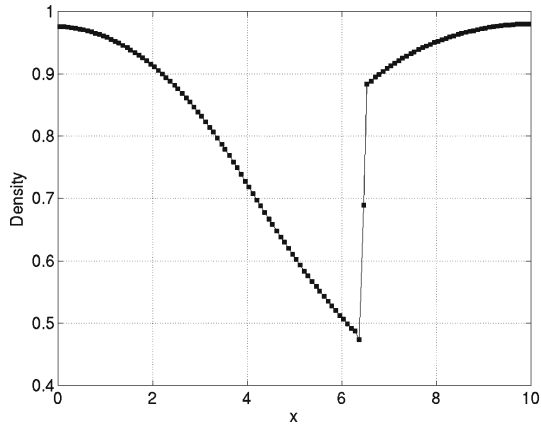


Fig. 5. Density solution for transonic quasi-1D nozzle flow.

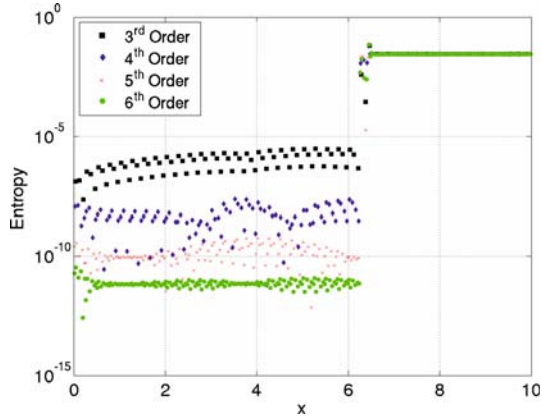


Fig. 6. Entropy error for transonic quasi-1D nozzle flow.

We have chosen a symmetric nozzle configuration on the interval [0,1] for this test case, where the area on the interval [0,1/2] is given by

$$A(x) = \begin{cases} 1, & x \leqslant 0.375, \\ 1 - (a_1\xi^5 + a_2\xi^6 + a_3\xi^7 + a_4\xi^8 + a_5\xi^9), & x > 0.375. \end{cases}$$

Here $\xi = (x - 0.375)/0.125$ and the area on the interval [1/2,1] is obtained by symmetry. Obviously the first four derivatives of the polynomial expression for the nozzle throat vanish at the intersection with the constant part of the geometry definition. The remaining coefficients are chosen such that

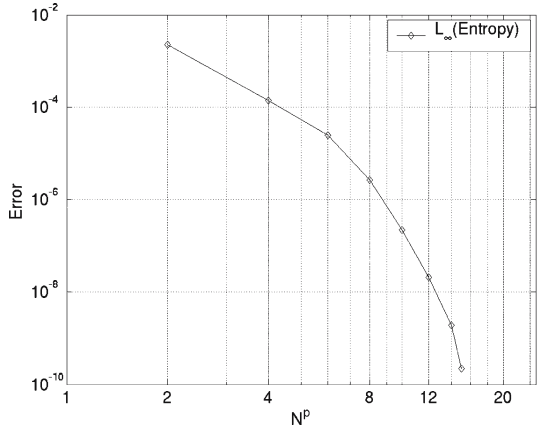


Fig. 7. Entropy error in p-refinement for smooth nozzle flow (16 cells), where p is the polynomial order, and $Np = p + 1$ is the order of the scheme.

the nozzle throat is given by $d_t = 0.85$, and all derivatives up to order four correspond to a circle with radius $r = 1 - d_t$ at the throat. Figure 7 shows the convergence of the entropy error with increasing order of the scheme on a mesh with 16 cells. Over a wide range of polynomial refinement the convergence can be seen to be spectral (i.e., exponential).

