JOURNAL OF COMPUTATIONAL PHYSICS **144,** 325–357 (1998)
ARTICLE NO. CP975831

Spectral/hp Methods for Viscous Compressible Flows on Unstructured 2D Meshes

I. Lomtev, C. B. Quillen, and G. E. Karniadakis*

Center for Fluid Mechanics, Division of Applied Mathematics, Brown University, Providence, Rhode Island 02912
E-mail: gk@cfm.brown.edu

Received December 16, 1996

In this paper we describe the foundation of a spectral/hp method suitable for simulating viscous compressible flows with shocks on standard unstructured meshes. It is based on a discontinuous Galerkin formulation for the hyperbolic contributions combined with a mixed Galerkin formulation for the diffusive contributions. High-order accuracy is achieved by using a recently developed hierarchical spectral basis. This basis is formed by combining Jacobi polynomials of high-order weights written in a new coordinate system that retains a tensor product property and accurate numerical quadrature. The formulation is conservative, and monotonicity is enforced by high-order limiters and by appropriately lowering the basis order around discontinuities. Convergence results are shown for benchmark solutions of the advection, Euler, and Navier–Stokes equations that demonstrate exponential convergence of the new method. Flow simulations for subsonic and supersonic flows are also presented that demonstrate discretization flexibility using $h - p$ type refinement. Unlike other high-order methods the new method uses standard finite volume meshes consisting of arbitrary triangulizations. © 1998 Academic Press

1. INTRODUCTION

There has been recently an interest in computational aerodynamics to extend finite volume methods to high-order accuracy. This is due primarily to the shift of emphasis from steady inviscid Euler flow simulations toward accurate simulations of *time-dependent, viscous* flows (see [1, 2]). Also, new fields such as computational electromagnetics for aerospace design involve the solution of time-dependent highly oscillatory solutions for which high-order discretization is more efficient [3]. In particular, for the *long-time* integration of time-dependent solutions it has been argued in [4] that high-order numerical methods provide the most cost-effective approach.

---

* Author to whom all correspondence should be addressed.

There are several fundamental issues that limit such a straightforward extension of finite volume or other low-order methods to high-order in the context of aerodynamic simulations. First, *monotonicity* is not preserved in high-order methods in discretizing hyperbolic conservation laws. Second, *conservativity* is not easily implemented. Third, *geometric complexity* requires the use of unstructured meshes. Fourth, *computational complexity* is increased significantly.

Despite these difficulties, progress has been made in the last few years in addressing these issues. For shock-fitting methods the multidomain spectral methods developed by Kopriva [5] have been successful in simulating very accurately supersonic flows at very high Mach numbers [6]. However, their generality is somewhat limited as shock-fitting methods work best for well defined sharp shocks and relatively regular geometries. For shock-capturing methods, the issue of monotonicity and the associated Gibbs phenomena caused by solution discontinuities has been addressed in [7], where nonoscillatory reconstruction algorithms were developed and implemented in the spectral element context in [8]. Their implementation, however, in multidimensions is quite difficult. A more robust method was developed in [9], where a flux-corrected-transport (FCT) limiter was combined with spectral element discretizations but its computational complexity was two to three times higher than standard low-order methods.

In these approaches as well as in the work of [10], staggered grids are used to preserve conservativity, assigning fluxes on one grid and cell averages on the other. This too introduces extra computational complexity as it relies on expensive cell averaging and reconstruction procedures. A novel spectral multidomain technique was proposed more recently in [11, 12] based on the penalty method [13], but this scheme does not preserve conservativity.

High-order methods have been used extensively in transition and turbulence simulations both for incompressible as well as compressible flows [14, 15], but they are practically limited to simple geometries and they require special meshes. The spectral element method, as it was first developed [16, 17], employs a *nodal* spectral basis which, in practice, necessitates the use of relatively undeformed subdomains. For a new numerical method to become useful for CFD problems of industrial complexity, it has to utilize the *existing* technology of mesh generators for unstructured and hybrid meshes [18–20].

In previous work [21, 22] we developed a spectral/hp Galekin method for the numerical solution of the two- and three-dimensional unsteady *incompressible* Navier–Stokes equations on unstructured meshes. This formulation was implemented in the code $\mathcal{N}\varepsilon\kappa\mathcal{T}\alpha r$. A similar approach was used in [23] in the context of geophysical fluid dynamics applications. The discretization is based on arbitrary triangulizations/tesselations of (complex-geometry) domains. On each triangle/tetrahedron a spectral expansion basis is employed consisting of Jacobi polynomials of mixed weight that accommodate exact numerical quadrature. The hierarchical expansion basis is of variable order per element and retains the tensor product property (similar to standard spectral expansions), which is key in obtaining computational efficiency via the sum factorization technique.

In the current work we develop a new formulation for *compressible* Navier–Stokes solutions employing the aforementioned hierarchical basis for triangular subdomains. In particular, we develop techniques to deal with monotonicity and conservativity in two-dimensional domains of arbitrary geometric complexity. Unlike the work for incompressible flows where a standard Galerkin formulation was employed, here we use a *discontinuous* Galerkin formulation to treat the hyperbolic contributions and a mixed discontinuous/continuous Galerkin formulation to treat the diffusive contributions. Correspondingly, two sets of basis functions are employed: the first one is a discontinuous orthogonal basis first proposed by Dubiner [24]; and the second one is a C^0 continuous semi-orthogonal basis used in [21]. The conservativity property is maintained automatically by the discontinuous Galerkin formulation, whereas monotonicity is controlled by high-order limiters and/or by varying the order of the the spectral expansion around discontinuities. The formulation for the Euler equations presented here was motivated by the work on discontinuous finite elements presented in a series of papers [25–28]. A similar implementation for quadrilateral Legendre spectral elements was used in [29].

An example of a simulation obtained with the methods developed herein is shown in Fig. 1 that shows a supersonic flow at Mach number $\mathrm{Ma} = 2$ past a NACA4420 airfoil at a large angle of attack. The simulation is time-dependent, but after some time the solution is settled to the steady state shown in the plot. The important point to note here is that this simulation was obtained on the unstructured mesh of Fig. 2, which is typical of the meshes generated using standard mesh generator codes, e.g. [30]. To test convergence of the solution, however, p-refinement is pursued that does not require any remeshing, and thus, it avoids the overhead associated with the mesh generation. We will return to this flexibility in discretization for a similar application in Section 6.

The paper is organized as follows: In Section 2 we present the discontinuous and the C^0-continuous spectral basis. In Section 3 we review the discontinuous Galerking formulation, and in Section 4 we implement it in the context of a multidimensional advection equation; in Section 5 we apply it to the Euler equations. In Section 6 we develop a mixed Galerkin formulation and consider the Navier–Stokes equations. Several convergence results and flow simulations in the subsonic and supersonic regime are presented. We finish in Section 7 with a brief summary.

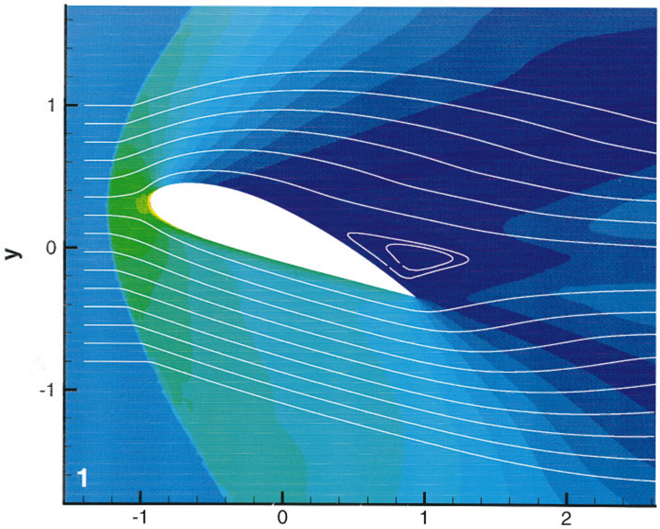
2. SPECTRAL BASES

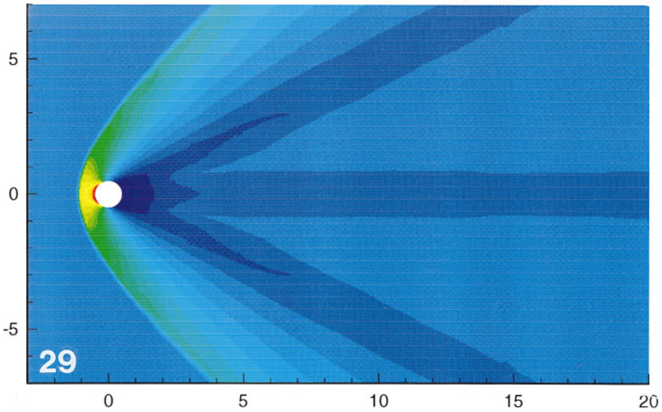
To implement the discontinuous Galerking method [31] using spectral discretizations we need to work with an appropriate expansion basis. To this end, we will adopt the spectral basis for triangles first developed by Dubiner [24]. This polynomial orthogonal basis, however, cannot be used in multidomain discretizations if continuity of functions is required at interelemental interfaces. This is the situation with the Navier–Stokes equations, where a C^0 continuity condition is required in the variational statement. A new basis can then be derived that can accommodate continuity at the expense of partial loss of orthogonality. Such a basis has been developed in [32] in the context of spectral element method. It has been implemented in two and three dimensions in algorithms solving the incompressible Navier–Stokes equations in [21, 22]. In the following, we review these two spectral bases as we will use them both: the first one (discontinuous) in the context of the Euler equations; and the second one (continuous) in the context of the Navier–Stokes equations.

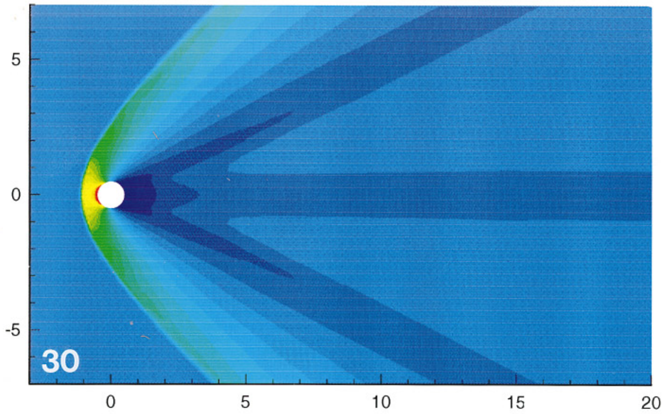
We first define a set of mappings that are useful in defining the triangular bases in terms of Cartesian coordinates attached to the transformed domain. We define the standard triangular and rectangular domains as shown in Fig. 3, which are mathematically expressed as

$$T \equiv \{(r, s) | -1 \leq r, s; r + s \leq 0\}$$

$$R \equiv \{(a, b) | -1 \leq a, b \leq 1\}.$$


1
y
1
0
-1
-1
0
1
2


29
5
0
-5
0
5
10
15
20


30
5
0
-5
0
5
10
15
20

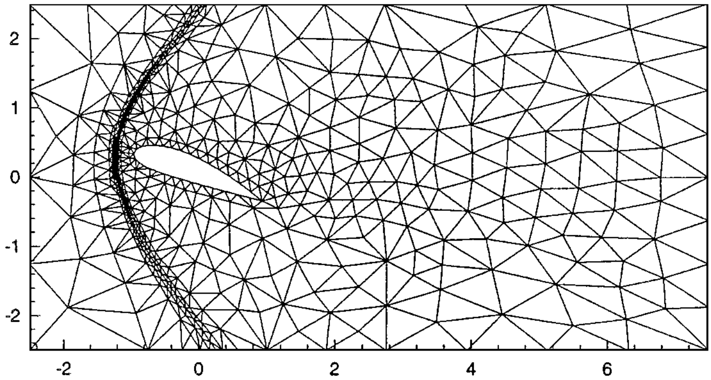


FIG. 2. Unstructured mesh for the NACA4420 airfoil supersonic flow shown in the previous figure. The number of elements in 1128 and each element can support different p-order.

