LA-UR- 98-5561

Title: Properties of Space-Time Discontinuous Galerkin

Author(s): Robert B. Lowrie (X-HM)
Philip L. Roe (University of Michigan)
Bram van Leer (University of Michigan)

Submitted to: Applied Numerical Mathematics

PROPERTIES OF SPACE-TIME DISCONTINUOUS GALERKIN

ROBERT B. LOWRIE*, PHILIP L. ROE†, AND BRAM VAN LEER†

Abstract.

A method is developed for the simulation of nonlinear wave propagation where high accuracy is required. The approach is based on the space-time Discontinuous Galerkin (DG) finite-element method. In deriving the method, the idea of *compactness* is strictly followed. That is, that the discrete domain of dependence should contain a minimum amount of data outside the physical domain of dependence. Compactness is attained by carrying higher-order solution moments in each spatial-mesh cell and through the definition of a family of space-time meshes. For any order-of-accuracy, we prove the method is L_2-stable for Courant numbers less than 1 and satisfies an entropy condition.

A Fourier analysis of the DG(k) method is discussed, where k is the order of the polynomial used in each element. The method has a 'superconvergence' property, in that a component of the error converges as $\mathcal{O}(h^{2k+1})$, and this component dominates the error for long-time calculations. Numerical experiments indicate that the superconvergence property extends to some nonlinear cases.

The consequences of mesh staggering in time, in an effort to eliminate the need for the solution of the Riemann problem, are also investigated.

Numerical results are shown that verify the error convergence rate for scalar advection and the Euler equations. A companion paper will show more extensive numerical results.

1. Introduction. Two major challenges for computational fluid dynamics are problems that involve wave propagation over long times and problems with a wide range of amplitude scales. An example with both of these characteristics is the propagation *and* generation of acoustic waves, where the mean-flow amplitude scales are typically orders-of-magnitude larger than those of the generated acoustics. Other examples include vortex evolution and the direct simulation of turbulence. All of these problems require greater than second-order accuracy, whereas for nonlinear equations, most current methods are at best second-order accurate. Of the higher-order (greater than second-order) methods that do exist, most are

*Applied Theoretical and Computational Physics Division (X-HM), MS D413, Los Alamos National Laboratory, Los Alamos, New Mexico, 87545 (lowrie@lanl.gov).

†W. M. Keck Foundation Laboratory for Computational Fluid Dynamics, Aerospace Engineering Department, University of Michigan, Ann Arbor, Michigan, 48109.

tailored to high-spatial resolution, coupled with time integrators that are only second or third-order accurate. But for wave phenomena, time accuracy is as important as spatial accuracy.

One property of many successful second-order methods is that they attempt to follow the physics of hyperbolic problems. To develop higher-order methods, particularly for unsteady problems, it is tempting to violate this philosophy. Typically, higher accuracy is obtained by increasing the size of the update stencil. The consequence is that the stencil must stretch far outside of the physical domain of dependence. Instead, our aim is to develop time-accurate methods that minimize the size of the update stencil.

For these reasons, we have chosen to study the Discontinuous Galerkin (DG) method. The DG method was first proposed for solving the linear neutron transport problem [24, 33]. Thorough global error estimates are derived by Johnson and Pitkaranta [21]. For hyperbolic conservation laws, a semi-discrete form of DG has been developed by Cockburn et al. [10, 11, 12, 14]. Refinement and parallel implementation strategies have been studied by Bey et al. [7, 8], Biswas, Devine, and Flaherty [9], and Devine [15]. Some interesting issues regarding curved-wall boundaries have recently been studied by Bassi and Rebay [5]. Finally, the method has been extended to include diffusion terms in [4, 6, 13, 30].

The DG method in this study follows the 'space-time' approach. Control volumes (elements) are defined in space and time, and then a polynomial representation (which includes the time variable) of the solution is found in each element. In this way, the temporal error can be kept on the same order as the spatial error, which is critical for wave phenomena. The elements are arranged so that when solving a linear equation, each element can be solved for explicitly in a marching procedure. This study proposes two space-time meshes for solving 1-D *nonlinear* problems using the DG method. These meshes result in a *point-implicit* method, in that the solution in each element requires the solution of a small system of equations and only a small number of elements may be coupled. Moreover, the implicitness is weak, arising solely from the nonlinearity, so that rapid iteration is possible. To avoid non-physical oscillations, a limiter has also been developed. The method is then extended to 2-D problems using a novel space-time mesh.

The concept of a space-time approach has been used in finite-element methods for quite some time [29].

More recently, the space-time approach of the related SUPG method is given by Shakib and Hughes [37, 38], Jaffre, Johnson, and Szepessy [18], and Perrochet [32]. Note that Reference [18] also refers to their approach as "Discontinuous Galerkin," presumably because space-time SUPG uses the DG methodology in the time-variable and possibly at a few select spatial locations (such as to track discontinuities; see also [20]).

In this study, one of the space-time meshes considered uses mesh staggering, in time, to eliminate the need for the Riemann solver necessary in most upwind methods. The first-order accurate Lax-Friedrichs scheme [22] effectively uses the same idea. A higher-order extension has been developed by Nessyahu and Tadmor [28]. Others that use staggering include Sanders and Weiser [36] and Arminjon, Viallon, and Madrane [1]. Although staggering is attractive from a cost perspective, the resultant schemes are diffusive at low Courant numbers. A discussion of the advantages and disadvantages of staggering, and in particular the issues for multi-dimensional problems, will be given in this paper.

The remainder of this paper is organized as follows: §2 will state the moment equations that DG numerically solves. Also, this section covers the element-face definitions, which are an integral part of the method. An overview of the numerical implementation is given in §3. The properties of the DG method, such as accuracy and stability, are discussed in §4 with special reference to the phenomenon of superconvergence in the error norm. Some numerical results are presented in §5, which at this point are used to confirm the theoretical properties. Further numerical results will be presented in a companion paper [26] and can also be found in [25, 27].

2. Discontinuous Galerkin Method.

2.1. Formulation. Consider a conservation law of N_{eqn}-equations in d-space dimensions:

$$\partial_t \mathbf{u} + \underline{\nabla} \cdot \underline{\mathbf{f}}(\mathbf{u}) = \mathbf{0}, \tag{2.1}$$

where $\mathbf{u}$ is the vector of conserved quantities, and

$$\underline{\nabla} = (\partial_{x_1}, \ldots, \partial_{x_d}), \qquad \underline{\mathbf{f}} = (\mathbf{f}_1, \ldots, \mathbf{f}_d).$$

A more compact form used in this study is

$$\vec{\nabla} \cdot \vec{\mathbf{f}} = \mathbf{0}, \tag{2.2}$$

where

$$\vec{\nabla} = (\partial_{x_1}, \ldots, \partial_{x_d}, \partial_t), \qquad \vec{\mathbf{f}} = (\mathbf{f}_1, \ldots, \mathbf{f}_d, \mathbf{u}).$$

The notation here is $\underline{v}$ is a vector in space, while $\vec{v}$ is a space-time vector, so that

$$\vec{v} = (v_{x_1}, \ldots, v_{x_d}, v_t) = (\underline{v}, v_t).$$

Assume that each component of $\underline{\mathbf{f}}(\mathbf{u})$ is smooth enough that (2.1) may be written as

$$\partial_t \mathbf{u} + \underline{\mathbf{A}} \cdot \underline{\nabla} \mathbf{u} = \mathbf{0}, \tag{2.3}$$

where each component of $\underline{\mathbf{A}}$ is a $N_{\text{eqn}} \times N_{\text{eqn}}$ Jacobian matrix. Define the matrix

$$\mathbf{A}_\ell = \underline{\ell} \cdot \underline{\mathbf{A}}, \tag{2.4}$$

with $\underline{\ell}$ a unit vector. This study assumes that (2.1) is hyperbolic; that is, for every $\underline{\ell}$, $\mathbf{A}_\ell$ has real eigenvalues, $\lambda_{\ell,m}$, $m = 1, 2, \ldots, N_{\text{eqn}}$, and distinct eigenvectors.

Divide the solution domain Ω into a set of N_e non-overlapping control volumes (elements), $\{\Omega_e\}$. Each

Ω_e is allowed to be any type of polyhedron, with boundary $\partial\Omega_e$. A sample space-time mesh for $d = 1$, between two time levels, is shown in Figure 2.1. On each element the DG method integrates the weak form of eq. (2.2):

$$\oint_{\partial\Omega_e} \Phi\, \vec{\mathbf{f}}_b \cdot \vec{n}_e \, dS - \int_{\Omega_e} \vec{\mathbf{f}}_e \cdot \vec{\nabla}\Phi \, dV = \mathbf{0}, \tag{2.5}$$

where $\vec{n}_e \equiv (n_{e,x_1}, \ldots, n_{e,x_d}, n_{e,t})$ is the outward boundary unit normal, and the differentials dV and dS are defined in Ω_e and $\partial\Omega_e$, respectively.

Within each element the solution is approximated as

$$\mathbf{w}_e(\vec{x}) = \sum_{i=1}^{N} \phi_{e,i}(\vec{x}) \mathbf{c}_{e,i}, \quad \forall \vec{x} \in \Omega_e, \tag{2.6}$$

where $\{\phi_{e,i}(\vec{x})\}$ are basis functions defined on Ω_e and $\{\mathbf{c}_{e,i}\}$ the expansion coefficient vectors. We stress that $\vec{x}$ includes the time variable. Also, $\mathbf{w}$ is the *parameter vector* [35]. For many conservation laws, the components of $\mathbf{u}$ and $\vec{\mathbf{f}}$ can be written as quadratic functions of the components of $\mathbf{w}$. This property is exploited to evaluate the integrals *exactly* in eq. (2.5) [25].

DG(k) will refer to the DG method with each $\phi_{e,i} \in \mathcal{P}_k(\Omega_e)$, where $\mathcal{P}_k(\Omega_e)$ is the space of polynomials of maximum order k defined on Ω_e. There are N-unknown N_{eqn}-vectors in each element, namely the expansion coefficient vectors, $\mathbf{c}_{e,i}$. The Galerkin approach generates N-vector equations from (2.5) by selecting the test function $\Phi(\vec{x})$ to span the set $\{\phi_{e,i}(\vec{x})\}$.

The value $\vec{\mathbf{f}}_b$ denotes the flux on the element boundary, which may be equal to one of the two neighboring element values, or some combination thereof. Note that the numerical solution is continuous within each element, but possibly discontinuous across element boundaries. That is, there are no constraints on the coefficients $\mathbf{c}_{e,i}$ in (2.6) to maintain continuity with neighboring elements. The precise definition of the boundary value is given in the next section.

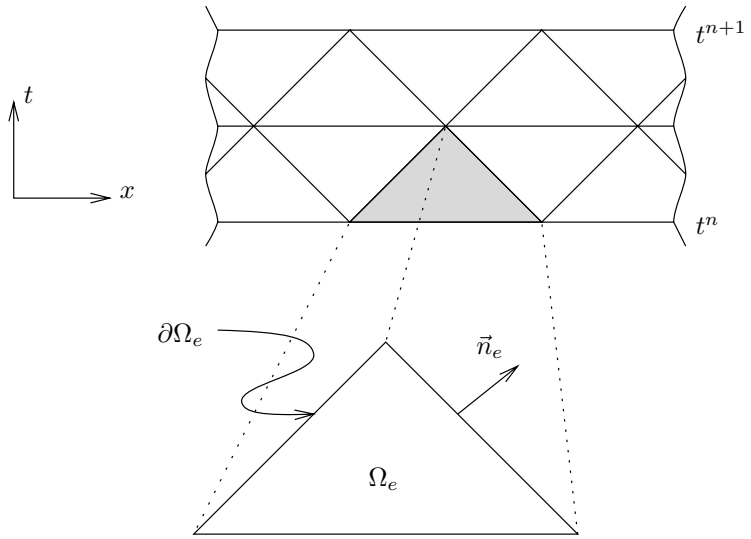


FIG. 2.1. *Sample Space-Time Mesh and Element Definition.*

2.2. Face Definitions. In this section, the element-boundary values, $(\cdot)_b$, will be defined. Each element is a polyhedron with boundary $\partial\Omega_e$, which is subdivided into faces, with $\partial\omega_e$ a specific face that separates two elements Ω_e and Ω_{e^*}. Define the interface value as a function of the values in the adjacent elements:

$$\mathbf{w}_b(\vec{x}) = \mathcal{F}(\mathbf{w}_e(\vec{x}), \mathbf{w}_{e^*}(\vec{x})), \quad \forall \vec{x} \in \partial\omega_e.$$

The function $\mathcal{F}$ is evaluated using an upwind approach. Following upwind finite-volume methods, on a face that is aligned with the t-axis ($n_{e,t} = 0$) a (approximate) Riemann solver is used. Such a face is referred to as a ***Riemann Face***.

For faces where $n_{e,t} \neq 0$, our present implementation requires that all characteristic paths cross the face in the same direction. Such a face is referred to as an ***Explicit Face***. Specifically, on an Explicit Face the quantity

$$\mu = (\underline{\ell}, \lambda_{\ell,m}) \cdot \vec{n}_e$$

is either strictly positive (also referred to as an "Outflow Face" with respect to element-e), or strictly negative ("Inflow Face"), for all m and $\underline{\ell}$, and all $\vec{x} \in \partial\omega_e$. The vector $(\underline{\ell}, \lambda_{\ell,m})$ is defined via (2.4). The boundary value is set as

$$\mathbf{w}_b(\vec{x}) = \begin{cases} \mathbf{w}_{e^*}(\vec{x}) & \text{if } \mu < 0: \quad \text{"Inflow"} \\ \mathbf{w}_e(\vec{x}) & \text{if } \mu > 0: \quad \text{"Outflow"} \end{cases} \tag{2.7}$$

for all $\vec{x} \in \partial\omega_e$. In practice, the calculation of $\lambda_{\ell,m}$ for all $\underline{\ell}$ and $\vec{x} \in \partial\omega_e$ is not needed, as long as a reasonable local value is used, along with a safety factor. So that the elements can be solved for in a time-marching procedure, on Explicit Faces we require that

$$\text{sgn}(\mu) = \text{sgn}(n_{e,t}), \tag{2.8}$$

where again, $n_{e,t}$ is the time component of the normal vector $\vec{n}_e$. This requirement translates into a Courant condition. In fact, for scalar conservation laws and a reasonable estimate for the eigenvalues, in §4.1 we show that (2.8) is sufficient for L_2-stability.

