INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 20, 393–414 (1995)

A STAGGERED SPECTRAL ELEMENT MODEL WITH APPLICATION TO THE OCEANIC SHALLOW WATER EQUATIONS

MOHAMED ISKANDARANI AND DALE B. HAIDVOGEL

Institute of Marine and Coastal Sciences, Rutgers University, PO Box 231, New Brunswick, NJ 08903-0231, U.S.A.

AND

JOHN P. BOYD

Department of Atmospheric Oceanic and Space Sciences, University of Michigan, 2455 Hayward Avenue, Ann Arbor, MI 48109-2143, U.S.A.

SUMMARY

A staggered spectral element model for the solution of the oceanic shallow water equations is presented. We introduce and compare both an implicit and an explicit time integration scheme. The former splits the equations with the operator–integration factor method and solves the resulting algebraic system with generalized minimum residual (GMRES) iterations. Comparison of the two schemes shows the performance of the implicit scheme to lag that of the explicit scheme because of the unpreconditioned implementation of GMRES. The explicit code is successfully applied to various geophysical flows in idealized and realistic basins, notably to the wind-driven circulation in the North Atlantic Ocean. The last experiment reveals the geometric versatility of the spectral element method and the effectiveness of the staggering in eliminating spurious pressure modes when the flow is nearly non-divergent.

KEY WORDS shallow water equations; spectral element; implicit scheme; GMRES solver; staggered mesh; North Atlantic

1. INTRODUCTION

The simulation of large-scale ocean circulation raises particularly challenging issues in computational fluid dynamics. Perhaps most importantly, the geometrical complexity of the ocean basins demands algorithms that can faithfully represent the highly irregular coastline. These algorithms must be flexible enough to allow variable resolution so as to resolve effectively regions with important dynamics, e.g. the western boundary currents (a feature of Earth's rotation), the thermocline and the continental shelf, without wasting computational resources in dynamically less significant regions.

The geometric problem is further complicated by the non-linear nature of the free surface whose location is unknown *a priori* and must be computed as part of the solution process. In addition, the free surface gives rise to surface gravity waves whose speed of propagation may reach 200 m s^{-1} and which limit explicit models to extremely small time steps even though many phenomena of interest have much longer time scales. Implicit models can circumvent the stability restriction but have to contend with the solution of large systems of simultaneous algebraic equations; their computational advantage is therefore largely determined by the efficiency of their solver.

CCC 0271–2091/95/050393–22

Received 17 November 1993
Revised 6 July 1994

Large-scale ocean circulation models have traditionally relied on well-established finite difference techniques to discretize the equations of motion. Motivated by a desire for better accuracy and geometric flexibility, several models based on more novel numerical methods have emerged in recent years. For example, the two-dimensional model of Le Provost and Poncet[1] and the three-dimensional shallow water model of Lynch and Werner[2] employ low-order finite elements, the semi-spectral primitive equation model of Haidvogel *et al.*[3] discretizes the vertical direction spectrally with Chebyshev polynomials and the horizontal directions with centred finite differences and orthogonal curvilinear co-ordinates, Kelly[4] solves the reduced gravity shallow water equations with a fully spectral method and Ma[5] solves the same equations with a spectral element method.

The present work is part of an ongoing effort to build a spectral element model to solve the three-dimensional primitive equations governing large-scale ocean circulation. The spectral element (SE) method was developed in order to combine the geometrical flexibility of the traditional low-order finite element methods with the accuracy and high convergence rates of spectral methods.[6] It has been applied successfully in numerous engineering problems.[7,8] Ma[5] has recently adapted the SE method to geophysical flows and applied it successfully to the study of the non-linear reflection of Rossby solitons off a western coast. However, Ma's model uses a non-staggered grid unsuitable for non-divergent flows owing to the generation of spurious pressure modes.[9,10]

The numerical solution of the three-dimensional primitive equations poses numerous difficulties that are best resolved individually. A 'simplified' version of the primitive equations was needed to act as a test-bed for initial model development and evaluation. It was thus decided to start by developing a code to solve the oceanic shallow water equations, a depth-integrated, two-dimensional version of the original equations. The two-dimensional version inherits most of the numerical difficulties encountered in three dimensions, namely mesh generation, choice of interpolating polynomials, time integration, treatment of advection terms and non-linearities, treatment of the free surface gravity wave and finally solution of a system of equations if one is generated.

The paper is organized as follows. The spectral element formulation and discretization of the shallow water equations are described in Section 2. The discretization steps are given in some detail to show the staggering of the velocity and pressure nodes. The explicit time integration is outlined briefly in Section 3. An implicit treatment of the equations is described in Section 4, where the splitting technique and GMRES are introduced. Section 5 is devoted to the equatorial Rossby soliton problem which has been used as an initial verification of the model and a test to compare the implicit and explicit schemes. Finally, Section 6 illustrates the performance and versatility of the SE model for oceanic applications in rectangular and realistic ocean basin geometries.

2. FINITE ELEMENT FORMULATION

Let Ω be the two-dimensional region occupied by the fluid and let Γ denote its boundary. The shallow water equations in Ω are given by*

$$\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u} + f\hat{\mathbf{k}} \times \mathbf{u} + g\nabla\zeta + \gamma\mathbf{u} - \nu\nabla^2\mathbf{u} = \frac{\tau}{\rho(h+\zeta)}, \tag{1}$$

$$\zeta_t + \nabla \cdot [(h+\zeta)\mathbf{u}] = 0, \tag{2}$$

where $\mathbf{u} = (u, v)$ is the horizontal velocity vector, $\hat{\mathbf{k}}$ is a unit vector in the vertical direction, h is the resting depth of the fluid, ζ is the free surface elevation, f is the Coriolis parameter, g is the

* Gent[11] discusses the energetically consistent form of the shallow water equations, particularly when mass sources and sinks are present.

gravitational acceleration, γ is the bottom drag coefficient, ν is the diffusion coefficient, ρ is the density of the fluid, $\tau = (\tau^x, \tau^y)$ is the wind stress acting on the surface of the fluid and ∇ is the two-dimensional gradient operator. The subscript t denotes differentiation with respect to time. The boundary conditions are Dirichlet conditions on u, v and/or ζ:

$$u = u^b \quad \text{on} \quad \Gamma_D^u, \qquad v = v^b \quad \text{on} \quad \Gamma_D^v, \qquad \zeta = \zeta^b \quad \text{on} \quad \Gamma_D^\zeta; \tag{3}$$

and Neumann conditions on u and v:

$$\nu \nabla u \cdot \mathbf{n} = q^x \quad \text{on} \quad \Gamma_N^1, \qquad \nu \nabla v \cdot \mathbf{n} = q^y \quad \text{on} \quad \Gamma_N^2. \tag{4}$$

Here $\mathbf{n}$ is the unit normal to the boundary, q^x and q^y are the applied stresses in the x- and y-directions respectively (if free-slip BCs are applied on a rectangular domain, for example, $q^x = q^y = 0$) and u^b, v^b and ζ^b are the values of u, v and ζ imposed on specified segments of the boundary. If $\nu = 0$, only the normal component of the velocity at the boundary can be specified: $\mathbf{u} \cdot \mathbf{n} = \mathbf{u}^b \cdot \mathbf{n}$. Γ_D^u, Γ_D^v and Γ_D^ζ are those portions of the boundary where Dirichlet conditions on u, v and ζ are applied. In the case of no-slip conditions $\Gamma_D^u = \Gamma_D^v$. Γ_N^u and Γ_N^v refer to the boundary segments where Neumann conditions on u and v are applied; we must have $\Gamma_D^u \cap \Gamma_N^u = \emptyset$ and $\Gamma_D^v \cap \Gamma_N^v = \emptyset$. We refer the reader to Reference 12 for a discussion on the appropriate boundary conditions to apply for different forms of the shallow water equations.

When the surface displacement ζ is much smaller than the resting depth h, the flow is nearly incompressible. This can be easily seen by considering the dimensionless continuity equation

$$\nabla \cdot (h\mathbf{u}) + \frac{d}{H}\left[\zeta_t + \nabla \cdot (\zeta \mathbf{u})\right] = 0, \tag{5}$$

where we have scaled the free surface displacement ζ with d, the depth h with H, the velocity $\mathbf{u}$ with U, length with L and time with L/U. For oceanic flows typically $d \leqslant 10$ m and $H \geqslant 1000$ m, yielding a ratio $d/H \leqslant 0{\cdot}01$. The ratio d/H is even smaller in the deep ocean, where the depth can exceed 5000 m. In the incompressible limit the pressure interpolation must comply with the so-called Babuska–Brezzi or div-stability condition,[9,13,14] otherwise spurious pressure modes appear and can destroy the solution. Schumack *et al.*[14] describe several alternatives to circumvent the spurious modes. The one adopted here is to use a staggered mesh and interpolate the pressure with a polynomial of degree two less than that used for the velocity.[9]

The starting point of the spectral element model is the Galerkin formulation of the shallow water equations

$$\begin{aligned} \int_A u_t w^u \, \mathrm{d}A = &\int_A \left(-\left(uu_x + vu_y\right) - g\zeta_x + fv - \gamma u + \frac{\tau^x}{\rho(h+\zeta)}\right) w^u \, \mathrm{d}A \\ &- \int_A \nu \nabla u \cdot \nabla w^u \, \mathrm{d}A + \int_{\Gamma_N^u} q^x w^u \, \mathrm{d}S, \quad \forall w^u \in V^u, \end{aligned} \tag{6}$$

$$\begin{aligned} \int_A v_t w^v \, \mathrm{d}A = &\int_A \left(-\left(uv_x + vv_y\right) - g\zeta_y - fu - \gamma v + \frac{\tau^y}{\rho(h+\zeta)}\right) w^v \, \mathrm{d}A \\ &- \int_A \nu \nabla v \cdot \nabla w^v \, \mathrm{d}A + \int_{\Gamma_N^v} q^y w^v \, \mathrm{d}S, \quad \forall w^v \in V^v, \end{aligned} \tag{7}$$

$$\int_A \zeta_t w^p \, \mathrm{d}A = -\int_A \left\{[(h+\zeta)u]_x + [(h+\zeta)v]_y\right\} w^p \, \mathrm{d}A, \quad \forall w^p \in Z, \tag{8}$$

where V and Z are the Sobolev spaces defined by

$$V^i(\Omega) = \{v \in H^1(\Omega),\ v(\Gamma_D^i) = 0\}, \quad i = u,\ v,$$
$$Z(\Omega) = \left\{v \in H^1(\Omega),\ v\left(\Gamma_D^\zeta\right) = 0\right\}.$$

H^1 is the set of square integrable functions whose first derivatives are also square integrable; w^u, w^v and w^p are the weight functions associated with the x- and y-components of the velocity and the surface elevation respectively.