5.2. Accuracy Study in 2D

The vortex propagation problem has an exact solution for the Euler equations, and was used by Shu [22]. The mean flow is $\{\rho, u, v, p\} = \{1, 1, 1, 1\}$. An isotropic vortex is then added to the mean flow, i.e., with perturbations in u, v, and temperature $T = p/\rho$, and no perturbation in entropy $S = p/\rho^\gamma$:

$$
\begin{aligned}
(\delta u, \delta v) &= \frac{\varepsilon}{2\pi} e^{0.5(1-r^2)}(-\bar{y}, \bar{x}), \\
\delta T &= -\frac{(\gamma - 1)\varepsilon^2}{8\gamma\pi^2} e^{1-r^2}, \\
\delta S &= 0,
\end{aligned}
$$

where $r^2 = \bar{x}^2 + \bar{y}^2$, $\bar{x} = x - 5$, $\bar{y} = y - 5$, and the vortex strength $\varepsilon = 5$. If the computational domain is infinitely big, the exact solution of the Euler equations with the above initial conditions is just the passive convection of the isotropic vortex with the mean velocity (1, 1). In the following accuracy study, the computational domain is taken to be $[0, 10] \times [0, 10]$,

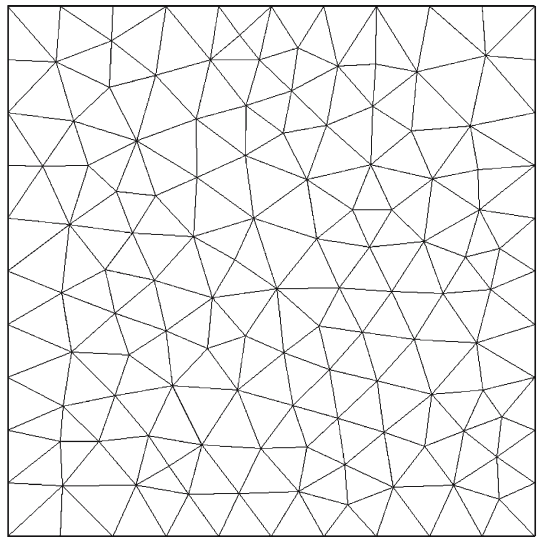


Fig. 8. Irregular "$10 \times 10 \times 2$" Grid.

with characteristic inflow and outflow boundary conditions imposed on the boundaries.

The numerical simulations were carried out until $t = 2$ on a set of irregular meshes. The coarsest mesh is shown in Fig. 8. The finer meshes are generated recursively by cutting each coarser grid cell into four finer grid cells. The Rusanov flux was used in the simulations. The L_1 and L_∞ norms in density are presented for SD schemes of second and third order in Table III. The errors presented in the tables were made time step independent by using sufficiently small time steps. Note that all the simulations have reached roughly the desired order of accuracy in the L_1 and L_∞ norms. The third-order scheme is about 0.3–0.4 orders from order 3. The SD method is about 25% faster than the SV method for this case.

A simpler density wave problem was also run with the third-order scheme on a set of regular triangular grids generated from cutting uniform Cartesian grids into triangles to see whether we could achieve closer to third-order accuracy. The problem has the following analytical solutions

$$\begin{aligned} \rho &= 2. + \sin(x + y - 0.3t), \\ u &= 1, \\ v &= -0.7, \\ p &= 1. \end{aligned}$$

Table III. Accuracy on the Vortex Propagation case ($t=2$) (Irregular Grids), Rusanov Flux

Order of accuracy	*Grid*	L_1 error	L_1 order	L_∞ error	L_∞ order
2	$10\times10\times2$	$6.97e-3$		$1.01e-1$	
	$20\times20\times2$	$1.80e-3$	1.95	$3.42e-2$	1.56
	$40\times40\times2$	$5.01e-4$	1.85	$9.08e-3$	1.91
	$80\times80\times2$	$1.30e-4$	1.95	$2.38e-3$	1.93
	$160\times160\times2$	$3.28e-5$	1.99	$6.36e-4$	1.90
3	$10\times10\times2$	$1.94e-3$	–	$5.43e-2$	–
	$20\times20\times2$	$3.55e-4$	2.45	$7.50e-3$	2.86
	$40\times40\times2$	$5.97e-5$	2.57	$1.32e-3$	2.51
	$80\times80\times2$	$9.79e-6$	2.61	$1.94e-4$	2.77
	$160\times160\times2$	$1.52e-6$	2.69	$3.22e-5$	2.59

Table IV. Accuracy on the Density wave Problem at $t=1$ (Regular Grids), Rusanov Flux

