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

Z.J. Wang (zjw@egr.msu.edu)
Department of Mechanical Engineering
Michigan State University, East Lansing, MI 48824

Yen Liu (liu@nas.nasa.gov)
MS T27B-1
NASA Ames Research Center
Moffett Field, CA 94035-1000

An Extended Abstract Submitted for
40th AIAA Aerospace Sciences Meeting and Exhibit
14-17, Jan. 2002, Reno, NV

Abstract

A time accurate, high-order, conservative, yet *efficient* method named Finite Spectral Volume (*FSV*) is 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. In addition, each spectral volume is further sub-divided into control volumes (CVs), and cell-averaged data from these control volumes is used to reconstruct a high-order 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 *FSV* method to remove/reduce spurious oscillations near discontinuities. A very desirable feature of the *FSV* method is that the reconstruction is carried out only once, and analytically, and is the same for all cells of the same type, and that the reconstruction stencil is always non-singular, in contrast to the memory and CPU-intensive reconstruction in a high-order finite volume (FV) method. Discussions are made concerning why the FSV method is significantly more efficient than high-order finite volume and the Discontinuous Galerkin (DG) methods. Fundamental properties of the *FSV* method are studied and high-order accuracy is demonstrated for several model problems with and without discontinuities.

1 Framework of the Finite Spectral Volume Method

To present the basic idea, we consider the following multi-dimensional scalar conservation laws:

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

on domain Ω with the following initial condition

$$u(x,y,0) = u_0(x,y) \tag{1b}$$

and appropriate boundary conditions on. Domain Ω is discretized into N non-overlapping cells which are called spectral volumes (*SVs*), i.e.

$$\Omega = \bigcup_{i=1}^{N} S_i \tag{2}$$

The reason why the cells are called *SVs* will be clear later. Integrating (1) on a *SV* S_i, we obtain

$$\int_{S_i} \frac{\partial u}{\partial t} dV + \oint_{\partial S_i} (F \bullet \boldsymbol{n}) dA = 0 \tag{3}$$

where $F = (f, g)$, and $\mathbf{n}$ is the unit outward normal of ∂S_i, the boundary of S_i. Define the cell averaged state variable for S_i as

$$\overline{u}_i = \frac{\int_{S_i} u dV}{V_i} \tag{4}$$

where V_i is the volume (area in 2D) of S_i. Then (3) becomes

$$\frac{d\overline{u}_i}{dt} + \frac{1}{V_i}\sum_{r=1}^{L}\int_{A_r}(F \bullet \boldsymbol{n})dA = 0 \tag{5}$$

where L is the number of faces (edges in 2D) in S_i, and A_r represents the r-th face. The surface integration on each face can be performed with a *k-th* order accurate Gauss quadrature formula, i.e.

$$\int_{A_r}(F \bullet \boldsymbol{n})dA = \sum_{j=1}^{J} w_{rj} F(u(x_{rj}, y_{rj})) \bullet \boldsymbol{n}_r A_r + O(A_r h^k) \tag{6}$$

where w_{rj} are the Gauss quadrature weights, (x_{rj}, y_{rj}) are the Gauss quadrature points, h is the maximum span of all the *SVs* in x and y directions, time t is omitted whenever there is no confusion. If $F = constant$, the following identity exists:

$$\sum_{r=1}^{L}\int_{A_r}(F \bullet \boldsymbol{n})dA = 0 \tag{7}$$

Therefore, we will gain an extra order if we sum up the surface integrals for all faces of S_i, i.e.,

$$\sum_{r=1}^{L}\int_{A_r}(F \bullet \boldsymbol{n})dA = \sum_{r=1}^{L}\sum_{j=1}^{J} w_{rj} F(u(x_{rj}, y_{rj})) \bullet \boldsymbol{n}_r A_r + O(A_r h^{k+1}) \tag{8}$$

Since $O(V_i) = O(A_r h)$, therefore we have

$$\frac{1}{V_i}\sum_{r=1}^{L}\oint_{A_r}(F \bullet \boldsymbol{n})dA = \frac{1}{V_i}\sum_{r=1}^{L}\sum_{j=1}^{J} w_{rj} F(u(x_{rj}, y_{rj})) \bullet \boldsymbol{n}_r A_r + O(h^k) \tag{9}$$