The rectangular domain R can be mapped into the triangular domain T by the transformation:

$$\begin{aligned} s &= b, \\ r &= \frac{(1+a)(1-b)}{2} - 1, \end{aligned}$$

and, similarly, the triangular domain T can be mapped into the rectangular domain R by the inverse transformation

$$\begin{aligned} b &= s, \\ a &= 2\frac{1+r}{1-s} - 1. \end{aligned} \tag{1}$$

2.1. Discontinuous Basis

We wish to define a polynomial basis, denoted by $\phi_{lm}(r, s)$, so that we can approximate the function $f(r, s)$ in the domain T, i.e.

$$f(r, s) = \sum_{l} \sum_{m} \bar{f}_{lm} \phi_{lm}(r, s).$$

Here $\bar{f}_{lm}$ is the expansion coefficient corresponding to polynomial ϕ_{lm} and (r, s) are the local coordinates within the triangle T. The polynomial expansion basis for triangular domains expressed in [24] is orthogonal in the Legendre inner product. The principal idea

---

FIG. 1. Supersonic flow past a NACA4420 airfoil at 20° angle of attack and Mach number $Ma = 2$. Density contours and streamlines are plotted.

FIG. 29. Density countours for supersonic flow ($\mathrm{Ma} = 2$) past a cylinder. Low resolution with second-order elements in the wake.

FIG. 30. Density countours for supersonic flow ($\mathrm{Ma} = 2$) past a cylinder. High resolution with fifth-order elements in the wake.

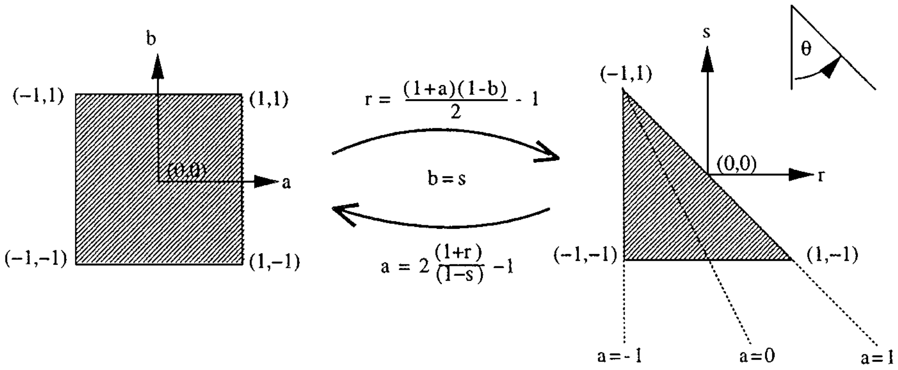


FIG. 3. General rectangle to triangle transformation.

is to express the expansion basis in terms of a function, which is a polynomial in both the standard coordinates and the transformed coordinates. A Gram–Scmidt orthonormalization procedure of polynomial powers in the standard triangle applied in the right order results in the Appel polynomials that form the basis. We briefly review this basis next.

Let us denote by $P_l^{\alpha,\beta}(x)$ the nth-order Jacobi polynomial in the $[-1, 1]$ interval with the orthogonality relationship,

$$\int_{-1}^{1} P_l^{\alpha,\beta}(x) P_m^{\alpha,\beta}(x)(1-x)^{\alpha}(1+x)^{\beta}\, dx = \delta_{lm}, \tag{2}$$

where δ_{lm} is the Kronecker delta. The triangular orthogonal expansion basis is given by

$$\phi_{lm}(r,s) = P_l^{0,0}\left(2\frac{(1+r)}{(1-s)} - 1\right)(1-s)^l P_m^{2l+1,0}(s).$$

We note that this is a polynomial in (r, s) since the $(1-s)^l$ factor acting on the $P_l^{0,0}(2((1+r)/(1-s)) - 1)$ Jacobi polynomial produces an lth-order polynomial in (r, s). (Note that $P^{0,0}$ is the often-used Legendre polynomial.) The basis can also be expressed as the product of two polynomials in (a, b) space; i.e.,

$$\phi_{lm}(r,s) = \phi_l^1\left(2\frac{(1+r)}{(1-s)} - 1\right)\cdot \phi_{lm}^2(s) = \phi_l^1(a)\cdot\phi_{lm}^2(b)$$

where

$$\phi_l^1(a) = P_l^{0,0}(a), \quad \phi_{lm}^2(b) = (1-b)^l P_m^{2l+1,0}(b).$$

Dubiner [24] refers to this property as a *warped product* to differentiate it from the standard tensor product associated with quadrilateral domains; it is a *generalized* tensor product. The significance to this property is that the inner product between two polynomial bases which both span a two-dimensional space can be expressed as the product of two one-dimensional inner products multiplied by a constant. This is particularly important when evaluating integrals involving $\phi_{lm}(r, s)$ with itself over T, since it is possible to write the integral as the product of two line integrals as explained in [21]. Integrals involving the inner product of $\phi_{lm}(r,s)$ with a function $f(r,s)$ can also be efficiently evaluated using the sum factorization technique. Finally, we note that the basis is complete in a polynomial space $\mathcal{P}_L$, where $\mathcal{P}_L$ is defined by

$$\mathcal{P}_L = \mathrm{Span}\{r^l, s^m\}_{(lm)\in\mathcal{Q}},$$

where

$$\mathcal{Q} = \{(lm) | 0 \le l, m;\, l < L, l+m < M\}, \quad L \le M.$$

We also note that $\phi_{lm}(r,s)$ is orthogonal in the Legendre internal product defined by

$$(\phi_{lm}(r,s), \phi_{pq}(r,s))_T = \int\int_T \phi_{lm}(r,s)\phi_{pq}(r,s)\,dr\,ds = \delta_{lp}\delta_{mq}.$$

This can be evaluated after we apply the transormation $T \mapsto R$.

2.1.1. *Numerical Integration*

Numerical integration will be performed in the rectangle R based on the (a,b) coordinates using some variant of Gaussian quadrature. The two varieties explored here will be Gauss quadrature and Gauss–Lobatto quadrature, both using uniform weight functions. As we shall see, both versions lead to essentially identical numerical methods, with some differences in practical implementation.

Let us begin with the *Gauss quadrature* case. For L quadrature points it is exact for polynomials up to order $(2L-1)$. The ordinates for this quadrature will be denoted by q^L, and w^L will be the weights for $0 \le i \le L-1$. In the a-coordinate direction, the quadrature points will be $a_i = q_i^L$. For the b-direction, the points $b_j = q_j^M$ will be applied. The inner product

$$(\phi_{lm}, \phi_{pq})_T = \frac{1}{2}\int_{-1}^{1}\left(\int_{-1}^{1} P_l^{0,0}(a)P_p^{0,0}(a)\,da\right)(1-b)^{l+p+1}P_m^{2l+1,0}(t)P_q^{2p+1,0}(b)\,db$$

may be exactly computed by the sum

$$\frac{1}{2}\sum_{j=0}^{M-1}\left(\sum_{i=0}^{L-1} P_l^{0,0}\left(q_i^L\right)P_p^{0,0}\left(q_i^L\right)w^L\right)\left(1-q_j^M\right)^{l+p+1}P_m^{2l+1,0}\left(q_j^M\right)P_q^{2p+1,0}\left(q_j^M\right)w_j^M.$$

This implies a discrete inner product,

$$(f,g)_T = \frac{1}{2}\sum_{j=0}^{M-1}\sum_{i=0}^{L-1} w^L w^M f\left(q_i^L, q_j^M\right)g\left(q_i^L, q_j^M\right). \tag{3}$$

Projections may be performed with essentially the same technique using *Gauss–Lobatto* quadrature. At first it might seem that we might need more quadrature points, as Gauss–Lobatto with L quadrature points is only accurate for polynomials of order $(2L-3)$. Two tricks may be used to avoid this. For the a-variable integration, if $g(a)$ is an $(L-1)$ degree polynomial, all inner products,

$$\left(g(a), P_j^{0,0}(a)\right)$$

for $j \leq L-2$ are calculated exactly by Gauss–Lobatto quadrature. Thus, $g(a)$ can be projected on $\{P_0^{0,0} \cdots P_{L-2}^{0,0}\}$ and may be expressed as

$$g(a) = \sum_{j=0}^{L-2} \alpha_j P_j^{0,0}(a) + \alpha_{L-1} P_{L-1}^{0,0}(a).$$

All coefficients α_j are known, except α_{L-1}, which can be solved for by evaluating at a particular point a. Taking $a = 1$ and using the fact that $P_j^{0,0}(1) = 1$, this becomes

$$\alpha_{L-1} = g(1) - \sum_{j=0}^{L-2} \alpha_j.$$

The b-direction also can be integrated using M quadrature points. The trick is to actually use $(M+1)$ and note that the point $b = 1$ is one of these points, and at that point the integrand will always be zero, due to the fact that the Jacobian of the tranformation $T \mapsto R$, which is $J = (1-b)/2$, vanishes there. Thus, that particular quadrature point may be ignored. This is, in fact, equivalent to using Gauss–Radau integration with a weight function of $(1-b)$.

2.1.2. *Matrix Notation*

Having defined the projection, i.e. inner products, and the numerical quadrature we can summarize these operations using matrix formalism. For now, we limit the discussion to the simpler case where Gauss integration is used. As a convention, the mesh points (q_i^L, q_j^M) will be ordered with the i index changing fastest, and they will be written as a vector $\mathbf{x}_k = (q_{k \bmod L}^L, q_{[k/L]}^M)$, for $0 \leq k \leq LM$.

Let

$$E : f(x, y) \mapsto \left(f(\mathbf{x}_0), \ldots, f\left(\mathbf{x}_{L(M_{-1})}\right)\right)^t$$

be the operator that evaluates functions at the grid points. Let W be the diagonal matrix

$$W_{jj} = w_{j \bmod L}^L w_{[j/L]}^M.$$

Then the inner product (3) may be written

$$(f, g)_T = (Eg)^t W E f.$$

Let G be the matrix of basis element values,

$$G_{jk} = v_{(k \bmod L)([k/L])}(\mathbf{x}_j),$$

and S the diagonal inverse mass matrix with ones in the diagonal. Then the projection operator may be written

$$E\mathbb{P}f = GSG^t WEf = E\sum_{l,m} \phi_{lm}(\phi_{lm}, f)_T.$$

2.1.3. *Differentiation*

Derivative operators are most conveniently constructed for the basis $\{\phi_{lm}\}$ by noting that the basis is spanned by the polynomials and by realizing that the derivative matrices derived from polynomial interpolation on the quadrature points may be used to compute a derivative at each grid point.

Let D^L be the differentiation matrix

$$D^L : \begin{pmatrix} P\left(q_0^L\right) \\ \vdots \\ P\left(q_{L-1}^L\right) \end{pmatrix} \mapsto \begin{pmatrix} P'\left(q_0^L\right) \\ \vdots \\ P'\left(q_{L-1}^L\right) \end{pmatrix},$$

where P is a polynomial of order $L-1$ or less (we have dropped the sup-indices), and D^M is the corresponding operator for the points $q_0^M \cdots q_{M-1}^M$. Then the derivative operators D_a and D_b operating on the grid points $\mathbf{x}$ may be defined

$$D_a = \begin{pmatrix} D^L & & & \\ & D^L & & \\ & & \ddots & \\ & & & D^L \end{pmatrix};$$

that is, $(D_a)_{ij} = \delta_{[l/M][j/M]} D^L_{(i \bmod L)(j \bmod L)}$ and

$$(D_b)_{ij} = \delta_{(i \bmod M)(j \bmod M)} D^M_{[i/L][j/L]}.$$

In terms of these operators, the operator $D_r \approx \partial/\partial r$ and $D_s \approx \partial/\partial s$ may be constructed by

$$\frac{\partial}{\partial r} = \frac{\partial a}{\partial r}\frac{\partial}{\partial a} + \frac{\partial b}{\partial r}\frac{\partial}{\partial b} = \frac{2}{1-b}\frac{\partial}{\partial a}$$

and, similarly,

$$\frac{\partial}{\partial s} = \frac{a+1}{1-b}\frac{\partial}{\partial a} + \frac{\partial}{\partial b},$$

so

$$\begin{aligned} D_r &= A D_a \\ D_s &= B D_a + D_b, \end{aligned}$$

where A and B are diagonal and

$$\begin{aligned} A_{jj} &= \frac{2}{1 - q^M_{[j/L]}}, \\ B_{jj} &= \frac{q^L_{j \bmod L} + 1}{1 - q^M_{[j/L]}}. \end{aligned}$$

2.2. Continuous Basis

The high-order discontinuous basis presented above can be modified to produce a C^0 continuous basis for a multidomain discretization. Such a construction can be achieved at the expense of partial orthogonality loss. The key idea is to decompose the basis into three sets of modes in two dimensions i.e., *vertices, edges, and interior*. The interior modes are similar to the modes of the orthogonal basis, the vertex modes are linear, and the edge modes start with quadratic order:

