AIAA-2002-1058

A HIGH-ORDER SPECTRAL (FINITE) VOLUME METHOD FOR CONSERVATION LAWS ON UNSTRUCTURED GRIDS

Z.J. Wang[¶]
Department of Mechanical Engineering
Michigan State University, East Lansing, MI 48824

Yen Liu[*]
MS T27B-1, NASA Ames Research Center
Moffett Field, CA 94035-1000

ABSTRACT

A time accurate, high-order, conservative, finite volume method named Spectral Volume (SV) method has been developed for conservation laws on unstructured grids. The concept of a "spectral volume" is introduced to achieve high-order accuracy in an efficient manner similar to spectral element and multi-domain spectral methods. Each spectral volume is partitioned into control volumes (CVs), and cell-averaged data from these control volumes is used to reconstruct a high-order polynomial approximation in the spectral volume. Riemann solvers are used to compute the fluxes at spectral volume boundaries. Then cell-averaged state variables in the control volumes are updated independently. Furthermore, TVD (Total Variation Diminishing) and TVB (Total Variation Bounded) limiters are introduced in the SV method to remove spurious oscillations near discontinuities. A very desirable feature of the SV method is that the reconstruction is identical for cells of the same type with similar partitions, and that the reconstruction stencil is always non-singular, in contrast to the memory and CPU-intensive reconstruction in a high-order k-exact finite volume (FV) method. The high-order accuracy of the SV method is demonstrated for several model problems with and without discontinuities.

1. INTRODUCTION

Numerical algorithms for conservation laws have been extensively researched in the last three decades because conservation laws govern many physical disciplines such as fluid dynamics, electromagnetics, aeroacoustics, to name just a few. One of the most successful algorithms for conservation laws is the Godunov method [12], which laid a solid foundation for the development of modern upwind methods including MUSCL [30], PPM [10], ENO [13] and weighted ENO (WENO) schemes [16,20]. There are two key components in a Godunov-type scheme. One is data reconstruction, and the other is the Riemann solver. The Godunov scheme employed a piece-wise constant data reconstruction, and the resultant scheme was only first-order accurate. Van Leer extended the first-order Godunov scheme to second-order [30] by using a piecewise linear data reconstruction. In addition, limiters were also introduced to remove spurious numerical oscillations near steep gradients. Meanwhile the exact Riemann solver used in the Godunov scheme was sometimes replaced by approximate "Riemann solvers" or flux-splitting procedures for better efficiency [29,26,31,24,15,19].

The difficulty in generating smooth structured grids for complex geometries has prompted intensive research and development of unstructured grid algorithms in the last two decades [1-3,17,22,23,32-33]. Most of the unstructured grid methods are second-order accurate because they are relatively easy to implement, and are quite memory efficient. Several high-order schemes have been developed for unstructured grids. For example, a high-order k-exact finite volume scheme was developed by Barth and Frederickson in [3], an ENO scheme for unstructured grid was developed by Abgrall in [1], and a WENO scheme was developed by Hu and Shu in [14]. Although high-order accurate finite volume schemes can be obtained theoretically for an unstructured grid by using high-order polynomial data reconstructions, higher than linear reconstructions are rarely used in three dimensions in practice. This is mainly because of the difficulty in finding valid (non-singular) stencils, and the enormous memory required to store the coefficients used in the reconstruction. For each control volume, the reconstruction stencil is

[¶] Associate Professor (zjw@egr.msu.edu), AIAA Senior Member

[*] Research Scientist (liu@nas.nasa.gov)

unique for an unstructured grid as shown in Figure 1a. A data reconstruction must be performed at each iteration for each control volume. This reconstruction step is the most memory and time consuming in higher than second-order schemes. In a recent implementation of a third-order *FV* scheme with a quadratic reconstruction in three dimensions by Delanaye and Liu [11], the average size of the reconstruction stencils is about 50-70. Still there are many singular reconstruction stencils. The size of the reconstruction stencils usually increases non-linearly with the order of accuracy. For a fourth order *FV* scheme, the average stencil size is estimated to be at least 120. It is very memory and CPU intensive to perform the reconstruction.

More recently, another high-order conservative algorithm called the Discontinuous Galerkin (DG) method was developed by Cockburn, Shu, et al and others [7-9,4]. In the DG method, a high-order data distribution is assumed for each element. As a result, the state variable is usually not continuous across element boundaries. The fluxes through the element boundaries are computed using an approximate Riemann solver, similar to a *FV* method. The residual is then minimized with a Galerkin approach. Due to the use of Riemann fluxes cross element boundaries, the DG method is fully conservative. A disadvantage of the DG method is that very high-order surface and volume integrals are necessary, which can be expensive to compute. Another high-order conservative scheme for unstructured quadrilateral grids is the multi-domain spectral method on staggered grids developed by Korpriva and Kolias [18]. The multi-domain spectral method is similar to the spectral element method by Patera [25], which is not conservative. Although very high-order of accuracy was achievable with both methods, the methods are difficult to extend to other cell types such as triangles, or tetrahedra.

In this paper, a new conservative high-order *SV* method is developed for conservation laws on unstructured grids. In the next section, we present the basic framework of the *SV* method on triangular grids together with a TVD Runge-Kutta time integration scheme. In Section 3, the reconstruction problem based on *CV*-averaged solutions is studied, and it is shown that the reconstruction problems on all triangles with a similar partition are identical. In addition, convergent linear to cubic SVs are presented. Section 4 discusses issues related to discontinuity capturing and TVD and TVB limiters are presented. In Section 5, numerical implementations of the *SV* method for both linear and non-linear scalar conservation laws are carried out, and accuracy studies are performed for both linear and non-linear wave equations to verify the numerical order of accuracy. The shock-capturing capability of the method is also demonstrated with the Burger's equation. Finally, conclusions and recommendations for further investigations are summarized in Section 6.

2. SPECTRAL (FINITE) VOLUME METHOD

Consider a mesh of unstructured triangular cells. Each cell is called a Spectral Volume, denoted by S_i, which is further partitioned into subcells named Control Volumes (CVs), indicated by $C_{i,j}$, as shown in Figure 1b. To represent the solution as a polynomial of degree m in two dimensions (2D) we need $N = (m+1)(m+2)/2$ pieces of independent information, or degrees of freedom (DOFs). The DOFs in a SV method are the volume-averaged mean variables at the N CVs. For example, to build a quadratic reconstruction in 2D, we need at least (2+1)(3+1)/2 = 6 DOFs. There are numerous ways of partitioning a SV, and not every partition is admissible in the sense that the partition may not be capable of producing a degree m polynomial. Such partitions are also called singular ones. Once N mean solutions in the CVs of an admissible SV are given, a unique polynomial reconstruction can be obtained from

$$p_i(x,y) = \sum_{j=1}^{N} L_j(x,y)\overline{u}_{i,j}\ , \tag{1}$$

where $p_i(x,y) \in P_m$ (the space of polynomials of degree m or less), $L_j(x,y) \in P_m,\ j = 1,\cdots,N$ are the "shape" functions satisfying

