Improved scalar transport for unstructured finite volume methods using simplex superposition

By F. Ham

1. Motivation and objectives

The application of second-order finite volume methods to large eddy simulation (LES) is becoming increasingly common. Finite volume methods support the complex geometries associated with most problems of engineering interest, and are also relatively efficient. When finite volume methods are applied to LES, it is common to use symmetric (central difference) discretizations of the convective term in the momentum equations. This reduces the numerical dissipation of the scheme and, for certain discretization choices, leads to a discrete equivalent of kinetic energy conservation in the inviscid, incompressible limit (Mahesh *et al.* 2004; Ham & Iaccarino 2004). When additional scalars are required (for energy, temperature, species transport, etc.), however, the analogous scalar energy conservation (in the limit of zero diffusion) is normally disregarded, and the associated convective terms are discretized with much more dissipative upwind approaches. This is true even for finite volume methods on high quality structured grids – see for example Pierce & Moin (2004). The reason for this discrepency is that the large dispersion errors associated with second-order central difference scalar transport lead to unacceptable scalar oscillations, often outside of their physical bounds – e.g. negative mixture fractions. Present sub-grid scale models for turbulent diffusion appear unable to address this problem adequately, and dissipative convective discretizations will likely be the norm until models can be improved or dispersion errors reduced.

In the present brief, we develop a solution to this problem by presenting a second-order finite volume scheme for general polyhedral grids with significantly reduced dispersion errors. This reduction in error is achieved, in part, by discretizing the time term using a consistent, non-diagonal mass matrix, similar to that used by the Galerkin finite element method with linear elements. The details of the spatial discretization and the production of an appropriate consistent mass matrix for grids of arbitrary polyhedra can be accomplished using the approach of "simplex superposition" described in this brief.

2. Background

It is generally accepted that discretization schemes suitable for LES must have relatively low numerical dissipation. One way to achieve a stable low-dissipation scheme is to discretize the governing equations using summation-by-parts (SBP) operators (Kreiss & Scherer 1974). SBP operators are sometimes referred to as mimetic operators because they mimic certain properties of vector calculus – specifically integration-by-parts. When combined with a suitable treatment of the boundary conditions (e.g. the simultaneous approximation term or SAT approach of Carpenter *et al.* (1994)) it can be proven that SBP/SAT discretizations are energy stable.

In the realm of second-order finite volume methods on unstructured grids, symmetric edge-based spatial discretizations are in fact SBP (Nordstrom *et al.* 2003). In edge-based

methods, unknowns are stored at the nodes of the mesh, and fluxes are discretized along the edges that connect two nodes. These simple, efficient discretizations are also (perhaps somewhat surprisingly) consistent on grids based on simplex elements (triangles in 2D, tetrahedra in 3D), due to an equivalence with the Galerkin finite element discretization using linear elements. This equivalence was realized and reported by Jameson *et al.* (1986), Rostand & Stoufflet (1988) as well as Barth (1991), and used to develop very efficient schemes with relatively sound mathematical foundations. Symmetric edge-based schemes are currently being developed by ours and other research groups for LES of compressible flows, where the stability derived from the mimetic properties of the operators is particularly crucial – see for example Georges *et al.* (2008). Unfortunately, however, their suitability for LES on general unstructured grids is severely hampered by two very fundamental problems:

- The equivalence to Galerkin finite elements is only present for grids based on simplex elements. For grids involving hexahedral or other more complex polyhedral elements, edge-based discretizations lose this equivalence and in many cases are no longer even consistent (i.e. do not converge with refinement).

- It is not possible to pack 3-dimensional space with arrays of self-similar tetrahedra.

The first of these problems explains why any uninformed effort to transition an edge-based solver from simplex grids to general unstructured grids will likely fail, due either to unacceptable accuracy and/or numerical instability. The second problem is an unfortunate consequence of three-dimensional geometry. While simplex elements are very convenient and flexible for building body-fitted grids around complex three-dimensional geometries, the fact that it is impossible to produce grids of isotropic elements in large regions of the volume will make it extremely challenging and expensive to minimize the grid-related errors present in LES. Systematic studies of the effect of unstructured grid quality on LES are not very common, perhaps because they are so depressing for practitioners of the unstructured schemes – see for example Ham *et al.* (2006) and Moreau *et al.* (2006). The basic conclusion is that grids with mainly hexahedral elements and with as much uniformity as possible produce the best solutions.