- interior modes $(2 \le l, 1 \le m; l < L, l + m < M)$,

$$\phi_{lm}^{interior} = \left(\frac{1+a}{2}\right)\left(\frac{1-a}{2}\right)P_{l-2}^{1,1}(a)\left(\frac{1-s}{2}\right)^{l}\left(\frac{1+s}{2}\right)P_{m-1}^{2l-1,1}(s);$$

- edge modes $(2 \le l, 1 \le m; l < L, l + m < M)$,

$$\phi_{l}^{side-1} = \left(\frac{1+a}{2}\right)\left(\frac{1-a}{2}\right)P_{l-2}^{1,1}(a) \cdot \left(\frac{1-s}{2}\right)^{l}$$

$$\phi_{1m}^{side-2} = \left(\frac{1+a}{2}\right) \cdot \left(\frac{1-s}{2}\right)\left(\frac{1+s}{2}\right)P_{m-1}^{1,1}(s)$$

$$\phi_{1m}^{side-3} = \left(\frac{1-a}{2}\right) \cdot \left(\frac{1-s}{2}\right)\left(\frac{1+s}{2}\right)P_{m-1}^{1,1}(s);$$

- Vertex modes,

$$\phi^{vert-A} = \left(\frac{1-a}{2}\right) \cdot \left(\frac{1-s}{2}\right)$$

$$\phi^{vert-B} = \left(\frac{1+a}{2}\right) \cdot \left(\frac{1-s}{2}\right)$$

$$\phi^{vert-C} = 1 \cdot \left(\frac{1+s}{2}\right).$$

The location of sides 1, 2, and 3 as well as vertices A, B, and C are indicated in Fig. 4. The interior modes are zero at the boundaries while edge modes have a nonzero value along one

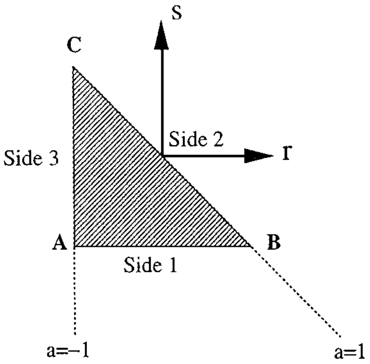


FIG. 4. Definition of the standard triangle and coordinate system.

edge and are zero at all vertices. The vertex modes have a unit value at one vertex and decay linearly to zero at the other vertices. Every mode is a polynomial in (a, b) space as well as (r, s) space, since any lth-order polynomial $f_l(a)$ is a polynomial in (r, s) when multiplied by the factor $(1 - s)^l$. As can be seen all polynomials in the a-variable are multiplied by appropriate factors of $(1 - s)$. As a final point we note that along each side the edge modes have the same shape which allows the basis to be combined to form a C^0 expansion by matching the expansion coefficients of these modes (see [32] for details). This way triangles with a different order per edge can be used, as long as adjacent edges match.

3. DISCONTINUOUS GALERKIN FORMULATION

We now consider the linear two-dimensional equation for advection of a conserved quantity u in a region Ω,

$$\frac{\partial u}{\partial t} + \nabla \cdot \boldsymbol{F}(u) = 0, \tag{4a}$$

where $\boldsymbol{F}(u) = (f(u), g(u))$ is the *flux* vector which defines the transport of $u(\mathbf{x}, t)$. In the standard Galerkin formulation of this equation, u is approximated by u_δ and $u_\delta \in \boldsymbol{\mathcal{X}}_\delta$, where $\boldsymbol{\mathcal{X}}_\delta$ is a finite-dimensional subspace of the space of compactly supported continuous functions. The variational statement of the Galerkin formulation of (4a) is derived by multiplying by a test function v and integrating by parts:

$$\int_\Omega \frac{\partial u_\delta}{\partial t} v\, dx + \int_{\partial\Omega} v\, \hat{n} \cdot \boldsymbol{F}(u_\delta)\, ds - \int_\Omega \nabla v \cdot \boldsymbol{F}(u_\delta)\, dx = 0. \tag{4b}$$

The solution u_δ satisfies this equation for all $v \in \boldsymbol{\mathcal{X}}_\delta$. The requirement that $\boldsymbol{\mathcal{X}}_\delta$ consist of continuous functions naturally leads to a basis consisting of functions with overlapping support, which implies Eq. (4b) leads eventually to inverting a large banded matrix. This is not a trivial task for parallel implementations, and therefore, a different type of formulation is desirable.

Another consideration from the point of view of advection is that continuous function spaces are not the natural place to pose the problem. Mathematically, hyperbolic problems of this type tend to have solutions in spaces of bounded variation. In physical problems, the best one can hope for in practice is that solutions will be piecewise smooth, that is, be smooth in regions separated by discontinuities (shocks). The main consideration is that the formulation presented next preserves conservativity in the element-wise sense automatically, and thus, we avoid dealing with staggered grids as in the formulation developed in [9].

These considerations suggest immediately a formulation where $\boldsymbol{\mathcal{X}}_\delta$ may contain discontinuous functions. These are typically taken to be polynomial functions within each "element," and zero outside the element. Here the "element" is, for example, an individual triangular region T_i in the computational mesh applied to the problem. Thus, the computational domain $\Omega = \cup_i T_i$, and T_i, T_j overlap only on edges as shown in Fig. 5. In summary, the appropriate approximation space is defined as

$$\boldsymbol{\mathcal{X}}_\delta = \{v \in L_2(\Omega) : v|T_i \in \mathcal{P}(T_i)\ \forall T_i\}, \tag{5}$$

where $\mathcal{P}(\Omega)$ is the polynomial space defined on the domain Ω.

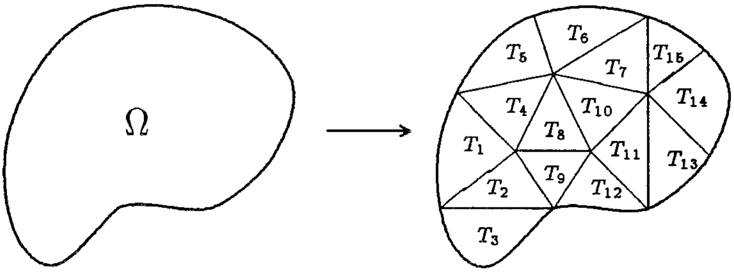


FIG. 5. A computational domain Ω tessellated by triangles T_e.

Contending with the discontinuities requires a somewhat different approach to the variational formulation. Each element is treated separately, giving a variational statement at each element ($u_\delta \in \boldsymbol{\mathcal{X}}_\delta$ and $\forall v \in \boldsymbol{\mathcal{X}}_\delta$),

$$\frac{\partial}{\partial t}(u_\delta, v)_e + \int_{\partial T_e} v\boldsymbol{F}(u_\delta) \cdot \hat{n}\, ds - (\boldsymbol{F}(u_\delta), \nabla v)_e = 0. \tag{6}$$

Computations on each element are performed *separately*, and the connection between elements is a result of the way boundary conditions are applied. Here, boundary conditions are enforced via the flux $\boldsymbol{F}(u_\delta)$ that appears in Eq. (6). Because this value is computed at the boundary between adjacent elements, it may be computed from the value of u_δ given at either element. These two possible values are denoted here as u_δ^- (left) and u_δ^+ (right), and the boundary flux written $f(u_\delta^-, u_\delta^+)$. Upwinding considerations dictate how this flux is computed. In the more complicated case of a hyperbolic system of equations, an approximate Riemann solver would be used to compute a value of f, g (in two-dimensions) based on u_δ^- and u_δ^+.

To illustrate how the discontinuous Galerkin formulation works, we consider the one-dimensional version of Eq. (4a), which we put in weak form and integrate by parts (to simplify notation $u_\delta \to u$, etc.)

$$(\partial_t u, v) - (f(u), v_x) + vf(u)|_{x_L}^{x_R} = 0, \tag{7a}$$

where $x \in [x_L, x_R]$, which represents the left and right boundaries of a single element.

The treatment of the boundary terms is important as it justifies the *conservativity property* reported earlier. To wit, the last term in Eq. (7a) expands to

$$v_R^- f_R^- - v_L^+ f_L^-,$$

which implies an *upwind* treatment (see flux of second term), and the test function v is evaluated inside the interval $[x_L, x_R]$. Note that f_L^- is a function of (u_L^-, u_L^+) and similarly for f_R^-. Integrating Eq. (7a) by parts again we obtain

$$(\partial_t u, v) + (f_x(u), v) + v_R^- f_R^- - v_L^+ f_L^- - v_R^- f_R^- + v_L^+ f_L^+, \tag{7b}$$

which reduces to the form

$$(\partial_t u, v) + (f_x(u), v) + v_L^+(f_L^+ - f_L^-). \tag{7c}$$

This final equation is of the form that is used in [31], and it represents the so-called weak imposition of boundary conditions (through the jump term). In the case that we use test functions which are constant along each element (in an equidistant mesh with spacing Δx), we recover the upwind (Euler backwards) finite difference formulation for the linear advection equation; i.e.,

$$(\partial_t u)_j + V\frac{u_j - u_{j-1}}{\Delta x} = 0,$$

where V is the constant advection velocity.

4. SCALAR ADVECTION EQUATION

4.1. Implementation

Using the derivative and projection operators described above, it is straightforward to implement a numerical method for Eq. (4a) using formula (6), which can be rewritten as

$$\partial_t(u_\delta, v)_e + \int_{\partial T_e} v\,\tilde{f}(u_\delta^-, u_\delta^+)\cdot\hat{n}\,ds - (\boldsymbol{F}(u_\delta), \nabla v)_e = 0, \tag{8}$$

where $\tilde{f}$ denotes the surface flux appearing in (6).

Suppose u has an expansion

$$u = \sum_{l,m} \alpha_{(l+mL)}\phi_{lm},$$

and if we substitute basis elements ϕ_{lm} for v above and use the discrete inner product $(\cdot,\cdot)_e$ then we have the equation

$$\frac{\partial\alpha}{\partial t} \approx S[(D_r G)^t W E f_r(u) + (D_s G)^t W E f_s(u)] - SE\int_{\partial T} \boldsymbol{F}\cdot\hat{n}\phi_{lm}\,ds.$$

Now $(D_c G)^t W E f_c = G^t D_c^t W E f_c = (G^t W)W^{-1}D_c^t W E f_c$ for $c = r, s$, so the first term may be efficiently computed by use of transpose derivative operators D_r^t and D_s^t.

The only remaining requirement is a discretization of the boundary integrals $\int_{\partial T}\boldsymbol{F}\cdot\hat{n}\phi_{lm}ds$. If $\boldsymbol{F}$ is interpolated to lie on the Gauss points along edges, then this edge integral may be handled by Gauss–Legendre integration. There are two reasons for doing this instead of using the same quadrature (Gauss–Radau, or Gauss–Lobatto) as in the interior. Theoretically, higher accuracy is needed at the edge quadrature; this is a result of interior integrals being carried out over a volume h^2, and edge integrals over h. In the error analysis, edge errors are multiplied by a larger constant. (See [27] for an account of truncation errors.) Another reason is to avoid the corner points of the triangle in quadrature. For nonlinear problems computing the flux accurately there may require the solution of a multidimensional Riemann problem.

An alternative implementation of the discontinuous Galerkin method is possible if (8) is integrated by parts again; this formulation does not rely upon transposed derivative operators. The formulation then becomes

$$\partial_t(u_\delta, v)_e + \int_{\partial T_e} v[\tilde{f}(u_\delta^-, u_\delta^+) - \boldsymbol{F}(u_\delta)]\cdot\hat{n}\,ds + (v, \nabla\cdot\boldsymbol{F}(u_\delta))_e = 0. \tag{9}$$

This formulation is similar to the formulation in one dimension in Eq. (7c) and in [31]. It is the one that we have implemented in the current work.

4.2. Eigen-Spectrum of the Advection Operator

To understand the stability properties of the numerical discretization described above, we use the linear advection equation

$$\partial_t u + \nabla \cdot (u\mathbf{V}) = 0,$$

where **V** is a constant velocity vector.

Figure 6 provides a plot of the spectra of the advection operator on a single standard triangle, where the left and bottom edges are inflow boundaries and the diagonal edge is the outflow. This plot changes little as the flow direction is changed from $\theta = 0$ to $\theta = 90°$(see Fig. 7). This numerical method shares the property of one-dimensional Legendre spectral methods in that the maximum eigenvalue magnitude grows linearly as the polynomial order $(M - 1)$ increases. Similar to the one-dimensional case, the high degree of nonnormality in the matrix equations implies that in practice the practical time step in a numerical scheme is inversely proportional to M^2, not M as the von Neumann stability analysis predicts [33]. In fact, this linear growth of eigenvalue magnitude is destroyed as the problem is perturbed.