$$\int_{C_{i,j}} L_k(x,y)dV = V_{i,j}\delta_{jk}\ . \tag{2}$$

where $V_{i,j}$ is the volume of $C_{i,j}$. This high-order polynomial reconstruction facilitates a high-order update for the mean solution of each CV. Consider the following hyperbolic conservation law

$$u_t + \nabla \bullet F = 0\ , \tag{3}$$

where F is the flux vector. Integrating (3) in each *CV*, we obtain

$$\frac{d\overline{u}_{i,j}}{dt}V_{i,j} + \sum_{r=1}^{K}\int_{A_r}(F \bullet \boldsymbol{n})dA = 0, \tag{4}$$

where K is the total number of faces in $C_{i,j}$, and $\overline{u}_{i,j}$ is the volume-averaged solution at $C_{i,j}$. The flux integral in (4) is then replaced by a Gauss-quadrature formula which is exact for polynomials of degree m

$$\int_{A_r}(F \bullet \boldsymbol{n})dA \approx \sum_{q=1}^{J} w_{rq}F(u(x_{rq,}y_{rq})) \bullet \boldsymbol{n}_r A_r, \tag{5}$$

where J is the number of quadrature points on the *r-th* face, w_{rq} are the Gauss quadrature weights, (x_{rq}, y_{rq}) are the Gauss quadrature points. Since the reconstructed polynomials are piece-wise continuous, the solution is usually discontinuous across the boundaries of a SV, although it is continuous across interior CV faces. The fluxes at the interior faces can be computed directly based on the reconstructed solutions at the quadrature points. The fluxes at the boundary faces of a SV are computed using approximate Riemann solvers given the left and right reconstructed solutions, i.e.,

$$\begin{aligned} &F(u(x_{rj}, y_{rj})) \bullet \boldsymbol{n_r} \approx \\ &F_{Riem}(p_i(x_{rj}, y_{rj}), p_{i,r}(x_{rj}, y_{rj}), \boldsymbol{n_r}) \end{aligned} \tag{6}$$

where $p_{i,r}(x_{rq}, y_{rq})$ is the reconstructed polynomial in a neighboring SV sharing face *r* with the *SV* in consideration, S_i. Obviously, the approximate Riemann solver must satisfy

$$\begin{aligned} &F_{Riem}(p_i(x_{rq}, y_{rq}), p_{i,r}(x_{rq}, y_{rq}), \boldsymbol{n_r}) = \\ &-F_{Riem}(p_{i,r}(x_{rq}, y_{rq}), p_i(x_{rq}, y_{rq}), -\boldsymbol{n_r}), \end{aligned}$$

to achieve discrete conservation. It has been shown [34] that order of accuracy of this SV scheme is *(m+1)-th* order. In addition, the scheme is compact in the sense that a high-order polynomial is reconstructed in each SV without using any data from neighboring SVs. This property can potentially translate into significantly reduced communication cost compared to a k-exact FV scheme (for example) when implemented on parallel computers.

Note that one of the subtle differences between a *FV* method and a *SV* method is that all the *CVs* in a *SV* use the *same* data reconstruction. As a result, it is not necessary to use a Riemann flux or flux splitting for the interior boundaries between the *CVs* inside a particular *SV* because the state variable is continuous across the interior *CV* boundaries. Riemann fluxes are only necessary at the boundaries of the *SV*.

For time integration, we use a third-order TVD Runge-Kutta scheme from [27]. We first rewrite (4) in a concise ODE form

$$\frac{d\overline{u}}{dt} = R_h(\overline{u}), \tag{7}$$

Then the third-order TVD Runge-Kutta scheme can be expressed as:

$$\begin{aligned} \overline{u}^{(1)} &= \overline{u}^n + \Delta t R_h(\overline{u}^n)\ ; \\ \overline{u}^{(2)} &= \frac{3}{4}\overline{u}^n + \frac{1}{4}[\overline{u}^{(1)} + \Delta t R_h(\overline{u}^{(1)})]; \\ \overline{u}^{n+1} &= \frac{1}{3}\overline{u}^n + \frac{2}{3}[\overline{u}^{(2)} + \Delta t R_h(\overline{u}^{(2)})]. \end{aligned} \tag{8}$$

The SV method shares many advantages with the DG method [7-9] in that it is compact which is suitable for parallel computing, high-order accurate, conservative, and capable of handling complex geometries. The SV method is expected to be more efficient than the DG method because high-order volume integrals are avoided, and lower order surface Gauss integral formula can be used (*m-th* order vs. *2m-th* order). In addition, the SV method should have higher resolution than the DG method for discontinuities because of the availability of local cell-averaged state variables at the CVs.

3. DATA RECONSTRUCTION

The reconstruction problem reads: Given a continuous function in S_i, $u \in \mathcal{C}(S_i)$ (the space of continuous functions in S_i), and a partition Π_m of S_i, find $p_i \in P_m$, such that

$$\int_{C_{i,j}} p_i(x,y)dV = \int_{C_{i,j}} u(x,y)dV, \quad j = 1,\cdots,N\ . \tag{9}$$

To actually solve the reconstruction problem, we introduce the complete polynomial basis, $e_l(x,y) \in P_m$, where $P_m = span\{e_l(x,y)\}_{l=1}^{N}$. Therefore p_i can be expressed as

$$p_i = \sum_{l=1}^{N} a_l e_l(x,y), \tag{10}$$

or in the matrix form

$$p_i = e\, a, \tag{11}$$

where e is the basis function vector $[e_1,...,e_N]$ and a is the reconstruction coefficient vector $[a_1,...,a_N]^T$. Substituting (10) into (9), we then obtain

$$\frac{1}{V_{i,j}}\sum_{l=1}^{N} a_l \int_{C_{i,j}} e_l(x,y)dV = \overline{u}_{i,j}, \quad j = 1,\cdots,N \tag{12}$$

Let $\overline{u}_i$ denote the column vector $[\overline{u}_{i,1},...,\overline{u}_{i,N}]^T$, Eq. (12) can be rewritten in the matrix form

$$R\,a = \overline{u}_i, \tag{13}$$

where the reconstruction matrix

$$R = \begin{bmatrix} \frac{1}{V_{i,1}}\int_{C_{i,1}} e_1(x,y)\, dV & \cdots & \frac{1}{V_{i,1}}\int_{C_{i,1}} e_N(x,y)\, dV \\ \vdots & \cdots & \vdots \\ \frac{1}{V_{i,N}}\int_{C_{i,N}} e_1(x,y)\, dV & \cdots & \frac{1}{V_{i,N}}\int_{C_{i,N}} e_N(x,y)\, dV \end{bmatrix}.$$

The reconstruction coefficients a can be solved as

$$a = R^{-1}\overline{u}_i, \tag{14}$$

provided that the reconstruction matrix R is nonsingular. Substituting Eq. (14) into Eq. (11), p_i is then expressed in terms of *cardinal* basis functions (or shape functions) $L = [L_1,...,L_N]$

$$p_i = \sum_{j=1}^{N} L_j(x,y)\,\overline{u}_{i,j} = L\,\overline{u}_i. \tag{15}$$

Here L is defined as

$$L \equiv e\,R^{-1}. \tag{16}$$

Equation (15) gives the functional representation of the state variable *u* within the *SV*. The function value of *u* at a quadrature point or any point (x_{rq}, y_{rq}) within the *SV* is thus simply

$$p_i(x_{xq}, y_{rq}) = \sum_{j=1}^{N} L_j(x_{rq}, y_{rq})\,\bar{u}_{i,j}. \qquad (17)$$

The above equation can be viewed as an interpolation of a function value at a point using a set of cell averaged values with each weight equal to the corresponding cardinal basis functional value evaluated at that point.