The spatial discretization proceeds by subdividing the domain into a set of conforming quadrilateral isoparametric elements. The elements may have curved sides in the physical domain; in the computational domain (ξ, η) they are individually mapped into the square $-1 \leqslant \xi \leqslant 1$, $-1 \leqslant \eta \leqslant 1$. On each element the variables u, v and ζ are interpolated as

$$\begin{aligned} u(\xi, \eta) &= \sum_{i=1}^{N^v} \sum_{j=1}^{N^v} u_{i,j} h_i^v(\xi) h_j^v(\eta), \\ v(\xi, \eta) &= \sum_{i=1}^{N^v} \sum_{j=1}^{N^v} v_{i,j} h_i^v(\xi) h_j^v(\eta), \\ \zeta(\xi, \eta) &= \sum_{i=1}^{N^p} \sum_{j=1}^{N^p} \zeta_{i,j} h_i^p(\xi) h_j^p(\eta), \end{aligned} \tag{9}$$

where (u_{ij}, v_{ij}) is the velocity vector at the velocity collocation nodes $(\xi_{ij}^v, \eta_{ij}^v)$, $(i, j) = 1, \ldots, N^v$, and ζ_{ij} is the surface elevation at the pressure collocation nodes $(\xi_{ij}^p, \eta_{ij}^p)$, $(i, j) = 1, \ldots, N^p$. The u_{ij}, v_{ij} and ζ_{ij} are functions of time only. N^v and N^p are the numbers of nodes per element in the ξ- and η-directions for the velocity and pressure interpolations respectively; therefore $N^p = N^v - 2$.

The interpolation functions h_i^v are the Legendre cardinal functions[15]

$$h_i^v(\xi) = \frac{-(1 - \xi^2) L'_{N^v - 1}(\xi)}{N^v (N^v - 1) L_{N^v - 1}(\xi_i^v)(\xi - \xi_i^v)}, \quad i = 1, 2, \ldots, N^v. \tag{10}$$

$L_{N^v - 1}$ denotes the Legendre polynomial of degree $N^v - 1$ and $L'_{N^v - 1}$ denotes its derivative. The ξ_i^v are the N^v Gauss–Lobatto nodes: they are the roots of $(1 - \xi^2) L'_{N^v - 1}(\xi)$. It is easy to show that $h_i^v(\xi_j^v) = \delta_{ij}$, where δ_{ij} is the Kronecker delta. The pressure interpolation functions h_i^p are defined similarly but with the superscript v replaced by p. Figure 1 shows a typical element and the lay-out of the velocity and pressure Gauss–Lobatto nodes. The Gauss–Lobatto nodes are not evenly spaced as sketched in the figure but are actually more crowded near the boundaries. The present staggered mesh, unlike the meshes of Ronquist[10] and Schumack *et al.*,[14] preserves the continuity of the pressure at the inter-element boundaries.

A system of ordinary differential equations (for u, v and ζ) is obtained after inserting (9) in (6)–(8) and substituting $h_i^v h_j^v$ for w^u and w^v, and $h_i^p h_j^p$ for w^p:

$$\begin{aligned} M^v \frac{\mathrm{d}u}{\mathrm{d}t} + (\gamma M^v + vD)u - Fv + gP^x \zeta + A^x &= r^x, \\ M^v \frac{\mathrm{d}v}{\mathrm{d}t} + Fu + (\gamma M^v + vD)v + gP^y \zeta + A^y &= r^y, \\ M^p \frac{\mathrm{d}\zeta}{\mathrm{d}t} + C^x u + C^y v + A^z &= 0, \end{aligned} \tag{11}$$

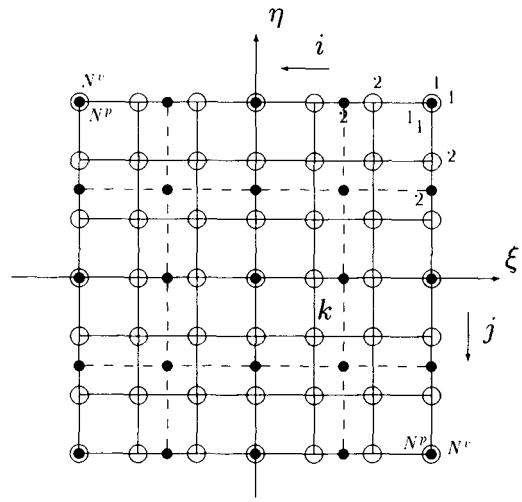

Figure 1. Isoparametric element in computational plane. The Gauss–Lobatto points of the velocity nodes are shown by open circles and the pressure nodes by full circles. The local node numbers of the velocity nodes are shown outside the element, while those of the pressure nodes are shown inside. $N^p = N^v - 2$

where the matrices are defined as

$$
\begin{aligned}
M^v_{ij,\,kl} &= \int_A h^v_i(\xi)h^v_j(\eta)h^v_k(\xi)h^v_l(\eta)\,\mathrm{d}A,\\
M^p_{ij,\,kl} &= \int_A h^p_i(\xi)h^p_j(\eta)h^p_k(\xi)h^p_l(\eta)\,\mathrm{d}A,\\
F_{ij,\,kl} &= \int_A fh^v_i(\xi)h^v_j(\eta)h^v_k(\xi)h^v_l(\eta)\,\mathrm{d}A,\\
D_{ij,\,kl} &= \int_A \nabla\left[h^v_i(\xi)h^v_j(\eta)\right]\cdot\nabla\left[h^v_k(\xi)h^v_l(\eta)\right]\mathrm{d}A,\\
P^s_{ij,\,kl} &= \int_A h^v_i(\xi)h^v_j(\eta)\,\frac{\partial h^p_k(\xi)h^p_l(\eta)}{\partial s}\,\mathrm{d}A, \quad s = x,\ y,\\
C^s_{ij,\,kl} &= \int_A h^p_i(\xi)h^p_j(\eta)\,\frac{\partial hh^v_k(\xi)h^v_l(\eta)}{\partial s}\,\mathrm{d}A, \quad s = x,\ y,\\
r^s_{ij} &= \int_A \frac{\tau^s}{\rho(h+\zeta)}\,h^v_i(\xi)h^v_j(\eta)\,\mathrm{d}A + \int_{\Gamma^1_N} h^v_i(\xi)h^v_j(\eta)q^s\,\mathrm{d}S, \quad s = x,\ y,\\
A^x_{ij} &= \int_A \left(uu_x + vu_y\right)h^v_i(\xi)h^v_j(\eta)\,\mathrm{d}A,\\
A^y_{ij} &= \int_A \left(uv_x + vv_y\right)h^v_i(\xi)h^v_j(\eta)\,\mathrm{d}A,\\
A^z_{ij} &= \int_A \left[(\zeta u)_x + (\zeta v)_y\right]h^p_i(\xi)h^p_j(\eta)\,\mathrm{d}A.
\end{aligned}
\tag{12}
$$

Equations (11) hold at the elemental level. The assembly procedure adds the contributions of the different elements to the system of equations. The non-linear terms are grouped in A^x, A^y and A^z in anticipation of the splitting scheme; the rest of the terms represent the action of linear operators only. The matrices M^v and M^p are the mass matrices associated with the velocity and pressure interpolation

functions respectively. The mass matrices are positive definite, symmetric and densely populated if the integrals are evaluated analytically; they are banded with a large bandwidth at the global level.

Note that even an explicit time-integration scheme requires the inversion of the matrices M^v and M^p. A judicious choice of the interpolation polynomial and the integration rule can reduce substantially the amount of memory and CPU time needed to store and invert the system of equations. This choice is Legendre cardinal functions for the interpolation and Gauss–Lobatto quadrature for the integration (exact for polynomials of order $2N^v - 3$). The combination of Lagrangian interpolation and numerical quadrature yields a diagonal mass matrix. Tremendous savings in computations and storage can thus be achieved for a negligible loss in accuracy (see e.g. References 10, 16 and 17). To show how the diagonal mass matrix is obtained, we derive the expression for M^v as

$$\begin{aligned} M^v_{ij,\,kl} &= \int_{-1}^{1}\int_{-1}^{1} |J| h_i^v(\xi) h_j^v(\eta) h_k^v(\xi) h_l^v(\eta)\, \mathrm{d}\xi\, \mathrm{d}\eta \\ &= \sum_{p=1}^{N^v}\sum_{q=1}^{N^v} |J_{pq}| w_p^v w_q^v h_i^v(\xi_p^v) h_j^v(\eta_q^v) h_k^v(\xi_p^v) h_l^v(\eta_q^v) \\ &= \delta_{ik}\delta_{jl}|J_{kl}| w_k^v w_l^v, \end{aligned}$$

where $J_{kl} = (x_\xi y_\eta - x_\eta y_\xi)_{kl}$ is the Jacobian of the mapping from the physical space to the computational space evaluated at the node (ξ_k^v, η_l^v), and π_p^v is the pth weight of the Gauss–Lobatto quadrature ($p = 1, \ldots, N^v$). We have used the identity $h_i^v(\xi_k^v) = \delta_{ik}$ to eliminate the sums on p and q and to obtain the final diagonal form. Since the elemental mass matrix is diagonal, so is the global mass matrix. Hence the global mass matrix can be stored in a one-dimensional array and its inverse is just the inverse of the diagonal entries. The same procedure applies to the pressure mass matrix M^p with a Gauss–Lobatto quadrature of order N^p.