For example, consider the advection problem on the meshes in Fig. 10 with an upwind flux being used for $\tilde{f}$ in Eq. (8), and inflow boundaries on the bottom and left boundaries, outflow on the top and right. With an upwind flux, the computation on each individual triangle is very close to the case above with a single triangle case, and the eigenvalue spectrum looks the same. If $\tilde{f}$ is changed to be a centered flux, i.e. $\tilde{f}(u_\delta^-, u_\delta^+) = f(u_\delta^-)/2 + f(u_\delta^+)/2$, the eigenvalue magnitudes will suddenly grow as M^2. If the problem is changed to be periodic on all four sides, or even just periodic in one axis direction, the eigenvalue magnitudes will

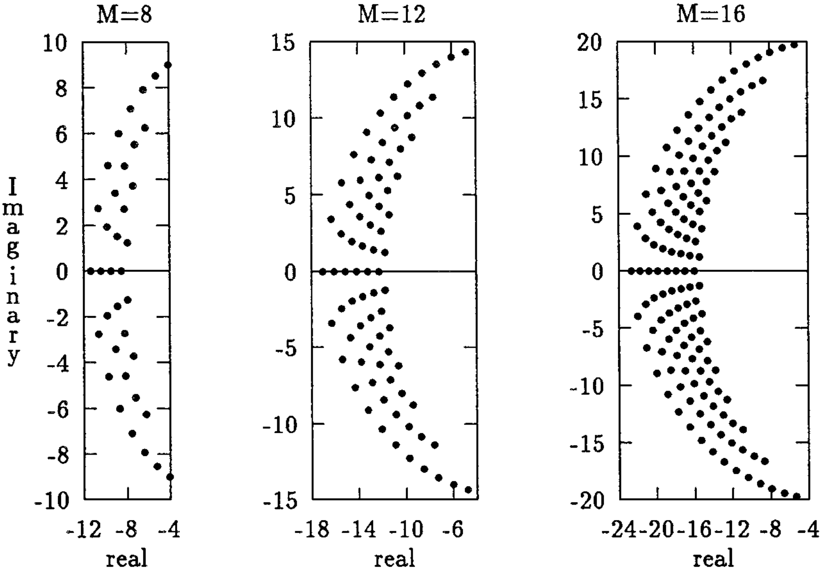


FIG. 6. Spectrum for the linear advection operator on a single triangle, for a wave speed of magnitude one travelling $\theta = 45°$ from the horizontal (i.e., $\mathbf{V} = (\cos\theta, \sin\theta)$).

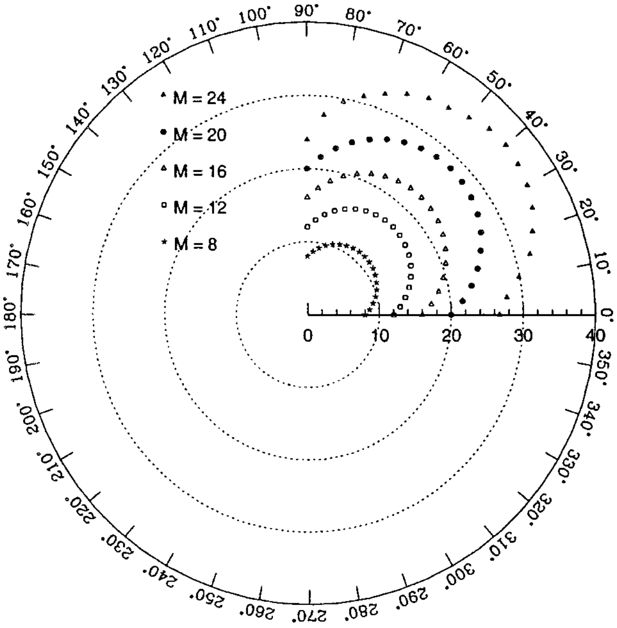


FIG. 7. Maximum eigenvalue magnitude for the linear advection operator on a single triangle for $0 \leq \theta \leq 90^\circ$.

grow as M^2. Figures 8 and 9 demonstrate both the dependence for this magnitude on θ and the polynomial order M for the 2-triangle mesh depicted in Fig. 10A.

4.3. Spectral Convergence

The discretization described above was used to implement a numerical method, using a third-order TVD Runge–Kutta solver to integrate in time. A periodic convection problem

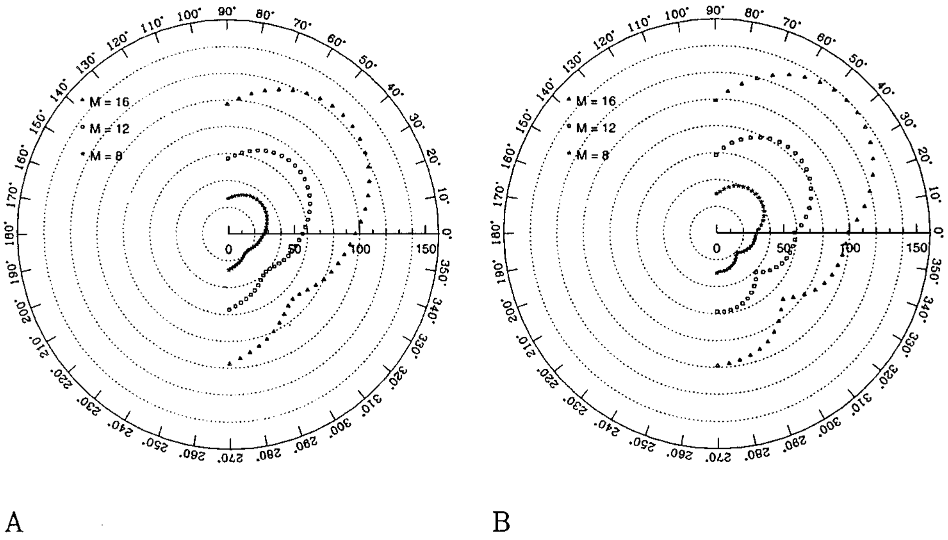


FIG. 8. Maximum eigenvalue magnitude for the linear advection operator on a two triangle periodic box. A, centered flux; B, upwind flux for $0 \leq \theta \leq 90^\circ$.

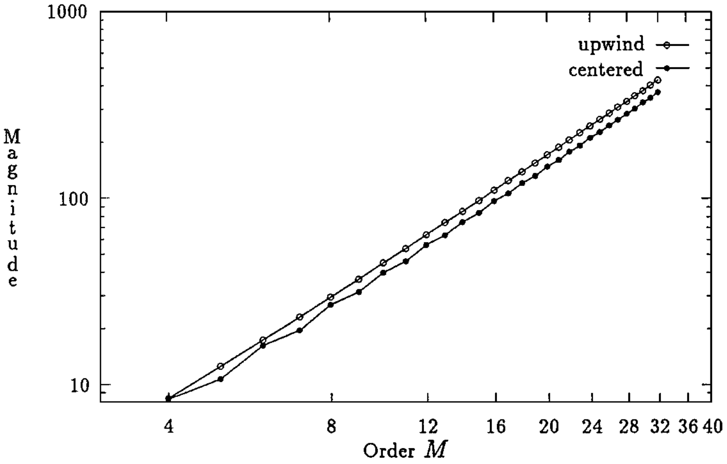


FIG. 9. Eigenvalue magnitude for the two triangle periodic box at $\theta = 0$. The exponent given by a linear regression fit is 1.88 for the centered flux and 1.91 for the upwind flux.

with $\theta = 0$ and initial condition

$$u(x) = \sin(\cos(\pi x))$$

was solved for the meshes in Fig. 10. The L_∞ error at $t = 2\pi$ is plotted in Fig. 11. The time step in all cases was $\Delta t = 1/500$, except for mesh C where $\Delta t = 1/1000$ was used. In all cases, the time step provides the ultimate limit on accuracy, which is governed by

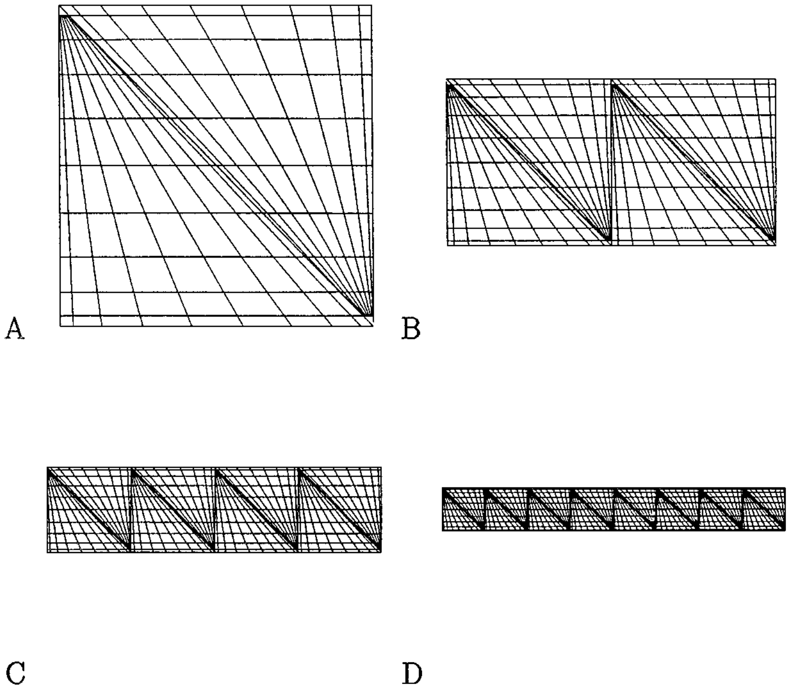


FIG. 10. Computational meshes for the convection problem. Dimensions are A, $-1 \le y \le 1$; B, $-\frac{1}{2} \le y \le \frac{1}{2}$; C, $-\frac{1}{4} \le y \le \frac{1}{4}$; D, $-\frac{1}{8} \le y \le \frac{1}{8}$. In all cases $-1 \le x \le 1$. Quadrature points are shown for ninth-order polynomials.

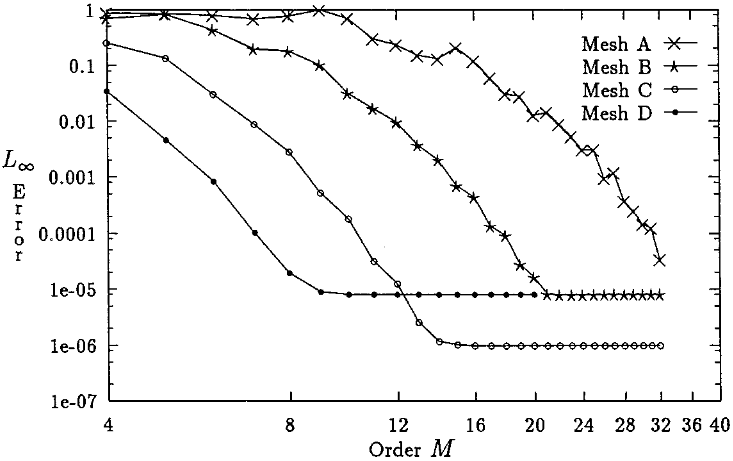


FIG. 11. L_∞ error for periodic convection on meshes A, B, C, and D.

exponential (spectral) convergence as shown in the plots. The effect of the finite time step is confirmed by case C, where reducing the time step by a factor of two increases limiting accuracy by a factor of 8. This is what one would expect based on the third-order time-stepping accuracy.

4.4. Limiting

To preserve monotonicity in the solution we can either apply flux limiters or use elements of zero order. We will describe the latter in the simulations so here we present the limiting procedure. The method used here is an extension of the one proposed by Cockburn *et al.* [27] to high-order accuracy. We give a specific example for third-order accuracy. The basic idea is to modify function values of u at the boundary of the element in such a way that the resulting numerical method obeys a maximum principle. The modification is carried out essentially by comparing the resulting flux through the element with that which would be obtained by a stable low-order method. No changes are made in smooth parts of the flow field, but in areas near shocks, the method performs comparably to a low-order method with a grid size given by the individual elements.

The limiting algorithm is accomplished in the following way. Values of u on a single element are interpolated on to a grid of 10 evenly spaced points (see Fig. 12). The

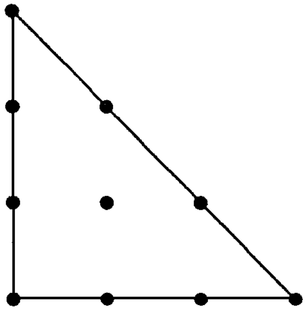


FIG. 12. Evenly spaced interpolation points on the standard triangle.

