Calhoun: The NPS Institutional Archive

Faculty and Researcher Publications Faculty and Researcher Publications

2003

Strong and weak Lagrange-Galerkin spectral element methods for the shallow water equations

Giraldo, F. X.

Calhoun is a project of the Dudley Knox Library at NPS, furthering the precepts and goals of open government and government transparency. All information contained herein has been approved for release by the NPS Public Affairs Officer.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

An International Journal
computers & mathematics with applications

PERGAMON Computers and Mathematics with Applications 45 (2003) 97–121
www.elsevier.com/locate/camwa

Strong and Weak Lagrange-Galerkin Spectral Element Methods for the Shallow Water Equations

F. X. Giraldo
Naval Research Laboratory
Monterey, CA 93943, U.S.A.
giraldo@nrlmry.navy.mil

Abstract—The Lagrange-Galerkin spectral element method for the two-dimensional shallow water equations is presented. The equations are written in conservation form and the domains are discretized using quadrilateral elements.

Lagrangian methods integrate the governing equations along the characteristic curves, thus being well suited for resolving the nonlinearities introduced by the advection operator of the fluid dynamics equations.

Two types of Lagrange-Galerkin methods are presented: the strong and weak formulations. The strong form relies mainly on interpolation to achieve high accuracy while the weak form relies primarily on integration. Lagrange-Galerkin schemes offer an increased efficiency by virtue of their less stringent CFL condition. The use of quadrilateral elements permits the construction of spectral-type finite-element methods that exhibit exponential convergence as in the conventional spectral method, yet they are constructed locally as in the finite-element method; this is the spectral element method.

In this paper, we show how to fuse the Lagrange-Galerkin methods with the spectral element method and present results for two standard test cases in order to compare and contrast these two hybrid schemes. © 2003 Published by Elsevier Science Ltd.

Keywords—Finite element, Lagrange-Galerkin, Shallow water equations, Semi-Lagrangian, Spectral element.

1. INTRODUCTION

The advection terms in the governing equations of fluid motion present formidable challenges to many spatial discretization methods, including Galerkin methods. These terms prevent the operator from being self-adjoint and as a result, Galerkin methods are no longer optimal for the spatial discretization. Researchers have tried circumventing this problem by using high-order Eulerian methods and characteristic-based methods. In this section, a brief overview of some of the more interesting methods is given.

A limited number of new Eulerian methods has been introduced in recent years for solving hyperbolic partial differential equations, perhaps the best being the spectral element method [1,2].

The support of my sponsor, the Office of Naval Research, through Program PE-0602435N is gratefully acknowledged.

0898-1221/03/$ - see front matter
© 2003 Published by Elsevier Science Ltd. Typeset by $\mathcal{AMS}$-TEX
PII: S0898-1221(02)00330-9

The spectral element method combines the benefits extracted from both the spectral method and the finite-element method. It combines the high-order accuracy of the spectral method with the geometric flexibility of the finite-element method. This method offers high accuracy solutions and spectral convergence (provided the solution is smooth) but at the price of having to use small time steps (due to the explicit solvers typically used) and structured grids (due to the restriction that the elements be quadrilaterals). Recently, Giraldo [3] showed how to increase the time step by combining the strong Lagrange-Galerkin method with the spectral element method.

In contrast, there exists a multitude of characteristic-based methods. These methods combine standard spatial discretization methods (such as the finite-element method) with the method of characteristics. By integrating the equations in time along the characteristics, the resulting spatial operator becomes self-adjoint which then justifies the use of Galerkin spatial discretizations. Below, we discuss the characteristic-based methods that are explored in this paper and those that are related to the methods that we shall use.

In the *strong* Lagrange-Galerkin method the time derivative and the advection terms are combined into the Lagrangian derivative and the resulting operator is then discretized using the standard finite-element method. This is the approach used by Bercovier and Pironneau [4], Bermejo [5], Douglas and Russell [6], and Priestley [7], for example. In this method, the basis functions are the typical Lagrange polynomials used in the standard finite-element method which are only dependent on the spatial coordinates. The success of this method hinges on determining (interpolating) the values at the feet of the characteristics.

Related to this method are the semi-Lagrangian [8], characteristic-Galerkin [9], and Taylor-Galerkin methods [10]. The semi-Lagrangian method is essentially the strong Lagrange-Galerkin method with the exception that the spatial discretization is achieved through finite differencing; this method is quite ubiquitous in the meteorology community.

The characteristic-Galerkin method avoids interpolations by using Taylor series expansions to approximate the values at the feet of the characteristics. Therefore, the computations all take place in terms of the Eulerian grid which eliminates the difficulties of the Lagrangian grid but at the expense of having to use smaller time steps due to the stricter stability restrictions which govern all Eulerian methods.

Taylor-Galerkin methods are essentially Lax-Wendroff schemes designed for finite element spatial discretizations and are quite similar to characteristic-Galerkin methods. In fact, they can be shown to be equivalent for scalar equations [11].

The *weak* Lagrange-Galerkin method, on the other hand, uses basis functions which are dependent on both space and time by forcing the basis functions to vanish along the adjoint of the transport operator (the transport operator referring to a homogeneous conservation law); this results in the basis functions vanishing along the characteristics. Using the conservation form of the governing equations simplifies the resulting discretization by introducing the Reynolds' transport theorem [12] which then eliminates the boundary terms arising from the integration by parts used to obtain the adjoint of the transport operator. This is the method introduced by Benque *et al.* [13,14].

Lagrange-Galerkin methods have increased in popularity in the last ten years because they offer increased accuracy and efficiency by virtue of their independence on the CFL condition; however, almost all research has involved the *strong* method as opposed to the *weak* method. Currently, the only applications (of which this author is aware of) using the *weak* method or variations thereof are the following: advection [13,15], shallow water on the plane [15,16], Navier-Stokes [14], and shallow water on the sphere [17,18].

In this paper, we show how to combine the Lagrange-Galerkin methods with the spectral element method. This paper builds on our previous paper [3] in which we showed theoretically how to combine the strong Lagrange-Galerkin and the spectral element methods. In this paper, we combine both the *strong* and *weak* Lagrange-Galerkin methods with the spectral element method and apply these schemes to the two-dimensional shallow water equations.

2. SHALLOW WATER EQUATIONS

The two-dimensional shallow water equations in conservation form are

$$\frac{\partial}{\partial t}\begin{bmatrix}\varphi\\ \varphi u\\ \varphi v\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}\varphi u\\ \varphi u^2+\frac{1}{2}\varphi^2\\ \varphi uv\end{bmatrix}+\frac{\partial}{\partial y}\begin{bmatrix}\varphi v\\ \varphi uv\\ \varphi v^2+\frac{1}{2}\varphi^2\end{bmatrix}=\begin{bmatrix}0\\ +f\varphi v\\ -f\varphi u\end{bmatrix},$$

but note that if we move the pressure terms to the right-hand side, we get

$$\frac{\partial}{\partial t}\begin{bmatrix}\varphi\\ \varphi u\\ \varphi v\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}\varphi u\\ \varphi u^2\\ \varphi uv\end{bmatrix}+\frac{\partial}{\partial y}\begin{bmatrix}\varphi v\\ \varphi uv\\ \varphi v^2\end{bmatrix}=\begin{bmatrix}0\\ -\varphi\frac{\partial\varphi}{\partial x}+f\varphi v\\ -\varphi\frac{\partial\varphi}{\partial y}-f\varphi u\end{bmatrix}. \tag{1}$$

This system can now be written more compactly as

$$\frac{\partial\boldsymbol{\varphi}}{\partial t}+\boldsymbol{\nabla}\cdot(\boldsymbol{\varphi}\mathbf{u})=\mathbf{S}(\boldsymbol{\varphi}), \tag{2}$$

where

$$\boldsymbol{\varphi}=\begin{bmatrix}\varphi\\ \varphi u\\ \varphi v\end{bmatrix},\qquad \mathbf{u}=\begin{bmatrix}u\\ v\end{bmatrix},\qquad \mathbf{S}(\boldsymbol{\varphi})=\begin{bmatrix}0\\ -\varphi\frac{\partial\varphi}{\partial x}+f\varphi v\\ -\varphi\frac{\partial\varphi}{\partial y}-f\varphi u\end{bmatrix}. \tag{3}$$

The equations are solved for the three conservation variables φ, φu, and φv. Before proceeding to the discretization of the weak Lagrange-Galerkin method, let us first look at the discretization obtained by an Eulerian method. This will serve both for contrast and also because we require one Eulerian time step before using the Lagrange-Galerkin methods.

3. SPECTRAL ELEMENT METHOD

3.1. Basis Functions

The conservation variables belong to the following spaces:

$$\varphi\in H^1(\Omega),\qquad (\varphi u,\varphi v)\in H_0^1(\Omega),$$

and their basis functions, likewise, are defined as

$$\psi_i^{\varphi}\in H^1(\Omega),\qquad \psi_i^{\varphi u,\varphi v}\in H_0^1(\Omega);$$

in other words, they belong to the set of square integrable functions whose first derivatives are also square integrable. The Hilbert space $H_0^1(\Omega)$ is defined as

$$H_0^1(\Omega)=\left\{\psi\in H^1(\Omega)\mid\psi(\Gamma)=0\right\},$$

where Γ denotes the boundary of the domain Ω. Because we are only using quadrilateral elements, we can construct the two-dimensional basis functions as a tensor product of the one-dimensional Legendre cardinal functions such as

$$\psi_i(\xi,\eta)=h_j(\xi)\,h_k(\eta), \tag{4}$$

where $i=1,\ldots,P^2$ and $j,k=1,\ldots,P$. The integer P denotes the number of Legendre-Gauss-Lobatto (LGL) points in each direction (ξ and η), and is equal to $P=p+1$, where p denotes the polynomial order of the Legendre cardinal functions.

The one-dimensional Legendre cardinal functions are defined as

$$h_i(\xi) = -\frac{(1-\xi^2)\,L_p'(\xi)}{p(p+1)L_p(\xi_i)(\xi-\xi_i)}, \tag{5}$$

where L_p is the p^{th}-order Legendre polynomial and L_p' is its derivative, and the mapping from physical space to computational space is achieved by the transformation

$$\xi = \frac{2}{x_2 - x_1}(x - x_1) - 1, \qquad x \in [x_1, x_2], \tag{6}$$

where x_1 and x_2 are the physical coordinates defining the spectral element. But, in order to keep the algorithms as general as possible, it is best to construct the one-dimensional basis functions and their derivatives in the following form:

$$h_i(\xi) = \prod_{\substack{j=1\\ j\neq i}}^{P} \left(\frac{\xi - \xi_j}{\xi_i - \xi_j}\right), \tag{7}$$

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

where the ξ_i, ξ_j, ξ_k are the permutations of the collocation points. The reason why it is best to write the cardinal functions in this form is that we may not always write the functions as functions of the Gauss-Lobatto points. In Lagrange-Galerkin methods, we need to compute the values of the unknown variables at the feet of the characteristics and since we shall use the basis functions for these interpolations, it is best to define these cardinal functions in a general form.