The drawback of a staggered mesh is the need to move information between the pressure and velocity grids. This exchange is necessary to evaluate those integrals in (12) that involve variables defined on both grids, namely P^x, P^y, C^x, C^y and A^z. The interpolation between the two grids involves matrix–matrix products of the form

$$\zeta\left(\xi_i^v, \eta_j^v\right) = \sum_{k=1}^{N^p}\sum_{l=1}^{N^p} h_k^p\left(\xi_i^v\right) h_l^p\left(\eta_j^v\right) \zeta_{k,l}. \tag{13}$$

The number of floating point operations required to perform the above interpolation is rather costly (compared with finite difference methods) and is approximately, after taking advantage of the tensor product nature of the multiplications, $2N^vN^p(N^v + N^p)$ per element. Cross-grid interpolation must be avoided as much as possible. Our current implementation interpolates ζ and its gradient onto the velocity grid. The gradient calculations are done in one step, i.e. immediately from the pressure grid to the velocity grid, e.g.

$$\left.\frac{\partial \zeta}{\partial \xi}\right|_{\xi_i^v,\,\eta_j^v} = \sum_{k=1}^{N^p}\sum_{l=1}^{N^p} \left.\frac{\partial h_k^p}{\partial \xi}\right|_{\xi_i^v} h_l^p\left(\eta_j^v\right) \zeta_{k,l}. \tag{14}$$

The above calculation is slightly more expensive than calculating ζ_ξ on the pressure grid:

$$\left.\frac{\partial \zeta}{\partial \xi}\right|_{\xi_i^p,\,\eta_j^p} = \sum_{k=1}^{N^p} \left.\frac{\partial h_k^p}{\partial \xi}\right|_{\xi_i^p} \zeta_{k,j}. \tag{15}$$

Note that the gradient of ζ on the pressure grid is not needed. We should also point out that the order

N^p quadrature associated with the pressure grid is used in the calculations of M^p only. All other integrals are approximated with the higher-order quadrature of the velocity grid (order N^v).

3. EXPLICIT TIME INTEGRATION

The explicit time integration of equations (11) is performed with a third-order Adams–Bashforth (AB3) scheme which we outline briefly. Each of the equations in (11) can be written in the generic form $M\,\mathrm{d}u/\mathrm{d}t = r$, where u and r are the vector of unknowns and the vector of right-hand sides respectively and M is one of the mass matrices. The AB3 scheme takes the form[18]

$$u^{n+1} = u^n + \Delta t M^{-1}\left(\frac{23}{12}r^n - \frac{16}{12}r^{n-1} + \frac{5}{12}r^{n-2}\right). \tag{16}$$

The calculations require information at two previous time levels and thus a start-up method is needed at the initial time step; we choose a fourth-order Runge–Kutta scheme. Notice that there is no need to compute or assemble explicitly the matrices in equation (11), since u, v and ζ are known at the previous time levels. Instead, the integrals in (12) are computed individually (e.g. the pressure gradient terms, the Coriolis force, the advective acceleration) and then summed up and assembled into the right-hand-side vector r. The vector r is the only quantity, besides the inverse of the mass matrices, that needs to be assembled.

The stability of the explicit scheme is determined by the smallest of the stability limits of the advection operator, Δt_a, and diffusion operator, Δt_d. Ma[5] gives these stability limits as $\Delta t_a \leqslant \alpha_1 L/UN^2K$ and $\Delta t_\mathrm{d} \leqslant \alpha_2 L^2/\nu N^4 K^2$, where U is the speed of the fastest-moving wave in the problem (in oceanic flows this is the free surface gravity wave $U = \sqrt{(gh)}$, L is the length scale, ν is the kinematic viscosity, N is the number of polynomials in the interpolation, K is the number of elements and α_a and α_d are proportionality constants. The ratio $\Delta t_\mathrm{d}/\Delta t_\mathrm{a}$ is proportional to Re/N^2K, where $Re = UL/\nu$ is the Reynolds number. Hence, as long as the Reynolds number is larger than N^2K, as is the case in most ocean flows, the advection operator determines the largest allowable time step. Typical values of $U = 200$ m s^{-1}, $L = 1000$ km, $\nu = 2000$ m^2 s^{-1}, $N = 12$ and $K = 30$ yield a ratio of $10^5/4000 = 20$. The time step for a grid with medium to high resolution varies from 2 to 6 min when the model is run in reduced gravity mode (surface wave speed of about 2–5 m s^{-1}; Reference 5 and present work) and from 5 to 15 s for a full gravity model with surface wave speeds between 100 and 200 m s^{-1} (present simulations). Given that the time scales of interest are much longer than those associated with the stability limits, it seems worthwhile to look for an implicit scheme that relaxes the severe stability restriction imposed by the free surface gravity wave.

4. IMPLICIT TIME INTEGRATION SCHEME

An ideal implicit integration scheme would have two main properties: (1) unconditional stability and (2) a fast and efficient means of solving the resulting system of equations. We have tried to reach these two goals by splitting the equations into their linear and non-linear parts. The splitting is carried out with the operator–integration factor method of Maday *et al.*[19] The linear terms are integrated implicitly in time with a second-order backward differentiation scheme, while the non-linear terms are treated explicitly with a second-order Runge–Kutta scheme. The implicit integration generates a system of algebraic equations that is solved iteratively at each time step with the GMRES method of Saad and Schultz.[20]

4.1. *Operator splitting*

A straightforward explicit treatment of the non-linear terms might mitigate the advantages of an implicit formulation, since the stability limit, as set by the non-linear advection operator, may force the implicit solver to be called upon more often than desired. This is a serious drawback when the equations to be solved are stiff and/or the solver is not very efficient. Semi-Lagrangian schemes[21] have proven very successful at relaxing the advective stability limit and increasing the length of the integration time step. In the present work we adopt the operator–integration factor method of Maday *et al.*[19] to separate the integration of the linear terms from that of the non-linear terms. The former are integrated implicitly, while the latter are integrated explicitly. The conditional stability of the explicit scheme does not affect the stability of the implicit part, because the two integrations are uncoupled. The distinct advantage of the operator–integration factor method is that it can consistently generate high-order time-splitting schemes for a variety of operators. (Maday *et al.*, for example, apply it to decouple the pressure–velocity integration of the Stokes problem.) In the case of the advection operator Maday *et al.* show that the splitting is tantamount to computing an approximation to the material derivative and that the splitting can be interpreted as a characteristic/Lagrangian scheme.[19,21,22] A brief description of the splitting procedure is given in the Appendix. Here we give the equations in their split form.

The time discretization of the linear operators (the **B**-operator in the Appendix) with a second-order backward difference scheme yields

$$\frac{a_0}{\Delta t}u^{n+1} - fv^{n+1} + g\zeta_x^{n+1} + \gamma u^{n+1} - \nu\nabla^2 u^{n+1} = \frac{a_1}{\Delta t}\tilde{u}^{n,M} - \frac{a_2}{\Delta t}\tilde{u}^{n-1,2M} + \frac{(\tau^x)^{n+1}}{\rho(h+\zeta^n)}, \tag{17}$$

$$\frac{a_0}{\Delta t}v^{n+1} - fu^{n+1} + g\zeta_y^{n+1} + \gamma v^{n+1} - \nu\nabla^2 v^{n+1} = \frac{a_1}{\Delta t}\tilde{v}^{n,M} - \frac{a_2}{\Delta t}\tilde{v}^{n-1,2M} + \frac{(\tau^y)^{n+1}}{\rho(h+\zeta^n)}, \tag{18}$$

$$\frac{a_0}{\Delta t}\zeta^{n+1} + \left(hu^{n+1}\right)_x + \left(hv^{n+1}\right)_y = \frac{a_1}{\Delta t}\tilde{\zeta}^{n,M} - \frac{a_2}{\Delta t}\tilde{\zeta}^{n-1,2M}. \tag{19}$$

If a_0, a_1 and a_2 are equal to 1, 1 and 0 respectively, we have the first-order backward difference scheme, while if they are equal to $\frac{3}{2}$, 2 and $\frac{1}{2}$ respectively, we have the second-order backward difference scheme. The $\tilde{\mathbf{u}}^{n-q+1,qM}$ and $\tilde{\zeta}^{n-q+1,qM}$, where q equals 1 or 2, are the velocity vector and surface elevation respectively computed from the auxiliary problem (the **A**-operator of the Appendix); we will give the auxiliary problem shortly. The wind stress term is linearized by evaluating ζ in the denominator at the previous time level. The neglected terms are $O(\Delta t\zeta_t/(h+\zeta))$, which are quite small in the deep ocean. Note that the wind stress is identically zero in our Rossby soliton test problem and hence the method is second-order in time. The spatial discretization of (17)–(19) proceeds as in Section 2 and the final system of equations becomes

$$\left[\left(\frac{a_0}{\Delta t}+\gamma\right)M^v + \nu D\right]u^{n+1} - Fv^{n+1} + gP^x\zeta^{n+1} = R^x, \tag{20}$$

$$Fu^{n+1} + \left[\left(\frac{a_0}{\Delta t}+\gamma\right)M^v + \nu D\right]v^{n+1} + gP^y\zeta^{n+1} = R^y, \tag{21}$$

$$C^x u^{n+1} + C^y v^{n+1} + \frac{a_0}{\Delta t}M^p\zeta^{n+1} = R^z, \tag{22}$$

$$R^x = M^v\left(\frac{a_1}{\Delta t}\tilde{u}^{n,M} - \frac{a_2}{\Delta t}\tilde{u}^{n-1,2M} + \frac{(\tau^x)^{n+1}}{\rho(h+\zeta^n)}\right), \tag{23}$$

$$R^v = M^v\left(\frac{a_1}{\Delta t}\,\tilde{v}^{n,M} - \frac{a_2}{\Delta t}\,\tilde{v}^{n-1,2M} + \frac{(\tau^v)^{n+1}}{\rho(h+\zeta^n)}\right), \tag{24}$$

$$R^z = M^p\left(\frac{a_1}{\Delta t}\,\tilde{\zeta}^{n,M} - \frac{a_2}{\Delta t}\,\tilde{\zeta}^{n-1,2M}\right). \tag{25}$$