“cell average” of the mass

$$\bar{u} = \int_T u\, dx$$

is computed on the local and nearby elements. By comparing $\bar{u}$ from these different elements, the nine edge values in Fig. 12 are changed so as to ensure a maximum principle for the cell averages $\bar{u}$. Then the value at the center grid point is modified so that the Lagrange interpolant function has the same total mass as $\bar{u}$ on the local element.

In order to understand the performance of the limiter and the accuracy of the method for nonlinear scalar problems, a test problem from Cockburn *et al.* [27] was used. This consists of the inviscid Burgers’ equation calculated for a sinusoidal initial condition on a square:

$$\begin{aligned} &\partial_t u + (\partial_x + \partial_y)\frac{1}{2}u^2 = 0 && \text{for } (t, x, y) \in (0, T) \times \Omega, \\ &u(t = 0, x, y) = \frac{1}{4} + \frac{1}{2}\sin(\pi(x + y)) && \text{for } (x, y) \in \Omega, \end{aligned} \tag{10}$$

where the computational domain is defined as $\Omega = [-1, 1] \times [-1, 1]$ and periodic boundary conditions are applied.

In order to calculate the nonlinear flux accurately while avoiding aliasing errors, enough collocation points were used to compute the u^2 flux exactly. For example, in the case of the third-order scheme, where u is approximated with second-order polynomials, u^2 was computed by

1. projecting from the nine quadrature points to the second-order polynomial basis (with six modes);
2. evaluating on the 25 quadrature points used for the fourth-order polynomial basis;
3. computing u^2 on the 25 quadrature points.

Derivatives would then be computed on the 25 quadrature points, and the result then projected back to second-order polynomials.

The test problem was computed to time $t = 0.1$ without limiting and to time $t = 0.45$ with limiting applied. At time $t = 0.1$, the solution is smooth, and the L^∞ errors demonstrate uniform achievement of the theoretical order of accuracy, as can be seen in Table 1. At time $t = 0.45$ high accuracy is retained away from the discontinuity, as can be seen in Table 2.

TABLE 1

L^∞ Error for Initial Value Problem (10) at Time $t = 0.1$

h	3rd order L^∞	4th order L^∞	Δt
1/4	0.0134949	0.00278605	.01
1/8	0.00294017	0.000252722	.01
1/16	0.000450087	1.69009e-05	.005
1/32	6.20611e-05	1.1896e-06	.0025
	Order: 2.86	Order: 3.83	

Note. Time integration is an insignificant source of error with the given Δt. The observed order of accuracy (2.86/3.83) is computed based on the last two data points.

TABLE 2
L^∞ Error for Initial Value Problem (10) at Time $t = 0.45$

h	4th order L^∞	Δt
1/4	4.81807e-05	.01
1/8	8.17886e-06	.01
1/16	7.93379e-07	.005
1/32	6.39865e-08	.0025
	Order: 3.63	

Note. Errors are computed in the region $[-0.2, 0.4] \times [-0.2, 0.4]$.

5. EULER EQUATIONS

The preceding methods can be also applied to systems of equations, where u in Eq. (4a) is a vector of conserved quantities. The Euler equations may be written

$$\partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) + \partial_y \mathbf{G}(\mathbf{U}) = 0,$$

where

$$\mathbf{U} = [\rho, \rho u, \rho v, E]^t$$

and (u, v) is the local fluid velocity, ρ the fluid density, and E is the total internal energy. For an ideal gas the pressure p is related to E by

$$p = (\gamma - 1)[E - \rho(u^2 + v^2)/2].$$

The flux $\boldsymbol{F} = (\mathbf{F}, \mathbf{G})$ may be written

$$\begin{aligned} \mathbf{F} &= [\rho u, \rho u^2 + p, \rho u v, u(E + p)]^t \\ \mathbf{G} &= [\rho v, \rho u v, \rho v^2 + p, v(E + p)]^t. \end{aligned}$$

The question that needs to be addressed in order to solve this system using the formulation (8) is how to compute the upwind flux $\tilde{f}(u_\delta^-, u_\delta^+)$. An (approximate) Riemann solver needs to be applied to arrive at a physical flux across the discontinuities between elements.

The simplest approach is to approximately diagonalize the system. For a one-dimensional constant-coefficient hyperbolic system

$$\partial_t \mathbf{U} + A \partial_x \mathbf{U} = 0,$$

where A can be diagonalized as $A = \mathcal{R} \Lambda \mathcal{L}$,

$$\partial_t \mathcal{L}\mathbf{U} + \Lambda \partial_x \mathcal{L}\mathbf{U} = 0.$$

Here $\mathcal{R}$ and $\mathcal{L}$ are matrices containing the right- and left-eigenvectors of A, and Λ contains the corresponding eigenvalues in its diagonal. So transforming to variables $(L\mathbf{U})$ decouples

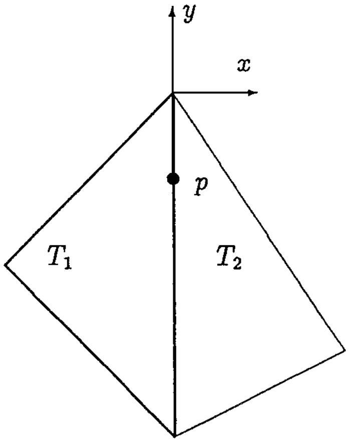


FIG. 13. Construction of approximate interface flux.

the system of equations creating independent scalar equations that may be treated individually. The same approach may be used to approximately decouple multidimensional nonlinear systems. For example, suppose at a point p along the common edge of two triangles T_1 and T_2 (see Fig. 13) $\mathbf{U}$ takes on values $\mathbf{U}_1$ and $\mathbf{U}_2$. For simplicity assume the common edge is parallel to the y axis. Then if we assume that the flow is independent of y locally to p, we may ignore the y derivative:

$$\partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) \approx 0.$$

This equation may then be approximately be decoupled by taking an average value of $\bar{U}$ intermediate between $\mathbf{U}_1$ and $\mathbf{U}_2$, diagonalizing A, where

$$A = \mathcal{R}\Lambda\mathcal{L} = \partial_{\bar{U}} \mathbf{F}(\bar{U}).$$

Then upwinding may be performed based on the sign of the eigenvalues Λ. For example, the Roe flux splitting [34] then would be

$$\tilde{f}(\mathbf{U}_1, \mathbf{U}_2) = \frac{\mathbf{F}(\mathbf{U}_1) + \mathbf{F}(\mathbf{U}_2)}{2} + \mathcal{R}|\Lambda|\mathcal{L}\frac{\mathbf{U}_1 - \mathbf{U}_2}{2}.$$

For a system of equations, *limiting* may be performed for each component separately. Better quality solutions are obtained if each element is locally projected to characteristic variables. The procedure used here is as follows:

1. For each triangle, take the cell averages of the state vector $\bar{U}$.
2. Compute using these values the matrices $\mathcal{R}$ and $\mathcal{L}$.
3. Project the edge values in figure 12 applying $\mathcal{L}$ to the edge points of adjacent triangles. Project also the cell averages.
4. Limit each characteristic variable independently using the scalar limiting algorithm.
5. Apply $\mathcal{R}$ to return to conserved variables.

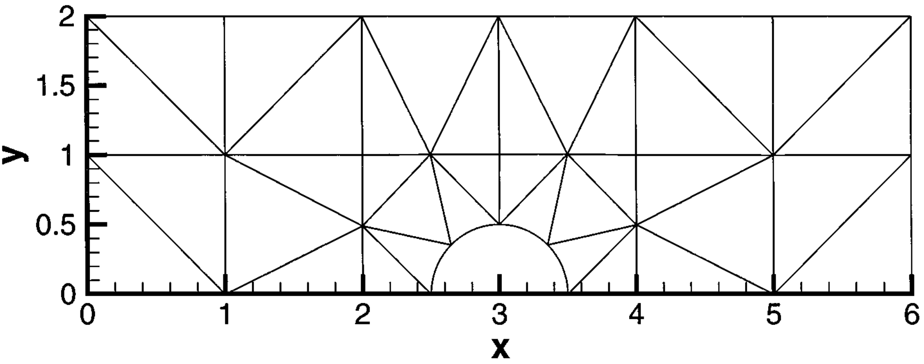


FIG. 14. Computational domain for internal flow over a semi-circular bump consisting of 26 triangular elements. The radius of the bump is 0.5.

5.1. Convergence

A benchmark problem that tests the accuracy of the method described above for the Euler equations is inviscid flow over a semi-circular bump. For adiabatic and irrotational flow conditions the entropy should remain zero everywhere in the domain. In practice, this is difficult to achieve with low-order methods because of the numerical boundary layer creation near the walls due to discretization error. The convergence of the numerically obtained entropy approaching zero should then determine the order of accuracy of the method. Figure 14 shows the computational domain used in the current test, which consists of a parallel channel with a semi-circular bump on the lower wall. The flow parameters at inflow are chosen to approximate a Mach 0.3 air flow at STP. The top and bottom walls are specified as reflecting boundary conditions. These were imposed by computing an adjacent element flux computed from a flow condition with zero normal velocity and pressure identical to that of the element on the inner surface of the wall. Inflow and outflow boundary conditions were imposed by specifying an adjacent flux computed using the reference flow conditions.

Contours of the Mach number are plotted in Fig. 15. The relative error in maximum entropy is plotted in Fig. 16 and it demonstrates that the numerical error approaches zero exponentially fast. In Fig. 17 the convergence history at a wall point with coordinates $(x = 4.0, y = 0.0)$ is plotted; it shows that the method is stable after a very long integration time. We note here that *no limiting* or other filtering is applied and that the expansion order is the same in each element. Finally, we compare a simulation with expansion order fixed

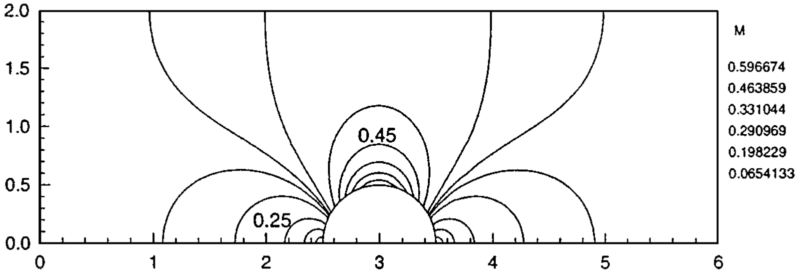


FIG. 15. Mach contours for inviscid flow over a semi-circular bump. The inlet Mach number is $Ma = 0.3$.

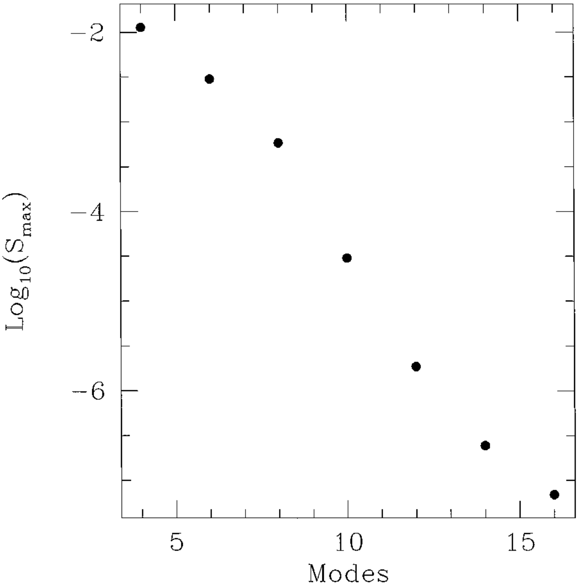


FIG. 16. Maximum entropy (logarithm) as a function of the expansion order per element. The number of elements is fixed.