The recurrence relation for Legendre polynomials is given by

$$L_p(\xi) = \frac{2p-1}{p}\,\xi L_{p-1}(\xi) - \frac{p-1}{p}\,L_{p-2}(\xi),$$

where $L_0(\xi) = 1$ and $L_1(\xi) = \xi$. The derivatives are given by

$$L_p'(\xi) = \frac{2p-1}{p}\left[L_{p-1}(\xi) + \xi L_{p-1}'(\xi)\right] - \frac{p-1}{p}\,L_{p-2}'(\xi)$$

and

$$L_p''(\xi) = \frac{2p-1}{p}\left[2L_{p-1}'(\xi) + \xi L_{p-1}''(\xi)\right] - \frac{p-1}{p}\,L_{p-2}''(\xi),$$

where $L_0'(\xi) = L_0''(\xi) = L_1''(\xi) = 0$ and $L_1'(\xi) = 1$. In order for the basis functions to be cardinal functions, the collocation points must satisfy the relation

$$(1-\xi^2)\,L_p'(\xi) = 0, \tag{9}$$

which means that they will be the roots of the numerator in (5); these points are the Gauss-Lobatto points. The roots of (9) can be obtained by Newton's method. Taking a Taylor series expansion with respect to ξ and rearranging, we get the following iterative relation:

$$\xi_j^{m+1} = \xi_j^m - \frac{(1-\xi^2)\,L_p'(\xi)}{-2\xi L_p'(\xi) + (1-\xi^2)\,L_p''(\xi)}, \qquad j = 0, \dots, p,$$

which has the following corresponding Gauss-Lobatto quadrature weights:

$$w_j = \frac{2}{p(p+1)[L_p(\xi_j)]^2}. \tag{10}$$

By using equations (5)–(10), we can automatically generate any order set of Legendre cardinal basis functions along with their associated collocation points and Gauss-Lobatto integration weights. The coordinates within an element are approximated by the basis functions as

$$x = \sum_{i=1}^{P^2} x_i \psi_i(\xi, \eta), \qquad \text{where} \quad \sum_{i=1}^{P^2} \psi_i(\xi, \eta) = \sum_{j=1}^{P} \sum_{k=1}^{P} h_j(\xi) h_k(\eta),$$

and so its derivatives can be approximated by

$$\frac{\partial x}{\partial \xi} = \sum_{i=1}^{P^2} x_i \frac{\partial \psi_i}{\partial \xi}(\xi, \eta) \qquad \text{and} \qquad \frac{\partial x}{\partial \eta} = \sum_{i=1}^{P^2} x_i \frac{\partial \psi_i}{\partial \eta}(\xi, \eta),$$

where $P = p + 1$ represents the number of grid points in the ξ and η directions. The remainder of the derivatives are obtained following the same procedure.

3.2. Integrals

To keep the algorithm as general and as automatic as possible, we evaluate the integrals numerically. Therefore, as you will see, the mass matrix can be evaluated as

$$M_{ij} = \sum_{q=1}^{Q^2} |J(\xi_q, \eta_q)| \, w_q \, \psi_i(\xi_q, \eta_q) \psi_j(\xi_q, \eta_q),$$

where Q represents the number of Legendre-Gauss-Lobatto quadrature points in the ξ and η directions and J is the Jacobian of the mapping from physical to computational space and is given by

$$|J| = \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi}.$$

You will note that the mass matrix only contains polynomials of the order $2p$ in each direction. However, this is not the maximum polynomial order contained in the equations. The Coriolis matrix F_{ij} contains polynomials of the order $3p$ and so we require at least $Q = 3(p + 1)/2$ quadrature points. These quadrature rules guarantee the exact integration of all of the matrices in the equations.

4. EULER-GALERKIN METHOD

Beginning with (2), we can define an Eulerian finite-element method by taking the weak form

$$\int_{\Omega} \psi \left[\frac{\partial \varphi}{\partial t} + \nabla \cdot (\varphi \mathbf{u}) - \mathbf{S}(\varphi) \right] d\Omega = 0$$

and integrating by parts (Green's theorem) such as

$$\nabla \cdot (\psi \varphi \mathbf{u}) = \psi \nabla \cdot (\varphi \mathbf{u}) + (\varphi \mathbf{u}) \cdot \nabla \psi$$

to arrive at

$$\int_{\Omega} \psi \frac{\partial \varphi}{\partial t} \, d\Omega + \int_{\Gamma} \mathbf{n} \cdot \mathbf{u} \psi \varphi \, d\Gamma - \int_{\Omega} \nabla \psi \cdot (\varphi \mathbf{u}) \, d\Omega - \int_{\Omega} \mathbf{S}(\varphi) \, d\Omega = 0.$$

In the case of no-flux boundary conditions, the second integral vanishes, in other words

$$\int_{\Gamma} \mathbf{n} \cdot \mathbf{u} \psi \varphi \, d\Gamma = 0,$$

and we are left with

$$\int_\Omega \psi \frac{\partial \varphi}{\partial t}\, d\Omega = \int_\Omega \nabla\psi \cdot (\varphi \mathbf{u})\, d\Omega + \int_\Omega \mathbf{S}(\varphi)\, d\Omega.$$

The resulting system of ordinary differential equations can now be written as

$$\frac{\partial \varphi}{\partial t} = H(\varphi),$$

where

$$H(\varphi) = M_{ij}^{-1}[(\mathbf{A}_{ijk} \cdot \mathbf{u}_k)\, \varphi_j + S_i(\varphi)]$$

and

$$\mathbf{A}_{ijk} = \int_\Omega \nabla\psi_i \psi_j \psi_k \, d\Omega.$$

Integrating in time by a general family of Runge-Kutta schemes yields

$$\varphi^{k+1} = \varphi^n + \Delta t \beta \, H\left(\varphi^{k-1}\right),$$

where

$$\beta = \frac{1}{M-k+1}, \qquad k = 1, \dots, M \quad \text{and} \quad \varphi^0 = \varphi^n.$$

This scheme is required for the first time step because, as you will see, Lagrange-Galerkin methods require the variables to be known at two time levels before they can be used. Alternatively, we can also derive a second-order leapfrog scheme as follows:

$$\varphi^{n+1} = \varphi^{n-1} + 2\Delta t \, H\left(\varphi^n\right). \tag{11}$$

This scheme is used to obtain the time step used for comparison in the time step ratio σ in (22) in the results section.

5. STRONG LAGRANGE-GALERKIN METHOD

5.1. Spatial Discretization

In order to arrive at the strong form of (2) we first need to expand the divergence term

$$\nabla \cdot (\varphi \mathbf{u}) = \varphi \nabla \cdot \mathbf{u} + \mathbf{u} \cdot \nabla \varphi,$$

thereby obtaining

$$\int_\Omega \psi \left[\frac{\partial \varphi}{\partial t} + \mathbf{u} \cdot \nabla\varphi + \varphi \nabla \cdot \mathbf{u} - \mathbf{S}(\varphi)\right] d\Omega = 0,$$

where

$$\frac{d\varphi}{dt} = \frac{\partial \varphi}{\partial t} + \mathbf{u} \cdot \nabla \varphi$$

denotes the Lagrangian derivative. Substituting this relation results in the following final form:

$$\int_\Omega \psi \left(\frac{d\varphi}{dt}\right) d\Omega = \int_\Omega \psi \left(\mathbf{S}(\varphi) - \varphi \nabla \cdot \mathbf{u}\right) d\Omega, \tag{12}$$

where

$$\frac{d\mathbf{x}}{dt} = \mathbf{u} \tag{13}$$

is used to obtain the particle trajectories. Equation (13) is solved using the Runge-Kutta method presented in [15].

5.2. Time Discretization

A strong Lagrange-Galerkin spectral element method was introduced in our previous paper [3]. This method proved to be extremely accurate and stable. The time discretization implemented in that scheme was the θ algorithm. Let us apply this time discretization method to the shallow water equations. Integrating (12) in time by the θ algorithm yields

$$\begin{aligned} \int_{\Omega^{n+1}} \left(\psi\varphi^{n+1}\right) d\Omega^{n+1} &= \int_{\Omega^{n+1}} \left(\psi\varphi^{n}\right) d\Omega^{n+1} \\ &+ \Delta t \int_{\Omega^{n+1}} \psi \left[\theta \left(\mathbf{S}\left(\varphi^{n+1}\right) - \varphi^{n+1}\nabla\cdot\mathbf{u}^{n+1}\right)\right. \\ &+ \left.(1-\theta)\left(\mathbf{S}\left(\varphi^{n}\right) - \varphi^{n}\nabla\cdot\mathbf{u}^{n}\right)\right] d\Omega^{n+1}, \end{aligned} \tag{14}$$

which represents a two-time level scheme and gives the forward Euler, trapezoid rule, and backward Euler for $\theta = 0, 1/2, 1$, respectively. The scheme $\theta = 1/2$ yields a second-order accurate scheme and is unconditionally stable with respect to pure advection. The other two schemes are both first order and the $\theta = 0$ is only conditionally stable while the $\theta = 1$ is too diffusive. Let us now see what the element equations would look like using this formulation.

5.3. Element Equations

Returning to (14) and substituting the conservation variables from (3) gives

$$\begin{aligned} \int_{\Omega^{n+1}} \psi \begin{bmatrix} \varphi \\ \varphi u \\ \varphi v \end{bmatrix}^{n+1} d\Omega^{n+1} &= \int_{\Omega^{n+1}} \psi \begin{bmatrix} \varphi \\ \varphi u \\ \varphi v \end{bmatrix}^{n} d\Omega^{n+1} \\ &+ \Delta t\theta \int_{\Omega^{n+1}} \psi \begin{bmatrix} -\varphi\left(\dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y}\right) \\ -\varphi\dfrac{\partial\varphi}{\partial x} + f\varphi v - \varphi u\left(\dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y}\right) \\ -\varphi\dfrac{\partial\varphi}{\partial y} - f\varphi u - \varphi v\left(\dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y}\right) \end{bmatrix}^{n+1} d\Omega^{n+1} \\ &+ \Delta t(1-\theta) \int_{\Omega^{n+1}} \psi \begin{bmatrix} -\varphi\left(\dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y}\right) \\ -\varphi\dfrac{\partial\varphi}{\partial x} + f\varphi v - \varphi u\left(\dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y}\right) \\ -\varphi\dfrac{\partial\varphi}{\partial y} - f\varphi u - \varphi v\left(\dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y}\right) \end{bmatrix}^{n} d\Omega^{n+1}, \end{aligned}$$

which are the equations to be solved within each element. Approximating the conservation variables within each element Ω by the following expansion:

$$\varphi^{(e)} = \sum_{j=1}^{P^2} \psi_j\varphi_j$$

and letting

$$\begin{aligned} M_{ij} &= \int_\Omega \psi_i\psi_j \, d\Omega, \\ D_{ij} &= \int_\Omega \psi_i\psi_j \sum_{k=1}^{P^2} \left(\frac{\partial\psi_k}{\partial x} u_k + \frac{\partial\psi_k}{\partial y} v_k\right) d\Omega, \end{aligned}$$

$$F_{ij} = \int_\Omega \psi_i \psi_j \sum_{k=1}^{P^2} (\psi_k f_k) \, d\Omega,$$

$$P^u_{ij} = \int_\Omega \psi_i \psi_j \sum_{k=1}^{P^2} \left(\frac{\partial \psi_k}{\partial x} \varphi_k \right) d\Omega,$$

$$P^v_{ij} = \int_\Omega \psi_i \psi_j \sum_{k=1}^{P^2} \left(\frac{\partial \psi_k}{\partial y} \varphi_k \right) d\Omega,$$

gives the following matrix form:

$$\begin{bmatrix} M_{ij} + \Delta t \theta D_{ij} & 0 & 0 \\ \Delta t \theta P^u_{ij} & M_{ij} + \Delta t \theta D_{ij} & -\Delta t \theta F_{ij} \\ \Delta t \theta P^v_{ij} & +\Delta t \theta F_{ij} & M_{ij} + \Delta t \theta D_{ij} \end{bmatrix}^{n+1} \begin{bmatrix} (\varphi)_j \\ (\varphi u)_j \\ (\varphi v)_j \end{bmatrix}^{n+1} = \begin{bmatrix} b^\varphi_i \\ b^u_i \\ b^v_i \end{bmatrix} \tag{15}$$

where

$$\begin{bmatrix} b^\varphi_i \\ b^u_i \\ b^v_i \end{bmatrix} = \begin{bmatrix} [M_{ij} - \Delta t(1-\theta) D_{ij}](\varphi)_j \\ [M_{ij} - \Delta t(1-\theta) D_{ij}](\varphi u)_j - \Delta t(1-\theta) \left[P^u_{ij}(\varphi)_j - F_{ij}(\varphi v)_j \right] \\ [M_{ij} - \Delta t(1-\theta) D_{ij}] (\varphi v)_j - \Delta t(1-\theta) \left[P^v_{ij}(\varphi)_j + F_{ij}(\varphi u)_j \right] \end{bmatrix}^n,$$

which represents not only a strongly coupled system but also a nonlinear system. We can also see that for divergence free flow ($D_{ij} = 0$) the mass equation becomes linear. Case 1 represents this type of flow and so we will not encounter any nonlinearities in Case 1 and we could then use the θ algorithm. However, for divergent flow we have nonlinearities and in order to avoid them we replace the θ algorithm with the leap frog scheme given in (11), except that now the integration in time is done along the characteristics. This completely avoids having to invert a nonlinear matrix; however, there is a price to be paid for simplicity and this price comes in the form of inefficiency due to the smaller time step required to maintain stability. In the next section, we explore another type of Lagrange-Galerkin method that circumvents this problem.