The variables with the tilde on the right-hand side of the above equations are computed by integrating the auxiliary problem (the **A**-operator in the appendix)

$$\frac{\partial \tilde{u}^{n-q+1}}{\partial s} = -\tilde{u}^{n-q+1}\,\frac{\partial \tilde{u}^{n-q+1}}{\partial x} - \tilde{v}^{n-q+1}\,\frac{\partial \tilde{u}^{n-q+1}}{\partial y}, \tag{26}$$

$$\frac{\partial \tilde{v}^{n-q+1}}{\partial s} = -\tilde{u}^{n-q+1}\,\frac{\partial \tilde{v}^{n-q+1}}{\partial x} - \tilde{v}^{n-q+1}\,\frac{\partial \tilde{v}^{n-q+1}}{\partial y}, \tag{27}$$

$$\frac{\partial \tilde{\zeta}^{n-q+1}}{\partial s} = -\frac{\partial \tilde{\zeta}^{n-q+1}\tilde{u}^{n-q+1}}{\partial x} - \frac{\partial \tilde{\zeta}^{n-q+1}\tilde{v}^{n-q+1}}{\partial y}, \tag{28}$$

$$\tilde{u}^{n-q+1}(s=0) = u^{n-q+1}, \tag{29}$$

$$\tilde{v}^{n-q+1}(s=0) = v^{n-q+1}, \tag{30}$$

$$\tilde{\zeta}^{n-q+1}(s=0) = \zeta^{n-q+1} \tag{31}$$

between $s = 0$ and $s = t^{n+1} - t^{n-q+1}$. The integration is done explicitly with a second-order Runge–Kutta scheme. The discretization and integration of (26)–(28) are similar to the explicit procedure of Section 2 and will not be repeated here. The integration is performed twice for a second-order scheme: once for $q = 1$ and once for $q = 2$. The accuracy can easily be made third-order at the expense of integrating (26)–(28) a third time with $q = 3$; the order of the integration in the auxiliary problem would also have to be increased, by moving to a third- or fourth-order Runge–Kutta scheme for example. Our experience has been that most of the CPU time is spent in the iterative solver and hence increasing the order of the time differencing would not greatly increase the CPU time.

4.2. GMRES and FEM

The implicit scheme yields the system of equations (17)–(19) that needs to be solved at each time step for the unknowns u^{n+1}, v^{n+1} and ζ^{n+1}. This system is non-symmetric, indefinite and its bandwidth is large compared with the bandwidth of algebraic systems arising from low-order finite element methods. The memory required to solve large multidimensional problems precludes the use of direct solvers. An efficient iterative solver is clearly required. Few iterative solvers are well suited to non-symmetric and indefinite systems and ever fewer prove to be effective on general systems of equations. The solver we have tested is the GMRES method of Saad and Schultz.[20,23]

The main feature that makes GMRES attractive in the context of the FEM is that the global stiffness matrix need not be assembled. The algorithm requires the formation of matrix–vector products which can be performed at the elemental level and only the resulting vectors need to be assembled. Moreover, there is no need to store any elemental matrices, since the integrals in (12) can be recalculated cheaply at each iteration. In practice it is useful to store the mass matrices, since they are needed frequently in the course of the computations. It is also useful to assemble the global mass matrices and their inverses. The assembly, storage and inversion of the mass matrices are very efficient and simple, since these

matrices are diagonal. The sequence of operations required for the GMRES algorithm is very similar to the sequence of an explicit time-stepping procedure.

Table I illustrates the storage that different implementations of the implicit spectral element method require. The first implementation used Chebyshev cardinal functions and a banded LU solver, the second one substitutes GMRES for LU and the last one abandons the Chebyshev functions for the Legendre functions and their concomitant Gauss–Lobatto quadrature. The data are taken from an early run of the equatorial Rossby soliton experiment to be described more fully below. The basin is rectangular and the mesh consists of eight elements zonally and four meridionally; each element has 81 interpolation nodes and the total is 2145 (6438 degrees of freedom). The local matrix equations are stored for the GMRES solver and the Krylov subspace has dimension five. Most of the memory in the LU solver is wasted on storing the zero entries. The Legendre GMRES implementation requires the least amount of storage, because the numerical quadrature yields sparse elemental matrices that consume five times less storage than a comparable Chebyshev implementation.

5. EQUATORIAL ROSSBY SOLITON

We have reproduced the propagation of the equatorial Rossby soliton of Boyd[24] with our staggered SE model. The shallow water equations were solved in their nondimensional form. The time (T), length (L) and velocity (U) were scaled with the Lamb parameter E:[24]

$$T = \frac{E^{1/4}}{2\Omega}, \qquad L = \frac{a}{E^{1/4}}, \qquad U = \sqrt{(gh)}, \qquad E = \frac{4\Omega^2 a^2}{gh},$$

where a is the radius of Earth and Ω is its angular speed of rotation. The depth was chosen to be 41 cm so that the surface gravity wave speed was 2 m s^{-1} (the wave speed of the first baroclinic mode). These values yield a time scale of 41 h and a length scale of 295 km. Harmonic and bottom frictions were set to zero and the wind stress was turned off. The asymptotic solution predicts a balance between dispersive and non-linear effects that allows the soliton to preserve its shape while travelling westward at a constant phase speed. The numerical solution reproduced this behaviour quite faithfully. Figure 2 shows the surface displacement at the beginning and end of the simulation. The discrepancies between our solution and Boyd's asymptotic solution are negligible and are due to the asymptotic nature of Boyd's solution. The SE model gives a phase speed of 0·76 m s^{-1}, while the asymptotic solution predicts a value of 0·78 m s^{-1}.

We have used the equatorial Rossby soliton problem to compare the performances of the implicit and explicit schemes. For the implicit scheme the GMRES solver requires the storage of the basis vectors of the Krylov space. Each vector has the size of the number of degrees of freedom in the

Table I. Storage in megabytes required by the implicit version

	Solver		
	Banded LU	GMRES	GMRES
Cardinal functions	Chebyshev	Chebyshev	Legendre
Integration	Analytical	Analytical	GL quadrature
Memory (MB)	81·23	12·37	2·41

problem. Hence storage increases linearly with an increase in the dimension of the Krylov space, M. In general a larger M means a higher convergence rate and hence fewer iterations to achieve convergence. However, the operation count per iteration increases and so does the CPU time. Storage was not a problem in the present test, for larger problems, however, M should be kept as small as possible.

The efficiencies of the implicit and explicit schemes were compared based on the CPU time required to integrate the shallow water equations up to a fixed time $T = 17{\cdot}82$ non-dimensional time units. The CPU time was recorded at the end of each run, as were the average number of iterations required for the solver to converge and the maximum error in the solution. The errors were computed based on a reference calculation with the explicit code using a time step 10 times smaller than the stability limit. The latter was determined experimentally to be about $\Delta t_e = 0{\cdot}033$ non-dimensional time unit. The CPU time and error of the explicit code were obtained by running the explicit code with the time step set to the stability limit. Note that the AB3 scheme produces small time discretization errors, because the time step has to be small and the scheme is third-order-accurate. The CPU time of the explicit code was $CPU_e = 30{\cdot}28$ s as measured on an IBM RS/6000 model 550 (all runs were performed on this platform).

The question we ask is: is there a range of parameters where the implicit scheme would be more cost-effective than the explicit scheme? Three series of implicit runs were performed in order to tune the different parameters of the implicit solver, namely the dimension of the Krylov space, M, the convergence criterion of the GMRES solver, ε, and the optimal time step Δt_i. The first series was meant to determine M: ε and Δt_i were kept constant at 10^{-6} and $5\Delta t_e$ respectively, while M was increased from 5 to 50. Figure 3 shows the ratio CPU_i/CPU_e and the average number of iterations, i_c. The number of iterations decreases steadily and the dimension of the Krylov space is increased, but the curve flattens out considerably beyond $M = 20$. The CPU curve shows the same behaviour and so we have set M to 20 for the remainder of the computations.

