A Nonhydrostatic Finite-Element Model for Three-Dimensional Stratified Oceanic Flows. Part I: Model Formulation

R. FORD

Department of Mathematics, Imperial College, London, United Kingdom

C. C. PAIN, M. D. PIGGOTT, A. J. H. GODDARD, C. R. E. DE OLIVEIRA, AND A. P. UMPLEBY

Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College, London, United Kingdom

(Manuscript received 14 July 2003, in final form 22 January 2004)

ABSTRACT

The vast majority of advanced numerical ocean models in use today, while performing extremely well, especially for certain classes of problem, do not necessarily take full advantage of current trends in numerical analysis and scientific computing. Here, a three-dimensional finite-element model is presented for use in oceanic simulations. The main aim is to fully exploit the use of unstructured meshes in both the horizontal and vertical directions, in order to conform well to topography and coastlines, and also to enable the straightforward use of dynamic variable mesh resolution reflecting fluid flow. In addition, the model should be accurate and efficient under typical oceanographic conditions, and not make the hydrostatic approximation. For simplicity here however the model does assume the presence of a rigid lid. To cope with inherent instabilities present in finite-element simulations of incompressible flow, caused by the Lagrange multiplier role that pressure plays in satisfying incompressibility, a mixed formulation for representing velocity and pressure is employed. Additionally, instabilities occurring due to the advection-dominated nature of the flow are dealt with using linear Petrov–Galerkin methods. In the course of this work a different type of instability has also been observed, which has some similarities with the sigma coordinate pressure gradient problem. The instability results from the mixed nature of the finite-element formulation and consequent poor satisfaction of hydrostatic balance, which in turn manifests itself in errors and spurious velocities on distorted meshes, such as those typical over topography. A satisfactory solution to this problem is presented that involves a splitting of pressure and allows efficient computations, whether the flow be close to or far from a state of hydrostatic balance.

1. Introduction

The purpose of this paper is to propose a new approach to the numerical modeling of oceanic flows, using three-dimensional finite elements. Finite-element methods offer great geometric flexibility and hence are often used to model flows in complex domains. This is due to the fact that, as opposed to the pointwise nature of finite-difference methods typically defined on topologically Cartesian grids, they generally represent solutions as piecewise polynomials. The piecewise polynomials are formed by allowing a polynomial representation of the numerical solution over so-called *elements*. These elements, which define a mesh, are nonempty convex polyhedra in nature and are used to give a proper partition (in that they form a nonoverlapping and total covering) of the physical domain, and such that the intersection of any two elements is either empty or contains common faces, edges, or vertices. The flexibility of finite-element methods now stems from the fact that they are equally well defined on both *structured* and *unstructured* meshes. A structured mesh is so-called since it possesses a uniform topological structure, for example, one that is topologically equivalent to a uniform Cartesian mesh, so that in two dimensions (2D) every internal grid point has four connected neighbors. In contrast an unstructured mesh has no such constraints, an internal mesh point, again in 2D, can have anything from three-upward connected neighbors. For simplicity, and as a first step to a more complete model, it has been decided to employ 3D static (i.e., nonadaptive) hexahedral meshes exclusively in this work, along with the assumption of a rigid lid. The use of adaptive tetrahedral meshes and free surface formulations shall be reserved for future publications.

The flexibility afforded by the use of unstructured meshes allows for an accurate representation of highly irregular coastlines and topography. Compare this, for example, with restrictions imposed by a curvilinear-

Corresponding author address: Dr. M. D. Piggott, Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College, Prince Consort Road, London, SW7 2BP, United Kingdom.
E-mail: m.d.piggott@imperial.ac.uk

structured mesh representation for coastlines, and a staircase representation of topography. Of equal and perhaps even more importance, an unstructured mesh may be adapted to the solution with far more ease and generality than a structured mesh, either dynamically or a priori. Large-scale oceanic circulation possesses many localized features such as western boundary currents and regions of confined eddy activity. The use of an adapting variable resolution mesh is the only practical means to faithfully represent the physical processes at work, both locally in isolation as well as in relation to their coupling with the larger scale. Most numerical ocean models use a discretization in which horizontal resolution is essentially uniform. At a relatively coarse resolution of, say, ½° × ½°, it is reasonable to suppose that substantial improvements can be obtained simply by refining resolution everywhere. However, for highly nonuniform and anisotropic flows such as those present in ocean circulation, solution adaptive procedures should be developed and investigated. Although these techniques require additional computations, and indeed the finite-element method requires more computations than comparable methods with structured meshes, there will inevitably come a point at which the additional ‘‘overhead’’ of further uniform mesh refinement where it is not required is greater than the overhead associated with the use of finite elements and mesh adaptivity.

Nonetheless, finite-element methods have been relatively underused in numerical ocean modeling. It is reasonable to ask questions regarding the accuracy, efficiency, and applicability of finite-element methods, particularly in comparison with models currently in widespread use. The aim of this paper is to make a start at addressing these issues; the long-term goal being the development of a finite-element ocean model in which solution adaptive techniques can be employed, and which yields high-quality results for problems at a range of scales.

The possibility of applying the finite-element technique to oceanic problems has been discussed for roughly three decades. Fix (1975) investigated the properties of finite elements when applied to a barotropic quasi-geostrophic potential vorticity equation. Favorable results were given regarding stability, accuracy, aliasing errors, and conservative laws, along with the natural treatment of certain boundary conditions and geometric flexibility of the approach. Dumas et al. (1982) considered a similar system and noted the possibilities and advantages of increased resolution in western boundary currents for wind-driven gyre-type problems. An investigation of finite-difference, finite-element, and (pseudo) spectral models applied to open-ocean problems was conducted by Haidvogel et al. (1980). It was demonstrated that finite-element and spectral methods can be more than competitive with finite-difference methods for certain problems.

Until now the large proportion of serious and successful applications of the finite-element method in oceanic applications has been to tidal and coastal problems (Lynch and Gray 1979; Provost and Vincent 1986; Lynch and Werner 1987, 1991; Lynch et al. 1996). These have primarily been based on shallow-water models, often including reformulations in terms of Helmholtz operators or second-order wave equations.

More recently larger-scale and more complex diagnostic models have been developed and examined. For example, Myers and Weaver (1995) considered a discretization of a linearized steady-state barotropic potential vorticity equation in spherical coordinates; the advantages of irregular grids designed to conform to geographic features, and also the natural manner in which certain boundary conditions may be treated were highlighted. Greenberg et al. (1998) examined a similar linearized diagnostic model in spherical-polar coordinates. Finally, Nechaev et al. (2003) discussed a diagnostic model for a 3D hydrostatic problem on a mesh unstructured in the horizontal, making use of residual-free bubble stabilization techniques. This has recently been extended to a prognostic model by Danilov et al. (2004).

Research into the application of finite elements to 3D (hydrostatic) primitive equations is thriving. Early problems, cited by many as reasons to avoid finite elements have been, and are being, addressed. For example, Le Roux et al. (1998) investigated the ability of various element choices to describe states of geostrophic balance in a noise-free manner for the linear shallow-water equations. This work on spatial discretizations was extended by Le Roux et al. (2000) with a semi-Lagrangian treatment of the temporal discretization. Similarly, Hanert et al. (2002) performed a study to select a convenient (low order) mixed formulation for ocean modeling by considering the linear shallow-water equations. Low-order finite-element equivalents of the Arakawa A, B, and C grids were considered and seen to behave in a broadly analogous manner to their staggered counterparts. The generation of boundary-fitted Delaunay triangulations in 2D was considered by Legrand et al. (2000) with application to the global ocean.

In addition, work has been conducted on models that employ spectral representations of solution variables. Haidvogel et al. (1991) described a semispectral sigma coordinate primitive equation model that represents the solution in the vertical via a sum over a polynomial basis, whereas standard second-order finite differences are used in the horizontal. Iskandarani et al. (1995) considered a staggered spectral element discretization of a shallow-water model on a quadrilateral mesh, with pressure represented using Legendre polynomials of degree 2 less than those used for velocity. This was extended to the 3D hydrostatic primitive equations on unstructured meshes by Iskandarani et al. (2003). Three-dimensional finite elements were used by Lynch et al. (1996), but the discretization is not local to the elements, in the sense that the baroclinic pressure gradient is determined by a procedure of interpolation between the finite-element mesh used for integrating the equations of motion and a set of level surfaces on which the pressure gradient is determined, see also Naimie et al. (1994).

The work presented here seeks to further extend the investigation into the benefits and applicability of finite-element models for oceanic flows. As an extension to previous work, the desire here is to formulate a robust 3D model based on the Boussinesq equations on meshes unstructured in both the horizontal and vertical direction. In addition, since one aim of the approach is the accurate modeling of flow over topography, which may be steep in places and therefore not well represented by the hydrostatic approximation, the nonhydrostatic form of the equations are considered directly here. Also, since solution adaptive techniques will ultimately be employed, yielding the possibility of resolving small-scale (e.g., convective) processes, a nonhydrostatic approach seems judicious.

The primary focus of this paper, and its companion (Ford et al. 2004, hereafter Part II), in addition to the model formulation, is on the faithful representation of topography. A variety of methods exist to handle topography in ocean circulation models. It is now well-recognized that a discontinuous, or ''staircase,'' representation of topography employed since the earliest models (e.g., Bryan 1969) can lead to substantial errors when used to model flow over continuous topography. For example, any downslope flow is represented by a sequence of horizontal motion and vertical mixing, and as such is represented in a very diffusive way, regardless of the resolution employed. Sigma coordinates have been used, for example, by Haidvogel et al. (1991), but these can also lead to substantial errors in the calculation of the horizontal pressure gradient, especially where the topography is steep (Haney 1991; Mellor et al. 1998). Shaved and partial cell finite-volume methods have been used successfully by Adcroft et al. (1997) where height is the vertical coordinate.

The remainder of this paper is set out as follows. In section 2 a standard finite-element formulation for a Boussinesq fluid is presented. In section 3 it is demonstrated that a standard numerical approach is unsuitable for certain geophysical applications, including flows over varying topography. The reason for this is related to the classical sigma coordinate problem, in which horizontal pressure gradients must be represented as accurately as possible in order to prevent the production of large spurious velocities. This problem also occurs on irregular meshes, which are to be expected with an adaptive mesh model, it is thus doubly important to consider here. In section 4 it is shown that this problem can be solved, to a sufficient order of accuracy for the problems considered here, by splitting the pressure into ''hydrostatic'' and ''nonhydrostatic'' components. In section 5 a Petrov–Galerkin weighting is introduced, which gives more importance to the ''upwind'' direction in the discretization. It provides a means for avoiding additional instabilities often present in convectively dominated problems while preserving accuracy. Finally, in section 6 some concluding remarks are presented.

Part II of this work focuses on the validation of the model as presented here. Flow past a seamount is considered in detail, this is due to the fact that this problem has been the subject of exceptionally thorough numerical studies against which comparisons may be made. Also, the accurate representation of topography is crucial to the successful simulation of the flow.