Now assume a multi-dimensional polynomial in x and y of order at most $k - 1$ exists on S_i which is a *k-th* order approximation to the state variable, i.e.,

$$p_i(x, y) = u(x, y) + O(h^k), \qquad (x, y) \in S_i \tag{10}$$

This polynomial is called a reconstruction of the state variable. With the polynomial distribution on each *SV*, the state variable is most likely discontinuous across the *SV* boundaries, unless the state variable is a polynomial of order $k - 1$ or less. Therefore, the flux integration involves two discontinuous state variables just to the left and right of a face of the *SV* boundary. This flux integration is then carried out using an exact Riemann solver or one of the Lipschitz continuous approximate Riemann solvers or flux splitting procedures, i.e.,

$$\begin{aligned} F(u(x_{rj}, y_{rj})) \bullet \boldsymbol{n}_r &= F_{Riemann}(p_i(x_{rj}, y_{rj}), p_n(x_{rj}, y_{rj}), \boldsymbol{n}_r) \\ &+ O(p_i(x_{rj}, y_{rj}) - p_n(x_{rj}, y_{rj})) \end{aligned} \tag{11}$$

Here p_n is the reconstruction polynomial of a neighboring *SV* S_n, which shares face A_r with S_i. Both p_i and p_n are *k-th* order approximations of the exact state variable, i.e.,

$$p_i(x_{rj}, y_{rj}) = u(x_{rj}, y_{rj}) + O(h^k) \tag{12a}$$

$$p_n(x_{rj}, y_{rj}) = u(x_{rj}, y_{rj}) + O(h^k) \tag{12b}$$

Therefore

$$F(u(x_{rj}, y_{rj})) \bullet \boldsymbol{n} = F_{Riemann}\ (p_i(x_{rj}, y_{rj}), p_n(x_{rj}, y_{rj}), \boldsymbol{n}_r) + O(h^k) \tag{13}$$

Substituting (13) into (6), we obtain

$$\int_{A_r} (F \bullet \boldsymbol{n}) dA = \sum_{j=1}^{J} w_{rj} F_{Riemann}(p_i(x_{rj}, y_{rj}), p_n(x_{rj}, y_{rj}), \boldsymbol{n}_r) A_r + O(A_r h^k) \tag{14}$$

Summarizing (5)-(14), we obtain the following semi-discrete, *k-th* order accurate scheme on S_i for the conservation laws (1)

$$\frac{d\bar{u}_i}{dt} + \frac{1}{V_i} \sum_{r=1}^{L} \sum_{j=1}^{J} w_{rj} F_{Riemann}(p_i(x_{rj}, y_{rj}), p_n(x_{rj}, y_{rj}), \boldsymbol{n}_r) A_r = O(h^k) \tag{15}$$

What we have done so far follows exactly the finite volume doctrine. We, however, omitted a vital detail, i.e., how we build the high-order reconstruction polynomial given just the cell-averaged state variables for the SVs. Here is where the new method departs from the traditional FV scheme. In a FV method, a stencil (a group of neighboring cells and the cell under consideration) is used to build a high-order polynomial approximation to the state variable on the cell under consideration. Depending on the order of accuracy, a very large number of (up to 60-100) cells may be necessary to perform a non-singular quadratic data reconstruction. For an unstructured grid, each cell has a unique reconstruction stencil, and the reconstruction problem needs to be solved for each and every cell at each and every time step or iteration. The reconstruction can be very memory and CPU intensive especially for higher than linear reconstructions. This is probably why we have seen few attempts to perform quadratic reconstructions in three dimensions. A recent study on a 3-D quadratic reconstruction [5] showed that the cost in memory and CPU time does not justify the effort.

In this paper, the *FSV* method is developed to address this very drawback. Our solution is as follows. Instead of using a large stencil of neighboring cells to perform the reconstruction, we subdivide the *SV* into *control volumes* (*CVs*). The order of accuracy of the reconstruction determines the number of *CVs* to be generated in each *SV*. For example, for a linear reconstruction on a triangle, the triangle is divided into at least three *CVs* as shown in Figure 1a, and cell averaged state variables are defined on the *CVs*. Figures 1b and 1c give some possible *CV* subdivisions for quadratic and cubic data reconstructions. The number of *CVs* in Figure 1 is the minimum required for these polynomial reconstructions. Other *CV* subdivisions are definitely possible.