Order of accuracy	*Grid*	L_1 error	L_1 order	L_∞ error	L_∞ order
3	$10\times10\times2$	$1.69e-3$	–	$4.52e-3$	–
	$20\times20\times2$	$3.45e-4$	2.29	$9.64e-4$	2.23
	$40\times40\times2$	$5.66e-5$	2.61	$1.72e-4$	2.49
	$80\times80\times2$	$7.87e-6$	2.85	$2.50e-5$	2.78
	$160\times160\times2$	$1.02e-6$	2.95	$3.24e-6$	2.95

The computational domain is $[-\pi,\pi]\times[-\pi,\pi]$, and the simulation was carried out until $t=1$. The L_1 and L_∞ density errors are presented in Table IV. Note that the numerical orders of accuracy in both the L_1 and L_∞ norms have reached a full third order on the finest two meshes.

5.3. Double Mach Reflection

This problem is also a standard test case for high-order methods [31] and has been studied extensively by many researchers. The computational domain for this problem is chosen to be $[0,4]\times[0,1]$. The reflecting wall lies at the bottom of the computational domain starting from $x=1/6$. Initially a right-moving Mach 10 shock is positioned at $x=1/6, y=0$ and makes a 60° angle with the x-axis. For the bottom boundary, the exact post-shock condition is imposed for the region from $x=0$ to $x=1/6$ and a solid wall boundary condition is used for the rest. For the top boundary of the computational domain, the solution is set to describe

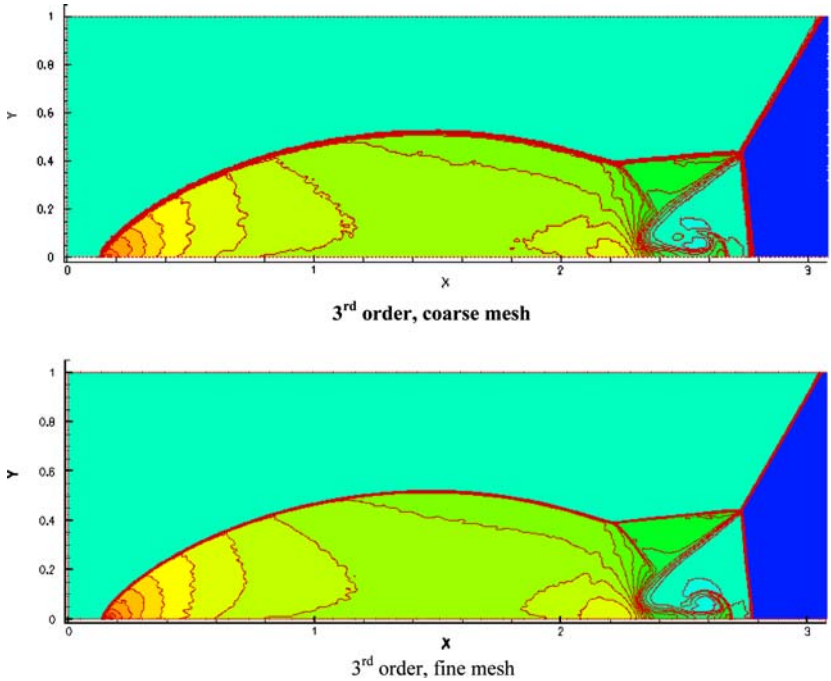


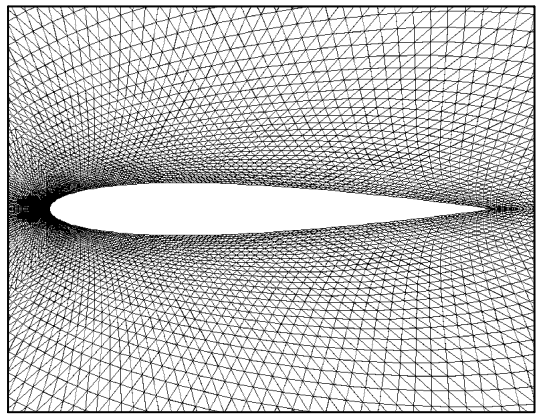
Fig. 9. Density contours for the double Mach reflection problem.

the exact motion of the Mach 10 shock. The left boundary is set as the exact post-shock condition, while the right boundary is set as outflow boundary. Two triangular grids were generated with 133,480, and 533,920 triangular cells, respectively. All the simulations were carried until $t = 0.2$ using the CUSP flux and TVB limiter similar to the one developed for the SV method [30]. Figure 9 shows the density contours computed with the third-order SD scheme on the coarse, and fine grids. This case clearly demonstrates the capability of crisp shock capturing on triangular meshes.

