Available online at www.sciencedirect.com
SCIENCE DIRECT®
Ocean Modelling 14 (2006) 139–173
Ocean Modelling
www.elsevier.com/locate/ocemod

An unstructured-grid, finite-volume, nonhydrostatic, parallel coastal ocean simulator

O.B. Fringer *, M. Gerritsen, R.L. Street

Environmental Fluid Mechanics Laboratory, Department of Civil and Environmental Engineering, Stanford University, Stanford, CA 94305-4020, United States

Received 30 August 2005; received in revised form 23 March 2006; accepted 23 March 2006
Available online 3 May 2006

Abstract

A finite-volume formulation is presented that solves the three-dimensional, nonhydrostatic Navier–Stokes equations with the Boussinesq approximation on an unstructured, staggered, z-level grid, with the goal of simulating nonhydrostatic processes in the coastal ocean with grid resolutions of tens of meters. In particular, the code has been developed to simulate the nonlinear, nonhydrostatic internal wave field in the littoral ocean. The method is based on the formulation developed by Casulli, in that the free-surface and vertical diffusion are semi-implicit, thereby removing stability limitations associated with the surface gravity wave and vertical diffusion terms. The remaining terms in the momentum equations are discretized explicitly with the second-order Adams–Bashforth method, while the pressure-correction method is employed for the nonhydrostatic pressure in order to achieve overall second-order temporal accuracy. Advection of momentum is accomplished with an Eulerian discretization which conserves momentum in cells that do not contain the free surface, and scalar advection is discretized in a way that ensures consistency with continuity, thereby ensuring local and global mass conservation using a velocity field that conserves volume on a local and global basis. The nonhydrostatic pressure field is solved efficiently using a block-Jacobi preconditioner, and while stability is limited by the internal gravity wave speed and vertical advection of momentum, applications requiring relatively small time steps due to accuracy or stability constraints are run efficiently on parallel computers, since the present formulation is written entirely with the message-passing interface (MPI). The ParMETIS libraries are employed in order to achieve a load-balanced parallel partitioning that minimizes interprocessor communication, and the grid is reordered to optimize per-processor performance by limiting cache misses while accessing arrays in memory. Test cases demonstrate the ability of the code to efficiently and accurately compute the nonhydrostatic lock exchange and internal waves in idealized as well as real domains, and we evaluate the parallel efficiency of the code using up to 32 processors.

Keywords: Nonhydrostatic; Unstructured grid; Parallel; Internal gravity wave

* Corresponding author. Tel.: +1 650 725 6878.
E-mail addresses: fringer@stanford.edu (O.B. Fringer), margot.gerritsen@stanford.edu (M. Gerritsen), street@stanford.edu (R.L. Street).

1463-5003/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ocemod.2006.03.006

1. Introduction

Recent measurements in the coastal ocean demonstrate the existence of highly nonlinear internal solitary waves that propagate along coastal shelves (see, e.g. Klymak and Moum, 2003; Carter et al., 2005). While their generation mechanism is not clear, it is hypothesized that they form when long, internal waves of tidal frequency (internal tides) interact with topography, leading to a nonlinear cascade that results in a transfer of energy from wavelengths of tens of kilometers and timescales of tens of hours down to highly nonlinear and nonhydrostatic solitary waves with wavelengths of tens of meters and timescales of seconds. To date, no simulation code has captured this cascade, simply because computational resources have made it intractable. However, recent advances in numerical methods and scientific computing have lead to simulation methods which now make a solution of the multiscale problem of internal waves, and a host of problems in coastal oceanography in general, feasible. This paper presents a numerical formulation, built on the successes of many others, that combines proven numerical and computational tools in an optimal fashion to yield an efficient simulation code designed to capture the entire multiscale internal wave energy cascade. To this end, the paper outlines a finite-volume, unstructured-grid, nonhydrostatic, parallel coastal-ocean solver, which was built from the beginning to be a highly efficient code on parallel processors, under the name of the Stanford Unstructured Nonhydrostatic Terrain-following Adaptive Navier–Stokes Simulator (SUNTANS). As the name implies, the ultimate goal is to have an unstructured-grid tool that adaptively simulates multiscale physics in the coastal ocean. Here we discuss the fundamental hydrodynamic kernel of SUNTANS, including the nonhydrostatic and Navier–Stokes methodology on an unstructured grid.

The nonhydrostatic formulation in SUNTANS is based on that of Casulli (1999), which has spawned numerous code efforts (Mahadevan et al., 1996a,b; Hodges et al., 2000; Wadzuk et al., 2004), all of which split the pressure into its hydrostatic and nonhydrostatic components. The hydrostatic, or predictor, velocity field that satisfies the depth-averaged continuity equation is computed first. It is then corrected with the nonhydrostatic pressure field to enforce local continuity. This method can be used to convert existing primitive equation models to their nonhydrostatic counterparts, since it can be done with the addition of a nonhydrostatic pressure solver and a vertical momentum equation, which is replaced by the continuity equation in the primitive equations. Kanarska and Maderich (2003) demonstrated this by extending the free-surface primitive equation code POM (Blumberg and Mellor, 1987), while Wadzuk et al. (2004) extended the CWR-ELCOM code (Hodges et al., 2000). Marshall et al. (1997a,b) describe a similar nonhydrostatic approach to that of Casulli, in that they split the hydrostatic and nonhydrostatic components of the pressure, and first solve a two-dimensional Poisson equation for the pressure at the free-surface, followed by a solution of the three-dimensional Poisson equation for the nonhydrostatic pressure field. They also describe a quasi-hydrostatic formulation that adds the cosine of latitude Coriolis terms to the primitive equations as an intermediate approximation between the fully hydrostatic and fully nonhydrostatic equation sets. Numerous other works, including the finite-element formulation of Ford et al. (2004), implement similar nonhydrostatic formulations, all of which rely on the essential splitting of the hydrostatic and nonhydrostatic pressure fields.

These pressure-split algorithms are often referred to as "quasi-hydrostatic" (not to be confused with the quasi-hydrostatic definition of Marshall et al., 1997a) algorithms because the position of the free surface at a given time step is updated in the absence of the nonhydrostatic pressure at that time level. Fully nonhydrostatic algorithms do not split the pressure, but instead compute the full pressure field and advance the free surface explicitly using the kinematic free-surface boundary condition, such as the large-eddy simulation of Hodges and Street (1999) and the finite-element coastal model of Labeur and Pietrzak (2005). The method of SUNTANS incorporates a pressure-split algorithm and this is inherently quasi-hydrostatic from the point of view that the nonhydrostatic pressure does not directly affect the free-surface at a given time step. However, we demonstrate that the present pressure-split algorithm is fully nonhydrostatic from the point of view of the temporal accuracy of the solution method. That is, the effect of the nonhydrostatic pressure on the free-surface at a given time step is negligible since it is the same order of magnitude as the error associated with the global second-order temporal accuracy of the overall solution procedure.

While the pressure-split formulation intelligently reduces the workload of the nonhydrostatic pressure solver, the three-dimensional elliptic equation that governs the nonhydrostatic pressure still accounts for

a bulk of the workload for most nonhydrostatic codes. The nonhydrostatic pressure solver in the laboratory-scale LES code of Zang et al. (1994) employs a multigrid solver, for which the performance is improved with the parallel implementation of Cui and Street (2000) using the message passing interface (MPI). For field-scale applications, the three-dimensional pressure-Poisson equation is poorly conditioned because the terms containing the vertical second derivatives are typically two orders of magnitude larger than those containing the horizontal second derivatives. This poses a stringent limitation on the efficiency of the numerical solver unless preconditioning is employed. For this reason, Casulli (1999) employs the preconditioned conjugate gradient algorithm both for the nonhydrostatic pressure as well as the free-surface solver. Marshall et al. (1997b) describe this preconditioning approach in their discussion of the block-Jacobi preconditioner for the preconditioned conjugate gradient algorithm. This preconditioner is ideally suited to their parallel implementation in MPI because it does not require any additional communication among processors since its solution is computed via a tridiagonal inverse in each water column, the data for which is contiguous on each processor. We adopt a similar solution strategy for the parallel unstructured-grid Poisson solver in SUNTANS, and demonstrate how the effectiveness of the preconditioner is strongly dependent on the grid aspect ratio.

In addition to developing computationally efficient solutions of the nonhydrostatic pressure, stability with respect to fast free-surface gravity waves in ocean models is paramount because propagation speeds can be 100 times faster than typical current or internal gravity wave speeds. POM (Blumberg and Mellor, 1987) and relatives, such as SCRUM (Song and Haidvogel, 1994), the unstructured finite-volume code FVCOM (Chen et al., 2003), and the higher-order split-explicit formulation of Shchepetkin and McWilliams (2005), stabilize the fast free-surface gravity waves by employing mode splitting, which advances the barotropic, or external, mode using smaller time steps than the baroclinic, or internal mode. Marshall et al. (1997b) ensure stability for the linear and nonlinear free-surface time advances in the MIT General Circulation Model (MITgcm) by employing a backward Euler scheme for the depth-averaged continuity equation and the barotropic term in the momentum equations, while the Casulli (1999) formulation employs the semi-implicit theta-method for these terms. In SUNTANS, the free-surface terms are also treated using the theta-method. Stability limitations associated with fast free-surface gravity waves can also be removed with the rigid-lid approximation. This is an option in MITgcm as well as the nonhydrostatic finite-element model of Ford et al. (2004), which was extended to include a free surface in Piggott et al. (2005).

The horizontal grid in SUNTANS is a Delaunay triangulation (see, e.g. Shewchuck, 1996), which enjoys the obvious advantages of unstructured grids as pointed out by Pain et al. (2005). In the vertical, like the Casulli and Zanolli (2002) method and the method of Marshall et al. (1997a), SUNTANS employs a z-level grid. Although sigma-coordinate grids can also be used in combination with horizontally unstructured grids, as in FVCOM (Chen et al., 2003), the use of z-levels avoids the well-known pressure-gradient error associated with sigma-coordinate (see, e.g. Mellor et al., 1994, 2004) and fully unstructured grids (Ford et al., 2004). Unlike fully unstructured grids, z-levels and sigma-coordinates are more efficient from a memory access point of view since water columns can be stored contiguously in memory. The primary disadvantage to using z-level grids is the stair-stepped resolution of bottom topography. While this error can be minimized by aligning unstructured grid faces with isobaths, the treatment of the bottom boundary can be improved either by rewriting the finite-volume equations at the bottom-most, or "shaved" cells, using the method of Adcroft et al. (1997), or by employing the immersed boundary method (Tseng and Ferziger, 2003). In the latter, which is much more feasible when unstructured prisms are involved, irregular boundaries are immersed into a computational domain which contains z-levels. The cells adjacent to and inside the immersed boundaries are computational cells, while the values in the cells adjacent to and outside the boundary are updated by extrapolations that use the values in the computational cells while satisfying boundary conditions at the immersed boundary.

In SUNTANS, the unstructured, finite-volume grid is staggered such that only the velocity components normal to the faces are stored, while all other variables are stored at the Voronoi points, which are the analogue of cell centers on Cartesian grids (see Section 3). This is similar to the method of Casulli and Zanolli (2002), while differing from FVCOM (Chen et al., 2003), since in that formulation the horizontal components of the velocity field are stored at the centroids and all other quantities are stored at the vertices. To advect momentum, Casulli and Zanolli employ the semi-Lagrangian, or Euler–Lagrangian method (ELM), which

is a natural discretization on unstructured, staggered grids because advection is computed with an interpolation from the velocity field at points in the vicinity of the Lagrangian traceback. As described in detail in the review of semi-Lagrangian advection schemes by Staniforth and Côté (1991), the accuracy of such schemes is then dictated by the accuracy of the spatial interpolation scheme, and, to a lesser degree, by the accuracy with which the position of the traceback is computed. Le Roux et al. (1997), for example, developed an accurate semi-Lagrangian advection scheme on unstructured grids using the Kriging interpolation scheme. The greatest advantage of the semi-Lagrangian methods is that they are stable. The drawback, however, is that most implementations do not conserve momentum. Eulerian schemes for momentum advection have the advantage that they can be conservative. For unstructured, collocated grids, Chen et al. (2003) devised a conservative Eulerian scheme for advection of momentum in FVCOM, while the same was done for unstructured, staggered grids by Perot (2000). The scheme of Perot (2000) is employed in SUNTANS and, while it has the attractive property of being conservative, it introduces a time step limitation on stability because it is computed explicitly with the second-order Adams–Bashforth scheme.

Finite-volume, scalar advection does not share the same conservation difficulty as momentum advection on unstructured, staggered grids because the scalar quantities are stored at the Voronoi points. This makes it possible to conserve mass on a local and global basis as long as the advection discretization is consistent with the three-dimensional and depth-averaged continuity equations, as pointed out by Gross et al. (2002), for the Casulli (1999) method, and Campin et al. (2004), for the method of Marshall et al. (1997b). Once consistency with continuity is ensured, however, monotonicity of the advected quantity is not necessarily guaranteed when using higher-order schemes to interpolate the scalar values onto the faces. Gross et al. (1999) compare a number of advection schemes on a Cartesian grid, including the total variation diminishing (TVD) schemes (see, e.g. Roe, 1984), while Casulli and Zanolli (2005) extend the concept of the flux limiter used in the TVD schemes to unstructured grids in order to improve the interpolation accuracy while guaranteeing the min–max principle of the advected quantity. The TVD concept is also employed on unstructured finite-volume grids by Darwish and Moukalled (2003). The scalar advection formulation in SUNTANS is consistent with the three-dimensional and depth-averaged continuity equations, and the interpolation method to obtain the scalar face values is also monotonic. While the consistent formulation is described in the present paper, details of the higher-order monotonic interpolation are left to a future paper.

In the next sections we describe SUNTANS and demonstrate its performance, paying particular attention to the formulation of the nonhydrostatic and Eulerian momentum advection terms. Section 2 gives the equations of motion, Section 3 describes the unstructured, finite-volume prisms, and Section 4 outlines the numerical formulation. Section 5 describes some test cases that demonstrate the numerical and physical aspects of the nonhydrostatic pressure formulation. First, the internal seiche test case highlights the differences between hydrostatic and nonhydrostatic internal wave propagation and presents fundamental features of nonhydrostatic physics that a nonhydrostatic code must capture. In addition to demonstrating accurate physics, this test case is used to demonstrate the second-order temporal accuracy of SUNTANS in Section 6. The internal seiche represents coupling between the nonhydrostatic pressure and the barotropic and baroclinic pressure fields in the linear regime, and hence is verifiable analytically. As a second test, the lock-exchange is employed to assess the coupling between the nonhydrostatic pressure and nonlinear advection of momentum. In the third test, we demonstrate the ability of the code to capture field-scale generation of internal waves in Monterey Bay, and use this as a test bed to demonstrate the parallel efficiency of SUNTANS in Section 6. Overall, these tests serve to demonstrate that SUNTANS is an effective and operational parallel simulation tool, and sets the stage for ongoing and future, production-scale simulations, an example of which is to simulate internal waves in Monterey Bay using horizontal grid resolutions of 60 m in 200 km domains. We draw our conclusions and describe future directions of the present formulation in Section 7.

2. Equations of motion

The three-dimensional Navier–Stokes equations with the Boussinesq approximation in a rotating frame, after filtering with either Reynolds-averaging or via a large-eddy simulation and employing an eddy viscosity model, are given by

$$\frac{\partial u}{\partial t}+\nabla\cdot(\mathbf{u}u)-fv+bw=-\frac{1}{\rho_0}\frac{\partial p}{\partial x}+\nabla_{\mathrm{H}}\cdot(\nu_{\mathrm{H}}\nabla_{\mathrm{H}}u)+\frac{\partial}{\partial z}\left(\nu_{\mathrm{V}}\frac{\partial u}{\partial z}\right), \tag{1}$$

$$\frac{\partial v}{\partial t}+\nabla\cdot(\mathbf{u}v)+fu=-\frac{1}{\rho_0}\frac{\partial p}{\partial y}+\nabla_{\mathrm{H}}\cdot(\nu_{\mathrm{H}}\nabla_{\mathrm{H}}v)+\frac{\partial}{\partial z}\left(\nu_{\mathrm{V}}\frac{\partial v}{\partial z}\right), \tag{2}$$

$$\frac{\partial w}{\partial t}+\nabla\cdot(\mathbf{u}w)-bu=-\frac{1}{\rho_0}\frac{\partial p}{\partial z}+\nabla_{\mathrm{H}}\cdot(\nu_{\mathrm{H}}\nabla_{\mathrm{H}}w)+\frac{\partial}{\partial z}\left(\nu_{\mathrm{V}}\frac{\partial w}{\partial z}\right)-\frac{g}{\rho_0}(\rho_0+\rho), \tag{3}$$

subject to the incompressibility constraint

$$\nabla\cdot\mathbf{u}=0, \tag{4}$$

where ρ_0 is the constant reference density and the total density is given by $\rho_0+\rho$, and the sine and cosine of latitude Coriolis terms are given by $f=2\omega\sin\phi$ and $b=2\omega\cos\phi$, respectively, where ϕ is the latitude, and ω is the Earth's angular velocity. The horizontal and vertical eddy viscosities are given by ν_{H} and ν_{V}, respectively, and the horizontal gradient operator is given by

$$\nabla_{\mathrm{H}}=\mathbf{e}_x\frac{\partial}{\partial x}+\mathbf{e}_y\frac{\partial}{\partial y}. \tag{5}$$

