JOURNAL OF COMPUTATIONAL PHYSICS **147,** 114–146 (1998)
ARTICLE NO. CP986078

The Lagrange–Galerkin Spectral Element Method on Unstructured Quadrilateral Grids

Francis X. Giraldo

Naval Research Laboratory, Monterey, California 93943
E-mail: giraldo@nrlmry.navy.mil

Received December 31, 1997; revised June 23, 1998

The purpose of this paper is to introduce a new method formed by fusing the Lagrange–Galerkin and spectral element methods. The Lagrange–Galerkin method traces the characteristic curves of the solution and, consequently, is very well suited for resolving the nonlinearities introduced by the advection operator of the fluid dynamics equations. Spectral element methods are essentially higher order finite element methods that exhibit spectral (exponential) convergence, provided that the solution is a smooth function. By combining these two methods, a numerical scheme can be constructed that resolves, with extremely high precision, the nonlinearities of the advection terms and the smooth regions of the flow generated by the diffusion terms. This paper describes the construction of the Lagrange–Galerkin spectral element method which permits the use of any grid type including unstructured grids. The only restriction at the moment is that the grid elements be quadrilaterals. The stability analysis of both methods demonstrates why these two methods are so powerful individually and how their fusion leads to an improved scheme. The Lagrange–Galerkin spectral element method is validated using the 1D and 2D advection and advection–diffusion equations. The results of the stability analysis and the numerical experiments demonstrate the utility of such an approach.

Key Words: advection–diffusion equation; finite element; flux-corrected transport (fct); icosahedral grid; Lagrange–Galerkin; Legendre polynomial; semi-Lagrangian; spectral element; unstructured grid.

1. INTRODUCTION

The spectral element method combines the benefits extracted from both the spectral method and the finite element method. The spectral element method can be described as a method that can automatically generate any order polynomial basis function, as in the spectral method, while allowing for the geometrical flexibility enjoyed by the finite element method. The spectral element method may use any type of Jacobi polynomial to define

0021-9991/98 $25.00

the basis functions but typically either Chebyshev or Legendre polynomials are used. In this paper, Legendre polynomials are used. The prosperity of the spectral element method can be attributed to the fact that any order polynomial can be generated automatically, concurrently with its numerical integration rule. If we select the Gauss quadrature points for the integration rules to be the collocation points we get orthogonal basis functions which means that the mass matrices are then diagonal. There is also no need to define the basis functions explicitly because we can define implicit relations *a priori* for the inner products of the functions and their derivatives. Since the collocation points are not equi-spaced, staggered grids can be generated automatically by using varying order polynomials for the different variables (say the pressure and velocity in Navier–Stokes) as is done in [9] for the shallow-water equations which then satisfies the Babuska–Brezzi condition, thereby avoiding the development of any spurious pressure modes. Many researchers have used the spectral element method successfully for Stokes flows [13], the shallow-water equations [9, 10], and Navier–Stokes.

The advection terms in the governing equations of fluid motion present formidable challenges to many discretization methods, including Galerkin methods. These terms make the operator non-self-adjoint and as a result, prevents the optimization of these methods. By combining the time derivative and the advection terms into the Lagrangian derivative and then discretizing the resulting operator, many of these difficulties are circumvented; this is the Lagrange–Galerkin method. This method not only increases the solution accuracy but also allows for much larger time steps therefore making it potentially more efficient than Eulerian methods. Lagrange–Galerkin methods have increased in popularity in the last 10 years because they offer increased accuracy and efficiency by virtue of their independence on the CFL condition. Researchers have used this method successfully for advection [5, 11], advection–diffusion [7], shallow water, and Navier–Stokes.

There have been few attempts at combining Lagrangian methods with high order spatial discretization methods. In [15], a method is presented which combines the spectral method with the method of characteristics. While this approach offers accuracy it does not offer geometric flexibility due to the restriction on the grid dictated by the spectral method. In [8], a characteristic-based spectral element method is introduced which uses an explicit time-stepping scheme in order to get the values at the foot of the characteristics. This approach preserves the Eulerian-like structure of the equations, thereby avoiding the search for the departure points that is typically associated with the Lagrange–Galerkin method. It is unclear how the structure-preserving schemes affect the stability. In any event, our approach is shown to be unconditionally stable and, because it is constructed in a general fashion, it is perfectly suited for all types of grids, including unstructured grids.

The following section describes the implementations of both the spectral element and Lagrange–Galerkin spectral element methods on the 1D advection–diffusion equation.

2. ONE-DIMENSIONAL ADVECTION-DIFFUSION EQUATION

2.1. Spectral Element Method

For stability analysis purposes and for describing the algorithms, we shall use the 1D advection–diffusion equation. The time discretization is handled in this paper by the θ algorithm which defines a family of explicit and implicit schemes. The governing equation

is defined as

$$\frac{\partial \varphi}{\partial t} + u\frac{\partial \varphi}{\partial x} = \nu \frac{\partial^2 \varphi}{\partial x^2}. \tag{1}$$

The spatial discretization by the spectral element method follows closely the discretization of the finite element method. After applying the method of weighted residuals we obtain the relation

$$\int_{x_0}^{x_0+\Delta x} \psi_i \psi_j \, dx \, \frac{\partial \varphi_j}{\partial t} + \int_{x_0}^{x_0+\Delta x} \psi_i \psi_k u_k \frac{\partial \psi_j}{\partial x} \, dx \, \varphi_j$$
$$= -\nu \int_{x_0}^{x_0+\Delta x} \frac{\partial \psi_i}{\partial x}\frac{\partial \psi_j}{\partial x} \, dx \, \varphi_j + \nu \left[\psi_i \frac{\partial \psi_j}{\partial x} \varphi_j \right]_{x_0}^{x_0+\Delta x},$$

where ψ are the basis functions in terms of physical space, and $i, j, k = 0, \ldots, N$, where N is the order of the basis functions. After mapping to computational space by virtue of $\xi = (2/\Delta x)(x - x_0) - 1$ we arrive at

$$M_{ij} = \int_{x_0}^{x_0+\Delta x} \psi_i \psi_j \, dx = \frac{\Delta x}{2} \int_{-1}^{+1} h_i(\xi) h_j(\xi) \, d\xi$$
$$A_{ij} = \int_{x_0}^{x_0+\Delta x} \psi_i \psi_k u_k \frac{\partial \psi_j}{\partial x} \, dx = \int_{-1}^{+1} h_i(\xi) h_k(\xi) u_k \frac{\partial h_j}{\partial \xi}(\xi) \, d\xi$$
$$D_{ij} = \nu \int_{x_0}^{x_0+\Delta x} \frac{\partial \psi_i}{\partial x}\frac{\partial \psi_j}{\partial x} \, dx = \nu \frac{2}{\Delta x} \int_{-1}^{+1} \frac{\partial h_i}{\partial \xi}(\xi) \frac{\partial h_j}{\partial \xi}(\xi) \, d\xi$$
$$R_i = \nu \left[\psi_i \frac{\partial \psi_j}{\partial x} \varphi_j \right]_{x_0}^{x_0+\Delta x} = \nu \frac{2}{\Delta x} \left[h_i(\xi) \frac{\partial h_j}{\partial \xi}(\xi) \varphi_j \right]_{-1}^{+1},$$

where $h(\xi)$ are the Legendre cardinal basis functions, M is the mass matrix, A is the advection matrix, D is the diffusion matrix, and R is the boundary vector. The time discretization by the θ algorithm yields the relation

$$[M_{ij} + \Delta t \theta (A_{ij} + D_{ij})]\varphi_j^{n+1} = [M_{ij} - \Delta t (1-\theta)(A_{ij} + D_{ij})]\varphi_j^n$$
$$+ \Delta t \left[\theta R_i^{n+1} + (1-\theta) R_i^n \right] \tag{2}$$

which for $\theta = 0$ yields the forward Euler (explicit and first order in time), for $\theta = \frac{1}{2}$ yields the trapezoidal rule (implicit and second order), and for $\theta = 1$ yields the backward Euler (implicit and first order). Because of the higher order achieved with the trapezoidal rule ($\theta = \frac{1}{2}$) this is the value that is used throughout the paper.

Typically, explicit methods have been used in conjunction with the spectral element method, although there have been some attempts at using implicit methods [9]. The advantage in using explicit methods arise from the nonsymmetry and large bandwidths of the system of equations resulting from the spectral element discretization. Because of this nonsymmetry, it makes it very difficult to select a robust and efficient matrix solver. In [9], a GMRES solver was used but was not found sufficiently cost-effective. In this paper, an LU decomposition (direct solver) is used for simplicity. Therefore, an implicit

time discretization may not be the most efficient possibility, but it is nonetheless used in order to show how the Lagrange–Galerkin formulation is constructed on top of this Eulerian version. In the Lagrange–Galerkin version described in the next section, the time discretization must be implicit in order for the method to be unconditionally stable.

The spatial accuracy is determined by the order of the basis functions and will be of order $O(\Delta x^{N+1})$. As an example, for $N=1$ we have linear elements, but second-order spatial accuracy. In this article, the element basis functions are the Legendre cardinal functions (for details on these basis functions see [13]). To keep the algorithm as general and as automatic as possible, we evaluate the integrals numerically. In other words, the mass matrix is evaluated as

$$M_{ij} = \frac{\Delta x}{2} \sum_{q=0}^{Q} w_q h_i(\xi_q) h_j(\xi_q),$$

where Q represents the number of Legendre–Gauss–Lobatto quadrature points. The remaining matrices are evaluated similarly. In order to obtain exact integrations for this matrix, at least $Q = N+1$ quadrature points are required. This is the minimum required because an integration rule of order N will integrate exactly any polynomial of order $2N-1$, but since the mass matrix contains polynomials of order $2N$, then higher order integration rules are required. The rules given above (i.e., $Q = N+1$) will integrate exactly polynomials of order $2N+1$, which is more than sufficient. In typical implementations of the spectral element method, $Q = N$ is used. In Section 3 (Stability Analysis), the effect of using exact versus nonexact integration is discussed.

2.2. Lagrange–Galerkin Spectral Element Method

Lagrange–Galerkin methods belong to the general class of upwinding methods. These methods incorporate characteristic information into the numerical scheme. The Lagrangian form of (1) is

$$\frac{d\varphi}{dt} = \nu \frac{\partial^2 \varphi}{\partial x^2} \tag{3}$$

$$\frac{dx}{dt} = u(x,t), \tag{4}$$

where d/dt denotes the total derivative and is defined as

$$\frac{d}{dt} = \frac{\partial}{\partial t} + u \frac{\partial}{\partial x}.$$

Applying the method of weighted residuals directly onto the Lagrangian operator yields the two-time level direct Lagrange–Galerkin method (for an explanation of the weak versus the direct Lagrange-Galerkin method see [5])

$$[M_{ij} + \Delta t \theta D_{ij}]\varphi_j^{n+1} = [M_{ij} - \Delta t(1-\theta)D_{ij}]\tilde{\varphi}_j^n + \Delta t\left[\theta R_i^{n+1} + (1-\theta)\tilde{R}_i^n\right], \tag{5}$$

where $\varphi^{n+1} = \varphi(x, t+\Delta t)$ and $\tilde{\varphi}^n = \varphi(x-\alpha, t)$ are the solutions at the arrival $(x_a = x)$

and departure points ($x_d = x - \alpha$), respectively. Integrating (4) by the mid-point rule yields

$$\alpha = \Delta t u\left(x - \frac{\alpha}{2}, t + \frac{\Delta t}{2}\right) \tag{6}$$

which defines a recursive relation for the Lagrange–Galerkin departure points.

As in the Eulerian version described previously, the value $\theta = \frac{1}{2}$ is used throughout for the Lagrange–Galerkin method. Note that, although this defines an implicit scheme, it yields a much easier system of linear equations to solve because the global coefficient matrix is symmetric. This is different from the Eulerian version and this property is critical to the efficiency of the Lagrange–Galerkin method. Now, instead of using an LU decomposition or some nonsymmetric iterative solver, we can use classic methods that are robust and efficient such as the conjugate gradient method. However, for the sake of fairness in comparing the computational costs (Table VII) of the algorithms an LU decomposition is also used for the Lagrange–Galerkin method.

We still need to consider how we are going to interpolate the values of φ at the departure points ($x_d = x - \alpha$). Interpolation is required because, in general, the departure points will not fall on grid points but rather between them. Typically, the interpolations are constructed using one of the following three types of functions: Lagrange, Hermite, or spline polynomials. The difficulty with these methods is that they require some structure in the grid (i.e., curvilinear coordinates) and while this does not affect the 1D case, it would be quite limiting in the 2D case where the elements are not required to have a true structure. (A caveat is in order here: cubic splines can be constructed from unstructured data, but these methods are generally impractical for CFD-type computations because they are prohibitively expensive.) The incentive for combining the spectral element method with the Lagrange–Galerkin method is due to the high order polynomials of the spectral element method that are also locally defined within each element. We can determine in which spectral element a given departure point lies and then use the basis functions within the element to construct the interpolation required by the Lagrange–Galerkin method. Since the order of the basis functions are typically high order, the interpolation will be of a sufficiently high order to ensure the high order spatial and temporal accuracy of the numerical scheme.