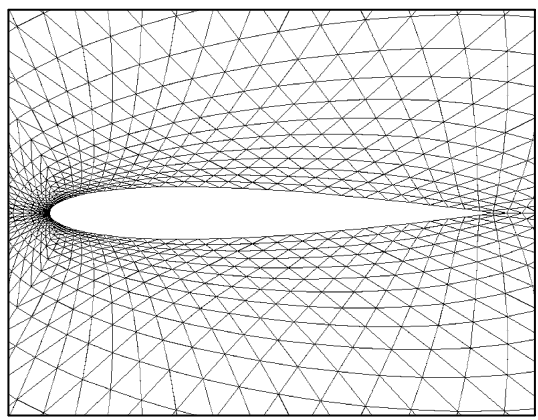
5.4. Subsonic Flow over a NACA0012 Airfoil

As a final demonstration for a more realistic geometry, subsonic flow around a NACA0012 airfoil at Mach = 0.4, and angle of attack of five degrees is simulated. In this simulation, the computational results using the third-order SD scheme on a coarse mesh with $72 \times 24 \times 2$ triangles are compared with those using a second-order MUSCL type FV method [26]

on a much finer mesh with $192 \times 64 \times 2$ triangles. Therefore the number of DOFs in the FV simulation is 24,576 while it is 20,736 in the SD simulation. The entropy production in the solution is used as the indicator for the solution accuracy. For the third-order SD scheme, the boundary is approximated with 72 piece-wise quadratic segments. For the second-order FV scheme, the airfoil surface is approximated with 192 linear segments. The computational meshes used for both the SD and FV methods are displayed in Fig. 10. The outer boundary is 20 chords away from the center of the airfoil. The computed Mach contours computed with both the

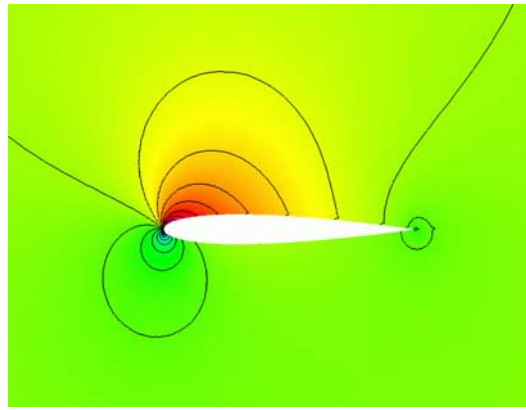


(a) 2nd FV grid

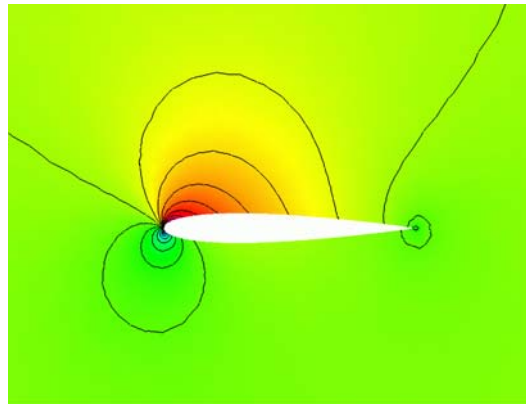


(b) 3rd SD grid

Fig. 10. Computational grids for subsonic flow around a NACA0012 airfoil.



(a) 2nd order FV



(b) 3rd order SD

Fig. 11. Comparison of Mach contours between the FV and SD methods.

SD and FV schemes are plotted in Fig. 11. Note that the agreement is very good. The average entropy error with the second-order FV method is $1.04e-5$, while the average entropy error with the third-order SD scheme is $4.86e-6$, which is more than a factor of 2 smaller. The entropy errors along the airfoil surface are plotted for both computational results in Fig. 12. Note that although the second-order FV scheme used a much finer grid, the solution quality of the third-order SD scheme is superior.