Following the work of Casulli (1999), the pressure is split into its hydrostatic p_{h} and nonhydrostatic q components with $p=p_{\mathrm{h}}+q$, where the hydrostatic pressure is defined by

$$\frac{\partial p_{\mathrm{h}}}{\partial z}=-(\rho_0+\rho)g. \tag{6}$$

Integrating this equation from z to the free surface, h, and substitution into the momentum equations (1)–(3) yields

$$\frac{\partial u}{\partial t}+\nabla\cdot(\mathbf{u}u)-fv+bw=-\frac{1}{\rho_0}\frac{\partial q}{\partial x}-\frac{1}{\rho_0}\frac{\partial p_s}{\partial x}-g\frac{\partial}{\partial x}(h+r)+\nabla_{\mathrm{H}}\cdot(\nu_{\mathrm{H}}\nabla_{\mathrm{H}}u)+\frac{\partial}{\partial z}\left(\nu_{\mathrm{V}}\frac{\partial u}{\partial z}\right), \tag{7}$$

$$\frac{\partial v}{\partial t}+\nabla\cdot(\mathbf{u}v)+fu=-\frac{1}{\rho_0}\frac{\partial q}{\partial y}-\frac{1}{\rho_0}\frac{\partial p_s}{\partial y}-g\frac{\partial}{\partial y}(h+r)+\nabla_{\mathrm{H}}\cdot(\nu_{\mathrm{H}}\nabla_{\mathrm{H}}v)+\frac{\partial}{\partial z}\left(\nu_{\mathrm{V}}\frac{\partial v}{\partial z}\right), \tag{8}$$

$$\frac{\partial w}{\partial t}+\nabla\cdot(\mathbf{u}w)-bu=-\frac{1}{\rho_0}\frac{\partial q}{\partial z}+\nabla_{\mathrm{H}}\cdot(\nu_{\mathrm{H}}\nabla_{\mathrm{H}}w)+\frac{\partial}{\partial z}\left(\nu_{\mathrm{V}}\frac{\partial w}{\partial z}\right), \tag{9}$$

where p_s is the pressure at the free surface, and the baroclinic pressure head is given by

$$r=\frac{1}{\rho_0}\int_z^h\rho\,\mathrm{d}z. \tag{10}$$

Integrating the continuity equation (4) from the bottom, defined by $z=-d(x,y)$, to the free-surface at $z=h(x,y,t)$ yields the depth-averaged continuity equation

$$\frac{\partial h}{\partial t}+\frac{\partial}{\partial x}\left(\int_{-d}^h u\,\mathrm{d}z\right)+\frac{\partial}{\partial y}\left(\int_{-d}^h v\,\mathrm{d}z\right)=0, \tag{11}$$

where we have employed the kinematic boundary conditions at $z=h$ and $z=-d$, viz.,

$$\frac{\partial h}{\partial t}+\mathbf{u}_{\mathrm{H}}|_{z=h}\cdot\nabla_{\mathrm{H}}h=w|_{z=h}, \tag{12}$$

$$-\mathbf{u}_{\mathrm{H}}|_{z=-d}\cdot\nabla_{\mathrm{H}}d=w|_{z=-d}, \tag{13}$$

and the horizontal velocity vector is given by

$$\mathbf{u}_{\mathrm{H}}=u\mathbf{e}_x+v\mathbf{e}_y. \tag{14}$$

Eqs. (7)–(11) along with the continuity equation (4) comprise a set of equations for the velocity components u, v, and w, the free surface h, and nonhydrostatic pressure q. The density field is determined by an equation of state in terms of the salinity s and temperature T such that

$$\rho = \rho(s, T), \tag{15}$$

where s and T represent salinity and temperature anomalies from reference states s_0 and T_0, respectively. After filtering and employing a scalar diffusivity law, the transport equations for salinity and temperature are given by

$$\frac{\partial s}{\partial t} + \nabla \cdot (\mathbf{u}s) = \nabla_{\mathrm{H}} \cdot (\gamma_{\mathrm{H}} \nabla_{\mathrm{H}} s) + \frac{\partial}{\partial z}\left(\gamma_{\mathrm{V}} \frac{\partial s}{\partial z}\right), \tag{16}$$

$$\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{u}T) = \nabla_{\mathrm{H}} \cdot (\kappa_{\mathrm{H}} \nabla_{\mathrm{H}} T) + \frac{\partial}{\partial z}\left(\kappa_{\mathrm{V}} \frac{\partial T}{\partial z}\right), \tag{17}$$

where γ_{H}, γ_{V}, κ_{H} and κ_{V} are the horizontal and vertical turbulent mass and thermal diffusivities, respectively. In this paper we neglect the effects of temperature stratification and assume a linear equation of state of the form $\rho = \beta s$, which implies a transport equation for density of the form

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\mathbf{u}\rho) = \nabla_{\mathrm{H}} \cdot (\gamma_{\mathrm{H}} \nabla_{\mathrm{H}} \rho) + \frac{\partial}{\partial z}\left(\gamma_{\mathrm{V}} \frac{\partial \rho}{\partial z}\right). \tag{18}$$

3. Unstructured, finite-volume prism grid cells

We confine ourselves to three-dimensional z-level grids, for which vertical grid spacings remain constant in the horizontal. In plan, the grid is composed of a two-dimensional Delaunay triangulation (see, e.g., Shewchuck, 1996), in which no pointset (which is comprised of the three vertices) of a particular triangle lies within the circumcircle of any other triangle within the triangulation, as shown in Fig. 1. The dual of the Delaunay triangulation is the Voronoi diagram, which connects the circumcenters of the Delaunay triangles. The Voronoi points make up the nodes of the Voronoi diagram, and the edges that connect the Voronoi points are perpendicular to the faces of the Delaunay triangles, thus forming an orthogonal, unstructured grid. All discretizations in the present paper assume this orthogonality condition.

The eddy viscosity, scalar diffusivities, scalars, and nonhydrostatic pressure are defined at the Voronoi points and vertical centers of the prismatic cells, the free-surface and surface pressure are defined at the Voronoi points on the surface of the top cells, and the depth is defined at the Voronoi points at the bottom of the bottom-most cells. The horizontal velocity U is defined normal to each vertical cell face at the intersection of the Voronoi and Delaunay edges, and the vertical velocity w is defined at the Voronoi points at the top and bottom of each cell, as shown in Fig. 2.

Each vertical face with index j has a predefined normal (whose orientation is arbitrary), $\mathbf{n}_j$, which indicates the positive direction of the velocity vector defined on that face, so that, if $\mathbf{u}_j$ is the velocity vector at face j, then

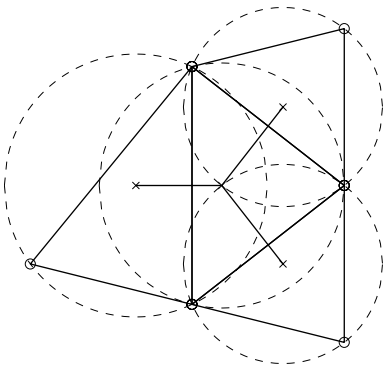

Fig. 1. A Delaunay triangulation, in which the circumcircles (denoted by – –) of the triangles do not contain the pointset of any other triangle in the triangulation. The circumcircles can, however, contain the Voronoi points of neighboring triangles. Delaunay points/edges: – ○ –, Voronoi points/edges: – × –.

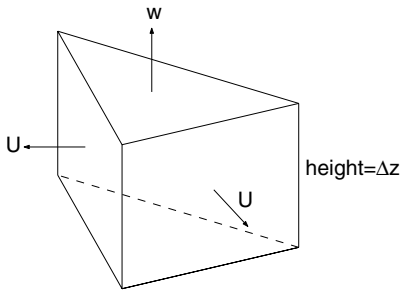

Fig. 2. Depiction of a three-dimensional prismatic grid cell, showing the horizontal velocity U defined normal to the vertical cell faces and the vertical velocity defined at the Voronoi points at the top and bottom of the cell with height Δz.

$$\mathbf{u}_j \cdot \mathbf{n}_j = U_j. \tag{19}$$

Every Delaunay edge j with a normal $\mathbf{n}_j$ has two neighboring cells defined by the Voronoi points that make up the jth Voronoi edge. The indices to these two cells are given by the pointers G_{2j} and G_{2j+1}. The first index, G_{2j}, provides the index of the cell in the direction of $\mathbf{n}_j$, while G_{2j+1} provides the index of the cell in the opposite direction. If x_i and y_i correspond to the Voronoi points of a given planform cell i, then we can define the components of the normal vector $\mathbf{n}_j = n_{1j}\mathbf{e}_x + n_{2j}\mathbf{e}_y$ with

$$n_{1j} = \frac{x_{G_{2j}} - x_{G_{2j+1}}}{D_j}, \tag{20}$$

$$n_{2j} = \frac{y_{G_{2j}} - y_{G_{2j+1}}}{D_j}, \tag{21}$$

where the gradient distance is defined by

$$D_j^2 = (x_{G_{2j}} - x_{G_{2j+1}})^2 + (y_{G_{2j}} - y_{G_{2j+1}})^2. \tag{22}$$

Because G_{2j+1} and G_{2j} are indices to cells, if there are N_c triangular cells that make up the unstructured grid, then for computational edges, $1 \leqslant G_{2j} \leqslant N_c$ and $1 \leqslant G_{2j+1} \leqslant N_c$. By adopting the convention that face-normals always point into the domain, boundary edges are identified by $G_{2j+1} = -1$.

Using the present notation, we can identify the upwind cell i_{iw} corresponding to a given edge j with

$$i_{iw} = \begin{cases} G_{2j+1} & U_j > 0, \\ G_{2j} & \text{otherwise.} \end{cases} \tag{23}$$

We can also define gradients normal to an edge face using the G pointer. As an example, the magnitude of the free-surface gradient $\nabla_H h$ in the direction of the normal $\mathbf{n}_j$ and perpendicular to Delaunay edge j is given by

$$(\nabla_H h)_j \cdot \mathbf{n}_j = \frac{h_{G_{2j}} - h_{G_{2j+1}}}{D_j} + E_g, \tag{24}$$

where E_g is the truncation error. If the grid is composed of equilateral triangles, then the truncation error E_g in Eq. (24) is second order in D_j. Otherwise, face-normal gradients are not centered about the Delaunay edges.

The three outward-pointing normals of each cell can be defined by $\mathbf{n}_1^o$, $\mathbf{n}_2^o$, and $\mathbf{n}_3^o$, as shown in Fig. 3. Rather than storing the components of each of these normals, we store the dot product of the outward normal with the unique normal at every edge $\mathbf{n}_j$, and define this as

$$N_j = \mathbf{n}_j^o \cdot \mathbf{n}_j = \pm 1. \tag{25}$$

This is also used to specify the gradient in the direction of the outward pointing normal at a cell face. Since the gradient of a cell-centered quantity ϕ in the direction of the unique normal $\mathbf{n}_1$ in Fig. 3 is defined as

$$(\mathbf{n}_1 \cdot \nabla\phi)\mathbf{n}_1 = \frac{\phi_{G_{2j}} - \phi_{G_{2j+1}}}{D_j}\mathbf{n}_1, \tag{26}$$

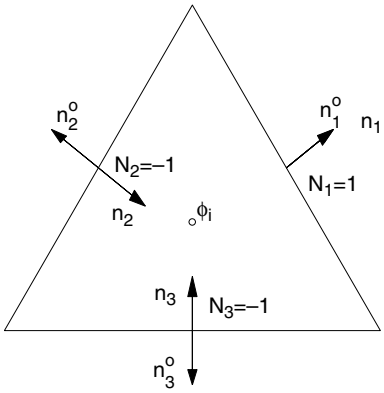

Fig. 3. Depiction of the relationship between the unique normals $\mathbf{n}_j$ defined on every edge, the outward normals $\mathbf{n}_j^{\mathrm{o}}$ defined on the faces of a given cell, and N_j, the dot product of the unique normal with the outward normal.

where index j corresponds to the edge with normal $\mathbf{n}_1$, then the component of the gradient in the direction of the outward normal of edge j with respect to the cell in Fig. 3 is given by

$$\frac{\partial \phi}{\partial n^{\mathrm{o}}} = (\mathbf{n}_1 \cdot \nabla \phi)\mathbf{n}_1 \cdot \mathbf{n}_1^{\mathrm{o}} = \left(\frac{\phi_{G_{2j}} - \phi_{G_{2j+1}}}{D_j}\right) N_1. \tag{27}$$

Denoting the cell for which this outward gradient is defined as cell i and the cell which neighbors this cell on the opposite side of edge j as cell Ne_1, then we can simplify the expression for the outward normal as

$$\frac{\partial \phi}{\partial n^{\mathrm{o}}} = \frac{\phi_{Ne_1} - \phi_i}{D_j}. \tag{28}$$

In this manner we can store values of cell neighbors in the Ne vectors and use these to simplify the calculations of the outward normals.

4. Numerical discretization

4.1. Discretized momentum equations

The horizontal momentum equations are solved at the vertical faces of each cell. To obtain the equation for the normal component of momentum at each face, the dot product of the face-normal vector $\mathbf{n}$, which is horizontal, is taken with the horizontal momentum equations (7) and (8) to obtain

$$\frac{\partial U}{\partial t} + \mathbf{n} \cdot \nabla \cdot (\mathbf{u}\mathbf{u}) = -\frac{1}{\rho_0}\frac{\partial q}{\partial n} - \frac{1}{\rho_0}\frac{\partial p_s}{\partial n} + (fv - bw)n_1 - fun_2 - g\frac{\partial}{\partial n}(h + r) + \nabla_{\mathrm{H}} \cdot (\nu_{\mathrm{H}} \nabla_{\mathrm{H}} U) + \frac{\partial}{\partial z}\left(\nu_{\mathrm{V}} \frac{\partial U}{\partial z}\right), \tag{29}$$

where $\partial/\partial n$ is the face-normal gradient and n_1 and n_2 are the components of the normal vector. The vertical momentum equation is evaluated at the top and bottom of each cell, and is given by

$$\frac{\partial w}{\partial t} + \nabla \cdot (\mathbf{u} w) = -\frac{1}{\rho_0}\frac{\partial q}{\partial z} + bu + \nabla_{\mathrm{H}} \cdot (\nu_{\mathrm{H}} \nabla_{\mathrm{H}} w) + \frac{\partial}{\partial z}\left(\nu_{\mathrm{V}} \frac{\partial w}{\partial z}\right). \tag{30}$$

The first step in the discretization procedure is to use the pressure at the old time step to compute a predicted velocity field at edge j and level k with

$$\frac{U^*_{j,k}-U^n_{j,k}}{\Delta t}=\frac{1}{2}(3F^n_{j,k}-F^{n-1}_{j,k})-\frac{1}{\rho_0}\frac{\partial q^{n-1/2}}{\partial n}-g\theta\left.\frac{\partial h}{\partial n}\right|^{n+1}_j-g(1-\theta)\left.\frac{\partial h}{\partial n}\right|^{n}_j$$
$$+\theta\frac{\partial}{\partial z}\left(\nu^n_{\mathrm{V}}\frac{\partial U^*_{j,k}}{\partial z}\right)+(1-\theta)\frac{\partial}{\partial z}\left(\nu^n_{\mathrm{V}}\frac{\partial U^n_{j,k}}{\partial z}\right), \tag{31}$$

$$\frac{w^*_{i,k}-w^n_{i,k}}{\Delta t}=\frac{1}{2}(3H^n_{i,k}-H^{n-1}_{i,k})-\frac{1}{\rho_0}\frac{\partial q^{n-1/2}}{\partial z}+\theta\frac{\partial}{\partial z}\left(\nu^n_{\mathrm{V}}\frac{\partial w^*_{i,k}}{\partial z}\right)+(1-\theta)\frac{\partial}{\partial z}\left(\nu^n_{\mathrm{V}}\frac{\partial w^n_{i,k}}{\partial z}\right), \tag{32}$$

where the vertical diffusion terms and the free-surface term have been discretized semi-implicitly with the theta-method (Casulli, 1999) and the explicit term has been discretized with the second order Adams–Bashforth method, as in Zang et al. (1994). The old nonhydrostatic pressure is discretized at time step $n-1/2$ to be consistent with the pressure-correction method (Armfield and Street, 2000), in which second-order temporal accuracy is attained for q when the old pressure field is assumed to exist at step $n-1/2$. This is discussed in more detail in Section 6.1. The explicit terms are given by

$$F_{j,k}=-C_{\mathrm{H}}(U_{j,k})-C_{\mathrm{V}}(U_{j,k})-\frac{1}{\rho_0}\frac{\partial p_s}{\partial n}-g\frac{\partial r}{\partial n}+(fv-bw)n_1-(fu)n_2+D_{\mathrm{H}}(U_{j,k}), \tag{33}$$

$$H_{i,k}=-C_{\mathrm{H}}(w_{i,k})-C_{\mathrm{V}}(w_{i,k})+bu+D_{\mathrm{H}}(w_{i,k}), \tag{34}$$

where $C_{\mathrm{H}}(\,)$, $C_{\mathrm{V}}(\,)$, and $D_{\mathrm{H}}(\,)$ are the horizontal and vertical advection and horizontal diffusion operators, respectively, which are defined in Sections 4.3 and 4.4. The horizontal components of the Cartesian velocity vector that are required for the Coriolis terms are defined at the Voronoi points, as described in Section 4.3, and these are interpolated onto the appropriate cell faces.

Boundary conditions on the horizontal velocity at the bottom are given by the drag law

$$\nu_{\mathrm{V}}\left.\frac{\partial U}{\partial z}\right|^{n+1}_{z=-d}=C_{d,B}|U^n|U^{n+1}, \tag{35}$$

while at the free surface, depending on the value of s_{top}, we impose either a drag law ($s_{\mathrm{top}}=0$), or a wind stress ($s_{\mathrm{top}}=1$), such that

$$\nu_{\mathrm{V}}\left.\frac{\partial U}{\partial z}\right|^{n+1}_{z=h}=-(1-s_{\mathrm{top}})C_{d,T}|U^n|U^{n+1}+s_{\mathrm{top}}\mathbf{n}\cdot\boldsymbol{\tau}_{\mathrm{s}}, \tag{36}$$

where $\boldsymbol{\tau}_{\mathrm{s}}$ is the imposed wind shear stress. Boundary conditions on the vertical velocity are no-flux at the bottom and gradient-free at the free surface. Following Casulli and Zanolli (2002), the discrete momentum equations (31) and (32) are solved by inverting tridiagonals for the vertical diffusion operators at each water column after solving for the free surface, for which the solution procedure is described in Section 4.2.