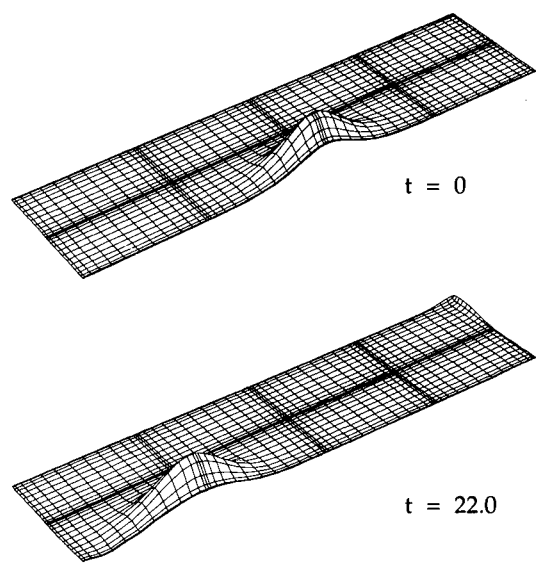

Figure 2. Propagation of an equatorial Rossby soliton: (a) initial surface displacement; (b) surface displacement after 8·5 days ($t = 22$ non-dimensional time units)

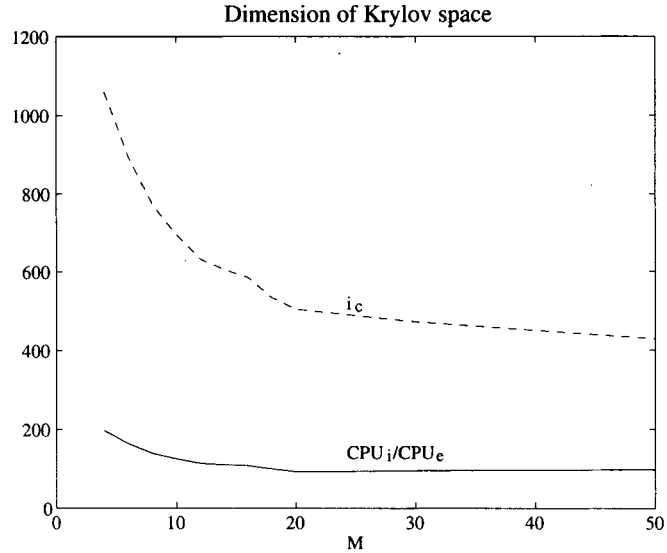

Figure 3. Effects of dimension of Krylov space, M, on convergence rate and CPU time: —, ratio of CPU_i to CPU_e; – –, number of iterations required for convergence. $\varepsilon = 10^{-6}$ and $\Delta t_i/\Delta t_e = 5$

A second series of tests was performed to determine the best convergence criterion: $\Delta t_i = 5\Delta t_e$ and $M = 20$ were kept constant, while ε was decreased from 10^{-2} to 10^{-10}. Figure 4(a) shows the increase in CPU time and iteration number as ε decreases. The convergence criterion ε must be chosen as large as possible without increasing the error in the solution. Figure 4(b) shows the ratios of the maximum errors in the implicit and explicit schemes for u, v and ζ. The ratios do not decrease for $\varepsilon < 10^{-5}$, because the errors are dominated by the time discretization error rather than by the error arising from the approximate solution of the system of equations.

The third series of tests was used to determine the optimal time steps: $\varepsilon = 10^{-5}$ and $M = 20$ were kept constant, while Δt_i was increased from $2\Delta t_e$ to $30\Delta t_e$. Figure 5(b) is a plot of $\Delta t_i/\Delta t_e$ versus the ratio of implicit and explicit errors. Figure 5(a) shows an alarming trend in the number of iterations and CPU time needed to achieve convergence as the time step is increased: both increase with the time step. Hence there are no benefits in increasing the size of the time step for the present implicit scheme. The cost could be partly alleviated by making the convergence criterion less stringent as the time step is increased. There would be no harm in doing so, since the temporal error would also increase.

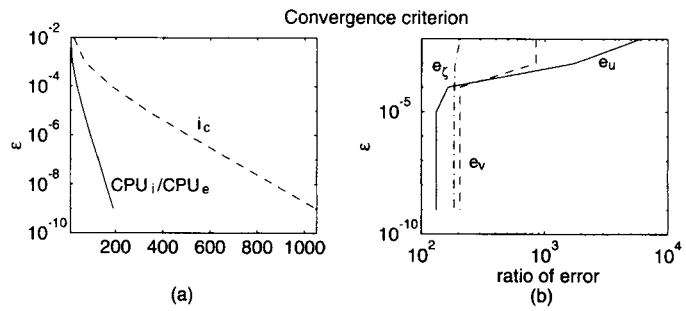

Figure 4. Effects of convergence criterion ε on CPU time and errors: (a) —, CPU_i/CPU_e; – –, number of iterations required for convergence; (b) ratio of errors (implicit error/explicit error) in u (—), v (– –) and ζ (– · –). $M = 20$ and $\Delta t_i/\Delta t_e = 5$

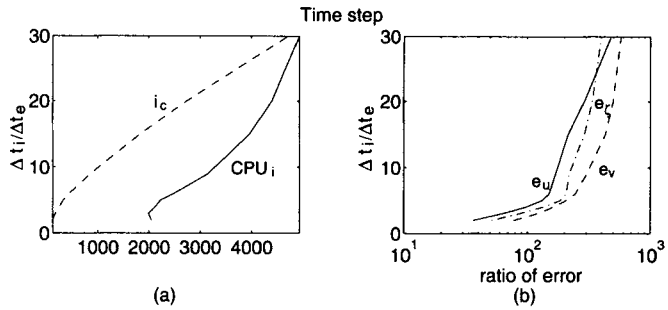

Figure 5. Effects of time step ratio $\Delta t_i/\Delta t_e$ on CPU time and errors: (a) —, CPU$_i$; – –, number of iterations required for convergence; (b) ratio of implicit and explicit maximum errors in u (—), v (– –) and ζ (– · –). $M = 20$ and $\varepsilon = 10^{-5}$

The present implicit implementation is clearly not satisfactory. We are currently evaluating alternatives to GMRES (e.g. the QMR algorithm) and possible simple preconditioning schemes. We do not pursue the issue further here. The rest of our results were obtained with the explicit scheme exclusively.

6. NUMERICAL APPLICATION

The accuracy and fast convergence rate of the SE model are best illustrated with a simple analytic problem; the convergence of the numerical solution can then be monitored as the spatial resolution is increased. We have chosen the problem of a linear and inviscid standing wave in a square basin of unit width and depth. The analytic solution of the inviscid linear shallow water equations is

$$\begin{aligned} u &= \frac{1}{\sqrt{2}} \sin(\pi x)\ \cos(\pi y)\ \sin\left(\sqrt{2}\pi t\right), \\ v &= \frac{1}{\sqrt{2}} \cos(\pi x)\ \sin(\pi y)\ \sin\left(\sqrt{2}\pi t\right), \\ \zeta &= \cos(\pi x)\ \cos(\pi y)\ \cos\left(\sqrt{2}\pi t\right). \end{aligned} \tag{32}$$

We have normalized the gravity coefficient to unity and set the viscosity, the bottom drag coefficient, the Coriolis parameter and the wind stress to zero. We have used very small time steps in our calculations so that the error is dominated by the spatial discretization errors even at the highest spatial resolution. The numerical and analytical solutions were compared at $t = 2$ (1·41 wave period) and the maximum errors in the velocity and surface height were computed. Figure 6(a) shows the exponential decrease in the error of the SE solution as the order of the interpolation polynomial is increased for one-, four- and nine-element partitions of the basin. The error decreases by several orders of magnitude when N' is doubled. The curve for the nine-element partition flattens out near $N' = 13$ because of finite precision arithmetic; the errors incurred are zero to machine accuracy. The velocity and pressure errors seem to decrease at the same rate, which must then be dictated by the resolution of the pressure mesh. Figure 6(b) shows the algebraic (fixed-order) rate of convergence of the SE solution as the number of elements is increased for constant N'. Notice that this rate is not fixed but increases with N'. The present example illustrates how the accuracy of the spectral element method can be improved either by increasing the order of the interpolation polynomial while keeping the element partition fixed or by increasing the number of elements while keeping the order of the interpolation polynomial fixed; in either case the convergence is more rapid than in low-order methods.

The SE solution of the full shallow water equations has also been compared with a finite difference solution computed by Milliff and McWilliams.[25] Their numerical solution relies on the well-

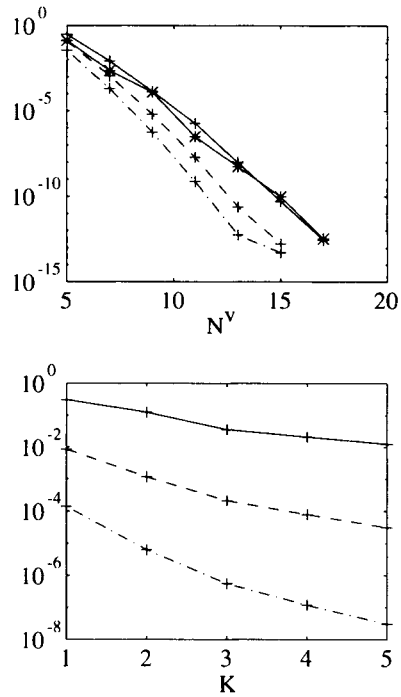

Figure 6. Maximum error in the spectral element (SE) method for the linear sloshing problem: (a) SE error versus N^v for one-element (—), four-element (– –) and nine-element (– · –) partitions (*, error in ζ; +, error in u, v); (b) SE errors versus number of elements in each direction for $N^v = 5$ (—), $N^v = 7$ (– –) and $N^v = 9$ (– · –)

established Sadourny model[26] to discretize the governing equations. Milliff and McWilliams study the reflection of a monopole vortex off the western coast of an idealized ocean basin in order to elucidate the role that boundary pressure plays in coupling coastal and interior dynamics.

We use the same boundary conditions (free-slip) and physical parameters as Milliff and McWilliams except for the friction coefficient. The basin size is 3600×2800 km^2 (from 23°N to 48°N), the basin depth is 1000 m, the reduced gravity is $0{\cdot}081$ m^2 s^{-1}, the Coriolis parameter at mid-latitude is 9×10^{-5} s^{-1} and the linear variation in the Coriolis parameter with latitude is $1{\cdot}8 \times 10^{-11}$ m^{-1} s^{-1}. Milliff and McWilliams employ a biharmonic friction with a hyperviscosity of $9{\cdot}3 \times 10^9$ m^4 s^{-1}. We use harmonic friction with a viscosity of 50 m^2 s^{-1}. The two frictions have the same spin-down time on waves with length scales of about 14 km. Our grid consists of 266 elements, with 121 nodes per element for the velocity grid and 81 for the pressure. The total numbers of nodes are 26,931 and 17,289 for the velocity and pressure grids respectively. The elements are uniform in the interior of the basin (200×200 km^2). We have refined the elements along the western boundary by using elements of 100 km width. This was necessary in order to resolve the western boundary current (estimated width of 23 km) and the short waves arising from the reflection of the Rossby wave.