Figure 2.2 shows an example of both a Riemann and Explicit Face for a 1-D space-time mesh. Note that the same face definitions extend to multiple-space dimensions.

3. Implementation.

3.1. Space-Time Meshes. This section defines the space-time meshes on which the discrete form of the conservation law is solved. How the mesh is defined will greatly influence the cost of the DG method. There are roughly three criteria we follow when designing a space-time mesh:

1. Use only Riemann and Explicit Faces.
2. The method should be at worst point implicit.
3. The method should be compact.

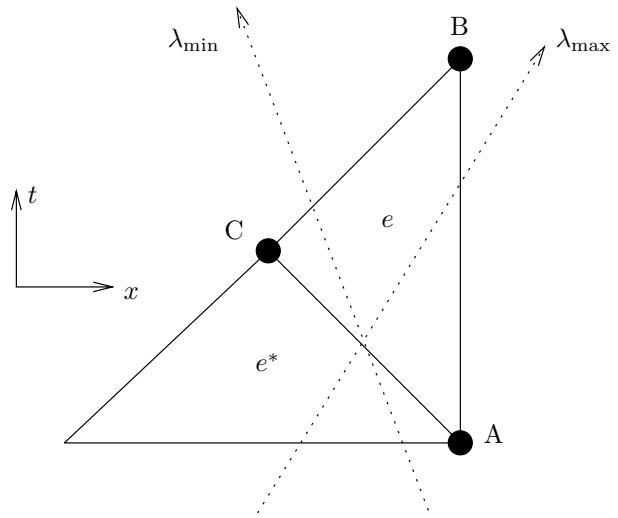
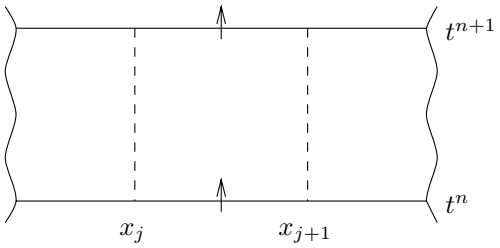


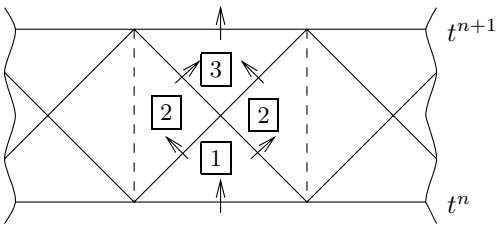
FIG. 2.2. *Example Faces on Two Triangular Elements. Face A-B is a Riemann Face. The other four faces are Explicit Faces. The paths labeled* $\lambda_{\min}$ *and* $\lambda_{\max}$ *correspond to the local minimum and maximum eigenvalues, respectively. The Explicit Face A-C may also be referred to as an Inflow Face with respect to element-e or an Outflow Face with respect to element-*e^**, with the solution on the face taken as* $\mathbf{w}_{e^*}(\vec{x})$.

By *point implicit* we mean that the solution in each element requires the solution of a small system of equations and only a small number of elements may be coupled. In other words, the update cost per time-step scales linearly with the total number of elements. By *compact*, we mean that the solution of an element should use a minimum of information outside of its physical domain of dependence.

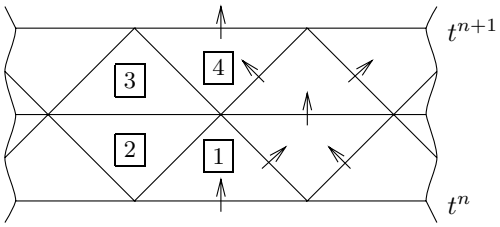
3.1.1. 1-D Meshes. First consider space-time meshes in one-space dimension ($d = 1$). Let the interval $0 \leq x \leq 1$ be discretized into N_x cells of size h_j, $j = 1, 2, \ldots, N_x$. Let the solution be given at some time t^n, and the solution is to be advanced to $t^{n+1} = t^n + \Delta t$. The most straightforward space-time mesh is shown in Figure 3.1a. Bar-Yoseph and Elata [3] have applied DG using such a mesh with promising results. However, at each time level the Rectangular Mesh couples the entire spatial domain through the Riemann Faces; eq. (2.5) becomes an $N_x \times N_x$ block tri-diagonal matrix. In other words, the DG method is fully implicit on the Rectangular Mesh. One advantage of such an approach is that there is no time-step restriction, which for problems with a large range in eigenvalues can offset the additional cost of a fully-implicit method. For problems without such disparate time scales, or for which the fastest waves must be resolved, an explicit or point-implicit method is more natural and cost effective.



(a) Rectangular.



(b) Riemann.



(c) Staggered.

FIG. 3.1. *Various Space-Time Mesh Segments. Solid lines denote Explicit Faces, with arrows indicating the flow of information. Dashed lines are Riemann Faces. The element types are numbered according to the order in which they are solved (Rectangular mesh solves for all elements simultaneously).*

In 1-D, there are two canonical space-time meshes that satisfy our criteria above. The Riemann Mesh, shown in Figure 3.1b, divides each cell of the Rectangular Mesh into four triangular elements. In general, when the system has eigenvalues of both sign, each Riemann Face couples two elements implicitly. That is to say, there is a system of equations that couples each type-$\boxed{2}$ element and its neighbor, and this system is twice as large as that for type-$\boxed{1}$ or type-$\boxed{3}$ elements. To avoid Riemann Faces altogether (except possibly at boundaries), the Staggered Mesh in Figure 3.1c will also be used.

For linear systems and both the Riemann and Staggered Meshes, the diagonal Explicit Faces satisfy the Courant condition (2.8) as long as the time-step is restricted as

$$\nu \equiv \frac{|\lambda|_{\max}\Delta t}{h_j} < 1, \quad \forall j,$$

where $|\lambda|_{\max}$ is the maximum-absolute eigenvalue of the linear system. For nonlinear systems, a local estimate of $|\lambda|_{\max}$ is computed at the beginning of the time-step, along with a safety factor.

Figure 3.1 also indicates the order in which the elements are solved. Referring to Figure 3.1b, the solution procedure for the Riemann Mesh is as follows:

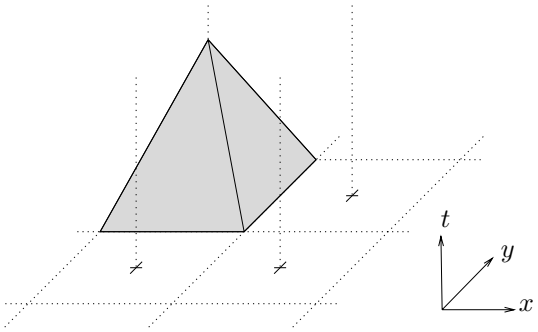
1. Solve in each of the elements of family $\boxed{1}$. The flux on the bottom face is either from the initial condition, or the solution in element $\boxed{3}$ from the previous time level.
2. Solve in the elements of family $\boxed{2}$. In general, these elements are coupled in pairs by the Riemann Faces. The two elements are *not* coupled implicitly if all of the eigenvalues are of the same sign.
3. Solve in each of the elements of family $\boxed{3}$.
4. Proceed to the next time level.

The Staggered Mesh is solved in a similar fashion, the only difference being that no elements are coupled implicitly. Consequently, the computational cost per time-step of the Staggered Mesh is cheaper than the Riemann Mesh.

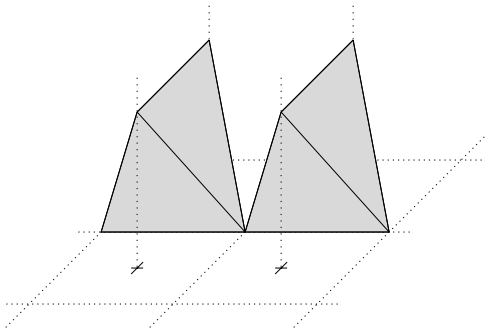
A mesh can also be defined that is made up of segments of the Riemann and Staggered Meshes. In particular, to enforce boundary conditions (aside from periodic) the Staggered Mesh uses a Riemann Mesh segment at the boundary [25].

3.1.2. 2-D Meshes. The 2-D space-time mesh is an extension of the 1-D Staggered Mesh of Figure 3.1c. The underlying 2-D spatial mesh is a quadrilateral mesh. To visualize the space-time mesh, Figure 3.2 shows the order in which elements are solved, over $\frac{1}{2}\Delta t$. In the next half time-step, the mesh is staggered; the Step (1) pyramid base, in the $\frac{1}{2}\Delta t < t \leq \Delta t$ interval, is coincident with the Step (3) pyramid base of the $0 < t \leq \frac{1}{2}\Delta t$ interval. Note that at each step, none of the element solutions are coupled, just as in 1-D Staggered Mesh. In particular, for Step (2), the same final solution results if the y-axis "valley" elements are solved before the x-axis "valley" elements. The method does not use "operator splitting." As a consequence, the results will show that this mesh does not exhibit a loss of accuracy when the advection direction is skewed with respect to the spatial mesh.

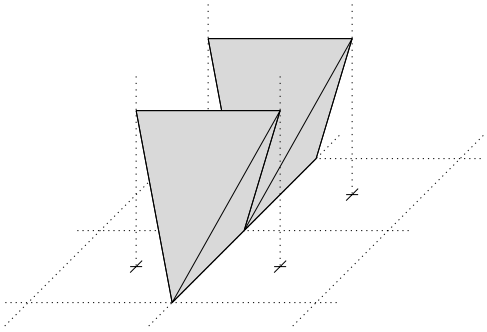
Reference [25] gives a 2-D analogy to the 1-D Riemann Mesh, meshes based on a 2-D triangular spatial mesh, and a 3-D hexahedral mesh. Only the Staggered Mesh has been used for 2-D problems, with Riemann



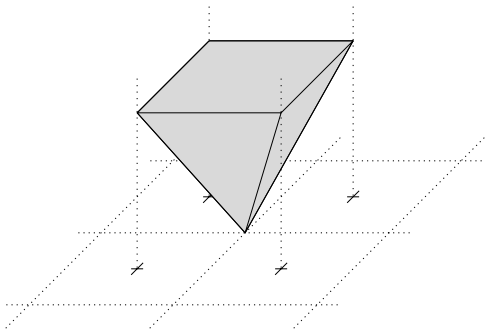
Step (1): Solve in the pyramid elements covering each x, y mesh cell.



Step (2): Solve in the tetrahedral elements that fill the x-axis valleys of the pyramid elements in Step [1].



Step (2)-continued: Solve in the tetrahedral elements that fill the y-axis valleys of the pyramid elements in Step [1].



Step (3): Solve in the inverted-pyramid elements.

FIG. 3.2. *Element Solution Order for the 2-D Staggered Mesh. Only* $\frac{1}{2}\Delta t$ *of mesh is shown.*

Faces used on the domain boundaries to enforce boundary conditions.

3.2. Algebraic System. For a nonlinear conservation law, eq. (2.5) results in a nonlinear system of equations for the coefficients, $\mathbf{c}_{e,i}$. This system is solved using Newton's method. Consider the general case where the solution of N_c elements are coupled. The linearized system of equations is of the form

$$\mathbf{M}(\delta\mathbf{c}) = -\mathbf{r}, \tag{3.1}$$

where $\delta\mathbf{c}$ is the update to the expansion coefficients, and if $\mathbf{r} = \mathbf{0}$, the discrete form of eq. (2.5) is satisfied for each of the coupled elements. The matrix $\mathbf{M}$ is a $N_c \times N_c$ block matrix; the structure of eq. (3.1) is given by

$$\begin{bmatrix} \ddots & & & & \\ & \mathbf{M}_e & \cdots & \mathbf{M}_{e,e^*} & \\ & \vdots & \ddots & \vdots & \\ & \mathbf{M}_{e^*,e} & \cdots & \mathbf{M}_{e^*} & \\ & & & & \ddots \end{bmatrix} \begin{pmatrix} \vdots \\ \delta\mathbf{c}_e \\ \vdots \\ \delta\mathbf{c}_{e^*} \\ \vdots \end{pmatrix} = - \begin{pmatrix} \vdots \\ \mathbf{r}_e \\ \vdots \\ \mathbf{r}_{e^*} \\ \vdots \end{pmatrix} .$$

Each submatrix $\mathbf{M}_{(\cdot)}$ is an $(N_{\text{eqn}}N) \times (N_{\text{eqn}}N)$ matrix, and

$$\delta\mathbf{c}_e = (\delta\mathbf{c}_{e,1}, \ldots, \delta\mathbf{c}_{e,N}) \quad .$$

From the boundary definitions, the coupling of elements can only occur across Riemann Faces. That is, the coupling matrices $\mathbf{M}_{e,e^*}$ and $\mathbf{M}_{e^*,e}$ are zero if no Riemann Faces separate elements e and e^*. For elements without Riemann Faces, eq. (3.1) reduces for a single element e to

$$\mathbf{M}_e(\delta\mathbf{c}_e) = -\mathbf{r}_e. \tag{3.2}$$

The formula for the residual vector $\mathbf{r}$ follows directly from eq. (2.5). The submatrices $\mathbf{M}_{(\cdot)}$ are derived by linearizing the fluxes. To compute the integrals in eq. (2.5), quadrature is used. A sufficient number of quadrature points is used so that the integrals are computed *exactly*; this is possible because the solution is expanded in terms of the parameter vector (eq. (2.6)). See [25] for the details.