The stability, accuracy, and efficiency of the edge-based schemes on simplex elements are highly desirable, however, we need to devise a way to use such schemes on hex-dominant grids. One possibility is to build a simplex grid that is based on a hex-dominant grid where every non-simplex element in the parent grid has been subdivided into tetrahedra. This approach is at the heart of a mesh generation technique called octree-based mesh generation (Yerry & Shephard 1984). In two dimensions, it is equivalent to introducing an additional edge along the diagonal of each quadrilateral. Because a quadrilateral element has two diagonals, one must decide in which direction to introduce the edge. This decision will introduce a preferential connection along one diagonal direction and not the other, thus introducing a bias in the grid that may affect the solution. This bias can be particularly problematic in boundary layer elements with large aspect ratio.

An alternative to making this decision (on which diagonal to choose) is to simply not decide and choose both, weighting each choice by a factor of $1/2$ – see figure 1. This is in fact a valid choice because any discretization of a PDE will generally be linear in the geometry, allowing the grid to be represented by an appropriately weighted superposition. We call this approach "simplex superposition". In the following sections it is used to develop stable and accurate finite volume discretizations on unstructured grids of arbitrary polyhedra. As will be shown below, by avoiding any directional bias in building the

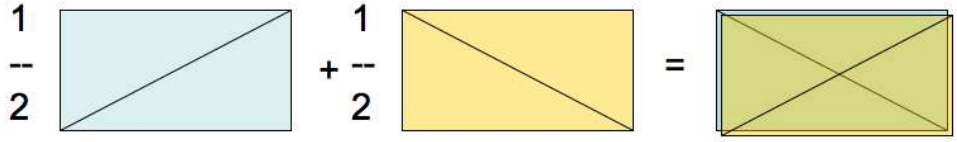

FIGURE 1. A valid two-dimensional "simplex superposition" for a quadrilateral element involving four triangles, each with weight 1/2.

underlying simplex grid, the accuracy of the resulting operators is improved significantly. In addition, because the underlying grid is based entirely on simplex elements, we can exploit the equivalence with Galerkin FEM to include a consistent positive definite mass matrix in a time dependent formulation, and dramatically reduce the dispersion errors associated with the scheme.

3. Simplex superposition

The problem of subdividing a grid consisting of arbitrary polyhedra into a valid superposition of simplex elements is subject to three requirements:

- Partition of unity requirement: all simplex weights must be positive, and the sum of weights associated with all simplex elements containing any point in the volume must be unity.
- Positive volume requirement: all individual tetrahedra must have positive volume.
- Face compatibility requirement: the sum of weights associated with the simplex elements sharing opposite sides of the same face (*e.g.* a triangle in 3D) must be equal.

The first and second of these requirements are the most obvious, ensuring that volume integration on the grid is consistent. The requirement for positive weights and volumes also ensures that the matrix properties of discrete operators based on traditional simplex elements (where all the weights are unity) are reproduced on grids based on simplex superposition (*e.g.* positive definite, symmetric or skew-symmetric, etc.) and thus the SBP properties are preserved. The third compatibility requirement ensures that the superposed grid satisfies the geometric conservation law at each node, that is $\sum_e \hat{n}_{i,e} A_e = 0$, where $\hat{n}_{i,e}$ are the edge outward unit normals and A_e the edge areas.

With these requirements stated, it is then possible to reduce the problem of building a globally valid simplex superposition to the more tractable problem of building a locally valid superposition in each element (by satisfying these three requirements at the element level), and constraining the triangle weights on the shared element faces to be the same. For the standard element types (tetrahedron, pyramid, prism, hexahedron) where faces are either triangles or quadrilaterals, this is relatively straightforward. The following subsections provide a valid (although not unique) superposition for each of these element types. These are excerpts directly from the code used to compute the examples given in this brief. Each row i of the `node[i][j]` matrix defines the 4 nodes of a right-handed tetrahedron based on the numbering provided in the accompanying figure, and `wgt[i]` is the weight of each tetrahedron in the simplex superposition.