With any of these high-order reconstructions, Eq. (15) can then be used to update the cell-averaged state variable on the *SVs*, i.e., the cell-averaged state variable $\bar{u}_i$ for S_i at a new time level $n+1$ (i.e. $\bar{u}_i^{n+1}$) can be obtained with an appropriate time integration scheme based on the solution at time level n with $t = n\Delta t$, where Δt is the time step. However, in order to use the same high-order reconstruction at time level $n+1$, it is necessary to "scatter" the update $\Delta\bar{u}_i \equiv \bar{u}_i^{n+1} - \bar{u}_i^{n}$ back to the cell-averaged state variables at all the *CVs* in S_i. This is how we perform the scattering operation. Each *CV* inside a *SV* is treated separately *as if it is independent* to update the cell-averaged state variable for the *CV*. Note that the subtle difference between a *FV* and a *FSV* 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*. To maintain a high-order accuracy, Gaussian quadrature formulas are again used, not only for the Riemann fluxes through the *SV* boundaries, but also for the fluxes through interior *CV* boundaries. The most significant advantage of the *FSV* method, as compared with the *FV* method, is that the reconstruction for a particular cell type (e.g. triangles) with a certain *CV* subdivision (e.g. those shown in Figure 1) is exactly the same. Even though the shape of the SVs may all be different, as long as they are geometric similarly subdivided, they all have the same

reconstruction (in the parameter space, to be explained in the final paper), and the weights for evaluating the state variables in term of cell-averaged unknowns at similar quadrature points are all the same. Therefore, the memory and CPU intensive reconstructions used in a *FV* method are solved analytically without taking any extra memory in the *FSV* method. Furthermore, exact fluxes rather than Riemann fluxes are used at the interior boundaries of the *CVs*, resulting again significant savings because the Riemann flux is usually several time more expensive to compute than the exact flux.

The idea can of course be easily extended to other cell types such as quadrilaterals, tetrahedra, hexahedra, prisms, etc. For cell types other than triangles and tetrahedra, it appears that symmetric *CV* subdivisions with the minimum number of *CVs* for a given order of accuracy are not easily found.

The FSV method shares many advantages with the Discontinuous Galerkin (DG) method [2-4] in that it is compact which is suitable for parallel computing, high-order accurate, conservative, and capable of handling complex geometries. Furthermore, the FSV method is expected to be much more efficient than the DG method, and has higher resolution than the DG method for discontinuities because of the availability of local cell-averaged state variables at the control volumes.

The main steps in a *k-th* order FSV method (with an order *k-1* polynomial reconstruction) are:
1. Compute the state variables at the quadrature points;
2. Use a *k-th* order accurate quadrature formula (exact for a polynomial of order *k - 1*) and a Riemann solver to compute the surface flux integrals at the spectral volume boundaries, and use a *k-th* order accurate quadrature formula for analytical fluxes for interior control volume boundaries because the state-variable is continuous across the interior CV boundaries;
3. Use a TVD Runge-Kutta scheme for time integration;

The main steps in a *k-th* order DG method are:
1. Compute the state variables at the quadrature points;
2. Use a Riemann solver and a *2k-th* order quadrature formula to compute the surface flux integrals;
3. Use a *(2k-1)-th* order quadrature formula to compute the volume integrals;
4. Left multiply the residual by the inverse of the mass matrix because the mass matrix is usually not diagonal for k > 2;
5. Use a TVD Runge-Kutta scheme for time integration;

Note that with the FSV method, the high-order volume integration required in a DG method is completely eliminated. Furthermore, the surface integral in the FSV method needs only to be *k-th* order accurate instead of the *2k-th* order accuracy required in a DG method. As a result, the FSV method requires only half the quadrature points required by a DG method to carry out the surface integration. For fourth order DG and FSV methods, there are 10 degrees of freedom (DOF) in 2D for a single variable. In a DG method, 4 quadrature points are required to compute the surface integral on a single edge to achieve the desired accuracy, and 12 quadrature points are required to compute the volume integral up to the desired accuracy [2]. To update all the DOFs (assuming a single variable) for a single element using the DG method, (3x4x10 + 12x10) = 240 variable evaluations at the quadrature points are required. In order to compute the surface integrals, 3x4x10 = 120 Riemann fluxes need to be computed. In contrast, only 2 quadrature points are necessary to compute the surface integral on a single edge to achieve the desired accuracy with a fourth-order FSV method. Therefore, to update all the DOFs for a single spectral volume (element) using the FSV method, only 2x27 = 54 (27 being the number of total edges in the spectral volume shown in Figure 1c) variable evaluations at the quadrature points are required. In addition, only 2x12 = 24 (12 being the total number of spectral volume boundary edges) Riemann fluxes are required, and the rest 2x15 = 30 (15 being the number of interior control volume boundary-edges) fluxes are analytical fluxes because the reconstructed state variable is continuous across the interior control volume boundaries inside the spectral volume. It is well known that a Riemann flux is usually several times more