4. Properties of the DG Method. The DG(k) method described in this study has the following properties:

1. Local conservation. See [25] for the details.
2. Under certain restrictions, both stability and an entropy condition are satisfied for any order-of-

accuracy. At this time, the proof has been carried out for any scalar conservation law that has a convex flux function, in any number of space dimensions.

3. High accuracy. Specifically, the method has a *superconvergence* property, in that under certain circumstances the error converges as $\mathcal{O}(h^{2k+1})$.

An overview of items 2 and 3 will be given in this section.

4.1. Entropy Condition and Stability. This section will show that the space-time DG(k) method, with certain constraints, satisfies an entropy condition for scalar conservation laws. The particular choice of entropy function will also imply L_2-stability. Jiang and Shu have already showed the semi-discrete DG method satisfies an entropy condition [19]. Summarizing their results:

- For the entropy function $\eta(u) = \frac{1}{2}u^2$, the entropy condition is satisfied without limiting, for any order of accuracy.
- To satisfy the entropy condition for all convex entropy functions, some sort of modification to the scheme is required.
- Several time discretizations also satisfy the entropy condition, including implicitness factors ranging from Crank-Nicholson to fully-implicit, and leapfrog.

It should then be no surprise that the space-time DG(k) also satisfies an entropy condition. However, the proof is illuminating, in that a Courant condition will fall out of the analysis. Our proof is valid for any number of space dimensions.

Consider the scalar conservation law

$$\partial_t u + \underline{\nabla} \cdot \underline{f}(u) = 0. \tag{4.1}$$

The corresponding entropy condition is written as

$$\partial_t \eta(u) + \underline{\nabla} \cdot \underline{q}(u) \leq 0. \tag{4.2}$$

In the construction of eq. (4.2), $\eta(u)$ is a strictly convex function, and

$$\underline{q}(u) = \int^u \eta'(\xi)\underline{f}'(\xi)\, d\xi.$$

Note that both (4.1) and (4.2) are in the weak sense; for smooth u, (4.2) is an equality.

The analysis in this section will only be for the entropy function $\eta(u) = \frac{1}{2}u^2$. When each component of $\underline{f}(u)$ is convex, convergence is assured if the entropy condition is satisfied for a single entropy function [16, 23]. For non-convex $\underline{f}(u)$, all possible convex $\eta(u)$ must satisfy the entropy condition. For this study, only convex flux functions will be considered.

For $\eta(u) = \frac{1}{2}u^2$, the entropy flux is given by

$$\underline{q}(u) = u\underline{f}(u) - \underline{g}(u), \tag{4.3}$$

where

$$\underline{g}(u) = \int^u \underline{f}(\xi)\, d\xi.$$

With $\Phi = u_e$, eq. (2.5) in this case can be written as

$$\oint_{\partial\Omega_e} u_e\, (\underline{f}, u)_b \cdot \vec{n}_e\, dS + \int_{\Omega_e} \vec{\nabla}\cdot(\underline{g}, \eta)_e\, dV = 0,$$

which by the Divergence Theorem reduces to

$$\oint_{\partial\Omega_e} (u_e\underline{f}_b - \underline{g}_e, u_e u_b - \eta_e)\cdot\vec{n}_e\, dS = 0. \tag{4.4}$$

First, consider elements that have no Riemann Faces. The boundary values on each face are then given by eq. (2.8), so that

$$(\underline{f}, u)_b = \begin{cases} (\underline{f}, u)_{e^*} & \text{if } \mu < 0, \\ (\underline{f}, u)_e & \text{if } \mu > 0, \end{cases} \tag{4.5}$$

with $\mu = (\underline{\hat{a}}, 1) \cdot \vec{n}_e$, and where $\underline{\hat{a}}$ corresponds to an average wave velocity that will be defined as part of satisfying the entropy condition. The boundary-entropy fluxes are defined by simply replacing $(\underline{f}, u)$ by $(\underline{q}, \eta)$ in the expression above. Equation (4.4) becomes

$$\int_{\mu>0} (u_e \underline{f}_e - \underline{g}_e, u_e^2 - \eta_e) \cdot \vec{n}_e \, dS + \int_{\mu<0} (u_e \underline{f}_{e^*} - \underline{g}_e, u_e u_{e^*} - \eta_e) \cdot \vec{n}_e \, dS = 0.$$

Re-arranging this expression, and using eq. (4.3) gives

$$\int_{\mu>0} (\underline{q}, \eta)_e \cdot \vec{n}_e \, dS + \int_{\mu<0} (\underline{q}, \eta)_{e^*} \cdot \vec{n}_e \, dS = E, \tag{4.6}$$

where

$$E = \int_{\mu<0} \left(\underline{f}_{e^*} \Delta u - \Delta \underline{g}, \frac{1}{2} (\Delta u)^2 \right) \cdot \vec{n}_e \, dS. \tag{4.7}$$

and $\Delta(\cdot) \equiv (\cdot)_{e^*} - (\cdot)_e$. But, using the boundary-value definitions of eq. (4.5), the left-hand side of eq. (4.6) is in the form of the left side of the following:

$$\oint_{\partial \Omega_e} (\underline{q}, \eta)_b \cdot \vec{n}_e \, dS \leq 0, \tag{4.8}$$

which is a weak form of the entropy condition, (4.2). Therefore, the entropy condition is satisfied if

$$E \leq 0. \tag{4.9}$$

An easy case to analyze is when $\underline{f}(u) = (\frac{1}{2}u^2, c)$, where c is a constant. Equation (4.7) then reduces to

$$E = \oint_{\mu<0} \frac{1}{2}(\Delta u)^2 (\underline{\hat{a}}, 1)\cdot\vec{n}_e \, dS, \tag{4.10}$$

where $\hat{a} = ((2u_{e^*} + u_e)/3, c)$. The condition (4.9) is then satisfied if

$$(\underline{\hat{a}}, 1)\cdot\vec{n}_e \leq 0. \tag{4.11}$$

But this constraint is simply a variant of the Courant condition, eq. (2.8), and is the loosest possible constraint consistent with causality.

For a more general $\underline{f}(u)$, the trapezoidal rule can be used to estimate $\Delta\underline{g}$. Then E takes the form of eq. (4.10) with an average wave velocity given by

$$\underline{\hat{a}} = \frac{\Delta \underline{f}}{\Delta u} + \frac{1}{6}\underline{f''}|_{\underline{\xi}} \Delta u, \tag{4.12}$$

where $\underline{f''}|_{\underline{\xi}} = (f_1''(\xi_1), \ldots, f_d''(\xi_d))$ and each ξ_i is between u_e and u_{e^*}. At least for small Δu, it easy to show that each component of $\underline{\hat{a}}$ is bounded between its respective component in $\underline{f'}(u_e)$ and $\underline{f'}(u_{e^*})$. In theory, the time-step could be determined by maximizing $(\underline{\hat{a}}, 1)\cdot\vec{n}_e$ over $\underline{\xi}$ and ensuring that (4.11) is satisfied. In practice some average wave velocity over the element is used, along with a safety factor. Regardless, with the Courant condition (4.11), the entropy condition is satisfied.

The contribution of any Riemann Faces can now be included. A single Riemann Face contributes

$$\underline{n}_e \cdot \int_0^{\Delta t} (u_e \underline{f}^R - \underline{g}_e)\, dt \tag{4.13}$$

to the left-hand side of (4.4). Therefore, the contribution to eq. (4.6) will be of the form

$$\underline{n}_e \cdot \int_0^{\Delta t} \underline{q}^R\, dt - E^R, \tag{4.14}$$

where E^R is the contribution to E. It remains to show that $E^R \leq 0$, and therefore that the contribution from Riemann Faces only helps in satisfying the entropy condition.

Just as in [19], require that the Riemann flux $\underline{f}^R$ be an E-flux [31]:

$$\left(\underline{f}^R(u^-, u^+) - \underline{f}(u)\right)(u^+ - u^-) \leq 0$$

for all u between u^- and u^+. Define an entropy flux, consistent with (4.3), as

$$\underline{q}^R = \begin{cases} u_e \underline{f}^R - \underline{g}_e & \text{if } \sigma > 0, \\ u_{e^*} \underline{f}^R - \underline{g}_{e^*} & \text{if } \sigma < 0, \end{cases}$$

with

$$\sigma \equiv \underline{n}_e \cdot (\underline{x}^+ - \underline{x}^-),$$

where $(\underline{x}^+ - \underline{x}^-)$ denotes a direction pointing from u^- to u^+. Note that since $\underline{n}_e = -\underline{n}_{e^*}$, the above definition is conservative. With this definition, for $\sigma > 0$, by comparing eqs. (4.13) and (4.14) obviously $E^R = 0$. For

$\sigma < 0$,

$$E^R = -\int_0^{\Delta t} (\underline{f}^R \Delta u - \Delta \underline{g})\, dt.$$

By the Mean Value Theorem, $\Delta \underline{g} = \underline{f}(u_1)\Delta u$ for some u_1 between u_e and u_{e^*}. So,

$$E^R = \int_0^{\Delta t} (u_e - u_{e^*}) \left(\underline{f}^R - \underline{f}(u_1) \right) dt.$$

From the definition of σ, it follows that $u^+ \equiv u_e$ and $u^- \equiv u_{e^*}$, and so $E^R \leq 0$.

This completes the proof that DG(k) satisfies an entropy condition for all convex flux functions. The important constraint is the Courant condition, eq. (4.11). If Riemann Faces are used, the Riemann solver must also be an E-flux.

Note that satisfying eq. (4.8) also implies stability in L_2 [19]; if $u(\underline{x}, \cdot)$ has compact support, for every $t_2 > t_1$,

$$\int_{-\infty}^{\infty} \eta(u(\underline{x}, t_2))\, dx \leq \int_{-\infty}^{\infty} \eta(u(\underline{x}, t_1))\, dx.$$

We stress that this stability property is independent of the order k.

4.2. Accuracy. For a method using an order-k interpolant, the expectation is that at best the error will converge as $\mathcal{O}(h^{k+1})$, where h is some measure of the mesh size. A norm typically used to measure the solution accuracy at a given time-level is

$$L_p(v) = \left\{ \frac{1}{|\Omega_d|} \int_{\Omega_d} |v_{\text{exact}}(\underline{x}) - v(\underline{x})|^p \, dx \right\}^{1/p}, \tag{4.15}$$

where Ω_d is the spatial domain, p is an integer with $p \geq 1$, and v, v_{exact} are the numerical and exact solutions of a representative variable of the conservation law. A finite-element analysis shows that DG(k) converges

in L_2 as $\mathcal{O}(h^{k+1/2})$ [18]. However, for many smooth solutions on 'regular' meshes, practitioners often realize $\mathcal{O}(h^{k+1})$ accuracy [18, 34].

As alluded to earlier, for a fixed Courant number, for some calculations we realize a convergence rate of $\mathcal{O}(h^{2k+1})$. This *superconvergence* most often occurs with long-time calculations. In this section, an argument is given as to why superconvergence can occur. We do this simply through a Fourier analysis. Although a Fourier analysis is much more restrictive than the finite-element analysis of Reference [18], in terms of assumptions on the regularity of the solution and mesh, a Fourier analysis will distinguish various components of the error. Most importantly in the present context, the analysis will show the dominating error component for long-time calculations. Although the Fourier analysis here will be for a linear scalar equation, the analysis will also help explain some of the results in §5.2 for a nonlinear system. We stress that our objective here is simply to give a rough explanation as to why superconvergence can occur, not to develop specific conditions for its existence, nor to develop rigorous error estimates for the most general case.

Consider numerically solving linear advection,

$$\partial_t u + a\partial_x u = 0, \tag{4.16}$$

on an equally-spaced mesh with $a > 0$. Let the x-coordinate in cell-j be written as

$$x = x_j + h\xi,$$

where h is the mesh spacing, x_j the cell centroid, and $\xi \in \left[-\frac{1}{2}, \frac{1}{2}\right]$. The DG($k$) numerical solution at time-level n in cell-j can be written as

$$U_j^n(\xi) = \sum_{m=1}^{k+1} \bar{c}_{j,m}^n \bar{\phi}_m(\xi). \tag{4.17}$$

The basis functions $\{\bar{\phi}_m(\xi)\}$ are Legendre polynomials mapped onto the domain of ξ. The use of Legendre

polynomials is not necessary to realize the accuracy claims in this section, but simplifies the discussion. Referring to Figure 3.1, the above expansion corresponds to the space-time expansion (2.6) evaluated at the previous time level, on the upper face of triangle $\boxed{3}$ for the Riemann Mesh and triangle $\boxed{4}$ for the Staggered Mesh.

A property of the Legendre basis is that it is orthogonal. Therefore, the expansion coefficients can be initialized as

$$\bar{c}^0_{j,m} = \int_{-\frac{1}{2}}^{\frac{1}{2}} \bar{\phi}_m(\xi) u^0_j(\xi)\, d\xi \Bigg/ \int_{-\frac{1}{2}}^{\frac{1}{2}} \bar{\phi}^2_m(\xi)\, d\xi \,, \tag{4.18}$$

where $u^0_j(\xi)$ is the initial condition in cell-j. It is easy to show that these coefficients minimize the difference between $U^0_j(\xi)$ and $u^0_j(\xi)$ in a least-squares sense.