Since the Legendre cardinal basis functions have to be able to interpolate any departure point, they have to be constructed explicitly. The Legendre cardinal basis functions can be written using the definition for Lagrange polynomials,

$$h_i(\xi) = \prod_{\substack{j=0,\\ j\neq i}}^{N} \left(\frac{\xi - \xi_j}{\xi_i - \xi_j}\right) \tag{7}$$

and its derivatives are

$$\frac{\partial h_i}{\partial \xi}(\xi) = \sum_{\substack{k=0,\\ k\neq i}}^{N} \prod_{\substack{j=0,\\ j\neq i}}^{N} \left(\frac{1}{\xi_i - \xi_k}\right) \cdot \left(\frac{\xi - \xi_j}{\xi_i - \xi_j}\right), \tag{8}$$

where the ξ_i, ξ_j, ξ_k are the permutations of the Legendre–Gauss–Lobatto (collocation) points. These two relations are very general and valid for any order Legendre cardinal

basis function and can be used quite easily to generate the desired interpolating polynomial for the Lagrange–Galerkin method. In the next section, the stability analysis of both methods is performed on the 1D advection–diffusion equation.

3. STABILITY ANALYSIS

3.1. Spectral Element Method

Equation (2) can be written in the following more explicit form:

$$\sum_{q=0}^{Q} w_q \left[h_i(\xi_q)h_j(\xi_q) + 2\theta\frac{\Delta t u}{\Delta x} h_i(\xi_q)\frac{\partial h_j}{\partial \xi}(\xi_q) + 4\theta\frac{\Delta t \nu}{\Delta x^2}\frac{\partial h_i}{\partial \xi}(\xi_q)\frac{\partial h_j}{\partial \xi}(\xi_q)\right]\varphi_j^{n+1}$$
$$= \sum_{q=0}^{Q} w_q \left[h_i(\xi_q)h_j(\xi_q) - 2(1-\theta)\frac{\Delta t u}{\Delta x} h_i(\xi_q)\frac{\partial h_j}{\partial \xi}(\xi_q) \right.$$
$$\left. - 4(1-\theta)\frac{\Delta t \nu}{\Delta x^2}\frac{\partial h_i}{\partial \xi}(\xi_q)\frac{\partial h_j}{\partial \xi}(\xi_q)\right]\varphi_j^{n}.$$

These are element equations but in order to study the stability at the nodes the grid point equations must be obtained. Therefore, the contributions of the elements to each node must be summed. Below, the analysis for the external and internal nodes (collocation points) are derived.

DEFINITION 1. External nodes are those points which define the boundary of a spectral element. In 1D, the external nodes are the left- and right-most collocation points corresponding to $i = 0$ and N.

DEFINITION 2. Internal nodes are any other collocation points corresponding to $i = 1, \ldots, N-1$.

By substituting the nondimensional parameters

$$\sigma = \frac{\Delta t u}{\Delta x} \text{ (Courant number)}, \quad \mu = \frac{\Delta t \nu}{\Delta x^2} \text{ (diffusion coefficient)}$$

and introducing the Fourier modes

$$\varphi_{\jmath+(1+\xi_j)/2}^{n+1} = G^{n+1} e^{\imath(\jmath+(1+\xi_j)/2)\phi}, \quad \varphi_{\jmath-(1-\xi_j)/2}^{n+1} = G^{n+1} e^{\imath(\jmath-(1-\xi_j)/2)\phi},$$

where ϕ is the phase angle and $\imath = \sqrt{-1}$ gives the total contributions to the external node $\jmath$ to be

$$\sum_{j=0}^{N} \left[\left(M_j^l + 2\theta\sigma A_j^l + 4\theta\mu D_j^l\right) G^{n+1} e^{\imath(\jmath-(1-\xi_j)/2)\phi} \right.$$
$$\left. + \left(M_j^r + 2\theta\sigma A_j^r + 4\theta\mu D_j^r\right) G^{n+1} e^{\imath(\jmath+(1+\xi_j)/2)\phi} \right]$$
$$= \sum_{j=0}^{N} \left[\begin{array}{l} \left(M_j^l - 2(1-\theta)\sigma A_j^l - 4(1-\theta)\mu D_j^l\right) G^n e^{\imath(\jmath-(1-\xi_j)/2)\phi} \\ \quad + \left(M_j^r - 2(1-\theta)\sigma A_j^r - 4(1-\theta)\mu D_j^r\right) G^n e^{\imath(\jmath+(1+\xi_j)/2)\phi} \end{array} \right],$$

where

$$M_j^l = \sum_{q=0}^{Q} w_q h_N(\xi_q) h_j(\xi_q), \qquad M_j^r = \sum_{q=0}^{Q} w_q h_0(\xi_q) h_j(\xi_q)$$

$$A_j^l = \sum_{q=0}^{Q} w_q h_N(\xi_q) \frac{\partial h_j}{\partial \xi}(\xi_q), \qquad A_j^r = \sum_{q=0}^{Q} w_q h_0(\xi_q) \frac{\partial h_j}{\partial \xi}(\xi_q)$$

$$D_j^l = \sum_{q=0}^{Q} w_q \frac{\partial h_N}{\partial \xi}(\xi_q) \frac{\partial h_j}{\partial \xi}(\xi_q), \quad D_j^r = \sum_{q=0}^{Q} w_q \frac{\partial h_0}{\partial \xi}(\xi_q) \frac{\partial h_j}{\partial \xi}(\xi_q),$$

and the superscripts l and r denote the contributions from the left and right elements to the node j. Upon dividing by $G^n e^{\imath j\phi}$, using Euler's formula $e^{\pm \imath\phi} = \cos\phi \pm \imath \sin\phi$, and multiplying by the complex conjugate, we get

$$G = \mathrm{Re}(G) - \imath\, \mathrm{Im}(G),$$

where

$$\mathrm{Re}(G) = \frac{\sum_{i=0}^{N}\sum_{j=0}^{N}\left(\mathrm{Re}_i^r\,\mathrm{Re}_j^l + \mathrm{Im}_i^r\,\mathrm{Im}_j^l\right)}{\sum_{i=0}^{N}\sum_{j=0}^{N}\left(\mathrm{Re}_i^l\,\mathrm{Re}_j^l + \mathrm{Im}_i^l\,\mathrm{Im}_j^l\right)}, \quad \mathrm{Im}(G) = \frac{\sum_{i=0}^{N}\sum_{j=0}^{N}\left(\mathrm{Re}_j^l\,\mathrm{Im}_i^r - \mathrm{Re}_i^r\,\mathrm{Im}_j^l\right)}{\sum_{i=0}^{N}\sum_{j=0}^{N}\left(\mathrm{Re}_i^l\,\mathrm{Re}_j^l + \mathrm{Im}_i^l\,\mathrm{Im}_j^l\right)}, \tag{9}$$

and

$$\begin{aligned}
\mathrm{Re}_j^r &= \left(M_j^l - 2(1-\theta)\sigma A_j^l - 4(1-\theta)\mu D_j^l\right)\cos\left(\frac{1-\xi_j}{2}\right)\phi \\
&\quad + \left(M_j^r - 2(1-\theta)\sigma A_j^r - 4(1-\theta)\mu D_j^r\right)\cos\left(\frac{1+\xi_j}{2}\right)\phi \\
\mathrm{Im}_j^r &= \left(M_j^l - 2(1-\theta)\sigma A_j^l - 4(1-\theta)\mu D_j^l\right)\sin\left(\frac{1-\xi_j}{2}\right)\phi \\
&\quad - \left(M_j^r - 2(1-\theta)\sigma A_j^r - 4(1-\theta)\mu D_j^r\right)\sin\left(\frac{1+\xi_j}{2}\right)\phi \\
\mathrm{Re}_j^l &= \left(M_j^l + 2\theta\sigma A_j^l + 4\theta\mu D_j^l\right)\cos\left(\frac{1-\xi_j}{2}\right)\phi \\
&\quad + \left(M_j^r + 2\theta\sigma A_j^r + 4\theta\mu D_j^r\right)\cos\left(\frac{1+\xi_j}{2}\right)\phi \\
\mathrm{Im}_j^l &= \left(M_j^l + 2\theta\sigma A_j^l + 4\theta\mu D_j^l\right)\sin\left(\frac{1-\xi_j}{2}\right)\phi \\
&\quad - \left(M_j^r + 2\theta\sigma A_j^r + 4\theta\mu D_j^r\right)\sin\left(\frac{1+\xi_j}{2}\right)\phi
\end{aligned}$$

For the internal node j the corresponding relation is

$$\sum_{j=0}^{N}\left[\left(M_j^r + 2\theta\sigma A_j^r + 4\theta\mu D_j^r\right)G^{n+1}e^{\imath(\jmath+(1+\xi_j)/2)\phi}\right]$$
$$= \sum_{j=0}^{N}\left[\left(M_j^r - 2(1-\theta)\sigma A_j^r - 4(1-\theta)\mu D_j^r\right)G^{n}e^{\imath(\jmath+(1+\xi_j)/2)\phi}\right]$$

along with

$$\mathrm{Re}_j^r = +\left(M_j^r - 2(1-\theta)\sigma A_j^r - 4(1-\theta)\mu D_j^r\right)\cos\left(\frac{1+\xi_j}{2}\right)\phi$$

$$\mathrm{Im}_j^r = -\left(M_j^r - 2(1-\theta)\sigma A_j^r - 4(1-\theta)\mu D_j^r\right)\sin\left(\frac{1+\xi_j}{2}\right)\phi$$

$$\mathrm{Re}_j^l = +\left(M_j^r + 2\theta\sigma A_j^r + 4\theta\mu D_j^r\right)\cos\left(\frac{1+\xi_j}{2}\right)\phi$$

$$\mathrm{Im}_j^l = -\left(M_j^r + 2\theta\sigma A_j^r + 4\theta\mu D_j^r\right)\sin\left(\frac{1+\xi_j}{2}\right)\phi.$$

Using these relations the amplification factor is defined as

$$|G| = \sqrt{\mathrm{Re}^2(G) + \mathrm{Im}^2(G)} \tag{10}$$

and the dispersion error as

$$\epsilon_{\mathbf{\Phi}} = \mathbf{\Phi}/\sigma\phi, \tag{11}$$

where

$$\mathbf{\Phi} = \arctan\left(\frac{\mathrm{Im}(G)}{\mathrm{Re}(G)}\right)$$

is the phase angle.

3.2. Lagrange–Galerkin Spectral Element Method

Equation (5) can be written in the more explicit form:

$$\sum_{q=0}^{Q} w_q\left[h_i(\xi_q)h_j(\xi_q) + 4\theta\frac{\Delta t\nu}{\Delta x^2}\frac{\partial h_i}{\partial\xi}(\xi_q)\frac{\partial h_j}{\partial\xi}(\xi_q)\right]\varphi_j^{n+1}$$
$$= \sum_{q=0}^{Q} w_q\left[h_i(\xi_q)h_j(\xi_q) - 4(1-\theta)\frac{\Delta t\nu}{\Delta x^2}\frac{\partial h_i}{\partial\xi}(\xi_q)\frac{\partial h_j}{\partial\xi}(\xi_q)\right]\tilde{\varphi}_j^{n}.$$

However, we still need to introduce the interpolation $\tilde{\varphi}_j^n$. In the Lagrangian version of this algorithm, the spectral element basis functions are used as the interpolation functions. In other words, first we need to determine in which element the departure point lies which can be performed efficiently with the bisection method in 1D and the quadtree method in 2D.

The interpolation within the element is written as

$$\tilde{\varphi}_j^n = \begin{cases} \sum_{k=0}^N \varphi^n_{\jmath-p-\Delta p-(1-\xi_k)/2} h_k(\tilde{\xi}_j), & \text{if } \tilde{\alpha} \le 0, \\ \sum_{k=0}^N \varphi^n_{\jmath-p-\Delta p+(1+\xi_k)/2} h_k(\tilde{\xi}_j), & \text{if } \tilde{\alpha} > 0, \end{cases}$$

where $\tilde{\xi}_j = \xi_j - (\tilde{\alpha}+1)$ and $\tilde{\alpha} \in [-1,1]$ is the real part of the Courant number σ and is related to it by the relation

$$\sigma = p + \frac{\tilde{\alpha}+1}{2}, \tag{12}$$

where p is the integer part of σ. In addition, the following definitions are also required:

$$\tilde{\xi}_j = \begin{cases} \tilde{\xi}_j = \tilde{\xi}_j + 2, & \Delta p = 1 \quad \text{if } \tilde{\xi}_j < -1, \\ \tilde{\xi}_j, & \Delta p = 0 \quad \text{if } \tilde{\xi}_j \ge -1. \end{cases} \tag{13}$$

The parameter Δp is required because in most instances the departure points of the external nodes, of a given element, may lie within two different grid elements; in this situation, Δp determines in which of the two grid elements the departure points of the internal nodes lie.