2. The numerical basis of the model

a. Underlying equations and model formulation

In the finite-element numerical model to be presented the three-dimensional nonhydrostatic Boussinesq equations, as follows, are discretized in a domain $\Omega \subset \mathbb{R}^3$:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla}\cdot\boldsymbol{uu} + 2\boldsymbol{\Omega}\times\boldsymbol{u} = -\boldsymbol{\nabla}p - \rho g\mathbf{k} + \boldsymbol{\nabla}\cdot\boldsymbol{\tau} + \mathbf{F}, \tag{1a}$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0, \tag{1b}$$

$$\frac{\partial T}{\partial t} + \boldsymbol{\nabla}\cdot(\boldsymbol{u}T) = \boldsymbol{\nabla}\cdot(\boldsymbol{\kappa}\boldsymbol{\nabla}T), \tag{1c}$$

$$\rho = -T. \tag{1d}$$

Here, $\boldsymbol{u}$ represents the 3D velocity, p is the perturbation pressure, ρ is the perturbation density, and T is the temperature. The rotation vector is $\boldsymbol{\Omega}$, and here the f-plane assumption $\boldsymbol{\Omega} = (0, 0, f/2)^{\mathrm{T}}$ is made, where f is the (constant) Coriolis parameter.

The stress tensor $\boldsymbol{\tau}$ is used to represent the viscous terms in (1a). A simple model of subgrid-scale turbulence is employed, in which $\boldsymbol{\tau}$ is related to the deformation rate tensor S_{ij}, given by

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{1}{3}\frac{\partial u_k}{\partial x_k}, \tag{2}$$

with summation over repeated indices. In this simple model (Tennekes and Lumley 1972), the elements τ_{ij} and S_{ij} are related by

$$\tau_{ij} = 2\mu_{ij}S_{ij}, \tag{3}$$

where the ''viscosities'' μ_{ij} are constants [there is no summation over repeated indices in (3)]. The horizontal viscosity (μ_{11} and μ_{22}) and vertical viscosity (μ_{33}) are allowed to take two different, uniform values, and the off-diagonal components of μ_{ij} satisfy $\mu_{ij} = (\mu_{ii}\mu_{jj})^{1/2}$. The diagonal components of the diffusivity tensor $\boldsymbol{\kappa}$ in (1c) are also independent, but in this model the off-diagonal components of $\boldsymbol{\kappa}$ vanish.

The simplified equation of state (1d) has been used here, but there is no difficulty, in principle, in incorporating a nonlinear equation of state and including additional constituents, such as salinity.

The system is closed by imposing suitable initial and boundary conditions. With regard to boundary conditions, in this work the normal component of velocity on all boundaries shall be given, $\boldsymbol{u} \cdot \mathbf{n} = g$ on $\partial\Omega$, with typically $g = 0$ at all parts of the boundary other than inflow/outflow, where $g = \mp$Const. Stress-free traction boundary conditions are then applied to all boundaries, $\boldsymbol{\tau} \cdot \mathbf{n} - p\mathbf{n} = 0$, where $\mathbf{n}$ denotes the outward unit normal vector. The normal component of these traction conditions is overridden by the imposition of the normal component of velocity. A Poisson equation for pressure may be derived, if desired, by taking the divergence of the momentum equation (1a). The appropriate boundary condition for pressure (see Gresho and Sani 1987) is then obtained by taking the normal component of (1a), this yielding a Neumann boundary condition for pressure. This only serves to define the pressure field up to an arbitrary additive constant, the pressure is thus fixed by defining it to be zero at an arbitrary node. The initial condition on the velocity field is given by an imposed field $\boldsymbol{u}_0(\boldsymbol{x})$, which for well-posedness must satisfy both continuity (1b) and the boundary conditions imposed on the problem. Since the normal component of velocity is imposed on the entire boundary in this work the additional compatibility constraint of global mass conservation must be satisfied:

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

where Γ represents the boundary of Ω.

b. Basis functions and mixed formulation for a standard Galerkin finite-element model

In a finite-element formulation, the infinitely many degrees of freedom of the continuous equations (1) are reduced to a finite number by representing each of the variables as a sum over a finite set of spatial basis functions. The basis functions are defined on elements, as defined in the introduction. In the present model, hexahedral finite elements have been employed. Examples of basis functions that can be used with this choice include piecewise-linear and piecewise-constant functions, both of which are defined in detail later.

At this point, increased flexibility and generality is achieved by allowing pressure to be represented using a different basis from the velocity and temperature, additional reasons for this shall be explained presently. As such, numerical approximations to the velocity and temperature are represented using a finite set of basis functions $\{N_i\}$:

$$\begin{aligned} \boldsymbol{u} &\equiv (u, v, w)^{\mathrm{T}} \\ &\approx \sum_{j=1}^{N} [\mathbf{u}_j(t), \mathbf{u}_{N+j}(t), \mathbf{u}_{2N+j}(t)]^{\mathrm{T}} N_j(\mathbf{x}), \\ T &\approx \sum_{j=1}^{N} \mathbf{T}_j(t) N_j(\mathbf{x}), \end{aligned} \tag{4}$$

while the pressure is represented using a different basis $\{M_i\}$:

$$p \approx \sum_{j=1}^{M} \mathbf{p}_j M_j(\mathbf{x}). \tag{5}$$

Thus $\mathbf{u}$ is a vector of length $3N$, while $\mathbf{T}$ is a vector of length N, and $\mathbf{p}$ is a vector of length M. This form for $\boldsymbol{u}$, T, and p is substituted into the momentum equation (1a). A *Galerkin projection* of that equation is then taken by multiplying by each basis function $N_i(\mathbf{x})$ in turn and applying Green's theorem where appropriate to reduce the regularity required of basis functions. This leads to a discretization of the following form:

$$\mathsf{M}\frac{d\mathbf{u}}{dt} + [\mathsf{N}(\mathbf{u}) + \mathsf{L} - \mathsf{K}]\mathbf{u} = \mathsf{C}\mathbf{p} + \mathsf{G}\mathsf{T}'. \tag{6}$$

In (6) the $3N \times 3N$ *mass matrix* M is given by

$$\mathsf{M} \equiv \begin{pmatrix} \mathsf{M}' & 0 & 0 \\ 0 & \mathsf{M}' & 0 \\ 0 & 0 & \mathsf{M}' \end{pmatrix}, \tag{7}$$

where

$$M'_{ij} = \int_\Omega dV\, N_i(\mathbf{x}) N_j(\mathbf{x}), \quad \text{for } 1 \leq i, j \leq N.$$

The Coriolis term is represented by

$$\mathsf{L} \equiv f \begin{pmatrix} 0 & -\mathsf{M}' & 0 \\ \mathsf{M}' & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{8}$$

which may be generalized for a more complex rotation vector. Buoyancy forcing is represented by G and T', which are given by

$$\mathsf{G} \equiv \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \mathsf{M}' \end{pmatrix}, \qquad \mathsf{T}' \equiv \begin{pmatrix} 0 \\ 0 \\ \mathbf{T} \end{pmatrix}. \tag{9}$$

Discretization of the nonlinear advection terms results in the matrix $\mathsf{N}(\tilde{\mathbf{u}})$ given by

$$\mathsf{N}(\tilde{\mathbf{u}}) \equiv \begin{bmatrix} \mathsf{N}'(\tilde{\mathbf{u}}) & 0 & 0 \\ 0 & \mathsf{N}'(\tilde{\mathbf{u}}) & 0 \\ 0 & 0 & \mathsf{N}'(\tilde{\mathbf{u}}) \end{bmatrix}, \tag{10}$$

where

$$N'_{ij}(\tilde{\mathbf{u}}) = \int_\Omega dV\, N_i(\mathbf{x}) \tilde{\mathbf{u}} \cdot \boldsymbol{\nabla} N_j(\mathbf{x}), \quad \text{for } 1 \leq i,j \leq N. \tag{11}$$

The matrix K in (6) that represents a discretization of the viscous terms and for simplicity is not given here; various details may be found in Gresho and Sani (1998). Finally, $-\mathsf{C}\mathbf{p}$ is a discretization of the pressure gradient term in (1a), where

$$\mathbf{C} \equiv \begin{pmatrix} \mathbf{C}^1 \\ \mathbf{C}^2 \\ \mathbf{C}^3 \end{pmatrix}, \qquad (12)$$

and, for $1 \leq i \leq N$; $1 \leq j \leq M$; $q = 1, 2, 3$,

$$C^q_{ij} = \int_\Omega dV\, M_j(\mathbf{x}) \frac{\partial N_i(\mathbf{x})}{\partial x_q} - \int_\Gamma d\Gamma\, M_j(\mathbf{x}) N_i(\mathbf{x}). \qquad (13)$$

Green's theorem has been applied here due to the discontinuous nature of the basis $\{M_i\}$. The continuity equation (1b) is also discretized using a Galerkin projection, but now the pressure basis functions $M_i(\mathbf{x})$ must be used to form the projection. The result is that the discrete continuity equation takes the form

$$\int_\Omega dV\, M_i \nabla \cdot \mathbf{u} = 0, \quad \text{for } 1 \leq i \leq M. \qquad (14)$$

The boundary condition of no normal flow is added to (14), to give

$$\int_\Omega dV\, M_i \nabla \cdot \mathbf{u} - \int_\Gamma d\Gamma\, M_i \mathbf{n} \cdot \mathbf{u} = 0,$$
$$\text{for } 1 \leq i \leq M, \qquad (15)$$

where $\mathbf{n}$ is the unit outward normal to the boundary. This may be written in the following matrix form, where $\mathbf{C}$ is precisely as defined in (13):

$$\mathbf{C}^{\mathrm{T}}\mathbf{u} = \mathbf{0}. \qquad (16)$$

The fact that the same matrix $\mathbf{C}$ forms $\mathbf{Cp}$ for the pressure terms in (6) and $\mathbf{C}^{\mathrm{T}}\mathbf{u}$ for the continuity condition (16) will prove important later in deriving an equation for $\mathbf{p}$, and hence separating the velocity time step from the solution for pressure.

In principle, one could now attempt to integrate the system forward in time, and a natural choice might seem to be to represent all the fields using piecewise-linear basis functions. However, it is well-known that, if the velocities and pressure are all represented using equal-order interpolants then it is not possible, in general, to find a solution for $\mathbf{p}$ that ensures that the velocity at the end of the time step satisfies (16); correspondingly, if a discrete pressure equation is formulated, then that equation is in general singular. This is due to the intimate coupling of velocity and pressure through the incompressibility constraint (1b) and (16). Consequently, since the use of an inappropriate selection of basis functions may lead to numerical instabilities, this issue is termed *LBB stability* following (Ladyzhenskaya 1969; Babuška 1971; Brezzi 1974), and a compatibility condition between the pair of discrete spaces used to define velocity and pressure, which ensures LBB stability, is known as the the *inf-sup condition.* Therefore, to represent the variables in (1), the procedure of using the piecewise-linear basis $\{N_i\}$ for velocity and temperature, and the piecewise-constant basis $\{M_j\}$ for pressure is adopted. This discussion may be thought of as the analog of staggering [or choice of Arakawa grid (Arakawa 1966)] in a finite-difference procedure.