6. WEAK LAGRANGE-GALERKIN METHOD

6.1. Spatial Discretization

The weak form of (2) is

$$\int_\Omega \psi \left[\frac{\partial \varphi}{\partial t} + \nabla \cdot (\varphi \mathbf{u}) - \mathbf{S}(\varphi) \right] d\Omega = 0$$

and using the calculus identities

$$\frac{\partial}{\partial t} (\psi \varphi) = \psi \frac{\partial \varphi}{\partial t} + \varphi \frac{\partial \psi}{\partial t} \qquad \text{and} \qquad \nabla \cdot (\psi \varphi \mathbf{u}) = \psi \nabla \cdot (\varphi \mathbf{u}) + (\varphi \mathbf{u}) \cdot \nabla \psi$$

yields

$$\int_\Omega \left[\frac{\partial}{\partial t} (\psi \varphi) + \nabla \cdot (\psi \varphi \mathbf{u}) \right] - \varphi \left[\frac{\partial \psi}{\partial t} + \mathbf{u} \cdot \nabla \psi \right] - [\psi \mathbf{S}(\varphi)] \, d\Omega = 0. \tag{16}$$

The first bracketed term of (16), using Reynold's transport theorem, can be written as

$$\frac{d}{dt} \int_\Omega (\psi \varphi) \, d\Omega = \int_\Omega \left[\frac{\partial}{\partial t} (\psi \varphi) + \nabla \cdot (\psi \varphi \mathbf{u}) \right] d\Omega \tag{17}$$

and the second bracketed term is actually the characteristic equation

$$\frac{d\psi}{dt} \equiv \frac{\partial \psi}{\partial t} + \mathbf{u} \cdot \nabla \psi = 0, \tag{18}$$

where

$$\frac{d\mathbf{x}}{dt} = \mathbf{u}$$

is used to predict the particle trajectories along which the basis functions vanish. The characteristic equation is equal to zero because we are constraining the basis functions to be constant along the particle trajectories. Therefore, in essence, the basis functions ψ are functions of both space and time (see [15] for a more detailed discussion).

Substituting (17) and (18) into (16) yields the system

$$\frac{d}{dt}\int_{\Omega}(\psi\varphi)\,d\Omega = \int_{\Omega}\psi\mathbf{S}(\varphi)\,d\Omega. \tag{19}$$

Note that the advection operator has disappeared from the equations; however, the correct particle trajectories are accounted for by the trajectory equation (13). In addition, consider that the divergence of the velocity has also disappeared or rather has been absorbed by virtue of the Reynold's transport theorem.

6.2. Time Discretization

Integrating (19) in time by the θ algorithm yields

$$\int_{\Omega^{n+1}}(\psi\varphi)\,d\Omega^{n+1} = \int_{\Omega^n}(\psi\varphi)\,d\Omega^n + \Delta t\left[\theta\int_{\Omega^{n+1}}\psi\mathbf{S}(\varphi)\,d\Omega^{n+1} + (1-\theta)\int_{\Omega^n}\psi\mathbf{S}(\varphi)\,d\Omega^n\right], \tag{20}$$

which represents a two-time level scheme. In this paper, $\theta = 1/2$ is used because it yields a second-order accurate scheme and is unconditionally stable with respect to pure advection. The trajectory equation (13) is required for closure. The accurate solution of the trajectory equation is perhaps the most important part of Lagrangian methods. Once again, we solve this equation using the Runge-Kutta trajectory calculation scheme described in one of our previous papers [15].

6.3. Element Equations

Returning to (20) and substituting in the conservation variables from (3) gives

$$\begin{aligned}\int_{\Omega^{n+1}}\psi\begin{bmatrix}\varphi\\ \varphi u\\ \varphi v\end{bmatrix}^{n+1} d\Omega^{n+1} &= \int_{\Omega^n}\psi\begin{bmatrix}\varphi\\ \varphi u\\ \varphi v\end{bmatrix}^{n} d\Omega^{n}\\ &+ \Delta t\theta\int_{\Omega^{n+1}}\psi\begin{bmatrix}0\\ -\varphi\dfrac{\partial\varphi}{\partial x} + f\varphi v\\ -\varphi\dfrac{\partial\varphi}{\partial y} - f\varphi u\end{bmatrix}^{n+1} d\Omega^{n+1}\\ &+ \Delta t(1-\theta)\int_{\Omega^n}\psi\begin{bmatrix}0\\ -\varphi\dfrac{\partial\varphi}{\partial x} + f\varphi v\\ -\varphi\dfrac{\partial\varphi}{\partial y} - f\varphi u\end{bmatrix}^{n} d\Omega^{n},\end{aligned}$$

which are the equations to be solved within each element. Once again approximating the conservation variables within each element Ω by the relation

$$\varphi^{(e)} = \sum_{j=1}^{P^2}\psi_j\varphi_j$$

and using the following notations for the different matrices:

$$
\begin{aligned}
M_{ij} &= \int_\Omega \psi_i \psi_j \, d\Omega = \int_{-1}^{+1} \int_{-1}^{+1} |J| \, \psi_i(\xi,\eta) \, \psi_j(\xi,\eta) \, d\xi \, d\eta, \\
F_{ij} &= \int_\Omega \psi_i \psi_j \sum_{k=1}^{P^2} (\psi_k f_k) \, d\Omega = \int_{-1}^{+1} |J| \, \psi_i(\xi,\eta) \, \psi_j(\xi,\eta) \sum_{k=1}^{P^2} \left[\psi_k(\xi,\eta) \, f_k \right] d\xi \, d\eta, \\
P_{ij}^u &= \int_\Omega \psi_i \psi_j \sum_{k=1}^{P^2} \left(\frac{\partial \psi_k}{\partial x} \varphi_k \right) d\Omega \\
&= \int_{-1}^{+1} \int_{-1}^{+1} |J| \, \psi_i(\xi,\eta) \, \psi_j(\xi,\eta) \sum_{k=1}^{P^2} \left[\left(\frac{\partial \psi_k(\xi,\eta)}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial \psi_k(\xi,\eta)}{\partial \eta} \frac{\partial \eta}{\partial x} \right) \varphi_k \right] d\xi \, d\eta, \\
P_{ij}^v &= \int_\Omega \psi_i \psi_j \sum_{k=1}^{P^2} \left(\frac{\partial \psi_k}{\partial y} \varphi_k \right) d\Omega \\
&= \int_{-1}^{+1} \int_{-1}^{+1} |J| \, \psi_i(\xi,\eta) \, \psi_j(\xi,\eta) \sum_{k=1}^{P^2} \left[\left(\frac{\partial \psi_k(\xi,\eta)}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial \psi_k(\xi,\eta)}{\partial \eta} \frac{\partial \eta}{\partial y} \right) \varphi_k \right] d\xi \, d\eta,
\end{aligned}
$$

yields the following matrix form:

$$
\begin{bmatrix} M_{ij} & 0 & 0 \\ \Delta t \theta P_{ij}^u & M_{ij} & -\Delta t \theta F_{ij} \\ \Delta t \theta P_{ij}^v & +\Delta t \theta F_{ij} & M_{ij} \end{bmatrix}^{n+1} \begin{bmatrix} (\varphi)_j \\ (\varphi u)_j \\ (\varphi v)_j \end{bmatrix}^{n+1} = \begin{bmatrix} b_i^\varphi \\ b_i^u \\ b_i^v \end{bmatrix}, \tag{21}
$$

where

$$
\begin{bmatrix} b_i^\varphi \\ b_i^u \\ b_i^v \end{bmatrix} = \begin{bmatrix} M_{ij}(\varphi)_j \\ M_{ij}(\varphi u)_j - \Delta t(1-\theta) \left[P_{ij}^u (\varphi)_j - F_{ij} (\varphi v)_j \right] \\ M_{ij}(\varphi v)_j - \Delta t(1-\theta) \left[P_{ij}^v (\varphi)_j + F_{ij} (\varphi u)_j \right] \end{bmatrix}^n,
$$

which appears to be nonlinear. However, note that the mass equation (the first row) is completely decoupled from the momentum equations (second and third rows). Therefore, we can solve for the mass and then use the now known values of φ^{n+1} in the momentum equations. Therefore, the terms P_{ij}^u and P_{ij}^v can now be moved to the right-hand side which then yields a linear but skewed symmetric matrix. The difference between the strong (15) and weak (21) methods is that the weak method does not contain the velocity divergence term. Note also that in the strong method all of the integrals occur within the domain (elements) at the time $n+1$. However, in the weak method the left-hand matrices occur at this new time but the right-hand side matrices are all integrated at the feet of the characteristics, namely at n. Because in the weak method, we are actually integrating the control volumes at the feet of the characteristics, this operation then takes the place of the velocity divergence; in other words, the integrals account for the dilation (expansion or contraction) of the fluid volume. This process is depicted in Figure 1. In the strong method, as in any Eulerian scheme this has to be accomplished explicitly by including the divergence term. Therefore, it can be said that the weak method relies heavily on integration as opposed to the strong method, which relies more on interpolation of the values at the feet of the characteristics.

The news is not all good for the weak method, however. Because the resulting system is not symmetric we cannot utilize many of the best known solvers such as the conjugate gradient method. There are, however, variants of these methods that can be used such as the biconjugate gradient method. Because the system is quite sparse, direct methods are inefficient. For the moment, we are employing a simple Jacobi iteration method, although we are also currently exploring more sophisticated methods.