Introducing the Fourier modes

$$\varphi^n_{\jmath-p-\Delta p+(1+\xi_j)/2} = G^n e^{\imath(\jmath-p-\Delta p+(1+\xi_j)/2)\phi}, \quad \varphi^n_{\jmath-p-\Delta p-(1-\xi_j)/2} = G^n e^{\imath(\jmath-p-\Delta p-(1-\xi_j)/2)\phi}$$

we get the total contributions to the external node $\jmath$ to be

$$\begin{aligned} &\sum_{j=0}^N \left[\left(M_j^l + 4\theta\mu D_j^l\right) G^{n+1} e^{\imath(\jmath-(1-\xi_j)/2)\phi} + \left(M_j^r + 4\theta\mu D_j^r\right) G^{n+1} e^{\imath(\jmath+(1+\xi_j)/2)\phi} \right] \\ &\quad = \sum_{j=0}^N \left[\begin{array}{l} \left(M_j^l - 4(1-\theta)\mu D_j^l\right) \left\{ \sum_{k=0}^N G^n e^{\imath(\jmath-p-\Delta p-(1-\xi_k)/2)\phi} h_k(\tilde{\xi}_j) \right\} \\ \quad + \left(M_j^r - 4(1-\theta)\mu D_j^r\right) \left\{ \sum_{k=0}^N G^n e^{\imath(\jmath-p-\Delta p+(1+\xi_k)/2)\phi} h_k(\tilde{\xi}_j) \right\} \end{array} \right] \end{aligned}$$

and after simplifying they yield

$$G = [\mathrm{Re}(G) - \imath\, \mathrm{Im}(G)] e^{-\imath p\phi},$$

where $\mathrm{Re}(G)$ and $\mathrm{Im}(G)$ are given by (9) and

$$\begin{aligned} \mathrm{Re}_j^r = &\left(M_j^l - 4(1-\theta)\mu D_j^l\right) \cos\left(\Delta p + \frac{1-\xi_j}{2}\right)\phi \\ &+ \left(M_j^r - 4(1-\theta)\mu D_j^r\right) \cos\left(-\Delta p + \frac{1+\xi_j}{2}\right)\phi \end{aligned}$$

$$\begin{aligned}
\mathrm{Im}_j^r &= \left(M_j^l - 4(1-\theta)\mu D_j^l\right)\sin\left(\Delta p + \frac{1-\xi_j}{2}\right)\phi \\
&\quad - \left(M_j^r - 4(1-\theta)\mu D_j^r\right)\sin\left(-\Delta p + \frac{1+\xi_j}{2}\right)\phi \\
\mathrm{Re}_j^l &= \left(M_j^l + 4\theta\mu D_j^l\right)\cos\left(\Delta p + \frac{1-\xi_j}{2}\right)\phi \\
&\quad + \left(M_j^r + 4\theta\mu D_j^r\right)\cos\left(-\Delta p + \frac{1+\xi_j}{2}\right)\phi \\
\mathrm{Im}_j^l &= \left(M_j^l + 4\theta\mu D_j^l\right)\sin\left(\Delta p + \frac{1-\xi_j}{2}\right)\phi \\
&\quad - \left(M_j^r + 4\theta\mu D_j^r\right)\sin\left(-\Delta p + \frac{1+\xi_j}{2}\right)\phi.
\end{aligned}$$

For the internal node $\jmath$ the corresponding relation is

$$\begin{aligned}
&\sum_{j=0}^{N}\left[\left(M_j^r + 4\theta\mu D_j^r\right)G^{n+1}e^{\imath(\jmath+(1+\xi_j)/2)\phi}\right] \\
&\quad = \sum_{j=0}^{N}\left[\left(M_j^r - 4(1-\theta)\mu D_j^r\right)\left\{\sum_{k=0}^{N} G^n e^{\imath(\jmath-p-\Delta p+(1+\xi_k)/2)\phi} h_k(\tilde{\xi}_j)\right\}\right],
\end{aligned}$$

along with

$$\begin{aligned}
\mathrm{Re}_j^r &= +\left(M_j^r - 4(1-\theta)\mu D_j^r\right)\cos\left(-\Delta p + \frac{1+\xi_j}{2}\right)\phi \\
\mathrm{Im}_j^r &= -\left(M_j^r - 4(1-\theta)\mu D_j^r\right)\sin\left(-\Delta p + \frac{1+\xi_j}{2}\right)\phi \\
\mathrm{Re}_j^l &= +\left(M_j^r + 4\theta\mu D_j^r\right)\cos\left(-\Delta p + \frac{1+\xi_j}{2}\right)\phi \\
\mathrm{Im}_j^l &= -\left(M_j^r + 4\theta\mu D_j^r\right)\sin\left(-\Delta p + \frac{1+\xi_j}{2}\right)\phi.
\end{aligned}$$

The amplification factor is given by (10), but the dispersion error is now defined as

$$\epsilon_\Phi = \frac{p\phi + \Phi}{\sigma\phi}. \tag{14}$$

Remark 1. The 2D stability analysis follows quite easily from the 1D analysis. That is, the theory behind the 2D analysis follows closely the 1D analysis but the 2D analysis requires much more algebra and book keeping due to the 2D analogs of Eqs. (12) and (13).

The 1D analysis yields much insight into the stability of the 2D numerical scheme; however, the 1D analysis tells us nothing about the effects of using distorted quadrilateral elements on the stability and accuracy of the scheme.

3.3. Stability Analysis Results

3.3.1. *Spectral Element Method*

The amplitude factors and dispersion errors for both methods are illustrated up to a Courant number of $\sigma = 4$. In addition, results are illustrated for exact ($Q = (2N + 1)/2$) and nonexact ($Q = N$) integration for the spectral element method. For brevity, comparisons for the Lagrange–Galerkin spectral element method using exact and non–exact integration are not shown because the spectral element method contains all of the matrices contained within the Lagrangian version. However, the results for the external and internal nodes of both methods are presented.

Figures 1 and 2 show the dispersion errors for the spectral element method for pure advection using exact and nonexact integration, respectively. (The amplification factors are not illustrated because this method experiences no amplification errors.) For $N = 1$, large differences exist between the exact and nonexact methods. Note that a discontinuity appears to be present for all phase angles at $\sigma = 0$. This just shows that at this Courant number nothing is happening (i.e. stationary flow) and so the amplitude factor and the normalized dispersion error should be 1, meaning that no errors exist at all. For $N = 2$, the differences are obvious but they are not too great. Finally, for $N \geq 4$ no differences exist between the two methods. This is also true for the internal nodes. From here on, only the nonexact integration results are illustrated.

Figure 2 shows that for $N = 1$, the scheme is atrocious and in fact is never used. However, the exact integration method is equivalent to the finite element method with linear elements. At $N = 2$, the situation is much improved but lagging and leading errors are still present, albeit quite small, for $\sigma \leq \frac{1}{2}$. For $N \geq 4$, the scheme suffers absolutely no phase errors for $\sigma \leq \frac{1}{4}$. This is also true for the internal nodes (Fig. 3). For $\frac{1}{4} \leq \sigma \leq \frac{1}{2}$ the scheme suffers errors for the short waves (large ϕ), but is nondispersive for the long waves. As we increase σ, especially beyond $\frac{1}{2}$, the scheme continues to experience lagging errors for the short waves and for a large part of the long waves. The scheme is well behaved only for phase angles $\phi \leq \pi/4$. This analysis shows that it does not make sense to use Courant numbers that are too large not because of stability reasons but for accuracy. It does make sense, however, to use high Courant numbers if some sort of dissipation mechanism is introduced which will hinder the propagation of the dispersive waves. Lagrange–Galerkin methods do precisely this.

3.3.2. *Lagrange–Galerkin Spectral Element Method*

Figures 4 and 5 show the external node amplitude factors and dispersion errors, respectively, for the Lagrange–Galerkin spectral element method for pure advection while Fig. 6 shows the results for the internal nodes. For $N = 1$, the amplification factors are clearly cyclical and do not change for different σ, but rather are only a function of the real part $\tilde{\alpha}$ (as opposed to the integer part p) of σ. These results show that for near $\sigma = 0$ or 1

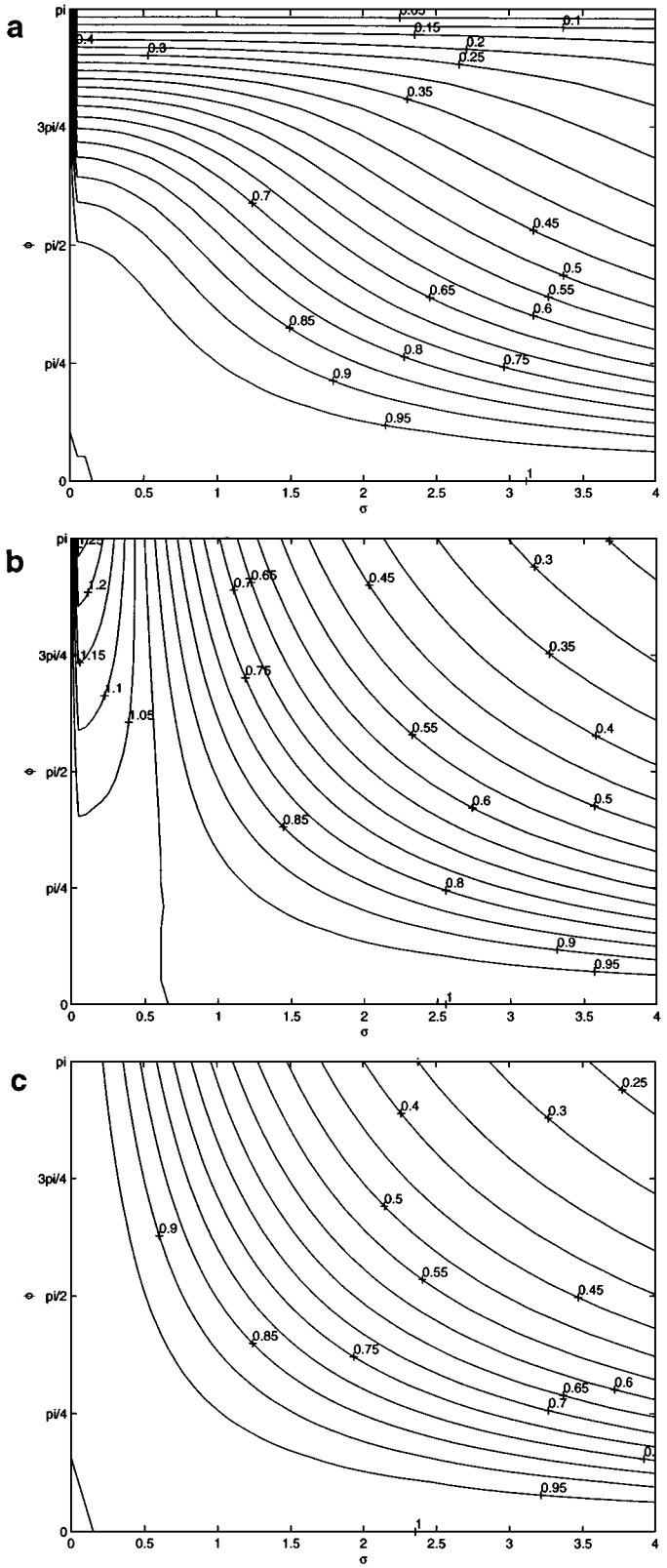

FIG. 1. The external node dispersion error for the 1D advection equation as a function of σ (Courant number) and ϕ (phase angle) for the spectral element method using exact integration for (a) $N = 1$; (b) $N = 2$, and (c) $N \geq 4$.