4.2. Discretized depth-averaged continuity equation

In vector form, the depth-averaged continuity equation (11) is given by

$$\frac{\partial h}{\partial t}+\nabla_{\mathrm{H}}\cdot\int_{-d}^{h}\mathbf{u}_{\mathrm{H}}\,\mathrm{d}z=0, \tag{37}$$

where the horizontal gradient and velocity vectors are given in Eqs. (5) and (14). If A_i is the area of the unstructured-grid cell with N_s sides and free-surface height h_i, following Casulli and Zanolli (2002), a semi-implicit discretization of Eq. (37) using the theta-method is given by

$$\frac{h^{n+1}_i-h^n_i}{\Delta t}+\frac{1}{A_i}\theta\sum_{m=1}^{N_s}\sum_{k=1}^{N_{ke}}U^{n+1}_{m,k}\Delta z^{\mathrm{uw}}_{i,k}N_m\,\mathrm{d}f_m+\frac{1}{A_i}(1-\theta)\sum_{m=1}^{N_s}\sum_{k=1}^{N_{ke}}U^{n}_{m,k}\Delta z^{\mathrm{uw}}_{i,k}N_m\,\mathrm{d}f_m=0, \tag{38}$$

where the thickness of the uppermost wet cell is evaluated at time level n, and where N_{ke} is the number of faces in the vertical at side m, $\mathrm{d}f_m$ is the length of the mth side of the polygon, and $U_{m,k}$ is the normal component of the velocity at the mth side of the polygon at the kth depth level. Because the cell thickness $\Delta z_{i,k}$ is stored at

the Voronoi points, we approximate the face-values with their upwind quantities $\Delta z^{\mathrm{uw}}_{i,k}$. In this formulation the free surface h_i represents the location of the top face of the uppermost wet cell, and cells above it are assumed to have zero height when they are dry. Although the present applications do not incorporate wetting and drying because of the stability limitation imposed by vertical advection of momentum when Δz becomes small (see Section 4.7), we have implemented a semi-Lagrangian momentum advection scheme (Le Roux et al., 1997, 2000) for use in estuarine studies with extensive wetting and drying. We leave a discussion of this formulation and its application to a future manuscript.

Since Eq. (38) employs the velocity at the new time step, this requires that the free surface at the new time step include the effects of the nonhydrostatic pressure at that time step. If the corrector step for the horizontal velocity is given by

$$U^{n+1}_{j,k} = U^{*}_{j,k} - \Delta t\left(\frac{qc_{G_{2j},k} - qc_{G_{2j+1},k}}{D_j}\right), \tag{39}$$

where qc denotes the pressure correction (see Section 4.6), substitution into Eq. (38) gives

$$\begin{aligned}\frac{h^{n+1}_i - h^{n}_i}{\Delta t} + \frac{1}{A_i}\theta\sum_{m=1}^{N_s}\sum_{k=1}^{N_{ke}} U^{*}_{m,k}\Delta z^{\mathrm{uw}}_{i,k}N_m\,\mathrm{d}f_m + \frac{1}{A_i}(1-\theta)\sum_{m=1}^{N_s}\sum_{k=1}^{N_{ke}} U^{n}_{m,k}\Delta z^{\mathrm{uw}}_{i,k}N_m\,\mathrm{d}f_m \\ = \frac{\Delta t}{A_i}\theta\sum_{m=1}^{N_s}\frac{1}{D_m}\sum_{k=1}^{N_{ke}}(qc_{G_{2m},k} - qc_{G_{2m+1},k})\Delta z^{\mathrm{uw}}_{i,k}N_m\,\mathrm{d}f_m,\end{aligned} \tag{40}$$

which gives the position of the free-surface at time step $n+1$ in terms of the predictor velocity field and the nonhydrostatic pressure-correction qc. A "fully" nonhydrostatic method would require an iteration in order to solve this equation, since qc is computed from the divergence of the predictor velocity field U^*, which in turn requires the new free-surface h^{n+1}. However, as described by Armfield and Street (2000), because $qc = \mathcal{O}(\Delta t)$ in the pressure correction algorithm, an iteration is not required in order to attain second-order accuracy. Therefore, in the present formulation we omit this term from the free-surface equation as the added expense associated with a fully nonhydrostatic iteration does not affect the time accuracy of the method.

In a manner similar to that of Casulli (1999) and Casulli and Zanolli (2002), the free-surface equation is solved via substitution of the discrete horizontal momentum equation for U^* given by Eq. (31) into (40). This yields a symmetric, positive-definite, linear system in which the bandwidth is a strong function of the ordering of the unstructured cells, and the linear system is solved with the preconditioned conjugate gradient algorithm using diagonal preconditioning. Once this equation is solved for h^{n+1}, the predicted horizontal velocity field can be obtained from Eq. (31).

4.3. Advection of momentum

The control volume used to discretize the advection and diffusion terms for the horizontal and vertical momentum equations is shown in Fig. 4. For a Delaunay triangulation, the Voronoi edges, which are the lines connecting the cell circumcenters (denoted by the open circles in Fig. 4(a)) are perpendicular to the faces of the cells, which are the Delaunay edges. The distance from the circumcenter of cell i to face j is given by $d_{i,j}$, and the distance between two circumcenters adjacent to face j is D_j, as was shown in Section 3. Fig. 4 depicts face j and its two neighboring cells, whose indices are given by G_{2j} and G_{2j+1}. In what follows, for ease of notation we assume that $G_{2j} = 1$ and $G_{2j+1} = 2$, so that $d_{G_{2j},j} = d_{1,j}$ and $d_{G_{2j+1},j} = d_{2,j}$.

Following the work of Perot (2000), a discretization of the vertical advection of horizontal momentum at face j in Fig. 4 is given by

$$C_{\mathrm{V}}(U_{j,k}) = \frac{1}{D_j}(d_{1,j}C^{vu}_{1,k} + d_{2,j}C^{vu}_{2,k}). \tag{41}$$

The component of vertical advection at the Voronoi point in cell i in the direction of the normal vector $\mathbf{n}_j$ is given by

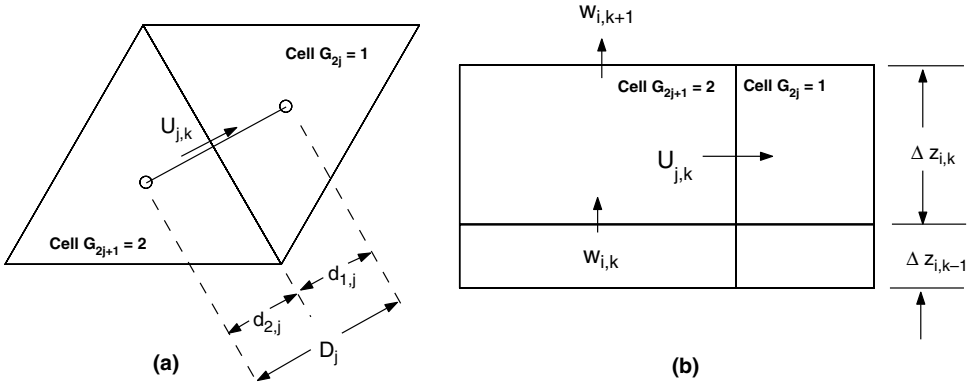

Fig. 4. The control volume used to discretize the horizontal momentum equation at face j, showing the (a) planform and (b) side views.

$$C_{i,k}^{vu} = \frac{1}{\Delta z_{i,k}}(w_{i,k+1}\mathbf{n}_j \cdot \mathbf{u}_{i,k+1/2} - w_{i,k}\mathbf{n}_j \cdot \mathbf{u}_{i,k-1/2}), \tag{42}$$

where $\mathbf{u}_{i,k}$ is the approximation of the velocity vector at the Voronoi point, noting that $\mathbf{u}_{i,k-1/2}$ is located at the same point as $w_{i,k}$. Following Perot, the velocity vector is defined at the Voronoi points with

$$\mathbf{u}_{i,k} = \frac{1}{A_i}\sum_{m=1}^{N_s} U_m \mathbf{n}_m d_{i,m}\,\mathrm{d}f_m. \tag{43}$$

Approximating this velocity vector with a linear interpolation at the upper and lower faces yields

$$\mathbf{u}_{i,k-1/2} = \frac{\Delta z_{i,k-1}}{\Delta z_{i,k} + \Delta z_{i,k-1}}\mathbf{u}_{i,k} + \frac{\Delta z_{i,k}}{\Delta z_{i,k} + \Delta z_{i,k-1}}\mathbf{u}_{i,k-1}, \tag{44}$$

$$\mathbf{u}_{i,k+1/2} = \frac{\Delta z_{i,k+1}}{\Delta z_{i,k} + \Delta z_{i,k+1}}\mathbf{u}_{i,k} + \frac{\Delta z_{i,k}}{\Delta z_{i,k} + \Delta z_{i,k+1}}\mathbf{u}_{i,k+1}. \tag{45}$$

Vertical advection of vertical momentum is obtained with a similar discretization, which reduces to a second-order central difference of the form

$$C_{\mathrm{V}}(w_{i,k}) = \frac{1}{\Delta z_{i,k-1} + \Delta z_{i,k}}(w_{i,k+1}^2 - w_{i,k-1}^2). \tag{46}$$

Conservation of momentum and energy is satisfied everywhere except for cells containing the free-surface, where conservation is only approximate. This is because the free surface is stored at the Voronoi points, while the free surface in the formulation of Perot, which employs fully unstructured tetrahedra, is stored at the Delaunay points, thereby enabling conservation.

Horizontal advection of horizontal momentum at face j in Fig. 4 is discretized in a conservative manner (Perot, 2000) with

$$C_{\mathrm{H}}(U_{j,k}) = \frac{1}{D_j}(d_{1,j}C_{1,k}^{hu} + d_{2,j}C_{2,k}^{hu}). \tag{47}$$

The component of horizontal advection of horizontal momentum in the direction of the normal vector $\mathbf{n}_j$ within cell i is obtained by integrating the horizontal advection term over the finite volume V_i to yield

$$C_{i,k}^{hu} = \frac{\mathbf{n}_j}{V_i} \cdot \int_{V_i} \nabla_{\mathrm{H}} \cdot (\mathbf{u}_{\mathrm{H}}\mathbf{u})\,\mathrm{d}V, \tag{48}$$

which, after application of Green's theorem, becomes

$$C_{i,k}^{hu} = \frac{\mathbf{n}_j}{A_i \Delta z_{i,k}} \cdot \int_S \mathbf{u}(\mathbf{u}_{\mathrm{H}} \cdot \mathbf{n})\,\mathrm{d}S, \tag{49}$$

where S is the surface of the control volume. This is approximated with

$$C^{hu}_{i,k} = \frac{1}{A_i \Delta z_{i,k}} \sum_m^{N_s} \mathbf{n}_j \cdot \mathbf{u}_{m,k} U_{m,k} \Delta z^{\mathrm{uw}}_{m,k} N_m \, \mathrm{d}f_m, \tag{50}$$

where the face-centered velocity vector is given by a linear interpolation with the velocity vectors on either side of face m (which are computed using Eq. (43)) with

$$\mathbf{u}_{m,k} = \frac{1}{D_m} (d_{G_{2m+1},m} \mathbf{u}_{G_{2m},k} + d_{G_{2m},m} \mathbf{u}_{G_{2m+1},k}). \tag{51}$$

Horizontal advection of vertical momentum is computed in a similar manner with

$$C_{\mathrm{H}}(w_{i,k}) = \frac{1}{\Delta z_{i,k} + \Delta z_{i,k-1}} (\Delta z_{i,k-1} C^{hw}_{i,k-1} + \Delta z_{i,k} C^{hw}_{i,k}), \tag{52}$$

where

$$C^{hw}_{i,k} = \frac{1}{A_i \Delta z^{\mathrm{uw}}_{i,k}} \sum_m^{N_s} w_{m,k+1/2} U_{m,k} \Delta z^{\mathrm{uw}}_{i,k} N_m \, \mathrm{d}f_m. \tag{53}$$

Because the vertical velocity is stored at the horizontal faces, it is interpolated to the vertical faces with

$$\mathbf{w}_{m,k+1/2} = \frac{1}{2D_m} (d_{G_{2m+1},m} w_{G_{2m},k} + d_{G_{2m},m} w_{G_{2m+1},k}) + \frac{1}{2D_m} (d_{G_{2m+1},m} w_{G_{2m},k+1} + d_{G_{2m},m} w_{G_{2m+1},k+1}). \tag{54}$$

4.4. Horizontal diffusion of momentum

Horizontal diffusion of horizontal momentum is discretized using the same conservative discretization as for advection, such that

$$D_{\mathrm{H}}(U_{j,k}) = \frac{1}{D_j} (d_{1,j} D^{hu}_{1,k} + d_{2,j} D^{hu}_{2,k}). \tag{55}$$

The component of horizontal diffusion of horizontal momentum in the direction of the normal vector $\mathbf{n}_j$ within cell i is obtained by integrating the horizontal diffusion term over the finite volume V_i to yield

$$D^{hu}_{i,k} = \frac{\mathbf{n}_j}{V_i} \cdot \int_{V_i} \nabla_{\mathrm{H}} \cdot (\nu_{\mathrm{H}} \nabla_{\mathrm{H}} \mathbf{u}) \, \mathrm{d}V, \tag{56}$$

which, after application of Green's theorem, becomes

$$D^{hu}_{i,k} = \frac{\mathbf{n}_j}{A_i \Delta z_{i,k}} \cdot \int_S \nu_{\mathrm{H}} \frac{\partial \mathbf{u}}{\partial n} \mathrm{d}S, \tag{57}$$

where S is the surface of the control volume. This is approximated with

$$D^{hu}_{i,k} = \frac{1}{A_i \Delta z_{i,k}} \sum_m^{N_s} (\nu_{\mathrm{H}})_{m,k} \mathbf{n}_j \cdot \left. \frac{\partial \mathbf{u}}{\partial n} \right|_{m,k} \Delta z^{\mathrm{uw}}_{i,k} N_m \, \mathrm{d}f_m, \tag{58}$$

where the face-normal gradient is given by

$$\left. \frac{\partial \mathbf{u}}{\partial n} \right|_{m,k} = \frac{1}{D_m} (\mathbf{u}_{G_{2m},k} - \mathbf{u}_{G_{2m}+1,k}), \tag{59}$$

and the eddy viscosity is approximated with the linear interpolation

$$(\nu_{\mathrm{H}})_{m,k} = \frac{1}{D_m} (d_{G_{2m+1},m} (\nu_{\mathrm{H}})_{G_{2m},k} + d_{G_{2m},m} (\nu_{\mathrm{H}})_{G_{2m+1},k}). \tag{60}$$

For horizontal diffusion of vertical momentum, the discretization is obtained in a similar manner and is given by

$$D_{\mathrm{H}}(w_{i,k}) = \frac{1}{\Delta z_{i,k} + \Delta z_{i,k-1}} (\Delta z_{i,k} D^{hw}_{i,k} + \Delta z_{i,k-1} D^{hw}_{i,k-1}), \tag{61}$$

where

$$D^{hw}_{i,k} = \frac{1}{A_i \Delta z_{i,k}} \sum_{m}^{N_s} (\nu_{\mathrm{H}})_{m,k} \left.\frac{\partial w}{\partial n}\right|_{m,k+1/2} \Delta z^{\mathrm{uw}}_{m,k} N_m \, \mathrm{d}f_m. \tag{62}$$

The horizontal gradient of the vertical velocity is approximated with

$$\left.\frac{\partial w}{\partial n}\right|_{m,k+1/2} = \frac{1}{2D_m} (w_{G_{2j},k+1} - w_{G_{2j+1},k+1} + w_{G_{2j},k} - w_{G_{2j+1},k}). \tag{63}$$

For solid vertical boundaries, we impose a drag law for the horizontal and vertical velocities of the form

$$\nu_{\mathrm{H}} \left.\frac{\partial \mathbf{u}}{\partial n}\right|_b = C_{\mathrm{d}W} |\mathbf{u}_{G_{2b}}| \mathbf{u}_{G_{2b}}, \tag{64}$$

$$\nu_{\mathrm{H}} \left.\frac{\partial w}{\partial n}\right|_b = C_{\mathrm{d}W} |w_{G_{2b}}| w_{G_{2b}}, \tag{65}$$

where the b index denotes a solid vertical wall, and $C_{\mathrm{d}W}$ is the drag coefficient for vertical walls. By definition, the gradient pointer G_{2b} points to the cell adjacent to boundaries and $G_{2b+1} = -1$.

4.5. Discretized transport equations

The general advection–diffusion equation for an arbitrary scalar quantity ϕ, which may be either s, T, or ρ, is given by

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = \nabla \cdot (\alpha_0 \nabla \phi), \tag{66}$$

where $\alpha_0 = \alpha_{0,\mathrm{H}} + \alpha_{0,\mathrm{V}}$ is the scalar diffusivity. We split α_0 into its horizontal and vertical parts later in the discretization for ease of notation. Integrating (66) over a cross-sectional area A and from z_1 to $z_1 + h$ and applying Green's theorem gives

$$\int_A \int_{z_1}^{z_1+h} \frac{\partial \phi}{\partial t} \, \mathrm{d}z \, \mathrm{d}A = \int_S (\alpha_0 \nabla \phi - \phi \mathbf{u}) \cdot \mathbf{n} \, \mathrm{d}S, \tag{67}$$

where S is the surface encompassing the elemental control volume $\mathrm{d}V = \mathrm{d}z\,\mathrm{d}A$. Using Leibniz' rule, we have

$$\frac{\partial}{\partial t} \left[\int_{\mathrm{V}} \phi \, \mathrm{d}V \right] - \int_A \phi(h) \frac{\partial h}{\partial t} \, \mathrm{d}A = \int_S (\alpha_0 \nabla \phi - \phi \mathbf{u}) \cdot \mathbf{n} \, \mathrm{d}S. \tag{68}$$

If S_{NF} represents all control volume surfaces other than the free surface, this can be rewritten as

$$\frac{\partial}{\partial t} \left[\int_{\mathrm{V}} \phi \, \mathrm{d}V \right] - \int_A \phi(h) \left(\frac{\partial h}{\partial t} - \mathbf{u}(h) \cdot \mathbf{n}_h \right) \mathrm{d}A = \int_{S_{NF}} (\alpha_0 \nabla \phi - \phi \mathbf{u}) \cdot \mathbf{n} \, \mathrm{d}S, \tag{69}$$

where $\mathbf{n}_h$ is the vector normal to the free surface, and we have assumed no scalar gradients normal to the free surface, such that $\nabla \phi \cdot \mathbf{n}_h = 0$. Since $\mathbf{n}_h$ is given by

$$\mathbf{n}_h = -\nabla h + \mathbf{e}_3, \tag{70}$$

applying the kinematic free-surface boundary condition at $z = h$ (Eq. (12)) to Eq. (69) yields

$$\frac{\partial}{\partial t}\left[\int_V \phi \,\mathrm{d}V\right] = \int_{S_{NF}} (\alpha_0 \nabla\phi - \phi\mathbf{u})\cdot\mathbf{n}\,\mathrm{d}S. \tag{71}$$

In this manner a conservative scalar transport algorithm can be derived for control volumes that include the free surface. Control volumes that do not include the free surface employ Eq. (71), except with $S_{NF} = S$.