expensive to compute than the analytical flux for the Euler equations. Let's assume that a Riemann flux is only three times as expensive as an analytical flux (a very conservative estimate indeed). Then the FSV method requires 24+30/3 = 34 Riemann fluxes. The third-order TVD Runge-Kutta scheme takes negligible CPU time because of the very few number of operations. Finally in a fourth DG method, the residual vector has to left-multiply by a 10x10 matrix at each iteration.

If we assume that the Riemann flux computation dominates the total CPU time, then the FSV method is about 4 times as fast as the DG method. If on the other hand, variable evaluations dominate the CPU time, the FSV method can be close to five times as fast as the DG method. Overall, we expect the fourth-order FSV method to be about 4-5 times as fast as the fourth-order DG method in 2D. If one is interested in even higher order accurate schemes, the availability of very high-order quadrature formulas may become an issue in a DG method. For example, a sixth order DG scheme necessitates a 12^{th} order quadrature formula for surface integration, and an 11^{th} order quadrature formula for volume integration. In three-dimensions, it is expected the advantage of the FSV method is even more pronounced because high-order quadrature formulas for volume integration in a tetrahedron are required in the DG method.

2. Numerical Tests

We have implemented the FSV method in both 1D and 2D, with a variety of limiters (control-volume-wise (CV-wise) and spectral-volume-wise (SV-wise) TVDM and TVBM limiters) to eliminate spurious oscillations. In the final paper, detailed formulations will be given. Here we just show several demonstration cases to demonstrate its capability.

1 Test with the Burger's Equation

In this test, we solve the non-linear Burger's equation with a periodic boundary condition:

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

$$u(x,0) = u_0(x) = 1 + \frac{1}{2}\sin(\pi x).$$

The exact solution is smooth up to $t = 2/\pi$, then it develops a moving shock, which interacts with rarefaction waves. At t = 0.3, the solution is still smooth. FSV schemes from second to sixth order of accuracy are tested, and the L_1 and L_∞ errors are listed in Table 1, together with the numerical order of accuracy. Note that the expected formal orders of accuracy for all the tested schemes are achieved in both the L_1 and L_∞ norms. The computed solution with a second-order FSV scheme on 6 SVs is compared with the solution with a fourth-order FSV scheme on 3 SVs in Figure 2. The numerical solutions therefore have the same number of degrees-of-freedom. Note that the fourth-order scheme gave a visibly better solution than the second-order scheme.

At $t = 2/\pi$, a shock starts to form in the solution. The numerical solution would be oscillatory without limiters. Figure 3 displays the computed solutions with a 4^{th}-order FSV scheme on 20 SVs using various limiters. Note that the SVTVDM limiter strongly dissipated the numerical solution, while the CVTVDM limiter gave a much better solution. Both TVBM limiters with M = 20 gave reasonable results, with the CVTVBM limiter showing a slightly more accurate prediction. The solution with the SVTVBM limiter is slightly oscillatory.

At t = 1, a shock wave has formed in the solution. The numerical solutions computed with a fourth-order FSV scheme on 20 and 40 SVs using both TVBM limiters are presented in Figure 4. Note that the shock-wave is generally captured in one spectral volume, and the CVTVBM limiter once again produced a solution with a better resolution for the shock wave.

Finally to see whether TVBM limiters affect the solution accuracy away from the shock wave, the local solution errors with a 4th-order FSV scheme on different grids are plotted in Figure 5. It is shown clearly in the Figure that the expected order of accuracy (4th-order) is retained away from the shock-wave with both the CVTVBM and SVTVBM limiters. The CVTVBM limiter is shown again to have a better resolution for the shock wave.