Now, although this choice (termed Q_1P_0) does not satisfy the inf-sup condition, and is therefore not strictly LBB stable, implying limitations from a theoretical point of view, it does work well in practice and is often used in engineering applications. See Griffiths and Silvester (1994) where it is commented that it ''invariably works well in practice,'' and also Gresho and Sani (1998) for a detailed discussion of this element where it is termed an ''often-excellent and cost-effective'' choice [see also Girault and Raviart (1986) and Brezzi and Fortin (1991)]. Care must be taken with its use due to the presence of spurious pressure modes. An example is the *hydrostatic* mode (the use of hydrostatic here is standard and should not be confused with its use later in this work) that occurs here due to the choice of velocity boundary conditions, that is, the arbitrary additive constant mentioned in section 2a, where an effective technique for removing this mode is also given. *Checkerboard* modes are not an issue here since the boundary conditions employed do not support them. Finally, *Pesky* modes that have nonzero $\mathcal{O}(h)$ or smaller LBB constants (as opposed to zero LBB constants for the previous two modes), where h is an indication of the mesh spacing, may affect both solution accuracy as well as the conditioning of the linear systems that must be solved, see (Griffiths and Silvester 1994; Griffiths 1996). For the applications considered here this was not found to be a serious issue. Additionally, due to the desire for mesh adaptivity the model will ultimately make use of tetrahedral elements. The choice of element pair will then be based both upon stability issues as well as the pair's ability to represent well the solution characteristics present in oceanographic applications. Hence, in this work comparisons with LBB stable element pairs, or stabilized versions of the Q_1P_0 choice have not been pursued.

The spatial basis functions mentioned previously, along with a discontinuous piecewise-linear basis $\{Q_k\}$ of use later, are defined as follows, where (ξ_1, ξ_2, ξ_3) represent local coordinates on a hexahedral element, in which each of the ξ_i runs from -1 to 1 along the edges of an element.

1) Continuous piecewise-linear basis functions, $\{N_i\}$. These functions are equal to unity at vertex i, are equal to zero at all neighboring vertices, and take a trilinear form throughout each element; for example, if the vertex number i corresponds to the vertex $(\xi_1, \xi_2, \xi_3) = (-1, -1, -1)$ in a given element, then N_i takes the trilinear form $(1 - \xi_1)(1 - \xi_2)(1 - \xi_3)/8$ throughout the element. The basis functions $\{N_i\}$ vanish outside of the elements that have vertex i as one of their vertices. Let N denote the total number of these basis functions defined on the mesh.
2) Discontinuous piecewise-linear basis functions, $\{Q_k\}$. These functions take the same trilinear spatial

form as the functions $\{N_i\}$ within the elements. However, whereas the functions $\{N_i\}$ are continuous, and nonzero in all elements that have vertex i as one of their vertices, the discontinuous basis functions Q_k are defined such that they are nonzero in only one element. Therefore, with the exception of elements at the boundary of the domain, there are eight discontinuous basis functions Q_k for every continuous basis function N_i. Let Q denote the total number of these defined on the mesh.

3) Piecewise-constant basis functions, $\{M_j\}$. These functions are equal to unity throughout element j, and equal to zero in all other elements. Let M denote the total number of these defined on the mesh.

c. *Solution method for velocity and pressure*

In principle, the momentum and continuity equations can now be assembled into a large coupled nonlinear system for $\mathbf{u}^{n+1}$, $\mathbf{p}^{n+1/2}$, and $\mathbf{T}^{n+1}$ as a function of $\mathbf{u}^n$, $\mathbf{p}^{n-1/2}$, and $\mathbf{T}^n$ [where superscripts represent discrete time levels following a temporal discretization of (6), here the velocities and temperatures $\mathbf{u}^n$, $\mathbf{T}^n$, and $\mathbf{u}^{n+1}$, $\mathbf{T}^{n+1}$ are held at times $t = t_n$ and $t = t_{n+1}$, while the pressure $\mathbf{p}^{n+1/2}$ is held at the midpoint of the time step, $t = t_{n+1/2} := \frac{1}{2}(t_n + t_{n+1})$]. This approach, however, results in an extremely large system of equations to solve, for which an efficient solution strategy is difficult to devise. Therefore, a technique is used in which pressure and velocity are solved for independently, thus reducing the total dimension of the systems that must be solved.

In order to avoid computational drift, it is important that the velocity at the end of each time step be divergence free or incompressible in a discrete sense, that is, it must satisfy (16). First, a time stepping scheme is constructed that is accurate to first order in time, this scheme may then be adapted to one that is accurate to second order in time. The computed velocity field is then projected onto a divergence-free subspace of the discrete space in which it is constructed. The overall approach taken here of solving separately for velocity and pressure is an example of what is known variously as a projection method, fractional-step method, pressure correction method, or Chorin's method (Chorin 1968).

A time stepping scheme that is first-order accurate in time is defined, as follows. In (6), the time derivative is approximated by

$$\frac{d\mathbf{u}}{dt} \approx \frac{\mathbf{u}^{n+1} - \boldsymbol{u}^n}{\Delta t}, \tag{17}$$

where n is the time level and Δt is the time step size. Also take

$$\mathsf{L}\mathbf{u} \approx \frac{1}{2}L(\mathbf{u}^n + \mathbf{u}^{n+1}), \tag{18}$$

which means that the Coriolis terms are treated implicitly with second-order accuracy across the time step, the viscous terms $K\mathbf{u}$ are treated analogously. The nonlinear advection terms are also treated implicitly, but only to first-order accuracy:

$$\mathsf{N}(\tilde{\mathbf{u}})\mathbf{u} \approx \mathsf{N}(\mathbf{u}^n)\left(\frac{\mathbf{u}^{n+1} + \mathbf{u}^n}{2}\right). \tag{19}$$

Picard iteration may be performed if desired, where in subsequent iterations $\tilde{\mathbf{u}} = (\mathbf{u}^n + \hat{\mathbf{u}}^{n+1})/2$ is used in the nonlinear matrix $\mathsf{N}(\tilde{\mathbf{u}})$, where $\hat{\mathbf{u}}^{n+1}$ is an approximation to the new velocity following the previous iterate (more precisely the divergence-free projection of $\mathbf{u}_*^{n+1}$, as defined presently). This process may be continued to convergence if desired, where it is equivalent (up to the choice of linearization of the nonlinear convective term) to the second-order Crank–Nicolson time stepping algorithm (Heywood and Rannacher 1990; Smith and Silvester 1997). Although in practice only two steps of this Picard iteration are required here to achieve second-order accuracy.

Thus, using (17)–(19), the discretized momentum equation (6) may be expressed in the form

$$\left(\mathsf{M} + \frac{1}{2}\Delta t\mathsf{A}\right)\left(\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t}\right) = -\mathsf{A}\mathbf{u}^n + \mathbf{f}^{\,n+1/2} + \mathsf{C}\mathbf{p}^{n+1/2}, \tag{20}$$

where the matrix $\mathsf{A} \equiv \mathsf{A}(\tilde{\mathbf{u}}) = \mathsf{N}(\tilde{\mathbf{u}}) + \mathsf{L} - \mathsf{K}$ does not depend on $\mathbf{u}^{n+1}$. The vector $\mathbf{f}$ contains the buoyancy forcing $\mathsf{G}\mathsf{T}'$, constructed from $\mathbf{T}^{n+1}$ which, like pressure, and due to time staggering being employed is taken to represent temperature at time $t = t_{n+1/2}$, [a discretization of the temperature equation (1c) is given in section 2e].

Now suppose that, as a first approximation to $\mathbf{p}^{n+1/2}$, the pressure $\mathbf{p}^{n-1/2}$ from the previous time step is used. This is a valid approximation to obtain first-order accuracy in time. A first approximation to $\mathbf{u}^{n+1}$ is then obtained by solving

$$\left(\mathsf{M} + \frac{1}{2}\Delta t\mathsf{A}\right)\left(\frac{\mathbf{u}_*^{n+1} - \mathbf{u}^n}{\Delta t}\right) = -\mathsf{A}\mathbf{u}^n + \mathbf{f}^{\,n+1/2} + \mathsf{C}\mathbf{p}^{n-1/2} \tag{21}$$

for $\mathbf{u}_*^{n+1}$. In general, $\mathbf{u}_*^{n+1}$ will not satisfy the discrete continuity equation (16), but it is nonetheless an approximation to $\mathbf{u}^{n+1}$ that is accurate to first order in time.

A *projection method* is now used to modify the velocity and pressure. The modified velocity will satisfy (16), and an updated estimate of the pressure can be obtained. The correction to the velocity is irrotational, and is obtained by supposing that a modified pressure acts over the interval $t_n < t < t_{n+1}$.

In the continuous limit, a Helmholtz decomposition (see, e.g., Girault and Raviart 1986) of the approximate solution $\mathbf{u}_*^{n+1}$ into divergence-free and irrotational parts is made. The correction step can then be written as

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}_*^{n+1} - \boldsymbol{\nabla}\varphi \tag{22}$$

(notice that vorticity is preserved by this operation). The corresponding discrete form of (22) is

$$\mathbf{M}\mathbf{u}^{n+1} = \mathbf{M}\mathbf{u}_{*}^{n+1} + \mathbf{C}\boldsymbol{\varphi}, \tag{23}$$

where ϕ is the discrete representation of ϕ.

Moreover, it is evident that, if two velocities differ at the end of a time step only by an irrotational component then, all other forcing terms being equal, this difference can be obtained by a difference in pressure acting throughout the time step between the two.

Now, to determine $\boldsymbol{\varphi}$, (23) is used, requiring that $\mathbf{u}^{n+1}$ satisfies (16). Multiplying (23) by $\mathbf{C}^{\mathrm{T}}\mathbf{M}^{-1}$, and applying (16), yields the following discretized elliptic problem for $\boldsymbol{\varphi}$:

$$\mathbf{C}^{\mathrm{T}}\mathbf{M}^{-1}\mathbf{C}\boldsymbol{\varphi} = -\mathbf{C}^{\mathrm{T}}\mathbf{u}_{*}^{n+1}. \tag{24}$$

Using the solution $\boldsymbol{\varphi}$ from (24), $\mathbf{u}^{n+1}$ may be obtained from (23).

Finally, the correction to the pressure that is implied by this velocity projection must be determined. The correction to the velocity obtained by this approach is sufficient to preserve up to second-order accuracy in time (E and Liu 1996; Shen 1996; Gresho 1990), and some value of the pressure can in principle be found to render the entire scheme second-order accurate in time (see, e.g., Brown et al. 2001). However, in the implementation described above, the pressure is only required to first order. To obtain such a correction Δp to the pressure consider the continuous system, for which the following must hold

$$\frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}_{*}^{n+1}}{\Delta t} \equiv \frac{\boldsymbol{\nabla}\varphi}{\Delta t} = -\boldsymbol{\nabla}\Delta p, \tag{25}$$

and hence the correction to the pressure must be

$$\Delta p = -\frac{\varphi}{\Delta t}. \tag{26}$$

Equivalently, in the discrete system, the pressure is updated via

$$\mathbf{p}^{n+1/2} = \mathbf{p}^{n-1/2} - \frac{\boldsymbol{\varphi}}{\Delta t}. \tag{27}$$

As previously mentioned, a second iteration of the entire procedure outlined above results in an overall scheme which is [due to the more accurate handling of the advective and pressure gradient terms in (21)] formally second-order accurate in time and space for velocity, see also Heywood and Rannacher (1982) and Heywood and Rannacher (1990). Note that the semi-implicit treatment presented in this section allows the stable use of time steps up to a Courant number ($u\Delta t/h$) of order $O(10)$, dependent on the problem being solved (Gresho and Sani 1998).

d. Mass lumping of the continuity equation

The procedure described above must be modified in order to provide a computationally efficient scheme. The problem lies in Eq. (24), which must be solved by an iterative procedure. Here, $\mathbf{C}$ and $\mathbf{M}$ are sparse matrices, but $\mathbf{M}^{-1}$ is not. Now, given a linear system of the form $\mathbf{P}\mathbf{z} = \mathbf{q}$, where $\mathbf{P}$ is a sparse matrix, it is not necessary to form the matrix $\mathbf{P}$, but it must be possible to multiply a given vector by it. In the case of (24), this requires multiplication by $\mathbf{M}^{-1}$; each single iteration in the solution of (24) would then require an iterative solution of a linear system with matrix $\mathbf{M}$. While this may be possible without substantially increasing the storage required, it will lead to a method that requires a large amount of computational time for even a single time step.