3.1. *Hexahedron: based on 10 tetrahedra*

```
//
//        7------6
//       /      /|
//      4------5 |
//      | 3    | 2
//      |      |/
//      0------1
//

ntet = 10;

node[0][0] = 2; node[0][1] = 0; node[0][2] = 7; node[0][3] = 3;
node[1][0] = 0; node[1][1] = 2; node[1][2] = 5; node[1][3] = 1;
node[2][0] = 2; node[2][1] = 7; node[2][2] = 5; node[2][3] = 6;
node[3][0] = 0; node[3][1] = 7; node[3][2] = 5; node[3][3] = 2;
node[4][0] = 5; node[4][1] = 4; node[4][2] = 7; node[4][3] = 0;
node[5][0] = 7; node[5][1] = 3; node[5][2] = 6; node[5][3] = 4;
node[6][0] = 1; node[6][1] = 5; node[6][2] = 6; node[6][3] = 4;
node[7][0] = 0; node[7][1] = 1; node[7][2] = 3; node[7][3] = 4;
node[8][0] = 2; node[8][1] = 3; node[8][2] = 1; node[8][3] = 6;
node[9][0] = 3; node[9][1] = 4; node[9][2] = 1; node[9][3] = 6;

wgt[0] = 0.5;
wgt[1] = 0.5;
wgt[2] = 0.5;
wgt[3] = 0.5;
wgt[4] = 0.5;
wgt[5] = 0.5;
wgt[6] = 0.5;
wgt[7] = 0.5;
wgt[8] = 0.5;
wgt[9] = 0.5;
```

3.2. *Prism: based on 12 tetrahedra*

```
//           __
//       ___/  5
//    __/     /|
//   3-------4 |
//   |       | 2
//   |       |/
//   0-------1
//

ntet = 12;

node[0][0]  = 0; node[0][1]  = 5; node[0][2]  = 1; node[0][3]  = 2;
node[1][0]  = 1; node[1][1]  = 3; node[1][2]  = 2; node[1][3]  = 0;
node[2][0]  = 2; node[2][1]  = 4; node[2][2]  = 0; node[2][3]  = 1;
node[3][0]  = 4; node[3][1]  = 0; node[3][2]  = 3; node[3][3]  = 5;
node[4][0]  = 4; node[4][1]  = 1; node[4][2]  = 3; node[4][3]  = 5;
node[5][0]  = 4; node[5][1]  = 2; node[5][2]  = 3; node[5][3]  = 5;
node[6][0]  = 0; node[6][1]  = 5; node[6][2]  = 4; node[6][3]  = 1;
node[7][0]  = 1; node[7][1]  = 3; node[7][2]  = 5; node[7][3]  = 2;
node[8][0]  = 2; node[8][1]  = 4; node[8][2]  = 3; node[8][3]  = 0;
node[9][0]  = 3; node[9][1]  = 5; node[9][2]  = 1; node[9][3]  = 0;
```

```
node[10][0] = 4; node[10][1] = 3; node[10][2] = 2; node[10][3] = 1;
node[11][0] = 5; node[11][1] = 4; node[11][2] = 0; node[11][3] = 2;

wgt[0]  = 0.33333333333333;
wgt[1]  = 0.33333333333333;
wgt[2]  = 0.33333333333333;
wgt[3]  = 0.33333333333333;
wgt[4]  = 0.33333333333333;
wgt[5]  = 0.33333333333333;
wgt[6]  = 0.16666666666667;
wgt[7]  = 0.16666666666667;
wgt[8]  = 0.16666666666667;
wgt[9]  = 0.16666666666667;
wgt[10] = 0.16666666666667;
wgt[11] = 0.16666666666667;
```

3.3. *Pyramid: based on 4 tetrahedra*

```
//       4_
//      / \\_
//     / 3 \ \_2
//    /     \ /
//   0-------1

ntet = 4;

node[0][0] = 0; node[0][1] = 1; node[0][2] = 2; node[0][3] = 4;
node[1][0] = 1; node[1][1] = 2; node[1][2] = 3; node[1][3] = 4;
node[2][0] = 2; node[2][1] = 3; node[2][2] = 0; node[2][3] = 4;
node[3][0] = 3; node[3][1] = 0; node[3][2] = 1; node[3][3] = 4;

wgt[0] = 0.5;
wgt[1] = 0.5;
wgt[2] = 0.5;
wgt[3] = 0.5;
```

Any tetrahedral elements in the grid are of course already simplex elements, and are given a unit weight. Using this recipe, a simplex superposition can be computed from basically any three dimensional grid of reasonable quality primitive elements. For a fixed grid, this can be done once as a preprocessing step at the start of the simulation, leading to a list of tetrahedra and associated weights that fully describe the grid and facilitates the discretization of all operators.

3.4. *Support for hanging-node grids*

One of the benefits of developing a solver based on simplex superposition is that grids of arbitrary polyhedra can be accommodated without affecting the structure of the solver so long as a valid local superposition can be built. For example, consider the 2-dimensional elements and possible superposition shown in figure 2. Elements such as this occur in grids where local grid refinement is used. During the preprocessing stage, these transition elements can be identified and special rules applied to build the associated superposition. Once the weighted list of tetrahedra is constructed, however, the discretization and solver can be applied without any special knowledge that the grid contains these non-standard elements.