2. Accuracy Study with 2D Linear Wave Equation

In this case, we test the accuracy of the *FSV* method on the linear equation:

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

$$u(x, y,0) = u_0(x, y), \qquad \textit{periodic boundary condition}$$

The initial condition is $u_0(x, y) = \sin \pi(x + y)$. The numerical simulation is carried until t = 1 on a triangular grid generated from a uniform Cartesian grid by cutting each Cartesian cell into two triangles. In Table 2, we show the L_1 and L_∞ errors produced using second to fourth order *FSV* schemes with CVs shown in Figure 1. The third-order TVD Runge-Kutta time-integration scheme was used for all the computations presented here. 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. Again it is shown that the desired order of accuracy is obtained for all cases.

Conclusions

A high-order Finite Spectral Volume (*FSV*) method is developed in this study. The concept of "spectral volume" is introduced to achieve high-order accuracy in a very efficient manner. The *FSV* method is much more efficient in terms of both memory and CPU requirement than a high-order finite volume method because the reconstruction for a particular grid type is solved only once, and analytically, and is never explicitly carried out. Furthermore, the "reconstruction stencils" are never singular. We also discussed why the FSV method is significantly more efficient than the DG method. Control-volume-wise and spectral-volume-wise TVDM and TVBM limiters are developed to remove spurious oscillations near discontinuities. It has been shown that CV-wise limiters perform better than SV-wise limiters. Because of the availability of local data, the *FSV* method is expected to produce much sharper discontinuity profiles than the DG method.

Accuracy studies with 1D and 2D linear and non-linear scalar conservation laws have been carried out, and the order of accuracy claim has been numerically verified. The TVBM limiters were found to maintain uniformly high-order accuracy away from discontinuities.

References

1. 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.
2. F. Bassi and S. Rebay, High-order accurate discontinuous finite element solution of the 2D Euler equations, *J. Comput. Phys.* **138**, 251-285 (1997).
3. 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).
4. 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).
5. M. Delanaye and Y. Liu, Quadratic reconstruction finite volume schemes on 3D arbitrary unstructured polyhedral grids, AIAA Paper No. 99-3259-CP, 1999.

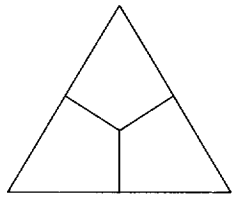
6. D.A. Kopriva, Multidomain spectral solutions of the Euler gas-dynamics equations, *J. Comput. Phys.* **96**, 428 (1991).
7. A.T. Patera, A Spectral element method for fluid dynamics: laminar flow in a channel expansion, *J. Comput. Phys.* **54** 468-488 (1984).
8. P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, *J. Comput. Phys.* **43** 357-372 (1981).

Table 1. Accuracy on $u_t + uu_x = 0$, *with* $u_0(x) = 1 + 0.5\sin(\pi x)$ at t = 0.3

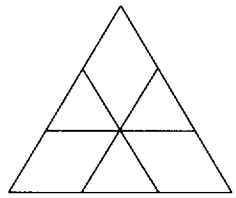
Order of Accuracy	*NDOF*	L_∞ *error*	L_∞ *order*	L_1 *error*	L_1 *order*
2	20	2.74e-2		1.16e-2	
	40	9.97e-3	1.46	3.07e-3	1.92
	80	3.09e-3	1.69	7.82e-4	1.97
	160	8.11e-4	1.93	1.95e-4	2.00
	320	2.08e-4	1.96	4.88e-5	2.00
	640	5.24e-5	1.99	1.22e-5	2.00
3	30	1.17e-2	-	1.44e-3	-
	60	2.11e-3	2.47	1.92e-4	2.91
	120	3.90e-4	2.44	2.70e-5	2.83
	240	5.77e-5	2.76	3.66e-6	2.88
	480	7.72e-6	2.90	4.80e-7	2.93
	960	9.91e-7	2.96	6.15e-8	2.96
4	20	1.64e-2	-	2.90e-3	-
	40	7.81e-4	4.39	1.09e-4	4.73
	80	2.59e-4	1.59	1.25e-5	3.12
	160	1.67e-5	3.96	7.17e-7	4.12
	320	1.11e-6	3.91	4.35e-8	4.04
	640	7.01e-8	3.99	2.71e-9	4.00
5	20	1.45e-2	-	2.09e-3	-
	40	8.15e-4	4.15	6.45e-5	5.02
	80	9.10e-5	3.16	3.45e-6	4.22
	160	4.56e-6	4.32	9.54e-8	5.18
	320	1.89e-7	4.59	3.78e-9	4.66
	640	6.73e-9	4.81	1.32e-10	4.84
6	30	2.96e-3	-	2.74e-4	-
	60	5.97e-5	5.63	3.66e-6	6.23
	120	5.48e-6	3.45	1.45e-7	4.66
	240	8.76e-8	5.97	1.77e-9	6.36
	480	1.55e-9	5.82	2.53e-11	6.13