The time required to take a time step could be substantially reduced, however, if only the *lumped* mass matrix $\mathbf{M}_l$ could be used in place of $\mathbf{M}$ in Eq. (24). The lumped mass matrix is defined by

$$M_{lii} := \sum_{j=1}^{3N} M_{ij}, \quad \text{and}$$
$$M_{lij} := 0 \quad \text{for } i \neq j, \qquad 1 \leq i, j \leq 3N. \tag{28}$$

In other words, $\mathbf{M}_l$ is a diagonal matrix, with each diagonal entry being equal to the sum of the entries in the corresponding row of $\mathbf{M}$. Following through the details of section 2c, it can be seen that this is possible provided that the discrete continuity equation (16) is replaced by

$$\mathbf{C}^{\mathrm{T}}\mathbf{M}_l^{-1}\mathbf{M}\mathbf{u} = 0. \tag{29}$$

Analysis shows that this makes errors $O(\Delta x^4)$ locally, and that global mass conservation is preserved.

A modified form of (24) can now be formed by multiplying (23) by $\mathbf{C}^{\mathrm{T}}\mathbf{M}_l^{-1}$, and requiring that $\mathbf{u}^{n+1}$ satisfies (29). The result is that $\boldsymbol{\varphi}$ must satisfy

$$\mathbf{C}^{\mathrm{T}}\mathbf{M}_l^{-1}\mathbf{C}\boldsymbol{\varphi} = -\mathbf{C}^{\mathrm{T}}\mathbf{M}_l^{-1}\mathbf{M}\mathbf{u}_{*}^{n+1}. \tag{30}$$

With this modified equation for $\boldsymbol{\varphi}$, the remaining equations (23) for $\mathbf{u}^{n+1}$ and (27) for $\mathbf{p}^{n+1/2}$ remain unchanged. In particular, $\mathbf{M}$ is not replaced by $\mathbf{M}_l$ in (23)—indeed it would be inconsistent with (29) to do so, since these two equations are used to derive (30). Since (23) is solved only once per time step (or twice in the case of a second-order scheme), the computational resources required are kept within more reasonable bounds. Importantly, the consistent mass matrix being retained in the momentum equations leads to the preservation of accuracy properties, in particular the dispersion errors are far lower than would be the case if the lumped mass was used throughout (Gresho and Chan 1990).

At the beginning of a simulation the pressure field necessary to start the approach outlined above is not immediately at hand. Therefore, equations are now derived for velocity and pressure, which may be solved by iteration at the first time step. Subsequent time steps may then follow the projection method approach given in the previous section. Multiplying the discretized momentum equation (20) by $\mathbf{C}^{\mathrm{T}}\mathbf{M}_l^{-1}$ gives

$$\left(\mathbf{C}^{\mathrm{T}}\mathbf{M}_l^{-1}\mathbf{M} + \frac{1}{2}\Delta t\mathbf{C}^{\mathrm{T}}\mathbf{M}_l^{-1}\mathbf{A}\right)\left(\frac{\mathbf{u}^{n+1} - \mathbf{u}^{n}}{\Delta t}\right) = \mathbf{C}^{\mathrm{T}}\mathbf{M}_l^{-1}(-\mathbf{A}\mathbf{u}^{n} + \mathbf{f}^{n+1/2}) + \mathbf{C}^{\mathrm{T}}\mathbf{M}_l^{-1}\mathbf{C}\mathbf{p}^{n+1/2}. \quad (31)$$

(This represents a discrete form of the divergence of the momentum equations, and will presently yield a discrete elliptic equation for pressure). Equations (31) and (20) can be used as an iterative scheme for $\mathbf{p}^{n+1/2}$ and $\mathbf{u}^{n+1}$. Given $\mathbf{u}^n$, a successive set of estimates $\mathbf{u}_*^{n+1}$ for $\mathbf{u}^{n+1}$ and $\mathbf{p}_*^{n+1/2}$ for $\mathbf{p}^{n+1/2}$ are obtained as follows.

First, set $\mathbf{u}_*^{n+1} = \mathbf{u}^n$. Second, use (29), with $\mathbf{u} = \mathbf{u}_*^{n+1}$, to derive the following equation from (31):

$$-\frac{1}{\Delta t}\mathbf{C}^{\mathrm{T}}\mathbf{M}_l^{-1}\mathbf{M}\mathbf{u}^{n} + \frac{1}{2}\Delta t\mathbf{C}^{\mathrm{T}}\mathbf{M}_l^{-1}\mathbf{A}\left(\frac{\mathbf{u}_*^{n+1} - \mathbf{u}^{n}}{\Delta t}\right) = \mathbf{C}^{\mathrm{T}}\mathbf{M}_l^{-1}(-\mathbf{A}\mathbf{u}^{n} + \mathbf{f}^{n+1/2}) + \mathbf{C}^{\mathrm{T}}\mathbf{M}_l^{-1}\mathbf{C}\mathbf{p}_*^{n+1/2}. \quad (32)$$

Solve (32) to obtain $\mathbf{p}_*^{n+1/2}$. Third, solve

$$\left(\mathbf{M} + \frac{1}{2}\Delta t\mathbf{A}\right)\left(\frac{\mathbf{u}_*^{n+1} - \mathbf{u}^{n}}{\Delta t}\right) = -\mathbf{A}\mathbf{u}^{n} + \mathbf{f}^{n+1/2} + \mathbf{C}\mathbf{p}_*^{n+1/2} \quad (33)$$

to obtain a new estimate for $\mathbf{u}_*^{n+1}$.

The second and third steps can then be repeated to obtain a succession of approximations: $\mathbf{p}_*^{n+1/2}$ and $\mathbf{u}_*^{n+1}$ to $\mathbf{p}^{n+1/2}$ and $\mathbf{u}^{n+1}$.

In the model as presented here, this approach is used to take only the first time step (i.e., at $n = 0$). Both (32) and (33) are solved by iterative methods. The velocity $\mathbf{u}_*^{n+1}$ will satisfy (29), to the accuracy of the iterative solution, only in the limit of infinitely many iterations. For practical reasons, only one iteration of this procedure is taken, followed by an application of the projection method as described in section 2c, to ensure that $\mathbf{u}^{n+1}$ satisfies (29).

e. Discretization of the temperature equation

The temperature equation (1c) is discretized in time using the Crank–Nicolson method, and in space using a standard Galerkin approach. The approximated temperature is represented as

$$T(\mathbf{x}, t^n) \approx \sum_{j=1}^{N} N_j(\mathbf{x})T_j^n. \quad (34)$$

Upon assuming homogeneous Neumann boundary conditions of the form $\mathbf{n}\cdot(\boldsymbol{\kappa}\boldsymbol{\nabla}T) = 0$ (i.e., no heat flux), the Crank–Nicolson time stepping scheme then takes the form

$$\left(\mathbf{M}' + \frac{1}{2}\Delta t\mathbf{H}\right)\mathbf{T}^{n+1} = \left(\mathbf{M}' - \frac{1}{2}\Delta t\mathbf{H}\right)\mathbf{T}^{n}, \quad (35)$$

where $\mathbf{M}'$ is defined in (7) and the matrix $\mathbf{H}$ is defined by

$$H_{ij} = \int_{\Omega} dV\left(N_j\mathbf{u}^n\cdot\boldsymbol{\nabla}N_i + \kappa_{lm}\frac{\partial N_j}{\partial x_m}\frac{\partial N_i}{\partial x_l}\right), \quad \text{for } 1 \le i, j \le N.$$

As mentioned earlier, since $\mathbf{u}^n$ is used in this calculation it is appropriate to consider $\mathbf{T}^{n+1}$ to be staggered with respect to velocities and thus to represent temperature at a time $t = t_{n+1/2}$. It follows then that second-order temporal accuracy of the overall method is not compromised.

f. Rotations and boundaries

For boundaries that are not aligned with the coordinate system and at which no normal flow and slip/partial slip boundary conditions are to be applied, it is necessary to rotate the velocities to a system aligned with the boundary. For each node on the boundary an associated 3 × 3 rotation matrix can be defined. These are collected together to form a $3N \times 3N$ rotation matrix $\mathbf{R}$ for the whole system of equations. The matrix $\mathbf{R}$ has the orthogonality property that $\mathbf{R}\mathbf{R}^{\mathrm{T}} = \mathbf{I}$. This must be done with care where the sea floor meets the sea surface (at coastal boundaries) because two components of the rotated velocity $\tilde{\mathbf{u}}^n := \mathbf{R}\mathbf{u}^n$ must be set to zero.

In addition, to avoid singularities in the discretized equations for certain limiting cases, the coordinate system is also rotated so that it is aligned with the new system. This is simply achieved by pre- and post-multiplication of the discretization matrices by the rotation matrices. The discretized momentum equation (20) then becomes

$$\mathbf{R}\left(\mathbf{M} + \frac{1}{2}\Delta t\mathbf{A}\right)\mathbf{R}^{\mathrm{T}}\mathbf{R}\left(\frac{\mathbf{u}^{n+1} - \mathbf{u}^{n}}{\Delta t}\right) = -\mathbf{R}\mathbf{A}\mathbf{R}^{\mathrm{T}}\mathbf{R}\mathbf{u}^{n} + \mathbf{R}\mathbf{f}^{n+1/2} + \mathbf{R}\mathbf{C}\mathbf{p}^{n+1/2}, \quad (36)$$

and the continuity equation (29) becomes

$$(\mathbf{R}\mathbf{C})^{\mathrm{T}}\mathbf{R}\mathbf{M}_l^{-1}\mathbf{R}^{\mathrm{T}}\mathbf{R}\mathbf{M}\mathbf{R}^{\mathrm{T}}\mathbf{R}\mathbf{u}^{n+1} = 0. \quad (37)$$

Now using $\tilde{\mathbf{u}}^n := \mathbf{R}\mathbf{u}^n$ and writing $\mathbf{M} \equiv \mathbf{R}\mathbf{M}\mathbf{R}^{\mathrm{T}}$, $\mathbf{A} \equiv \mathbf{R}\mathbf{A}\mathbf{R}^{\mathrm{T}}$, $\mathbf{C} \equiv \mathbf{R}\mathbf{C}$, $\mathbf{M}_l \equiv \mathbf{R}\mathbf{M}_l\mathbf{R}^{\mathrm{T}}$, $\tilde{\mathbf{f}}^{n+1/2} \equiv \mathbf{R}\mathbf{f}^{n+1/2}$, Eqs. (36) and (37) take the form (20) and (29), respectively. The solution methods described previously are then applied to this transformed system. In order to satisfy global mass conservation exactly, the normal to the boundary at mesh nodes must be calculated correctly. Following Engelman et al. (1982) this is achieved for an arbitrary mesh at node j by defining

$$\mathbf{n}_j := \frac{1}{n^*}\left(\sum_i \hat{C}_{ij}^1, \sum_i \hat{C}_{ij}^2, \sum_i \hat{C}_{ij}^3\right)^{\mathrm{T}}, \quad (38)$$

where the matrices $\hat{\mathbf{C}}^k$ are defined via Eq. (13) with the boundary integral removed, and n^* is a scalar that simply normalizes $\mathbf{n}_j$.