($M = 8$) for all elements with a simulation where the elements away from the bump have order $M = 6$ and the elements around the bump have order $M = 8$. If variable expansion order is employed care should be used in the construction of the interface fluxes. In particular, at the interface between elements the higher of the two values should be used as the low-order

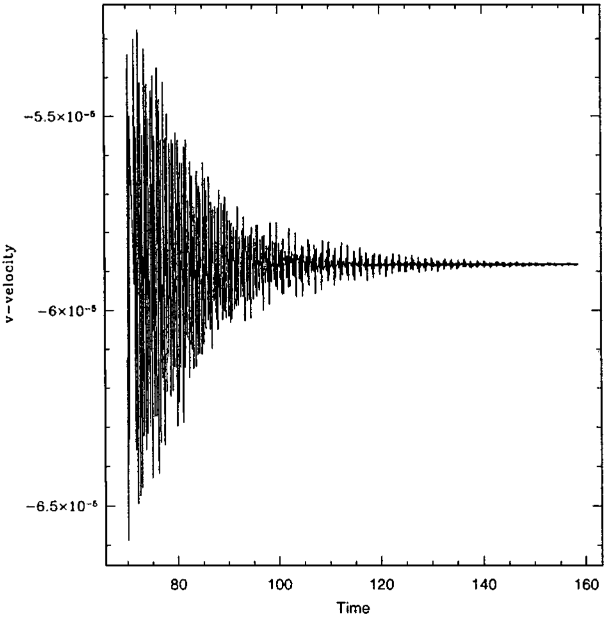


FIG. 17. Convergence history at a wall point ($x = 4.0$, $y = 0.0$) demonstrating time-asymptotic stability ($M = 9$).

produces oscillations rendering the numerical solution unstable. The oscillations originate at exactly the interface if it is not treated properly. The distribution of the error in entropy for the variable order and the fixed order simulations is shown in Fig. 18. Approximately the same levels of error are computed in both cases with the large errors located around the bump. Also, in the variable-order simulation larger entropy errors are seen in the rest of the domain due to the lower element resolution in that region.

6. NAVIER–STOKES EQUATIONS

In this section we consider the nondimensionalized Navier–Stokes equations, which in two dimensions can be written in conservation form as

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ E\end{pmatrix}+\frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ (E+p)u\end{pmatrix}+\frac{\partial}{\partial y}\begin{pmatrix}\rho v\\ \rho vu\\ \rho v^2+p\\ (E+p)v\end{pmatrix}$$

$$=\frac{1}{\mathrm{Re}_\infty}\left\{\frac{\partial}{\partial x}\begin{pmatrix}0\\ \frac{2}{3}\mu\left(2\frac{\partial u}{\partial x}-\frac{\partial v}{\partial y}\right)\\ \mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\\ \frac{2}{3}\mu\left(2\frac{\partial u}{\partial x}-\frac{\partial v}{\partial y}\right)u+\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)v+\kappa\frac{\gamma}{Pr_\infty}\frac{\partial T}{\partial x}\end{pmatrix}\right.$$

$$\left.+\frac{\partial}{\partial y}\begin{pmatrix}0\\ \mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\\ \frac{2}{3}\mu\left(2\frac{\partial v}{\partial y}-\frac{\partial u}{\partial x}\right)\\ \frac{2}{3}\mu\left(2\frac{\partial v}{\partial y}-\frac{\partial u}{\partial x}\right)v+\mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)u+\kappa\frac{\gamma}{Pr_\infty}\frac{\partial T}{\partial y}\end{pmatrix}\right\}. \tag{11}$$

Here μ is the dynamic viscosity and κ is the thermal conductivity. We nondimensionalized the equations by introducing the following reference quantities (the subscript "∞" denotes reference values): U_∞, ρ_∞, μ_∞, κ_∞, and a reference length L.

The reference Prandtl number and Reynolds number are then defined as

$$\mathrm{Pr}_\infty=\frac{\mu_\infty c_p}{\kappa_\infty},\quad \mathrm{Re}_\infty=\frac{\rho_\infty U_\infty L}{\mu_\infty}. \tag{12}$$

The left-hand side coincides with the Euler equations of the previous section, and the right-hand side includes the effects of dissipation through the viscosity and thermal conductivity. We can rewrite in compact form the compressible Navier–Stokes equations as

$$\mathbf{U}_t+\nabla\cdot\mathbf{F}=Re_\infty^{-1}\nabla\cdot\mathbf{F}^\nu, \tag{13}$$

where **F** and $\mathbf{F}^\nu$ correspond to inviscid and viscous contributions, respectively. Splitting the Navier–Stokes operator in this form allows for a separate treatment of the inviscid and viscous contributions, which in general exhibit different mathematical properties.

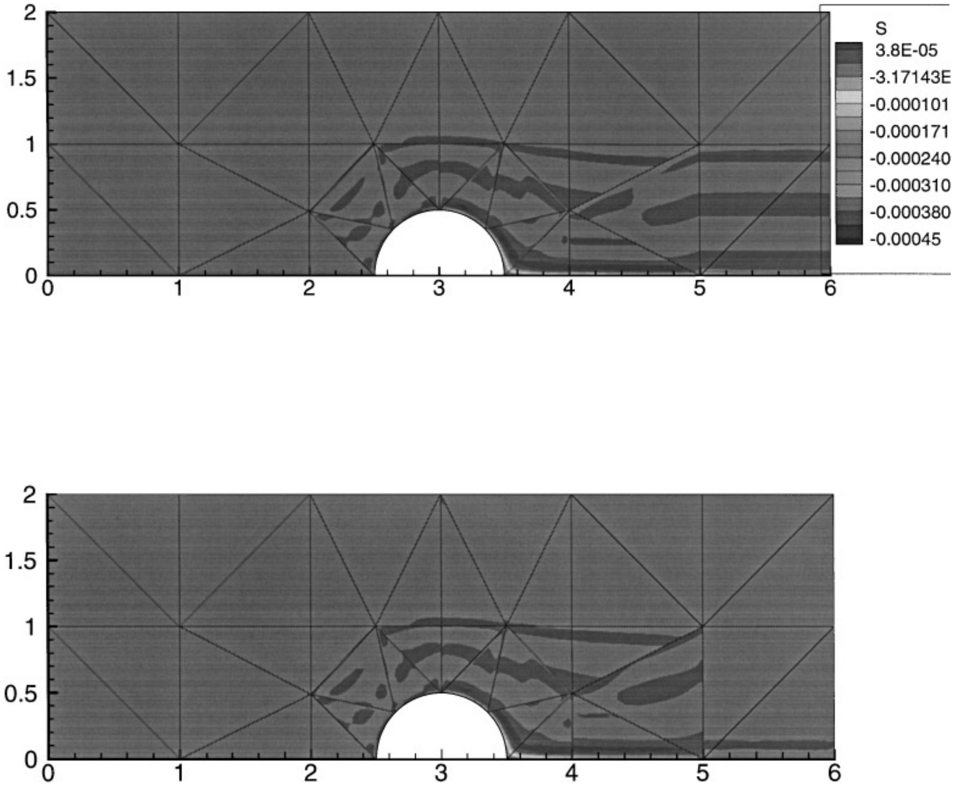


FIG. 18. Contours of entropy generated due to discretization error for a fixed ($M = 8$) (top) and a variable ($M = 6$–8) (botom) expansion order. The extra errors shown in the lower plot due to the variable-order discretization are bounded by the largest errors of the upper plot.

6.1. Mixed Spectral/hp Formulation

We will assume here that the Euler term $\nabla \cdot \mathbf{F}$ has been discretized first using a discontinuous Galerkin method, which implies that the solution is formally discontinuous, i.e.

$$\mathbf{U} \in L^2(\Omega).$$

With this in mind, we can subsequently discretize the viscous term $\nabla \cdot \mathbf{F}^{\nu}$ using a mixed Galerkin method involving two sets of test functions, one set in $L^2(\Omega)$ and the other one in $C^0(\Omega)$. We thus consider the parabolic model problem for the scalar field $u(\mathbf{x}, t)$:

$$u_t = \nabla \cdot (\nu \nabla u) + f, \quad \text{in } \Omega \tag{14a}$$

$$u = g(\mathbf{x}, t), \quad \text{in } \partial\Omega. \tag{14b}$$

To proceed we introduce the flux variable

$$\mathbf{q} = -\nu \nabla u$$

with $\mathbf{q}(\mathbf{x}, t) \in H(\text{div}; \Omega)$, where we define the new functional space as

$$H(\text{div}; \Omega) = \{\mathbf{v} \in L^2(\Omega); \nabla \cdot \mathbf{v} \in L^2(\Omega)\},$$

which is a Hilbert space equipped with the appropriate norm, i.e.

$$\|\mathbf{v}\|_{H(\mathrm{div};\Omega)} = \{\|\mathbf{v}\|^2 + \|\nabla\cdot\mathbf{v}\|^2\}^{1/2}.$$

We can discretize in time first using (for simplicity) a single-step integrator and, also, substitute in terms of the flux variable to obtain

$$\frac{u^{n+1}-u^n}{\Delta t} = -\nabla\cdot\mathbf{q}^m + f^m, \quad \text{in } \Omega \tag{15a}$$

$$\mathbf{q}^m = -\nu\nabla u^m \qquad \text{in } \Omega \tag{15b}$$

$$u^{n+1} = g(\mathbf{x}, t^m), \qquad \text{on } \partial\Omega, \tag{15c}$$

where $m = n$, $(n+1)$ correspond to explicit (implicit) time integration, respectively. The variational form can now be derived by testing Eq. (15a) with functions $w \in L^2(\Omega)$. Correspondingly, we test Eq. (15b) against functions $\mathbf{v} \in H(\mathrm{div};\,\Omega)$, and subsequently integrate by parts. The Dirichlet variational problem corresponding to Eqs. (15a), (15b) is then stated as:

Find $(\mathbf{q}, u) \in H(\mathrm{div};\,\Omega) \times L^2(\Omega)$ such that

$$(u^{n+1}, w) = (u^n, w) + \Delta t[-(\nabla\cdot\mathbf{q}^m, w) + (f^m, w)] \quad \forall w \in L^2(\Omega)$$

$$(\mathbf{q}^m, \mathbf{v}) = (\nabla\cdot\mathbf{v}, u^m) - (g^m, \mathbf{v}\cdot\mathbf{n})_{\partial\Omega} \qquad \forall\mathbf{v} \in H(\mathrm{div};\,\Omega),$$

where $\mathbf{n}$ is the outward unit normal and parentheses denote standard inner products. For the implicit scheme the coupled system for $\mathbf{q}$, u has to be solved, whereas for the explicit scheme, the unknown u^{n+1} is obtained from the first equation by a simple projection. If no projection is performed but, instead, the equation is solved in its strong form, then numerical instabilities develop. This has been verified by numerical experiments.

Next we need to define appropriate polynomial spaces for $\mathbf{v}$, w and corresponding unknowns $\mathbf{q}, u$ to guarantee stability of the approximation. This problem is similar to the Stokes problem which is treated using a mixed formulation in [35]. Following similar arguments we choose the polynomial space for $\mathbf{q}$, $\mathbf{v}$ to be $\mathcal{P}_N(\Omega^e)$ and the polynomial space for u, w to be $\mathcal{P}_{N-1}(\Omega^e)$. For the numerical quadrature the same Gauss–Lobatto–Jacobi (Gauss–Radau–Jacobi) points are used for the flux variable as well as for the velocity. An example of the stability of the approximation with this choice of polynomial spaces is shown in Fig. 19, where we integrate explicitly the one-dimensional version of (14a) for a long time. The initial condition corresponds to a step function in the interval [0, 1]. The proper choice of polynomial space produces the correct solution, whereas polynomial representation of equal order produces unphysical oscillations, which do not decay in time, as they should.

The mixed formulation results in exponential convergence of the error. To verify this we obtained the numerical solution for $u_t = \nabla^2 u$ in the two-dimensional domain of the semicircular bump (see Fig. 14). The exact solution $u = \sin x \sin y e^{-2t}$ was used and Dirichlet boundary conditions were applied on all boundaries. The final integration time was $t = 1.0$ and the time step $\Delta t = 10^{-5}$ was chosen small to avoid any temporal discretization errors. Third-order explicit time integration was employed. The plot of the L_∞ error is shown in

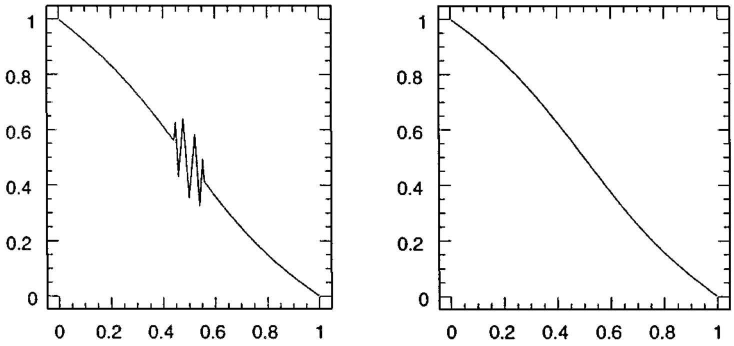


FIG. 19. Integration of the parabolic equation with discontinuous initial data for 50,000 time steps with $\Delta t = 10^{-6}$. A stable approximation is obtained with polynomial orders 7 and 6 for the flux variable and the solution, respectively (right), while oscillations prevail with equal polynomial order 7 (left).