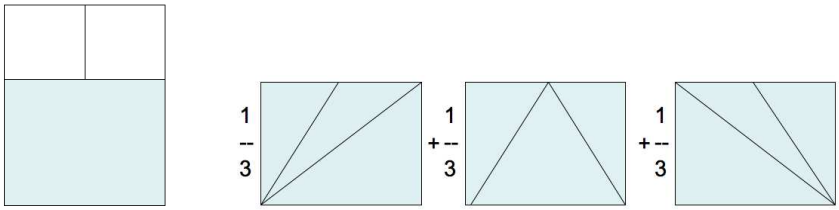

FIGURE 2. A valid two-dimensional "simplex superposition" for an element with a hanging node involving seven unique triangles.

3.5. *Mass matrix*

As mentioned previously, in additional to the SBP properties of the spatial operators built from discretizations based on simplex superposition, there is also the option to include a mass matrix for discretizing time-dependent problems and potentially reduce the dispersion error present in the scheme. Non-diagonal mass matrices are rarely used in finite volume methods. The reasons for this are likely some combination of:

- It is difficult (or impossible) to construct a symmetric positive-definite mass matrix consistent with the finite volume surrounding a node for grids based on arbitrary polyhedra. The symmetric positive definite property is required to form a valid norm and show stability of the resulting scheme.

- A non-diagonal mass matrix introduces neighbor coupling that necessitates the use of an implicit solver to advance each time step (or sub-step), even if an explicit spatial discretization is employed.

A notable exception to this trend in the finite volume community is the recent work of Georges *et al.* (2008) on simplex grids. On simplex grids, the non-diagonal mass matrix evaluated with linear Galerkin approximation is in fact consistent with the finite volume. In their paper, numerical evidence was provided in the context of compressible LES that solution accuracy was improved significantly when a mass matrix was employed, although no analysis was performed.

Similarly, an advantage of the present discretizations based on simplex superposition is that a symmetric positive definite mass matrix can readily be constructed that is consistent with the finite volume method due to the exclusive use of simplex elements. The mass matrix based on simplex superposition, however, will also have less directional bias than mass matrices based on standard simplex grids where element uniformity is not possible.

4. Results: scalar advection

To shed some light on why non-diagonal mass matrices improve dispersion errors, consider the propagation of a Gaussian bump at speed c in a 1D periodic domain. The governing equation is

$$\frac{\partial \phi}{\partial t} + c\,\frac{\partial \phi}{\partial x} = 0, \tag{4.1}$$

which we discretize in space using second-order central difference on a uniform 1D grid and introduce a potentially non-diagonal mass matrix for the time term parameterized by α as follows:

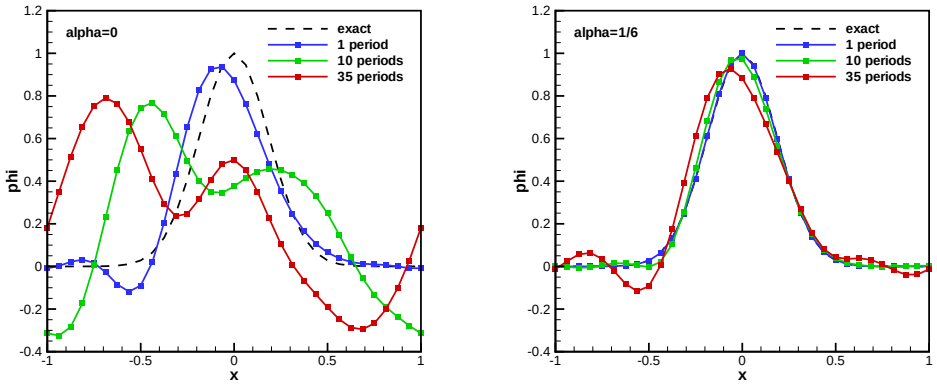

FIGURE 3. Results for advection of a 1D Gaussian bump in a periodic domain with 33 nodes (32 segments). The initial condition is $\phi\,(x, t_0) = e^{-x^2/0.07}$.

$$\alpha \left.\frac{\partial \phi}{\partial t}\right|_{j-1} + (1 - 2\,\alpha) \left.\frac{\partial \phi}{\partial t}\right|_{j} + \alpha \left.\frac{\partial \phi}{\partial t}\right|_{j+1} + c\,\frac{\phi_{j+1} - \phi_{j-1}}{2\Delta x} = 0. \tag{4.2}$$