A conservative time discretization of (71) is given by

$$\begin{aligned}\frac{\Delta z^{n+1}_{i,k}\phi^{n+1}_{i,k} - \Delta z^{n}_{i,k}\phi^{n}_{i,k}}{\Delta t} = {} & \frac{1}{2}(3\Phi^{n}_{i,k} - \Phi^{n-1}_{i,k}) - \frac{1}{A_i}\sum_{m}^{N_s}[\theta U^{n+1}_{m,k} + (1-\theta)U^{n}_{m,k}]\phi^{n}_{m,k}\Delta z^{\mathrm{uw}}_{m,k}N_m\,\mathrm{d}f_m \\ & - \theta\left(w^{n+1}_{i,k+1}\phi^{n+1}_{i,k+1/2} - w^{n+1}_{i,k}\phi^{n+1}_{i,k-1/2}\right) - (1-\theta)\left(w^{n}_{i,k+1}\phi^{n}_{i,k+1/2} - w^{n}_{i,k}\phi^{n}_{i,k-1/2}\right) \\ & + \theta D_\phi\phi^{n+1}_{i,k} + (1-\theta)D_\phi\phi^{n}_{i,k},\end{aligned} \tag{72}$$

where D_ϕ is the vertical diffusion operator and contains the vertical scalar diffusivity $\alpha_{0,\mathrm{V}}$. The vertical diffusion and vertical advection have been discretized with the theta-method (Casulli, 1999), and the horizontal diffusion term, $\Phi_{i,k}$, is discretized with the Adams–Bashforth method (Zang et al., 1993), and is given by

$$\Phi_{i,k} = \frac{1}{A_i}\int_{S_\mathrm{H}} \alpha_{0,H}\nabla\phi\cdot\mathbf{n}\,\mathrm{d}S, \tag{73}$$

where S_H implies integration only over the vertical faces. This is approximated with

$$\Phi_{i,k} = \frac{1}{A_i}\sum_{m}^{N_s}\frac{(\alpha_{0,H})_{m,k}}{D_m}(\phi_{G_{2m},k} - \phi_{G_{2m+1},k})\Delta z^{\mathrm{uw}}_{m,k}N_m\,\mathrm{d}f_m, \tag{74}$$

where the scalar diffusivity $\alpha_{0,\mathrm{H}}$ at the cell faces is obtained with a linear interpolation from the neighboring cells, as in Eq. (60). As discussed in Gross et al. (2002), the discretization given in Eq. (72) is guaranteed to conserve mass because it is consistent with the discrete depth-averaged continuity equation (38) and the discrete continuity equation (78) (see Section 4.6).

The value of the scalar quantity at the faces can be determined in a number of ways, depending on the desired scalar advection scheme. We employ first-order upwinding for the examples in this paper. Although highly diffusive, we have found it to be suitable to demonstrate code performance and other salient features of the present formulation.

4.6. Nonhydrostatic pressure

In general, while the predicted velocity field given by U^* and w^* in Eqs. (31) and (32) satisfies depth-averaged continuity, it does not satisfy local continuity. Therefore, a nonhydrostatic pressure field must be computed that enforces the continuity equation (4). This is done by correcting the horizontal and vertical velocity fields with the nonhydrostatic pressure-correction qc, as in

$$U^{n+1}_{j,k} = U^{*}_{j,k} - \Delta t\left(\frac{qc_{G_{2j},k} - qc_{G_{2j+1},k}}{D_j}\right), \tag{75}$$

$$w^{n+1}_{i,k} = w^{*}_{i,k} - 2\Delta t\left(\frac{qc_{i,k} - qc_{i,k-1}}{\Delta z_{i,k} + \Delta z_{i,k-1}}\right), \tag{76}$$

followed by an update of the nonhydrostatic pressure field with

$$q^{n+1/2}_{i,k} = q^{n-1/2}_{i,k} + qc_{i,k}. \tag{77}$$

Following Armfield and Street (2000), the time level of the nonhydrostatic pressure field is specified at the half-time steps in order to maintain second-order temporal accuracy. This method is termed the pressure-correction algorithm and differs from the methodology in Casulli (1999) in that it is second-order accurate in time

while the Casulli method is first-order accurate in time. Second-order temporal accuracy of the present method is demonstrated in Section 6.1.

A Poisson equation for the pressure correction is derived by integrating the continuity equation (4) at time step $n+1$ over a cell with N_s sides and assuming that the velocity field is divergence-free at time step $n+1$ to yield the discrete continuity equation

$$A_i\left(w_{i,k+1}^{n+1}-w_{i,k}^{n+1}\right)+\sum_{m=1}^{N_s}U_{m,k}^{n+1}\Delta z_{m,k}^{\mathrm{uw}}N_m\,\mathrm{d}f_m=0, \tag{78}$$

where $\Delta z_{m,k}^{\mathrm{uw}}$ is the upwind cell height at face m in order to be consistent with the discrete depth-averaged continuity equation (38), as discussed in Gross et al. (2002). Substitution of the corrector steps (75) and (76) into (78) yields the discrete Poisson equation for the nonhydrostatic pressure-correction field as

$$\mathscr{L}(qc_{i,k})=S_{i,k}^{*}, \tag{79}$$

where the source term is given by

$$S_{i,k}^{*}=\frac{1}{\Delta t}\left[A_i(w_{i,k+1}^{*}-w_{i,k}^{*})+\sum_{m=1}^{N_s}U_{m,k}^{*}\Delta z_{m,k}^{\mathrm{uw}}N_m\,\mathrm{d}f_m\right], \tag{80}$$

and the Poisson operator is given by

$$\begin{aligned}\mathscr{L}(qc_{i,k})=&\sum_{m=1}^{N_s}\left(\frac{qc_{Ne_m,k}-qc_{i,k}}{D_m}\right)\Delta z_{m,k}^{\mathrm{uw}}\,\mathrm{d}f_m+\frac{2A_i}{\Delta z_{i,k}+\Delta z_{i,k-1}}qc_{i,k-1}\\&-2A_i\left(\frac{1}{\Delta z_{i,k}+\Delta z_{i,k-1}}+\frac{1}{\Delta z_{i,k}+\Delta z_{i,k+1}}\right)qc_{i,k}+\frac{2A_i}{\Delta z_{i,k}+\Delta z_{i,k+1}}qc_{i,k+1}.\end{aligned} \tag{81}$$

We solve Eq. (79) with the preconditioned conjugate gradient algorithm with the preconditioner described in Section 6.2. Following a solution of (79), the divergence-free horizontal velocity field at time step $n+1$ can be obtained with Eq. (75) and the vertical velocity field is obtained by starting with $w_{i,1}^{n+1}=0$ at the lower boundary, and using the discrete continuity equation (78) to obtain the velocity field with

$$w_{i,k+1}^{n+1}=w_{i,k}^{n+1}-\frac{1}{A_i}\sum_{m=1}^{N_s}U_{m,k}^{n+1}\Delta z_{m,k}^{\mathrm{uw}}N_m\,\mathrm{d}f_m, \tag{82}$$

where $k=1,2,\ldots,N_{kc}$, and N_{kc} is the number of vertical levels within fluid column i. The nonhydrostatic pressure field is then updated with Eq. (77).

If the pressure field is hydrostatic, then $qc=0$ and we do not need to solve for the vertical momentum in Eq. (32). In this case, $U^{n+1}=U^{*}$, and the vertical velocity is obtained by continuity with Eq. (82). Boundary conditions on the nonhydrostatic correction are gradient-free at solid boundaries and $qc=0$ at the free surface. At open boundaries, since the velocity field at time step $n+1$ is known, this allows specification of the predictor velocity field $U_{\mathrm{boundary}}^{*}=U_{\mathrm{boundary}}^{n+1}$, which implies a no-gradient condition on the nonhydrostatic pressure after imposing Eq. (75). Justification and details on this strategy can be found in the work of Zang et al. (1994).

4.7. Stability

The numerical method in this paper is mass- and volume-conservative, and because of the use of the theta-method for vertical diffusion and the fast free-surface gravity waves, stability is not limited by these terms. However, in order to avoid oscillations associated with the Crank–Nicolson scheme for the barotropic terms that would result when $\theta=0.5$ (since it is neutrally stable, as in Casulli, 1990), we have found through trial and error that $\theta=0.55$ provides enough damping to eliminate the high-frequency oscillations without adversely affecting the simulations. Stability for the present formulation is limited by internal gravity waves because the gravity terms in the horizontal momentum equation are discretized explicitly with the Adams–Bashforth method. Stability is also limited by the explicit discretization of the horizontal and vertical

advection and horizontal diffusion terms for momentum and horizontal advection and horizontal diffusion for scalars. Stability of the advection–diffusion terms on unstructured grids can be approximated with the two-dimensional (horizontal–vertical) stability limitations on Cartesian grids with the horizontal axis parallel to a face-normal, **n**. Following the limitations derived by Fletcher (1997) when explicit central differencing is employed for advection of momentum and horizontal diffusion is treated explicitly, the time step is limited by

$$\left(\frac{|U|\Delta t}{D_j}+\frac{|w|\Delta t}{\Delta z}\right)^2 \leqslant \frac{2\nu_H \Delta t}{D_j^2} \leqslant 1. \tag{83}$$

Assuming a stable interpolation scheme is employed for horizontal advection of scalars, stability for the scalar advection equations is determined by

$$\Delta t \leqslant \min\left(\frac{D_j}{|U|}, \frac{D_j^2}{\max(\gamma_H, \kappa_H)}\right). \tag{84}$$

Furthermore, if c_i is the maximum internal gravity wave speed, then the time step is also limited by

$$\Delta t \leqslant \frac{D_j}{c_i}. \tag{85}$$

Although the time step is generally limited by accuracy in our simulations, the most limiting of these stability conditions for coastal applications is the internal gravity wave limitation, while for estuarine applications, the most limiting is the vertical advection of momentum. Computational overhead associated with small time steps is offset by the ability to attain significant speedup by performing simulations in parallel, as discussed in Section 6.3.

5. Test cases

We present here three test cases that explore the important features of the code, the role of nonhydrostatic pressure, and the range of applicability of the code from laboratory-scale to field-scale cases. In the internal seiche and lock-exchange examples, the nonhydrostatic and variable density capabilities are featured. The internal seiche case also shows the impact of the nonhydrostatic formulation on the free surface behavior. The examination of internal waves in Monterey Bay demonstrates a field-scale application and provides the basis for the estimates of code performance used in Section 6.3.

5.1. Nonhydrostatic internal seiche

As a test case, we compute the oscillations of a nonhydrostatic internal seiche to highlight some fundamental aspects of nonhydrostatic physics that must be captured correctly in nonhydrostatic simulations of internal waves. This test case is similar to the free-surface seiche computed by Casulli (1999), except the internal seiche case demonstrates the coupling of the density stratification as well as the free surface to the nonhydrostatic pressure. As depicted in Fig. 5, the initial density stratification is given by

$$\rho(x,z,t=0) = -\frac{\Delta\rho}{2\rho_0}\tanh\left[\frac{2\tanh^{-1}\alpha_s}{\delta}\left(z+\frac{D}{2}-a\cos(kx)\right)\right], \tag{86}$$

where $\Delta\rho/\rho_0$ is the top–bottom density difference, D is the depth, a is the amplitude of the seiche, k is the horizontal wavenumber, δ is the interface thickness, and $\alpha_s = 0.99$. The frequency of oscillation of a small-amplitude nonhydrostatic internal seiche between two fluid layers separated by a finite-thickness interface is given by (see, e.g. Kundu, 1990)

$$\omega^2 = \frac{g'k}{2}\tanh\left(\frac{kD}{2}\right)f(k\delta), \tag{87}$$

where $g' = g\Delta\rho/\rho_0$ is the reduced gravity, and $f(k\delta)$ accounts for finite-interface effects. From Thorpe (1968), this function is given to first order in $k\delta$ as

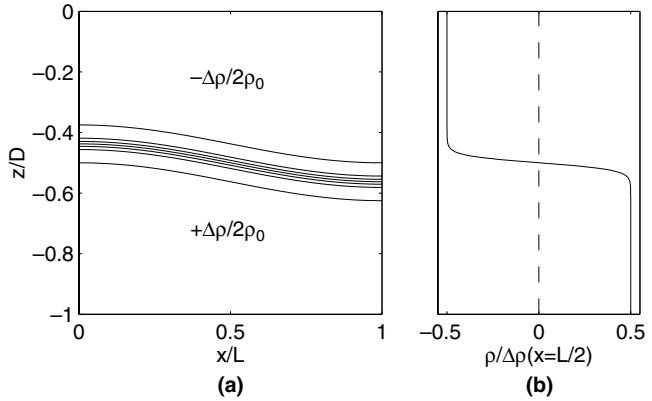

Fig. 5. Initial density field of the internal seiche test case showing (a) the interface with an amplitude five times that used in the tests, and (b) the density profile at $x = L/2$.

$$f(k\delta) = \left(1 + \frac{k\delta}{2}\right)^{-1}. \tag{88}$$

Eq. (87) represents the linearized small-amplitude frequency of the internal seiche in the limit of small steepness ka. The speed of propagation of the right- and left-propagating waves which superpose to form the standing wave is given by

$$c^2 = \frac{g'}{2k}\tanh\left(\frac{kD}{2}\right) f(k\delta), \tag{89}$$

which demonstrates the important property that the frequency-dispersive behavior of internal gravity waves, which is a nonhydrostatic effect, arises for linear waves in the limit of small steepness ka. This is not to be confused with amplitude dispersion for gravity waves, which is a nonlinear effect in the presence of finite steepness ka, and does not necessarily imply nonhydrostatic behavior. In the limit of deep water, for which $kD \to \infty$, the phase speed becomes independent of the depth, such that

$$\lim_{kD\to\infty} c^2 = c_{\mathrm{DW}}^2 = \frac{g'}{2k} f(k\delta), \tag{90}$$

whereas in the limit of shallow water, for which $kD \to 0$, the phase speed is no longer frequency dispersive and reverts to the shallow-water wave speed, for which

$$\lim_{kD\to 0} c^2 = c_{\mathrm{SW}}^2 = \frac{g'D}{4} f(k\delta). \tag{91}$$

While the finite interface thickness introduces frequency dispersion for a fixed interface thickness δ, its dispersive character is negligible when compared to the dispersive character of the infinitesimally thin interface case, for which $f(k\delta) = 1$. Dividing (89) by the deep-water limit (90) gives

$$\left(\frac{c}{c_{\mathrm{DW}}}\right)^2 = \tanh\left(\frac{\pi\epsilon}{2}\right), \tag{92}$$

and the ratio of the shallow-water to the deep-water limits is given by

$$\left(\frac{c_{\mathrm{SW}}}{c_{\mathrm{DW}}}\right)^2 = \frac{\pi\epsilon}{2}, \tag{93}$$

where $\epsilon = D/L$ is the aspect ratio of the wave and is a measure of its nonhydrostacy. That is, waves propagate as hydrostatic waves when the horizontal extent is much larger than the depth, for which $\epsilon \ll 1$, or when $c = c_{\mathrm{SW}}$, and they become frequency dispersive when the horizontal extent of the wave is on the order of the depth or greater.

To demonstrate the behavior of the numerical method outlined in this paper in the hydrostatic and nonhydrostatic limits, we compute the effect of varying the depth on the oscillations of the internal seiche depicted in Fig. 5 with $\Delta\rho/\rho_0 = 0.06$, $L = 100$ m, $\delta = 5$ m, and $a = 1$ m for a total of 250 s with a time step of $\Delta t = 0.025$ s. We employ an equispaced grid both in the horizontal (one dimension of equilateral triangles in the planform) and vertical directions, with $N_x = 100$ horizontal grid cells, and the number of vertical grid cells is set to maintain a vertical grid spacing of 0.5 m. The boundary conditions are free-slip on the walls, the free surface is initially at rest, the initial stratification is given in Eq. (86), and there is no eddy viscosity or scalar diffusivity. Although momentum advection in SUNTANS is theoretically unstable without viscosity, this inviscid test case remains stable because of the relatively weak effect of the nonlinear terms. Likewise, the numerical diffusion associated with first-order upwinding has a minimal effect on the results.

In order to analyze the nonhydrostatic behavior of the seiche, we compute the oscillations with and without the nonhydrostatic pressure with the length fixed at $L = 100$ m and with $D = 10$, 20, 40, 80, and 160 m, which represents aspect ratios of $\epsilon = 0.1$, 0.2, 0.4, 0.8, and 1.6. With these depths, the fastest nonhydrostatic wave is given by the deep-water limit of $c_{\mathrm{DW}} = 2.83$ m s^{-1}, and the fastest hydrostatic wave is given by the shallow-water limit for the deepest case, which is $c_{\mathrm{SW}} = 4.51$ m s^{-1}. Stability for the hydrostatic calculation is then limited by the shallow-water internal wave speed. With a time step of $\Delta t = 0.025$ s that is fixed throughout the calculations, this yields a maximum shallow-water Courant number of $C_{\mathrm{H}} = c_{\mathrm{SW}}\Delta t/\Delta x = 0.11$ for the hydrostatic calculations and a maximum deep-water Courant number of $C_{\mathrm{NH}} = c_{\mathrm{DW}}\Delta t/\Delta x = 0.07$ for the nonhydrostatic calculations.