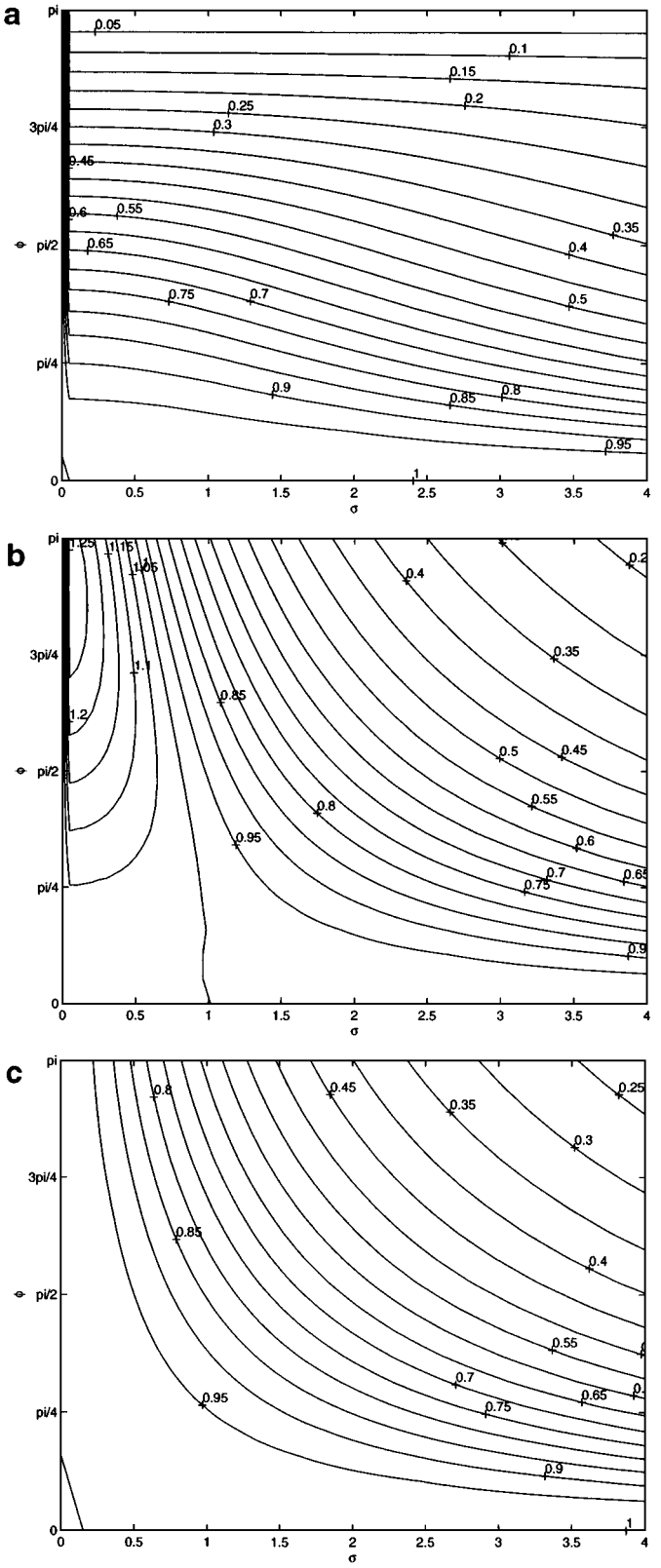

FIG. 2. The external node dispersion error for the 1D advection equation for the spectral element method using nonexact integration for (a) $N = 1$, (b) $N = 2$, and (c) $N \geq 4$.

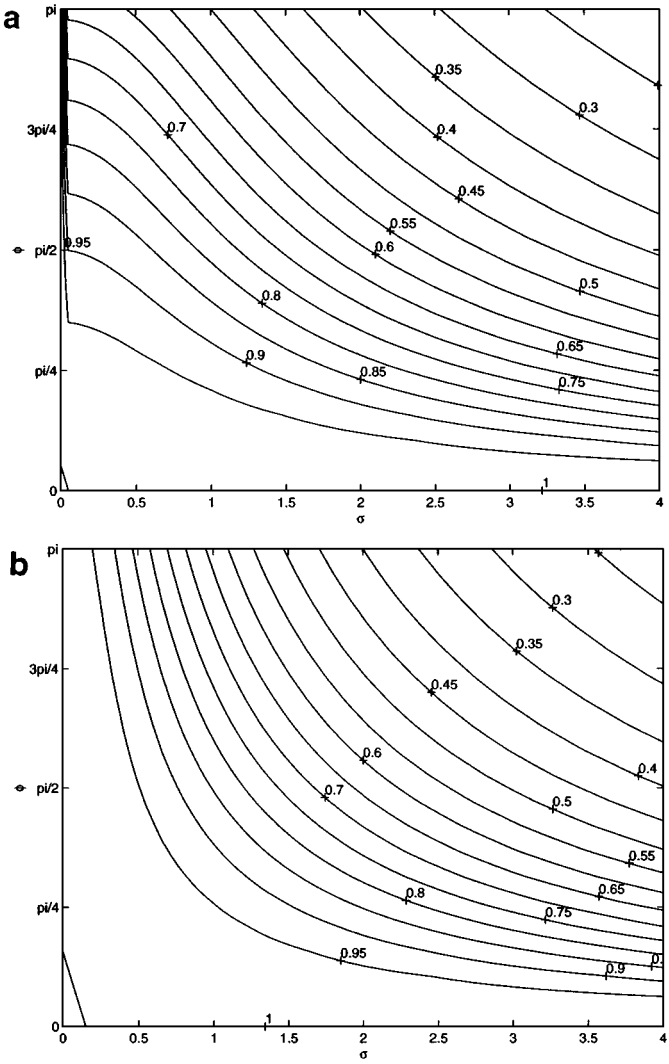

FIG. 3. The internal node dispersion error for the 1D advection equation for the spectral element method using nonexact integration for (a) $N = 2$ and (b) $N \geq 4$.

the scheme exhibits absolutely no dissipation and very little dispersion, but for $\sigma = \frac{1}{2}$ the scheme is excessively dissipative for the short waves, but this behavior is desirable because these tend to be where the scheme is most dispersive. For the long waves, the scheme is slightly dissipative and nondispersive. This scheme is in fact identical to the typical first-order upwinding scheme, at least for $\sigma \leq 1$, and the results for this scheme match the results given in [11] for a finite difference semi-Lagrangian method with linear interpolation. Thus the Lagrange–Galerkin method can be likened to the upwinding method but valid for all Courant numbers including those far greater than 1. For $N = 2$, the scheme becomes less dissipative but also less dispersive. For $N \geq 4$ the scheme exhibits neither dissipation nor dispersion errors for both the external and internal nodes; for this reason, the stability plots for the high order cases ($N \geq 4$) are not shown.

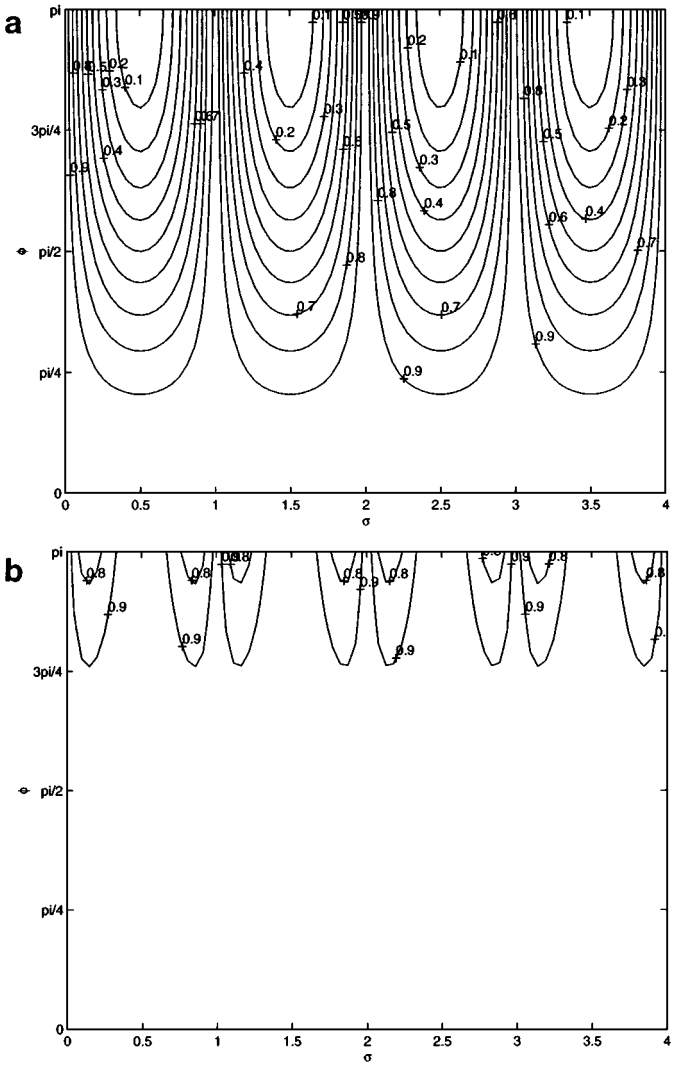

FIG. 4. The external node amplification factor for the 1D advection equation for the Lagrange–Galerkin spectral element method using nonexact integration for (a) $N = 1$ and (b) $N = 2$.

For the spectral element method with diffusion, the dispersion errors are identical to those obtained for pure advection but now due to the viscosity it now experiences some dissipation. For the Lagrange–Galerkin spectral element method with diffusion, the dispersion errors also look rather similar to the pure advection results while the amplitude errors are greater in this case (Fig. 7) and they linger for all σ and N but only for large phase angles.

This analysis shows that the Lagrange–Galerkin spectral element method for $N \geq 4$ works extremely well for advection-dominated flows. For advection–diffusion, there is some dissipation associated with the viscosity which is to be expected and in no way deters from the accuracy of the scheme. In fact, the addition of the Lagrange–Galerkin method to the spectral element method introduces the dissipation mechanism that the spectral element method needs for high Courant numbers without diminishing the accuracy of the scheme.

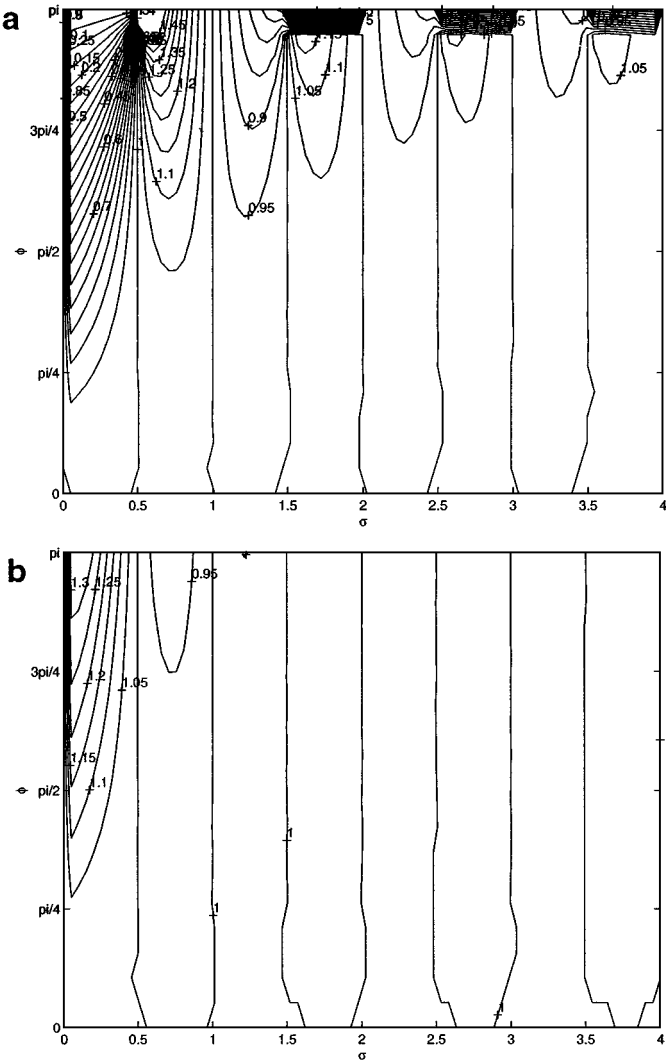

FIG. 5. The external node dispersion error for the 1D advection equation for the Lagrange–Galerkin spectral element method using nonexact integration for (a) $N = 1$ and (b) $N = 2$.

The fusion of the Lagrange–Galerkin method with the spectral element method on the 2D advection-diffusion equation is explored in the following section.