Here the subscripts represent the grid coordinate locations. By selecting $\alpha = 0$ we recover the standard second-order finite difference or (equivalently) finite volume discretization with diagonal time term. Selecting $\alpha > 0$ introduces off-diagonal immediate neighbors into the time term. For $\alpha = 1/6$ we get the mass matrix of Galerkin FEM with linear elements. Figure 4 compares the computed solution for the cases $\alpha = 0$ and $\alpha = 1/6$ after 1, 10, and 35 periods. The time step was reduced and/or time integration modified for all cases to ensure that time integration errors were not affecting the conclusions. In both cases, scalar energy is discretely conserved in the norm $\phi^T M \phi$ where ϕ is the vector of unknowns and M is the mass matrix. Clearly the scheme with the non-diagonal mass matrix is much better at preserving the original Gaussian shape and preventing the non-physical undershoot of ϕ below zero.

4.1. *Modified wave number*

Modified wave number analysis can be used to investigate the accuracy of the discrete spatial derivatives on the grid. Because the present scheme includes a spatial modification to the time term, the usual analysis must be modified slightly as follows. Assume the function $\phi\,(x, t)$ can be written

$$\phi\,(x, t) = \psi\,(t)\, e^{i\,k\,x} \tag{4.3}$$

where $i = \sqrt{-1}$. Substituting into the governing wave equation 4.1 and differentiating exactly gives:

$$\frac{\partial \psi\,(t)}{\partial t} + c\,i\,k\,\psi\,(t) = 0 \tag{4.4}$$

Substitution of our assumed expression for ϕ into the semi-discretization given in equation 4.2 yields:

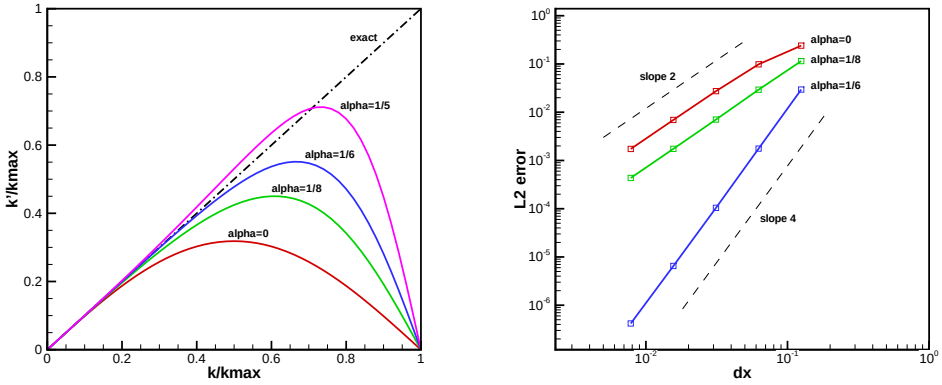

FIGURE 4. Left) Modified wave number for second-order central difference with α-weighted time term. Wavenumbers are plotted as ratios of $k_{max} = \pi/\Delta x$. Right) grid convergence study for 1D transport of a Gaussian bump.

$$\frac{\partial \psi(t)}{\partial t} + c\, i\, k'\, \psi(t) = 0 \tag{4.5}$$

where k' is the modified wavenumber:

$$k' = k\, \frac{sin(k\Delta x)}{(2\,\alpha\, cos(k\Delta x) - 2\,\alpha + 1)\, k\Delta x} \tag{4.6}$$

Note that when $\alpha = 0$, the well-known expression for second-order central difference is recovered:

$$k' = k\, \frac{sin(k\Delta x)}{k\Delta x} \tag{4.7}$$

Figure 4.1 (left) plots equation 4.6 for several different choices of α. Figure 4.1 (right) shows the computed L^2-error for the Gaussian bump transport problem as a function of grid spacing. Interestingly, for the choice $\alpha = 1/6$, the scheme is superconvergent at 4^{th} order.

4.2. *Equivalence to 4^{th}-order Pade scheme*

For the choice $\alpha = 1/6$, the expression for modified wave number is exactly the same as the standard $4th$-order Pade scheme (Lele 1992). In fact, for this problem of constant linear advection, the two schemes are basically equivalent. In the standard 4^{th}-order Pade scheme, discrete approximations to the spatial derivatives at each grid point $\delta\phi/\delta x$ are first computed by solving the following implicit system:

$$\frac{1}{6}\,\frac{\delta\phi}{\delta x}\bigg|_{j-1} + \frac{2}{3}\,\frac{\delta\phi}{\delta x}\bigg|_{j} + \frac{1}{6}\,\frac{\delta\phi}{\delta x}\bigg|_{j+1} = \frac{\phi_{j+1} - \phi_{j-1}}{2\,\Delta x}. \tag{4.8}$$

In a second step, the computed gradients are used as part of the explicit time advancement of the governing equation. For the case of constant advection speed c, substitution of the governing equation 4.1 into 4.8 gives the present 1D scheme, and the two are equivalent (apart from the differences in the time integration).