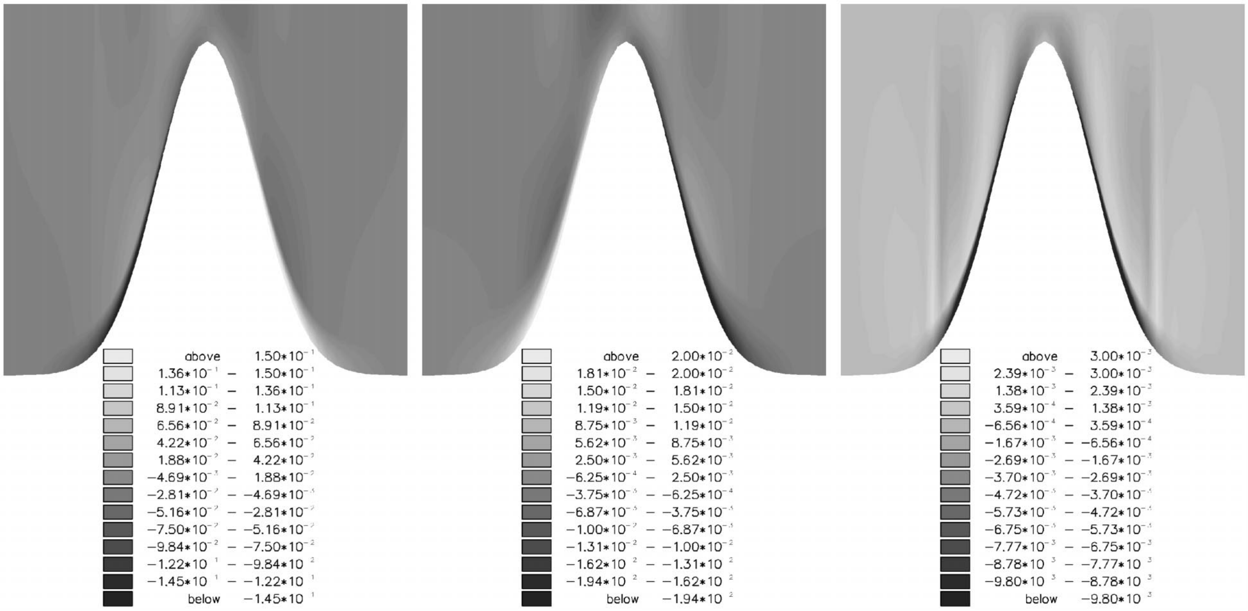

FIG. 1. (left to right, respectively) Spurious velocities *u, v, w* induced by pressure gradient errors in the standard Galerkin finite-element model.

g. *Solution of linear systems*

The symmetric-positive-definite matrix equations (30) and (32) are solved using conjugate gradients (Golub and van Loan 1989) with nine-step parameterized forward–backward Gauss–Seidel preconditioning (Adams 1983) within each subdomain in a parallel computation (three steps are used for serial computations). A large number of steps is used so that each processor does a considerable quantity of work before communicating the results (communication can be relatively real-time intensive). Since the quantity of time spent on each of these sweeps is predictable it is relatively easy to balance the load on each processor by, for example, assigning approximately equal numbers of pressure nodes to each processor. The block nonsymmetric matrix equations (21) and (33) are solved using restarted Generalized Minimal Residual (GMRES; Saad and Schultz 1986), typically here using 20 orthonormal vectors and a block symmetric successive over-relaxation (SSOR) preconditioner within each subdomain. The temperature equation (35) is solved in an analogous manner.

3. Pressure gradient errors

A standard test for pressure gradient errors in a numerical model is to initialize with a stable density profile that depends only on the vertical direction *z,* and to impose no inflow or outflow at boundaries. The model is then integrated for a few time steps, and any velocity induced by the discretization must be considered to be model error. It is well-known, for example, that sigma-coordinate models in the presence of topography may produce large spurious pressure gradients (Haney 1991; Mellor et al. 1998). In this section, the behavior of the ‘‘standard’’ model is investigated when it is initialized with a linear density profile, a Burger number of 1.5 is employed here. The Burger number is a measure of the strength of stratification and is defined as

$$B \equiv \frac{NH}{fL},$$

where H is the depth of the ocean, L is a measure of the horizontal scale, and the Brunt–Väisälä frequency N (not to be confused with the notation for the number of basis functions introduced in section 2b) is defined here via

$$N^2 \equiv -\frac{g\Delta_z\rho}{H\rho_0},$$

where $\Delta_z\rho$ is the density difference in the vertical between the sea surface and sea floor, and ρ_0 is a reference value for density (see, e.g., Turner 1973; Pedlosky 1987). The problem geometry is that of a channel 560 km long, 514 km wide, and 4.5 km deep. The topography is given by an isolated Gaussian seamount of maximum height 90% that of the channel, and a full-width at half maximum value of 41.63 km. A sigma-coordinate hexahedral mesh is used with 12 equally spaced levels in the vertical, and a mesh spacing of 3.125 km in the horizontal.

In Fig. 1, the three components of velocity are presented in a vertical section along the channel passing through the center of the seamount. The time step is Δt = 432 s, and in Fig. 1 the fields are shown after 10 time steps.

Although the standard finite-element model has a formally second-order accurate spatial discretization, the magnitude of errors does depend upon the distortion of elements. As the elements become distorted, computational accuracy reduces. Moreover and importantly, the predominant balance that describes the system under simulation is that of hydrostatic balance between the temperature (or density) and the vertical pressure gradient. Now, the temperature is represented (34) using the piecewise-linear basis $\{N_i\}$, and so hydrostatic balance,

$$\frac{\partial p}{\partial z} = -g\rho \equiv gT, \tag{39}$$

can in general only be satisfied exactly if the pressure is represented using basis functions that are of quadratic or higher order. However, as already discussed in section 2b, in order for the pressure equation to be nonsingular, the basis used for the pressure must generally be in fact of *lower* order than that used for velocity and temperature. Recall that here the piecewise-constant basis $\{M_j\}$ is used. Therefore, the model cannot satisfy the discrete version of (39) exactly, and hence the pressure required to balance the buoyancy forces in a nearly hydrostatic regime is not well approximated.

The ‘‘mixed’’ approach does work very well in many applications, but here it has been found that it is unsuitable for fluids in which the dynamics are close to hydrostatic balance. The result is that extremely large spurious velocities are generated on unstructured or distorted meshes, such as those required to accurately conform to coastlines or topography.

For example, in the model integration presented above horizontal pressure gradients are established that would not be present in the continuous system. These pressure gradients cause the fluid to come into motion, as shown in Fig. 1. The relatively large magnitude of the velocities induced indicates that this discretization is unsuitable for numerical ocean modeling. Of course, the discretization is consistent in the limit of small mesh intervals, but the aim must be to obtain ‘‘reasonable’’ results at ‘‘moderate’’ resolution. Although these are qualitative statements about what is practical with contemporary computational resources, it is certainly desirable to eliminate any discretization that can produce such large pressure gradient errors, even if the resolution is relatively modest.

To fully benefit from the use of finite elements and unstructured adaptive grids, the hydrostatic approximation should not be made *per force* in the model, however it should still be possible to simulate flows in regimes in which the fluid is near to hydrostatic balance over much of the computational domain, incorporating both coastlines and topography. A special treatment of the hydrostatic pressure is therefore required, which is presented in the following section.

4. A hydrostatic pressure solver

a. Formulation

Possible methods to deal with the pressure gradient errors in sigma-coordinate-type models include interpolation to z levels and spline-based reconstruction schemes (for a comparison see Ezer et al. 2002). In the present context, and based upon the discussion of the previous section, one approach to eliminating, or at least reducing, pressure gradient errors might be to use basis functions in which the hydrostatic relation (39) can hold exactly at the discrete level. While this is an interesting possibility, the implied hierarchy of basis functions for temperature and velocity is rather complex, and here an alternative simpler approach is proposed.

The overall pressure is split into two parts: a ‘‘hydrostatic pressure’’ p_h and a ‘‘nonhydrostatic pressure’’ $p_{\rm nh}$ so that the total pressure p is given at any point by $p = p_{\rm nh} + p_h$. A similar device was employed by Adcroft et al. (1997), but whereas their splitting was designed for computational efficiency, here the splitting results in pressures that are of a fundamentally different kind. The hydrostatic pressure p_h is required to balance the buoyancy force in the vertical momentum equation:

$$\frac{\partial p_h}{\partial z} = -\rho g. \tag{40}$$

The vertical momentum equation in (1a) is therefore

$$\frac{\partial w}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u}w) = -\frac{\partial p_{\rm nh}}{\partial z} + \frac{\partial \tau_{3j}}{\partial x_j} + F_3, \tag{41}$$

where $\mathbf{F} \equiv (F_1, F_2, F_3)^{\rm T}$. The horizontal momentum equations are now

$$\frac{\partial \boldsymbol{u}_i}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{u}\boldsymbol{u}_i) = -\frac{\partial p_h}{\partial x_i} - \frac{\partial p_{\rm nh}}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + F_i, \quad \text{for } i = 1, 2. \tag{42}$$

This means that, given ρ, it must be possible to evaluate p_h, or more specifically $\partial p_h/\partial x$ and $\partial p_h/\partial y$ for use in (42). In fact, $\partial p_h/\partial x$ and $\partial p_h/\partial y$ are solved for directly here. This is both more numerically robust than solving for p_h and then taking the horizontal derivatives, as well as resulting in an algorithm in which the balance (39) is consistent with regards to polynomial order. In Song (1998) and Iskandarani et al. (2003) the procedure of interchanging the order of integration and differentiation in the computation of baroclinic pressure gradients is shown to result in more accurate results for sigma coordinate hydrostatic models.

Using Eq. (40), equations for $\partial p_h/\partial x$ and $\partial p_h/\partial y$ are

$$\frac{\partial}{\partial z}\left(\frac{\partial p_h}{\partial x}\right) = -\frac{\partial \rho g}{\partial x}, \qquad \frac{\partial}{\partial z}\left(\frac{\partial p_h}{\partial y}\right) = -\frac{\partial \rho g}{\partial y}. \tag{43}$$

The required terms $\partial p_h/\partial x$ and $\partial p_h/\partial y$ are calculated by integrating (43) down from the top of the ocean at $z = 0$. Surface boundary conditions $\partial p_h/\partial x = 0$ and $\partial p_h/\partial y$

$= 0$ at $z = 0$ are employed. Consequently, the hydrostatic pressure p_h is not the same as the pressure that would exist in a hydrostatic ocean model with a rigid lid. Rather, p_h is the simplest pressure field consistent with (40). Even if the flow is nearly hydrostatic, the nonhydrostatic pressure p_{nh} is required to accommodate the surface pressure required by the condition of no normal flow at $z = 0$. In the case of nearly hydrostatic flow, p_{nh} is nearly independent of depth.