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

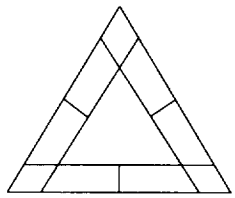
Order of Accuracy	*Grid*	L_1 *error*	L_1 *order*	L_∞ *error*	L_∞ *order*
2	10x10x2	1.64e-2	-	4.13e-2	-
	20x20x2	4.01e-3	2.03	9.59e-3	2.11
	40x40x2	9.85e-4	2.03	2.21e-3	2.18
	80x80x2	2.44e-4	2.01	5.18e-4	2.09
	160x160x2	6.09e-5	2.00	1.24e-4	2.06
3	10x10x2	4.18e-3	-	7.76e-3	-
	20x20x2	5.33e-4	2.97	1.01e-3	2.94
	40x40x2	6.70e-5	2.99	1.24e-4	3.03
	80x80x2	8.13e-6	3.04	1.51e-5	3.04
	160x160x2	1.05e-6	2.95	1.93e-6	2.97
4	10x10x2	9.33e-5	-	3.17e-4	-
	20x20x2	5.86e-6	3.99	1.94e-5	4.02
	40x40x2	3.70e-7	3.99	1.24e-6	3.97
	80x80x2	2.32e-8	4.00	7.78e-8	3.99
	160x160x2	1.45e-9	4.00	4.84e-9	4.01



(a)



(b)



(c)

Figure 1. Control Volumes in a Triangular Spectral Volume (a) Linear; (b) Quadratic; (c) Cubic.

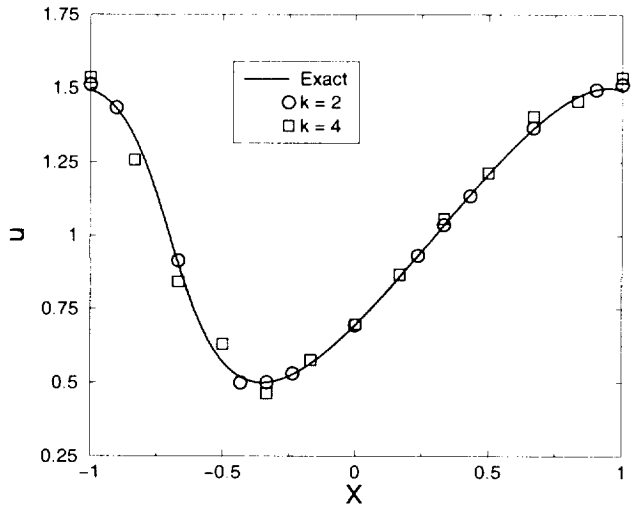


Figure 2. Computed Solutions to the Burger's Equation at t = 0.3 Using a Second-Order and Fourth Order FSV Schemes with 12 Degrees-of-Freedom without Limiters