Fig. 6 depicts the velocity vectors plotted over the density contours at $t = T/4$ for the hydrostatic and nonhydrostatic calculations when $\epsilon = 1.6$, where T is the wave period for each case. As expected, the hydrostatic solution approximates the wave as a shallow water wave and hence the velocity fields do not decay with distance from the interface, as they should and do for the nonhydrostatic calculation. This is demonstrated by the profile plots in Fig. 7, which compare the velocity profiles of the hydrostatic and nonhydrostatic results to the first-mode velocity profiles computed with a linearized eigenfunction analysis (see, e.g., Fringer and Street, 2003) of the density field given in Eq. (86). The first-mode eigenfunctions decay exponentially with distance from the interface, and this is a linear, nonhydrostatic effect, which is corroborated by the nonhydrostatic results. The hydrostatic horizontal velocity profile, on the other hand, is constant with depth away

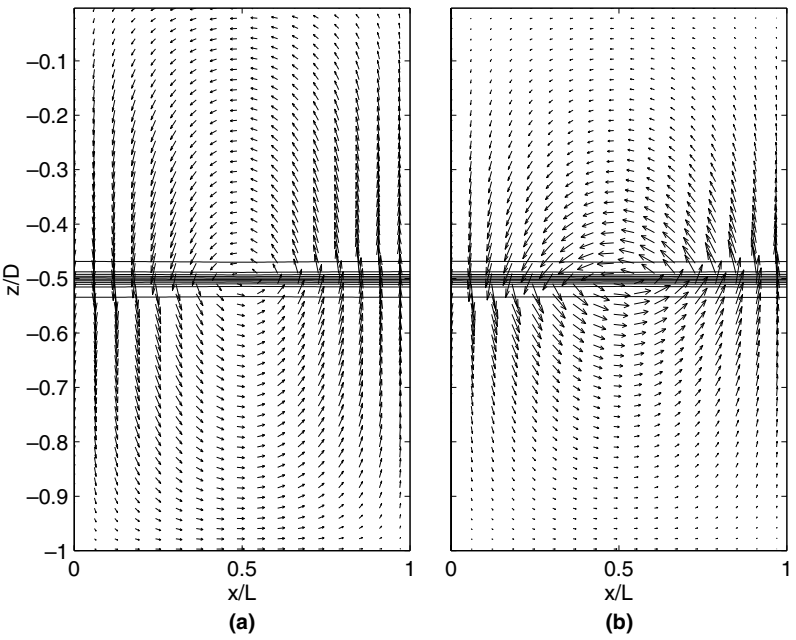

Fig. 6. Density and velocity fields of the internal seiche calculations at $t = T/4$ using the (a) hydrostatic and (b) nonhydrostatic solvers when the aspect ratio is $\epsilon = 1.6$. Density contours are plotted as solid black lines in increments of $0.1\Delta\rho$ between $+\alpha_s\Delta\rho/2$ and $-\alpha_s\Delta\rho/2$.

from the interface, while the hydrostatic vertical velocity profile decays linearly with distance from the interface and is nonzero at the free surface, indicating that the free-surface deflection for the hydrostatic case is much greater than it is for the nonhydrostatic case.

The hydrostatic solution under-approximates the maximum horizontal velocity by roughly 24%, while it over-approximates the vertical velocity by roughly 68%. This is consistent with the neglect of vertical inertia in the hydrostatic calculation. The lack of vertical inertia also results in an overprediction of the wave speed in the hydrostatic calculation when nonhydrostatic effects are important. This is depicted in Fig. 8, which compares the calculations of the internal wave speeds to the theoretical predictions given in Eqs. (92) and (93). The hydrostatic calculations always yield the shallow water wave speed and hence the hydrostatic results yield speeds that correspond to the shallow water relation (93). The nonhydrostatic calculations, on the other hand, include the effects of vertical inertia and as a result the waves are limited in speed by their wavelength once the aspect ratio becomes large. The nonhydrostatic calculations accurately predict the wave speeds and the results are consistent with the predictions of the linearized dispersion relation given in Eq. (92). Compared to the predicted results, the slight under-prediction of the calculated results is due to finite-interface and nonlinear effects not accounted for in relations (92) and (93).

The frequency and amplitude of the free-surface response are also overpredicted by the hydrostatic solver. Fig. 9 depicts the response of the free surface to the internal seiche at $x = 0$ over one period of the nonhydrostatic internal seiche when $\epsilon = 1.6$. This figure shows how the free-surface amplitude is six times greater for the hydrostatic calculation than it is for the nonhydrostatic calculation. Fig. 10 depicts the frequency spectrum of the free-surface response for $\epsilon = 1.6$ and shows how, in general, hydrostatic calculations tend to shift the free-surface spectrum up and to the right, since amplitudes as well as frequencies are overpredicted. The first peaks in the spectra correspond to the frequency of the internal seiche, while the second peaks correspond to the natural frequency of the free-surface seiche. Because the free-surface response is large for the hydrostatic

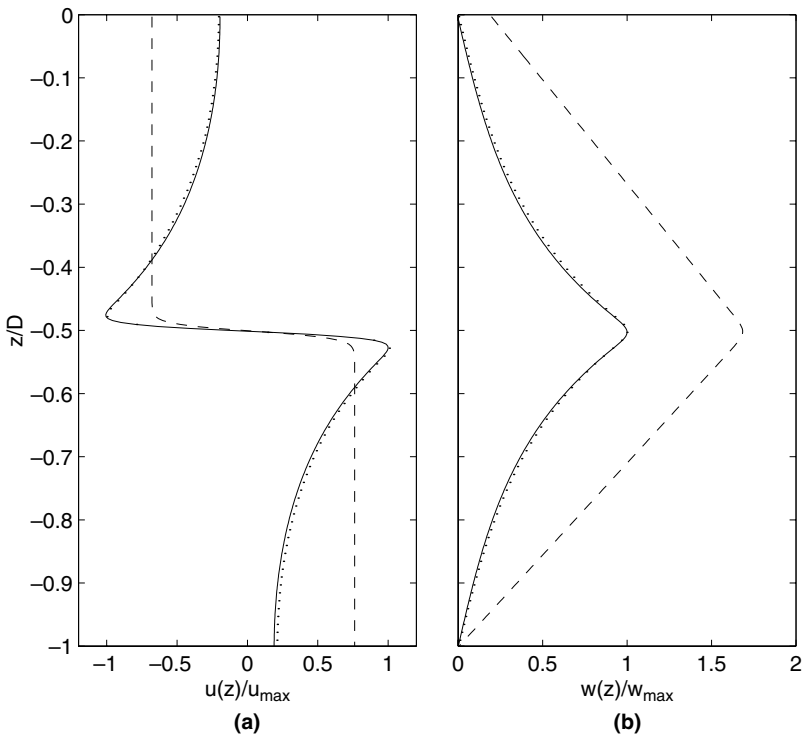

Fig. 7. Horizontal and vertical velocity profiles of the internal seiche calculations at $t = T/4$ using the hydrostatic and nonhydrostatic solvers when the aspect ratio is $\epsilon = 1.6$. Each is normalized by the maximum nonhydrostatic velocity along that particular profile at $t = T/4$. (a) Horizontal velocity, (b) vertical velocity. Legend: nonhydrostatic (–), hydrostatic (– –), first-mode linearized eigenfunction analysis of Fringer and Street (2003) (···).

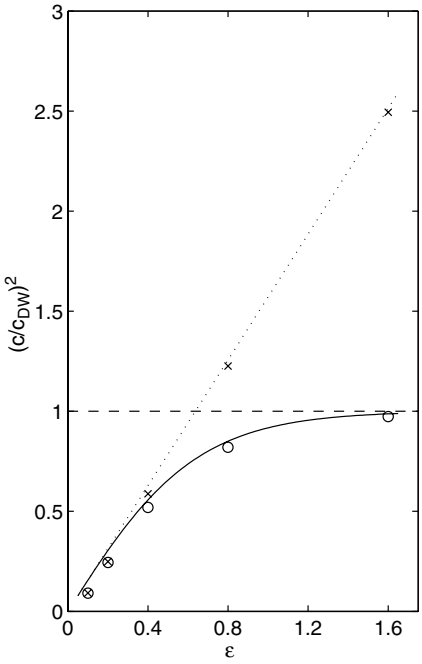

Fig. 8. Effect of the aspect ratio ϵ on the computed wave speed for the hydrostatic calculations (×) and the nonhydrostatic calculations (○). The solid line depicts the linearized dispersion relation given in Eq. (92), while the dotted line (···) depicts the ratio of the shallow-water wave speed to the deep-water speed in Eq. (93). The dashed line (– –) depicts the deep-water limit.

calculation, third and fourth peaks appear in the hydrostatic frequency spectrum, corresponding to the excitement of higher harmonics. The theoretical frequencies of the free-surface seiche can be calculated with relations similar to those in (92) and (93), as was done by Casulli (1999).

5.2. Nonhydrostatic lock exchange

Here we present the results of computing an exchange flow using the parameters of the direct numerical simulations of Hartel et al. (2000) and discuss the differences between the nonhydrostatic and hydrostatic results. The simulations are performed with SUNTANS in a two-dimensional domain of length $L = 0.8$ m and depth $D = 0.1$ m (one dimension of equilateral triangles in the planform) using 400×100 cells, a no-slip condition at the lower boundary, and free-slip at all other boundaries. This configuration allows the study of the propagation speed of the gravity currents under a no-slip condition at the bottom and a free-slip condition at the surface. The horizontal and vertical eddy-viscosities are constant and are set to their molecular values of $\nu_H = \nu_V = 10^{-6}\ \mathrm{m^2\ s^{-1}}$, and there is no physical scalar diffusivity, since the first-order upwind scheme provides

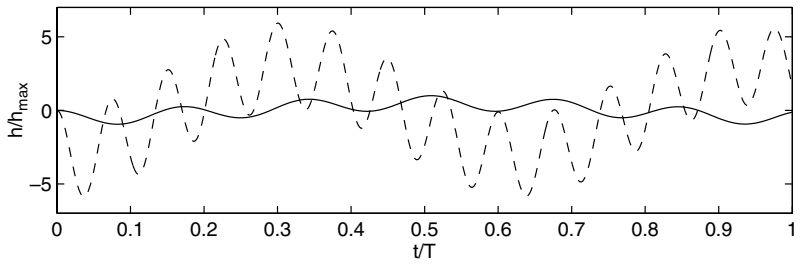

Fig. 9. Response of the free-surface at $x = 0$ to the internal seiche when $\epsilon = 1.6$ normalized by the maximum amplitude of the free surface (i.e. $\max(h(x = 0, t))$) for the nonhydrostatic calculation. The time is normalized by the period of the nonhydrostatic internal seiche. Legend: nonhydrostatic (–), hydrostatic (– –).

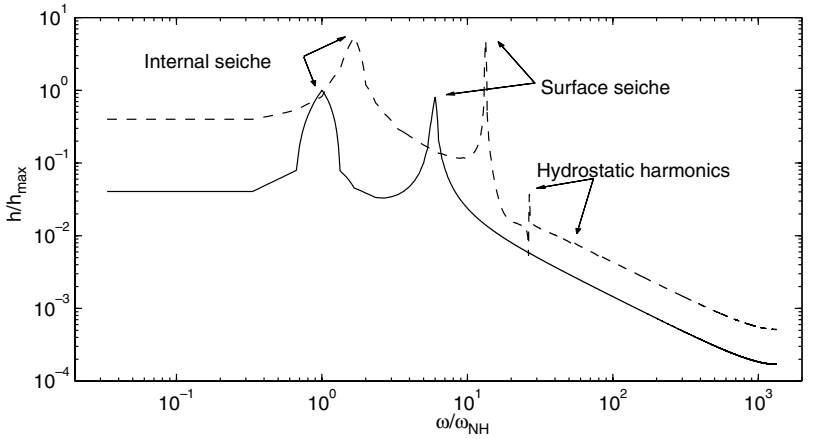

Fig. 10. Frequency spectrum of the response of the free surface to the internal seiche for the nonhydrostatic (–) and hydrostatic (– –) calculations. The first peaks correspond to the frequencies of the internal seiche, while the second peaks correspond to those of the free surface. The third and fourth peaks in the hydrostatic calculation are harmonics that do not appear in the nonhydrostatic calculation. The amplitude is normalized by the maximum nonhydrostatic free-surface response and the frequency is normalized by the nonhydrostatic internal wave frequency.

sufficient numerical scalar diffusion. This numerical diffusion can be estimated with an analysis of the one-dimensional advection equation which yields $\gamma_H = u_{max} D_j/2(1 - C_u)$ and $\gamma_V = w_{max}\Delta z/2(1 - C_w)$, where $C_u = u_{max}\Delta t/D_j$ and $C_w = w_{max}\Delta t/\Delta z$ are the horizontal and vertical Courant numbers, respectively, and u_{max} and w_{max} are estimates of the maximum horizontal and vertical velocity. For the nonhydrostatic simulation, $u_{max} = 0.024$ m s^{-1} and $w_{max} = 0.013$ m s^{-1}, while for the hydrostatic simulation, $u_{max} = 0.045$ m s^{-1} and $w_{max} = 0.210$ m s^{-1}, which yields $C_u = 0.27$ and $C_w = 0.29$ for the nonhydrostatic simulation and $C_u = 0.03$ and $C_w = 0.29$ for the hydrostatic simulation. Using these values, $\gamma_H = 1.76 \times 10^{-5}$ m^2 s^{-1} and $\gamma_V = 4.6 \times 10^{-6}$ m^2 s^{-1} for the nonhydrostatic simulation and $\gamma_H = 4.35 \times 10^{-5}$ m^2 s^{-1} and $\gamma_V = 7.41 \times 10^{-5}$ m^2 s^{-1} for the hydrostatic simulation.

Fig. 11(a) and (b) depict the density contours for the hydrostatic and nonhydrostatic simulations after $t = 10T$ s, where $T = \sqrt{D/2g'}$ and $g' = g\Delta\rho/\rho_0 = 0.01$ m s^{-2} is the reduced gravity. The hydrostatic simulation does not capture the generation of the Kelvin–Helmholtz billows and it also does not correctly capture the speed of the front. Table 1 compares the Froude numbers obtained for the present simulations to those of Hartel et al. (2000), where the Froude number is the ratio of the speed of the gravity current to the buoyancy velocity $u_b = \sqrt{g'D/2}$, and the Grashof number for these simulations is given by $Gr = (u_b D/2\nu)^2 = 1.25 \times 10^6$. The results demonstrate that the nonhydrostatic simulation captures the correct front speed, both for the no-slip and free-slip cases, while the hydrostatic simulation underpredicts both speeds. The slight discrepancy

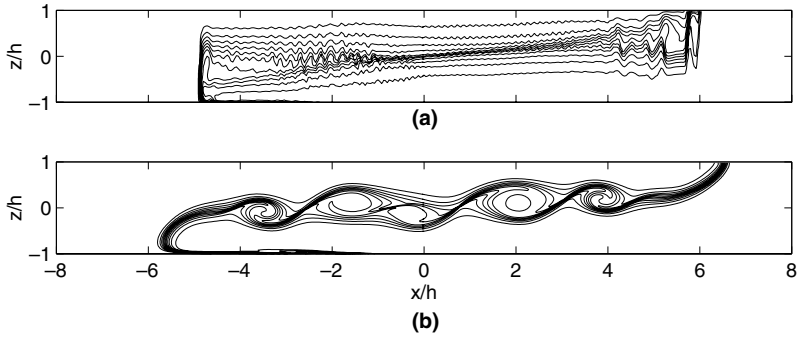

Fig. 11. Comparison of the resulting density contours for the (a) hydrostatic simulation and (b) nonhydrostatic simulation of the lock exchange test after $t = 10T$ s. Contours are plotted every $0.1\Delta\rho/\rho_0$, and $L = 0.8$ m and $h = D/2 = 0.05$ m.

Table 1
Comparison of the present Froude number calculations with those of Hartel et al. (2000)

Source	No-slip	Free-slip
Hartel et al. (2000)	0.574	0.675
Nonhydrostatic	0.562	0.654
Hydrostatic	0.470	0.605

The no-slip results represent the speed of the leftward-propagating front at the bottom while the free-slip results represent those of the rightward-propagating front at the free surface.

between the present nonhydrostatic results and the results of Hartel et al. (2000) is likely due to the use of first-order upwinding for scalar advection, which decreases the effective Schmidt number and hence decreases the front speed as a result of a thicker interface separating the layers. A small discrepancy also arises due to the asymmetry in the top–bottom boundary conditions in the present simulations, which leads to a right-propagating free-slip gravity current and a left-propagating no-slip gravity current.

It is interesting to note that, on average per time step, the nonhydrostatic simulation takes 5 times longer than the hydrostatic simulation. However, due to the hydrostatic approximation, the maximum vertical velocity for the hydrostatic simulation is 16 times larger than that for the nonhydrostatic simulation. Therefore, the hydrostatic simulation must be run with a time step 16 times smaller than the nonhydrostatic simulation to achieve the same vertical Courant number. Neglecting accuracy considerations, even though the nonhydrostatic simulation takes longer per time step, the hydrostatic simulation takes 3.2 times longer overall.

5.3. Internal waves in Monterey Bay

5.3.1. Simulation setup

As a demonstration of field-scale internal wave generation, we simulate the internal wave field in Monterey Bay by employing SUNTANS on a 100 km × 100 km domain (please see Fig. 19(a)) with a total of 72,700 cells in the planform and 100 cells in the vertical. Because SUNTANS does not store data associated with inactive cells that lie beneath the bathymetry, this yields substantial savings because only 59% of the 7.3 million cells are active. The total number of grid cells for this computation is then reduced to roughly 4.3 million. The cells were created with the Triangle package (Shewchuck, 1996), and the average Voronoi edge length is $D_j = 290$ m. The vertical grid is stretched to refine the grid near the surface, where the density gradients are larger. The minimum vertical grid spacing at the surface where index $k = N_{kc}$ can be expressed as

$$\Delta z_{N_{kc}} = D\frac{r-1}{r^{N_{kc}}-1}, \tag{94}$$

where $D = 3367.5$ m is the maximum depth, $r = 1.025$ is the algebraic stretching factor, and $N_{kc} = 100$ is the number of vertical levels, yielding $\Delta z_{N_{kc}} = 7.8$ m. The vertical grid spacings beneath the surface are then given by $\Delta z_{k-1} = r\Delta z_k$, where $k = N_{kc}-1, N_{kc}-2, \ldots, 1$, which yields a maximum vertical grid spacing of $\Delta z_1 = 89.7$ m. The depth is interpolated with data obtained from the MBARI Multibeam Survey CD-ROM (Hatcher et al., 1998).