In the case of triangular SV, it can be proven that the reconstruction coefficients in (17) are identical for all triangles with similar partitions [35]. We thus have a universal reconstruction formula, Eq. (17), for evaluating the state variable *u* at similar points. This also implies that the reconstruction needs to be carried out only once, and that can be performed using any shape of triangle. In our study, we use Mathematica to carry out the reconstruction analytically. The exact integrations of polynomials over arbitrary polygons can be found in [21].

Since the reconstruction problem is equivalent for all triangles, we focus our attention on the reconstruction problem in an equilateral triangle *E*. In partitioning *E* into *N* non-overlapping *CVs*, we further require that the *CVs* satisfy the following three conditions:

1. The *CVs* are "symmetric" with respect to all symmetries of the triangle;
2. All *CVs* are convex;
3. All *CVs* have straight sides, i.e., the *CVs* are polygons.

We believe the symmetry and convexity requirement is important for achieving the best possible accuracy and robustness. The requirement of polygons simplifies the formulation of the *SV* method. It is then obvious that a *CV* containing the centroid of *E* must be symmetric with respect to the three edges and vertices, and at most one such *CV* can exist. This *CV*, if it exists, is thus said to possess degree 1 symmetry (or 1 symmetry, in short). Similarly, *CVs* with degree 3 and 6 symmetries can also be defined. For example, if a *CV* is said to possess degree 3 symmetry, then two other symmetric *CVs* must exist in the same partition. We shall denote n_1, n_3 and n_6 the number of degree 1, 3 and 6 symmetry groups in a partition with $n_1 = 0$ or 1. Then the total number of *CVs* in the partition is then $n_1 + 3n_3 + 6n_6$. In order to support the unique reconstruction of a degree *m* polynomial, the total number of *CVs* must be identical to the dimension of the polynomial space, i.e.,

$$n_1 + 3n_3 + 6n_6 = \frac{(m+1)(m+2)}{2}. \qquad (18)$$

The solutions of (18) can be used to guide the partition of *E* once *m* is given. Some possible partitions of the standard triangle corresponding to these solutions for *m* = 1, 2, 3 are shown in Figures 2-4. Next the question of how these partitions perform in a data reconstruction needs to be answered.

In [34], the first paper on the *SV* method, it was shown that not all non-singular reconstructions are convergent. For example, high-order polynomial reconstructions based on equidistant *CVs* in one dimension are not convergent although the reconstructions are non-singular. We believe this is the direct consequence of the Runge phenomenon. Therefore some means to quantify the quality of the reconstructions needs to be identified.

For any $u \in \mathcal{C}(E)$, there exists a unique degree m polynomial p_i which satisfies (9) for any admissible partition. Denote $p_i = \Gamma_\Pi(u)$, where Γ_Π is a projection operator, which maps $\mathcal{C}(E)$ onto $P_m(E)$. When both spaces $\mathcal{C}(E)$ and $P_m(E)$ are equipped with the uniform norm, i.e., $\|\bullet\| = \|\bullet\|_\infty = \max|\bullet|$, the norm of this projection operator can be defined as

$$\|\Gamma_\Pi\| = \sup_{u \neq 0} \frac{\|\Gamma_\Pi u\|}{\|u\|}. \qquad (19)$$

It can be shown that

$$\|\Gamma_\Pi\| = \max_{(x,y)\in E} \sum_{j=1}^{N} \left|L_j(x,y)\right|. \qquad (20)$$

$\|\Gamma_\Pi\|$ is called the Lebesgue constant, which is of interest for the following reason [5-6]:

If p_m^* is the best uniform approximation to *u* on *E*, then

$$\|u - \Gamma_\Pi u\| \le \left(1 + \|\Gamma_\Pi\|\right)\|u - p_m^*\|. \qquad (21)$$

Thus $\|\Gamma_\Pi\|$ gives a simple method of bounding the interpolation polynomial. It is obvious from (21) that the smaller the Lebesgue constant, the better the interpolation polynomial is to be expected. Therefore the problem becomes finding the partition with a small Lebesgue constant, if not as small as possible. In this paper, our focus is to construct convergent *SV* partitions when the *SV* is refined. The Lebesgue constant is used as the criterion to judge the quality of the partitions. The optimization of the partitions will be the subject of a future publication. Through extensive testing, the following linear to cubic SVs have been found.

Linear Spectral Volume (*m* = 1)

Two partitions are possible, as shown in Figure 2a and 2b, which are named Type 1 and Type 2 partitions. Since the centroids of the *CVs* are non-co-linear, both partitions are admissible. Note that the *CVs* in both partitions possess a degree 3 symmetry. The Lebesgue constants are 13/3 (4.3333) and 43/15 (2.8667) for Type 1 and 2 partitions, respectively. Note that the Type 2 *SV* has a much smaller Lebesgue constant than the Type 1 *SV*, indicating that the L_∞ error with the Type 2 *SV* should be smaller than the error with the Type 1 *SV*.

Quadratic Spectral Volume (*m* = 2)

Two possible partitions for m = 2 are shown in Figure 3. The partition presented in Figure 2a is not unique in the sense that the position of one vertex on an edge of the triangle can change, i.e., the length *d* shown in Figure 3a can be any real number in (0, 0.5) assuming the length of the edge is 1. It seems that with any *d*, the partition is admissible. In our numerical studies, two different values of *d* were tested, namely *d* = 1/3 and *d* = 1/4, which are called Type 1 and Type 2 partitions, respectively. The Lebesgue constant for the Type 1 partition is 9.3333, and for the Type 2 partition is 8. Therefore, the Type 2 partition is expected to yield more accurate numerical results. Although the partition shown in Figure 3b looks reasonable, it is not admissible.

Cubic Spectral Volume (*m* = 3)

Three possible partitions for m = 3 are shown in Figure 4. For the partition shown in Figure 4b, the parameter *d* can be changed to obtain different partitions. In fact, the Lebesgue constants for partitions with a set of *d* values are presented in Table 1. Among this set of *d* values, it is interesting to note that the Lebesgue constant reaches a smallest value of 3.44485 at *d* = 1/15 from a value of 8.21499 at *d* = 1/6. When *d* is smaller than 1/15, the Lebesgue constant starts to increase. For presentation purpose, we call the partition shown in Figure 4a the Type 1 partition. The partition shown in Figure 4b with *d* = 1/6 is called the Type 2 partition, and with *d* = 1/15 the Type 3 partition. It is expected that the Type 3 partition should give the most accurate numerical solution in the uniform norm. The Lebesgue constant for the Type 1 partition is 167/12 (13.9), which is significantly larger than those for the Type 2 and 3 partitions. Numerical results to be presented later confirm that the Type 1 partition is not convergent with grid refinement. The partition shown in Figure 4c is singular.