We start the motion with an initial monopole vortex in gradient balance, as in Reference 25 (Figure 7), and track its evolution by plotting pressure contour at 40 day intervals. Figure 8 compares the finite difference and SE solutions at day 160 (the contour levels were chosen so as to reveal the presence of the scattered Rossby waves). The two models give the same results and phenomenology. The reflection of the vortex off the western boundary triggers Kelvin waves that propagate along the sides of the

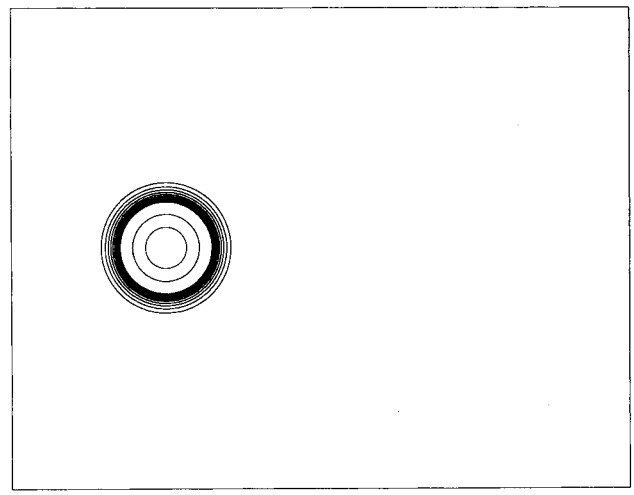

Figure 7. Contour plot of the depth anomaly for a monopole vortex in gradient balance at day 0

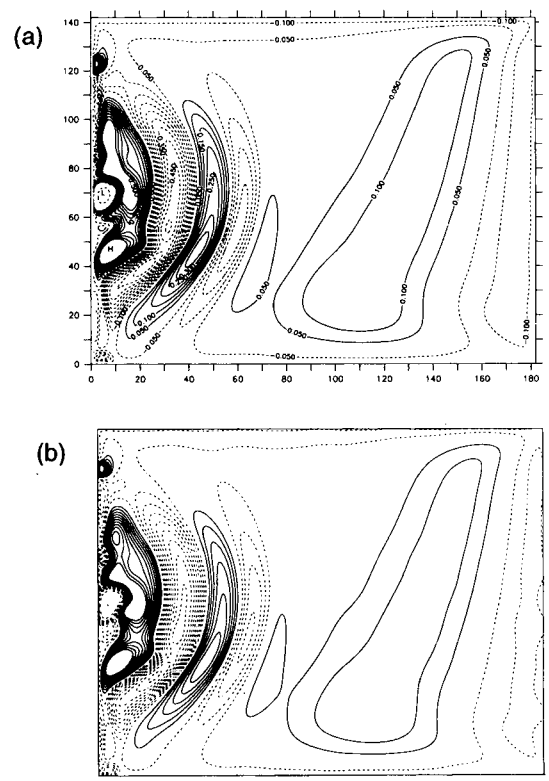

Figure 8. Contour plots of the depth anomaly for a monopole vortex reflecting off the western coast at day 160: (a) finite difference solution; (b) spectral element solution

basin. As the Kelvin waves travel along meridional boundaries, they scatter mass into interior Rossby waves (see the eastern boundary region in Figure 8).

Lastly, we demonstrate the ability of the spectral element method to handle complex geometries by studying the wind-driven circulation in the North Atlantic Ocean. The discretization of the region, with and without the interpolation nodes, is shown in Figure 9. The mesh has 118 elements, each containing 121 interpolation nodes for the velocity and 81 for the pressure. The total numbers of nodes are 12,120 and 7,808 for the velocity and pressure grids respectively. The minimum, maximum and average distances between interpolation points on the velocity grid are 5, 214 and 65 km respectively. The largest elements are located at the eastern side of the basin. On the western side we use small elements in order to resolve the western boundary current. The wind forcing is taken from the monthly ECMWF data sets and averaged over a period of 1 year; the wind forcing was then applied steadily in time.

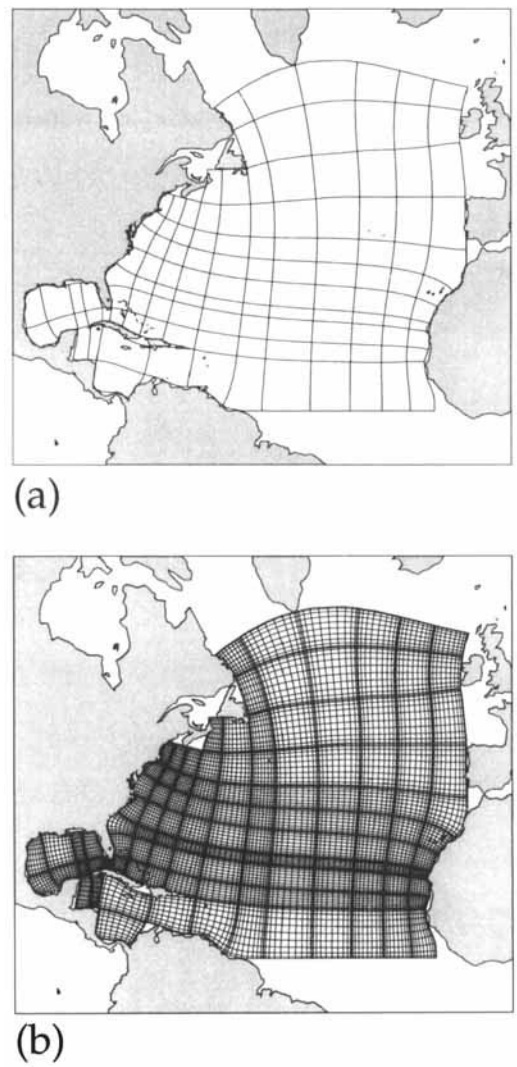

Figure 9. Spectral element mesh of the North Atlantic: (a) the elements; (b) the elements with the interpolation points

The first simulation shows what can go wrong when the non-staggered grid is used for the solution of the shallow water equations in the incompressible limit. The depth is constant at 1000 m and the full gravity is used (g = 9·81 m s^{-2}), yielding 100 m s^{-1} for the speed of the free surface gravity wave. We choose a very small time step, 3·6 s, to guard against instabilities. The friction is provided by linear bottom drag with coefficient γ set to 10^{-7} s^{-1} (a spin-down time of about 100 days) and by Laplacian friction with viscosity set to 50,000 m^2 s^{-1}. This viscosity is huge, but the non-staggered code would not run with a smaller value. The simulation is started from rest and the wind is increased from zero to full amplitude over 12 days. Figures 10(a) and 11(a) show the contours of the u and ζ-fields respectively 10 days after the start of the calculations. The velocity contours are smooth and betray only small oscillations in the centre of the basin. The ζ-contours, on the other hand, are contaminated by spurious pressure modes. The same run was repeated with exactly the same parameters but using the staggered grid; the results are shown in Figures 10(b) and 11(b). The u and ζ-fields are now smooth; the staggering we adopted has effectively eliminated the spurious pressure modes. We should note that the depth of the present basin, 1000 m, is rather shallow, but we had difficulties running the non-staggered scheme on a 5000 m deep basin; the computations failed catastrophically even with the

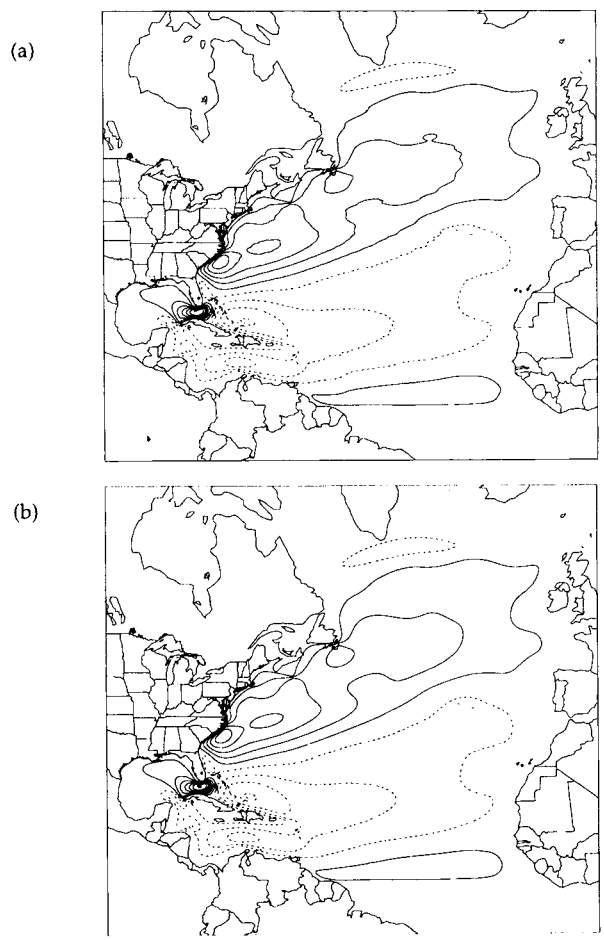

Figure 10. Contours of u for the wind-driven circulation in the North Atlantic at day 10 (Δt = 3·6 s, $\nu = 5 \times 10^4$ m^2 s^{-1}): (a) non-staggered grid; (b) staggered grid. $u_{min} = -2{\cdot}4$ cm s^{-1}, $u_{max} = 4{\cdot}0$ cm s^{-1} and $\Delta u = 0{\cdot}64$ cm s^{-1}