The model is initialized with a stagnant velocity field and free surface, and the initial density profie is given by the average of data from 50 CTD casts obtained in Monterey Bay between November 1989 and December 1992 (Rosenfeld et al., 1994). This density profile and associated buoyancy period are shown in Fig. 12. At the boundaries of the domain, the velocity field is specified with the semi-diurnal (M_2) component of the barotropic tidal currents obtained from the OSU Tidal Inversion Software (Egbert and Erofeeva, 2002). Because the barotropic tides at this boundary behave linearly, it suffices to impose a gradient-free condition on the free surface. We employ a sponge layer at the western boundary in order to prevent reflections of the internal tides from that boundary. The sponge layer is implemented by adding a source term to the horizontal momentum equation of the form

$$S_U = -\frac{U}{\tau_s}\exp\left(-\frac{x}{L_s}\right), \tag{95}$$

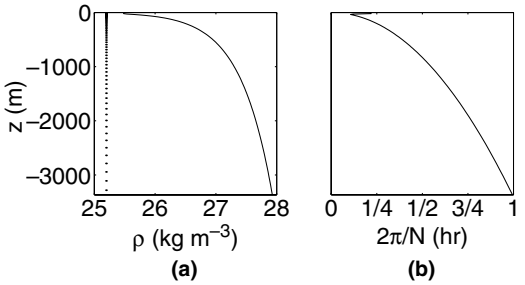

Fig. 12. (a) Density field and (b) buoyancy period $2\pi/N$ taken from Petruncio et al. (2002). The dots in subplot (a) indicate the vertical position of every other z-level.

where $\tau_s = 1000$ s and $L_s = 1$ km. The simulation is run for a total of four M_2 tides with a time step of $\Delta t = 9.315$ s, which is governed by the stability of the explicit treatment of the internal gravity waves from Eq. (85) and by desiring accuracy such that $c_i \Delta t / D_j = 0.1$, which implies an internal gravity wave Courant number of 0.1. The total simulation time is sufficient to generate semidiurnal internal tidal energy, yet it is short enough such that internal wave energy reflecting from the boundaries does not affect the results. We are currently implementing the Sommerfeld radiation condition to radiate internal wave energy from these boundaries in order to allow for longer simulation times. The horizontal and vertical eddy viscosities are constant and are chosen to stabilize the central-differencing scheme for advection of momentum, as in Eq. (83). Accordingly, the vertical eddy viscosity is constant and given by $\nu_V = 1.4 \times 10^{-3}$ m^2 s^{-1}, the horizontal eddy viscosity is $\nu_H = 1.18$ m^2 s^{-1}, and there is no physical scalar diffusivity, since the first-order upwinding scheme provides sufficient numerical diffusion. Following the estimation technique outlined in Section 5.2, the numerical diffusion coefficients associated with the first-order upwinding scheme are given by $\gamma_H = 14.5$ m^2 s^{-1} and $\gamma_V = 1.4$ m^2 s^{-1}, where we have assumed that $u_{max} = 0.1$ m s^{-1} and $w_{max} = 0.01$ m s^{-1}. The normalized residual of the conjugate gradient solver for the free-surface is 10^{-10} and that for the nonhydrostatic pressure is 10^{-5}, and we use $\theta = 0.55$ for the theta-method. Simulations take 4.48 s per time step, or twice as fast as real time, using 16 processors on the JVN cluster at the ARL Major Shared Resource Center.

5.3.2. Internal wave generation sites

In the continuously stratified ocean, internal wave energy propagates along beams at an angle θ with respect to the horizontal and slope dz/dx given by

$$\frac{dz}{dx} = \tan(\theta) = \pm\left(\frac{\omega^2 - f^2}{N^2 - \omega^2}\right)^{1/2}, \tag{96}$$

where ω is the internal wave frequency, f is the Coriolis parameter, and N is the buoyancy frequency. In the coastal ocean where the barotropic tides generate currents across isobaths, internal waves are generated in regions where the topographic slope is critical, or where θ matches the bottom slope (Prinsenberg et al., 1974).

Numerous works have documented the generation of internal waves in Monterey Bay. The field and numerical studies of Petruncio et al. (1998, 2002) and the field observations of Kunze et al. (2001) show that internal wave energy is generated beyond the shelf break in Monterey Bay and that this energy propagates towards the shore and is focused within the Monterey Submarine Canyon. The transect we use to compare the results of our model with their findings is shown as the along-canyon transect in Fig. 13. Lien and Gregg (2001) show that an internal wave beam that is generated at the shelf break just north of the along-canyon transect results in elevated dissipation and mixing. We compare the results of our model with their field results by studying the internal wave field in the along-ridge transect shown in Fig. 13.

Contours of the east–west baroclinic velocity field after 3.75 M_2 tides are shown in Fig. 14 for the along-canyon transect. From linear theory, internal waves are generated at regions where the internal wave ray path matches that of the local topography. Steep topographical ridges present effective regions of generation because the bottom slope passes through criticality in a narrow region where the vertical momentum is large. Linear theory predicts that the internal wave ray paths follow trajectories defined by the dispersion relation

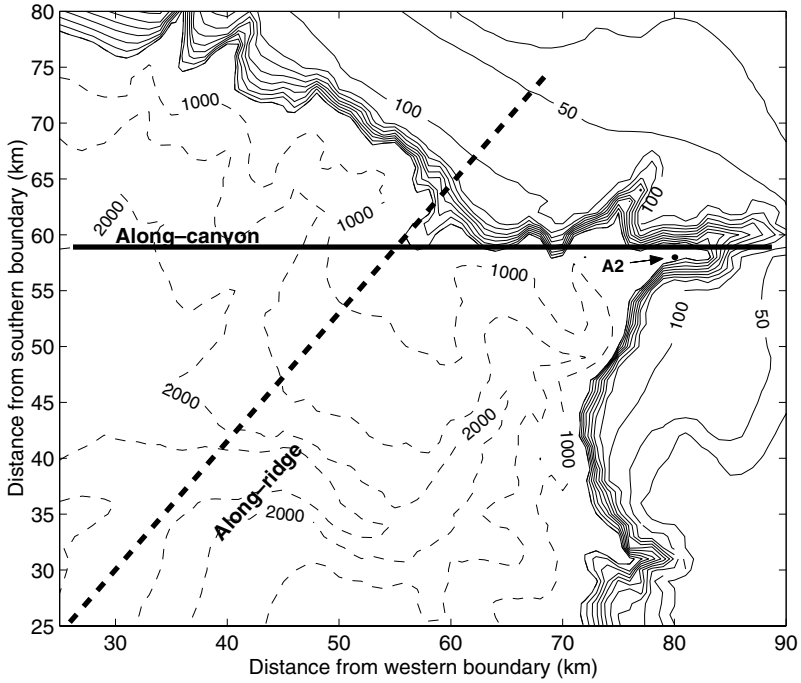

Fig. 13. Transects used to compare the results of SUNTANS to the numerical and field studies of Petruncio et al. (1998, 2002) and Kunze et al. (2001) (along-canyon transect: –) and Lien and Gregg (2001) (along-ridge transect: – –). Also shown is the location of the ITEX A2 mooring, where field data is compared to simulation results in Fig. 16.

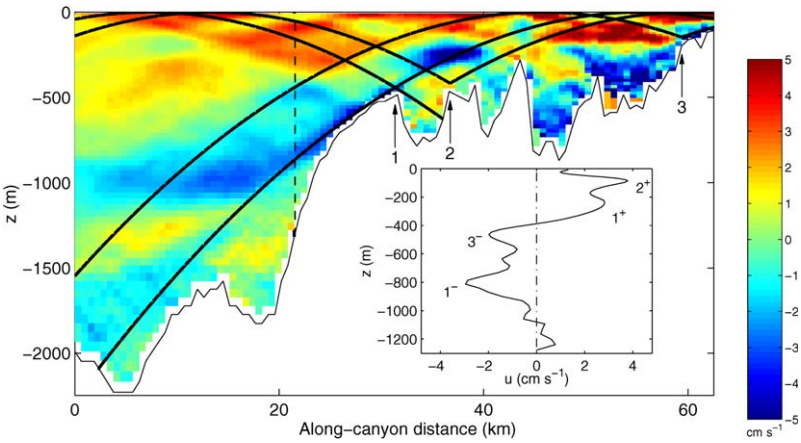

Fig. 14. East–west baroclinic velocity contours in the along-canyon transect depicted in Fig. 13 during peak ebb (simulation time of 3.75 M_2 tides). The numbers indicate possible generation sites for internal waves with the dark solid lines indicating the ray paths predicted by linear theory. The inset plot depicts the horizontal velocity profile at the vertical transect depicted by the dashed line, with the numbers indicating the likely origin of the upward- ($^+$) or downward-propagating ($^-$) internal wave beams.

(96). The internal wave ray paths are depicted by the solid lines that emit from likely generation sites at locations 1–3 in Fig. 14. As noted by Petruncio et al. (2002), most of the western side of the smooth ridge from which ray 1 is generated is critical, which is what influences the east–west velocity profile in the upper 400 m of the water column at the transect located at 20 km (the inset plot in Fig. 14). Both downward and upward propagating beams generated at location 1 affect the velocity field at the 20 km transect (denoted by the 1^- for the downward-propagating beam and 1^+ for the upward-propagating beam), as well as upward-propagat-

ing beams (denoted by the 2^+ and 3^+) emanating from ridges 2 and 3. Generation sites 2 and 3 also generate downward propagating energy but these beams are omitted for clarity.

Fig. 15 depicts the internal wave field in the along-ridge transect depicted in Fig. 13. Internal wave energy is generated at the shelf break (location 3) and this energy propagates on- and off-shore and generates the off-shore beam (detailed in the inset plot) measured by Lien and Gregg (2001) which contains a region of enhanced turbulent mixing and dissipation. They measured eddy viscosities in this region as large as 0.01 $m^2 s^{-1}$, which are likely due to turbulence generated by a shear instability that results from the internal wave field. The present simulations show that internal wave energy is also generated along the bottom between generation sites 1 and 2, where the bottom slope is almost exclusively critical. This internal wave energy is also contributing to the elevated levels of shear which generates the turbulence in the tidal beam measured by Lien and Gregg.

Using a much larger domain than that employed in the present paper in order to limit internal wave reflection from the boundaries for longer timescale simulations, a detailed analysis has been performed by Jachec et al. (2006) to show that the velocity field generated by the internal wave field is highly dependent on the grid

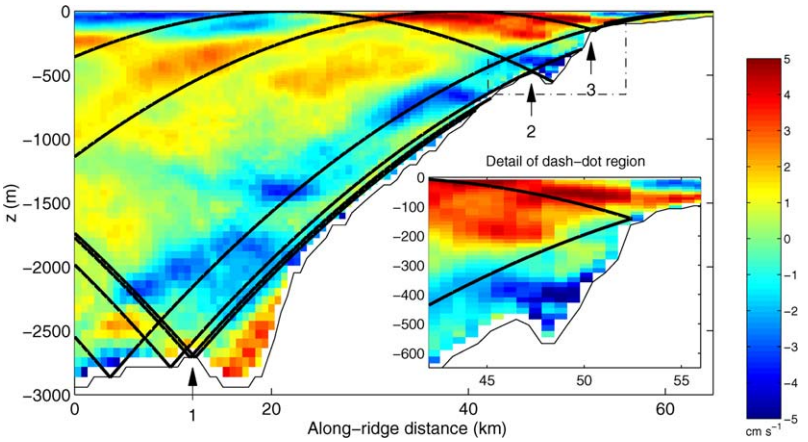

Fig. 15. Baroclinic velocity contours in the plane of the along-ridge transect depicted in Fig. 13 during peak ebb (simulation time of 3.75 M_2 tides). The numbers indicate possible generation sites for internal waves with the dark solid lines indicating the ray paths predicted by linear theory. The inset plot depicts a zoomed in view of the dash-dot region surrounding generation sites 2 and 3.

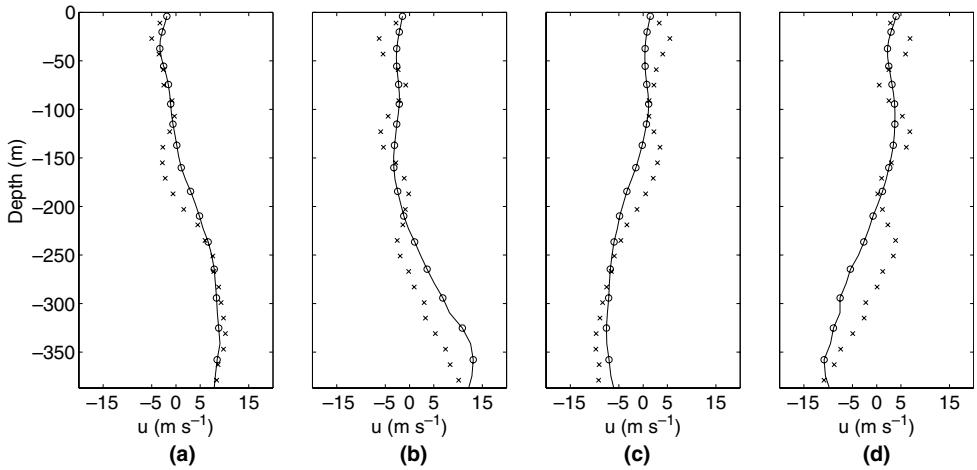

Fig. 16. Comparison of the computed horizontal east–west velocity field (– ○ –) to the measured $M2$ velocity field (– × –) at station A2 (see Fig. 13) at four different phases in the $M2$ tidal cycle. (a) $t/T_{M2} = 0$, (b) $t/T_{M2} = 0.25$, (c) $t/T_{M2} = 0.5$, (d) $t/T_{M2} = 0.75$.

resolution. They employ high-resolution simulations of the internal wave field in Monterey Bay using SUNTANS and show that the computed internal wave-induced velocity matches that in the field when using a grid with at least 300 m horizontal resolution (based on the Voronoi edge length), while coarser resolution leads to substantially lower internal wave-induced velocities. Results from those simulations are depicted in Fig. 16, and show the horizontal east–west velocity field computed at station A2 (see Fig. 13) compared to the ITEX1 (Internal Waves EXperiment) field data of Petruncio et al. (1998) from April of 1994. These simulations match the amplitude and phasing of the internal tidal wave field quite well. Details of these simulations, including a discussion of the internal wave energetics computed with high resolution, can be found in Jachec et al. (2006, submitted for publication).

The simulations of internal waves in the present work and in the work of Jachec et al. (submitted for publication) demonstrate the capability of SUNTANS to generate the internal wave field that is consistent with field measurements. Although the simulations were performed using the code in its nonhydrostatic mode, we have found that even at this grid resolution the results are quite similar to the hydrostatic results. Nonhydrostatic effects are manifested when the grid spacing is on the order of 50 m or less, since waves in Monterey Bay only start becoming nonhydrostatic when they transition to steep, highly nonlinear on-shelf waves, such as those measured by Klymak and Moum (2003) or Carter et al. (2005), which have wavelengths of a few hundred meters.

6. Code performance

6.1. Time accuracy

As shown by Armfield and Street (2000), the time accuracy of a numerical method for incompressible flows depends a great deal on the specific treatment of the pressure term in the discretization as well as the type of grid being used. While discretizing the viscous terms with the Crank–Nicolson method and assuming that all other terms are discretized with at least second-order accuracy, they show that the use of the projection algorithm on a nonstaggered grid results in first-order temporal accuracy, while the use of the correction algorithm increases the accuracy to second order. In the projection algorithm, the pressure is left out of the momentum equations and the pressure field required to enforce continuity is the full nonhydrostatic pressure field, such that the governing discrete equations can be summarized with

$$\mathbf{u}^* = \mathbf{u}^n + \Delta t \mathbf{R}^{n+1/2},$$
$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \mathbf{G} q^{n+1/2},$$

where $\mathbf{u}^*$ is the predicted velocity field, $\mathbf{R}^{n+1/2}$ is the portion of the momentum equations that contains terms other than the nonhydrostatic pressure, q is the nonhydrostatic pressure, and $\mathbf{G}$ is the discrete gradient operator. The pressure-correction algorithm, which is employed by the method in this paper, is given by

$$\mathbf{u}^* = \mathbf{u}^n + \Delta t \mathbf{R}^{n+1/2} - \Delta t \mathbf{G} q^{n-1/2},$$
$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \mathbf{G} qc,$$
$$q^{n+1/2} = q^{n-1/2} + qc,$$

where qc is the pressure correction. An alternative to these methods is to employ the theta-method of Casulli (1999), which is given by

$$\mathbf{u}^* = \mathbf{u}^n + \Delta t \mathbf{R}^{n+1/2} - (1-\theta)\Delta t \mathbf{G} q^n,$$
$$\mathbf{u}^{n+1} = \mathbf{u}^* - \theta \Delta t \mathbf{G} q^{n+1},$$

where, based on the underlying free-surface algorithm, $0.5 < \theta \leqslant 1$ for stability. In the absence of the nonhydrostatic pressure, employing the theta-method for the vertical diffusion and barotropic terms and the second-order Adams–Bashforth method for all other terms (i.e. advection, Coriolis, and gravity) results in second-order accuracy when $\theta = 0.5$. Although using $\theta = 0.5$ is only neutrally stable (Casulli, 1990), numerical tests can be performed over small enough times before instability develops in order to demonstrate second-order temporal accuracy.