Table 1 Lebesgue Constants for the Partition Shown in 4b

d	Lebesgue constant
1/6	8.21499
1/7	6.71178
1/8	5.71904
1/10	4.49231
1/15	3.44485
1/20	3.57595
1/25	3.65981
1/100	3.93353

4. MULTI-DIMENSIONAL LIMITERS

A limiter in a high-order numerical method such as the SV method should satisfy the following two requirements: 1) non-oscillatory, sharp resolution of discontinuities, and 2) recover the full formal order of accuracy away from the discontinuities. To that end, a TVB limiter [28] has been implemented.

Refer to (4), which is used to update the CV-averaged state variable. Denote

$$\Delta u_{rq} = p_i(x_{rq}, y_{rq}) - \bar{u}_{i,j},\ r = 1, \cdots, K;\ q = 1, \cdots, J.$$

Following the TVB idea, if

$$\left|\Delta u_{rq}\right| \le 4Mh_{rq}^2,\ r = 1, \cdots, K;\ q = 1, \cdots, J, \qquad (22)$$

it is not necessary to do any data limiting. In (22), *M* represents some measure of the second derivative of the solution, and h_{rq} is the distance from point (x_{rq}, y_{rq}) to the centroid of $C_{i,j}$. If for any value of *r* and *q*, (22) is violated, it is assumed that $C_{i,j}$ is near a steep gradient and data limiting is necessary. Instead of using the polynomial $p_i(x, y)$ in $C_{i,j}$, we assume linear data distribution in $C_{i,j}$, i.e.,

$$u_{i,j}(x, y) = \bar{u}_{i,j} + \nabla u_{i,j} \bullet (\boldsymbol{r} - \boldsymbol{r}_{i,j}),\quad \forall \boldsymbol{r} \in C_{i,j}, \qquad (23)$$

where $\boldsymbol{r}_{i,j}$ is the position vector of the centroid of $C_{i,j}$. In order to achieve the highest resolution, we need to maximize the magnitude of the solution gradient $\nabla u_{i,j}$. At the same time, we require that the reconstructed solutions at the quadrature points of $C_{i,j}$ satisfy the following monotonicity constraint:

$$\bar{u}_{i,j}^{\min} \le u_{i,j}(x_{rq}, y_{rq}) \le \bar{u}_{i,j}^{\max}, \qquad (24)$$

where $\bar{u}_{i,j}^{\min}$ and $\bar{u}_{i,j}^{\max}$ are the minimum and maximum cell-averaged solutions among all its neighboring *CVs* sharing a face with $C_{i,j}$. A very efficient approach can be used to compute the gradient. In this approach, we avoid a separate data reconstruction by reusing the polynomial reconstruction already available for the *SV*. For each *CV*, we use the gradient of the reconstructed polynomial at the *CV* centroid, i.e.,

$$\nabla u_{i,j} = \left(\frac{\partial p_i}{\partial x}, \frac{\partial p_i}{\partial y} \right)\bigg|_{\mathbf{r}_{i,j}} . \qquad (25)$$

This gradient is then limited if necessary to satisfy (24). If any of the reconstructed variable at the quadrature points is out of the range [$\overline{u}_{i,j}^{\min}, \overline{u}_{i,j}^{\max}$], the gradient is limited, i.e.,

$$\nabla u_{i,j} \Leftarrow \varphi \nabla u_{i,j},$$

where $\varphi \in [0,1]$ is calculated from:

$$\varphi = \begin{cases} \min\left(1, \dfrac{\Delta u_{rq}}{\overline{u}_{i,j}^{\max} - \overline{u}_{i,j}}\right) & if\ \Delta u_{rq} > 0 \\ \min\left(1, \dfrac{\Delta u_{rq}}{\overline{u}_{i,j} - \overline{u}_{i,j}^{\min}}\right) & if\ \Delta u_{rq} < 0. \\ 1 & otherwise \end{cases}$$

Note that if parameter M = 0, the *TVB* limiter is *TVD* (total variation diminishing), which strictly enforces monotonicity by sacrificing accuracy near extrema.

The availability of cell-averaged data on the *CVs* inside a *SV* makes this *CV*-based data limiting possible, whereas in the *DG* method, one can only perform an element based data limiting. Due to the increased local resolution, the *SV* method has been shown to have higher resolutions for discontinuities than the *DG* method [34].

5. NUMERICAL TESTS

Accuracy Study with 2D Linear Wave Equation

In this case, we test the accuracy of the *SV method* on the two-dimensional linear equation:

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = 0, \quad -1 \le x \le 1, \quad -1 \le y \le 1,$$

$$u(x, y, 0) = u_0(x, y), \qquad periodic \quad b.c.$$

The initial condition is $u_0(x, y) = \sin \pi(x + y)$. A fourth-order accurate Gauss quadrature formula [21] is used to compute the *CV*-averaged initial solutions. These *CV*-averaged solutions are then updated at each time step using the third-order TVD Runge-Kutta scheme presented earlier. The numerical simulation is carried until t = 1 on two different triangular grids as shown in Figure 5. One gird is generated from a uniform Cartesian grid by cutting each Cartesian cell into two triangles, and is named the regular grid. The other grid is generated with an unstructured grid generator, and is named the irregular grid. Note that the cells in the irregular grid have quite different sizes. In Table 2, we present the L_1 and L_∞ errors in the *CV*-averaged solutions produced using second to fourth order *SV method* schemes with *SVs* shown in Figures 2-4 on the regular grid. The errors presented in the table are time-step independent because the time step Δt was made small enough so that the errors are dominated by the spatial discretization. In this test, all *SVs* except the Type 1 cubic *SV* (shown in Figure 4a) are convergent with grid refinement on this regular grid. It is obvious that the expected order of accuracy is achieved by all the convergent *SVs*. It is not surprising that the Type 1 cubic *SV* is not convergent because of its rather large Lebesgue constant of 13.9. In contrast, the Type 2 and 3 cubic *SVs* have Lebesgue constants of 8.21 and 3.44 respectively. It is interesting to note that the Type 1 linear *SV* gives more accurate results in both the L_1 and L_∞ norms than the Type 2 linear *SV* even if the Type 1 *SV* has a larger Lebesgue constant of 4.33 versus that of 2.87, of the Type 2 *SV*. This indicates that the Lebesgue constant cannot serve as an absolute error estimator, but rather an estimate of the upper bound of the error. For the quadratic and cubic *SVs*, the partitions with smaller Lebesgue constants do give more accurate numerical solutions, as shown in Table 2.