Fig. 20 as a function of the expansion order for a fixed number of elements. Exponential convergence is demonstrated for this curvilinear geometry.

6.2. Convergence and Simulations

We present here numerical solutions of the Navier–Stokes equations using first an analytic solution in order to demonstrate the exponential convergence of the spectral/hp algorithm and, subsequently, simulations of external flows in order to demonstrate its flexibility in h-p refinement and unrefinement procedures.

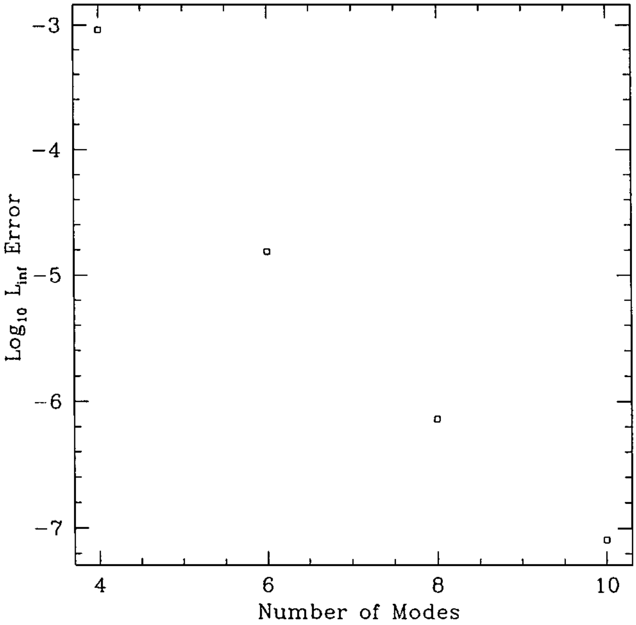


FIG. 20. Convergence of the mixed spectral/hp method for an exact solution on the semi-circular-bump domain. The parabolic equation was integrated up to time $t = 1$ with time step $\Delta = 10^{-5}$.

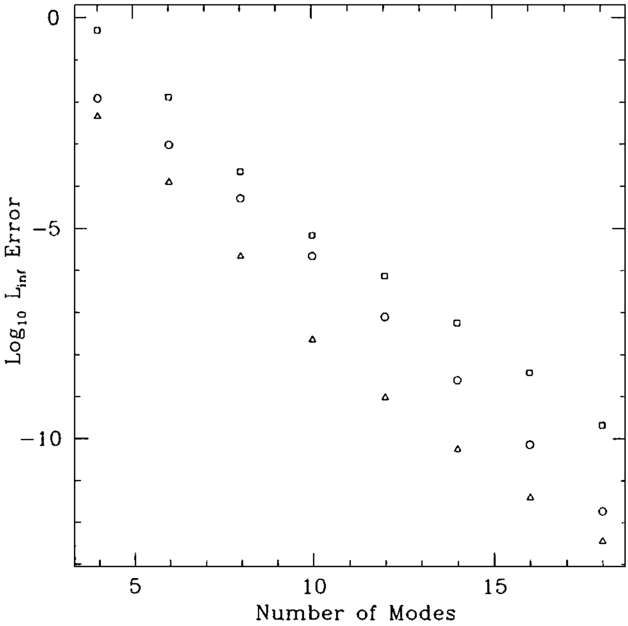


FIG. 21. L_∞ error as a function of the expansion order for an analytic solution of the steady Navier–Stokes equations obtained in a rectangular domain consisted of eight triangular elements: □ corresponds to total energy E; ○ corresponds to momentum flux ρu; and △ corresponds to density.

In the first example we use a square domain consisted of eight triangles. On the left and right sides periodic boundary conditions are assumed and on the top and bottom Dirichlet boundary conditions are prescribed. The analytic solution has the form

$$\begin{aligned}\rho &= A + B\sin(\omega x)\\ u &= C + D\cos(\omega x)\sin(\omega y)\\ T &= E + Fy,\end{aligned}$$

where $\omega = 8\pi$, $A = 1$, $B = 0.1$, $C = 1$, $D = 0.04$, $E = 84$, and $F = 28$. The Navier–Stokes equations are then integrated using a forcing term consistent with the above solution. In Fig. 21 we plot the L_∞ error (for a fixed number of elements) versus the expansion order for the conserved quantities ρ, ρu, and E. We see that exponential accuracy is achieved with the error in the total energy higher than the errors in the density and momentum flux. Next we examine the temporal accuracy of the method using an Adams–Bashforth (explicit) integrator assuming an analytic solution as before but varying in time. More specifically, the *sine* term in the density and the linear term in the temperature are multiplied by $\sin(10\pi t)$, and the second term in the velocity is multiplied by $\cos(10\pi t)$. The final integration time is $t = 0.2$. Numerical solutions were obtained for different sizes of time step, and the results are summarized in Fig. 22 for first-, second-, and third-order Adams–Bashforth integration. Correspondingly, first-, second-, and third-order accuracy is achieved.

Next we present simulations of flow past a cylinder at $\mathrm{Re} = 100$ for a subsonic $\mathrm{Ma} = 0.2$ case and a supersonic $\mathrm{Ma} = 2$ case. The objective is twofold: First, to validate the simulation results at the subsonic cases with experimental results and other simulations we have

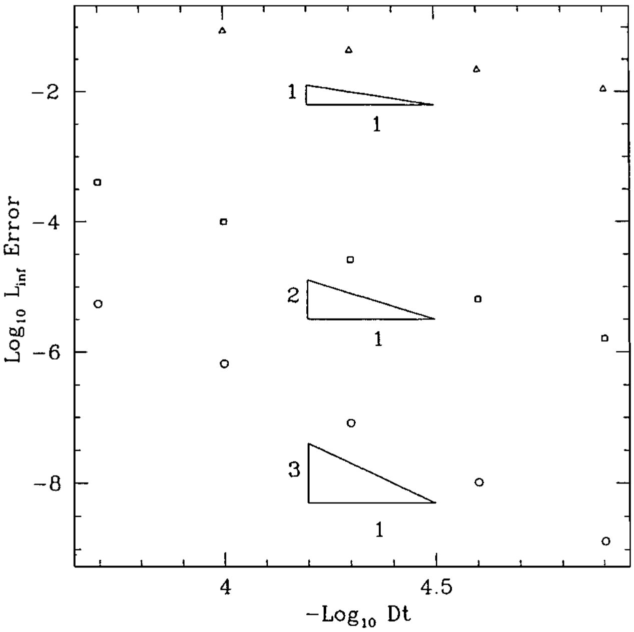


FIG. 22. L_∞ error as a function of the time step for an analytic solution of the unsteady Navier–Stokes equations obtained in a rectangular domain consisted of eight triangular elements.

performed with different methods; second, to demonstrate that the presence of shock waves presents no problem for the new method.

A more systematic study of the flow past a cylinder studying the effect of compressibility on suppressing the vortex street is presented elsewhere [36]. The simulations were performed on the domain shown in Fig. 23 along with the triangulization; 462 elements were used and two sets of simulations were performed one at order $M = 5$ and one at order $M = 7$.

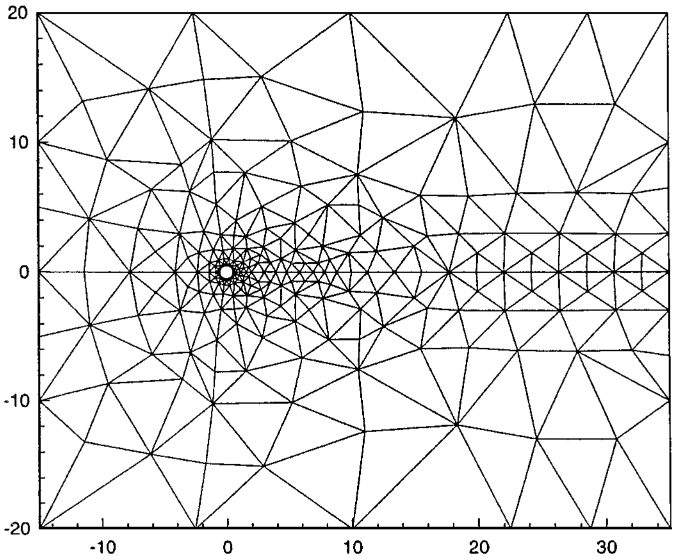


FIG. 23. Computational domain for subsonic flow past a cylinder; 462 elements are used in the discretization.

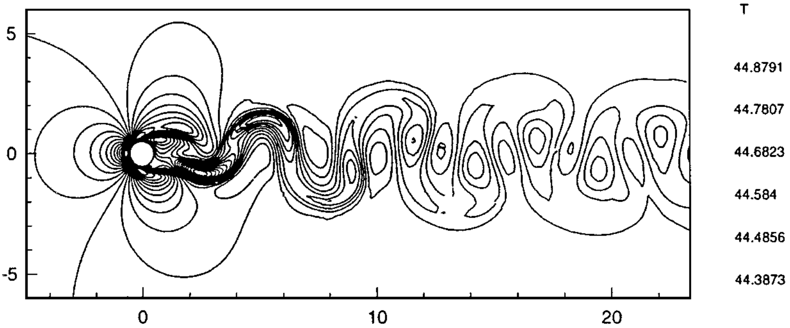


FIG. 24. Instantaneous temperature contours of flow past a cylinder at Re = 100 and Ma = 0.2.

In Fig. 24 we plot contours of temperature of the instantaneous field that shows the von Karman vortex street observed in low Mach number flows. The lateral and streamwise spacing of vortices agrees with simulation results using the $\mathcal{N}\varepsilon\kappa\mathcal{T}\alpha r$ code documented in [21]. The Strouhal number (nondimensional frequency) is St = 0.165, in excellent agreement with the experimental results reported in [37]. In Figs. 25 and 26 we plot the lift and drag coefficients as a function of time, indicating the separate contributions due to pressure and viscous forces. The peak-to-peak amplitude of the lift coefficient is 0.67, in good agreement with incompressible simulations (0.68), and the average drag coefficient is 1.37, in good agreement with the experimental value 1.35 and with simulations of the corresponding incompressible flow reported in [38]. The base pressure coefficient is −0.73, again in good agreement with the experimental value −0.73 [39] and with the incompressible simulations −0.72.

Finally, in Fig. 27 we plot the centerline average velocity versus the streamwise distance for both the low and high resolution as well as for another simulation based on a spectral

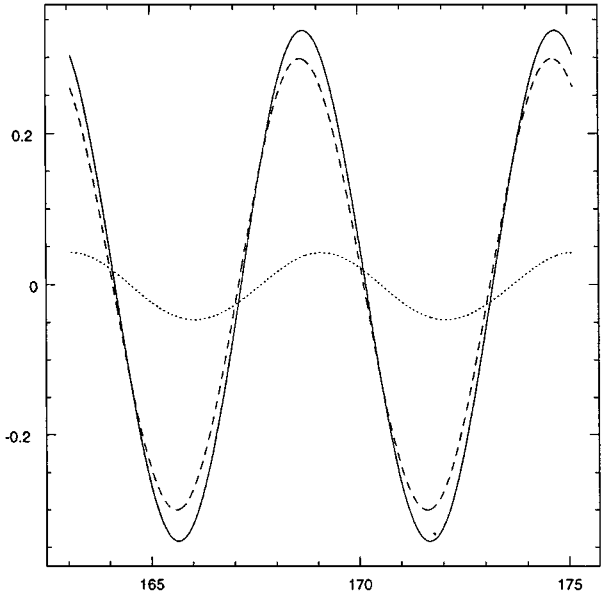


FIG. 25. Lift coefficient (solid line) as a function of time for flow past a cylinder at Re = 100 and Ma = 0.2. Shown with dash line is the pressure contribution and with dot line is the viscous contribution.

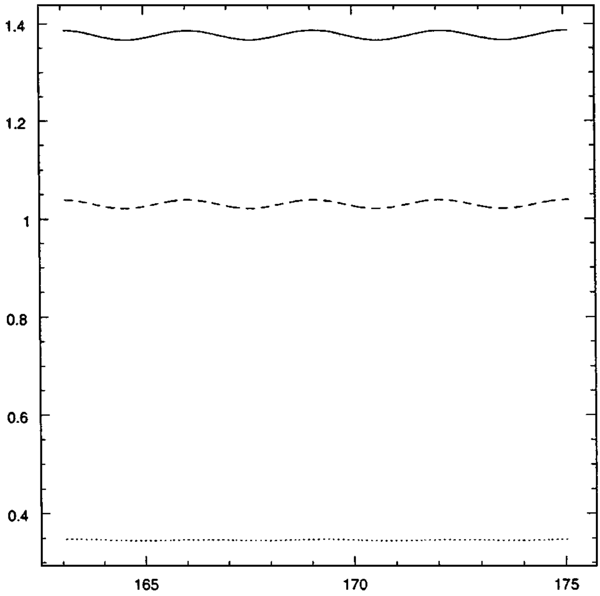


FIG. 26. Drag coefficient (solid line) as a function of time for flow past a cylinder at $\mathrm{Re} = 100$ and $\mathrm{Ma} = 0.2$. Shown with dash line is the pressure contribution and with dot line is the viscous contribution.

element (collocation) formulation for subsonic flows [40]. These simulations were tested for possible differences if an upwind, instead of a Roe approximate flux is used, and also if the arithmetic mean state, instead of the average Roe state, is used in the calculations. Similar results were obtained for $\mathrm{Ma} = 0.7$. As expected at these low Mach numbers no differences were found.