To achieve high spatial accuracy on unstructured meshes, the discontinuous Galerkin method (Johnson 1987; Cockburn et al. 2000) is used to obtain $\partial p_h/\partial x$ and $\partial p_h/\partial y$. The discontinuous Galerkin method is used extensively in the finite-element literature for transport equations of the form

$$\boldsymbol{\nabla} \cdot (\mathbf{U}\varphi) = S, \qquad (44)$$

where $\mathbf{U}$ is the transport velocity, φ is the quantity being transported, and S is a source. Here, the equations in (43) take the form (44) if $\mathbf{U} = (0, 0, -1)^{\mathrm{T}}$, $\varphi = (\partial p_h/\partial x, \partial p_h/\partial y)^{\mathrm{T}}$, and $S = g(\partial \rho/\partial x, \partial \rho/\partial y)^{\mathrm{T}}$.

The key to the discontinuous Galerkin method is to work with the discontinuous spatial basis functions $\{Q_k\}$. To obtain discretized equations, (44) is multiplied by Q_k and integrated over the element E in which Q_k is nonzero:

$$\int_E dV(\varphi \mathbf{U} \cdot \boldsymbol{\nabla} Q_k + SQ_k) = \int_{\Gamma_E} d\Gamma\, \mathbf{U} \cdot \mathbf{n}\varphi Q_k, \qquad (45)$$

where Γ_E represents the boundary of element E. Now, if $\mathbf{U} \cdot \mathbf{n} > 0$, then this corresponds to information flowing out of the element, and no further action is taken. However, if $\mathbf{U} \cdot \mathbf{n} < 0$, then the value of φ on the boundary from neighboring elements is imposed. This value is denoted by φ_{in}. This procedure is equivalent to one in which Green's theorem is not invoked, but boundary conditions are imposed weakly on inflow boundaries. Thus, the discontinuous Galerkin representation of (44) is, for $1 \leq k \leq Q$,

$$\int_E dV[Q_k \boldsymbol{\nabla} \cdot (\mathbf{U}\varphi) - Q_k S] + \int_{\{\Gamma_E : \mathbf{U} \cdot \mathbf{n} < 0\}} d\Gamma \mathrm{S}\, Q_k \mathbf{U} \cdot \mathbf{n}(\varphi_{\mathrm{in}} - \varphi) = 0, \qquad (46)$$

where $\{\Gamma_E\colon \mathbf{U} \cdot \mathbf{n} < 0\}$ represents the restriction of the boundary to that part on which $\mathbf{U} \cdot \mathbf{n} < 0$.

Here the direction of integration is down from the ocean surface, so $\mathbf{U} \cdot \mathbf{n} < 0$ wherever the normal to the element has a positive vertical component. In the interior of the fluid, information is transferred between neighboring elements according to (46). The faces at the surface of the ocean, $z = 0$, also satisfy this property, and here the boundary condition $\varphi_{\mathrm{in}} = 0$ is imposed. System (46) is solved element by element in a Gauss–Seidel iterative manner. Due to the ''flow of information'' from sea surface to bottom here on simple meshes, an ordering of the elements may be trivially found such that only one iteration is required. Thus, this hydrostatic pressure solver is very efficient and does not introduce a large computational overhead. Algorithms for obtaining efficient reorderings of the elements in the case of unstructured meshes shall be discussed in future work.

Once the horizontal gradients of p_h have been obtained in this way, the discrete versions of the momentum equations are formed. The same procedure is used as in section 2b, except that buoyancy forces are now absent from the vertical momentum equation: $\mathsf{G} = 0$. Instead, additional terms on the right-hand side of the horizontal momentum equations are introduced to represent the effects of the hydrostatic pressure gradients, and ensure consistency of the discretization.

Using the discontinuous Galerkin method, the hydrostatic pressure gradient has the form

$$\left(\frac{\partial p_h}{\partial x}, \frac{\partial p_h}{\partial x}\right)^{\mathrm{T}} = \sum_k Q_k(\mathbf{phx}_k, \mathbf{phy}_k)^{\mathrm{T}}. \qquad (47)$$

Now, the discrete momentum equation (6) is formed as a Galerkin projection by multiplying by the piecewise-linear basis functions. If this projection is applied directly to the hydrostatic pressure gradients, then the discrete momentum equation will take the form

$$\mathsf{M}\frac{d\mathbf{u}}{dt} + [\mathsf{N}(\mathbf{u}) + \mathsf{L} - \mathsf{K}]\mathbf{u} = \mathsf{C}\mathbf{p} + \mathsf{F}\mathbf{ph}. \qquad (48)$$

The solution vector $\mathbf{p}$ now represents just the nonhydrostatic pressure p_{nh}, and the term $\mathsf{F}\mathbf{ph}$ represents the hydrostatic pressure gradients, in which

$$\mathbf{ph} \equiv \begin{pmatrix} \mathbf{phx} \\ \mathbf{phy} \\ 0 \end{pmatrix} \quad \text{and} \qquad (49)$$

$$\mathsf{F} \equiv \begin{pmatrix} \mathsf{F}' & 0 & 0 \\ 0 & \mathsf{F}' & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad (50)$$

where

$$F'_{ij} = -\int_\Omega dV\, N_i(\mathbf{x})Q_j(\mathbf{x}).$$

In (48), again notice that the buoyancy term in the vertical has been replaced by additional source terms in the horizontal. It is important to note that, because there are eight elements of $\{Q_k\}$ for every element of the set $\{N_i\}$ (with the exception of elements that adjoin the boundary of the domain), the discontinuous Galerkin method results in approximations for both $\partial p_h/\partial x$ and $\partial p_h/\partial y$ that have many more nodal values than the number of elements. It has been found by experimentation that, due to the large number of degrees of freedom for the hydrostatic pressure gradients, direct Galerkin projection as above can lead to computational instabilities.

TABLE 1. Linear stratification with a Burger number $B = 1.5$. Spurious velocities observed in the pressure gradient error test for the standard and ocean models. The first column gives the height of the seamount in relation to the total depth of the domain, denoted δ. The second column gives the horizontal mesh spacing Δx, Δy and the number of levels used in the vertical V. The magnitudes of the maximum horizontal and vertical spurious velocities are given in columns three and four. The figures in parentheses represent exponents, i.e. 1.234(5) = 1.234×10^5.

δ	Resolution	Error in standard model	Error in ocean model
0.1	$\Delta x, \Delta y = 12.5, V = 12$	2.887(−2), 2.658(−3)	—
0.1	$\Delta x, \Delta y = 6.25, V = 12$	3.048(−2), 9.049(−4)	—
0.1	$\Delta x, \Delta y = 12.5, V = 24$	3.220(−2), 7.304(−3)	—
0.9	$\Delta x, \Delta y = 12.5, V = 12$	4.150(−1), 2.088(−1)	—
0.9	$\Delta x, \Delta y = 6.25, V = 12$	2.037(−1), 2.818(−2)	—
0.9	$\Delta x, \Delta y = 12.5, V = 24$	4.502(−1), 2.183(−1)	—

Therefore, for use in the discretized form of the horizontal momentum equations, the hydrostatic pressure gradients $\partial p_h/\partial x$ and $\partial p_h/\partial y$ obtained using the discontinuous Galerkin method as described, are projected onto the piecewise-constant basis $\{M_j\}$. The Galerkin projection is then taken to form the discrete momentum equations. The projection ($\mathbf{phx}^p$, $\mathbf{phy}^p$) of the hydrostatic pressure gradients onto the piecewise-constant basis is obtained by solving

$$\mathbf{Pphx}^p = \mathbf{Rphx}, \qquad \mathbf{Pphy}^p = \mathbf{Rphy}. \tag{51}$$

where

$$P_{ij} = \int_\Omega dV\, M_i(\mathbf{x})M_j(\mathbf{x}) \quad \text{and}$$

$$R_{ij} = \int_\Omega dV\, M_i(\mathbf{x})Q_j(\mathbf{x}).$$

The piecewise-constant basis functions are nonzero only within one element, therefore the matrix $\mathbf{P}$ is diagonal and can be inverted without iteration. These projected hydrostatic pressure gradients are then used in the horizontal momentum equations. The result is that $\mathbf{F}'$ in (50) is replaced by

$$\mathbf{F}' = \mathbf{SP}^{-1}\mathbf{R}, \tag{52}$$

where

$$S_{ij} = -\int_\Omega dV\, N_i(\mathbf{x})M_j(\mathbf{x}).$$

Hence, (6) in the standard numerical model is replaced with (48), with $\mathbf{F}$ defined by (50), but in which $\mathbf{F}'$ is defined by (52). To distinguish this from the standard model, it shall be referred to as the ‘‘ocean model.’’

b. Numerical experimentation

It can be seen immediately from the foregoing that if the temperature depends linearly on the vertical coordinate z and is independent of the horizontal coordinates x and y, then the model produces no pressure gradient errors whatsoever. This is because the piecewise-linear basis functions used for the temperature can represent such a field exactly on a skewed mesh. Consequently, the horizontal gradients of density used in the discrete form of (43) are identically zero, and so the horizontal pressure gradients are zero. The nonhydrostatic pressure p_{nh} is then also zero and so, if the fluid is at initially rest, it will remain at rest.

In Part II forced flows in which the initial stratification is linear shall be considered. However, as the integration proceeds and flow develops, this stratification is disturbed. It is therefore of interest to determine whether the ocean model is likely to be more accurate than the standard one for more general stratification.

To investigate this the test simulation from section 3 is repeated and both models are initialized with a quadratic temperature profile, with the same range of temperature as the linear profile, but with $\partial T/\partial z = 0$ at the sea floor away from the seamount. Both models were then integrated for 100 time steps of size $\Delta t = 432$ s and the maximum velocity in each was monitored over this time. There is flow in both models (confirming that the ocean model preserves a state of rest exactly only for a linear temperature profile) but the maximum velocity in the ocean model is approximately 2% of that in the standard model.

Further tabulated results are given in Tables 1, 2, and 3, where the simulation was stopped after 10 time steps of size $\Delta t = 432$ s. The horizontal components of any spurious velocity observed were very similar, and so only the maximum of these is given, along with the vertical component of the spurious velocity. As can be seen the magnitude of these errors increases both with distortion of the mesh (in relation to the fractional seamount height δ) and intensity of the stratification, as measured by the Burger number. It can be seen that at moderate resolutions it is typical to expect between one and two orders of magnitude improvement in the velocity errors when the hydrostatic pressure solver is used, these results are identified by calling them ocean model results.

In Beckmann and Haidvogel (1993) similar pressure gradient error tests are carried out, it is noted that for the semispectral model (SPEM), increased resolution in both the horizontal and vertical decreases this spurious behavior, however increased resolution in the vertical

TABLE 2. As in Table 1 but with a quadratic stratification and a Burger number $B = 1.0$.