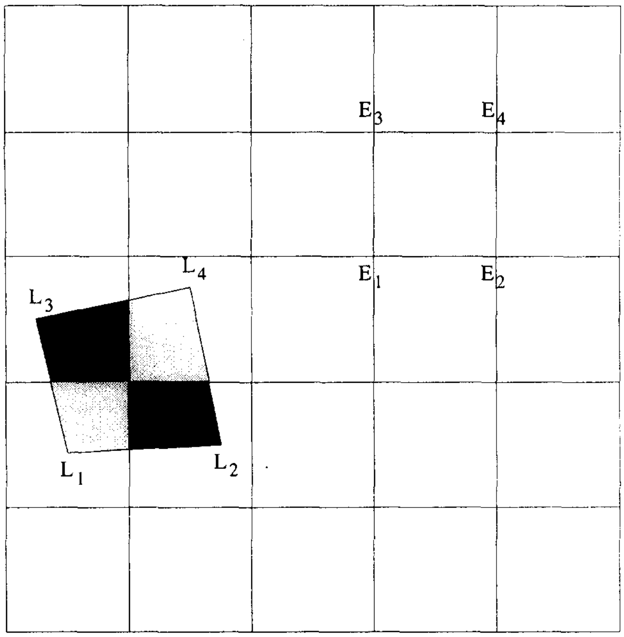

Figure 1. The dilation of the Eulerian element (E) area to the Lagrangian element (L) area traced by the particle trajectories.

7. SEARCHING ALGORITHMS

The crux of any Lagrangian scheme is the accurate calculation of the particle trajectories, in other words, the correct solution of the trajectory equation (13). Once the correct trajectories are computed, it then remains to interpolate either the departure points (in the strong method) or the quadrature points (in the weak method). For structured or quasi-structured grids, *ad hoc* searching algorithms can be constructed, but for unstructured or general grids quadtree algorithms are the best choice.

Note that the trajectory equation only gives us the departure point $\mathbf{x}_d$ and tells us nothing of where this point is located. Therefore, we need to devise some means of searching the elements of the grid to determine which element contains the departure point in order to then interpolate the variables (in this case φ and $\mathbf{u}$) at this point by virtue of that element's basis functions. Once we isolate the element claiming the departure point, 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 (ξ, η) and not the physical coordinates (x, y). Equation (13) will only give the coordinates of the departure point in terms of the physical space. We can write the coordinates in physical space of the departure point using the basis functions in the form

$$\mathbf{x}_d = \sum_{i=1}^{P^2} x_i \psi_i(\xi_d, \eta_d)$$

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

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

where

$$\varphi = \sum_{i=1}^{P^2} x_i \psi_i(\xi_d, \eta_d) - \mathbf{x}_d.$$

This leads to the solutions

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

where

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

which only requires five iterations at most for convergence. Thus, if $(\xi_d^{k+1}, \eta_d^{k+1}) \in [-1, 1]$, then we can be sure that the departure point is contained within this element. Clearly, as p increases, the cost of our searches will increase by an order of p^2. So instead of using the full polynomial of degree p, 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 can then be obtained using the full p^{th}-order basis functions of the element. The use of the linear basis functions to check whether (ξ_d, η_d) is contained within an element has absolutely no impact on the accuracy of the scheme, while costing far less than using the full p^{th}-order basis functions.

Because the Lagrange-Galerkin method requires the calculation of the departure points, the success of the method hinges on the rapidity of the searching algorithms. In this search, we need to determine in which elements the departure points lie. For general grids, the best strategy is to find the closest grid point to the departure point in question 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] stores 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, and
- quad_tree[i, 7] stores the number of points 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 points comprising the spectral element grid. In fact, we only need to use the vertices of each quadrilateral element (i.e., the four corner points) and we can omit the rest of the collocation points. This saves a lot of effort in the searching process. Upon finding this nearest neighbor (closest point), we then search through the list of elements which claim this point and check for inclusion using the iterative approach outlined in the previous section. There are usually no more than six elements claiming each point even for distorted unstructured grids, meaning 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 points is stored. 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 of the iterative approach are required. This can have adverse affects on the efficiency of the scheme if this case were to arise for many of the points being searched; for the grids used in this paper, this case did not arise at all.

8. 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,

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

where φ represents any of the conservation variables, and the following two moments:

$$M = \frac{\int_\Omega \varphi \, d\Omega}{\int_\Omega \varphi_{\text{exact}} \, d\Omega}, \qquad E = \frac{\int_\Omega \left[\varphi \left(u^2 + v^2\right) + \varphi^2\right] d\Omega}{\int_\Omega \left[\varphi_{\text{exact}} \left(u^2_{\text{exact}} + v^2_{\text{exact}}\right) + \varphi^2_{\text{exact}}\right] d\Omega}.$$

The L_2 error norm compares the root mean square percent error of the numerical and exact solutions, M measures the conservation property of the mass, and E measures the conservation of the total available energy. The ideal scheme should yield an L_2 error norm of zero, and mass and energy moments of one.

In addition, the following time-step ratio is used:

$$\sigma = \frac{\Delta t}{\Delta t_{\text{LF}}}. \tag{22}$$

This variable represents the time step ratio between the Lagrange-Galerkin spectral element methods and the maximum allowable time step for the Eulerian spectral element method with the leapfrog scheme given in (11).

8.1. Problem Statement

In the following sections, the two test cases used to measure the performance of the Lagrange-Galerkin spectral element methods are introduced. They are a solid body rotation with time-independent velocity field and a westward traveling soliton wave.

8.1.1. Case 1: Solid body rotation

The initial condition for this test case is given by the Gaussian wave

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

having the far boundary conditions

$$\varphi(x, y, t) = 0, \qquad \forall\, (x, y) \in \Gamma,$$

where

$$\lambda_o = \frac{1}{8}, \quad (x_o, y_o) = \left(-\frac{1}{2}, 0\right), \qquad (x, y) \in [-1, 1].$$

The velocity field is constant for all time and is given by

$$u = +y \qquad \text{and} \qquad v = -x,$$

which defines a clockwise rotation about the center of the domain. There are no Coriolis effects and since only the mass is allowed to vary, this problem simplifies to the passive advection of the quantity φ. The Gaussian wave rotates along this circular path with neither distortion nor dissipation. The exact solution is given by

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

where

$$\tilde{x} = x - x_o \cos t - y_o \sin t \qquad \text{and} \qquad \tilde{y} = y + x_o \sin t - y_o \cos t.$$

Results are given for one full revolution of the initial wave. The period for one revolution of the wave is 2π which means that one revolution corresponds to $t = 2\pi$. For the p-type grid with $N_P = 60$, the maximum allowable time step for the Eulerian spectral element leapfrog scheme is $\Delta t_{\mathrm{LF}} = 2\pi/1000$. The p-type $N_P = 60$ grid is comprised of a 10 by 10 element grid with each element having sixth-order polynomials. (See the results for a discussion on the h- and p-type spectral element methods.)

8.1.2. Case 2: Rossby soliton waves

This problem describes a pair of equatorially trapped Rossby soliton waves [19,20]. The pair of soliton waves start off in the center of the domain. They then begin to move westward together along the equator and should continue to move in this direction without changing shape and without moving either closer together or farther apart. The exact solution is given by

$$\begin{aligned}
\varphi(x,y,t) &= \varphi^{(0)} + \varphi^{(1)},\\
u(x,y,t) &= u^{(0)} + u^{(1)},\\
v(x,y,t) &= v^{(0)} + v^{(1)},
\end{aligned}$$

where the superscripts (0) and (1) denote the zeroth- and first-order asymptotic solutions of the shallow water equations, respectively. They are given by

$$\begin{aligned}
\varphi^{(0)} &= \eta \left(\frac{-9+6y^2}{4}\right) e^{-y^2/2},\\
u^{(0)} &= \frac{\partial \eta}{\partial \xi} \, (2y)\, e^{-y^2/2},\\
v^{(0)} &= \eta \left(\frac{3+6y^2}{4}\right) e^{-y^2/2},
\end{aligned}$$

and

$$\begin{aligned}
\varphi^{(1)} &= c^{(1)} \eta \frac{9}{16} \left(-5+2y^2\right) e^{-y^2/2} + \eta^2 \Phi^{(1)}(y),\\
u^{(1)} &= c^{(1)} \eta \frac{9}{16} \left(3+2y^2\right) e^{-y^2/2} + \eta^2 U^{(1)}(y),\\
v^{(1)} &= \frac{\partial \eta}{\partial \xi} \eta V^{(1)}(y),
\end{aligned}$$

where $\eta(\xi,t) = A\,\mathrm{sech}^2 B\xi$, $\xi = x - ct$, $A = 0.771\,B^2$, $B = 0.394$, and $c = c^{(0)} + c^{(1)}$ where $c^{(0)} = -1/3$ and $c^{(1)} = -0.395\,B^2$. The variable η is the solution to the equation

$$\frac{\partial \eta}{\partial \tau} + \alpha_n \eta \frac{\partial \eta}{\partial \xi} + \beta_n \frac{\partial^3 \eta}{\partial \xi^3} = 0,$$

which is the famous Korteweg-de Vries equation that yields soliton solutions. The shallow water equations can be simplified into this equation using the method of multiple scales [19]. Finally, the remaining terms are given by

$$\begin{pmatrix} \Phi^{(1)}(y) \\ U^{(1)}(y) \\ V^{(1)}(y) \end{pmatrix} = e^{-y^2/2} \sum_{n=0}^{\infty} \begin{pmatrix} \varphi_n \\ u_n \\ v_n \end{pmatrix} H_n(y),$$

where $H_n(y)$ are the Hermite polynomials and φ_n, u_n, v_n are the Hermite series coefficients given in [20]. The boundary conditions used are

$$(u, v) = 0, \qquad \forall\,(x, y) \in \Gamma.$$

The equations are integrated up to a nondimensional time of $t = 10$. For the p-type grid with $N_P = 64$ the maximum allowable time step for the Eulerian spectral element method with the leapfrog scheme is $\Delta t_{\mathrm{LF}} = 1/20$. The p-type $N_P = 64$ grid is comprised of a 16 by 8 element grid with each element having fourth-order polynomials.

8.2. Results

Figures 2 and 3 show the grid and φ contours for h- and p-type Lagrange-Galerkin spectral element methods for Case 2. The thicker lines of the grids denote the elements while the fainter lines denote the high-order collocation points. In the h-type method, the order of the basis functions is kept low such as $p = 1$ while the number of spectral elements is increased in order to refine the grid. In the p-type method, on the other hand, the number of spectral elements is held constant while the order of the basis functions is increased. It has been shown previously that for smooth solutions the p-type method exhibits better convergence over the h-type method. In fact, the order of convergence for the p-type method is exponential while that for the h-type method is only algebraic. However, since we cannot always guarantee that the solution will be smooth, using a combination of both the h- and p-type methods is optimal.

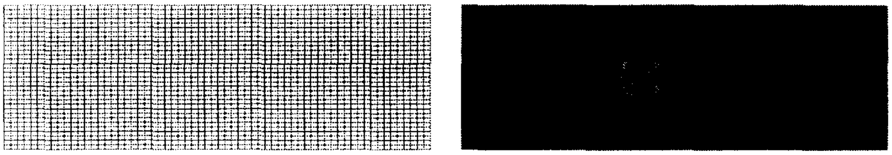

Figure 2. Case 2. The grid and φ contours for the h-type form of the Lagrange-Galerkin spectral element method with $N_P = 64$.

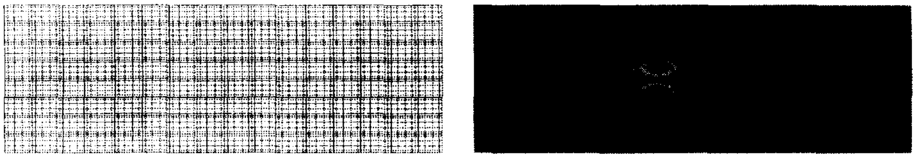

Figure 3. Case 2. The grid and φ contours for the p-type form of the Lagrange-Galerkin spectral element method with $N_P = 64$.