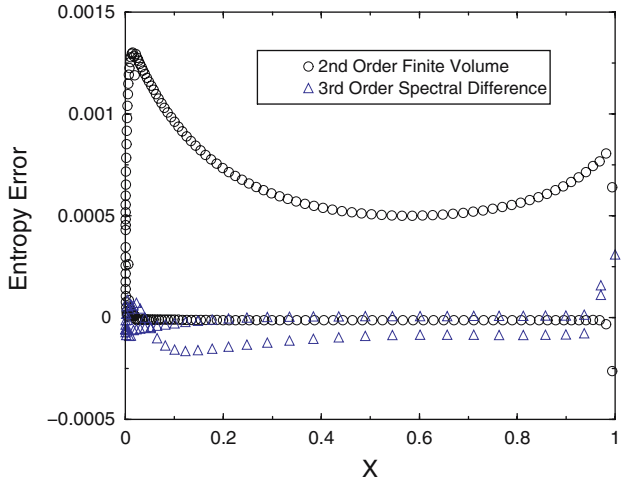


Fig. 12. Comparison of entropy error along the airfoil surface.

6. Conclusions

In this paper, the spectral difference method has been successfully extended to Euler equations on unstructured grids. The method combines the best features of structured and unstructured grid methods in which the structured distribution of discrete variables in each unstructured cell maintains computational efficiency and geometric flexibility. It utilizes the concept of discontinuous and high-order local representations to achieve conservation and high accuracy. Universal reconstructions are obtained by distributing unknown and flux points in a geometrically similar manner for all unstructured cells. The flux derivatives needed to update the conservative unknowns are expressed as universal weighted sums of the fluxes, leading to great computational efficiency. An important aspect of the method is that the number of Riemann solvers per unknown decreases as the order of accuracy increases, reducing the cost for higher order. Placements of the unknown and flux points with various orders of accuracy are given for triangular elements. Accuracy studies of the method are carried out with the vortex propagation problem and the order of accuracy is numerically verified. A monotonicity solution limiter has been implemented for discontinuity capturing, successfully employed to simulate 1D shock in a nozzle, and a double Mach reflection problem. The method is also applied to high-order boundary representations, and satisfactory results have been obtained for a subsonic flow around the NACA0012 airfoil. The third-order SD was shown to produce more accurate results than a second-order FV method on

a much coarser grid with fewer solution unknowns. Future research areas in the SD method include extension to even higher order of accuracy, and to the Navier-Stokes equations and three dimensions.

ACKNOWLEDGMENTS

The first author gratefully acknowledges partial support from Department of Energy grant DE-FG02-05ER25677, AFOSR grant FA9550-06-1-0146 and the start-up funding from the Department of Aerospace Engineering, College of Engineering of Iowa State University.

REFERENCES

1. Barth, J., and Frederickson, P. O. (1990). High-order solution of the Euler equations on unstructured grids using quadratic reconstruction. AIAA Paper No. 90-0013.
2. Bassi, F., and Rebay, S. (1997). High-order accurate discontinuous finite element solution of the 2D Euler equations. *J. Comput. Phys*. **138**, 251–285.
3. Canuto, C., Hussaini, M. Y., Quarteroni, A., and Zang, T. A. (1987). *Spectral Methods in Fluid Dynamics*, Springer-Verlag, New York.
4. Chen, Q., and Babuska, I. (1995). Approximate optimal points for polynomial interpolation of real functions in an interval and in a triangle. *Comput. Methods Appl. Mech. Engrg*. **128**, 405–417.
5. Cockburn, B., and Shu, C.-W. (1989). TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. Math. Comput. **52**, 411–435.
6. Cockburn, B., and Shu, C.-W. (1998). The Runge–Kutta discontinuous Garlerkin method for conservation laws V: multidimensional systems. *J. Comput. Phys.* **141**, 199–224.
7. Delanaye, M., and Liu, Y. (1999). Quadratic reconstruction finite volume schemes on 3D arbitrary unstructured polyhedral grids. AIAA Paper No. 99-3259-CP
8. Godunov, S. K. (1959). A finite-difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics. *Mater. Sb.* **47**, 271.
9. Harten, A. (1983). High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys.* **49**, 357–393.
10. Hesthaven, J. S. (1998). From electrostatics to almost optimal nodal sets for polynomial interpolation in a simplex. *SIAM J. Numer. Anal*. **35**, 655–676.
11. Jameson, A. (1995). Analysis and design of numerical schemes for gas dynamics 2: artificial diffusion and discrete shock structure. *Int. J. Comput. Fluid Dyn.* **5**, 1–38.
12. Kopriva, D. A. (1996). A conservative staggered-grid Chebyshev multidomain method for compressible flows. II semi-structured method. *J. Comput. Phys.* **128**, 475.
13. Kopriva, D. A. (1998). A staggered-grid multidomain spectral method for the compressible Navier–Stokes equations. *J. Comput. Phys.* **143**(1), 125.
14. Liu, Y., and Vinokur, M. (1998). Exact integration of polynomials and symmetric quadrature formulas over arbitrary polyhedral grids. *J. Comput. Phys.* **140**, 122–147.
15. Liu, Y., Vinokur, M., and Wang, Z. J. (2004). Discontinuous spectral difference method for conservation laws on unstructured grids. *Proceedings of the 3rd International Conference in CFD*, Toronto, Canada.