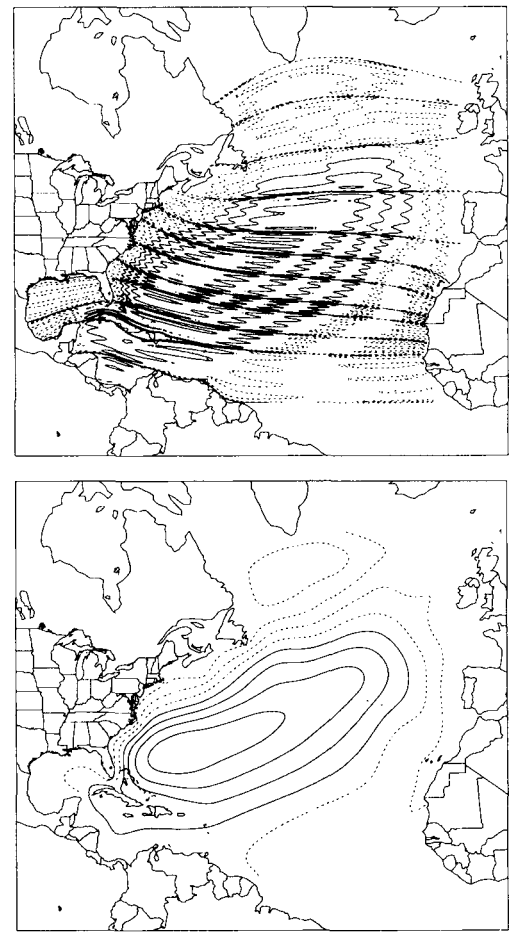

Figure 11. Contours of ζ for the wind-driven circulation in the North Atlantic at day 10 (Δt = 3·6 s, $\nu = 5 \times 10^4$ m^2 s^{-1}): (a) non-staggered grid; (b) staggered grid. $\zeta_{min} = -14$ cm, $\zeta_{max} = 18$ cm and $\Delta\xi = 3{\cdot}2$ cm

large viscosity. In contrast, the staggered grid model can be run with significantly reduced viscosity ($\geqslant$ 1000 m^2 s^{-1}) and with larger time steps. The present simulation with its 1000 m deep ocean illustrates the need for the staggered grid even at moderate depths.

To confirm that the staggered grid model is capable of extended integration at low values of the viscosity, we have repeated the above experiment with new parameters. The model is run in reduced gravity mode in order to simulate the response of the first baroclinic mode to the wind forcing. The gravity coefficient and the depth are set to 0·02 m s^{-2} and 1250 m respectively so that the gravity wave speed is 5 m s^{-1} and the Rossby radius of deformation is about 56 km at the centre of the basin. The time step is increased to 2 min and the viscosity is set to $\nu = 2000$ m^2 s^{-1}. The wind forcing is now provided by the monthly ECMWF winds instead of the annual mean. Figure 12 shows a snapshot of the depth anomaly contours, ζ, 8 years after the start of the simulation. One can see the development of a distinct western boundary current with an estimated Munk layer thickness of about 40 km at mid-latitudes. Notice also that the method does not require any special treatment to handle multiply connected regions (e.g. the islands of Cuba and Haiti were taken into account).

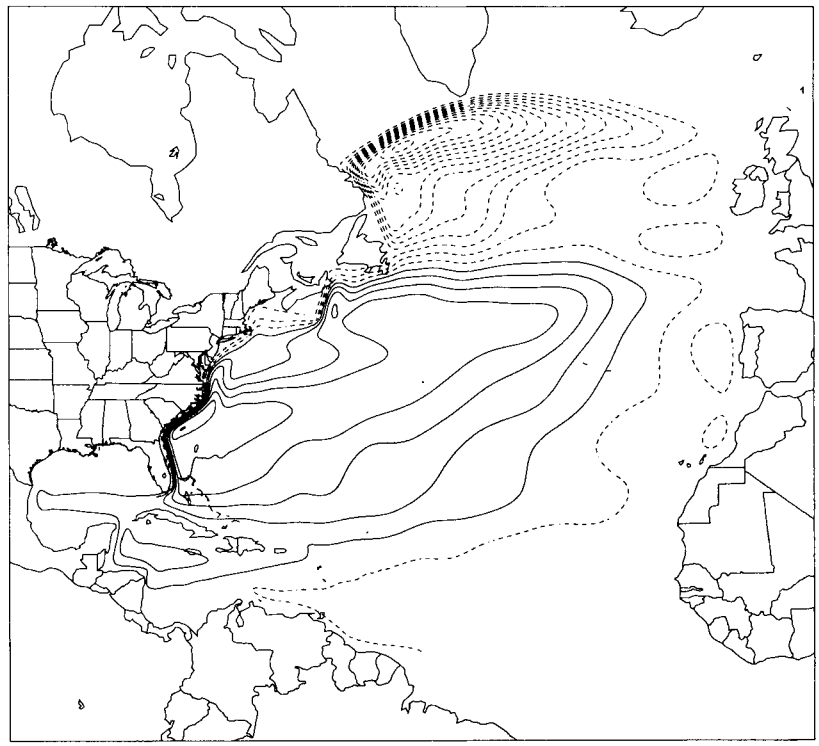

Figure 12. Depth anomaly contours, ζ, 8 years after the start of the simulation (Δt = 120 s, $\nu = 2 \times 10^3$ m^2 s^{-1}). $\zeta_{min} = -203$ m, $\zeta_{max} = 106$ m and $\Delta\xi = 15{\cdot}45$ m

7. DISCUSSION AND CONCLUSIONS

We have described a staggered spectral element model to solve the oceanic shallow water equations. Several numerical experiments were conducted to verify the model and to compare its behaviour with that of more established models. The North Atlantic simulation in particular illustrates the versatility of our spectral element method in handling complex geometries and nearly divergence-free flows. The choices of Legendre polynomials, numerical quadrature and explicit integration are imperative to make the method competitive with other numerical models. The implicit scheme we presented is handicapped by the lack of an efficient solver.

General application of our spectral element model to problems in regional and large scale ocean circulation modelling requires further implementation and improvement in several areas. For example, a robust, flexible and automatic grid generation code is an important prerequisite to exploit the full geometrical advantages of the spectral element model. Unstructured grid generation would be ideal; unfortunately, this technique is more suitable for triangles than quadrilaterals. Our efforts to transform a triangular mesh of the North Atlantic (based on Delaunay triangulation) into a quadrilateral mesh by merging triangles were fruitless. Some of the quadrilaterals were degenerate with large obtuse angles and the mapping of others turned out to be singular. Our current grid generation code relies on block-structured techniques. The basin is first divided into large blocks that are mapped conformally into rectangles. The spectral elements are generated in these rectangles and the locations of the spectral nodes in the physical domain are computed. The different blocks must then be pieced together. The whole process requires quite a bit of manual intervention at the present time, but we are writing several

programmes to automate the different steps. We are also evaluating several grid generation packages that would help improve our gridding capabilities.

Although an appropriate test-bed for the spectral element method and a relevant idealization to several topical problems in contemporary ocean modelling (see e.g. Reference 25), the shallow water equations are not suitable for the study of many important oceanographic phenomena such as the thermohaline circulation and the ocean's role in the climate system. Exploration of these problems will clearly require the utilization of the full three-dimensional primitive equations or an appropriate approximate system. Design criteria include the accurate representation of the effects of topography and of the surface and bottom boundary layers, the development of time-stepping procedures for the efficient treatment of the internal and external modes on an unstructured grid, and the utilization of high-performance (parallel) computing platforms to enable affordable basin- and global-scale integrations on (at least) decadal time scales. The domain decomposition philosophy underlying the spectral element technique and the large ratio of inter-element computation to intra-element communication suggest that these models will be well suited to the parallel computing environment. Conceptual development of the three-dimensional model and its parallel implementation are both under way.

ACKNOWLEDGEMENTS

This research was supported by the following grants: NSF grant OCE9119459 and DOE Grant KC070101 to the University of Michigan; NSF grant OCE8812300 and ONR grant N00014-93-1-0197 to Rutgers University. We would like to thank Dr. Ralph Milliff and Gabe Vecci for their help with the monopole vortex run, and Dr. John McCalpin for helpful discussions.

APPENDIX: OPERATOR SPLITTING

The spatial discretization gives rise to a system of equations of the general form

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{A}(t)\mathbf{u} + \mathbf{B}(t)\mathbf{u} + \mathbf{f}(t), \tag{33}$$

where $\mathbf{u}$ is the vector of unknowns, $\mathbf{A}$ is the matrix representing the discretized stiff operators, $\mathbf{B}$ is the matrix representing the discretization of the computationally intensive operators and $\mathbf{f}$ is the vector of forces. The aim of the splitting method is to separate the time integration of the $\mathbf{A}$- and $\mathbf{B}$-operators. To this end we multiply the above equation by the integration factor matrix $\mathbf{Q}(t^*, t)$:

$$\mathbf{Q}(t^*, t)\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{Q}(t^*, t)\mathbf{A}(t)\mathbf{u} + \mathbf{Q}(t^*, t)[\mathbf{B}(t)\mathbf{u} + \mathbf{f}(t)]. \tag{34}$$

$\mathbf{Q}(t^*, t)$ is defined as

$$\frac{\mathrm{d}\mathbf{Q}(t^*, t)}{\mathrm{d}t} = -\mathbf{Q}(t^*, t)\mathbf{A}(t), \qquad \mathbf{Q}(t^*, t^*) = \mathbf{I}, \tag{35}$$

where $\mathbf{I}$ is the identity matrix. The definition of $\mathbf{Q}$ implies that (34) can be rewritten as

$$\frac{\mathrm{d}[\mathbf{Q}(t^*, t)\mathbf{u}]}{\mathrm{d}t} = \mathbf{Q}(t^*, t)[\mathbf{B}(t)\mathbf{u} + \mathbf{f}(t)]. \tag{36}$$

Equation (36) is an ordinary differential equation in the unknown $\mathbf{Q}(t^*, t)\mathbf{u}$ which involves the $\mathbf{B}$- and $\mathbf{Q}$-matrices only. The problem becomes one of calculating $\mathbf{Q}$, which is defined as the exponential of matrix $\mathbf{A}$ by equation (35). $\mathbf{Q}$ is not constructed explicitly, since the exponential of a general matrix is not easy to form. Note that the integration of (36) involves the product $\mathbf{Q}\psi$, where ψ could be $\mathbf{u}$,

Bu or **f**. The formulation of these products can be accomplished through the integration of the auxiliary problem

$$\frac{d\tilde{\psi}}{ds} = \mathbf{A}(t+s)\tilde{\psi}, \quad 0 \leqslant s \leqslant t^* - t, \qquad \tilde{\psi}(t^*, t; s=0) = \psi(t), \tag{37}$$

where s is the independent variable and t^* and t are parameters. Multiplying both sides of (37) by $\mathbf{Q}(t^*, t+s)$ and invoking the definition (35), equation (37) becomes

$$\frac{d \leqslant ft[\mathbf{Q}(t^*, t+s)\tilde{\psi}(t^*, t; s)]}{ds} = 0,$$

which implies that the product $\mathbf{Q}(t^*, t+s)\tilde{\psi}$ $(t^*, t; s)$ is constant in s and hence $\mathbf{Q}(t^*, t)\tilde{\psi}(t^*, t; s=0) = \mathbf{Q}(t^*, t^*)\tilde{\psi}(t^*, t; t^* - t)$. Since $\mathbf{Q}(t^*, t^*) = \mathbf{I}$ and $\tilde{\psi}(t^*, t; s=0) = \psi(t)$, we have

$$\mathbf{Q}(t^*, t)\psi(t) = \tilde{\psi}(t^*, t; t^* - t). \tag{38}$$

Hence forming the products $\mathbf{Q}\psi$ amounts to integrating the auxiliary problem (37) from $s = 0$ to $t^* - t$.