8.2.1. Case 1: Solid body rotation

Figure 4 shows the results for both the strong and weak methods using the h-type method. In this figure, N_P denotes the total number of grid points in each direction (horizontal and vertical) while $\sigma = 1$ unless otherwise stated. Figure 4 shows that the weak method performs better as N_P increases. Figure 5 plots the results for both methods but now using the p-type method. In this case, the weak method performs better until about $N_P = 70$ corresponding to $p = 7$. After this point, the strong method performs better than the weak method.

Figures 6 and 7 compare the h- and p-type methods for the strong and weak methods, respectively. Figure 6 shows that there is a much larger gap in the accuracy between the h- and p-type methods for the strong method than for the weak method.

In summary, both methods perform far better when using the p-type method over the h-type method, the reason being that this case represents a smooth flow and for this type of flow the p-type method converges exponentially. This would also be the case for an Eulerian spectral element method; however, the Lagrange-Galerkin methods converge faster than the Eulerian

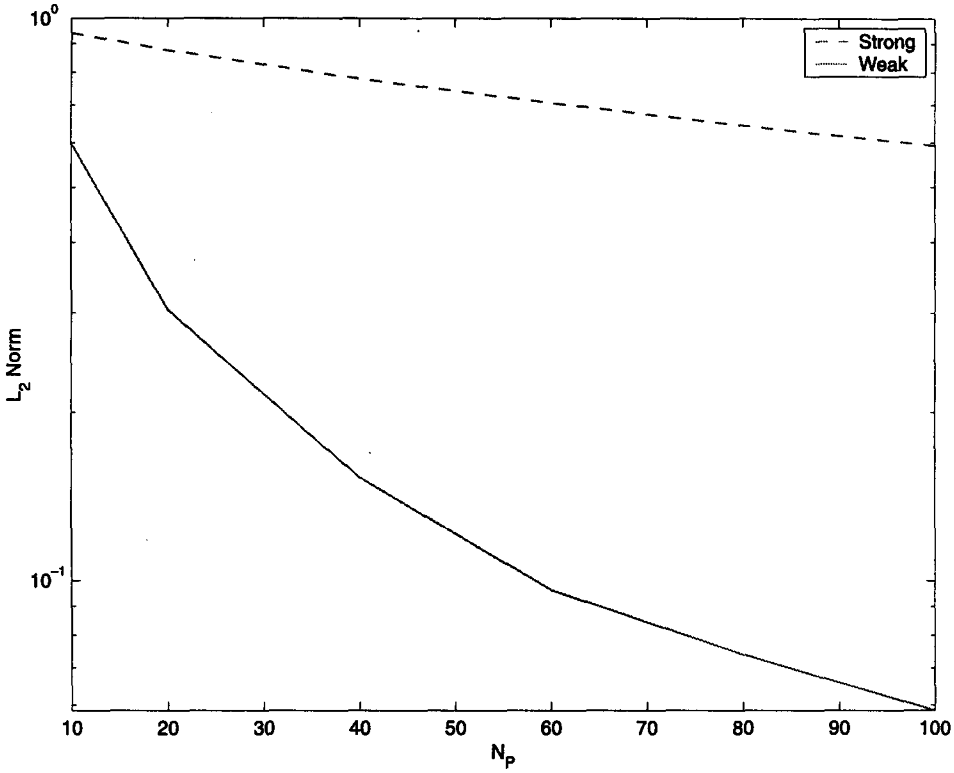

Figure 4. Case 1. The φ L_2 norm as a function of N_P for the h-type form of the strong and weak Lagrange-Galerkin spectral element methods.

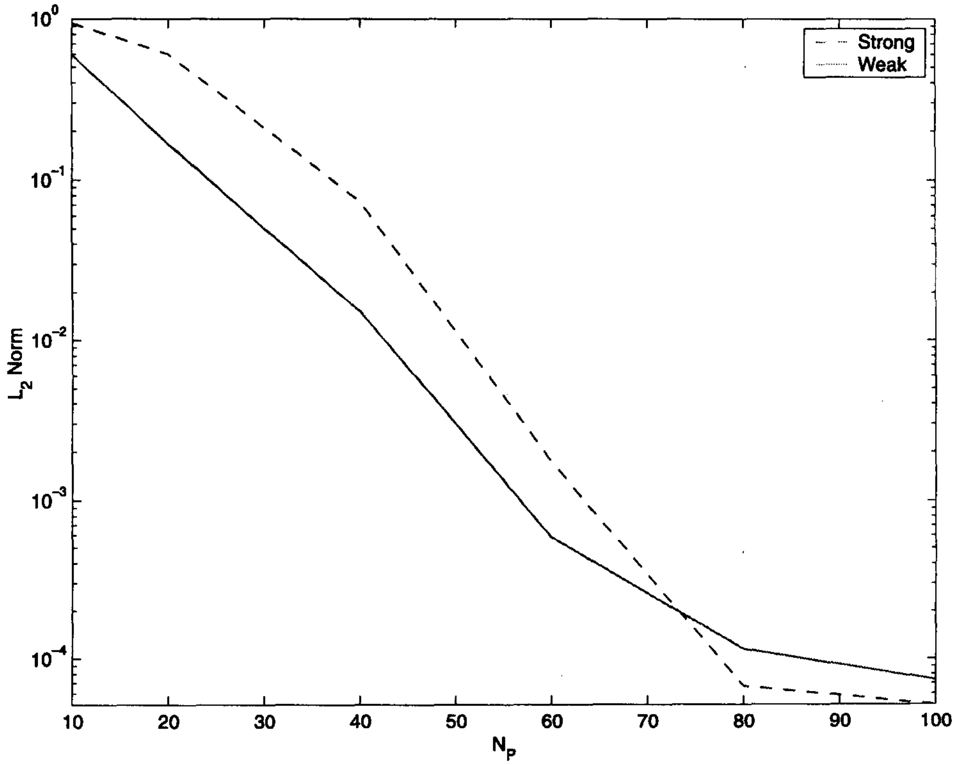

Figure 5. Case 1. The φ L_2 norm as a function of N_P for the p-type form of the strong and weak Lagrange-Galerkin spectral element methods.

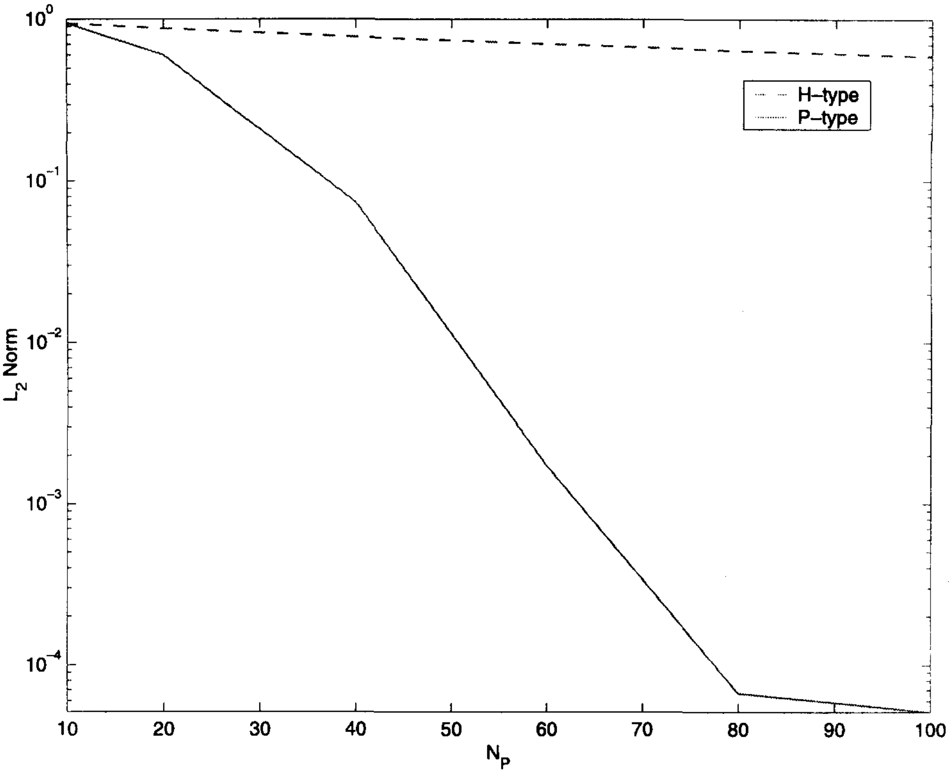

Figure 6. Case 1. The φ L_2 norm as a function of N_P for the h- and p-type forms of the strong Lagrange-Galerkin spectral element method.

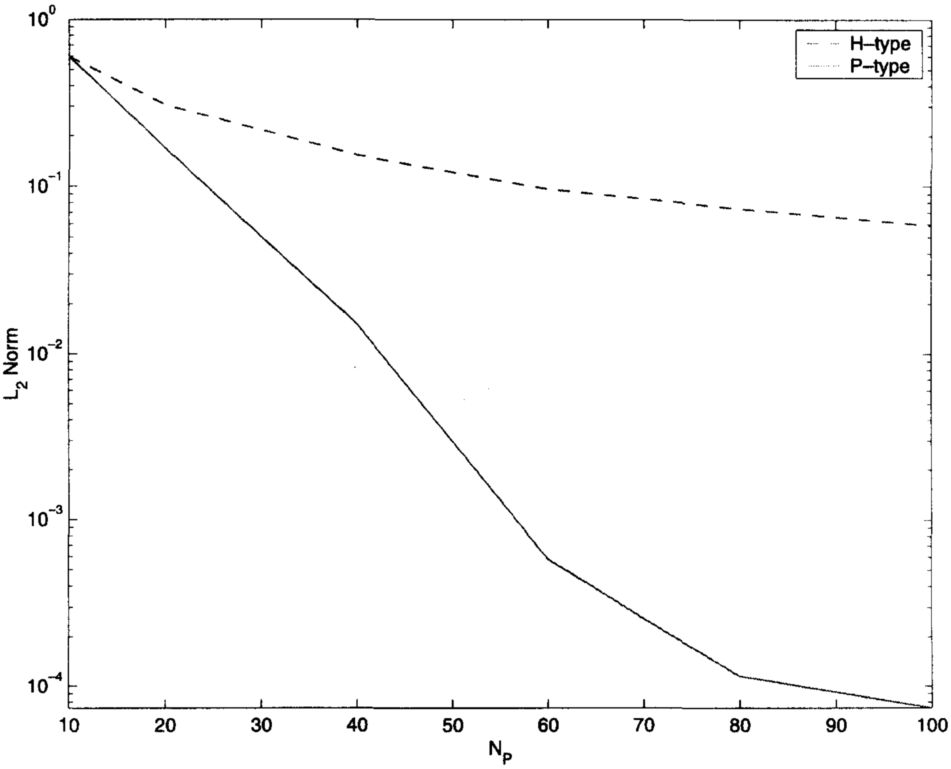

Figure 7. Case 1. The φ L_2 norm as a function of N_P for the h- and p-type forms of the weak Lagrange-Galerkin spectral element method.