4. TWO-DIMENSIONAL ADVECTION–DIFFUSION EQUATION

4.1. Spectral Element Method

The 2D advection–diffusion equation can be given in a similar fashion to (1) as

$$\frac{\partial \varphi}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla} \varphi = \nu \nabla^2 \varphi \tag{15}$$

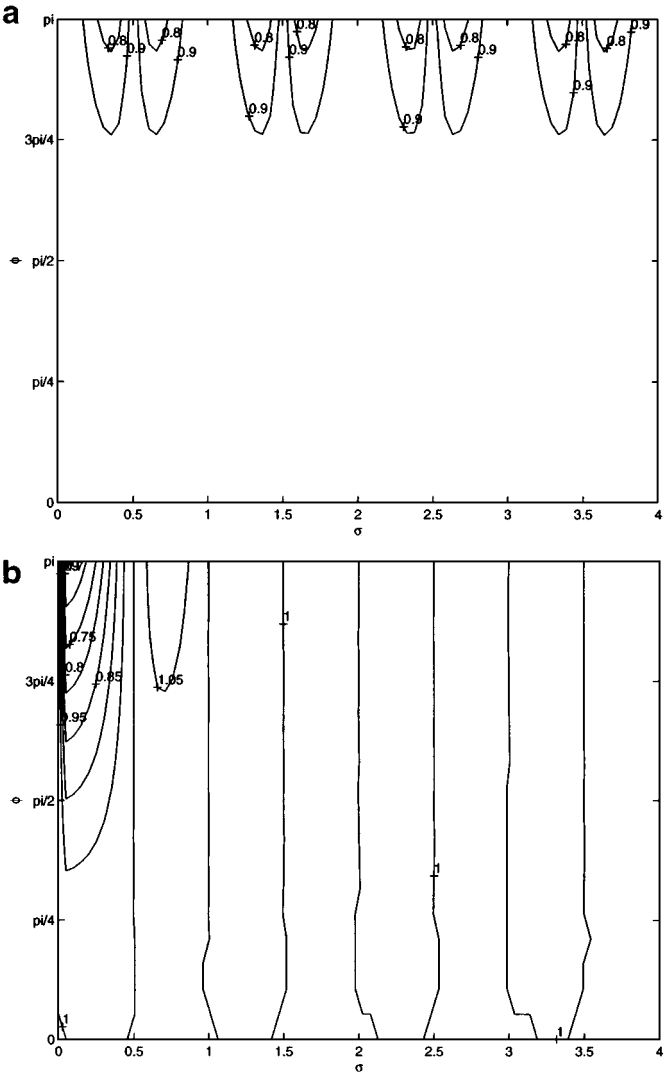

FIG. 6. The internal node (a) amplification factor and (b) dispersion error for the 1D advection equation for the Lagrange–Galerkin spectral element method using nonexact integration for $N = 2$.

and the spatial discretization is written as

$$\int_\Omega \psi_i \psi_j \, d\Omega \frac{\partial \varphi_j}{\partial t} + \int_\Omega \psi_i \psi_k \mathbf{u}_k \cdot \boldsymbol{\nabla} \psi_j \, d\Omega \, \varphi_j$$

$$= -\nu \int_\Omega \boldsymbol{\nabla} \psi_i \cdot \boldsymbol{\nabla} \psi_j \, d\Omega \, \varphi_j + \nu \int_\Gamma (\mathbf{n} \cdot \boldsymbol{\nabla} \varphi) \psi_i \, d\Gamma$$

which yields the time discretization

$$[M_{ijkl} + \Delta t \theta (A_{ijkl} + D_{ijkl})] \varphi_{kl}^{n+1} = [M_{ijkl} - \Delta t (1-\theta)(A_{ijkl} + D_{ijkl})] \varphi_{kl}^{n}$$
$$+ \Delta t \left[\theta R_{ij}^{n+1} + (1-\theta) R_{ij}^{n} \right],$$

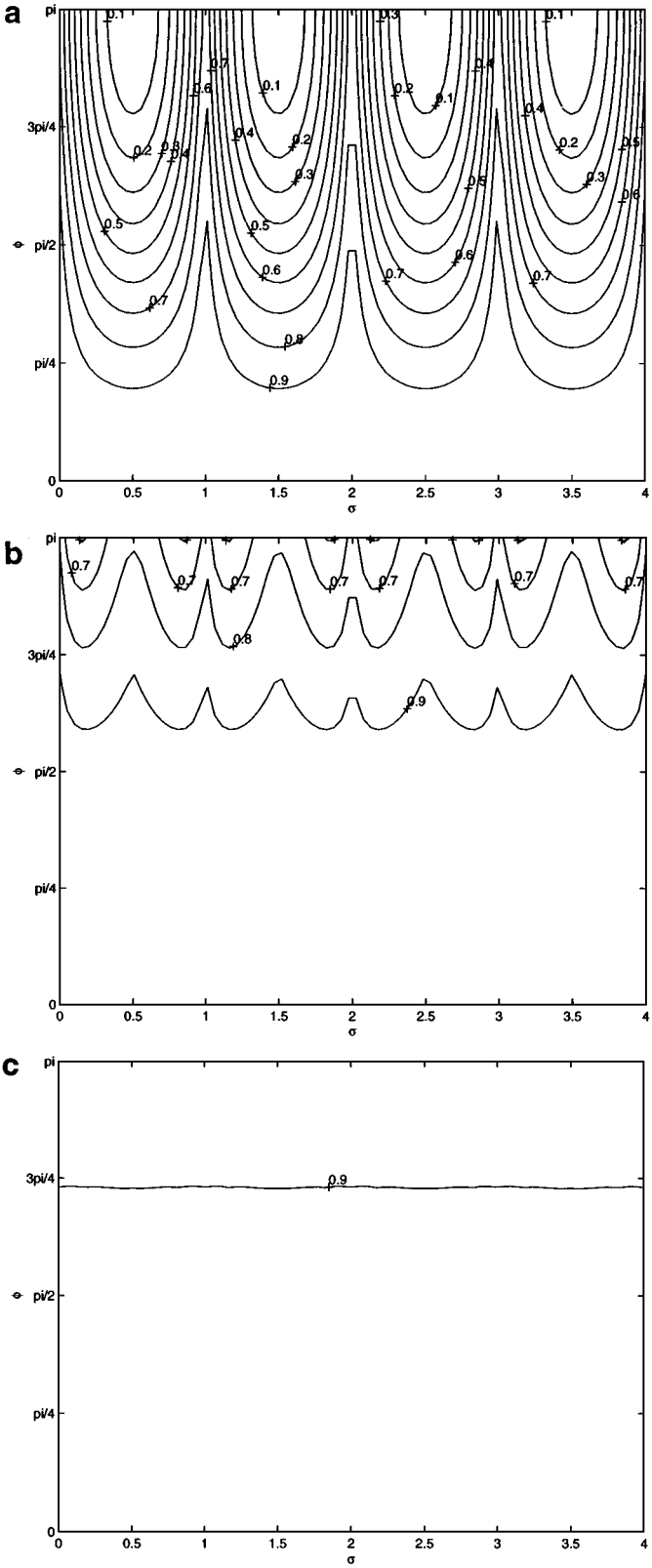

FIG. 7. The external node amplification factor for the 1D advection–diffusion equation for the Lagrange–Galerkin spectral element method using nonexact integration for $\mu = 0.01$ and (a) $N = 1$, (b) $N = 2$, and (c) $N \geq 4$.

where the matrix terms are

$$M_{ijkl} = \int_\Omega \psi_{ij}\psi_{kl}\,d\Omega = \int_{-1}^{+1}\int_{-1}^{+1} |J(\xi,\eta)| h_i(\xi)h_j(\eta)h_k(\xi)h_l(\eta)\,d\xi\,d\eta$$

$$\begin{aligned} A_{ijkl} &= \int_\Omega \psi_{ij}\psi_{mn}\mathbf{u}_{mn}\cdot\boldsymbol{\nabla}\psi_{kl}\,d\Omega \\ &= \int_{-1}^{+1}\int_{-1}^{+1} |J(\xi,\eta)| h_i(\xi)h_j(\eta)h_m(\xi)h_n(\eta)u_{mn}\frac{\partial h_k}{\partial\xi}(\xi)h_l(\eta)\frac{\partial\xi}{\partial x}\,d\xi\,d\eta \\ &\quad + \int_{-1}^{+1}\int_{-1}^{+1} |J(\xi,\eta)| h_i(\xi)h_j(\eta)h_m(\xi)h_n(\eta)v_{mn}h_k(\xi)\frac{\partial h_l}{\partial\eta}(\eta)\frac{\partial\eta}{\partial y}\,d\xi\,d\eta \end{aligned}$$

$$\begin{aligned} D_{ijkl} &= \nu\int_\Omega \boldsymbol{\nabla}\psi_{ij}\cdot\boldsymbol{\nabla}\psi_{kl}\,d\Omega \\ &= \nu\int_{-1}^{+1}\int_{-1}^{+1} |J(\xi,\eta)| \frac{\partial h_i}{\partial\xi}(\xi)h_j(\eta)\frac{\partial h_k}{\partial\xi}(\xi)h_l(\eta)\left(\frac{\partial\xi}{\partial x}\right)^2 d\xi\,d\eta \\ &\quad + \nu\int_{-1}^{+1}\int_{-1}^{+1} |J(\xi,\eta)| h_i(\xi)\frac{\partial h_j}{\partial\eta}(\eta)h_k(\xi)\frac{\partial h_l}{\partial\eta}(\eta)\left(\frac{\partial\eta}{\partial y}\right)^2 d\xi\,d\eta \end{aligned}$$

$$R_{ij} = \nu\int_\Gamma \mathbf{n}\cdot[\psi_i\boldsymbol{\nabla}\psi_j\varphi_j]\,d\Gamma = \nu\int_{-1}^{+1}\left[h_{ij}\frac{\partial h_{kl}}{\partial n}\varphi_{kl}\right]dn$$

for $i, j, k, l, m, n = 0, \ldots, N$, where $\mathbf{n}$ is the unit normal vector and J is the Jacobian of the transformation from physical to computational space. The coordinates within an element are approximated by the basis functions by

$$x = \sum_{i=0}^{N}\sum_{j=0}^{N} x_{ij}h_i(\xi)h_j(\eta)$$

and its derivative are given by

$$\frac{\partial x}{\partial\xi} = \sum_{i=0}^{N}\sum_{j=0}^{N} x_{ij}\frac{\partial h_i}{\partial\xi}(\xi)h_j(\eta),$$

where N represents the number of collocation points in the ξ and η directions. The remainder of the derivatives are obtained following the same procedure. Note that the extension to three dimensions is immediately obvious from the two-dimensional case.

To keep the algorithm as general and as automatic as possible, we evaluate the integrals numerically. Therefore, the mass matrix is evaluated as

$$M_{ijkl} = \sum_{q=0}^{Q}\sum_{r=0}^{Q} |J(\xi_q,\eta_r)| w_{qr}h_i(\xi_q)h_j(\eta_r)h_k(\xi_q)h_l(\eta_r),$$

where Q represents the number of Legendre–Gauss–Lobatto quadrature points in the ξ and η directions. In general, for exact integration of the matrices we require that $Q = (cN+1)/2$, where c is an integer constant denoting the factor of the maximum order matrix. In the case of the advection–diffusion equation, $c = 2$ which corresponds to the mass matrix. For the

case of the shallow water equations or the Navier–Stokes (and Euler) equations, $c = 3$ and corresponds to many terms, one of which is the advection operator. Note, however, that the 1D stability analysis showed that for $N \geq 4$ it makes no difference whether the exact or nonexact integration is used. Numerical results confirm this and for this reason the cheaper nonexact integration method is used throughout the paper.

4.2. Lagrange–Galerkin Spectral Element Method

The 2D advection–diffusion equation in Lagrangian form is written as

$$\frac{d\varphi}{dt} = \nu \nabla^2 \varphi$$

$$\frac{dx}{dt} = u, \quad \frac{dy}{dt} = v$$

which yields the matrix relation

$$[M_{ijkl} + \Delta t \theta D_{ijkl}]\varphi_{kl}^{n+1} = [M_{ijkl} - \Delta t(1-\theta)D_{ijkl}]\tilde{\varphi}_{kl}^{n} + \Delta t\left[\theta R_{ij}^{n+1} + (1-\theta)\tilde{R}_{ij}^{n}\right],$$

where the departure points are defined as $\mathbf{x}_d = \mathbf{x} - \boldsymbol{\alpha}$ and the trajectories are obtained from

$$\boldsymbol{\alpha} = \Delta t \mathbf{u}\left(\mathbf{x} - \frac{\boldsymbol{\alpha}}{2}, t + \frac{\Delta t}{2}\right) \tag{16}$$

and the matrices are defined as in the Eulerian case. Note that the trajectory relation only gives the departure point $\mathbf{x}_d$ and says nothing of where this point is located. Therefore, some means of searching the elements of the grid must be devised to determine which element contains the departure point in order to interpolate the variables (in this case φ and $\mathbf{u}$) using the element basis functions. For general grids, the best approach is to use a quadtree data structure. For the icosahedral grids used as one test case in this paper, the data structure described in [5] should be used. But once the element claiming the departure point is found, we still need to determine its coordinates in terms of the computational space; this is essential because all of the spectral element basis functions are written in terms of the computational coordinates. Equation (16) will only give the coordinates of the departure point in terms of the physical space. The departure point coordinates in physical space can be written using the basis functions in the form

$$\mathbf{x}_d = \sum_{i=0}^{N}\sum_{j=0}^{N} \mathbf{x}_{ij} h_i(\xi_d) h_j(\eta_d)$$

and by virtue of Newton's method, we write the iterative scheme for the roots (ξ_d, η_d) as

$$\mathbf{F}^{k+1} = \mathbf{F}^k + \boldsymbol{\nabla}\mathbf{F}^k\left(\xi_d^k, \eta_d^k\right) \cdot (d\xi, d\eta) = 0, \tag{17}$$

where

$$\mathbf{F} = \sum_{i=0}^{N}\sum_{j=0}^{N} \mathbf{x}_{ij} h_i(\xi_d) h_j(\eta_d) - \mathbf{x}_d.$$

This leads to the solutions

$$d\xi = \frac{\begin{bmatrix} -F_1^k & \frac{\partial F_1^k}{\partial \eta} \\ -F_2^k & \frac{\partial F_2^k}{\partial \eta} \end{bmatrix}}{\begin{bmatrix} \frac{\partial F_1^k}{\partial \xi} & \frac{\partial F_1^k}{\partial \eta} \\ \frac{\partial F_2^k}{\partial \xi} & \frac{\partial F_2^k}{\partial \eta} \end{bmatrix}}, \quad d\eta = \frac{\begin{bmatrix} \frac{\partial F_1^k}{\partial \xi} & -F_1^k \\ \frac{\partial F_2^k}{\partial \xi} & -F_2^k \end{bmatrix}}{\begin{bmatrix} \frac{\partial F_1^k}{\partial \xi} & \frac{\partial F_1^k}{\partial \eta} \\ \frac{\partial F_2^k}{\partial \xi} & \frac{\partial F_2^k}{\partial \eta} \end{bmatrix}},$$

where

$$\xi_d^{k+1} = \xi_d^k + d\xi, \quad \eta_d^{k+1} = \eta_d^k + d\eta$$

which only requires five iterations at most. Thus if $(\xi_d^{k+1}, \eta_d^{k+1}) \in [-1, 1]$, then the departure point is contained within this element. Clearly, as N increases, the cost of searching will increase by an order of N^2. So, instead of using the full polynomial of degree N, we use the vertices of the quadrilateral element (linear polynomial) to find the associated (ξ_d, η_d) for a given (x_d, y_d). Upon obtaining the departure point in terms of the computational space coordinates the interpolation is then obtained using the Nth-order basis functions of the element. The use of the linear basis functions to find (ξ_d, η_d) has absolutely no impact on the accuracy of the scheme, while costing far less than using the full Nth-order basis functions.