Next, the L_1 and L_∞ errors in the numerical results computed on the irregular grid using second- fourth order *SVs* are shown in Table 3. This should be a much tougher test case because of the truly unstructured nature of the computational grid. What is striking is that the Type 1 linear *SV* failed to achieve second-order accuracy on this grid. As a matter of fact, it is only first order accurate. This may be contributed to the acute angles of the *CVs* in the partition. Note that both quadratic *SVs* are convergent, and give similar results. Third order accuracy is achieved by both types of quadratic *SVs* in the L_1 norm although the numerical order of accuracy in the L_∞ norm is only slightly over second-order. We believe this is due to the non-smoothness of the computational grid. The Type 2 cubic *SV* also showed a non-convergent behavior in the L_∞ norm on the finest grid. It is nice to see that the Type 3 cubic *SV* is not only convergent, but also fourth-order accurate in both the L_1 and L_∞ norms.

Accuracy Study with 2D Burgers Equation

In this case, we test the accuracy of the *SV method* on the two-dimensional non-linear wave equation:

$$\frac{\partial u}{\partial t} + \frac{\partial u^2/2}{\partial x} + \frac{\partial u^2/2}{\partial y} = 0, \quad -1 \le x \le 1, -1 \le y \le 1,$$

$$u(x, y, 0) = \frac{1}{4} + \frac{1}{2} \sin \pi(x + y), \qquad periodic\ b.c.$$

The initial solution is smooth. Due to the non-linearity of the Burgers equation, discontinuities will develop in the solution. Therefore we also test the capability of the method to achieve uniform high-order accuracy away from discontinuities. At t = 0.1, the exact solution is still smooth, as shown in Figure 6a. The numerical simulation is therefore carried out until t = 0.1 without the use of limiters on the irregular grid as shown in Figure 5b. The numerical solution on the 20x20x2 irregular grid computed with the Type 1 quadratic *SV* (third-order accurate) is displayed in Figure 6b. Notice that the agreement between the numerical and exact solutions is excellent. In Table 4, the L_1 and L_∞ errors on the irregular grid are presented. The performance of the *SV method* on the non-linear Burgers equation is quite similar to the performance on the linear wave equation, although there is a slight loss of accuracy (from 0.1 – 0.6 orders) especially on the irregular grid in the L_∞ norm, probably due to the non-linear nature of the Burgers equation. Once again, the Type 1 linear *SV* has difficulty in achieving second-order accuracy on the irregular grid in both norms.

At t = 0.45, the exact solution has developed two shock waves as shown in Figure 7a. A limiter is necessary to handle the discontinuities. Shown in Figure 7 are the exact solution, and the computed numerical solutions with the Type 2 quadratic *SV* on the 40x40x2 irregular grid using the TVD limiter, i.e., *M* was taken to be 0. Note that the limiter produced a very good solution.

In order to estimate the numerical order of accuracy for the solution away from the discontinuities, L_1 and L_∞ errors in the smooth region [-0.2, 0.4]x[-0.2, 0.4] are computed. Computations were carried out on the irregular grid only. Without the use of the limiter, the solution quickly diverged after shock waves were developed in the solution. The parameter *M* was set to be 400 in the computation. If *M* is too small, the accuracy in the smooth region is degraded probably because limiting was carried out in the smooth region as well as near the shock. The L_1 and L_∞ errors with the bset performing *SVs* for a given order of accuracy are presented in Table 6. Obviously, with this choice of *M*, the designed order of accuracy was achieved away from discontinuities.

6. CONCLUSIONS

A high-order Spectral (Finite) Volume method has been developed for two-dimensional scalar conservation laws on unstructured triangular meshes. Each mesh cell forms a spectral volume, and the spectral volume is further partitioned into polygonal control volumes. High order schemes are then built based on the CV-averaged solutions. It was shown that a universal reconstruction can be obtained if all spectral volumes are partitioned in a similar manner [35]. However, as in the one-dimensional case, the way in which a *SV* is partitioned into *CVs* affects the convergence property of the resultant numerical scheme. A criterion based on the Lebesgue constant has been developed and used successfully to determine the quality of various partitions. Symmetric, stable, and convergent linear, quadratic and cubic *SVs* have been obtained, and many different types of partitions are evaluated based on the Lebesgue constants and their performance on model test problems.

Accuracy studies with 2D linear and non-linear scalar conservation laws have been carried out, and the order of accuracy claim has been numerically verified on both smooth and non-smooth triangular grids for convergent *SVs*. TVD and TVB limiters have been developed for non-oscillatory capturing of discontinuities, and found to perform well. The TVB limiters with a properly selected parameter (*M*) are capable of maintaining uniformly high-order accuracy away from discontinuities. The extension of the method to one and two dimensional hyperbolic systems is under way, and will be reported in future publications.

ACKNOWLEDGEMENT

The first author grateful acknowledges the start-up funding provided by the Department of Mechanical Engineering, College of Engineering of Michigan State University.

REFERENCES

1. R. Abgrall, On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation, *J. Comput. Phys*. **114**, 45-58 (1994).
2. T.J. Barth and D.C. Jespersen, The design and application of upwind schemes on unstructured meshes, AIAA Paper No. 89-0366.
3. T.J. Barth and P.O. Frederickson, High-order solution of the Euler equations on unstructured grids using quadratic reconstruction," AIAA Paper No. 90-0013, 1990.
4. F. Bassi and S. Rebay, High-order accurate discontinuous finite element solution of the 2D Euler equations, *J. Comput. Phys*. **138**, 251-285 (1997).
5. L.P. Bos, Bounding the Lebesgue functions for Lagrange interpolation in a simplex, J. Approx. Theo. **39**, 43 (1983).
6. Q. Chen and I. Babuska, Approximate optimal points for polynomial interpolation of real functions in an interval and in a triangle, Comput. Methods Appl. Mech. Engrg, 128, 405-417 (1995).