A compact notation for (4.17) is

$$U^n_j(\xi) = \left(\bar{\mathbf{c}}^n_j\right) \quad \mathbf{B}(\xi), \tag{4.19}$$

where $\bar{\mathbf{c}}_j \equiv (\bar{c}_{j,1}, \bar{c}_{j,2}, \ldots, \bar{c}_{j,k+1})$, and $\mathbf{B}(\xi) \equiv (\bar{\phi}_1(\xi), \bar{\phi}_1(\xi), \ldots, \bar{\phi}_{k+1}(\xi))$. In general, the update to the expansion coefficients can be written as

$$\bar{\mathbf{c}}^{n+1}_j = \mathcal{G}\bar{\mathbf{c}}^n_j, \tag{4.20}$$

where $\mathcal{G}$ is a $(k+1) \times (k+1)$ matrix. A Fourier analysis studies the influence of the update-matrix $\mathcal{G}$ on a wave of specified amplitude and frequency. The initial condition in the j'th cell is taken to be

$$u^0_j(\xi) = e^{i\theta(j+\xi)},$$

where $i = \sqrt{-1}$ and θ is the phase-change per mesh interval. The exact solution to (4.16) is

$$u_j^n(\xi) = g_{\text{exact}}^n u_j^0(\xi), \tag{4.21}$$

where

$$g_{\text{exact}} = e^{-i\nu\theta},$$

and $\nu = a\Delta t/h$. The projection of the exact solution onto the expansion (4.19) is then

$$\left[U_j^n(\xi)\right]_{\text{exact}} = g_{\text{exact}}^n e^{ij\theta} \mathbf{e}_{\text{exact}} \mathbf{B}(\xi),$$

where

$$\mathbf{e}_{\text{exact},m} = \int_{-\frac{1}{2}}^{\frac{1}{2}} \bar{\phi}_m(\xi) e^{i\xi\theta}\, d\xi \Bigg/ \int_{-\frac{1}{2}}^{\frac{1}{2}} \bar{\phi}_m^2(\xi)\, d\xi\ . \tag{4.22}$$

The notation $\mathbf{e}_{\text{exact},m}$ means the m'th component of the vector $\mathbf{e}_{\text{exact}}$. Also, from eq. (4.18),

$$\bar{\mathbf{c}}_j^0 = e^{ij\theta} \mathbf{e}_{\text{exact}}.$$

For $n > 0$, the coefficient vector is assumed to take the form

$$\bar{\mathbf{c}}_j^n = g^n e^{ij\theta} \mathbf{e},$$

where $\mathbf{e}$ is a constant vector and g the complex amplification. By substituting the above relation into (4.20), it is apparent that g is an eigenvalue of $\mathcal{G}$ with corresponding eigenvector $\mathbf{e}$. The $(k+1)$-eigenvalues are

numbered so that

$$|g_1| \geq |g_2| \geq |g_3| \ldots \geq |g_{k+1}|. \tag{4.23}$$

In other words, the g_1-eigenvalue represents the least-damped or *accurate* mode; the other modes are *spurious*. A rough measure of the accuracy of DG(k) is the 'closeness' of g_1 and $\mathbf{e}_1$ to g_{exact} and $\mathbf{e}_{\text{exact}}$.

The numerical solution can now be written as

$$U_j^n(\xi) = e^{ij\theta} \sum_{\ell=1}^{k+1} g_\ell^n \mathbf{e}_\ell \ \mathbf{B}(\xi). \tag{4.24}$$

Here the eigenvectors are scaled so that

$$\sum_{\ell=1}^{k+1} \mathbf{e}_\ell = \mathbf{e}_{\text{exact}}. \tag{4.25}$$

This unique scaling ensures that the projection of the initial condition is satisfied; specifically, that $U_j^0(\xi) = \left[U_j^0(\xi)\right]_{\text{exact}}$.

The norm (4.15) can be written in this case as

$$L_p = \left[\frac{1}{N_{\text{cell}}} \sum_{j=1}^{N_{\text{cell}}} \int_{-\frac{1}{2}}^{\frac{1}{2}} |u_j^n(\xi) - U_j^n(\xi)|^p \, d\xi \right]^{1/p},$$

where N_{cell} is the number of mesh cells. Substituting (4.21) and (4.24) into the above expression gives

$$L_p = \left\| g_{\text{exact}}^n e^{i\theta\xi} - \sum_{\ell=1}^{k+1} g_\ell^n \mathbf{e}_\ell \ \mathbf{B}(\xi) \right\|_p$$

where

$$\|f(\xi)\|_p \equiv \left[\int_{-\frac{1}{2}}^{\frac{1}{2}} |f(\xi)|^p \, d\xi\right]^{1/p}.$$

A straightforward bound on L_p is

$$L_p \leq (L_p)^{\text{init}} + (L_p)^{\text{evol}} + (L_p)^{\text{spur}}, \tag{4.26a}$$

where

$$(L_p)^{\text{init}} = \left\| e^{i\theta\xi} - \mathbf{e}_1 \, \mathbf{B}(\xi) \right\|_p, \tag{4.26b}$$

$$(L_p)^{\text{evol}} = |g_{\text{exact}}^n - g_1^n| \left\| \mathbf{e}_1 \, \mathbf{B}(\xi) \right\|_p, \tag{4.26c}$$

$$(L_p)^{\text{spur}} = \sum_{\ell=2}^{k+1} |g_\ell|^n \left\| \mathbf{e}_\ell \, \mathbf{B}(\xi) \right\|_p. \tag{4.26d}$$

The $(L_p)^{\text{init}}$-norm represents the error in projecting the initial condition onto the accurate mode. This is a one-time contribution to the overall error. The $(L_p)^{\text{evol}}$-norm represents the error in the evolution of the accurate mode. For small $|g_{\text{exact}} - g_1|$,

$$(L_p)^{\text{evol}} \leq n \, |g_{\text{exact}} - g_1| \left\| \mathbf{e}_1 \, \mathbf{B}(\xi) \right\|_p + \mathcal{O}\left(n^2 \, |g_{\text{exact}} - g_1|^2 \right). \tag{4.27}$$

The $(L_p)^{\text{spur}}$-norm represents the contribution of the spurious modes. Since DG(k) is stable, $|g_\ell| \leq 1$, and therefore a loose bound on $(L_p)^{\text{spur}}$ that will prove useful is

$$(L_p)^{\text{spur}} \leq \sum_{\ell=2}^{k+1} \left\| \mathbf{e}_\ell \, \mathbf{B}(\xi) \right\|_p \equiv (L_p)^{\text{spur}}_{\text{max}}. \tag{4.28}$$

Another norm we will study is the error in the cell averages, defined by

$$\bar{L}_p(v) = \left[\frac{1}{N_{\text{cell}}} \sum_{j=1}^{N_{\text{cell}}} |\bar{v}_{j,\text{exact}} - \bar{v}_j|^p \right]^{1/p}, \tag{4.29}$$

where $\bar{v}_j$ denotes the cell-average. A property of the Legendre basis used in (4.17) is that the coefficient $\bar{c}^n_{j,1}$ is the cell average in the j'th cell. Consequently, for the Fourier analysis the cell-average norm reduces to

$$\bar{L}_p = \left| g^n_{\text{exact}} \mathbf{e}_{\text{exact},1} - \sum_{\ell=1}^{k+1} g^n_\ell \mathbf{e}_{\ell,1} \right|. \tag{4.30}$$

This expression can be bounded in a similar manner as the L_p-norm;

$$\bar{L}_p \leq \left(\bar{L}_p\right)^{\text{init}} + \left(\bar{L}_p\right)^{\text{evol}} + \left(\bar{L}_p\right)^{\text{spur}}, \tag{4.31a}$$

where

$$\left(\bar{L}_p\right)^{\text{init}} = \left|\mathbf{e}_{\text{exact},1} - \mathbf{e}_{1,1}\right|, \tag{4.31b}$$

$$\left(\bar{L}_p\right)^{\text{evol}} = \left|g^n_{\text{exact}} - g^n_1\right| \left|\mathbf{e}_{1,1}\right|, \tag{4.31c}$$

$$\left(\bar{L}_p\right)^{\text{spur}} = \sum_{\ell=2}^{k+1} \left|g_\ell\right|^n \left|\mathbf{e}_{\ell,1}\right|. \tag{4.31d}$$

Analogous to the bounds (4.27) and (4.28),

$$\left(\bar{L}_p\right)^{\text{evol}} \leq n \left|g_{\text{exact}} - g_1\right| \left|\mathbf{e}_{1,1}\right| + \mathcal{O}\left(n^2 \left|g_{\text{exact}} - g_1\right|^2\right), \tag{4.32}$$

$$\left(\bar{L}_p\right)^{\text{spur}} \leq \sum_{\ell=2}^{k+1} \left|\mathbf{e}_{\ell,1}\right| \equiv \left(\bar{L}_p\right)^{\text{spur}}_{\text{max}}. \tag{4.33}$$

Further analysis requires expressions for g_ℓ and $\mathbf{e}_\ell$. For DG(1), the numerical solution can be written as

$$U_j^n(\xi) = \bar{U}_j^n + \bar{S}_j^n \xi,$$

so that in relation to eq. (4.19),

$$\bar{\mathbf{c}}_j^n = \begin{pmatrix} \bar{U} \\ \bar{S} \end{pmatrix}_j^n, \qquad \mathbf{B}(\xi) = \begin{pmatrix} 1 \\ \xi \end{pmatrix}.$$

The variable $\bar{U}_j^n$ is the cell-average, while $\bar{S}_j^n/h$ can be thought of as a slope; in fact, through eq. (4.18) it can be shown that

$$\bar{S}_j^0/h = \partial_x u^0(x_j) + \mathcal{O}(h^2). \tag{4.34}$$

The update matrix for DG(1) on the Riemann Mesh is

$$\mathcal{G}^R = \begin{bmatrix} 1-\nu+\nu T^{-1} & -\frac{1}{2}\nu(1-\nu)(1-T^{-1}) \\ 6\nu(1-\nu)(1-T^{-1}) & (1-\nu)(1-2\nu-2\nu^2) - \nu(3-6\nu+2\nu^2)T^{-1} \end{bmatrix}, \tag{4.35}$$

where $T^m u_j \equiv u_{j+m}$. Note that $\mathcal{G}^R_{1,1}$ is the first-order upwind scheme. A Fourier analysis for the matrix (4.35) is given by Van Leer, who referred to this method as Scheme III [40]. Here we will give the analysis for DG(1) on the Staggered Mesh, which has an update matrix given by

$$\mathcal{G}^S = \mathcal{G}^{\frac{1}{2}S}\mathcal{G}^{\frac{1}{2}S}, \tag{4.36a}$$

where

$$\mathcal{G}^{\frac{1}{2}S} = \begin{bmatrix} \frac{1}{2}(1+\nu) & \frac{1}{8}(1-\nu^2) \\ -\frac{3}{2}(1-\nu^2) & -\frac{1}{4}(1+\nu)(1-4\nu+\nu^2) \end{bmatrix} T^{-\frac{1}{2}} + \begin{bmatrix} \frac{1}{2}(1-\nu) & -\frac{1}{8}(1-\nu^2) \\ \frac{3}{2}(1-\nu^2) & -\frac{1}{4}(1-\nu)(1+4\nu+\nu^2) \end{bmatrix} T^{\frac{1}{2}}. \tag{4.36b}$$

In this case, $\mathcal{G}^{\frac{1}{2}S}_{1,1}$ is the Lax-Friedrichs scheme.

For a matrix in the form of (4.35) or (4.36), the analysis reduces to setting $T^m = e^{m\theta}$ and analyzing the eigenstructure. The eigenvalues of (4.36) are, for small θ,

$$g_1 = 1 - i\nu\theta - \frac{1}{2}(\nu\theta)^2 + \frac{1}{6}i(\nu\theta)^3 + \frac{1}{576}\left(25\nu^4 + 2\nu^2 - 3\right)\theta^4 + \mathcal{O}(\theta^5),$$
$$g_2 = \frac{1}{4}\left(1-3\nu^2\right)^2 + \frac{1}{4}i\left(3-\nu^2\right)\left(1-3\nu^2\right)\nu\theta + \mathcal{O}(\theta^2).$$

Comparing g_1 with g_{exact} gives

$$g_{\text{exact}} - g_1 = \frac{1}{576}(1-\nu^2)(3+\nu^2)\theta^4 + \mathcal{O}(\theta^5).$$

Assuming that $n\theta = \mathcal{O}(1)$, we obtain from (4.27) and (4.32) that

$$(L_p)^{\text{evol}} = \mathcal{O}(\theta^3), \qquad \left(\bar{L}_p\right)^{\text{evol}} = \mathcal{O}(\theta^3). \tag{4.37}$$

That is, the accurate mode is propagated with *third*-order accuracy, which is a property that is shared by DG(1) on the Riemann Mesh (Scheme III of [40]). The evolution of the 2-mode is inconsistent with (4.16); the 2-mode is a spurious mode. However, with the condition $|\nu| < 1$, the 2-mode is damped faster than the accurate mode. Values for the damping will be given in §4.2.1.