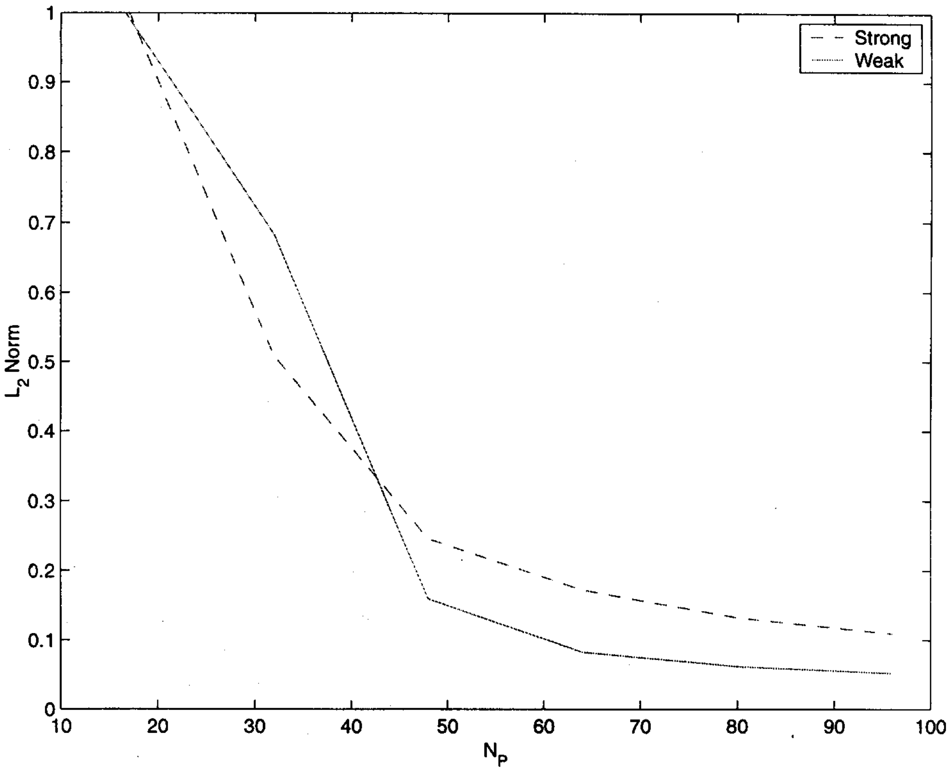

Figure 8. Case 2. The φ L_2 norm as a function of N_P for the h-type form of the strong and weak Lagrange-Galerkin spectral element methods.

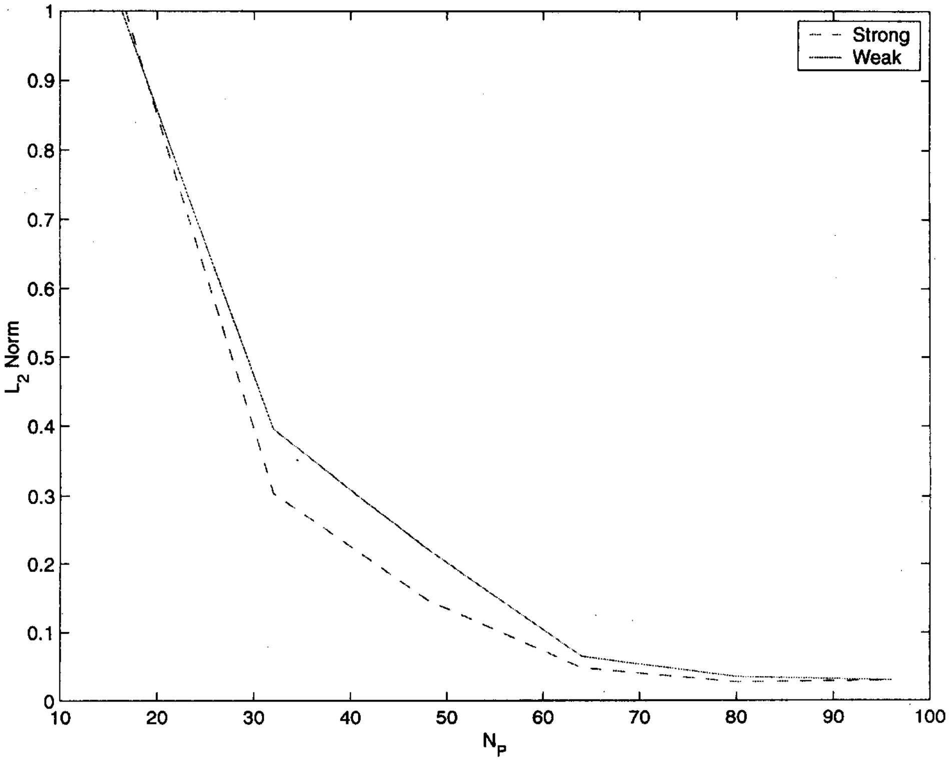

Figure 9. Case 2. The φ L_2 norm as a function of N_P for the p-type form of the strong and weak Lagrange-Galerkin spectral element methods.

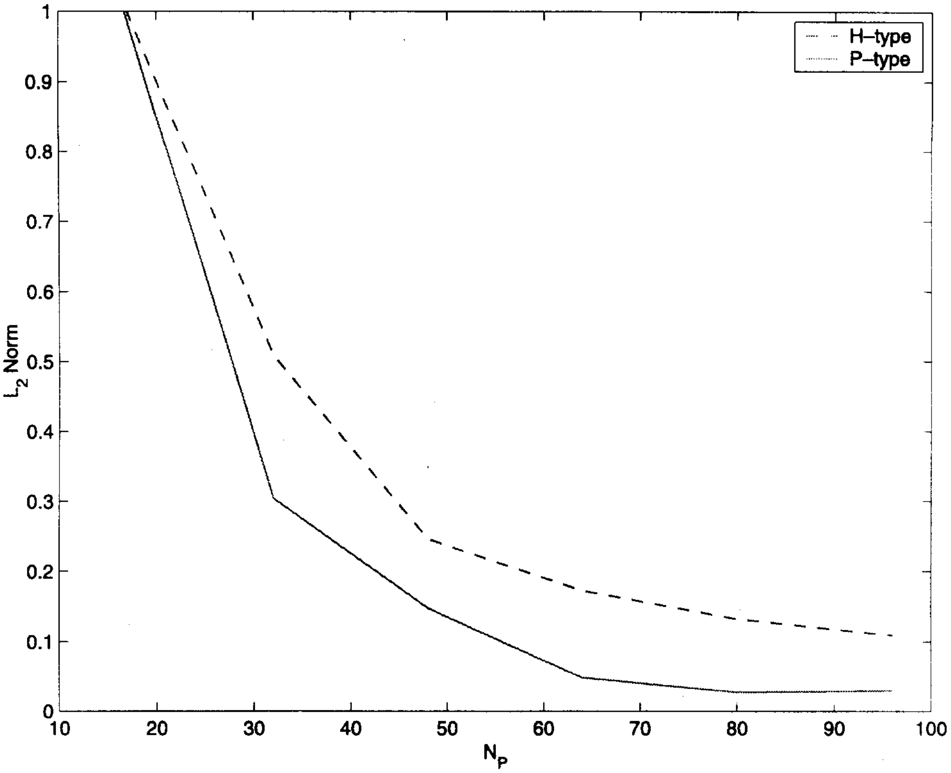

Figure 10. Case 2. The φ L_2 norm as a function of N_P for the h- and p-type forms of the strong Lagrange-Galerkin spectral element method.

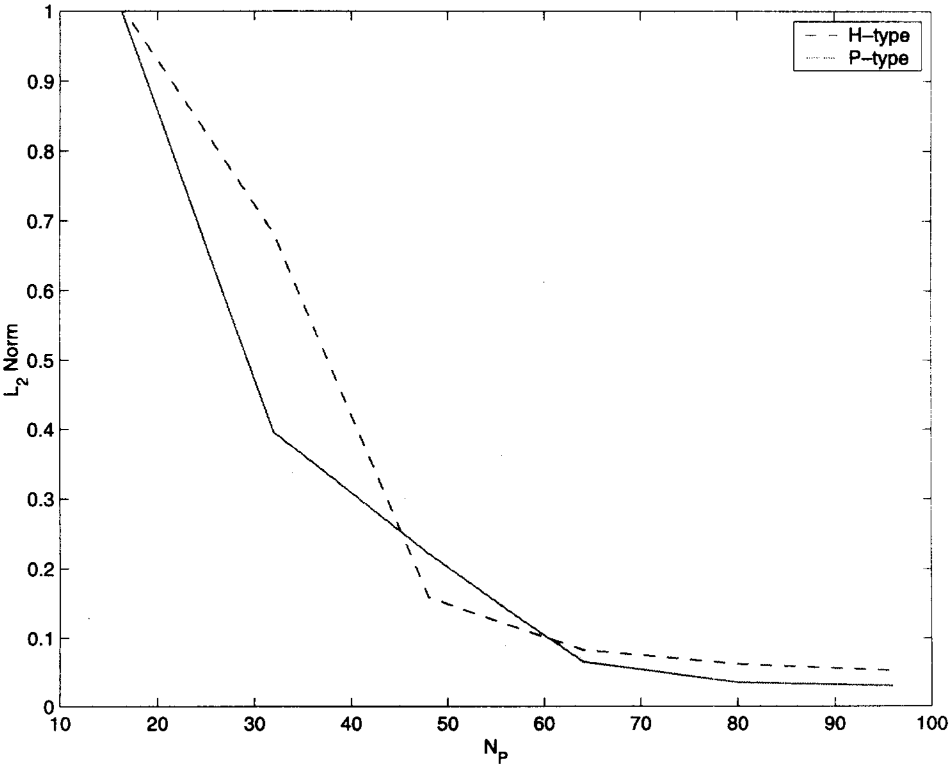

Figure 11. Case 2. The φ L_2 norm as a function of N_P comparing the h- and p-type forms of the weak Lagrange-Galerkin method.

methods (see [3]). While the weak method can obtain fair results with the h-type method, the strong method cannot. Since the strong method relies on interpolation, it cannot give good results with the h-type method, because this implies using linear interpolation functions. Therefore, the strong method is much more sensitive to the order of the interpolation functions than the weak method.

8.2.2. Case 2: Rossby soliton waves

Figure 8 shows the results for both the strong and weak methods using an h-type method. In this figure, N_P denotes the total number of horizontal grid points in the grid. In all cases, the number of vertical grid points is $N_P/2$ and $\sigma = 1$ unless otherwise stated. Figure 8 shows that the weak method performs better than the strong method as N_P increases. Figure 9 plots the results for both methods, but now using the p-type method. This figure shows that the strong and weak methods give very similar results. It is in fact not surprising since the strong method relies heavily on interpolation and so as the interpolation functions (basis functions) increase in order, then so will the accuracy of the method, whereas the weak method does not benefit as much from an increase in p. To further stress the point, let us look at Figures 10 and 11. These figures compare the h- versus p-type methods for the strong and weak methods, respectively. Figure 10 shows that there is a much larger gap in the accuracy between the h- and p-type methods for the strong method. Clearly, this gap is not as large for the weak method as is illustrated in Figure 11.

From these figures we can surmise that when using the h-type form, the weak method is superior to the strong method and when using the p-type form both methods are more or less equivalent. This parity in accuracy is not only true for the L_2 norm but also for the two conservation measures M and E. In fact, these two measures were conserved exactly. So the question remains: are there any advantages of one scheme over the other? Recall that the strong method had to be solved by an explicit time integration scheme (the leapfrog method) in order to avoid having to invert a nonsymmetric nonlinear matrix. Although explicit schemes are simpler than implicit ones, there is always a price associated with explicit schemes and it usually comes in the form of inefficiency and/or inaccuracies. The inefficiency in this case arises due to the more stringent CFL condition governing explicit schemes, meaning that smaller time steps must be used with these methods. For example, consider that all of the numerical experiments shown above were run with the time step ratio $\sigma = 1$. While this is not the maximum time step for the weak method, it is the maximum for the strong method. However, the time step for the strong method can be increased if we used the θ algorithm instead of the leapfrog scheme. The point of this study was to compare the strong and weak methods and then discuss their strengths and weaknesses without necessarily determining which one is better. Both methods are impressive in their own right, but if simplicity of coding is the primary concern then the strong method should be used. This method, however, will not be as efficient as the weak method because the strong method requires the solution of a nonlinear coupled system. This apparent disadvantage can be turned into an advantage by using a semi-implicit scheme [8], thereby substituting the u and v momentum equations into the mass equation. In essence, we then only need to solve one equation (a Helmholtz equation), albeit a nonlinear one. Clearly, this strategy will not work with the weak method because the mass completely decouples from the momentum. These are issues that need to be further explored. We are currently exploring these issues for the shallow water equations on the sphere. These new schemes are being built directly onto our current Eulerian spectral element model on the sphere [21] with our final goal being the development of a new atmospheric model for numerical weather prediction.