7. B. Cockburn and C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, *Mathematics of Computation* **52**, 411-435 (1989).
8. B. Cockburn, S.-Y. Lin and C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems, *J. Comput. Phys.* **84**, 90-113 (1989).
9. B. Cockburn, S. Hou and C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case, *Mathematics of Computation* **54**, 545-581 (1990).
10. P. Colella and P. Woodward, The piecewise parabolic method for gas-dynamical simulations, *J. Comput. Phys.* **54**, (1984).
11. M. Delanaye and Y. Liu, Quadratic reconstruction finite volume schemes on 3D arbitrary unstructured polyhedral grids, AIAA Paper No. 99-3259-CP, 1999.
12. S.K. Godunov, A finite-difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics, *Mat. Sb*. **47**, 271 (1959).
13. A. Harten, B. Engquist, S. Osher and S. Chakravarthy, Uniformly high order essentially non-oscillatory schemes III, *J. Comput. Phys.* **71**, 231 (1987).
14. C. Hu and C.-W. Shu, "Weighted essentially non-oscillatory schemes on triangular meshes," *J. Comput. Phys.* **150**, 97-127 (1999).
15. A. Jameson, Analysis and design of numerical schemes for gas dynamics 2: artificial diffusion and discrete shock structure, *Int. J. of Comput. Fluid Dynamics* **5**, 1-38 (1995).
16. G. Jiang and C.-W. Shu, Efficient implementation of weighted ENO schemes, *J. Comput. Phys.* **126**, 202 (1996).
17. Y. Kallinderis, A. Khawaja and H. McMorris, Hybrid prismatic/tetrahedral grid generation for complex geometries, *AIAA Journal*, **34**, 291-298 (1996).
18. D.A. Kopriva and J.H. Kolias, A conservative staggered-grid Chebyshev multidomain method for compressible flows, *J. Comput. Phys.* **125**, 244 (1996).
19. M.-S. Liou, Mass flux schemes and connection to shock instability, *J. Comput. Phys.* **160**, 623-648 (2000).
20. X.D. Liu, S. Osher and T. Chan, Weighted essentially non-oscillatory schemes, *J. Comput. Phys.* **115,** 200-212 (1994).
21. Y. Liu and M. Vinokur, Exact integration of polynomials and symmetric quadrature formulas over arbitrary polyhedral grids, *J. Comput. Phys.* **140** 122-147 (1998).
22. H. Luo, D. Sharov, J.D. Baum and R. Lohner, On the computation of compressible turbulent flows on unstructured grids, AIAA Paper No. 2000-0927, Jan. 2000.
23. D.J. Mavriplis and A. Jameson, Multigrid solution of the Navier-Stokes equations on triangular meshes, *AIAA J.* **28**, 1415-1425 (1990).
24. S. Osher, Riemann solvers, the entropy condition, and difference approximations, *SIAM J. on Numerical Analysis* **21**, 217-235 (1984).
25. A.T. Patera, A Spectral element method for fluid dynamics: laminar flow in a channel expansion, *J. Comput. Phys.* **54** 468-488 (1984).
26. P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, *J. Comput. Phys.* **43** 357-372 (1981).
27. C.-W. Shu, Total-Variation-Diminishing time discretizations, *SIAM Journal on Scientific and Statistical Computing* **9**, 1073-1084 (1988).
28. C.-W. Shu, TVB uniformly high-order schemes for conservation laws, Math. Comp. **49**, 105-121 (1987).
29. J.L. Steger and R.F. Warming, Flux vector splitting of the inviscid gasdynamics equations with application to finite difference methods, *J. Comput. Phys.* **40** 263 (1981).
30. B. van Leer, Towards the ultimate conservative difference scheme V. a second order sequel to Godunov’s method, *J. Comput. Phys*. **32,** 101-136 (1979).
31. B. van Leer, Flux-vector splitting for the Euler equations, *Lecture Notes in Physics*, **170**, 507 (1982).
32. V. Venkatakrishnan and D. J. Mavriplis, Implicit solvers for unstructured meshes, *J. Comput. Phys.* **105**, No. 1, 83-91 (1993).
33. Z.J. Wang and R.F. Chen, Anisotropic Cartesian Grid Method for Viscous Turbulent Flow, AIAA Paper No. 2000-0395, 2000.
34. Z.J. Wang, Finite spectral volume method for conservation laws on unstructured grids: basic formulation, submitted to *J. Comput. Phys.*
35. Z.J. Wang and Yen Liu, Spectral volume method for conservation laws on unstructured grids II: extension to 2D scalar equation, submitted to *J. Comput. Phys.*

Table 2. Accuracy on $u_t + u_x + u_y = 0,\ with\ u_0(x, y) = \sin\pi(x + y),\ at\ t = 1$ (regular grids)

Order of Accuracy	*Grid*	L_1 *error*	L_1 *order*	L_∞ *error*	L_∞ *order*
2 (Type 1 *SV*)	10x10x2	3.04e-2	-	4.97e-2	-
	20x20x2	7.68e-3	1.98	1.24e-2	2.00
	40x40x2	1.92e-3	2.00	3.10e-3	2.00
	80x80x2	4.81e-4	2.00	7.75e-4	2.00
	160x160x2	1.20e-4	2.00	1.93e-4	2.00
2 (Type 2 *SV*)	10x10x2	4.03e-2	-	6.68e-2	-
	20x20x2	1.06e-2	1.93	1.78e-2	1.91
	40x40x2	2.71e-3	1.97	4.54e-3	1.97
	80x80x2	6.83e-4	1.99	1.14e-3	1.99
	160x160x2	1.71e-4	2.00	2.87e-4	1.99
3 (Type 1, d = 1/3)	10x10x2	4.18e-3	-	7.76e-3	-
	20x20x2	5.33e-4	2.97	1.01e-3	2.94
	40x40x2	6.73e-5	2.99	1.25e-4	3.01
	80x80x2	8.45e-6	2.99	1.55e-5	3.01
	160x160x2	1.06e-6	2.99	1.93e-6	3.00
3 (Type 2, d = 1/4)	10x10x2	4.73e-3	-	7.88e-3	-
	20x20x2	4.77e-4	3.31	9.83e-4	3.00
	40x40x2	6.04e-5	2.98	1.23e-4	3.00
	80x80x2	7.58e-6	2.99	1.53e-5	3.01
	160x160x2	9.57e-7	2.99	1.91e-6	3.00
4 (Type 1 *SV*)	10x10x2	1.38e-4	-	4.86e-4	-
	20x20x2	8.64e-6	4.00	1.98e-5	4.62
	40x40x2	5.47e-7	3.98	1.51e-6	3.71
	80x80x2	3.46e-8	3.98	1.17e-7	3.69
	160x160x2	4.19e-8	Negative	5.15e-7	Negative
4 (Type 2, d = 1/6)	10x10x2	9.33e-5	-	3.17e-4	-
	20x20x2	5.86e-6	3.99	1.94e-5	4.03
	40x40x2	3.70e-7	3.99	1.24e-6	3.95
	80x80x2	2.32e-8	4.00	7.78e-8	3.99
	160x160x2	1.45e-9	4.00	4.84e-9	4.01
4 (Type 3, d = 1/15)	10x10x2	7.36e-5	-	2.51e-4	-
	20x20x2	4.52e-6	4.03	1.61e-5	3.96
	40x40x2	2.81e-7	4.01	1.01e-6	3.99
	80x80x2	1.75e-8	4.01	6.30e-8	4.00
	160x160x2	1.10e-9	3.99	3.94e-9	4.01

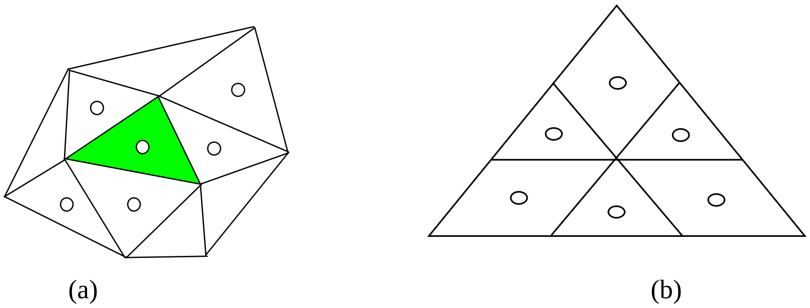


Figure 1. (a) A Possible “Stencil” of Cells Used to Build a Quadratic Reconstruction in an Unstructured Grid; (b) A Quadratic Triangular Spectral Volume