The eigenvectors corresponding to g_1 and g_2 can be written as

$$\mathbf{e}_1 = \begin{pmatrix} 1 - \frac{1}{24}\theta^2 + \frac{1}{72}i\theta^3 \\ i\theta + \frac{1}{6}\nu\theta^2 - \frac{1}{72}i\left(3 + 2\nu^2\right)\theta^3 \end{pmatrix} + \mathcal{O}(\theta^4), \tag{4.38}$$

$$\mathbf{e}_2 = \begin{pmatrix} -\frac{1}{72}i\theta^3 \\ -\frac{1}{6}\nu\theta^2 + \frac{1}{180}i\left(3 + 5\nu^2\right)\theta^3 \end{pmatrix} + \mathcal{O}(\theta^4). \tag{4.39}$$

An evaluation of (4.22) results in

$$\begin{aligned} \mathbf{e}_{\text{exact}} &= \frac{2}{\theta}\begin{pmatrix} \sin\left(\frac{\theta}{2}\right) \\ \frac{6i}{\theta}\left[2\sin\left(\frac{\theta}{2}\right) - \theta\cos\left(\frac{\theta}{2}\right)\right] \end{pmatrix}, \\ &= \begin{pmatrix} 1 - \frac{1}{24}\theta^2 \\ i\theta - \frac{1}{40}i\theta^3 \end{pmatrix} + \mathcal{O}(\theta^4). \end{aligned}$$

Substituting the eigenvector expansions into (4.26b) and (4.28) gives that

$$(L_p)^{\text{init}} = \mathcal{O}(\theta^2), \qquad (L_p)^{\text{spur}}_{\text{max}} = \mathcal{O}(\theta^2),$$

while from (4.31b) and (4.33),

$$\left(\bar{L}_p\right)^{\text{init}} = \mathcal{O}(\theta^3), \qquad \left(\bar{L}_p\right)^{\text{spur}}_{\text{max}} = \mathcal{O}(\theta^3).$$

Combined with the result (4.37), DG(1) is second-order accurate with respect to the L_p-norm, but third-order accurate with respect to the $\bar{L}_p$-norm. However, in numerical experiments we sometimes realize third-order accuracy in L_p, which indicates that in these cases the $(L_p)^{\text{evol}}$-norm is the dominating component. A discussion of this phenomenon will be given in §4.2.1.

Note that one interpretation of $\mathbf{e}_{1,2}$ is that the accurate mode considers the 'slope' to be

$$\frac{S_1}{h} = \partial_x u - \frac{1}{6}\nu h \partial_{xx} u + \mathcal{O}(h^2). \tag{4.40}$$

That is, once the 2-mode is fully damped, the solution has a slope that is only first-order accurate. One difficulty with interpreting S/h as a slope is when developing an accuracy-preserving limiter. Comparing S/h to $\mathcal{O}(h^2)$-slopes (formed by differencing $\{\bar{U}_j\}$) will, by (4.40), result in S/h appearing as an outlier.

The same trends observed in this section for the Staggered Mesh are also true for the Riemann Mesh, with changes only to the error coefficients.

For $k = 2$ and $k = 3$, the Fourier analysis has been computed numerically [25]. We observe the following convergence rates:

$$(L_p)^{\text{evol}} = \mathcal{O}(\theta^{2k+1}), \qquad \left(\bar{L}_p\right)^{\text{evol}} = \mathcal{O}(\theta^{2k+1}), \tag{4.41a}$$

$$(L_p)^{\text{spur}}_{\text{max}} = \mathcal{O}(\theta^{k+1}), \qquad \left(\bar{L}_p\right)^{\text{spur}}_{\text{max}} = \mathcal{O}(\theta^{k+2}). \tag{4.41b}$$

In addition, for the $(L_2)^{\text{init}}$ and $\left(\bar{L}_p\right)^{\text{init}}$-norms, we computed the same convergence rate as their respective $(\cdot)^{\text{spur}}_{\text{max}}$-norm. The reason is that for the cell-average norm, using (4.25),

$$\left(\bar{L}_p\right)^{\text{init}} = \left|\sum_{\ell=2}^{k+1} \mathbf{e}_{\ell,1}\right| \leq \left(\bar{L}_p\right)^{\text{spur}}_{\text{max}}.$$

Similarly,

$$\begin{aligned}(L_p)^{\text{init}} &= \left\| e^{i\theta\xi} - \mathbf{e}_{\text{exact}}\mathbf{B}(\xi) + \sum_{\ell=2}^{k+1} \mathbf{e}_\ell\, \mathbf{B}(\xi) \right\|_p \\ &\leq \left\| e^{i\theta\xi} - \mathbf{e}_{\text{exact}}\mathbf{B}(\xi) \right\|_p + (L_p)^{\text{spur}}_{\text{max}}.\end{aligned}$$

For any reasonable basis $\mathbf{B}(\xi)$ that uses order-k polynomials,

$$\left\| e^{i\theta\xi} - \mathbf{e}_{\text{exact}}\mathbf{B}(\xi) \right\|_p = \mathcal{O}(\theta^{k+1});$$

that is, the same convergence rate observed for the $(L_p)^{\text{spur}}_{\text{max}}$-norm.

From the eqs. (4.41), it is apparent that the convergence is dictated by the $(\cdot)^{\text{spur}}_{\text{max}}$-norm. Therefore,

$$L_p = \mathcal{O}(\theta^{k+1}), \tag{4.42a}$$

$$\bar{L}_p = \mathcal{O}(\theta^{k+2}), \quad k \geq 1. \tag{4.42b}$$

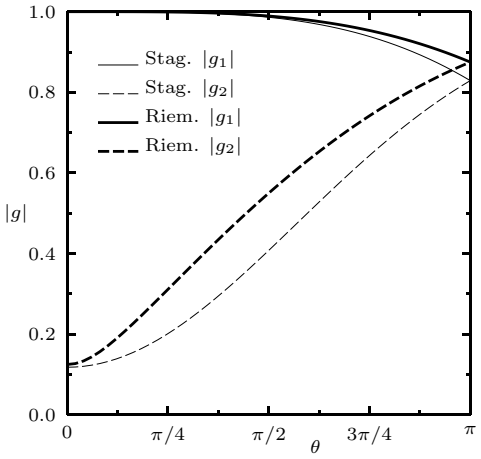
Note that DG(0), although not discussed here, converges as $\bar{L}_p = \mathcal{O}(\theta)$, since in this case the convergence is determined by the $(\bar{L}_p)^{\text{evol}}$-norm.

4.2.1. Conditions for Superconvergence. Using the analysis of the previous section, in this section an argument is given as to why superconvergence can occur. The argument will be made for the L_p-norm with the understanding that the same argument can be made for the $\bar{L}_p$-norm.

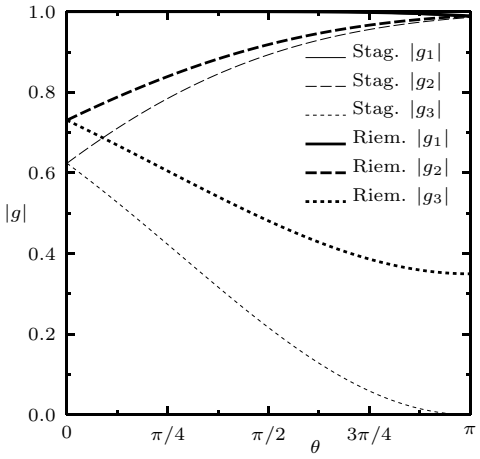
In cases where $\mathcal{O}(h^{2k+1})$ convergence is observed, it is apparent that the $(\cdot)^{\text{evol}}$-norm dictates the convergence. For long-time calculations, there are two main reasons the error is dictated by the $(L_p)^{\text{evol}}$-norm:

1. The spurious modes damp out, so that the contribution of the $(L_p)^{\text{spur}}$-norm vanishes.
2. The $(L_p)^{\text{init}}$-norm represents a one-time contribution to the overall error. Given enough time, the $(L_p)^{\text{evol}}$-norm will overwhelm this contribution, even though the $(L_p)^{\text{evol}}$-norm converges at a faster rate than the $(L_p)^{\text{init}}$-norm.

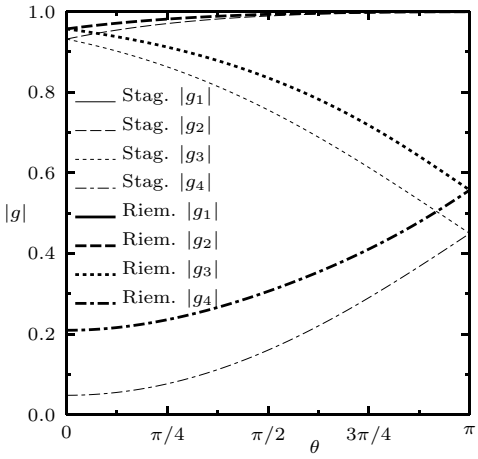
Consider the first item above. As an example of the dependence of the damping on the mesh size, Figure 4.1 shows the damping rates calculated for $\nu = 0.75$. Now, directly comparing $|g_\ell|$ at different Courant numbers can be misleading, particularly for small Courant numbers. For a computation to some



(a) DG(1)



(b) DG(2)



(c) DG(3)

FIG. 4.1. *Damping Rates, $\nu = 0.75$. For DG(2) and DG(3), on the axis scale above, the accurate mode (1-mode) is indistinguishable from $|g| = 1$.*

time $t_{\text{final}} > 0$, the number of time-steps required is given by

$$n = \frac{t_{\text{final}}}{\Delta t} = \frac{a t_{\text{final}}}{\nu h}. \tag{4.43}$$

Since the total damping of the ℓ'th mode is proportional to $|g_\ell|^n$, then for a given h and t_{final}, a useful

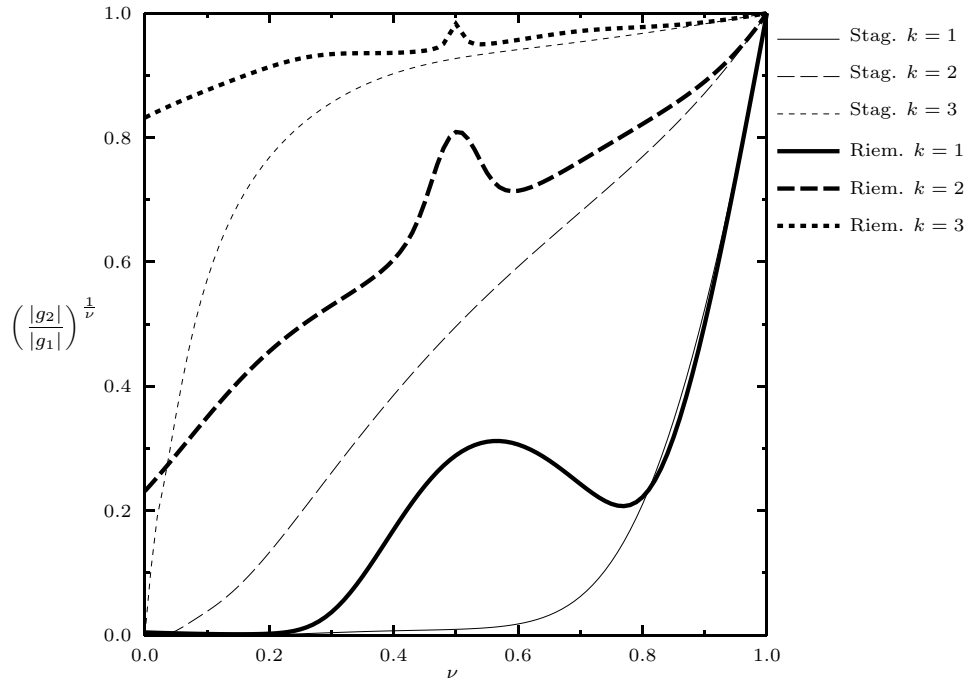


FIG. 4.2. *Damping Measure of 2-mode Relative to Accurate Mode for a Fixed Computation Time, Staggered and Riemann Meshes,* $\theta = \pi/4$.

measure of the damping is the quantity

$$|g_\ell|^{\frac{1}{\nu}} . \tag{4.44}$$

The above quantity allows for a meaningful comparison of the damping for different Courant numbers. Figure 4.2 shows the damping measure (4.44) for the 2-mode relative to the accurate mode as a function of ν. Note that by definition the 2-mode is the least-damped spurious mode.

A negative conclusion of Figures 4.1 and 4.2 is that as k increases, the damping of the spurious modes decreases. However, the initial magnitudes of the spurious modes also decrease with increasing k. After all, $(L_p)^{\text{spur}}_{\max} = \mathcal{O}(\theta^{k+1})$. Another concern is when the 2-mode is damped at the same rate as the accurate mode, which occurs in the following instances:

1. $\underline{\theta = \pm\pi, \text{ for all } \nu}$. This condition corresponds to two cells per wavelength, the highest frequency resolved by the Fourier analysis. In this case g_2 is the complex conjugate of g_1, so that the 2-mode

moves in the opposite direction of the accurate mode. This property is undesirable. However, for $0 < |\nu| < 1$,

$$|g_1|_{\theta=\pm\pi} < |g_1|_{|\theta|<\pi}.$$

In other words, the highest-frequency accurate mode is the most heavily damped. Note also that for large k, it is feasible that DG(k) could adequately resolve the $\theta = \pm\pi$ modes, but would require $k > 3$, which is beyond the scope of this study.

2. $|\nu| = 1$, for all θ. In this case, all of the g_ℓ are equal with $|g_\ell| = 1$, and DG(k) shifts the t^n-solution exactly by a single-mesh cell. This is a desirable property.

Therefore, the only problematic case is when $\theta = \pm\pi$. The hope is that for a general solution made up of all Fourier modes, the contribution of the $\theta = \pm\pi$ modes are small; that is, that the solution is at least moderately resolved.

As an aside, for the Riemann Mesh at $\nu = 0$, all of the g_ℓ are equal with $|g_\ell| = 1$. This property is similar to Item 2 above, but in this case the DG(k) solution is stationary. However, on the Staggered Mesh, DG(k) at $\nu = 0$ damps *all* of the modes [25], which is a property of any time-staggered (Lax-Friedrichs based) method. At low Courant numbers, the damping results in diffusion, with consequences that are unavoidable for 2-D problems. See the results of §5.3 for further discussion.