8.2.3. Time step study

Many in the meteorology community have argued that the semi-Lagrangian (i.e., strong Lagrange-Galerkin) method can only be run using time steps a few times higher than Eulerian methods [8,22,23]; these time steps are usually three to six times larger than those permitted by Eulerian schemes. However, this result was based on the low-order interpolations and particle trajectory approximations used. Typically third-order interpolations and only second-order trajectory approximations have been used for these methods. A recent paper by Falcone and Ferretti [24] has shown that the error for the semi-Lagrangian method is of the order

$$O\left(\Delta t^k + \frac{\Delta x^{p+1}}{\Delta t}\right), \tag{23}$$

where k and p represent the order of the trajectory approximation and the order of the interpolation functions, respectively. In other words, for the trajectory equation (13), k represents the order of accuracy of the numerical scheme used to solve this equation while p represents the order of the interpolation functions used to compute the variables at the feet of the characteristics (i.e., departure points). Researchers had previously investigated high k but with low p, thereby allowing the second term in (23) to dominate. Thus, many erroneously assumed that low order k was sufficient; specifically $k = 2$ has been used by almost all semi-Lagrangian algorithms. In this section, we present a time step study for the strong and weak Lagrange-Galerkin spectral element methods to see how they behave when we increase k, p, and Δt.

Figures 12–15 illustrate the results for the h- and p-type forms of the strong and weak methods for the grid $N_P = 60$ for various time step ratios σ for Case 1. For all of the methods presented herein, the order k represents a k^{th}-order Runge-Kutta method for the trajectory equation (see [15] for details on this scheme). In Figure 12, we see that the h-type strong Lagrange-Galerkin method does not benefit very much from an increase in k; note that the $k = 4$ and $k = 8$ curves are directly on top of each other, thus, illustrating that increasing k beyond 4 has no effect on the accuracy. Since in the h-type method $p = 1$, then obviously the second term in (23) will dominate regardless of the order of k. In contrast we see in Figure 13 that the p-type method clearly benefits from using a high-order k. At about $\sigma = 2$ the $k = 4$ and $k = 8$ schemes diverge from the $k = 2$ scheme which begins to lose accuracy. The $k = 4$ and $k = 8$ schemes give equivalent results until about $\sigma = 11$ at which point the $k = 4$ error begins to increase while the $k = 8$ error remains level. Because the order of p is sufficiently high, all of the error is dominated by the first term in (23). Since $p = 6$ is greater than $k = 4$, at some time step the Δt^k term will dominate. Thus, by using the $k = 8$ scheme we can delay the onset of this error increase. In fact, for the $k = 8$ scheme the error will eventually arise through the second term (because $k > p$).

Clearly, our results show that the strong Lagrange-Galerkin method is governed by the theory presented in [24]. A recent paper by Xiu and Karniadakis [25] shows very detailed results illustrating this phenomenon of the strong method for the advection-diffusion and Navier-Stokes equations. However, the question that we would like to address in our paper is whether the weak method behaves similarly to the strong method. Figure 14 shows that the $k = 4$ scheme for the weak method is sufficient to yield the maximum accuracy; the $k = 8$ scheme offers no additional accuracy which we can see from the figure where the $k = 4$ and $k = 8$ curves are directly on top of each other. The $k = 4$ and $k = 8$ schemes diverge from the $k = 2$ scheme at around $\sigma = 5$. However, the general trend of all three k schemes is an increasing error with time step. Figure 15 again shows the same kind of behavior for the p-type method although the method yields much more accurate results than the h-type method; the difference is almost two orders of magnitude. However, as we increase the time step both the h- and p-type methods lose their accuracy rather quickly. As in the strong method, the weak method benefits greatly from having a scheme greater than second order ($k = 2$). Therefore, it can be concluded that if a second-order scheme is used for the trajectory equation of either the Lagrange-Galerkin or semi-Lagrangian methods, then we will not extract the maximum level of accuracy from these schemes. The maximum accuracy can only be obtained with $k \geq 4$. In fact, we can obtain the optimal time step for a given Δx in closed form. From (23), we can show [26] that the optimal time step occurs when the trajectory

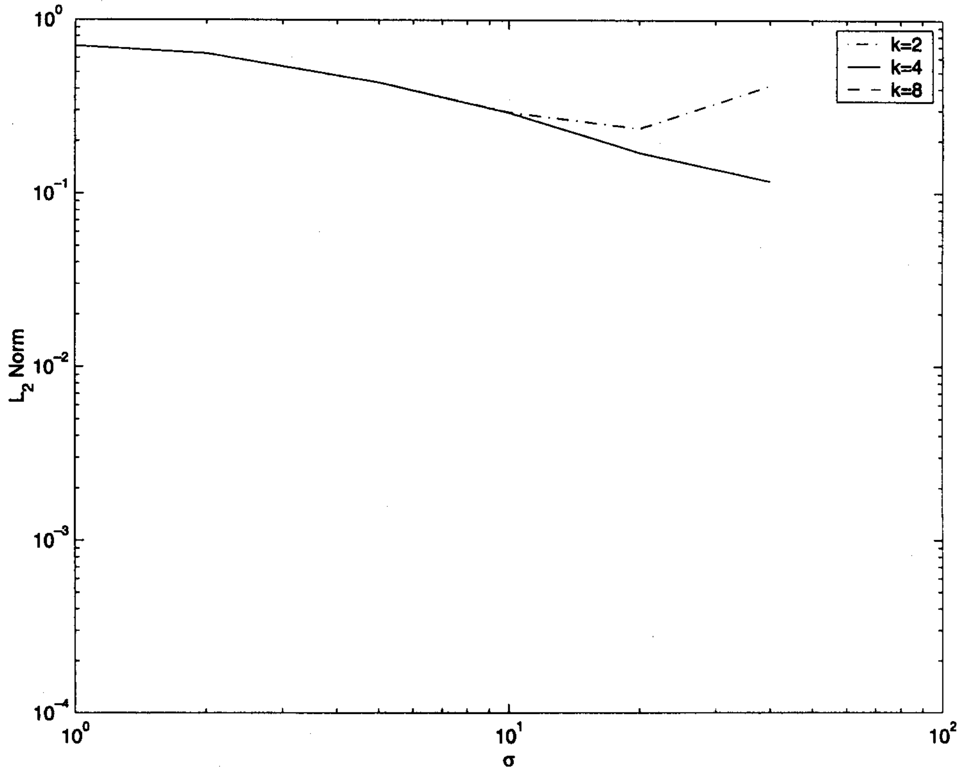

Figure 12. Case 1. The φ L_2 norm as a function of σ for the h-type form of the strong Lagrange-Galerkin spectral element methods with $N_P = 60$.

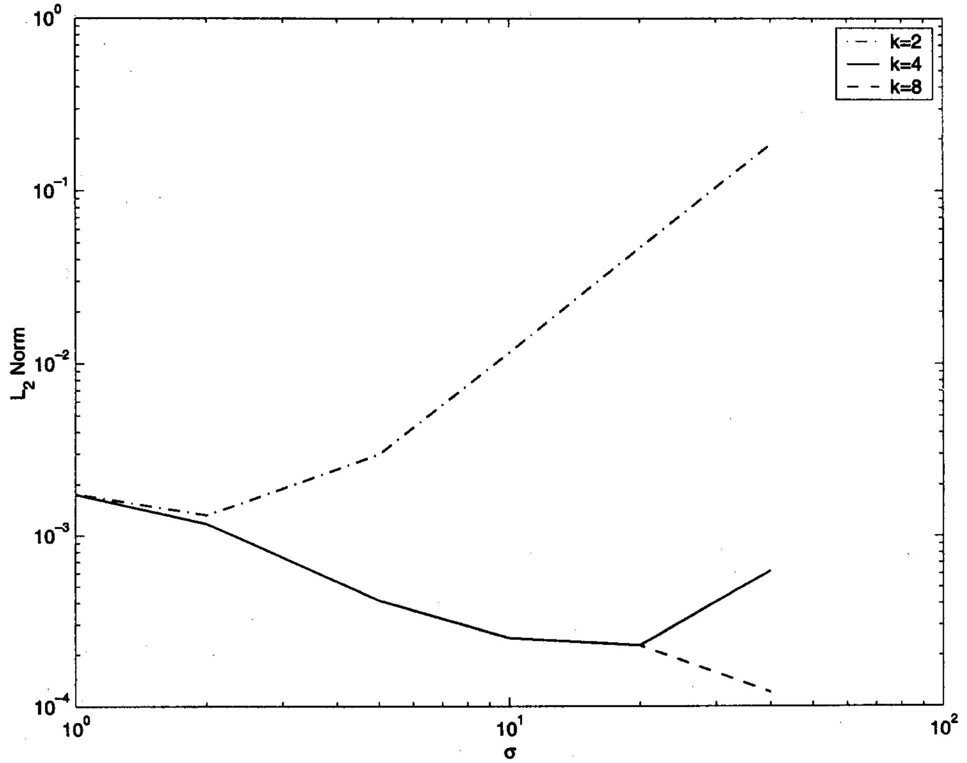

Figure 13. Case 1. The φ L_2 norm as a function of σ for the p-type form of the strong Lagrange-Galerkin spectral element methods with $N_P = 60$.

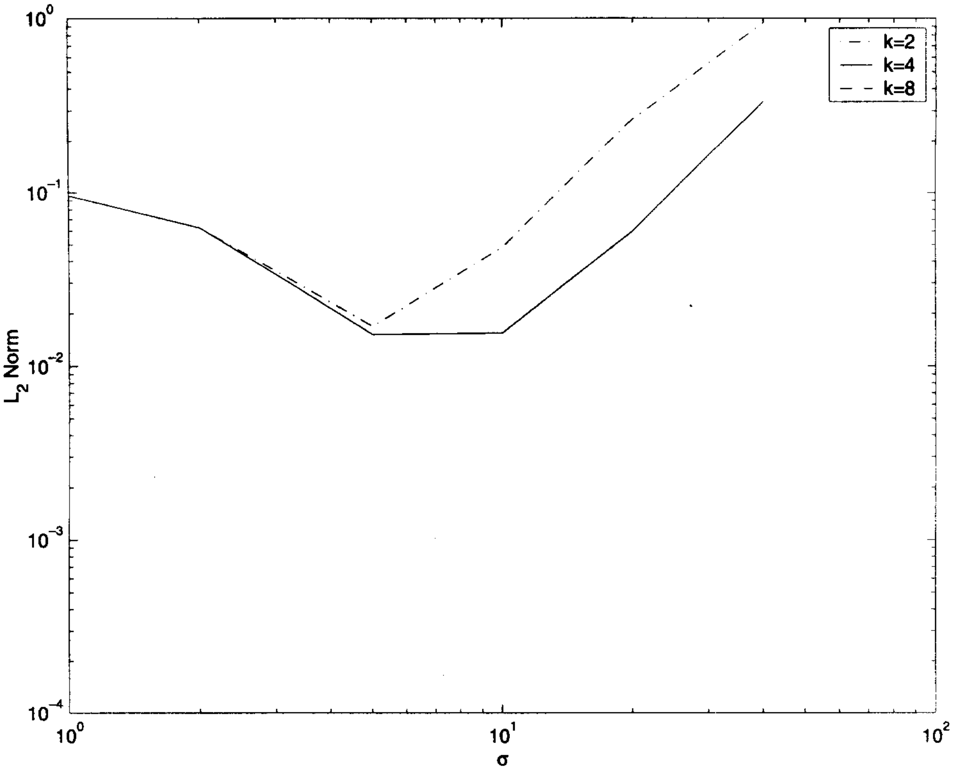

Figure 14. Case 1. The φ L_2 norm as a function of σ for the h-type form of the weak Lagrange-Galerkin spectral element methods with $N_P = 60$.