δ	Resolution	Error in standard model	Error in ocean model
0.1	Δx, Δy = 12.5, V = 12	1.352(−2), 4.150(−3)	1.894(−4), 3.453(−5)
0.1	Δx, Δy = 12.5, V = 24	1.528(−2), 4.474(−3)	1.665(−4), 1.409(−5)
0.1	Δx, Δy = 6.25, V = 6	2.774(−2), 1.442(−3)	1.118(−3), 2.217(−4)
0.1	Δx, Δy = 6.25, V = 12	1.161(−2), 4.327(−4)	2.958(−4), 5.608(−5)
0.1	Δx, Δy = 3.125, V = 6	3.167(−2), 1.921(−3)	1.201(−3), 2.312(−4)
0.1	Δx, Δy = 3.125, V = 12	1.477(−2), 4.268(−4)	3.254(−4), 5.989(−5)
0.9	Δx, Δy = 12.5, V = 12	2.732(−1), 1.821(−1)	3.260(−2), 6.010(−3)
0.9	Δx, Δy = 12.5, V = 24	2.958(−1), 1.888(−1)	3.576(−2), 6.983(−3)
0.9	Δx, Δy = 6.25, V = 6	2.010(−1), 2.812(−2)	8.228(−2), 1.162(−3)
0.9	Δx, Δy = 6.25, V = 12	9.158(−2), 1.334(−2)	1.239(−2), 1.174(−3)
0.9	Δx, Δy = 3.125, V = 6	2.307(−1), 3.124(−2)	4.356(−3), 5.449(−4)
0.9	Δx, Δy = 3.125, V = 12	1.073(−1), 1.506(−2)	3.237(−3), 4.323(−4)

alone does not. Analogous behavior for a finite-difference method is found in Haney (1991). Similar behavior is witnessed here, where increased resolution in both the horizontal and vertical does indeed decrease the magnitude of the observed spurious velocities. Also, at lower-horizontal resolutions it is found that increasing vertical resolution alone does not necessarily decrease the error. Hence it appears that a phenomenon very similar to the sigma-coordinate error is being witnessed. However, at higher-horizontal resolutions the opposite is found, where further increasing the resolution in the horizontal alone actually increases the errors, whereas extra vertical resolution again decreases the error. Thus an additional component of error seems present.

Now, in the finite-difference sigma-coordinate approach, where horizontal derivatives are computed as the gradient along sigma coordinates plus a correction term, discrepancies in computing these two terms leads to pressure gradient errors. However, the finite-element approach represents all fields in a global polynomial rather than pointwise form and horizontal derivatives can be computed naturally whatever the mesh. So although mesh distortion can have an impact on the computation of pressure gradients and is an important factor to consider (e.g., the apparent sigma coordinate type error at low-horizontal resolutions), it appears that the ability of the method to represent accurately the vertical structure of the density field also plays a crucial role in this error. An example of this being the ability of the (piecewise linear) model to represent exactly a density field linear in z. Further investigation of this procedure for dealing with pressure gradient errors in the model is ongoing.

It has been further observed that errors in the standard model increase with time more rapidly than in the ocean model, leading ultimately to instability that affects all solution fields. The ocean model on the other hand is far better behaved, demonstrating its ability to compute accurate solutions in a wide range of oceanic applications, some of which will be presented in Part II. This leads to the conviction that the ocean model should be capable of simulating flow at a small Rossby number and an order-unity Burger number.

5. A Petrov–Galerkin discretization

In situations where advection dominated flow [as indicated by a grid Reynolds number, $uh/2\mu$, greater than unity (Donea and Huerta 2003)] past a seamount is simulated (see Part II), one further modification to the standard model is required. It is well known that a standard (centered) second-order spatial and temporal discretization of advection may lead to spurious oscillations in the vicinity of sharp gradients in the solution. These may well globally pollute the numerical solution, as well as under/over shoots possibly yielding physically un-

TABLE 3. As in Table 1 but with a quadratic stratification and a Burger number $B = 1.5$.

δ	Resolution	Error in standard model	Error in ocean model
0.1	Δx, Δy = 12.5, V = 12	2.975(−2), 9.036(−3)	4.232(−4), 7.653(−5)
0.1	Δx, Δy = 12.5, V = 24	3.366(−2), 9.751(−3)	3.727(−4), 3.099(−5)
0.1	Δx, Δy = 6.25, V = 6	6.227(−2), 3.223(−3)	2.483(−3), 4.898(−4)
0.1	Δx, Δy = 6.25, V = 12	2.613(−2), 9.544(−4)	6.576(−4), 1.232(−4)
0.1	Δx, Δy = 3.125, V = 6	7.097(−2), 4.283(−3)	2.666(−3), 5.109(−4)
0.1	Δx, Δy = 3.125, V = 12	3.319(−2), 9.565(−4)	7.226(−4), 1.319(−4)
0.9	Δx, Δy = 12.5, V = 12	5.863(−1), 3.528(−1)	7.265(−2), 1.336(−2)
0.9	Δx, Δy = 12.5, V = 24	6.341(−1), 3.662(−1)	7.971(−2), 1.553(−2)
0.9	Δx, Δy = 6.25, V = 6	4.496(−1), 6.263(−2)	1.824(−2), 2.559(−3)
0.9	Δx, Δy = 6.25, V = 12	2.063(−1), 2.980(−2)	2.733(−2), 3.839(−3)
0.9	Δx, Δy = 3.125, V = 6	5.120(−1), 6.911(−2)	9.584(−3), 1.183(−3)
0.9	Δx, Δy = 3.125, V = 12	2.390(−1), 3.334(−2)	7.123(−3), 9.517(−4)

realistic, indeed unstable, solutions. A finite-element approach to dealing with this, known as a Petrov–Galerkin formulation, is to alter the discretization by weighting the test functions in, for example, streamline and cross-streamline directions. This may be regarded as a finite-element analog of upwind differencing in a gridpoint scheme and results in inclusion of some solution dependent artificial dissipation in appropriate directions. Since this is done within the projection to finite dimensions, consistency and accuracy of the discretization can be preserved.

First, it is convenient to extend the definitions of the spatial basis functions $\{N_i\}$ and $\{M_j\}$ to the space–time domain. That is, introduce spatiotemporal basis functions

$$\mathcal{N}_i^n(\mathbf{x}, t) = \begin{cases} N_i(\mathbf{x})(t - t_n) & \text{for } t_{n-1} < t < t_n \\ N_i(\mathbf{x})(t_{n+1} - t) & \text{for } t_n < t < t_{n+1} \\ 0 & \text{for } t < t_{n-1} \text{ or } t > t_{n+1} \end{cases} \tag{53}$$

$$\hat{\mathcal{N}}_i^n(\mathbf{x}, t) = \begin{cases} N_i(\mathbf{x})(t - t_n) & \text{for } t_n < t < t_{n+1} \\ 0 & \text{for } t < t_n \text{ or } t > t_{n+1} \end{cases} \tag{54}$$

and

$$\mathcal{M}_j^n(\mathbf{x}, t) = \begin{cases} M_j(\mathbf{x}) & \text{for } t_{n-1} < t < t_n \\ 0 & \text{for } t < t_{n-1} \text{ or } t > t_n, \end{cases} \tag{55}$$

where t is time and the t_n are the discrete time levels. A discretization of the momentum equations (1a) is now obtained by requiring

$$\int dt \int dV\, \mathcal{W}_i^n(\mathbf{x}, t)[L(\boldsymbol{u}) - \mathbf{F}] = -\int dt \int dV\, \mathcal{V}_i^n(\mathbf{x}, t)\boldsymbol{\nabla} p, \tag{56}$$

where

$$L(\boldsymbol{u}) = \frac{\partial \boldsymbol{u}}{\partial t} + \tilde{\boldsymbol{u}} \cdot \boldsymbol{\nabla}\boldsymbol{u} + 2\boldsymbol{\Omega} \times \boldsymbol{u} - \boldsymbol{\nabla} \cdot \boldsymbol{\tau}. \tag{57}$$

Over a time step $t_n < t < t_{n+1}$, the velocity $\boldsymbol{u}$ is, for example, the velocity at $t = t_n$. Equation (57) thus represents a stable linearization of the momentum equation (1a) over a time step.

In (56) the $\mathcal{W}_i^n$ and $\mathcal{V}_i^n$ are spatiotemporal weight functions, and so (56) represents a Galerkin projection of the momentum equations (1a) in space and time. In a classical Galerkin method, the $\mathcal{W}_i^n$ and $\mathcal{V}_i^n$ would be identical to the spatiotemporal basis functions $\hat{\mathcal{N}}_i^n$. However, here a Petrov–Galerkin method is employed, in which

$$\mathcal{W}_i^n = \hat{\mathcal{N}}_i^n + h_u(\mathbf{x})\left(\tilde{\boldsymbol{u}} \cdot \boldsymbol{\nabla}\hat{\mathcal{N}}_i^n + \frac{\partial \hat{\mathcal{N}}_i^n}{\partial t}\right), \tag{58a}$$

$$\mathcal{V}_i^n = \hat{\mathcal{N}}_i^n \tag{58b}$$

where

$$h_u(\mathbf{x}) = \left(\left\|\frac{J^{-1}\boldsymbol{u}}{\dfrac{2}{\Delta t}}\right\|_\infty\right)^{-1}. \tag{59}$$

Here, J is the Jacobian of the transformation from the finite-element coordinates (ξ_1, ξ_2, ξ_3) to the physical coordinates (x_1, x_2, x_3).

The additional term on the right-hand side of (58a), which makes $\mathcal{W}_i^n$ different from $\mathcal{N}_i^n$, may be regarded as an upwind term. To appreciate this, consider the case of one spatial dimension. In that case,

$$(\|J^{-1}\boldsymbol{u}\|_\infty)^{-1} = \frac{h_e}{2|u|},$$

in which h_e is the element length and u is the 1D velocity.

Note that, although viscous stress terms have been included in $L(\boldsymbol{u})$, when integrated in (56) terms of the form $(\boldsymbol{\nabla} \cdot \boldsymbol{\tau})(\tilde{\boldsymbol{u}} \cdot \boldsymbol{\nabla} N_i)$, which appear when the weight functions $\mathcal{W}_i^k$ are used, are not included. Upon integration by parts, it can be seen that these terms actually vanish in the case of tetrahedral elements, and this motivates their neglect in the present case of hexahedral elements. Similar schemes are described in detail in Hughes and Mallet (1986).

6. Concluding remarks

A finite-element approach to the numerical modeling of three-dimensional nonhydrostatic stratified oceanic flows has been presented. The guiding principle for the work here has been the need to examine the application of alternative and new numerical techniques to challenging geophysical fluid dynamics problems. The motivation being the desire to employ the full power and flexibility of unstructured grids, both in terms of faithful geometry representation and the ability to use variable model resolution and ultimately to use dynamic mesh adaptivity. The model as presented is temporally and spatially second-order accurate. It is based on a standard Galerkin approach, but with the option for Petrov–Galerkin upwind-type corrections. It employs a factorization of the incompressibility constraint resulting in an accurate and efficient method. The result is an efficient and robust model, suitable for simulating flows in a wide range of parameter regimes, and fit for the inclusion of additional numerical and ocean modeling options. The result should be an immensely powerful tool for the modeling community.

In the course of this model development problems related to the importance of the buoyancy force and hydrostatic balance in oceanic regimes have been highlighted and addressed. These issues are similar in nature to sigma-coordinate pressure gradient errors present over steep topography. A technique for solving this problem has been presented that allows meaningful simulations to be made. This involves splitting pressure into hydrostatic and nonhydrostatic parts, and then accurately solving for additional source terms in the momentum equations that ensure consistency of the approximation, as well as allowing the model to represent states of hydrostatic balance.

In Part II of this work continued validation of the model shall be presented. This is based on further simulation of flow past a tall isolated seamount, consideration of convergence of the model, and through comparisons with other numerical approaches.

Further work is under way on the use of the model with tetrahedral elements and unstructured meshes on complex domains. The model is also being used with optimization-based anisotropic mesh adaptivity. Analysis of the hydrostatic pressure gradient errors and the solver developed here are being carried out on unstructured tetrahedral meshes. Some interesting outstanding questions involve the apparent combined nature of this error in terms of a classical sigma-coordinate-type error, and a term emanating from the representation of the density field, particularly in the vertical; also the effect the high-order accuracy of the hydrostatic pressure gradient has on the temporal and spatial second-order accuracy of the model.