Here we demonstrate second-order temporal accuracy of SUNTANS by performing numerical experiments. The internal seiche described in Section 5.1 is simulated with 100×100 equilateral prisms, using a depth of $D = 100$ m, horizontal and vertical eddy viscosities of 10^{-4} m^2 s^{-1}, and a Prandtl number of $Pr = 1$. To test the behavior of the method, the flow is integrated over a time interval of 5.0 s using six different time step sizes, ranging from $\Delta t = 2.5 \times 10^{-1}$ s to $\Delta t = 7.8125 \times 10^{-3}$ s, corresponding to $N_{\max} = 20 \times 2^n$ time steps for each integration, with $n = 0, 1, 2, \ldots, 5$. The error is expressed as the L_2 norm of the difference between these solutions and a reference, or benchmark solution, obtained with a time step size of $\Delta t = 3.90625 \times 10^{-3}$ s, corresponding to $N_{\max} = 20 \times 2^6 = 1280$ time steps. The error results are shown in Fig. 17, where the error given by

$$E^2(\Delta t) = \frac{\sum_{i,k}^{N_c,N_k} (\phi_{i,k} - \phi_{i,k}^{\text{ref}})^2}{\sum_{i,k}^{N_c,N_k} (\phi_{i,k}^{\text{ref}})^2}, \tag{97}$$

where ϕ is either h, U, w, ρ, or q, and ϕ^{ref} is the reference solution. All errors converge with second-order accuracy. However, because the nonhydrostatic pressure is staggered in time with respect to the velocity ($q^{n\pm1/2}$ about U^n), the second-order Adams–Bashforth extrapolation scheme is used to compute the pressure at the final time step (Armfield and Street, 2000). For each simulation, the final pressure field is computed with the last two offset time steps using

$$p_{N_{\max}} = \frac{3}{2} p_{N_{\max}-1/2} - \frac{1}{2} p_{N_{\max}-3/2} + \mathcal{O}(\Delta t^2). \tag{98}$$

As shown in Fig. 17, failure to perform this extrapolation results in convergence at a rate of roughly $\mathcal{O}(\Delta t^{1.5})$, while its use ensures second-order accuracy. Fig. 17 also shows how all of the other terms also converge to second-order accuracy, indicating that the present formulation is truly a second-order accurate method.

6.2. Block-Jacobi preconditioner

Eq. (79) can be written in matrix–vector form as

$$\mathbf{M}_i \mathbf{q}_i - \sum_{m=1}^{N_s} \frac{\mathrm{d}f_m}{D_m} (\mathbf{\Delta Z}_m \mathbf{e}_1^{\mathrm{T}}) \mathbf{q}_{Ne_m} = -\mathbf{S}_i^*, \tag{99}$$

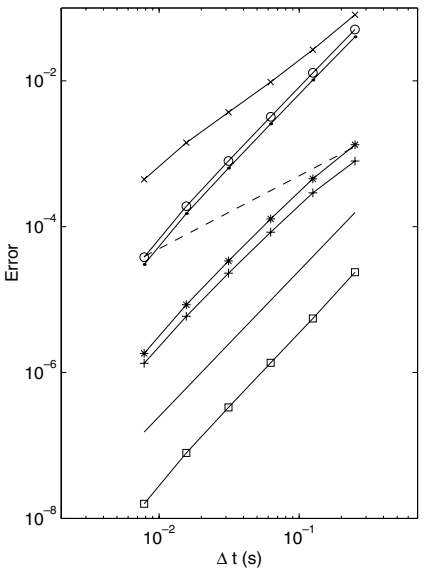

Fig. 17. Temporal convergence for the sloshing internal wave test case. Legend: (×) nonhydrostatic pressure without extrapolation, (○) with extrapolation, (+) u-velocity, (∗) w-velocity, (·) free surface, (□) density, (–) $\mathcal{O}(\Delta t^2)$, (– –) $\mathcal{O}(\Delta t)$.

where $\mathbf{M}_i$ is the symmetric tridiagonal matrix with diagonal entries $-a_{i,k}$, $a_{i,k}+a_{i,k+1}+\sum_{m=1}^{N_s}\frac{\mathrm{d}f_m}{D_m}\Delta z^{\mathrm{uw}}_{m,k}$, and $-a_{i,k+1}$, with

$$a_{i,k}=\frac{2A_i}{\Delta z_{i,k}+\Delta z_{i,k-1}}. \tag{100}$$

Eq. (99) is a symmetric, positive-definite, unstructured system of equations for the pressure correction. As noted in Marshall et al. (1997b), who implemented a preconditioner for their curvilinear grid nonhydrostatic model, this matrix is poorly conditioned for grids with small grid aspect ratios, or when $\epsilon_g=\Delta z^{\mathrm{uw}}_{m,k}/\sqrt{A_i}\ll 1$, which is the case for most environmental flows of interest. To combat this conditioning problem, a preconditioner is employed by noting that, when $\epsilon_g\ll 1$, a good approximation to the system (99) is given by a solution to

$$\overline{\mathbf{M}}_i\mathbf{q}_i=-\mathbf{S}^*_i, \tag{101}$$

where $\overline{\mathbf{M}}_i$ contains $-a_{i,k}$, $a_{i,k}+a_{i,k+1}$, and $-a_{i,k+1}$ on its diagonals. The matrices $\overline{\mathbf{M}}_i$ form the basis for a block-Jacobi preconditioner for the full system (99). Since $\overline{\mathbf{M}}_i$ is tridiagonal, the preconditioner is simple to implement because it only requires a tridiagonal inversion at each fluid column during each conjugate gradient iteration.

Applying the preconditioner (101) to the system (99) yields

$$\overline{\mathbf{M}}_i^{-1}\mathbf{M}_i\mathbf{q}_i-\sum_{m=1}^{N_s}\frac{\mathrm{d}f_m}{D_m}\overline{\mathbf{M}}_i^{-1}(\mathbf{\Delta}Z_m\mathbf{e}_1^{\mathrm{T}})\mathbf{q}_{Ne_m}=-\overline{\mathbf{M}}_i^{-1}\mathbf{S}^*_i, \tag{102}$$

which shows that, in the limit as $\epsilon_g\to 0$, $\overline{\mathbf{M}}_i^{-1}\mathbf{M}_i=\mathbf{I}$, and the solution is given by

$$\mathbf{q}_i=-\overline{\mathbf{M}}_i^{-1}\mathbf{S}^*_i. \tag{103}$$

Although the system (102) is no longer symmetric in general, the preconditioned conjugate gradient algorithm can still be used as it does not directly apply the preconditioner to the original governing equation (Demmel, 1997). The only added cost of PCG is additional storage of one temporary array and the calculation of $\overline{\mathbf{M}}_i^{-1}$, which is a straight-forward tridiagonal inversion at each water column. Furthermore, the parallel implementation of PCG does not require additional synchronization points because the inverse of $\overline{\mathbf{M}}_i$ requires data that is contiguous among processors since domain decomposition occurs only in the horizontal. Synchronization takes place twice at each iteration to compute the inner products, and interprocessor communication of boundary cells is overlapped with the calculation of $\mathscr{L}(p_k)$ on the inner cells.

The efficiency of the preconditioner is demonstrated by calculating the nonhydrostatic pressure field during the first time step of the internal seiche outlined in Section 5.1. All parameters are identical to those employed for the temporal accuracy runs in Section 6.1, except the time step is fixed at $\Delta t=0.001$ s, and, while keeping the number of vertical levels constant, the depth is varied in order to study the effects of the grid aspect ratio ϵ_g on the efficiency of the standard and preconditioned conjugate gradient algorithms. For this particular problem, because a uniform grid of equilateral triangles is employed with constant depth, then the grid aspect ratio is related to the domain aspect ratio $\epsilon=D/L$ via $\epsilon_g=2\epsilon/3^{1/4}=1.52\epsilon$. To compare the two algorithms and their sensitivity to ϵ_g, we compute the number of floating point operations required to achieve a converged solution of the discrete pressure-Poisson Eq. (79) by varying D such that $0.04\leqslant\epsilon_g<1$.

Fig. 18 depicts the number of working units (WU) required to converge to a normalized residual of 10^{-10} as a function of ϵ_g for the CG and PCG algorithms. The number of working units, or floating point operations (FLOPs), is proportional to the number of CG or PCG iterations required for convergence (N), but it is a better measure of the performance than N because of the added workload per iteration associated with the PCG algorithm. The present implementation of CG requires 20.8 FLOPs per iteration per cell while PCG requires 33.7 FLOPs per iteration per cell. For startup calculations before the iteration, CG requires 24.9 FLOPs per cell and PCG requires 35.8 FLOPs per cell. Although PCG requires 62% more operations per cell per iteration, when $\epsilon_g=0.024$ (the leftmost open circle in Fig. 18), PCG requires over 30 times fewer FLOPs to converge than CG because it converges in just 21 iterations, as opposed to 1105 for CG. As depicted in Fig. 18, without the preconditioner, the number of FLOPs to converge increases as ϵ_g decreases, reflecting the adverse

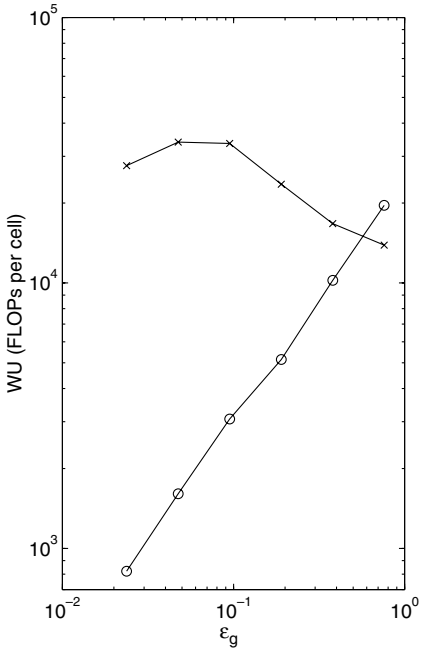

Fig. 18. Effect of the grid aspect ratio ϵ_g on the number of floating point operations required to solve the discrete nonhydrostatic pressure-Poisson equation (79) with a normalized residual of 10^{-10} using the non-preconditioned conjugate gradient algorithm (×) and the preconditioned conjugate gradient algorithm (○).

effect of ϵ_g on the conditioning of the problem. However, with the preconditioner, the number of FLOPs decreases linearly with decreasing ϵ_g, demonstrating that, in the limit as $\epsilon_g \to 0$, the solution to the problem is given by an inverse of the approximate system (103).

As expected, Fig. 18 implies that the preconditioner does not accelerate convergence for large grid aspect ratios, and hence should not be used when $\epsilon_g > \mathcal{O}(1)$. For real, field-scale applications, however, ϵ_g is rarely close to 1, even for extremely high-resolution grids. Assuming a well-designed grid, the distribution of ϵ_g throughout a given domain reflects the distribution of nonhydrostacy, since the horizontal grid size should be smaller where the flow is nonhydrostatic. The point of the preconditioner is to reduce the workload of computing the pressure-Poisson Eq. (99) associated with regions where $\epsilon_g \ll 1$, so that, naturally, its workload depends solely on the percentage of the flow that is truly nonhydrostatic. For example, suppose that α_{NH} represents the ratio of cells on a given grid where the grid aspect ratio is relatively large, so that $\epsilon_g = \epsilon_{g,\mathrm{large}}$ where the flow is mostly nonhydrostatic, while $1 - \alpha_{\mathrm{NH}}$ represents the ratio of cells on a given grid that are relatively small, where $\epsilon_g = \epsilon_{g,\mathrm{small}}$ and the flow is mostly hydrostatic. A rough estimate of the number of working units required to solve for the nonhydrostatic pressure is then given by

$$\mathrm{WU}_{\mathrm{total}} = (1 - \alpha_{\mathrm{NH}})\mathrm{WU}(\epsilon_{g,\mathrm{small}}) + \alpha_{\mathrm{NH}}\mathrm{WU}(\epsilon_{g,\mathrm{large}}). \tag{104}$$

Now suppose that 10% of a domain is nonhydrostatic, such that $\alpha_{\mathrm{NH}} = 0.1$, and $\epsilon_{g,\mathrm{large}} = 0.4$ and $\epsilon_{g,\mathrm{small}} = 0.02$. Employing Eq. (104) and using values obtained from Fig. 18, the number of working units required to solve for the nonhydrostatic pressure is reduced by 94% when PCG is used over CG. This is because use of PCG leads to a reduction in working units by 97% over most of the domain.

6.3. Parallel performance

6.3.1. Grid partitioning and load balancing

Of critical importance to the study of parallel unstructured grid simulations is the method by which the grids are partitioned among the processors. Since the grids for SUNTANS are unstructured in the planform and z-leveled in the vertical, we partition in the horizontal to ensure that water columns remain contiguous on

given processors. In addition to greatly simplifying the underlying parallel implementation, partitioning in this way allows contiguous allocation of water columns in memory which enhances performance.

When partitioning a grid among processors, it is important to balance the workload so that each processor does not do more work than any of the others involved in the simulation. While load-balancing ensures an equal workload distributed among the processors, the parallel performance of a particular unstructured grid code depends highly on the communication required between neighboring processors, and communication between processors is directly proportional to the number of cells that abut the interprocessor boundaries. Therefore, an optimal partitioning would equally distribute the cells among the processors while minimizing the number of cells that are adjacent to the interprocessor boundaries.

A load-balanced partitioning that minimizes the interprocessor communication can be obtained with the ParMETIS software package (Karypis et al., 1998) that partitions the unstructured two-dimensional planform grid. As an example, Fig. 19(a) depicts a typical unstructured grid of Monterey Bay. Using the number of active cells in each water column as the weight at each Voronoi point for the partitioning, a balanced partitioning among eight processors is depicted in Fig. 19(b). Using ParMETIS, it is guaranteed that this partitioning results in a balanced workload and minimizes the communication time by minimizing the surface area between each processor. It is worth noting that, although the planform area of the northeast subdomain depicted in Fig. 19(b) is substantially larger than that of the southwest subdomain, both subdomains have the same workload because there are fewer active cells in the vertical in the northeast subdomain because of the shallower water depth in the northeast region of Monterey Bay.

Performance can be further improved by ordering the cells such that the physical distance in memory between adjacent cells is minimized. The physical distance in main memory can be visualized with the boolean connectivity matrix. Specifically, row i of the connectivity matrix is populated by ones in the columns which correspond to neighbors of cell i, and zeros elsewhere. The connectivity matrix for a 1089-cell unstructured grid of Monterey Bay as output by the Triangle package of Shewchuck (1996) with no specific ordering is shown in Fig. 20(a). The same connectivity matrix after using the grid cell ordering routines of ParMETIS is shown in Fig. 20(b). Comparing the two shows how the reordered grid significantly reduces the average distance in main memory between neighboring cells. Typical speedups due to reordering the cells result in a 20–30% reduction of the per-processor computation time.

6.3.2. Parallel speedup

As a demonstration of the parallel performance of our code, we performed simulations of the Monterey Bay test case outlined in Section 5.3 in nonhydrostatic mode on the ARL MSRC LinuxNetworx Evelocity II cluster, which is comprised of 2048 Intel Xeon EM64T (3.6 GHz) processors interconnected with Myrinet

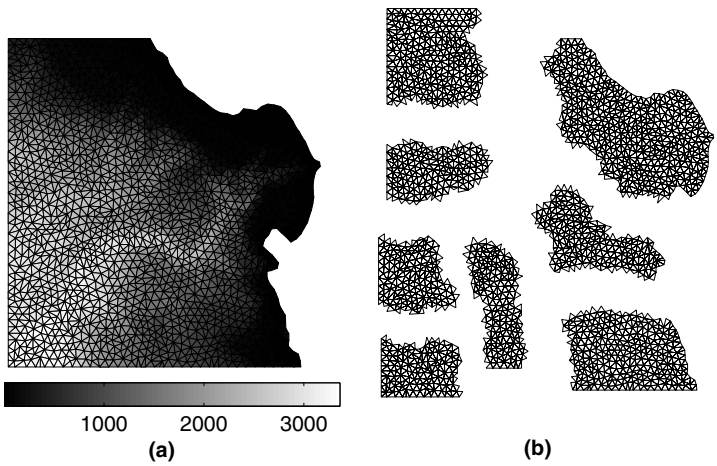

Fig. 19. (a) Typical unstructured grid of the Monterey Bay region with 3026 grid cells, showing the depth in meters. The domain is 100 km × 100 km. (b) Load-balanced partitioning using eight processors.

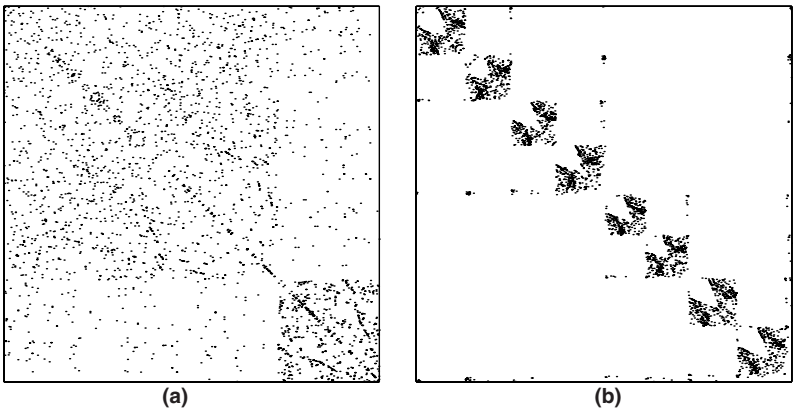

Fig. 20. Connectivity matrices for the (a) original and (b) ordered cells.

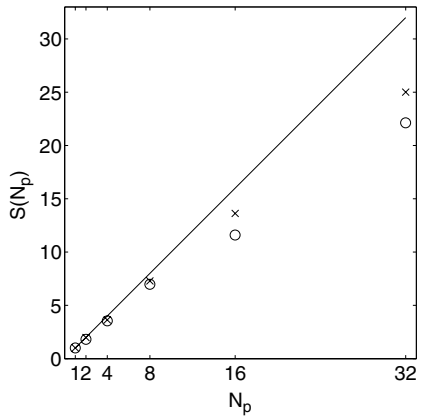

Fig. 21. Speedup of the Monterey Bay simulation as a function of number of processors with 1 million (○) and 4 million (×) cells, compared to the ideal speedup (–).