4.2.1. *Searching Algorithms*

Because the Lagrange–Galerkin method requires the calculation of departure points, the success of the method hinges on the rapidity of the searching algorithms. The object of the searching algorithm is to determine in which element a particular departure point lies.

QuadTree. For general grids, the best strategy is to find the closest node (grid point) to the departure point by virtue of a quadtree data structure. Let *quad_tree*$[1:ntree, 1:7]$ be an integer array which stores this quadtree. This array stores the following information:

- *quad_tree*$[i, 1-4]$ store the four children of this quad.
- *quad_tree*$[i, 5]$ stores the position of this quad with respect to its parent.
- *quad_tree*$[i, 6]$ stores the location of its parent.
- *quad_tree*$[i, 7]$ stores the number of nodes contained within this quad.

This defines a standard quadtree data structure; however, it is important to note that there is no need to use all of the nodes comprising the spectral element grid. In fact, only the vertices of each quadrilateral element (i.e., the four corner nodes) are required while the rest of the collocation points can be omitted. This saves a lot of effort in the searching process especially for high order grids (i.e. large N). Upon finding this nearest neighbor (closest node), we then search through the list of elements which claim this node and check for inclusion using the iterative approach defined in (17). There are usually no more than six elements claiming each node, even for distorted unstructured grids, which means that the iterative approach does not dominate the computational cost of the scheme. For highly distorted grids, however, the departure point may not necessarily lie within one of the

elements claiming the nearest neighbor. In this case, during the sweep through the element list claiming the nearest neighbor, the minimum distance between the departure point and the element nodes is stored. The element node yielding the minimum distance is considered to be the new nearest neighbor. If no inclusion is found, then the new nearest neighbor is used and the process is continued. Therefore, in the worst case scenario, only two nearest neighbor loops through the iterative approach are required. This can have adverse affects on the efficiency of the scheme if this case arises often; for the grids used in this paper, this situation did not present itself.

IcoTree. When a specific type of grid is being used, it is imperative to exploit any properties that it may have. For instance, the icosahedral grid used as one test case in the next section has an inherent tree data structure associated with it. In this case, the quadtree is abandoned for the ad-hoc data structure that can be constructed for this type of grid. The icosahedral grid on the plane is constructed by dividing a hexagon into six equilateral triangles. Then each triangle can be subdivided into four smaller equilateral triangles and so on. The quadrilateral spectral elements are then obtained by dividing each triangle into three quadrilaterals. Let $ico\_tree[1:ntree, 1:8]$ be an integer array which stores the icosahedral tree. This array stores the following information:

- $ico\_tree[i, 1-3]$ store the location of the three nodes which defines this triangle.
- $ico\_tree[i, 4]$ stores the location of its parent.
- $ico\_tree[i, 5-8]$ store the locations of its four children.

In addition, another array is required which stores the spectral elements contained within each of the triangles. Let $itree\_intma[1:ntree, 1:3]$ be this array. Clearly the icosahedral tree, like the quadtree, is also of order $\log_4 ntree$. The difference between the two data structures is that the icosahedral tree finds the triangle which contains the departure point, while the quadtree finds the nearest neighbor. Once this triangle is found, the array $itree\_intma[i, 1-3]$ is used to loop through the three spectral elements contained within this triangle. Thus the iterative approach given in (17) only requires sweeping through three elements which will always cost far less than the quadtree which may require looping through six elements.

5. NUMERICAL EXPERIMENTS

For the numerical experiments, the following terms are used in order to compare the performance of the schemes: the L_2 error norm,

$$\|e\|_{L_2} = \frac{\int_\Omega (\varphi_{exact} - \varphi)^2 \, d\Omega}{\int_\Omega \varphi_{exact}^2 \, d\Omega},$$

and the first and second moments of mass,

$$M_1 = \frac{\int_\Omega \varphi \, d\Omega}{\int_\Omega \varphi_{exact} \, d\Omega}, \quad M_2 = \frac{\int_\Omega \varphi^2 \, d\Omega}{\int_\Omega \varphi_{exact}^2 \, d\Omega}.$$

The L_2 error norm compares the root mean square percentage error of the numerical and exact solutions, while the first moment measures the percentage mass of the system, and the

second moment measures the amount of dissipation inherent within the numerical scheme. The ideal scheme should yield an error norm of zero and first and second moments of one.

5.1. One-Dimensional Advection–Diffusion Equation

For the spectral element method, the governing equation is given by (1) and the discretized equation by (2). For the Lagrange–Galerkin spectral element method, the governing equation is given by (3) and (4), while the discretized equation is given by (5).

The initial condition is given as

$$\varphi(x, 0) = e^{-(x-x_o)^2/2\lambda_o^2}$$

with periodic boundary conditions which yields the exact solution

$$\varphi(x, t) = \frac{\lambda_o}{\sqrt{\lambda_o^2 + 2\nu t}}\, e^{-(x-x_o-ut)^2/2\left(\lambda_o^2+2\nu t\right)},$$

where $\lambda_o = \frac{1}{8}$, $x_o = 0$, and $x \in [-1, 1]$ with the velocity $u = 2$. The results are given for one revolution ($t = 1.0$) of the initial wave.

Table I lists the pure advection ($\mu = 0$) results for the spectral element (SEM) and Lagrange–Galerkin spectral element methods (L-G SEM) for $\sigma \approx 0.25$. Because of the variable distance between any two nodes due to the Legendre–Gauss–Lobatto collocation points, the Courant numbers are never exactly the same when using varying orders of N (polynomial order) and N_E (number of elements) for a given number of N_P (total grid points). However, the Courant numbers used are equal at least to one decimal place. The results show that both methods give impressive results, but for $N \geq 4$ the Lagrange–Galerkin spectral element method is superior to the spectral element method. Table II shows the same results but for $\mu = 0.01$. These results are qualitatively similar to the $\mu = 0$ case. Table III shows the $\mu = 0$ results for various σ for the Lagrange–Galerkin spectral element method. From the stability analysis, the spectral element method is adversely affected by the increase in σ, but the Lagrange–Galerkin method is not. The numerical results also seem to indicate that the Lagrange–Galerkin method, in fact, improves with increased Courant numbers.

TABLE I
Results for the 1D Advection Equation for the Spectral Element and Lagrange–Galerkin Spectral Element Methods for $\sigma \approx 0.25$

Method	N	N_E	Δt	σ	$\|e\|_{L_2}$	M_1	M_2
SEM	1	10	0.025000	0.25	1.21×10^{-0}	1.0000	1.0000
	2	10	0.012500	0.25	2.84×10^{-1}	1.0000	1.0000
	4	10	0.004000	0.23	1.47×10^{-1}	1.0000	1.0000
	6	10	0.002000	0.24	1.87×10^{-3}	1.0000	1.0000
	8	10	0.001250	0.25	7.30×10^{-4}	1.0000	1.0000
	10	10	0.000800	0.24	2.99×10^{-4}	1.0000	1.0000
L-G SEM	1	10	0.025000	0.25	8.01×10^{-1}	1.0000	0.2316
	2	10	0.012500	0.25	4.13×10^{-1}	0.9996	0.7061
	4	10	0.004000	0.23	4.23×10^{-2}	1.0000	1.0179
	6	10	0.002000	0.24	1.06×10^{-3}	1.0000	0.9997
	8	10	0.001250	0.25	2.04×10^{-5}	1.0000	1.0000
	10	10	0.000800	0.24	3.32×10^{-7}	1.0000	1.0000

TABLE II

Results for the 1D Advection–Diffusion Equation for the Spectral Element and Lagrange–Galerkin Spectral Element Methods for $\sigma \approx 0.25$ and $\mu = 0.01$

Method	N	N_E	Δt	ν	$\|e\|_{L_2}$	M_1	M_2
SEM	1	10	0.025000	0.016000	8.89×10^{-1}	1.0009	1.1268
	2	10	0.012500	0.008000	1.13×10^{-1}	0.9997	0.9978
	4	10	0.004000	0.002980	5.16×10^{-3}	1.0000	1.0001
	6	10	0.002000	0.001500	1.44×10^{-3}	1.0000	1.0000
	8	10	0.001250	0.000800	6.30×10^{-4}	1.0000	1.0000
	10	10	0.000800	0.000545	2.74×10^{-4}	1.0000	1.0000
L-G SEM	1	10	0.025000	0.016000	6.48×10^{-1}	1.0009	0.4126
	2	10	0.012500	0.008000	2.63×10^{-1}	0.9993	0.8413
	4	10	0.004000	0.002980	1.96×10^{-2}	1.0000	1.0082
	6	10	0.002000	0.001500	5.67×10^{-4}	1.0000	0.9998
	8	10	0.001250	0.000800	1.28×10^{-5}	1.0000	1.0000
	10	10	0.000800	0.000545	2.20×10^{-7}	1.0000	1.0000

Remark 2. Table III shows that the Lagrange–Galerkin spectral element method improves as the Courant number is increased. This would be the case for ever increasing Courant numbers, if and only if the trajectories could be computed exactly. If this were possible, then the scheme could be run with infinitely large Courant numbers without any deterioration in accuracy. But since this is not the case, taking large enough Courant numbers will eventually lead to large errors in the trajectory calculation which will then severely diminish the accuracy of the scheme. In fact, the reason why we see an increase in accuracy for large Courant numbers is because fewer interpolations are being used.

TABLE III

Results for the 1D Advection Equation for the Lagrange–Galerkin Spectral Element Method for Various σ

Method	N	N_E	Δt	σ	$\|e\|_{L_2}$	M_1	M_2
L-G SEM	4	10	0.004000	0.23	4.23×10^{-02}	1.0000	1.0179
	4	10	0.008000	0.46	2.03×10^{-02}	1.0000	1.0077
	4	10	$0.0\overline{33333}$	1.93	1.75×10^{-03}	0.9999	1.0005
	6	10	0.002000	0.24	1.06×10^{-03}	1.0000	0.9997
	6	10	0.004000	0.47	4.82×10^{-04}	1.0000	1.0000
	6	10	0.015625	1.84	1.44×10^{-04}	1.0000	1.0000
	8	10	0.001250	0.25	2.04×10^{-05}	1.0000	1.0000
	8	10	0.002500	0.49	8.29×10^{-06}	1.0000	1.0000
	8	10	0.010000	1.99	2.41×10^{-06}	1.0000	1.0000
	10	10	0.000800	0.24	3.32×10^{-07}	1.0000	1.0000
	10	10	0.001600	0.48	1.40×10^{-07}	1.0000	1.0000
	10	10	0.006250	1.89	2.44×10^{-08}	1.0000	1.0000

5.2. Two-Dimensional Advection–Diffusion Equation

5.2.1. *Problem Statement*

The initial condition is given by

$$\varphi(x, y, 0) = e^{-\left[(x-x_o)^2+(y-y_o)^2\right]/2\lambda_o^2}$$

which yields the exact solution

$$\varphi(x, y, t) = \frac{\lambda_o^2}{\lambda_o^2 + 2\nu t}\, e^{-[\tilde{x}^2+\tilde{y}^2]/2\left(\lambda_o^2+2\nu t\right)}$$

having the far boundary conditions

$$\varphi(x, y, t) \to 0 \quad \text{for } (x, y) \to \infty,$$

where $\lambda_o = \frac{1}{8}$, $(x_o, y_o) = (-\frac{1}{2}, 0)$, $(x, y) \in [-1, 1]$ with the velocity field

$$u = +y, \quad v = -x$$

and

$$\tilde{x} = x - x_o \cos t - y_o \sin t, \quad \tilde{y} = y + x_o \sin t - y_o \cos t.$$