Acknowledgments. We acknowledge the contributions of the lead author, Rupert Ford, who died on 30 March 2001 when he and his research were flourishing. His enthusiasm and excitement at each achievement made in this project is greatly missed. This paper was completed in his absence, though not without his vision. We hope that it does justice to one aspect of his work.

We would like to thank both Dr. Marco Luchini and Mr. Nick James for initiatives that gave us access to a distributed parallel machine, which greatly assisted the development of the model. This research was carried out under funding from the National Environmental Research Council of the United Kingdom under Grants NER/A/S/2000/00375 and GR9/02518 for which we are grateful. We are very grateful to three anonymous referees whose comments helped improve the quality of this paper.

REFERENCES

Adams, L. M., (1983): An m-step preconditioned conjugate gradient method for parallel computations. *IEEE Parallel Computation Proc.,* Bellaire, MI, IEEE, 36–43.

Adcroft, A., C. Hill, and J. Marshall, 1997: Representation of topography by shaved cells in a height coordinate ocean model. *Mon. Wea. Rev.,* **125,** 2293–2315.

Arakawa, A., 1966: Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow. *J. Comput. Phys.,* **1,** 119–143.

Babuška, I., 1971: Error-bounds for finite element method. *Numer. Math.,* **16,** 322–333.

Beckmann, A., and D. B. Haidvogel, 1993: Numerical simulation of flow around a tall isolated seamount. Part I: Problem formulation and model accuracy. *J. Phys. Oceanogr.,* **23,** 1736–1753.

Brezzi, F., 1974: On the existence, uniqueness and approximation of saddle-point problems arising from Lagrange multipliers. *RAIRO Ser. Rouge Anal. Numer.,* **8** (R-2), 129–151.

——, and M. Fortin, 1991: *Mixed and Hybrid Finite Element Methods.* Springer-Verlag, 350 pp.

Brown, D. L., R. Cortez, and M. L. Minion, 2001: Accurate projection methods for the incompressible Navier–Stokes equations. *J. Comput. Phys.,* **168,** 464–499.

Bryan, K., 1969: A numerical method for the study of the circulation of the world ocean. *J. Comput. Phys.,* **4,** 347–376.

Chorin, A. J., 1968: Numerical solution of the Navier–Stokes equations. *Math. Comput.,* **22,** 745–762.

Cockburn, B., G. E. Karniadakis, and C. W. Shu, Eds., 2000: *Discontinuous Galerkin Methods: Theory, Computation and Applications.* Springer-Verlag, 470 pp.

Danilov, S., G. Kivman, and J. Schröter, 2004: A finite element ocean model: Principles and evaluation: *Ocean Modell.,* **6,** 125–150.

Donea, J., and A. Huerta, 2003: *Finite Element Methods for Flow Problems.* John Wiley & Sons, 350 pp.

Dumas, E., C. L. Provost, and A. Poncet, 1982: Feasability of finite element methods for oceanic general circulation modelling. *Proc. Fourth Int. Conf. on Finite Elements in Water Resources,* Hanover, Germany, Deutsche Forschungsgmeinschaft/ISCME/IAHR, 5.43–5.55.

E, W., and J. G. Liu, 1996: Projection method II: Godunov–Ryabenki analysis. *SIAM J. Numer. Anal.,* **33,** 1597–1621.

Engelman, M. S., R. L. Sani, and P. M. Gresho, 1982: The implementation of normal and/or tangentital boundary conditions in finite element codes for incompressible fluid flow. *Int. J. Numer. Methods Fluids,* **2,** 225–238.

Ezer, T., H. Arango, and A. F. Shchepetkin, 2002: Developments in terrain-following ocean models: Intercomparion of numerical aspects. *Ocean Modell.,* **4,** 249–267.

Fix, G. J., 1975: Finite element models for ocean circulation problems. *SIAM J. Appl. Math.,* **29,** 371–387.

Ford, R., C. C. Pain, M. D. Piggott., A. J. H. Goddard, C. R. E. de Oliveira, and A. P. Umbleby, 2004: A nonhydrostatic finite-element model for three-dimensional stratified oceanic flows. Part II: Model validation. *Mon. Wea. Rev.,* **132,** 2832–2844.

Girault, V., and P. A. Raviart, 1986: *Finite Element Methods for Navier–Stokes Equations: Theory and Algorithms.* Springer-Verlag, 374 pp.

Golub, G. H., and C. F. van Loan, 1989: *Matrix Computations.* 2d ed. Johns Hopkins University Press, 642 pp.

Greenberg, D. A., F. E. Werner, and D. R. Lynch, 1998: A diagnostic finite-element ocean circulation model in spherical-polar coordinates. *J. Atmos. Oceanic Technol.,* **15,** 942–958.

Gresho, P. M., 1990: On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 1: Theory. *Int. J. Numer. Methods Fluids,* **11,** 587–620.

——, and R. Sani, 1987: On pressure boundary conditions for the incompressible Navier–Stokes equations. *Int. J. Numer. Methods Fluids,* **7,** 1111–1145.

——, and S. T. Chan, 1990: On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 2: Implementation. *Int. J. Numer. Methods Fluids,* **11,** 621–659.

——, and R. L. Sani, 1998: *Incompressible Flow and the Finite Element Method: Advection-Diffusion and Isothermal Laminar Flow.* John Wiley and Sons, 1021 pp.

Griffiths, D. F., 1996: Discretised eigenvalue problems, LBB constants and stabilization. *Pitman Research Notes in Mathematics,* D. F. Griffiths and G. A. Watson, Eds., Vol. 334, Addison Wesley Longman, 105 pp.

——, and D. Silvester, 1994: Unstable modes of the $Q_1 - P_0$ element. Technical Rep., Numerical Analysis Rep. 257, Department of Mathematics, University of Manchester, 17 pp.

Haidvogel, D. B., A. R. Robinson, and E. E. Schulman, 1980: The accuracy, efficiency, and stability of three numerical models with application to open ocean problems. *J. Comput. Phys.,* **34,** 1–53.

——, J. L. Wilkin, and R. Young, 1991: A semi-spectral primitive equation ocean circulation model using vertical sigma coordinates and orthogonal curvilinear horizontal coordinates. *J. Comput. Phys.,* **94,** 151–185.

Hanert, E., V. Legat, and E. Deleersnijder, 2002: A comparison of three finite elements to solve the linear shallow water equations. *Ocean Modell.,* **5,** 17–35.

Haney, R. L., 1991: On the pressure gradient force over steep topography in sigma coordinate models. *J. Phys. Oceanogr.,* **21,** 610–619.

Heywood, J. G., and R. Rannacher, 1982: Finite element approximation of the nonstationary Navier–Stokes problem. I. Regularity of solutions and second-order error estimates for spatial discretization. *SIAM J. Numer. Anal.,* **19,** 275–311.

——, and ——, 1990: Finite element approximation of the nonstationary Navier–Stokes problem. IV. Error analysis for second-order time discetization. *SIAM J. Numer. Anal.,* **27,** 353–384.

Hughes, T. J. R., and M. Mallet, 1986: A new finite element formulation for computational fluid dynamics: IV. A discontinuity-capturing operator for multidimensional advective-diffusion systems. *Comput. Methods Appl. Mech. Eng.,* **58,** 329–336.

Iskandarani, M., D. B. Haidvogel, and J. P. Boyd, 1995: A staggered spectral element model with application to the oceanic shallow water equation. *Int. J. Numer. Method Fluids,* **20,** 393–414.

——, ——, and J. C. Levin, 2003: A three-dimensional spectral element method for the solution of the hydrostatic primitive equations. *J. Comput. Phys.,* **186,** 397–425.

Johnson, C., 1987: *Numerical Solution of Partial Differential Equations by the Finite Element Method.* Cambridge University Press, 278 pp.

Ladyzhenskaya, O., 1969: *Mathematical Theory of Viscous Incompressible Flow.* Gordon and Breach Science Publishers, 256 pp.

Legrand, S., V. Legat, and E. Deleersnijder, 2000: Delaunay mesh generation for an unstructured-grid ocean general circulation model. *Ocean Modell.,* **2,** 17–28.

Le Roux, D. Y., A. Staniforth, and C. A. Lin, 1998: Finite elements for shallow-water equation ocean models. *Mon. Wea. Rev.,* **126,** 1931–1951.

——, C. A. Lin, and A. Staniforth, 2000: A semi-implicit semi-Lagrangian finite-element shallow-water ocean model. *Mon. Wea. Rev.,* **128,** 1384–1401.

Lynch, D. R., and W. R. Gray, 1979: A wave equation model for finite element tidal computations. *Comput. Fluids,* **7,** 207–228.

——, F. E. Werner, 1987: Three-dimensional hydrodynamics on finite elements. Part I: Linearized harmonic model. *Int. J. Numer. Methods Fluids,* **7,** 871–909.

——, and ——, 1991: Three-dimensional hydrodynamics on finite elements. Part II: Non-linear time-stepping model. *Int. J. Numer. Methods Fluids,* **12,** 507–533.

——, J. T. C. Ip, C. E. Naimie, and F. E. Werner, 1996: Comprehensive coastal circulation model with application to the Gulf of Maine. *Cont. Shelf Res.,* **16,** 875–906.

Mellor, G. L., L.-Y. Oey, and T. Ezer, 1998: Sigma coordinate pressure gradient errors and the seamount problem. *J. Atmos. Oceanic Technol.,* **15,** 1122–1131.

Myers, P. G., and A. J. Weaver, 1995: A diagnostic barotropic finite-element ocean circulation model. *J. Atmos. Oceanic Technol.,* **12,** 511–526.

Naimie, C. E., J. W. Loder, and D. R. Lynch, 1994: Seasonal variation of the three-dimensional residual circulation on Georges Bank. *J. Geophys. Res.,* **99** (C8), 15 967–15 989.

Nechaev, D., J. Schröter, and M. Yaremchuk, 2003: A diagnostic stabilized finite-element ocean circulation model. *Ocean Modell.,* **5,** 37–63.

Pedlosky, J., 1987: *Geophysical Fluid Dynamics.* 2d ed. Springer-Verlag, 710 pp.

Provost, C. L., and P. Vincent, 1986: Some tests of precision for a finite element model of ocean tides. *J. Comput. Phys.,* **110,** 273–291.

Saad, Y., and M. H. Schultz, 1986: GMRES, a generalised minimum residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.,* **7,** 856–869.

Shen, J., 1996: On error estimates of the projection methods for the Navier–Stokes equations: Second-order schemes. *Math. Comput.,* **65,** 1039–1065.

Smith, A., and D. Silvester, 1997: Implicit algorithms and their linearization for the transient incompressible Navier–Stokes equations. *SIAM J. Numer. Anal.,* **17,** 527–545.

Song, Y. T., 1998: A general pressure gradient formulation for ocean models. Part I: Scheme design and diagnostic analysis. *Mon. Wea. Rev.,* **126,** 3213–3230.

Tennekes, H., and J. L. Lumley, 1972: *A First Course in Turbulence.* MIT Press, 300 pp.

Turner, J. S., 1973: *Buoyancy Effects in Fluids.* Cambridge University Press, 368 pp.