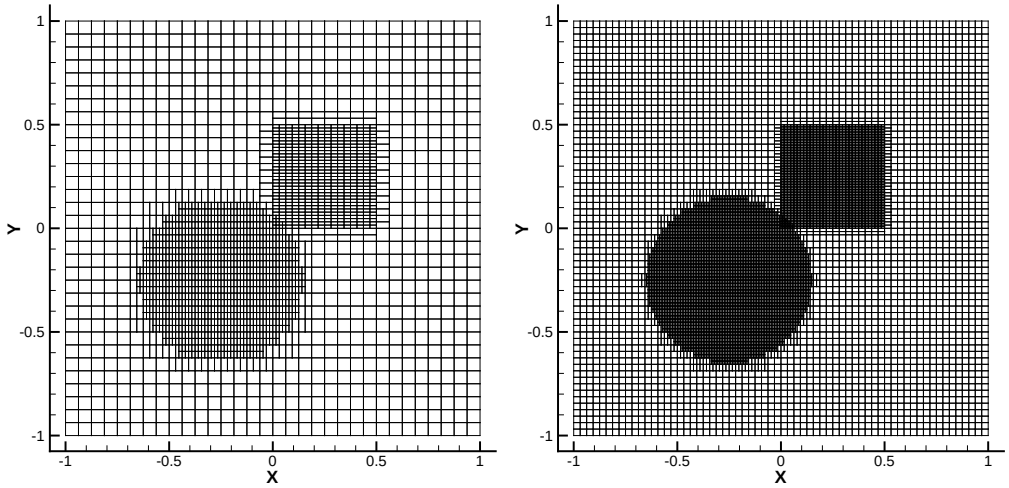

FIGURE 5. Sample grids with anisotropic local grid refinement: left) coarse grid 32×32 with local refinement; right) medium grid 64×64 with local grid refinement. Finest grid with local grid refinement is not shown: 128×128

Numerical experiments in 2D and 3D indicate that this 4^{th}-order behavior is retained for constant linear advection on uniform Cartesian grids. On general non-uniform or unstructured grids, however, the scheme converges at 2^{nd}-order with error levels approximately one order of magnitude below the same scheme with a diagonal time term.

5. Results: compressible flow on arbitrary grids

To illustrate the accuracy and flexibility of the method of simplex superposition, and to highlight the problems associated with traditional edge-based discretizations on non-simplex grids, we solve the compressible flow of an Euler vortex on a hierarchy of both uniform Cartesian grids and Cartesian grids with regions of local grid refinement – see figure 5. The discretization of the Euler flux along each edge of the parent grid (for the case of the edge-based scheme), or along each edge of the simplex superposition was based on the kinetic-energy conserving discretization described by Jameson (2008). Time integration was based on the 3^{rd}-order Runge-Kutta method of Shu & Osher (1988). For the simplex superposition with mass matrix, the implicit solve required in each sub-step was performed using the conjugate-gradient method with diagonal preconditioning.

The initial condition for the isentropic Euler vortex was:

$$\rho = \rho_0 \, (1 - c \, e^f)^{1/(\gamma-1)} \tag{5.1}$$

$$u = U_0 \, (cos \, \theta - \epsilon \, (y - y_0) \, e^{f/2}) \tag{5.2}$$

$$v = U_0 \, (sin \, \theta - \epsilon \, (x - x_0) \, e^{f/2}) \tag{5.3}$$

$$p = p_0 \, (1 - c \, e^f)^{\gamma/(\gamma-1)} \tag{5.4}$$

Where the coefficients $c = \epsilon^2 \, Ma_0^2 \, (\gamma - 1)/2$, $f = 1 - (x - x_0)^2 - (y - y_0)^2$. The vortex strength parameter ϵ was selected to be small to make the problem challenging: $\epsilon = 0.08$. Ma_0 is related to the other conditions through $Ma_0 = U_0/\sqrt{\gamma p_0/\rho_0}$. To eliminate any effect of boundary conditions on our conclusions, the problem was solved with periodic