Table 3. Accuracy on $u_t + u_x + u_y = 0, \mathit{with}\ u_0(x, y) = \sin\pi(x + y), \mathit{at}\ t = 1$ (irregular grids)

Order of Accuracy	*Grid*	L_1 *error*	L_1 *order*	L_∞ *error*	L_∞ *order*
2 (Type 1 *SV*)	10x10x2	1.30e-1	-	3.60e-1	-
	20x20x2	6.66e-2	0.96	1.91e-1	0.91
	40x40x2	3.51e-2	0.92	9.84e-2	0.96
	80x80x2	1.85e-2	0.92	4.91e-2	1.00
	160x160x2	9.74e-3	0.93	2.86e-2	0.78
2 (Type 2 *SV*)	10x10x2	6.71e-2	-	1.36e-1	-
	20x20x2	1.83e-2	1.87	4.42e-2	1.62
	40x40x2	4.71e-3	1.96	1.15e-2	1.94
	80x80x2	1.19e-3	1.98	2.94e-3	1.97
	160x160x2	3.00e-4	1.99	8.85e-4	1.73
3 (Type 1, d = 1/3)	10x10x2	9.17e-3	-	3.67e-2	-
	20x20x2	1.25e-3	2.87	5.28e-3	2.80
	40x40x2	1.64e-4	2.93	8.32e-4	2.67
	80x80x2	2.15e-5	2.93	1.84e-4	2.18
	160x160x2	2.79e-6	2.95	4.05e-5	2.18
3 (Type 2, d = 1/4)	10x10x2	8.36e-3	-	3.76e-2	-
	20x20x2	1.15e-3	2.86	5.63e-3	2.74
	40x40x2	1.52e-4	2.92	1.00e-3	2.49
	80x80x2	2.01e05	2.92	2.14e-4	2.22
	160x180x2	2.64e-6	2.93	5.31e-5	2.01
4 (Type 2, d = 1/6)	10x10x2	3.04e-4	-	2.58e-3	-
	20x20x2	2.02e-5	3.91	1.73e-4	3.90
	40x40x2	1.34e-6	3.91	1.42e-5	3.61
	80x80x2	9.61e-8	3.80	1.03e-6	3.79
	160x160x2	2.30e-8	2.06	1.23e-6	Negative
4 (Type 3, d = 1/15)	10x10x2	2.71e-4	-	1.51e-3	-
	20x20x2	1.61e-5	4.07	1.14e-4	3.73
	40x40x2	9.91e-7	4.02	8.28e-6	3.78
	80x80x2	6.17e-8	4.01	5.40e-7	3.94
	160x160x2	3.87e-9	3.99	3.79e-8	3.83

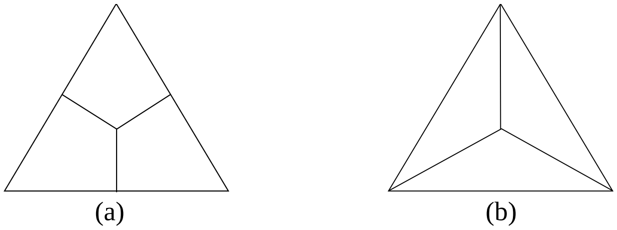
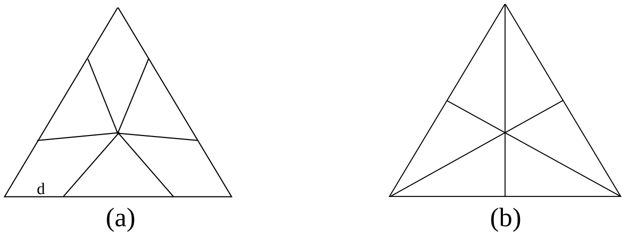

(a) (b)

Figure 2. Possible Partitions of a Linear Spectral Volume



(a) (b)

Figure 3. Possible Partitions of a Quadratic Spectral Volume

Table 4. Accuracy on $u_t + uu_x + uu_y = 0, \textit{ with } u_0(x,y) = \frac{1}{4} + \frac{1}{2}\sin\pi(x+y), \textit{ at } t = 0.1$ with irregular grid

Order of Accuracy	*Grid*	L_1 *error*	L_1 *order*	L_∞ *error*	L_∞ *order*
2 (Type 1 *SV*)	10x10x2	4.26e-3	-	4.32e-2	-
	20x20x2	1.52e-3	1.49	2.34e-2	0.88
	40x40x2	5.82e-4	1.38	1.22e-2	0.94
	80x80x2	2.42e-4	1.27	6.09e-3	1.00
	160x160x2	1.06e-4	1.19	2.99e-3	1.03
2 (Type 2 *SV*)	10x10x2	5.79e-3	-	2.96e-2	-
	20x20x2	1.46e-3	1.99	9.15e-3	1.69
	40x40x2	3.67e-4	1.99	2.87e-3	1.67
	80x80x2	9.40e-5	1.97	8.78e-4	1.71
	160x160x2	2.39e-5	1.98	3.54e-4	1.31
3 (Type 1, d = 1/3)	10x10x2	6.37e-4	-	4.71e-3	-
	20x20x2	1.21e-4	2.40	1.26e-3	1.90
	40x40x2	2.02e-5	2.58	3.52e-4	1.84
	80x80x2	3.22e-6	2.65	8.21e-5	2.10
	160x160x2	5.02e-7	2.68	1.66e-5	2.31
3 (Type 2, d = 1/4)	10x10x2	6.28e-4	-	3.93e-3	-
	20x20x2	1.17e-4	2.42	1.09e-3	1.85
	40x40x2	1.91e-5	2.61	3.05e-4	1.84
	80x80x2	3.01e-6	2.67	7.16e-5	2.09
	160x160x2	4.63e-7	2.70	1.43e-5	2.32
4 (Type 2, d = 1/6)	10x10x2	7.87e-5	-	1.02e-3	-
	20x20x2	6.07e-6	3.70	1.00e-4	3.35
	40x40x2	4.55e-7	3.74	9.62e-6	3.38
	80x80x2	3.44e-8	3.73	8.55e-7	3.49
	160x160x2	2.79e-9	3.62	8.75e-8	3.29
4 (Type 3, d = 1/15)	10x10x2	9.71e-5	-	1.29e-3	-
	20x20x2	7.17e-6	3.76	1.24e-4	3.38
	40x40x2	5.20e-7	3.79	1.07e-5	3.53
	80x80x2	3.79e-8	3.79	9.34e-7	3.52
	160x160x2	2.88e-9	3.72	8.34e-8	3.49