Assuming that $0 < |\nu| < 1$, the condition (4.23) for the spurious modes becomes

$$1 > |g_2| > |g_3| \ldots > |g_{k+1}|.$$

Consequently, from the definition (4.26d),

$$\lim_{n\to\infty} (L_p)^{\text{spur}} = 0. \tag{4.45}$$

Note that for a given computation time, by (4.43), $n \to \infty$ as $h \to 0$. In terms of the time required to damp the spurious modes, let the time t_{damp} be defined such that $(L_2)^{\text{spur}} < \varepsilon$ if $t > t_{\text{damp}}$. Then

$$t_{\text{damp}}(\varepsilon) = \frac{\nu h}{a} \frac{\log\left(\varepsilon \left\|\mathbf{e}_2 \, \mathbf{B}(\xi)\right\|_2^{-1}\right)}{\log|g_2|}. \tag{4.46}$$

Note that $t_{\text{damp}}(\varepsilon) \to 0$ as $h \to 0$.

For superconvergence to occur requires that, roughly speaking,

$$(L_p)^{\text{evol}} > (L_p)^{\text{init}} + (L_p)^{\text{spur}}. \tag{4.47}$$

Because of (4.45), the right-hand side of the above relation decreases with n, whereas $(L_p)^{\text{evol}}$ grows with n. Also, if $|\theta| > 0$ then $|g_1| < 1$, so that by (4.26c)

$$\lim_{n\to\infty} (L_p)^{\text{evol}} = \left\|\mathbf{e}_1 \, \mathbf{B}(\xi)\right\|_p > (L_p)^{\text{init}}.$$

Therefore, coupled with the result (4.45), there exists some n such that (4.47) is satisfied. In fact, using (4.27) and (4.41a), an estimate for when (4.47) is satisfied is

$$n > \mathcal{O}(h^{k+1-(2k+2)}) = \mathcal{O}(h^{-(k+1)}).$$

In terms of the time required, this estimate translates to

$$t > t_{\text{super}} \equiv \mathcal{O}(h^{-k}).$$

We stress that, strictly speaking, for Fourier data DG(k) converges as $\mathcal{O}(h^{k+1})$ in L_p. After all, $t_{\text{super}} \to \infty$ as $h \to 0$. Our only claim is that under certain instances, $\mathcal{O}(h^{2k+1})$ convergence may be realized in the course of finite-mesh refinement.

The arguments above also apply to the $\bar{L}_p$-norm, except that $t_{\text{super}} = \mathcal{O}(h^{-k+1})$. Therefore, superconvergence occurs sooner in $\bar{L}_p$ than L_p.

We conjecture that if the semi-discrete version of DG is used with a $\mathcal{O}(\Delta t^{2k+1})$ accurate time integrator, then superconvergence may be obtained. For example, Atkins and Shu [2] have observed $\mathcal{O}(h^3)$ convergence in the cell-averages for $k = 1$, using third-order time integration.

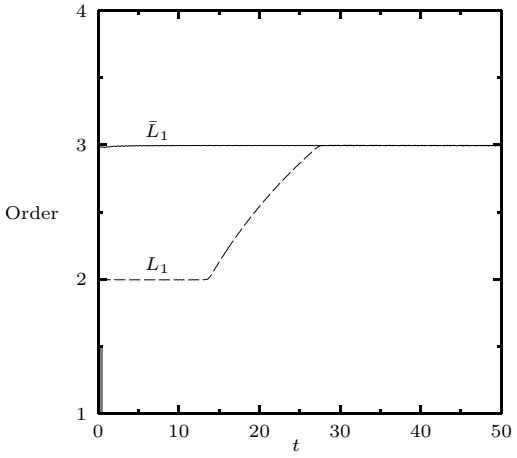
5. Results. In this section, preliminary results are presented for several model problems. The focus will be on confirming the accuracy claims of §4.2. More extensive results will be shown in the companion paper [26] and can also be found in [25, 27].

All of the results presented in this study were computed such that the maximum residual in each element is converged to a tolerance of 10^{-12}. For details on the iteration method, see [25]. We should note that there are several other iteration methods covered in [25]. Some of these methods reduce the cost by not fully converging the residual, yet for many problems these methods have sufficient accuracy. However, for strongly non-linear problems, in order to obtain the superconvergence property, in our experience the residual must be sufficiently converged.

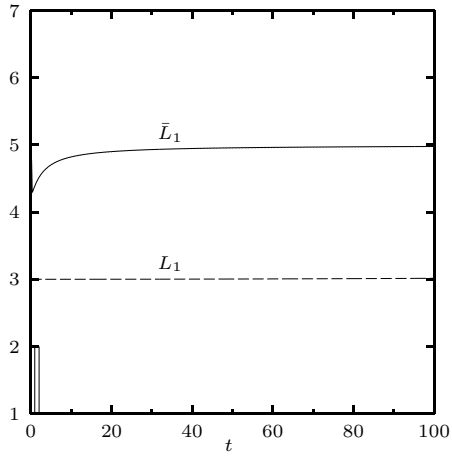
5.1. Scalar Advection. Numerical solutions are presented in this section for eq. (4.16) with periodic boundary conditions, $a = 1$, $0 \leq x \leq 1$, and the initial condition $u^0(x) = \sin(2\pi x)$. Figure 5.1 shows the order-of-accuracy history for the Staggered Mesh, $\nu = 0.8$, using the L_1 and $\bar{L}_1$-norms. Similar results are obtained for the Riemann Mesh. Indicated on each plot is time t_{damp} evaluated at $\varepsilon = 10^{-16}$.

At least for early times, DG(k) follows the Fourier analysis' prediction of $\mathcal{O}(h^{k+1})$ in L_p. An interesting phenomenon can occur, however, as shown by the results for DG(1). Given enough time, the spurious modes damp out, while the evolution error accumulates and overcomes the initial-projection error. Past this time, for a given stage in mesh refinement, the accuracy convergence is dictated by the evolution error, $(L_1)^{\text{evol}}$, which converges as $\mathcal{O}(h^{2k+1})$.

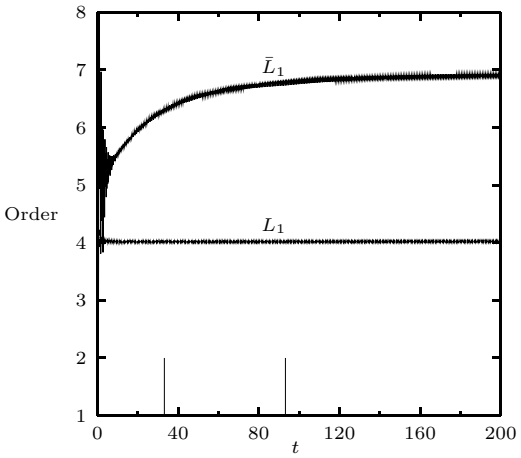
Superconvergence is also evident in the $\bar{L}_1$-norm for $k = 2, 3$. This is because the initial error satisfies $\left(\bar{L}_1\right)^{\text{init}} \ll (L_1)^{\text{init}}$, as discussed in §4.2. Therefore, the order asymptotes to apparent $\mathcal{O}(h^{2k+1})$-accuracy



(a) DG(1), Mesh 40 → 80.



(b) DG(2), Mesh 40 → 80.



(c) DG(3), Mesh 10 → 20.

FIG. 5.1. *Order-of-Accuracy History, Sine Wave, Staggered Mesh, $\nu = 0.8$, using L_1 and $\bar{L}_1$-norms. The two mesh sizes used for the order calculation are indicated. The vertical lines indicate the values of $t_{damp}(10^{-16})$ for the two meshes, where t_{damp} is greatest on the coarser mesh.*

Case	$\log_{10}(L_1)$	$\log_{10}(\bar{L}_1)$
DG(1), Mesh 80, $t = 50$	-3.560815	-3.560852
DG(2), Mesh 80, $t = 100$	-6.084043	-7.511184
DG(3), Mesh 20, $t = 200$	-6.082852	-7.513638

TABLE 5.1
Errors for DG(k) at final time, corresponding to Figure 5.1.

much more quickly in $\bar{L}_1$ than L_1. Note that the time at which the evolution error overcomes the initial-projection error, t_{super}, increases with mesh size, and therefore strictly speaking the convergence in $\bar{L}_1$ is $\mathcal{O}(h^{k+2})$. However, the finest mesh used in each case here gives results that are extremely well-resolved, as shown in Table 5.1. For many practical problems where long-time integration is required, there is the possibility of realizing $\mathcal{O}(h^{2k+1})$-accuracy, particularly in $\bar{L}_p$.

Referring to Figure 5.1c, our conjecture is that the oscillations at early time are caused by the spurious modes.

5.2. Simple Wave Solutions. In this section, solutions to the 1-D Euler equations are presented. The numerical results given in this section will show how nonlinear effects change the error estimates stated in §4.2. The error estimates were derived assuming that spurious modes are only created during the initial projection, and that the accurate and spurious modes do not interact. For nonlinear problems, this is not the case, as spurious modes are excited at each time-step.

Consider the solution domain $0 \leq x \leq 1$, with periodic boundary conditions, and the initial condition

$$\frac{u^0(x)}{a_\infty} = M_\infty + \frac{\cos(2\pi x)}{(\gamma+1)\pi t_{\text{shock}}}, \tag{5.1a}$$

$$\frac{a^0(x)}{a_\infty} = 1 + \frac{\gamma-1}{2}\left(\frac{u^0(x)}{a_\infty} - M_\infty\right), \tag{5.1b}$$

where u is the velocity, a the soundspeed, γ the ratio of specific heats, M the Mach number, and t_{shock} is the time at which a shock forms. The '∞'-conditions correspond to the mean flow state. The initial pressure and density are computed from isentropic conditions:

$$\frac{p^0(x)}{p_\infty} = \left(\frac{\rho^0(x)}{\rho_\infty}\right)^\gamma = \left(\frac{a^0(x)}{a_\infty}\right)^{\frac{2\gamma}{\gamma-1}}, \tag{5.1c}$$

where p is the pressure and ρ the density. For $t < t_{\text{shock}}$, the resulting flow is a simple-wave region with an analytic solution.

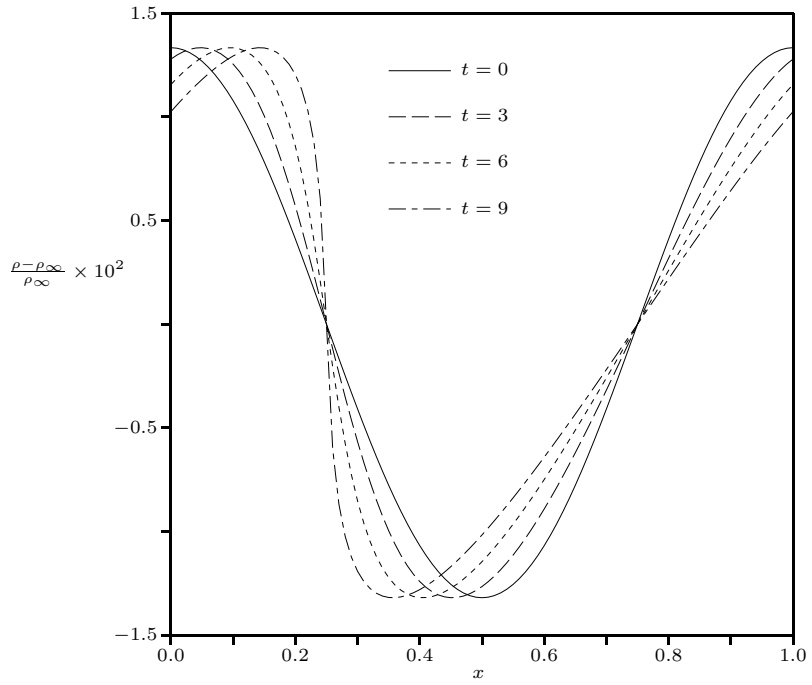


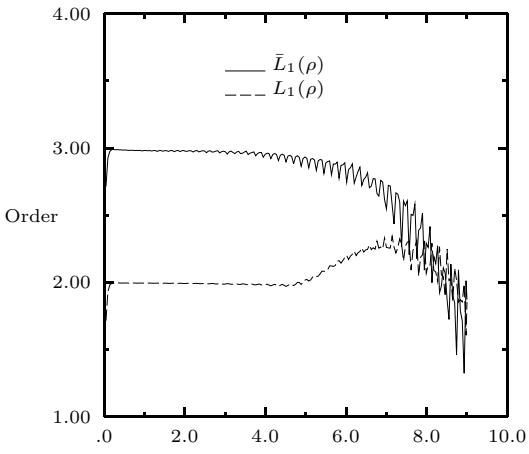
FIG. 5.2. *Exact Solution for Simple-Wave Problem,* $M_\infty = 1$, $t_{shock} = 10$, $\gamma = 1.4$. *Note that each wave has traveled t-periods.*

An exact solution is shown in Figure 5.2. Solutions for DG(k), $k = 1, 2, 3$, were generated using the Staggered Mesh and the order-of-accuracy history is shown in Fig. 5.3. The order is computed using both the density $\bar{L}_1$ and L_1-norms. In $\bar{L}_1$, the results roughly follow a $\mathcal{O}(h^{2k+1})$ convergence rate for early times, but then loose this property as the wave steepens. Only $\mathcal{O}(h^{k+1})$ convergence is realized in $L_1(\rho)$. For this problem, apparently the wave steepens before the superconvergence criterion (4.47) can be satisfied in the $L_1(\rho)$-norm.