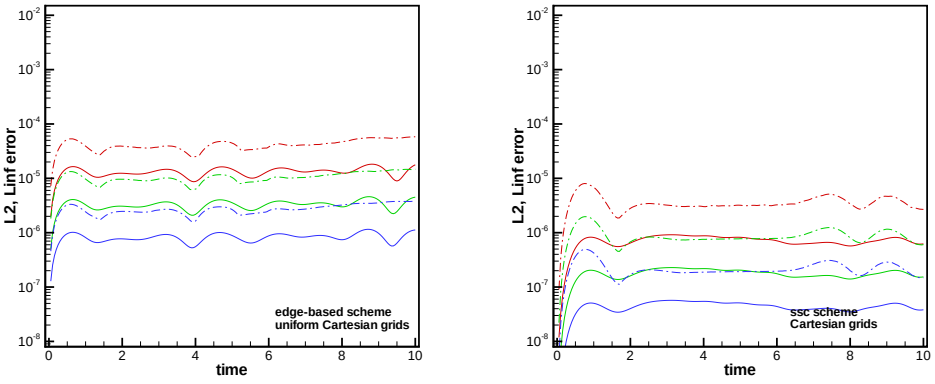

FIGURE 6. Time history of error levels in computed density on heirarchy of uniform Cartesian grids: left) edge-based scheme; right) Simplex Superposition scheme. Solid lines are L_2 error in density on 3 grids, 64×64, 128×128, and 256×256. Dashed lines are L_∞ errors in density for the same cases.

boundary conditions in x and y, with the grid size set $-5 \leq x, y \leq 5$. Other parameters were selected as follows: $\rho_0 = 1$, $p_0 = 1$, $\gamma = 1.4$, $U_0 = 1/2$, $\theta = \pi/3$. The exact solution to this problem on an infinite domain involves the vortex propagating unchanged at speed U_0 and angle θ.

Figure 6 compares the computed L_2 and L_∞ errors in density on uniform Cartesian grids. Both schemes exhibit the expected 2^{nd}-order convergence, with the absolute level of error approximately one order of magnitude lower with the discretization based on simplex superposition.

Figure 7 compares the computed L_2 and L_∞ errors in density on the hierarchy of Cartesian grids with local grid refinement. In this case the standard edge-based scheme is not converging in L_∞, indicative of the inconsistency of edge-based discretizations on poor-quality non-simplex grids. Figure 8 compares the computed density at time $t = 10$ for the two schemes. In the edge-based method, errors develop at the transition elements and rapidly contaminate the entire solution. No such behavior is seen in the discretization based on simplex superposition.

6. Future work

A discretization technique called simplex superposition has been described that results in edge-based finite volume operators that are both accurate and SBP on grids of arbitrary polyhedral elements. The method also provides a consistent, symmetric positive definite mass matrix that has been shown to significantly reduce dispersion errors for certain time dependent problems. The method is now being applied to LES of both incompressible and compressible flows to improve the spectral resolution of scalar transport problems.

Acknowledgments

This work was supported by the Advanced Simulation and Computing program of the United States Department of Energy.

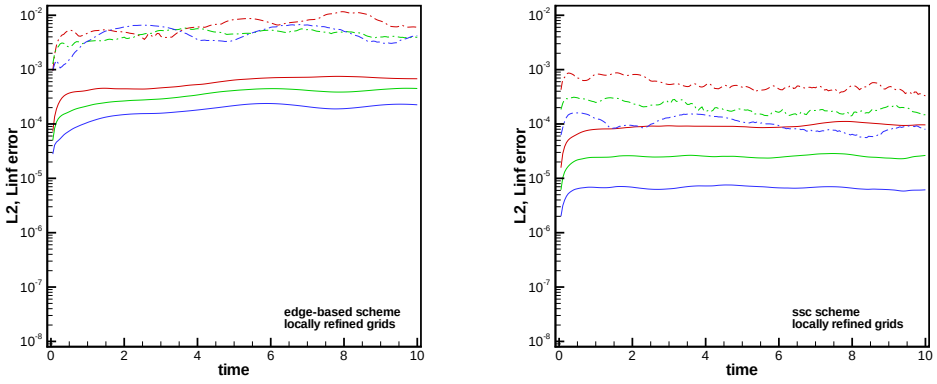

FIGURE 7. Time history of error levels in computed density on heirarchy of uniform Cartesian grids: left) edge-based scheme; right) Simplex Superposition scheme. Solid lines are L_2 error in density on 3 grids, 64×64, 128×128, and 256×256. Dashed lines are L_∞ errors in density for the same cases.