The following sections show the results after one revolution of the initial wave for the spectral element and Lagrange–Galerkin spectral element methods on various grids. The period for one revolution of the wave is 2π which means that one revolution corresponds to $t = 2\pi$ so that the actual time step taken for each experiment is defined as

$$\Delta t^{\text{actual}} = 2\pi\, \Delta t^{\text{reported}},$$

where $\Delta t^{\text{reported}}$ is the time step reported throughout the paper. The Courant number for all test cases is defined as

$$\sigma = \max\left(\frac{\Delta t U}{\Delta s}\right),$$

where $U = \sqrt{u_m^2 + v_m^2}$ and $\Delta s = \sqrt{\Delta x^2 + \Delta y^2}$ with $(\Delta x, \Delta y)$ being the distances between any two adjacent nodes and (u_m, v_m) being the velocities at the midpoint of the two adjacent nodes. The nondimensional diffusion coefficient is defined in a similar fashion as

$$\mu = \max\left(\frac{\Delta t \nu}{\Delta s^2}\right).$$

5.2.2. *Square Grid*

Tables IV and V list the results for the spectral element (SEM) and Lagrange–Galerkin spectral element methods (L-G SEM) for various values of N for $\mu = 0$ and $\mu = 0.01$ on a uniform square grid. In these tables N and N_E denote the polynomial order and the number of elements in each direction. In other words, for the case $N = 4$, $N_E = 10$, there are 10×10 total elements (N_E^T) and 41×41 total grid points (N_P) in the grid. The results obtained for the 2D cases are quantitatively similar to the 1D results, where the Lagrange–Galerkin

TABLE IV
Results for the 2D Advection Equation for the Spectral Element and Lagrange–Galerkin Spectral Element Methods on a Square Grid for $\sigma \approx 0.50$

Method	N	N_E	Δt	σ	$\|e\|_{L_2}$	M_1	M_2
SEM	1	10	0.012500	0.53	1.25×10^{-0}	0.9742	1.0000
	2	10	0.006250	0.54	3.98×10^{-1}	0.9854	1.0000
	4	10	0.002000	0.51	2.21×10^{-2}	1.0003	1.0000
	6	10	0.001000	0.52	2.04×10^{-3}	1.0000	1.0000
	8	10	0.000625	0.55	7.52×10^{-4}	1.0000	1.0000
	10	10	0.000400	0.54	3.12×10^{-4}	1.0000	1.0000
L-G SEM	1	10	0.012500	0.53	9.41×10^{-1}	0.4529	0.0211
	2	10	0.006250	0.54	6.02×10^{-1}	1.0119	0.4666
	4	10	0.002000	0.51	7.38×10^{-2}	1.0002	1.0448
	6	10	0.001000	0.52	1.72×10^{-3}	1.0000	0.9994
	8	10	0.000625	0.55	6.70×10^{-5}	1.0000	1.0000
	10	10	0.000400	0.54	5.13×10^{-5}	1.0000	1.0000

spectral element method now surpasses the spectral element method in accuracy for $N \geq 6$. Note that this behavior is experienced by the pure advection and the advection–diffusion problems. In fact, the diffusion assists both numerical algorithms by smoothening out the waves. Table VI shows the results for $\mu = 0$ for various values of σ. These results are once again in direct agreement with the 1D results and, hence, with the stability analysis. In the next sections, the grid and contours for various grids are illustrated for the pure advection problem only as this is the more difficult case.

5.2.3. *Icosahedral Grid*

The icosahedral grid used in this test case is the planar analog of the icosahedral triangular grid on the sphere presented in [5]. In the spherical case, the initial icosahedron contains

TABLE V
Results for the 2D Advection–Diffusion Equation for the Spectral Element and Lagrange–Galerkin Spectral Element Methods on a Square Grid for $\sigma \approx 0.50$ and $\mu = 0.01$

Method	N	N_E	Δt	ν	$\|e\|_{L_2}$	M_1	M_2
SEM	1	10	0.012500	0.005100	7.13×10^{-1}	0.9424	0.9751
	2	10	0.006250	0.002550	6.48×10^{-2}	0.9890	0.9922
	4	10	0.002000	0.000950	4.51×10^{-3}	0.9989	1.0000
	6	10	0.001000	0.000460	1.37×10^{-3}	0.9997	1.0000
	8	10	0.000625	0.000255	6.39×10^{-4}	0.9998	1.0000
	10	10	0.000400	0.000174	3.21×10^{-4}	1.0000	1.0000
L-G SEM	1	10	0.012500	0.005100	8.06×10^{-1}	0.4173	0.0771
	2	10	0.006250	0.002550	2.79×10^{-1}	0.9944	0.8010
	4	10	0.002000	0.000950	1.83×10^{-2}	0.9990	1.0099
	6	10	0.001000	0.000460	8.50×10^{-4}	0.9997	0.9998
	8	10	0.000625	0.000255	2.85×10^{-4}	0.9998	1.0000
	10	10	0.000400	0.000174	1.99×10^{-4}	1.0000	1.0000

TABLE VI
Results for the 2D Advection Equation for the Lagrange–Galerkin Spectral Element Method on a Square Grid for Various σ

Method	N	N_E	Δt	σ	$\|e\|_{L_2}$	M_1	M_2
L-G SEM	4	10	0.0010000	0.26	8.82×10^{-02}	1.0001	1.0539
	4	10	0.0020000	0.51	7.38×10^{-02}	1.0002	1.0448
	4	10	0.0080000	2.04	1.96×10^{-02}	1.0002	1.0059
	6	10	0.0005000	0.26	2.04×10^{-03}	1.0000	0.9993
	6	10	0.0010000	0.52	1.72×10^{-03}	1.0000	0.9994
	6	10	0.0040000	2.08	9.70×10^{-04}	1.0000	1.0000
	8	10	0.0003125	0.28	4.14×10^{-04}	1.0000	1.0000
	8	10	0.0006250	0.55	6.70×10^{-05}	1.0000	1.0000
	8	10	0.0025000	2.21	3.71×10^{-05}	1.0000	1.0000
	10	10	0.0002000	0.27	2.05×10^{-04}	1.0000	1.0000
	10	10	0.0004000	0.54	5.13×10^{-05}	1.0000	1.0000
	10	10	0.0016000	2.15	1.60×10^{-05}	1.0000	1.0000

12 grid points and 20 equilateral triangles; whereas in the planar case, the initial grid is a hexagon containing 7 grid points and 6 equilateral triangles. Upon generating the refined grid as is described in [5] the resulting triangles are then subdivided to form quadrilaterals in order for the spectral element discretization to be used. In order to split the triangles into quadrilaterals, we find the midpoints of each of the triangle edges and form quadrilaterals by connecting these midpoints to the centroid of the triangle.

Figure 8 shows the result for the Lagrange–Galerkin spectral element method with $N = 8$, $N_E^T = 72$, and $N_P = 4705$ with a Courant number of $\sigma = 2.43$. The error norm for this problem is 5.50×10^{-3}. The uniform square grid which has the closest number of grid points (N_P) is the case $N = 6$, $N_E = 10$ which has $N_P = 3721$ grid points. Table IV gives an error norm of 1.72×10^{-3} for this case for $\sigma = 0.52$. Thus the icosahedral grid has given slightly less accurate results than the uniform square grid, but, considering the unstructured nature of the grid and the Courant number being used, this result is quite good. In addition,

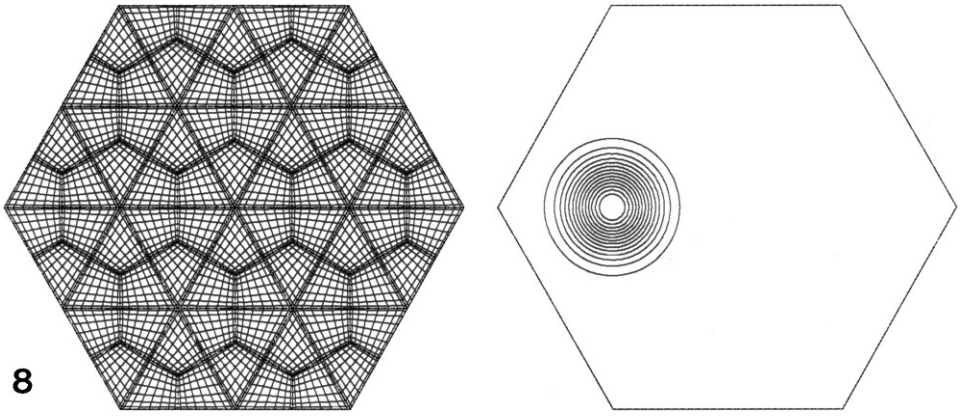

FIG. 8. The grid and contours for the Lagrange–Galerkin spectral element method on an icosahedral grid with $N = 8$, $N_E^T = 72$, $N_P = 4705$, and $\sigma = 2.43$.

although results for other orders of N are not shown (for brevity), the method shows spectral convergence even for this type of grid.

This icosahedral grid has a very efficient data structure associated with it that can be used for searching. The other beneficial property of this grid is that all the quadrilateral elements are exactly the same size because the parent triangles are equilaterals. This is an important property if a uniform representation throughout the domain is desired, which is usually the case for geophysical flows on the surface of the sphere where grid biasing is typically undesirable. However, if totally unstructured grids are desired, as is the case with adaptive grids in computational fluid dynamics, will this strategy work? The following section addresses this case.

5.2.4. *Unstructured Grid*

Figure 9 shows the grid and contours for the Lagrange–Galerkin spectral element method using an unstructured grid with $N = 8$, $N_E^T = 96$, and $N_P = 6273$. The error norm for this example is 3.08×10^{-4} for $\sigma = 2.50$. The unstructured grid generator is described in [4]. However, this grid generator creates only triangles and so the triangles have to be subdivided into quadrilaterals. The uniform square grid case $N = 8$, $N_E = 10$ having $N_P = 6561$ corresponds the closest to this unstructured grid case. The uniform square grid case yields an error norm of 6.70×10^{-5} for $\sigma = 0.55$. Once again, better results are obtained for the uniform square grid but, considering the lack of structure of the current grid and the large σ, this result is quite impressive. In fact, the results on the unstructured grid also show spectral convergence as N is increased.

The results for this example show that the Lagrange–Galerkin spectral element method does indeed work, even on such an irregular grid as the one presented here. In addition, the scheme yields an extremely accurate result, even when large Courant numbers are used. This example truly shows the power and flexibility of the proposed strategy of combining the Lagrange–Galerkin method with the spectral element method.

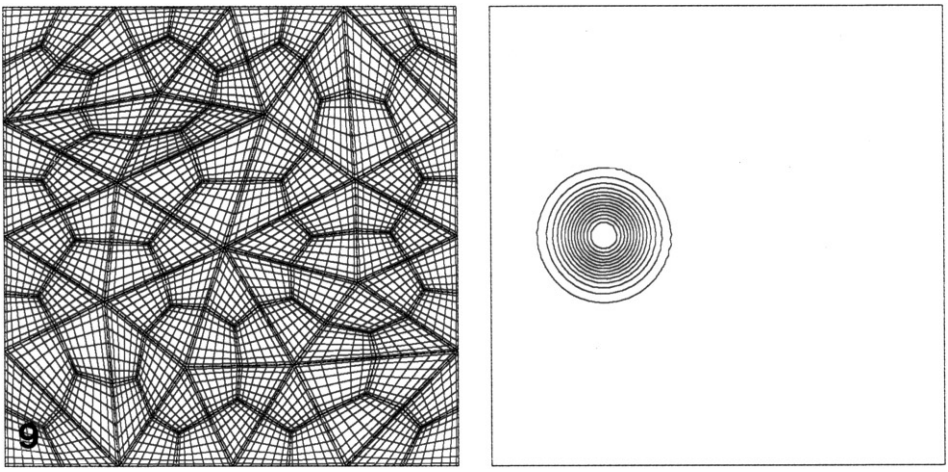

FIG. 9. The grid and contours for the Lagrange–Galerkin spectral element method on an unstructured grid with $N = 8$, $N_E^T = 96$, $N_P = 6273$, and $\sigma = 2.50$.

5.2.5. *Deformational Flow*

In this section, the Smolarkiewz deformational flow problem is used to test the stability of the numerical scheme. The initial function is assumed to be a cone given by

$$\varphi(x, y, 0) = h\left(1 - \frac{r}{R}\right),$$

where $h = 1, r = \sqrt{(x - x_o)^2 + (y - y_o)^2}$, $R = 0.15$, $(x_o, y_o) = (\frac{1}{2}, \frac{1}{2})$, $(x, y) \in [0, 1]$ with the velocity field

$$u = 0.32\pi \sin 4\pi x \sin 4\pi y, \quad v = 0.32\pi \cos 4\pi x \cos 4\pi y.$$

This velocity field defines a set of 16 symmetrical vortices. The fluid particles remain for all time within the vortices in which they resided initially. Therefore, because each fluid particle is trapped within its vortex it is constrained to move along streamlines and for this reason the cone deforms. The analytic solution to this problem is presented in [14]. Figure 4a in [14] corresponds to the solution at $t = 0.65625$ for our grid dimensions.