2000. Fig. 21 depicts the speedup as a function of the number of processors for the present case and a smaller problem size, which contains four times fewer grid cells. The speedup is defined by $S(N_p) = Tc_1/Tc_{N_p}$, and Tc_{N_p} is the total execution time when N_p processors are used. Consistent with Amdahl's law, the speedup deviates from the ideal speedup line as the number of processors increases because there is a portion of code that is not parallelizeable for which the execution time remains constant, independent of the number of processors. This overhead results from an increased communication time as the number of processors increases, but the relative effect of the increased communication overhead can be reduced by increasing the problem size. As a result, the speedup for the larger problem size, depicted by the × in Fig. 21, is closer to the ideal speedup line. A good measure of the parallel performance is the parallel efficiency, which is given by $E(N_p) = S(N_p)/N_p$, which is unity in the ideal case. As depicted in Table 2, the parallel efficiency is 0.69 for the smaller problem when 32 processors are used, and this efficiency increases to 0.78 when the problem size increases by a factor of four. Using 32 processors, the present simulation is 15.7 times faster than real time, while increasing the problem size by a factor of four reduces this to 3.82 (using the same time step size), indicating that the present formulation and code yields execution times which scale very well with the problem size.

7. Discussion

We have presented a parallel implementation of a three-dimensional, nonhydrostatic, unstructured-grid, finite-volume model which is highly suitable for multiscale coastal applications. The model simulates internal

Table 2
Speedup and parallel efficiency of the Monterey Bay simulation on a small and larger grid, demonstrating how the efficiency improves with increased problem size

N_p	1×10^6 cells			4×10^6 cells		
	Speedup	Efficiency	$\Delta t/\Delta t_{\text{wall}}$	Speedup	Efficiency	$\Delta t/\Delta t_{\text{wall}}$
2	1.81	0.91	1.28	1.98	0.99	0.30
4	3.55	0.89	2.52	3.63	0.91	0.56
8	6.98	0.87	4.95	7.29	0.91	1.11
16	11.60	0.73	8.23	13.63	0.85	2.08
32	22.13	0.69	15.70	25.00	0.78	3.82

The ratio of simulated time to elapsed wall-clock time is given by $\Delta t/\Delta t_{\text{wall}}$ and when this ratio exceeds unity the computation is faster than real time.

waves both from an idealized and field-scale point of view, and the lock-exchange test produces results that compare well with other published simulations. We have used the model to demonstrate essential physics resulting from the nonhydrostatic solver. In particular, nonhydrostatic effects can arise in the absence of nonlinearity, as demonstrated by the frequency-dispersive nature of the internal seiche. Nonhydrostatic effects can also be important for highly nonlinear flows, such as the nonhydrostatic lock exchange. For both linear and nonlinear flows, the fundamental effect of employing a hydrostatic model when nonhydrostatic effects are important is the overprediction of vertical deflections and frequencies (or, alternatively, vertical velocities and phase speeds), which is evident when analyzing computational results in frequency space. This was demonstrated by the internal seiche test case when the hydrostatic model was used but the dynamics were strongly nonhydrostatic because the aspect ratio was $\epsilon = D/L = 1.6$. It was also evident in the hydrostatic simulation of the lock exchange, which resulted in vertical velocities that were 20 times larger than those in the nonhydrostatic simulation. This behavior is to be expected, since frontal and interfacial dynamics in the lock exchange problem are clearly nonhydrostatic.

The numerical method employed by the model is based on the ideas of Casulli (1999), in that it employs the theta-method for the free-surface and vertical diffusion. The present formulation differs most significantly with the implementation of the pressure-correction technique and the use of the Eulerian advection of momentum strategy of Perot (2000). This allows for momentum conservation properties near the bed (as opposed to using a semi-Lagrangian technique), and hence reduces numerically induced drag, but the time step limitation is more stringent due to its explicit temporal discretization. Another important feature of the present formulation is that we have focused intently on creating a second-order temporally accurate simulation tool. As a test case to prove temporal accuracy, the internal seiche was chosen because it includes all of the terms in the momentum equations (except for the rotation terms) and poses a stringent test on the underlying nonhydrostatic solver. Convergence plots show that all of the hydrodynamic variables converge to second-order accuracy. This proves that, even though the nonhydrostatic pressure appears in the depth-averaged continuity equation, an iteration is not required in order to achieve second-order accuracy for the free surface. It also demonstrates that the pressure correction indeed satisfies $qc = \mathcal{O}(\Delta t)$, as described by Armfield and Street (2000).

The explicit discretization employed by the Eulerian advection scheme introduces a time step stability limitation that is more stringent than allowed by the semi-Lagrangian advection schemes. However, the time step sizes used in SUNTANS are limited by accuracy considerations, since we are interested in capturing short-timescale processes associated with short length-scales that can be resolved on high-resolution unstructured grids. While this comes at the expense of an increased computational cost, this is offset by the parallel implementation using MPI. Significant computational performance gains are attained with the use of the ParMETIS libraries (Karypis et al., 1998), which enable load balancing while minimizing overhead incurred by interprocessor communication. The use of the ParMETIS reordering schemes also allows for significant per-processor performance gains by minimizing the cache miss rate while accessing array data from memory. Algorithmically, per-processor performance is also optimized with the use of the block-Jacobi preconditioning scheme for the nonhydrostatic pressure solver, which is based on the ideas put forth by Marshall et al. (1997b). This preconditioner is ideal from a parallel point of view because it circumvents interprocessor communication

by only requiring the inverse of a tridiagonal system for each water column. Furthermore, it reduces the workload associated with computing the pressure-Poisson Eq. (99) in regions where the flow is hydrostatic, which is usually a large percentage of the domain for most problems of interest. As a result, convergence is greatly accelerated since computational resources focus only on the small percentage of the flow that is truly nonhydrostatic.

The present description outlines the hydrodynamic kernel of our model, yet we are working on several noteworthy additions. Our current projects include the development of a total variation dimishing (TVD) formulation (see, e.g. Gross et al., 1999; Darwish and Moukalled, 2003), which allows for the advection of sharp scalar fronts in a monotonic way, as well as the implementation of the immersed boundary technique for accurate representation of complex bathymetry (see, e.g. Tseng and Ferziger, 2003). An important next step is the implementation of adaptive mesh refinement (AMR) that allows the unstructured grid to adapt to regions requiring more resolution as the simulation progresses (see, e.g. Piggott et al., 2005). We note that in parallel implementations this could come at the possible expense of some loss in parallel efficiency unless a robust dynamic load balancing algorithm is employed. Our goal is to employ existing libraries that incorporate parallel AMR along with mesh quality control and load balancing, such as Pyramid (Lou et al., 1998).

Acknowledgements

The authors wish to thank two anonymous reviewers whose detailed comments greatly improved the quality of this manuscript, and we gratefully acknowledge the support of NSF/ITR grant 0113111 (Program manager: Barbara Fossum) and ONR grants N00014-02-1-0204 and N00014-05-1-0294 (Scientific officers: Dr. C. Linwood Vincent, Dr. Terri Paluszkiewicz, and Dr. Scott Harper). We also thank Steve Armfield, Ed Gross, Yu-Heng Tseng, and Kraig Winters for their invaluable insight on numerical methods and practical aspects of code development, and Emil Petruncio, Leslie Rosenfeld, and Jeff Paduan for providing the field data for the Monterey Bay simulations. Simulations were carried out on the JVN cluster at the ARL Major Shared Resource Center and the Baywulf cluster at the Peter A. McCuen Environmental Computing Center at Stanford University.

References

Adcroft, A., Hill, C., Marshall, J., 1997. Representation of topography by shaved cells in a height coordinate ocean model. Month. Weather Rev. 125, 2293–2315.

Armfield, S.W., Street, R.L., 2000. Fractional step methods for the Navier–Stokes equations on non-staggered grids. ANZIAM J. 42 (E), C134–C156.

Blumberg, A.F., Mellor, G.L., 1987. A description of a three-dimensional coastal ocean circulation model. In: Heaps, N. (Ed.), Three-Dimensional Coastal Ocean Models. American Geophysical Union, pp. 1–16.

Campin, J.-M., Adcroft, A., Hill, C., Marshall, J., 2004. Conservation properties in a free-surface model. Ocean Modell. 6, 221–244.

Carter, G.S., Gregg, M.C., Lien, R.-C., 2005. Internal waves, solitary waves, and mixing on the Monterey Bay shelf. Cont. Shelf Res. 25, 1499–1520.

Casulli, V., 1990. Semi-implicit finite difference methods for the two-dimensional shallow water equations. J. Comput. Phys. 86, 56–74.

Casulli, V., 1999. A semi-implicit finite difference method for non-hydrostatic, free-surface flows. Int. J. Numer. Methods Fluids 30, 425–440.

Casulli, V., Zanolli, P., 2002. Semi-implicit numerical modeling of nonhydrostatic free-surface flows for environmental problems. Math. Comput. Model. 36, 1131–1149.

Casulli, V., Zanolli, P., 2005. High resolution methods for multidimensional advection–diffusion problems in free-surface hydrodynamics. Ocean Modell. 10, 137–151.

Chen, C., Liu, H., Beardsley, R.C., 2003. An unstructured grid, finite-volume, three-dimensional, primitive equations ocean model: Application to coastal ocean and estuaries. J. Atmos. Ocean. Technol. 20, 179–186.

Cui, A., Street, R.L., 2000. Parallel computing of laboratory-scale realizations of rotating and stratified fluid flows. In: The 2000 International Conference on Parallel and Distributed Processing Techniques and Applications, pp. 1423–1429.

Darwish, M., Moukalled, F., 2003. TVD schemes for unstructured grids. Int. J. Heat Mass Transfer 46, 599–611.

Demmel, J.W., 1997. Applied Numerical Linear Algebra. SIAM.

Egbert, G.D., Erofeeva, L., 2002. OTIS—OSU Tidal Inversion Software. Available from: <http://www.coas.oregonstate.edu/research/po/research/tide/otis.html>.

Ezer, T., Mellor, G.L., 2004. A generalized coordinate ocean model and a comparison of the bottom boundary layer dynamics in terrain-following and in z-level grids. Ocean Modell. 6, 379–403.

Fletcher, C.A.J., 1997. Computational Techniques for Fluid Dynamics, vol. 1. Springer-Verlag.
Ford, R., Pain, C.C., Piggott, M.D., Goddard, A.J.H., de Oliveira, C.R.E., Umpleby, A.P., 2004. A nonhydrostatic finite-element model for three-dimensional stratified oceanic flows. Part I: Model formulation. Month. Weath. Rev. 312, 2816–2831.
Fringer, O.B., Street, R.L., 2003. The dynamics of breaking progressive interfacial waves. J. Fluid Mech. 494, 319–353.
Gross, E.S., Koseff, J.R., Monismith, S.G., 1999. Evaluation of advective schemes for estuarine salinity simulations. J. Hydraul. Eng. 125, 32–46.
Gross, E.S., Bonaventura, L., Rosatti, G., 2002. Consistency with continuity in conservative advection schemes for free-surface models. Int. J. Numer. Methods Fluids 38, 307–327.
Hartel, C., Meiburg, E., Necker, F., 2000. Analysis and direct numerical simulation of the flow at a gravity-current head. Part 1. Flow topology and front speed for slip and no-slip boundaries. J. Fluid Mech. 418, 189–212.
Hatcher, G., Maher, N., Greene, H.G., 1998. MBARI multibeam survey cd-rom. Available from: <http://www.mbari.org>.
Hodges, B.R., Street, R.L., 1999. On simulation of turbulent nonlinear free-surface flows. J. Comput. Phys. 151, 425–457.
Hodges, B.R., Imberger, J., Saggio, A., Winters, K.B., 2000. Modeling basin-scale internal waves in a stratified lake. Limno. Oceangr. 45, 1603–1620.
Jachec, S.M., Fringer, O.B., Gerritsen, M., Street, R.L., 2006. Effects of grid resolution on the simulation of internal tides. In: 16th Offshore and Polar Engng. Conf. Paper 2006-JSC-231.
Jachec, S.M., Fringer, O.B., Gerritsen, M., Street, R.L., submitted for publication. Numerical simulation of internal tides and the resulting energetics within Monterey Bay and the surrounding area. Geophys. Res. Lett.
Kanarska, Y., Maderich, V., 2003. A non-hydrostatic numerical model for calculating free-surface stratified flows. Ocean Dynam. 53, 176–185.
Karypis, G., Schloegel, K., Kumar, V., 1998. ParMETIS: Parallel graph partitioning and sparse matrix ordering library.
Klymak, J.M., Moum, J.N., 2003. Internal solitary waves of elevation advancing on a shoaling shelf. Geophys. Res. Lett. 30, 2045.
Kundu, P.K., 1990. Fluid Mechanics. Academic Press.
Kunze, E., Rosenfeld, L.K., Carter, G.S., Gregg, M.C., 2001. Internal waves in Monterey submarine canyon. J. Phys. Oceanogr. 32, 1890–1913.
Labeur, R.J., Pietrzak, J.D., 2005. A fully three dimensional unstructured grid non-hydrostatic finite element coastal model. Ocean Modell., 51–67.
Le Roux, D.Y., Lin, C.A., Staniforth, A., 1997. An accurate interpolating scheme for semi-Lagrangian advection on an unstructured mesh for ocean modeling. Tellus 49A, 119–138.
Le Roux, D.Y., Lin, C.A., Staniforth, A., 2000. A semi-implicit semi-Lagrangian finite-element shallow-water ocean model. Month. Weath. Rev. 128, 1384–1401.
Lien, R.C., Gregg, M., 2001. Observations of turbulence in a tidal beam and across a coastal ridge. J. Geophys. Res. 106, 4575–4591.
Lou, J.Z., Norton, C.D., Cwik, T., 1998. A robust parallel adaptive mesh refinement software package for unstructured meshes. In: 5th Intl. Symp. on Solving Irregularly Structured Problems in Parallel. Available from: <https://www.openchannelsoftware.com/projects/pyramid>.
Mahadevan, A., Oliger, J., Street, R., 1996a. A nonhydrostatic mesoscale ocean model. Part I: Implementation and scaling. J. Phys. Oceanogr. 26, 1860–1879.
Mahadevan, A., Oliger, J., Street, R., 1996b. A nonhydrostatic mesoscale ocean model. Part II: Numerical implementation. J. Phys. Oceanogr. 26, 1880–1900.
Marshall, J., Hill, C., Perelman, L., Adcroft, A., 1997a. Hydrostatic, quasi-hydrostatic, and nonhydrostatic ocean modeling. J. Geophys. Res. 102, 5733–5752.
Marshall, J., Adcroft, A., Hill, C., Perelman, L., Heisey, C., 1997b. A finite-volume, incompressible Navier–Stokes model for studies of the ocean on parallel computers. J. Geophys. Res. 102, 5753–5766.
Mellor, G.L., Ezer, T., Oey, L.Y., 1994. The pressure gradient conundrum of sigma coordinate ocean models. J. Atmos. Ocean. Technol. 11, 1126–1134.
Pain, C.C., Piggott, M.D., Goddard, A.J.H., Fang, F., Gorman, G.J., Marshall, D.P., Eaton, M.D., Power, P.W., de Oliveira, C.R.E., 2005. Three-dimensional unstructured mesh ocean modeling. Ocean Modell. 10, 5–33.
Perot, B., 2000. Conservation properties of unstructured staggered mesh schemes. J. Comput. Phys. 159, 58–89.
Petruncio, E.T., Rosenfeld, L.K., Paduan, J.D., 1998. Observations of the internal tide in Monterey Canyon. J. Phys. Oceanogr. 28, 1873–1903.
Petruncio, E.T., Paduan, J.D., Rosenfeld, L.K., 2002. Numerical simulations of the internal tide in a submarine canyon. Ocean Modell. 4, 221–249.
Piggott, M., Pain, C., Gorman, G., Power, P., Goddard, A., 2005. h, r, and hr adaptivity with applications in numerical ocean modelling. Ocean Modell. 10, 95–113.
Prinsenberg, S.J., Wilmot, S.L., Rattray, J.M., 1974. Generation and dissipation of coastal internal tides. Deep-Sea Res. 21, 263–281.
Roe, P.L., 1984. Generalized formulation of TVD Lax–Wendroff schemes. Tech. rep., NASA Langley Research Center, iCASE Report 84-53. NASA CR-172478.
Rosenfeld, L.K., Schramm, R.E., Paduan, J.B., Hatcher, J.G.A., Anderson, T., 1994. Hydrographic data collected in Monterey Bay during 1 September 1988 to 16 December 1992. Tech. Rep. 94-15, Monterey Bay Aquarium Research Institute, 549 pp.
Shchepetkin, A.F., McWilliams, J., 2005. The regional oceanic modeling system: a split-explicit, free-surface, topography-following-coordinate ocean modell. Ocean Modell. 9, 347–404.

Shewchuck, J.R., 1996. Triangle: a two-dimensional quality mesh generator and Delaunay triangulator, version 1.3. Available from: <http://www.cs.cmu.edu/quake/triangle.html>.

Song, Y., Haidvogel, D., 1994. A semi-implicit circulation model using a generalized topography-following coordinate system. J. Comput. Phys. 115, 228–244.

Staniforth, A., Côté, J., 1991. Semi-Lagrangian integration methods for atmospheric models: A review. Mon. Weath. Rev. 119, 2206–2223.

Thorpe, S.A., 1968. On standing internal gravity waves of finite amplitude. J. Fluid Mech. 32, 489–528.

Tseng, Y.H., Ferziger, J.H., 2003. A ghost-cell immersed boundary method for flow in complex geometry. J. Comput. Phys. 192, 593–623.

Wadzuk, M., Hodges, B.R., 2004. Hydrostatic and non-hydrostatic internal wave models. Final report to ONR. CRWR Online Report 04-09, U. Texas Austin, 77 pp. Available from: <http://www.crwr.utexas.edu/online.shtml>.

Zang, Y., Street, R.L., Koseff, J.R., 1993. A dynamic mixed subgrid-scale model and its application to turbulent recirculating flows. Phys. Fluids A 5, 3186–3196.

Zang, Y., Street, R.L., Koseff, J.R., 1994. A non-staggered grid, fractional step method for time-dependent incompressible Navier–Stokes equations in curvilinear coordinates. J. Comput. Phys. 114, 18–33.