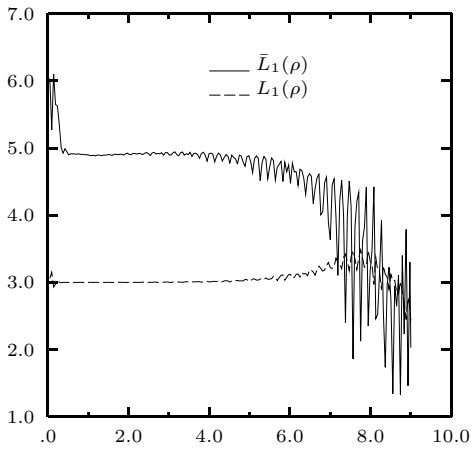
Note that for this problem, the Riemann Mesh gives nearly identical results as the Staggered Mesh [25].

At this point, it is still not clear that superconvergence is lost solely because of wave steepening, or if there is also a contribution from the accumulation of error. Moreover, all problems thus far used periodic boundary conditions. To help resolve this issue, consider the initial condition

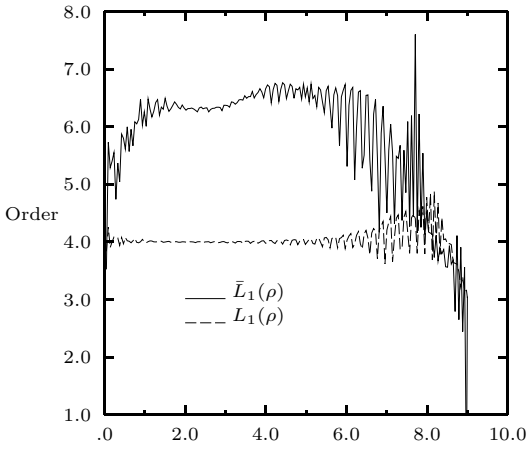
$$\frac{u^0(x)}{a_\infty} = M_\infty + \frac{0.04 \tanh\left(10(x - \frac{1}{2})\right)}{\gamma + 1},$$



(a) DG(1)



(b) DG(2)



(c) DG(3)

FIG. 5.3. *Order-of-Accuracy History, Simple-Wave Expansion, Staggered Mesh, $\nu = 0.9$, using L_1 and $\bar{L}_1$-norms. Mesh sizes 40 and 80 were used for the order calculation.*

with the remaining conditions computed the same as (5.1). A standard characteristic boundary condition is used at each endpoint; see [25] for the details. With $M_\infty = -1$, the resultant expansion wave remains centered in the domain. Figure 5.4 shows the initial condition and results at $t = 10$ for DG(3) on the Staggered Mesh, using 25 cells.

The order-of-accuracy history is plotted in Figure 5.5. For this problem, apparently the evolution error dominates and superconvergence is obtained for all k, even in L_1. Again, we stress that the *asymptotic*

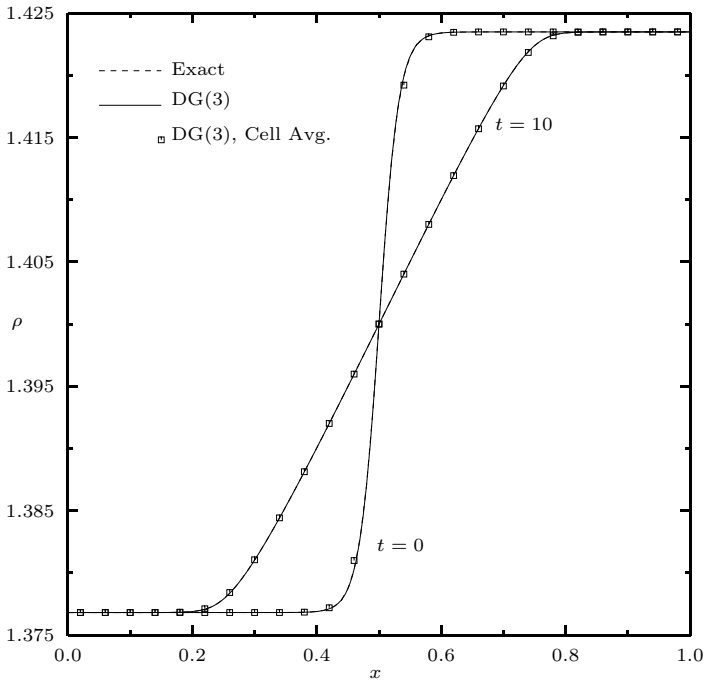


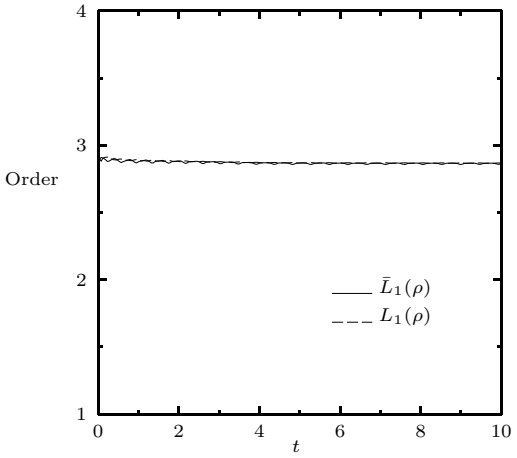
FIG. 5.4. *DG(3) Solution for Simple-Wave Expansion, Staggered Mesh, 25 cells, $\nu = 0.9$, $\gamma = 1.4$, $t = 0, 10$. The DG(3) and exact solutions are coincident on this scale.*

convergence of DG(k) in L_1 is at best $\mathcal{O}(h^{k+1})$. Nevertheless, h is small enough here that the problem is well resolved, as shown in Table 5.2 where the final errors on the finest mesh are given.

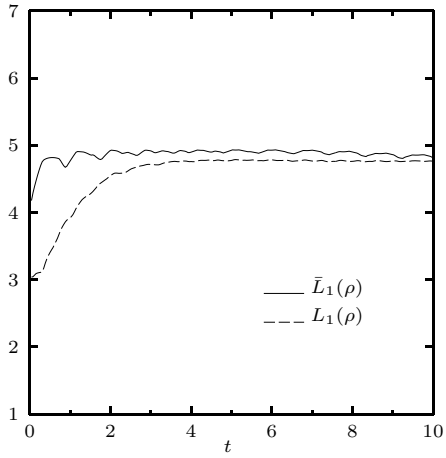
Case	$\log_{10}((L_1(\rho))$	$\log_{10}(\bar{L}_1(\rho))$
DG(1), Mesh 200	-5.311639	-5.311261
DG(2), Mesh 100	-7.360227	-7.325360
DG(3), Mesh 50	-8.065886	-7.951923

TABLE 5.2
Errors for DG(k) at $t = 10$, corresponding to Figure 5.5.

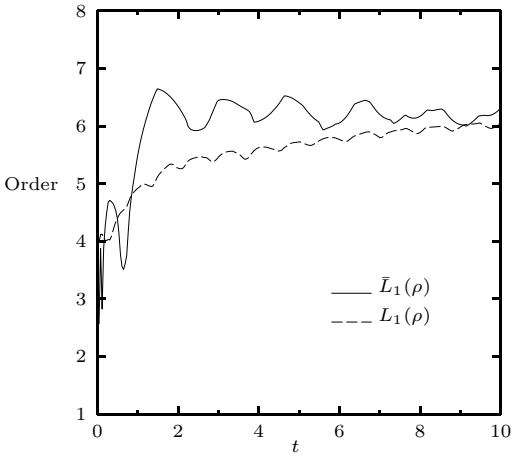
The results of this section show that given a long enough integration time, the benefits of superconvergence can be realized for nonlinear problems. It is encouraging that nonlinear effects by themselves do not destroy superconvergence. Once a wave profile steepens beyond what can be smoothly represented on the mesh, the superconvergence breaks down, but this is to be expected.



(a) DG(1), Mesh 100 → 200.



(b) DG(2), Mesh 50 → 100.



(c) DG(3), Mesh 25 → 50.

FIG. 5.5. *Order-of-Accuracy History, Simple-Wave Expansion, Staggered Mesh, $\nu = 0.9$, using L_1 and $\bar{L}_1$-norms. The two mesh sizes used for the order calculation are indicated.*

5.3. 2-D Scalar Advection. Consider the linear advection equation,

$$\partial_t u + \cos(\alpha)\partial_x u + \sin(\alpha)\partial_y u = 0. \tag{5.2}$$

The solution domain is $0 \leq x, y \leq 1$, with periodic boundary conditions, and an initial condition

$$u(\underline{x}, 0) = \sin(2\pi x)\sin(2\pi y).$$

In §3.1.2 the claim was made that the accuracy of DG(k) is fairly independent of how the advection direction, α, is oriented on the mesh. The numerical results of this section will discuss in what sense this α-independence is true.

Results here are compared with the Hancock finite-volume method [39]. Hancock is a fully-discrete method that is second-order accurate in both space and time. In 1-D, the method reduces to Fromm's scheme, which is designed to have low phase error [17]. Note that for eq. (5.2) solved on the Staggered Mesh, the Courant condition (2.8) requires that

$$\Delta t_{\mathrm{DG}} \leq \frac{h}{\max(|\sin(\alpha)|, |\cos(\alpha)|)},$$

while for the Hancock method

$$\Delta t_{\mathrm{Hancock}} \leq \frac{h}{|\sin(\alpha)| + |\cos(\alpha)|}.$$

Figure 5.6 shows the accuracy of DG(1) and Hancock as a function of α, on a 10×10-cell mesh, computed to a time $t = 10$. The advection direction is aligned with the mesh at $\alpha = 0$. Since the stability constraint of each method is a function of α, as α is varied, either the Courant number or the time-step may be fixed. For

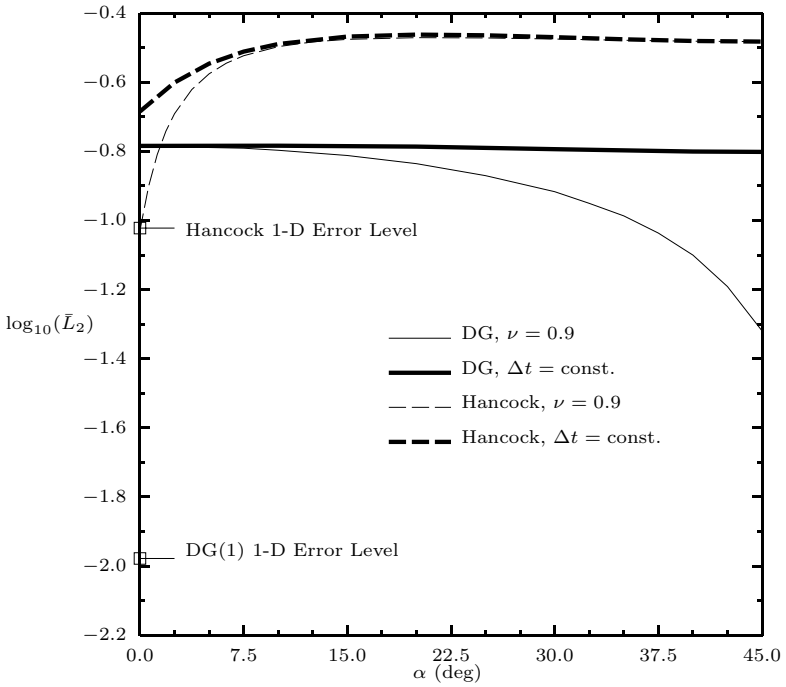


FIG. 5.6. *Accuracy of DG(1) and Hancock as a Function of Advection Direction, α. The Δt = const. curves denote a time-step corresponding to* $\max_\alpha(\nu) = 0.9$. *The 1-D Error Level indicates the error for the equivalent 1-D problem.*

more complicated conservation laws, a fixed time-step could correspond to problems where the maximum wavespeed is associated with some other wave family, or omni-directional waves. Fixing the Courant number is relevant to advection-dominated problems. Figure 5.6 shows the error levels for a fixed $\nu = 0.9$, and for a fixed time-step.

For a fixed time-step, DG(1) is fairly insensitive to α. Note that in this case, DG's time-step is restricted by its $\alpha = 0$ value, while Hancock is restricted by its $\alpha = 45°$ value. For a fixed Courant number, DG can take its largest time-step at $\alpha = 45°$, and therefore requires fewer time-steps to reach the final time level. Consequently, the error is lowest at $\alpha = 45°$.