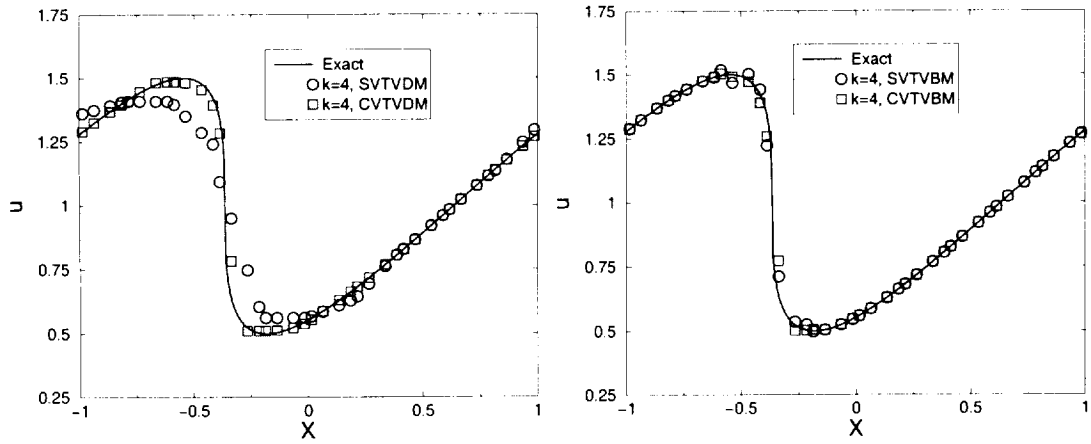


Figure 3. Computed Solutions to the Burger's Equation at t = $2/\pi$ Using a Fourth Order FSV Scheme with CVTVDM, SVTVDM, and CVTVBM, SVTVBM Limiters on 10 SVs

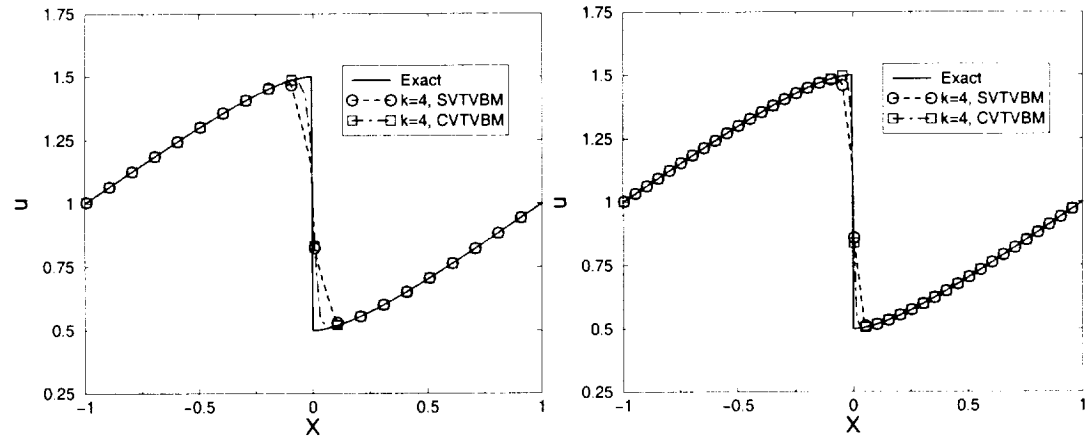


Figure 4. Computed Solutions to the Burger's Equations at t = 1 with CVTVBM and SVTVBM Limiters Using 20 and 40 Spectral Volumes

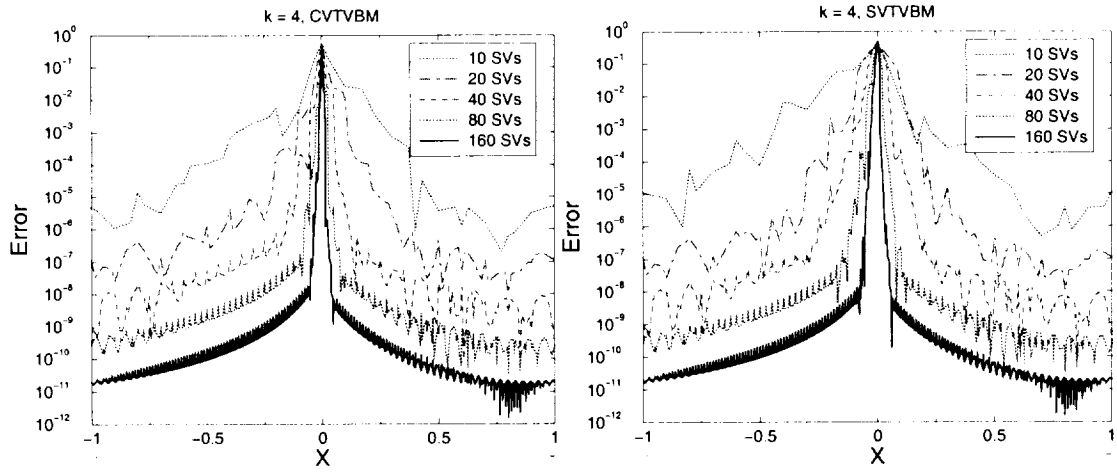


Figure 5. Local Error of Computed Solutions of the Burger's Equation at t = 1 with a Fourth-Order FSV scheme and CVTVBM and SVTVBM Limiters