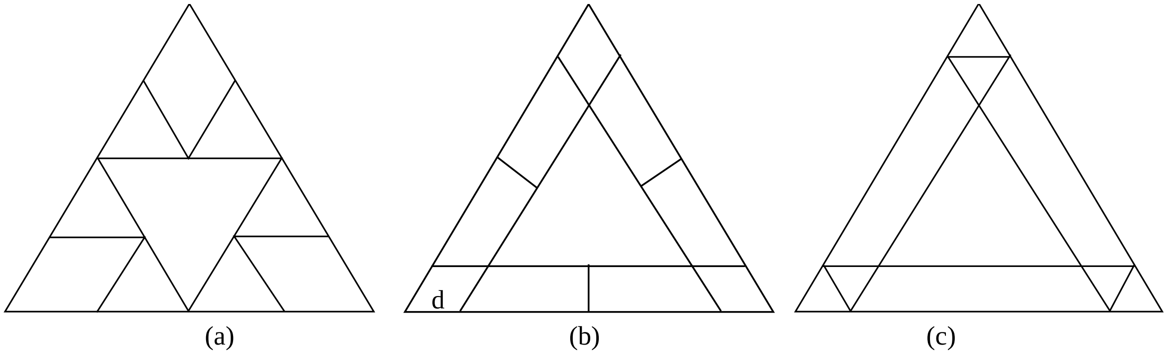
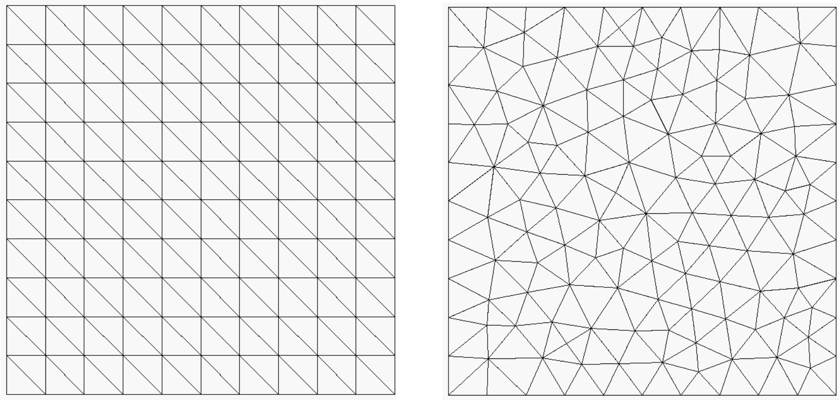


Figure 4. Possible Cubic Triangular Spectral Volumes

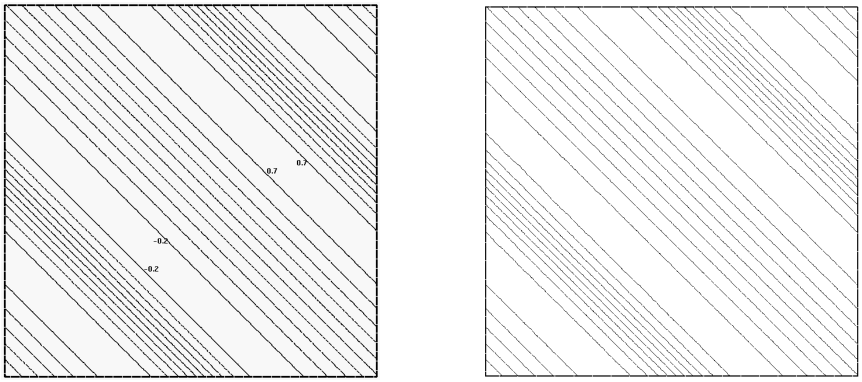
Table 5. Accuracy on $u_t + uu_x + uu_y = 0, \textit{ with } u_0(x, y) = \frac{1}{4} + \frac{1}{2}\sin\pi(x + y), \textit{ at } t = 0.45$ in [-0.2, 0.4]x[-0.2, 0.4] on irregular grid, TVB Limiter with M = 400

Order of Accuracy	*Grid*	L_1 *error*	L_1 *order*	L_∞ *error*	L_∞ *order*
2 (Type 2 *SV*)	10x10x2	1.68e-4	-	5.33e-3	-
	20x20x2	3.92e-5	2.10	1.65e-3	1.69
	40x40x2	9.66e-6	2.02	4.83e-4	1.77
	80x80x2	2.43e-6	1.99	1.58e-4	1.61
	160x160x2	6.01e-7	2.02	3.49e-5	2.18
3 (Type 2 *SV*)	10x10x2	6.23e-5	-	6.57e-3	-
	20x20x2	6.25e-6	3.32	5.86e-4	3.49
	40x40x2	6.06e-7	3.37	7.21e-5	3.02
	80x80x2	7.40e-8	3.03	1.29e-5	2.48
	160x160x2	9.47e-9	2.97	2.68e-6	2.27
4 (Type 3 *SV*)	10x10x2	7.81e-5	-	4.39e-2	-
	20x20x2	6.78e-7	6.85	1.31e-3	5.07
	40x40x2	6.38e-9	6.73	2.97e-6	8.78
	80x80x2	3.85e-10	4.05	6.65e-8	5.48
	160x160x2	2.84e-11	3.76	4.36e-9	3.93



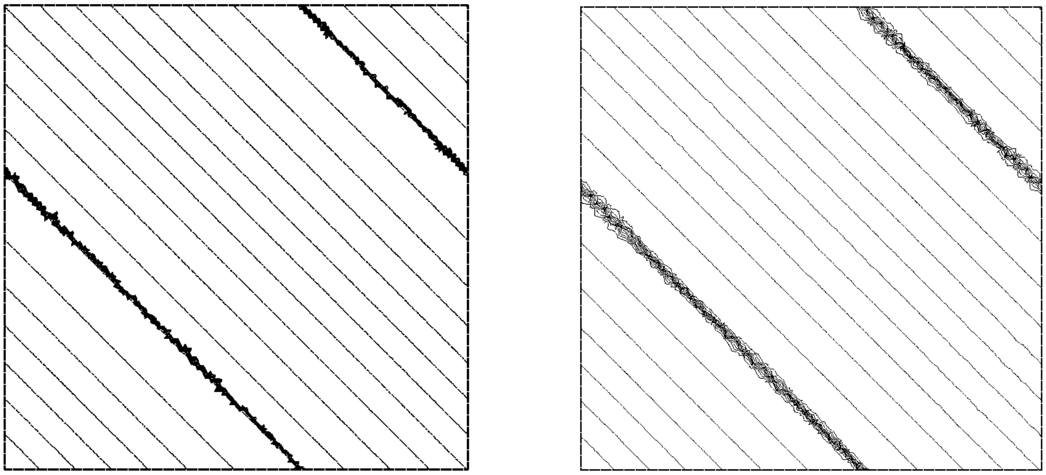
(a) (b)

Figure 5. Regular and Irregular "10x10x2" Computational Grids



(a) Exact Solution (b) Numerical Solution

Figure 6. Exact and Computational Solutions of the Burgers Equation at t = 0.1 on the 20x20x2 Irregular Mesh Using the Type 2 Quadratic Spectral Volume (Third-Order Accurate)



(a) Exact Solution (b) Numerical Solution

Figure 7. Exact and Computational Solutions of the Burgers Equation at t = 0.45 on the 40x40x2 Irregular Grid Using the Type 2 Quadratic *SV* (Third-Order Accurate)