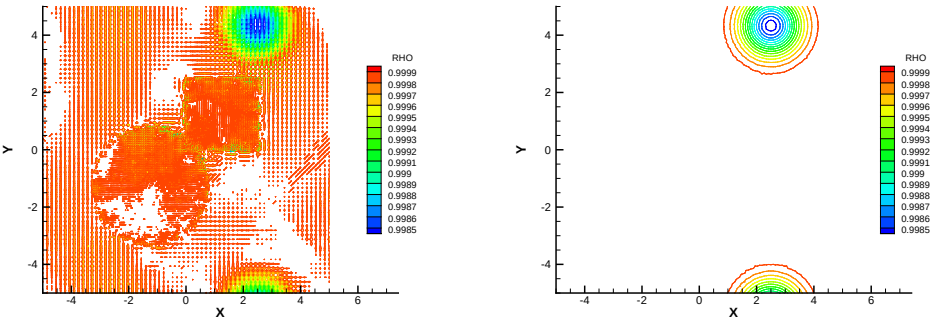

FIGURE 8. Comparison of computed density at $t = 10$: left) edge-based scheme on finest grid with local grid refinement; right) simplex superposition scheme on the same grid.

REFERENCES

BARTH, T. J. 1991 Numerical aspects of computing viscous high reynolds number flows on unstructured meshes. In *AIAA Paper 91-0721*.

CARPENTER, M. H., GOTTLIEB, D. & ABARBANEL, S. 1994 Time-stable boundary conditions for finite-difference schemes solving hyperbolic systems: Methodology and application to high-order compact schemes. *J. Comput. Phys.* **111(2)**.

GEORGES, L., WINCKELMANS, G. & GEUZAINE, P. 2008 Improving shock-free compressible RANS solvers for LES on unstructured meshes. *Journal of Computational and Applied Mathematics* **215**, 419 – 428.

HAM, F. & IACCARINO, G. 2004 Energy conservation in collocated discretization schemes on unstructured meshes. Annual Research Briefs 2004. Center for Turbulence Research, Stanford University / NASA Ames.

HAM, F., MATTSSON, K. & IACCARINO, G. 2006 Accurate and stable finite volume

operators for unstructured flow solvers. Annual Research Briefs 2006. Center for Turbulence Research, Stanford University.

Jameson, A. 2008 Formulation of kinetic energy preserving conservative schemes for gas dynamics and direct numerical simulation of one-dimensional viscous compressible flow in a shock tube using entropy and kinetic energy preserving schemes. *J. Sci. Comput.* **34**, 188–208.

Jameson, A., Baker, T. J. & Weatherill, N. P. 1986 Calculation of inviscid transonic flow over a complete aircraft. In *AIAA paper 86-0103, AIAA 24th Aerospace Sciences Meeting*, pp. 86–0103.

Kreiss, H.-O. & Scherer, G. 1974 Finite element and finite difference methods for hyperbolic partial differential equations. In *Mathematical Aspects of Finite Elements in Partial Differential Equations*. Academic Press Inc.

Lele, S. K. 1992 Compact finite difference schemes with spectral-like resolution. *J. Comput. Phys.* **103**, 16.

Mahesh, K., Constantinescu, G. & Moin, P. 2004 A numerical method for large-eddy simulation in complex geometries. *J. Comput. Phys.* **197**, 215–240.

Moreau, S., Neal, D., Khalighi, Y., Wang, M. & Iaccarino, G. 2006 Validation of unstructured-mesh les of the trailing-edge flow and noise of a controlled-diffusion airfoil. Proceedings of the 2006 Summer Program. Center for Turbulence Research, Stanford University / NASA Ames.

Nordstrom, J., Forsberg, K., Adamsson, C. & Eliasson, P. 2003 Finite volume methods, unstructured meshes and strict stability. *Applied Numerical Mathematics* **45**, 453–473.

Pierce, C. D. & Moin, P. 2004 Progress-variable approach for large-eddy simulation of non-premixed turbulent combustion. *J. Fluid Mech.* **504**, 73–97.

Rostand, P. & Stoufflet, B. 1988 Tvd schemes to compute compressible viscous flows on unstructured meshes. *Notes on Numerical Fluid Mechanics* **24**, 510–520.

Shu, C.-W. & Osher, S. 1988 Efficient implementation of essentially non-oscillatory shock capturing schemes. *JCP* **77**, 439.

Yerry, M. A. & Shephard, M. S. 1984 Automatic three-dimensional mesh generation by the modified-octree approach. *International Journal for Numerical Methods in Engineering* **20**, 1965–1990.