Any scheme which is not monotonic will not resolve this test case well because any small amount of dispersion is exacerbated by the vortical flow. In addition, this is not the best test case for the spectral element method because high order methods are only guaranteed to yield exponential convergence when the function it is meant to resolve is smooth. Clearly this is not the case for this test problem. Nonetheless, it makes for an interesting problem for showing the strengths and weaknesses of the Lagrange–Galerkin spectral element method.

The Lagrange–Galerkin spectral element was run on a uniform structured grid using a total of 81×81 grid points using various combinations of N and N_E. The low order schemes perform far better than the high order schemes due to their monotonicity property. In other words, the first-order scheme ($N = 1$) captures the essence of the analytic solution quite well, but it is far too diffusive to be of any use. Neither the Lagrange–Galerkin method nor the spectral element method are naturally monotonic. Therefore, in order to resolve this test case accurately some monotonicity-preserving mechanism must be introduced into the numerical scheme. The simplest approach is to use the flux-corrected transport (FCT) scheme. This is not a new idea and, in fact, has been used previously to suppress spurious oscillations near large gradients (such as shock waves) in high order spectral element methods [3]. In brief, the FCT scheme uses a limiting procedure between a high order and a low order scheme to get the highest possible order without introducing a new local extremum. Therefore, we write the interpolated value at the departure point $\tilde{\varphi}$ as the linear function

$$\tilde{\varphi} = \tilde{\varphi}^L + \gamma(\tilde{\varphi}^H - \tilde{\varphi}^L),$$

where the superscripts H and L denote the high and low order interpolations and $\gamma \in [0, 1]$ represents the value of the limiting (or weighting). The goal is to obtain the maximum value of γ which gives the highest order interpolation without producing a new maximum or

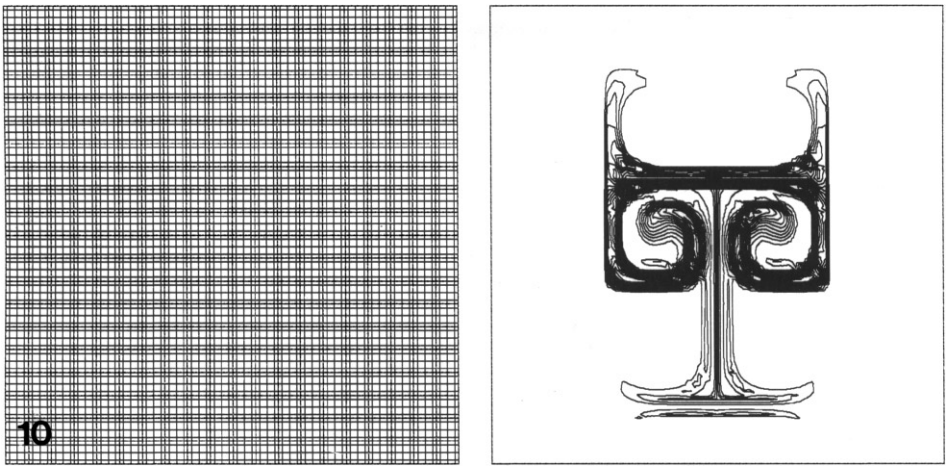

FIG. 10. The grid and contours for the Lagrange–Galerkin spectral element method for the deformational flow problem with $N = 4$, $N_E = 20$, and $\sigma = 1.02$.

minimum value within the element. The computation of γ can be written, as in [1], as

$$Q^+ = \varphi^+ - \tilde{\varphi}^L, \quad Q^- = \varphi^- - \tilde{\varphi}^L, \quad P = \tilde{\varphi}^H - \tilde{\varphi}^L$$

$$\gamma = \begin{cases} \min\left(1, \frac{Q^-}{P}\right), & \text{if } P < 0, \\ 0, & \text{if } P = 0, \\ \min\left(1, \frac{Q^+}{P}\right), & \text{if } P > 0, \end{cases}$$

where φ^+ and φ^- are the maximum and minimum values at the collocation points of the element which claims the departure point $\mathbf{x}_d$. The high order interpolation $\tilde{\varphi}^H$ is obtained by using the full polynomial of order N while the low order interpolation $\tilde{\varphi}^L$ is obtained by using only the four vertices of the quadrilateral element, thereby yielding a first-order interpolation which, by definition, is monotonic.

The Lagrange–Galerkin spectral element method with FCT was run on the grids

$$(N, N_E) = [(1, 80), (2, 40), (4, 20), (8, 10)]$$

with $\Delta t = 0.00875$ up to $t = 0.65625$. Once again, the low order schemes resolve the shape of the analytic solution much better, but they are quite diffusive. The higher order scheme $N = 8$ also resolves the shape of the solution, but the contours are much too densely packed and not nearly as finely defined as in the low order schemes. The scheme $N = 4$ results in the overall best scheme for this problem. It resolves the shape of the analytic solution quite well and does not suffer too much diffusion. Figure 10 shows the grid and contours for this case. Note that the time step results in a Courant number of $\sigma = 1.02$.

5.3. Computational Cost

Table VII summarizes the computational cost of using the spectral element both implicitly (trapezoidal rule) and explicitly (Adams–Bashforth), and the Lagrange–Galerkin

TABLE VII
Percentage Breakdown of the Computational Costs Incurred by the Spectral Element Method and the Lagrange–Galerkin Spectral Element Method

Method	N	Matrix inversion	Searching algorithms	Element operations	Total CPU (seconds)
SEM	4	0.0	0.0	99.7	271
Adams–Bashforth	6	0.0	0.0	99.6	868
(3rd order)	8	0.0	0.0	98.9	1589
	10	0.0	0.0	98.2	2776
SEM	4	52.3	0.0	47.6	304
Trapezoidal rule	6	42.2	0.0	57.6	2133
(2nd order)	8	72.1	0.0	27.6	2914
	10	55.8	0.0	43.8	4314
L-G SEM	4	45.2	9.7	44.8	344
Trapezoidal rule	6	38.2	12.3	49.4	2361
(2nd order)	8	46.1	19.7	33.1	4100
	10	46.3	25.7	27.6	5723

Note. These results are for the 2D advection–diffusion equation with $\sigma = 0.5$ (0.25 for SEM, Adams–Bashforth) and $\mu = 0.01$.

spectral element method. The results are illustrated for the 2D advection–diffusion equation. This case is chosen because the Lagrange–Galerkin method must solve a system of matrix equations, whereas for the advection equation it does not. The computational cost is divided into three categories: *matrix inversion*, *searching algorithms*, and *element operations*. *Matrix inversion* denotes the percentage of total time to invert the global matrix, *searching algorithms* is the percentage involving any operations required for the searching of departure points, and *element operations* is the percentage involving any operation typically associated with the spectral element method. The explicit spectral element method has to be run at a lower Courant number due to its limited stability. Nonetheless, it is by far the most efficient. The error norms between the implicit and explicit spectral element methods are virtually identical for this case. The interesting thing to note is that while the Lagrange–Galerkin method is more expensive than the implicit spectral element method, it is not prohibitively expensive. In fact, the cost involving the searching algorithms increases with increasing N but at a constant rate. Most of the cost, in fact, is in the matrix inversion. In other words, in order to optimize this method, the focus should be on the matrix inversions or in the element operations and not in the searching algorithms. Note that the resulting matrix for the Lagrange–Galerkin spectral element method is symmetric positive definite which lends itself to efficient iterative solvers such as conjugate gradient methods. Finally, the Lagrange–Galerkin spectral element method can be doubled in efficiency merely by using a Courant number twice as large. Table VI shows that the Lagrange–Galerkin spectral element method can be run using large Courant numbers up to 2 or greater without diminishing in accuracy.

6. CONCLUSIONS

A new method, the Lagrange–Galerkin spectral element method, is introduced whereby the Lagrange–Galerkin method and the spectral element method are combined. The main

attraction of this method is that it uses the basis functions of the spectral element method as the interpolating polynomial for the Lagrange–Galerkin calculation of the departure points and the corresponding interpolations of the variables at these points. This makes the method quite local in that the interpolation and calculations are all performed in an element per element basis. This property is very important, particularly if we have an interest in implementing the method with unstructured/adaptive grids and on distributed memory computers; this topic, however, is left for future work.

The stability analysis shows that while the spectral element method is stable for large σ, this range is not recommended as the accuracy decreases significantly. The analysis shows that when the Lagrange–Galerkin method is fused with the spectral element method, this decrease in accuracy no longer occurs, thereby allowing larger time steps to be used which in turn increases the efficiency.

The 2D results on the various grids show that the Lagrange–Galerkin spectral element method yields extremely accurate solutions, even while using large Cournat numbers and on different types of grids. The icosahedral grid results are especially encouraging, particularly because this grid has proven to be quite promising for applications on the sphere [5]. The results on the unstructured grid show that the Lagrange–Galerkin spectral element method yields very good results, even on distorted grids and while using large Courant numbers. However, because the Lagrange–Galerkin method relies heavily on interpolation, the spectral element basis functions must be of a sufficiently high order to obtain accurate values at the departure points. The analysis shows that N must be greater than four, in order to reap the full benefits of this new hybrid scheme. This is not the order which is recommended, but rather, it is the minimum order that ought to be used. Increasing the order of N beyond this minimum value increases the accuracy of the scheme but also incurs additional computational costs. The costs are not prohibitive, but they are nonetheless expensive. However, the dominant parts of the algorithm are the same operations which plague any implicit method, namely, the inversion of the global matrix. Efforts to optimize the method are currently underway. In addition, explicit time discretization methods such as those proposed in [6, 8] may have to be explored, thereby eliminating the need to invert a global matrix.

ACKNOWLEDGMENTS

I am grateful to my sponsor, the Office of Naval Research, for supporting this work through program PE-0602435N. Special thanks go to Mohamed Iskandarani of Rutgers University (New Brunswick, NJ) and Spencer Sherwin of the Imperial College (London, England) for discussing with me the details of the implementation of the spectral element method. I would also like to thank the two anonymous reviewers for their valuable comments and suggestions which led me to understand and then to clarify various key points of this work.

REFERENCES

1. R. Bermejo and A. Staniforth, The conversion of semi-Lagrangian advection schemes to quasi-monotone schemes, *Mon. Weather Rev.* **120**, 2622 (1992).
2. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics* (Springer-Verlag, New York, 1998).
3. J. G. Giannakouros, D. Sidilkover, and G. E. Karniadakis, Spectral element-FCT method for the one- and two-dimensional compressible Euler equations, *Comput. Methods Appl. Mech. Engrg.* **116**, 113 (1994).

4. F. X. Giraldo, Efficiency and accuracy of Lagrange–Galerkin methods on unstructured adaptive grids, *Math. Modeling Sci. Comput.* **8** (1997).
5. F. X. Giraldo, Lagrange–Galerkin methods on spherical geodesic grids, *J. Comput. Phys.* **136**, 197 (1997).
6. F. X. Giraldo, A spectral element semi-Lagrangian method for the shallow water equations, in *Proceedings, Fourth World Congress on Computational Mechanics, 1998.*
7. Y. Hasbani, E. Livne, and M. Bercouvier, Finite elements and characteristics applied to advection–diffusion equations, *Comput. Fluids* **11**, 71 (1983).
8. L. Ho, Y. Maday, A. Patera, and E. Ronquist, A high-order decoupling method for the incompressible Navier-Stokes equations, *Comput. Methods. Appl. Mech. Engrg.* **80**, 65 (1990).
9. M. Iskandarani, D. B. Haidvogel, and J. P. Boyd, Staggered spectral element model with application to the oceanic shallow water equations, *Int. J. Numer. Methods Fluids* **20**, 393 (1995).
10. H. Ma, A spectral element basin model for the shallow water equations, *J. Comput. Phys.* **109**, 133 (1993).
11. A. McDonald, Accuracy of multiply-upstream, semi-Lagrangian advective scheme, *Mon. Weather Rev.* **112**, 1267 (1984).
12. A. T. Patera, A spectral element method for fluid dynamics: laminar flow in a channel expansion, *J. Comput. Phys.* **54**, 468 (1984).
13. M. R. Schumack, W. W. Schultz, and J. P. Boyd, Spectral method solution of the Stokes equations on non-staggered grids, *J. Comput. Phys.* **94**, 30 (1991).
14. A. Staniforth and J. Côté, Comments on "Smolarkiewicz's Deformational Flow," *Mon. Weather Rev.* **115**, 894 (1987).
15. E. Suli and A. Ware, A spectral method of characteristics for hyperbolic problems, *SIAM J. Numer. Anal.* **28**, 423 (1991).