16. Liu, Y., Vinokur, M., and Wang, Z. J. (2006). Spectral difference method for unstructured grids I: basic formulation. *J. Comput. Phys*. **216**(2), 780–801.
17. Liu, Y., Vinokur, M., and Wang, Z. J. (2006). Spectral (finite) volume method for conservation laws on unstructured grids V: extension to three-dimensional systems. *J. Comput. Phys.* **212**, 454–472.
18. Roe, P. L. (1981). Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.* **43**, 357–372.
19. Rusanov, V. V. (1961). Calculation of interaction of non-steady shock waves with obstacles. *J. Comput. Math. Phys. USSR* **1**, 267–279.
20. Shu, C.-W. (1987). TVB uniformly high-order schemes for conservation laws. *Math. Comput.* **49**, 105–121.
21. Shu, C.-W. (1988). Total-variation-diminishing time discretizations. *SIAM J. Sci. Stat. Comput.* **9**, 1073.
22. Shu, C.-W. (1998). Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In B. Cockburn, C. Johnson, C.-W. Shu and E. Tadmor (A. Quarteroni, ed.), *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, Lecture Notes in Mathematics, Vol. 1697, Springer, Berlin, pp. 325–432.
23. Spiteri, R. J., and Ruuth, S. J. (2002). A new class of optimal high-order strong-stability-preserving time discretization methods. *SIAM J. Numer. Anal.* **40**, 469–491.
24. Sridar, D., and Balakrishnan, N. (2003). An upwind finite difference scheme for meshless solvers. *J. Comput. Phys.* **189**, 1–29.
25. van Leer, B. (1979). Towards the ultimate conservative difference scheme V. a second order sequel to Godunov's method. *J. Comput. Phys.* **32**, 101–136.
26. Wang, Z. J. (2000). A fast nested multi-grid viscous flow solver for adaptive Cartesian/quad grids. Int. J. Numer. Methods Fluids **33**, 657–680.
27. Wang, Z. J. (2002). Spectral (finite) volume method for conservation laws on unstructured grids: I. Basic formulation. *J. Comput. Phys.* **178**, 210.
28. Wang, Z. J., and Liu, Y. (2002). Spectral (finite) volume method for conservation laws on unstructured grids II: extension to two-dimensional scalar equation. *J. Comput. Phys.* **179**, 665–697.
29. Wang, Z. J., and Liu, Y. (2004).Spectral (finite) volume method for conservation laws on unstructured grids III: one-dimensional systems and partition optimization. *J. Scientific Comput.* **20**, 137–157.
30. Wang, Z. J., Zhang, L., and Liu, Y. (2004). Spectral (finite) volume method for conservation laws on unstructured grids IV: extension to two-dimensional systems. *J. Comput. Phys.* **194**, 716–741.
31. Woodward, P., and Colella, P. (1984). The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.* **54**, 115–173.
32. Wolfram, S. (1999). *The Mathematica Book*, 4th ed., Wolfram Media and Cambridge University Press, New York.