For both cases, Hancock has lowest error in the mesh-aligned case. It may then be tempting to conclude that this shows that DG(1) is superior, since it shows little preference for the mesh-aligned case. However, a more pessimistic conclusion can be also be drawn. As mentioned in §4.2.1, the DG(k) method on the 1-D Staggered Mesh is diffusive at a Courant number of zero. In 2-D, this symptom also means that in the

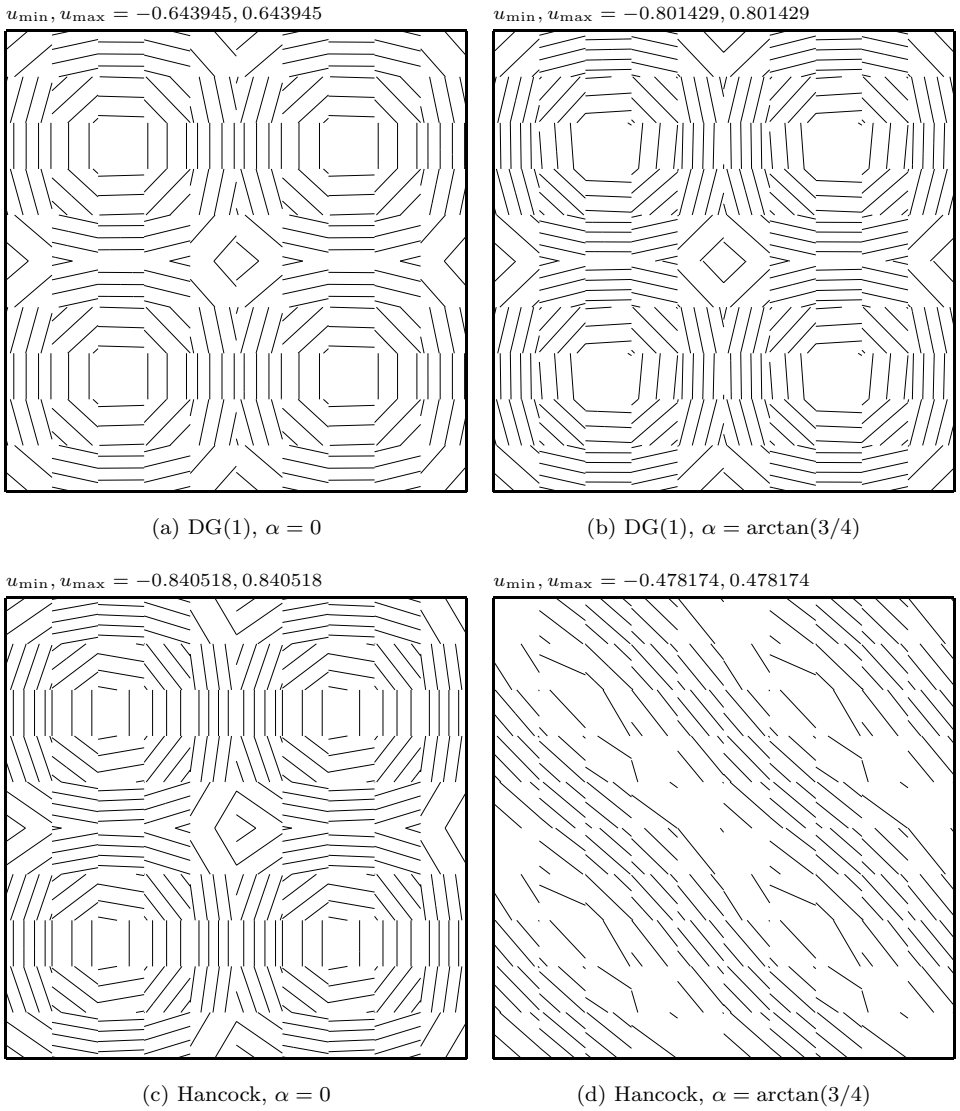
mesh-aligned case ($\alpha = 0°, 90°$), for any Courant number, DG(k) will have significant "cross-diffusion" in the advection-normal direction ($\alpha \pm 90°$). Hancock has no cross-diffusion in the mesh-aligned case. This is clearly shown in Figure 5.6 where the error levels for the equivalent 1-D problem at $\nu = 0.9$ are indicated; DG's 1-D error level is over an order-of-magnitude lower than its 2-D mesh-aligned result. The conclusion here is that even for simple problems, the drawbacks of mesh staggering are unavoidable in 2-D; the Courant number can not be increased to eliminate the problem, unlike in the 1-D case. Cross-diffusion is inevitable for *any* time-staggered method.

On the positive side for DG(k), the benefits of zero cross-diffusion are quickly lost by Hancock as the advection direction becomes misaligned with the mesh. For many practical problems, DG's cross diffusion should be tolerable. If not, one possible remedy is to use a Riemann Mesh in 2-D; or, at least use the Riemann Mesh in critical areas, such as for mesh-aligned shear flows. The present study implements this approach in a limited manner by requiring Riemann Faces at all mesh boundaries.

For this problem we also observe $\mathcal{O}(h^{2k+1})$ convergence for DG(k), $k = 1, 2, 3$ [25]. As a consequence, for mesh sizes larger than 25×25 and a fixed Courant number, DG(1) has a lower error level than Hancock for all α.

Sample numerical results for this problem are given in Figure 5.7. For $\alpha = 0$, the two methods give similar results, except that DG(1) has more damping. On the other hand, for $\alpha = \arctan(3/4)$, the Hancock results are very distorted. A Fourier analysis helps explain this distortion [25].

6. Summary. A time-accurate method has been developed that is based on the Discontinuous Galerkin method. Results were shown to verify that $\mathcal{O}(h^{2k+1})$-convergence occurs for certain long-time calculations. A Fourier analysis was given that argues why this superconvergence occurs. The analysis showed that the evolution error can dominate the error-convergence rate in the course of finite-mesh refinement, and that this error component converges as $\mathcal{O}(h^{2k+1})$. A more rigorous analysis is needed to determine the precise conditions for superconvergence to be realized. For any order-of-accuracy and a convex flux, we proved in the scalar case that the method is L_2-stable for Courant numbers less than 1 and satisfies an entropy condition.



(a) DG(1), $\alpha = 0$

(b) DG(1), $\alpha = \arctan(3/4)$

(c) Hancock, $\alpha = 0$

(d) Hancock, $\alpha = \arctan(3/4)$

FIG. 5.7. *Comparison of 2-D Scalar Advection Results, DG(1) and Hancock.* 10×10 *mesh,* $\nu = 0.9$*,* $t = 10$*. The contours are of the method's polynomial representation within each cell, which for Hancock is its reconstruction.*

A comparison of DG(k) with the Runge-Kutta version (DGRK) shows three advantages. The first advantage is that the Courant restriction for DGRK(k) is inversely proportional to k (at least for small k), whereas the space-time version is stable for Courant numbers less than 1. Secondly, DG(k) has a superconvergence property, although there is some evidence that DGRK(k) with a $\mathcal{O}(\Delta t^{2k+1})$ time integrator may give the same benefit. Thirdly, the space-time version gives a straightforward recipe to obtain a high order-of-accuracy in both space and time. Where the multi-stage version clearly shows an advantage is in its

simplicity, which we would certainly concede that unless time accuracy is critical, outweighs the advantages of the present space-time version.

The use of the Staggered Mesh allowed for the elimination of the Riemann problem. An analogous idea has been studied by many others [1, 28, 36]. The disadvantage of *any* staggered mesh approach is the presence of diffusion at low Courant numbers. Although seemingly innocuous in 1-D problems, in 2-D the diffusion effects are unavoidable, resulting in 'cross-diffusion' even when the mesh lines are aligned with the characteristics.

In the companion paper [26], we show more results for the 1-D Euler equations, including those with a limiter we have developed. 2-D Euler results will also be shown. Additional results can also be found in [25, 27].

References.

[1] P. Arminjon, M. C. Viallon, and A. Madrane. A finite-volume extension of the Lax-Friedrichs and Nessyahu-Tadmor schemes for conservation-laws on unstructured grids. *International Journal of Computational Fluid Dynamics*, 9(1):1–22, 1997.

[2] H. Atkins and C. W. Shu. Quadrature-free implementation of discontinuous Galerkin method for hyperbolic-equations. *AIAA Journal*, 36(5):775–782, 1998.

[3] P. Bar-Yoseph and D. Elata. An efficient L_2 Galerkin finite element method for multi-dimensional non-linear hyperbolic systems. *International Journal for Numerical Methods in Engineering*, 29:1229–1245, 1990.

[4] F. Bassi and S. Rebay. A high-order accurate discontinuous finite-element method for the numerical-solution of the compressible Navier-Stokes equations. *Journal of Computational Physics*, 131(2):267–279, 1997.

[5] F. Bassi and S. Rebay. High-order accurate discontinuous finite-element solution of the 2D Euler equations. *Journal of Computational Physics*, 138(2):251–285, 1997.

[6] C. E. Baumann. *An HP-Adaptive Discontinuous Finite Element Method for Computational Fluid Dy-*

namics. PhD thesis, University of Texas, Austin, 1997.

[7] K. S. Bey and J. T. Oden. HP-version discontinuous Galerkin methods for hyperbolic conservation-laws. *Computer Methods in Applied Mechanics and Engineering*, 133(3-4):259–286, 1996.

[8] K. S. Bey, J. T. Oden, and A. Patra. A parallel HP-adaptive discontinuous Galerkin method for hyperbolic conservation-laws. *Applied Numerical Mathematics*, 20(4):321–336, 1996.

[9] R. Biswas, K. D. Devine, and J. E. Flaherty. Parallel, adaptive finite element methods for conservation laws. Technical Report 93-5, Rensselaer Polytechnic Institute, Department of Computer Science, 1993.

[10] B. Cockburn, S. Hou, and C. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multi-dimensional case. *Mathematics of Computation*, 54:545–581, 1990.

[11] B. Cockburn, S. Lin, and C. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems. *Journal of Computational Physics*, 84:90–113, 1989.

[12] B. Cockburn and C. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework. *Mathematics of Computation*, 52:411–435, 1989.

[13] B. Cockburn and C. W. Shu. The local discontinuous Galerkin method for time-dependent convertion-diffusion systems. *SIAM Journal on Numerical Analysis*, 35(6):2440–2463, 1998.

[14] B. Cockburn and C. W. Shu. The Runge-Kutta discontinuous Galerkin method for conservation-laws V: Multidimensional systems. *Journal of Computational Physics*, 141(2):199–224, 1998.

[15] K. M. D. Devine. *An Adaptive HP-Finite Element Method with Dynamic Load Balancing for the Solution of Hyperbolic Conservations Laws on Massively Parallel Computers.* PhD thesis, Rensselaer Polytechnic Institute, 1994.

[16] R. J. DiPerna. Convergence of approximate solutions to conservations laws. *Archive for Rational Mechanics and Analysis*, 82:27–70, 1983.

[17] J. E. Fromm. A method for reducing dispersion in convective difference schemes. *Journal of Computa-*

tional Physics, 3:176–189, 1968.

[18] J. Jaffre, C. Johnson, and A. Szepessy. Convergence of the discontinuous Galerkin finite element method for hyperbolic conservation laws. *Mathematical Models and Methods in Applied Sciences*, 5:367–386, 1995.

[19] G. Jiang and C. W. Shu. On cell entropy inequality for discontinuous Galerkin methods. *Mathematics of Computation*, 62:531–538, 1994.

[20] C. Johnson. A new approach to algorithms for convection problems which are based on exact transport + projection. *Computer Methods in Applied Mechanics and Engineering*, 100:45–62, 1992.

[21] C. Johnson and J. Pitkaranta. An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation. *Mathematics of Computation*, 46:1–26, 1986.

[22] P. D. Lax. Weak solutions of nonlinear hyperbolic equations and their numerical computation. *Communications on Pure and Applied Mathematics*, VII, 1954.

[23] P. D. Lax. *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves*, volume 11 of *CMBS-NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia, PA., 1973.

[24] P. Lesaint and P. Raviart. On a finite element method for solving the neutron transport equation. In C. de Boor, editor, *Mathematical Aspects of Finite Elements in Partial Differential Equations*, 1974.

[25] R. B. Lowrie. *Compact Higher-Order Numerical Methods for Hyperbolic Conservation Laws*. PhD thesis, University of Michigan, 1996. Availiable at http://www.engin.umich.edu/research/cfd/research/-publications/.

[26] R. B. Lowrie, P. L. Roe, and B. van Leer. Numerical results of the space-time discontinuous Galerkin method. In preparation, 1998.

[27] R. B. Lowrie, P. L. Roe, and B. van Leer. Space-time methods for hyperbolic conservation laws. In *Barriers and Challenges in Computational Fluid Dynamics*, volume 6 of *ICASE/LaRC Interdisciplinary Series in Science and Engineering*, pages 79–98. Kluwer, 1998.

[28] H. Nessyahu and E. Tadmor. Non-oscillatory central differencing for hyperbolic conservation laws. *Journal of Computational Physics*, 87:408–463, 1990.

[29] J. T. Oden. A general theory of finite element I; II. *International Journal for Numerical Methods in Engineering*, 1:205–221;247–259, 1969.

[30] J. T. Oden, I. Babuska, and C. E. Baumann. A discontinuous HP finite-element method for diffusion-problems. *Journal of Computational Physics*, 146(2):491–519, 1998.

[31] S. Osher. Riemann solvers, the entropy condition, and difference approximations. *SIAM Journal on Numerical Analysis*, 21:217–235, 1984.

[32] P. Perrochet. A streamline-upwind-full-Galerkin method for space-time convection dominated transport problems. *International Journal for Numerical Methods in Engineering*, 36:4165–4183, 1993.

[33] W. F. Reed and T. R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos National Laboratory, 1973.

[34] G. R. Richter. An optimal-order error estimate for the discontinuous Galerkin method. *Mathematics of Computation*, 50:75–88, 1988.

[35] P. L. Roe. Approximate Riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics*, 43, 1981.

[36] R. Sanders and A. Weiser. A high resolution staggered mesh approach for nonlinear hyperbolic systems of conservation laws. *Journal of Computational Physics*, 101, 1992.

[37] F. Shakib. *Finite Element Analysis of the Compressible Euler and Navier-Stokes Equations*. PhD thesis, Stanford University, 1988.

[38] F. Shakib and T. J. R. Hughes. A new finite element formulation for computational fluid dynamics: IX. Fourier analysis of space-time Galerkin/least-squares algorithms. *Computer Methods in Applied Mechanics and Engineering*, 87:35–58, 1991.

[39] G. D. van Albada, B. van Leer, and W. W. Roberts. A comparative study of computational methods in cosmic gas dynamcs. *Astronomy and Astrophysics*, 108:76–84, 1982.

[40] B. van Leer. Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection. *Journal of Computational Physics*, 23, 1977.