Unlike the subsonic flow, the supersonic flow is steady at $\mathrm{Re} = 100$. In particular, we have performed a time-dependent simulation at Mach number $\mathrm{Ma} = 2$ on the unstructured mesh

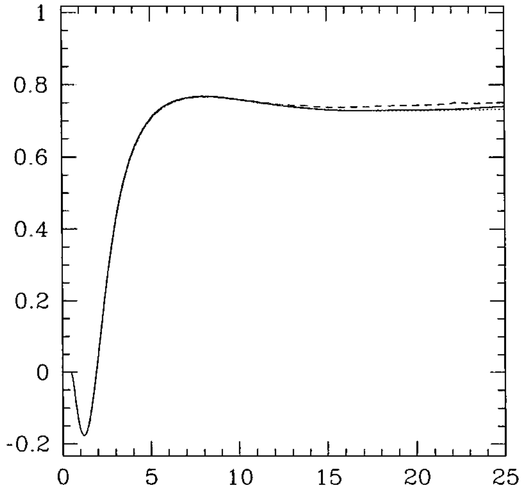


FIG. 27. Average centerline velocity in the wake of flow past a cylinder at $\mathrm{Re} = 100$ and $\mathrm{Ma} = 0.2$. Shown with solid line is the high resolution simulation ($M = 7$), with dash line the low resolution simulation ($M = 5$), and with dot line a simulation using a spectral element collocation scheme due to Beskok and Karniadakis [40].

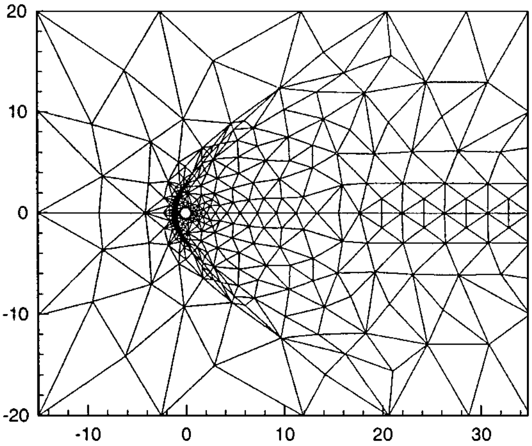


FIG. 28. Computational domain for supersonic flow past a cylinder; 1132 elements are used in the discretization.

shown in Fig. 28. The discretization is chosen on purpose to be irregular to demonstrate the robustness of the proposed method in handling non-Delaunay triangulizations. In Fig. 29 we plot density contours from a simulation with variable resolution, i.e. $M = 1$ in front of the shock and $M = 3$ behind the shock. The quality of the solution can be improved by p-refinement as shown in Fig. 30, where we use $M = 6$ in the wake.

7. SUMMARY

A discontinuous-Galerkin spectral/hp element method on triangles has been successfully applied to the computation of inviscid compressible flow. It was combined with a mixed Galerkin formulation for computing viscous compressible flows. Exponential convergence is space and third-order accuracy in time were demonstrated for analytic solutions of the advection, Euler, and Navier–Stokes equations. A limiting procedure may be used to stabilize the method in the presence of shocks. The applied limiting procedure in this case basically causes the method to degenerate to first-order accuracy within the shocked element. Adaptive h-refinement at that location then is necessary.

The method is characterized by algorithms comparable in efficiency to those used on quadrilateral elements because of the tensor-product property of the triangular spectral basis employed. The main computational expense is due to the inversion of the global mass matrix in the computation of the viscous term. In ongoing work we have addressed this issue and we have been able to also formulate a discontinuous Galerkin method for the second-order elliptic equation. This suggests that only local mass matrix inversions are necessary and that the discontinuous basic (instead of the C^0 basis) can be employed. Because of its orthogonality the local mass matrices are diagonal and their inversion is trivial.

The method has been implemented in parallel—most of the preceding test cases were calculated on up to eight nodes on an IBM SP2 using MPI. An embarrassingly parallel domain-decomposition algorithm is used for this fully explicit method, so very high parallel efficiencies may be expected for sufficiently large problems. Because high-order elements may be used, these parallel efficiencies may be achieved without necessarily using a large number of elements per node or large data-sets.

Additional geometric flexibility here has been gained by the use of triangular elements. It is fairly clear that there is some penalty in terms of performance for doing this. Theoretical operations counts will probably be higher by a factor of 2 over standard quadrilaterals in an equivalent computational domain. One should note, however, that there is no reason why triangular and quadrilateral elements may not be combined. Large uniform volumes in a flow field would then tend to be blocked out with quadrilaterals, and regions near the boundaries, regions undergoing adaptive refinement, or regions requiring gradients in refinement might then benefit from the geometric flexibility of triangles. Such work on spectral/hp methods on polymorphic domains is currently underway [41].

ACKNOWLEDGMENTS

We thank Professor C.-W. Shu, Dr. S. J. Sherwin, Dr. A. Beskok, and Mr. T. C. Warburton for many useful suggestions regarding this work. This work was supported by AFOSR.

REFERENCES

1. T. J. Barth, Recent developmets in high order k-exact reconstruction on unstructured meshes, AIAA-93-0668, 1993.
2. R. A. Shapiro and E. M. Murman, Higher-order and 3-d finite element methods for the Euler equations, AIAA-89-0655.
3. K. Morgan, O. Hassan, and J. Peraire, An unstructured grid algorithm for the solution of the Maxwell's equations in the time domain, *Int. J. Numer. Methods Fluids* **19**, 849 (1994).
4. H. O. Kreiss and J. Olinger, *Methods for the Approximate Solution of Time-Dependent Problems*, GARP Publ. Ser., Vol. 10 (GARP, Geneva, 1973).
5. D. A. Kopriva, Multidomain spectral solution of the Euler gas-dynamics equations, *J. Comput. Phys.* **96**(2), 428 (1991).
6. D. A. Kopriva, Spectral solutions of inviscid supersonic flows over wedges and axisymmetric cones, *Comput. & Fluids* **21**(2), 247 (1992).
7. W. Cai, D. Gottlieb, and C. W. Shu, Non-oscillatory spectral Fourier methods for shock wave calculations, *Math. Comput.* **52**, 389 (1989).
8. D. Sidilkover and G. E. Karniadakis, Non-oscillatory spectral element Chebyshev method for shock wave calculations, *J. Comput. Phys.* **107**, 1 (1993).
9. J. Giannakouros and G. E. Karniadakis, A spectral element-FCT method for the compressible Euler equations, *J. Comput. Phys.* **115**, 65 (1994).
10. D. A. Kopriva and J. H. Kolias, A conservative staggered-grid Chebyshev multidomain method for compressible flows, *J. Comput. Phys.* **125**, 244 (1996).
11. J. S. Hesthaven, A stable penalty method for the compressible Navier-Stokes equations. II. One dimensional domain decomposition schemes, *SIAM J. Sci. Comput.* **18**(2) (1997).
12. J. S. Hesthaven. A stable penalty method for the compressible Navier–Stokes equations. III. Multi-dimensional domain decomposition schemes, *SIAM J. Sci. Comput.* to appear.
13. J. S. Hesthaven and D. Gottlieb, A stable penalty method for the compressible Navier–Stokes equations. I. Open boundary conditions, *SIAM J. Sci. Comput.* **17**(3), 579 (1996).
14. G. E. Karniadakis and S. A. Orszag, Nodes, modes and flow codes, *Phys. Today* **46**, 34 (1993).
15. S. K. Lele, Compact finite difference schemes with spectral like resolution, *J. Comput. Phys.* **103**, 16 (1992).
16. A. T. Patera, A spectral method for fluid dynamics: Laminar flow in a channel expansion, *J. Comput. Phys.* **54**, 468 (1984).
17. G. E. Karniadakis, E. T. Bullister, and A. T. Patera, A spectral element method for solution of two- and three-dimensional time dependent Navier–Stokes equations, in *Finite Element Methods for Nonlinear Problems* (Springer-Verlag, New York/Berlin, 1985), p. 803.

18. N. P. Weatherill and O. Hassan, Efficient three-dimensional Delaurnay triangularisation with automatic point creation and imposed boundary constaints, *Int. J. Numer. Methods Eng.* **37**, 2005 (1994).
19. D. J. Mavriplis and V. Venkatakrishnan, A unified multigrid solver for the Navier–Stokes equations on mixed element meshes, in *AIAA-95-1666, San Diego, CA, 1995*.
20. V. Parthasarathy, Y. Kallindereris, and K. Nakajima, Hybrid adaption method and directional viscous multigrid with presmatic-tetrahedral meshes, in *AIAA-95-0670, Reno, NV, 1995*.
21. S. J. Sherwin and G. E. Karniadakis, A triangular spectral element method; applications to the incompressible Navier-Stokes equations, *Comput. Meth. Appl. Mech. Eng.* **23**, 83 (1995).
22. S. J. Sherwin and G. E. Karniadakis, Tetrahedral *hp* finite elements: Algorithms and flow simulations, *J. Comput. Phys.* **122**, 191 (1995).
23. B. A. Wingate and J. P. Boyd, Triangular spectral element methods for geophysical fluid dynamics applications, *Houston J. Math.*, 305 (1996).
24. M. Dubiner, Spectral methods on triangles and other domains, *J. Sci. Comput.* **6**, 345 (1991).
25. B. Cockburn and C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework, *Math. Comput.* **52**, 411 (1989).
26. B. Cockburn, S.-Y. Lin, and C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems, *J. Comput. Phys.* **84**, 90 (1989).
27. B. Cockburn, S. Hou, and C.-W. Shu, The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multi-dimensional case, *J. Comput. Phys.* **54**, 545 (1990).
28. B. Cockburn and C.-W. Shu, P^1*-RKDG* method for two-dimensional Euler equations of gas dynamics, in *Proc. Fourth Int. Symp. on CFD, UC Davis, 1991*.
29. R. Biswas, K. Devine, and J. Flaherty, Parallel, adaptive finite element methods for conservation laws, *Appl. Numer. Math.* **14**, 255 (1994).
30. J. Peiro, J. Peraire, and K. Morgan, *2D Felisa System—Reference Manual*, Dept. of Aeronautics, Imperial College, 1994.
31. C. Johnson, *Numerical Solution of Partial Differential Equations by the Finite Element Method*, Cambridge Univ. Press, Cambridge, 1994.
32. S. J. Sherwin and G. E. Karniadakis, A new triangular and tetrahedral basis for high-order finite element methods, *Int. J. Numer. Meth. Eng.* **38**, 3775 (1995).
33. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectra Methods in Fluid Mechanics* (Springer-Verlag, New York, 1987).
34. P. L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, *J. Comput. Phys.* **43**, 357 (1981).
35. Y. Maday and A. T. Patera, Spectral element methods for the Navier–Stokes equations, in *Proc. State of the Art Surveys in Computational Mechanics, ASME, 1987*.
36. I. Lomtev and G. E. Karniadakis, Simulations of viscous supersonic flows on unstructured meshes, AIAA-97-0754, 1997.
37. M. Hammache and M. Gharib, An experimental study of the parallel and oblique vortex shedding from circular cylinders, *J. Fluid Mech.* **232**, 567 (1991).
38. R. D. Henderson, Details of the drag curve near the onset of vortex shedding, *Phys. Fluids* **7**(9), 1 (1995).
39. C. H. K. Williamson and A. Roshko, Measurements of base pressure in the wake of a cylinder at low Reynolds number, *Z. Flugwiss. Weltraumforsch* **14**, 38 (1990).
40. A. Beskok and G. E. Karniadakis, A collocation spectral element method for compressible viscous flows, *in progress*.
41. T. C. E. Warburton, *Spectral/hp Methods on Polymorphic Domains*, Ph.D. thesis, Brown University, Division of Applied Mathematics, in progress.