The procedure outlined above effectively decouples the time integration of the **A**-and **B**-terms. The time integration of the **B**-operator is effected through equation (36) and that of the **A**-operator through equation (37). As noted in the original paper of Maday *et al.*,[19] the operator–integration factor procedure is not a time-stepping scheme in itself but is a mean to generate consistent time-splitting schemes.

Our implicit procedure uses a second-order backward differentiation scheme[18] to integrate (36) in t and a second-order Runge–Kutta (RK2) scheme to integrate (37) in s. The time discretization of (36) is

$$\frac{\frac{3}{2}\mathbf{u}^{n+1} - 2\mathbf{Q}(t^{n+1}, t^n)\mathbf{u}^n + \frac{1}{2}\mathbf{Q}(t^{n+1}, t^{n-1})\mathbf{u}^{n-1}}{\Delta t} = \mathbf{B}^{n+1}\mathbf{u}^{n+1} + \mathbf{f}^{n+1}.$$

There are two **Qu** products to compute with the auxiliary problems:

$$\frac{d\tilde{\mathbf{u}}^{n-1}}{ds} = \mathbf{A}(t^n - 1 + \mathbf{s})\tilde{\mathbf{u}}^{n-1}, \quad 0 \leqslant s \leqslant t^{n+1} - t^{n-1}, \qquad \tilde{\mathbf{u}}^{n-1}(\mathbf{s}=0) = \mathbf{u}^{n-1}, \tag{39}$$

$$\frac{d\tilde{\mathbf{u}}^{n}}{ds} = \mathbf{A}(t^n + s)\tilde{\mathbf{u}}^{n}, \quad 0 \leqslant s \leqslant t^{n+1} - t^{n}, \qquad \tilde{\mathbf{u}}^{n}(\mathbf{s}=0) = \tilde{\mathbf{u}}^{n}, \tag{40}$$

The discretization of (39) with the RK2 scheme gives

$$\tilde{\mathbf{u}}^{n-1, m+1/2} = \tilde{\mathbf{u}}^{n-1, m} + \frac{\Delta s}{2}\mathbf{A}^{n-1, m}\tilde{\mathbf{u}}^{n-1, m}, \tag{41}$$

$$\tilde{\mathbf{u}}^{n-1, m+1} = \tilde{\mathbf{u}}^{n-1, m} + \Delta s\mathbf{A}^{n-1, m+1/2}\tilde{\mathbf{u}}^{n-1, m+1/2}, \tag{42}$$

where $\Delta s = \Delta t/M$, $\tilde{\mathbf{u}}^{n-1,0} = \mathbf{u}^{n-1}$ and $m = 0, 1, \ldots, 2M - 1$. A similar set of equations can be obtained from (40):

$$\tilde{\mathbf{u}}^{n,m+1/2} = \tilde{\mathbf{u}}^{n,m} + \frac{\Delta s}{2}\mathbf{A}^{n,m}\tilde{\mathbf{u}}^{n,m}, \tag{43}$$

$$\tilde{\mathbf{u}}^{n,m+1} = \tilde{\mathbf{u}}^{n,m} + \Delta s\mathbf{A}^{n,m+1/2}\tilde{\mathbf{u}}^{n,m+1/2}, \tag{44}$$

where $\Delta s = \Delta t/M$, $\tilde{\mathbf{u}}^{n,0} = \mathbf{u}^n$ *and* $m = 0, 1, \ldots, M - 1$. Note that the choice of a backward difference scheme relieved us from computing terms such as **QB** and **Qf**.

REFERENCES

1. C. Le Provost and A. Poncet, 'Finite element method for spectral modelling of tides, *Int. j. numer. methods fluids*, **12**, 853–871 (1978).
2. D. R. Lynch and F. E. Werner, 'Three-dimensional hydrodynamics on finite elements. Part I: Linearized harmonic model', *Int. j. numer. methods fluids*, **7**, 871–909 (1987).
3. D. B. Haidvogel, J. L. Wilken and R. Young, 'A semi-spectral primitive equation ocean circulatin model using vertical sigma and orthogonal curvilinear horizontal coordinates', *J. Comput. Phys.*, **94**, 151–185 (1991).
4. B. Kelly, 'On the generation and dispersion of Yanai waves with a spectral Chebychev-collocation reduced gravity ocean model', *Tec. Rep.*, Mesoscale Air–Sea Interaction Group, Florida State University, Tallahassee, FL, 1993.
5. H. Ma, 'A spectral element basin model for the shallow water equations', *J. Comput. Phys.*, **109**, 133–149 (1993).
6. A. Patera, 'A spectral element method for fluid dynamics: laminar flow in a channel expansion', *J. Comput. Phys.*, **54**, 468–488 (1984).
7. G. E. Karniadakis, 'Spectral element simulation of turbulent flows in complex geometries', *Appl. Numer. Math.*, **6**, 85–105 (1989).
8. P. Fischer, L. W. Ho, G. E. Karniadakis, E. M. Ronquist and A. T. Patera, 'Recent advances in parallel spectral element simulation of unsteady incompressible flows', *Comput. Struct.*, **30**, 217–231 (1988).
9. Y. Maday and A. T. Patera, 'Spectral element methods for the incompressible Navier–Stokes equations', in A. K. Noor (ed.), *State of the Arts Surveys in Computational Mecahnics*, ASME, New York, 1988, pp.
10. E. M. Ronquist, 'Optimal spectral element methods for the unsteady three-dimensional incompressible Navier–Stokes equaqtions', *Ph.D. Thesis*, Massachusetts Institute of Technology, Cambridge, MA, 1988.
11. P. R. Gent, 'The energetically consistent shallow-water equations', *J. Atmos. Sci.*, **50**, 1323–1325 (1993).
12. C. Bernardi and O. Pironneau, 'On the shallow water equations at low Reynolds number', *Commun. Partial Diff. Eq.*, **16**, 59–104 (1991).
13. M. D. Gunzburger, *Finite Element Methods for Viscous Incompressible Flows, a Guide to Theory Practice and Algorithms*, Academic, New York, 1989.
14. M. R. Schumack, W. W. Shultz and J. P. Boyd, 'Spectral method solution of the Stokes equations on nonstaggered grids', *J. Comput. Phys.*, **94**, 30 (1991).
15. J. P. Boyd, in C. A. Brebbia and S. A. Orszag (eds), *Lecture Notes in Engineering*, Vol. ?, *Chebyshev and Fourier Spectral Methods*, Springer, New York, 1989, pp.
16. M. Y. Hussaini, D. A. Kopriva and A. T. Patera, 'Spectral collocation methods', *Appl. Numer. Math.*, **5**, 177–208 (1989).
17. G. E. Karniadakis, E. T. Bullister and A. T. Patera, 'A spectral element method for the two- and three-dimensional time-dependent incompressible Navier–Stokes equations', *Proc. Eur.–U.S. Symp. on Finite Element Methods for Nonlinear Problems*, Springer, Berlin, 1986, pp.
18. C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1971.
19. Y. Maday, A. T. Patera and E. M. Ronquist, 'An operator–integration-factor splitting method for time-dependent problems: applications to incompressible fluid flow', *J. Sci. Comput.*, **5**, 263–292 (1990).
20. Y. Saad and M. H. Schultz, 'GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems', *SIAM J. Sci. Stat. Comput.*, **7**, 856–869 (1986).
21. A. Staniforth and J. Cote, 'Semi-Lagrangian integration schemes for atmospheric models—a review', *Mon. Weather Rev.*, **119**, 2206–2223 (1991).
22. L. -W. Ho, Y. Maday, A. T. Patera and E. M. Ronquist, 'A high order Lagrangian-decoupling method for the incompressible Navier–Stokes equations', *Comput Methods Appl. Mech. Eng.* **80**, 65–90 (1990).
23. H. F. Walker, 'Implementation of the GMRES method', *Comput. Phys. Commun.*, **53**, 311–320 (1989).
24. J. P. Boyd, 'Equatorial solitary waves. Part I: Rossby solitons', *J. Phys. Oceanog.*, **10**, 1699–1717 (1980).
25. R. Milliff and J. McWilliams, 'The evolution of boundary pressure in ocean basins', *J. Phys. Oceanogr.*, **24**, 1317–1338, (1994).
26. R. Sadourny, 'The dynamics of finite-difference models of the shallow water equations', *J. Atmos. Sci.*, **32**, 680–689, 1975.
27. R. W. Freund and N. M. Nachtigal, 'QMR: a quasi-minimal residual method for non-Hermitian linear systems', *Numer. Math.*, **60**, 315 (1992).