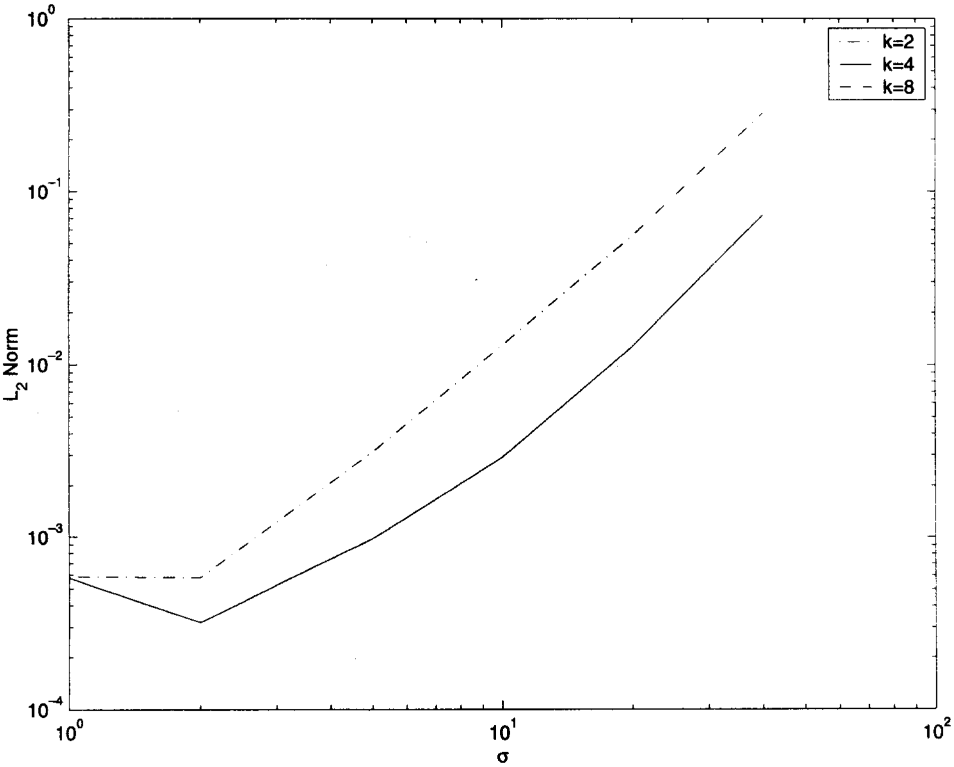

Figure 15. Case 1. The φ L_2 norm as a function of σ for the p-type form of the weak Lagrange-Galerkin spectral element methods with $N_P = 60$.

and interpolation errors are equivalent. Letting

$$\Delta t = \Delta x^{\alpha},$$

we can equate the first and second terms in (23) to obtain

$$\Delta t = \Delta x^{(p+1)/(k+1)}. \tag{24}$$

From this equation we can see that for a given Δx we can only use a very large time step if and only if $k > p$. The h-type form of the weak Lagrange-Galerkin method can actually be shown to be equivalent to the strong form with $p = 3$ [13] so that it makes sense to use $k = 4$ when using this scheme.

This time step study has shown that the weak method in either the h- or p-type forms does not benefit from using a high-order k (greater than 4) trajectory integration scheme. Clearly, it is better to use $k > 2$ (at least 4) but the error does not remain level for increasing time step as in the strong method. Based on this result alone it can be concluded that the strong method is the better candidate for use with the spectral element method.

9. CONCLUSIONS

Two Lagrange-Galerkin spectral element methods for the two-dimensional shallow water equations are compared. These methods are the strong and weak forms of the Lagrange-Galerkin method. The spectral element method presented is a high-order finite-element method that uses Legendre cardinal functions as its basis functions. The strong Lagrange-Galerkin spectral element method has been shown by the author [3] to achieve spectral (exponential) convergence as the order of the Legendre cardinal functions p is increased. In addition, the papers [24,25] have shown that this method with high-order p can use extremely large time steps (Courant numbers greater than 20) without losing accuracy as long as a sufficiently large k is used for the trajectory equation. Our results have confirmed this; in fact, we were able to use time steps 40 times larger than the explicit Eulerian scheme without any loss in accuracy (see Figure 13 for $k = 8$). The strong Lagrange-Galerkin method achieves its order of accuracy from the order of interpolation and so its accuracy increases with p, whereas, since the weak method achieves its accuracy from the order of integration, this method does not benefit as much from a high-order p. Both methods gave very good results, so it is very difficult to ascertain which of these two methods will work best with the spectral element method. For pure advection, the weak method performed better than the strong method for $p \leq 7$; for $p > 7$, the strong method prevailed. For the soliton test case, both methods yielded more,or less similar results. However, because efficiency is one of the major goals of all numerical models, it appears that the strong method is the better candidate for use with the spectral element method since the error does not increase with increasing time step provided that a sufficiently large k is used for the trajectory equation. However, for problems that have a physical limitation on the maximum allowable time step, the choice of methods is then governed by what order p one can use. For lower-order p the weak method seems to be superior but for high-order p (greater than or equal to 8) the strong method is superior. Numerical weather prediction models are a good example of problems that contain physical limitations. Ideally, one would like to use a larger time step than allowed by Eulerian schemes, but the time step cannot be so large that it exceeds the time scales of some physically limiting processes such as cloud and thunderstorm formations. These phenomena are confined to time scales on the order of minutes to an hour [27]. Thus, one cannot use time steps in the scale of days when considering these processes unless of course we are only interested in performing climate studies in the time frame of thousands of years.

Currently, we are testing both the strong and weak Lagrange-Galerkin spectral element methods for the shallow water equations on the sphere. The true test of the strong and weak Lagrange-Galerkin spectral element methods is how they will perform for the shallow water equations on the sphere. We hope to decide at this point which method will be selected for the development of a 3D atmospheric model for numerical weather prediction.

REFERENCES

1. M. Iskandarani, D.B. Haidvogel and J.P. Boyd, A staggered spectral element model with application to the oceanic shallow water equations, *International Journal for Numerical Methods in Fluids* **20**, 393–414, (1995).
2. H. Ma, A spectral element basin model for the shallow water equations, *Journal of Computational Physics* **109**, 133–149, (1993).
3. F.X. Giraldo, The Lagrange-Galerkin spectral element method on unstructured quadrilateral grids, *Journal of Computational Physics* **147**, 114–146, (1998).
4. M. Bercovier and O. Pironneau, Characteristics and the finite element method, In *Proceedings of the Fourth International Symposium on Finite Element Methods in Flow Problems*, (Edited by T. Kawai), p. 67, North-Holland, New York, (1982).
5. R. Bermejo, A Galerkin-characteristic algorithm for transport-diffusion equations, *SIAM Journal on Numerical Analysis* **32**, 425–454, (1995).
6. J. Douglas and T.F. Russell, Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures, *SIAM Journal on Numerical Analysis* **19**, 871–885, (1982).
7. A. Priestley, Exact projections and the Lagrange-Galerkin method: A realistic alternative to quadrature. *Journal of Computational Physics* **112**, 316–333, (1994).
8. A. Staniforth and J. Côté, Semi-Lagrangian integration schemes for atmospheric models—A review. *Monthly Weather Review* **119**, 2206–2223, (1991).
9. O.C. Zienkiewicz and P. Ortiz, A split-characteristic based finite element model for the shallow water equations, *International Journal for Numerical Methods in Fluids* **20**, 1061–1080, (1995).
10. J. Donéa, A Taylor-Galerkin method for convective transport problems, *International Journal for Numerical Methods in Engineering* **20**, 101–118, (1984).
11. R. Lohner, K. Morgan and O.C. Zienkiewicz, The solution of non-linear hyperbolic equation systems by the finite element method, *International Journal for Numerical Methods in Fluids* **4**, 1043–1063, (1984).
12. I.G. Currie, *Fundamental Mechanics of Fluids*, p. 9, McGraw-Hill, San Francisco, CA, (1974).
13. J.P. Benque and J. Ronat, Quelques difficultes de modules numerique en hydralique, In *Computing Method in Applied Sciences and Engineering, Volume V*, (Edited by R. Glowinski and J.L. Lions), p. 471, North-Holland, New York, (1982).
14. J.P. Benque, G. Labadie and J. Ronat, A new finite element method for Navier-Stokes equations coupled with a temperature equation, In *Proceedings of the Fourth International Symposium on Finite Element Methods in Flow Problems*, (Edited by T. Kawai), p. 295, North-Holland, New York, (1982).
15. F.X. Giraldo, The Lagrange-Galerkin method for the 2D shallow water equations on adaptive grids, *International Journal for Numerical Methods in Fluids* **33**, 789–832, (2000).
16. J. Petera and V. Nassehi, A new two-dimensional finite element model for the shallow water equations using a Lagrangian framework constructed along fluid particle trajectories, *International Journal for Numerical Methods in Fluids* **39**, 4159–4182, (1996).
17. G. Chukapalli, Weather and climate numerical algorithms: A unified approach to an efficient, parallel implementation, Ph.D. Thesis, University of Toronto, (1997).
18. F.X. Giraldo, Lagrange-Galerkin methods on spherical geodesic grids: The shallow water equations, *Journal of Computational Physics* **160**, 336–368, (2000).
19. J.P. Boyd, Equatorial solitary waves. Part 1: Rossby solitons, *Journal of Physical Oceanography* **10**, 1699–1717, (1980).
20. J.P. Boyd, Equatorial solitary waves. Part 3: Westward-traveling modons, *Journal of Physical Oceanography* **15**, 46–54, (1985).
21. F.X. Giraldo, A spectral element shallow water model on spherical geodesic grids, *International Journal for Numerical Methods in Fluids* **35**, 869–901, (2001).
22. A. McDonald and J.R. Bates, Semi-Lagrangian integration of a gridpoint shallow water model on the sphere, *Monthly Weather Review* **117**, 130–137, (1989).
23. H. Ritchie, Semi-Lagrangian advection on a Gaussian grid, *Monthly Weather Review* **115**, 608–619, (1987).
24. M. Falcone and R. Ferretti, Convergence analysis for a class of high-order semi-Lagrangian advection schemes, *SIAM Journal on Numerical Analysis* **35**, 909–940, (1998).
25. D. Xiu and G.E. Karniadakis, A semi-Lagrangian high-order method for Navier-Stokes equations. *Journal of Computational Physics* (to appear).
26. A.V. Malevsky and S.J. Thomas, Parallel algorithms for semi-Lagrangian advection, *International Journal for Numerical Methods in Fluids* **25**, 455–473, (1997).
27. T. Rosmond, Naval Research Laboratory, Marine Meteorology Division, (private communication).