INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **56**:1003–1015
Published online 5 December 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1663

A new computational framework for multi-scale ocean modelling based on adapting unstructured meshes‡

M. D. Piggott*, †, G. J. Gorman, C. C. Pain, P. A. Allison, A. S. Candy, B. T. Martin and M. R. Wells

Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College London, South Kensington Campus, London SW7 2AZ, U.K.

SUMMARY

A new modelling framework is presented for application to a range of three-dimensional (3D) multi-scale oceanographic problems. The approach is based upon a finite element discretization on an unstructured tetrahedral mesh which is optimized to represent highly complex geometries. Throughout a simulation the mesh is dynamically adapted in 3D to optimize the representation of evolving solution structures. The adaptive algorithm makes use of anisotropic measures of solution complexity and a load-balanced parallel mesh optimization algorithm to vary resolution and allow long, thin elements to align with features such as boundary layers. The modelling framework presented is quite different from the majority of ocean models in use today, which are typically based on static-structured grids. Finite element (and volume) methods on unstructured meshes are, however, gaining popularity in the oceanographic community. The model presented here is novel in its use of unstructured meshes and anisotropic adaptivity in 3D, its ability to represent a range of coupled multi-scale solution structures and to simulate non-hydrostatic dynamics.

Received 30 April 2007; Revised 6 October 2007; Accepted 8 October 2007

KEY WORDS: ocean modelling; multi-scale; finite element; unstructured mesh; adaptivity; anisotropic

1. INTRODUCTION

New numerical methods based on unstructured meshes, and used in conjunction with anisotropic mesh optimization methods, have the potential to revolutionize ocean modelling. Modelling the oceans accurately demands methods that can represent highly complex domains, capture multi-scale and anisotropic dynamics and replicate the strong coupling between the wide range of scales

*Correspondence to: M. D. Piggott, Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College London, South Kensington Campus, London SW7 2AZ, U.K.
†E-mail: m.d.piggott@imperial.ac.uk, URL: http://amcg.ese.imperial.ac.uk
‡Animations of presented results are available at http://amcg.ese.ic.ac.uk/~piggott/public/icfd07/.

Contract/grant sponsor: NERC; contract/grant numbers: NE/C52101X/ and NE/C51829X/1

that are present. Small-scale processes, whose effects often cannot be satisfactorily captured using subgrid-scale modelling, feed strongly into the large-scale dynamics and *vice versa*. Therefore, this application area demands the most advanced parallel numerical methods, adaptive algorithms and computational platforms.

Here a new methodology is presented, for which the goal is an ocean model that seeks to automatically maximize computational efficiency through the use of robust finite element discretizations on unstructured meshes in three dimensions (3D) in conjunction with parallel anisotropic mesh adaptivity algorithms. For background material and reviews of the literature, including discussions of other models and numerical approaches different to those presented here, see [1–4] and the references therein.

In the following sections the underlying equations and discretization are presented, as are methods for generating high-quality discrete domains. A brief background on the adaptive procedures used here is given for completeness. Issues associated with the standard discretization for problems where buoyancy and rotational effects are dominant are discussed and possible solutions given. Some preliminary examples in 2D and 3D are also presented.

2. BACKGROUND TO THE NUMERICAL MODEL

2.1. Underlying equations

The starting point for this work is the 3D non-hydrostatic Boussinesq equations in a domain $V \subset \mathbb{R}^3$:

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

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

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

$$\rho \equiv \rho(T) \tag{1d}$$

where $\mathbf{u}$ is the 3D velocity vector, $\hat{\mathbf{u}}$ is used to account for a moving reference frame, e.g. to account for a moving mesh representing a free surface whose height is given by η. t represents time, p is the perturbation pressure, g is the acceleration due to gravity and ρ is the perturbation density. The simplifying assumption that buoyancy acts in the $\mathbf{k} = (0, 0, 1)^{\mathrm{T}}$ direction has been made, this assumption can easily be generalized. When simulating large-scale domains, such as the entire globe, Cartesian (rather than spherical) coordinates are preserved and then buoyancy takes the form of a vector field acting in the direction pointing towards the globe's centre of mass. T is the temperature, but additional scalar fields such as salinity and tracers can be dealt with analogously, in which case they may be included in the equation of state (1d). The deviatoric stress tensor is used to model viscous effects and is given by $\boldsymbol{\tau}_{ij} = 2\boldsymbol{\mu}_{ij}(S_{ij} - \delta_{ij}\nabla \cdot \mathbf{u}/3)$, where $\boldsymbol{\mu}$ is the tensor of viscosities (the diagonal of which is user defined, with off diagonals defined *via* $\boldsymbol{\mu}_{ij} = \sqrt{\boldsymbol{\mu}_{ii}\boldsymbol{\mu}_{jj}}$), δ_{ij} is the Kronecker delta and $S_{ij} = \partial u_i/\partial x_j + \partial u_j/\partial x_i$ is the strain rate tensor. $\boldsymbol{\kappa}_T$ is the thermal (or saline for salinity, etc.) diffusivity tensor, assumed here to be diagonal. The rotation vector is $\boldsymbol{\Omega} = (0, 0, \Omega_z)^{\mathrm{T}}$, where $\Omega_z = 7.292 \times 10^{-5}\,\mathrm{rad\,s^{-1}}$ for calculations on the globe. For simulations

DOI: 10.1002/fld

in flat geometries the f-plane approximation can be used for relatively small domains where $2\boldsymbol{\Omega}=(0,0,\Omega_z\sin(\psi_0))^{\mathrm{T}}$ and ψ_0 is the assumed latitude for the domain; for larger 'flat' domains the β-plane approximation is often used, where $2\boldsymbol{\Omega}=(0,0,\Omega_z\sin(\psi_0)+\beta y)^{\mathrm{T}}$, $\beta=2\Omega_z\cos(\psi_0)/R_{\mathrm{E}}$ and R_{E} is the radius of the globe. Finally, $\mathbf{F}$ contains additional source terms such as astronomical tidal forcing.

2.2. A numerical discretization based on standard finite elements

The domain V is assumed to be partitioned with an unstructured mesh of tetrahedra. Let N denote the number of nodes in the mesh, which will be time dependent. For simplicity, continuous piecewise-linear finite element basis functions $\{\phi_i\}$ are assumed here. Numerical approximations to the velocity, pressure, free surface and temperature (and other scalars) are represented as

$$\mathbf{u}\equiv(u,v,w)^{\mathrm{T}}\approx\sum_{j=1}^{N}(\mathbf{u}_j(t),\mathbf{u}_{N+j}(t),\mathbf{u}_{2N+j}(t))^{\mathrm{T}}\phi_j(\mathbf{x}),\quad p\approx\sum_{j=1}^{N}\mathbf{p}_j\phi_j(\mathbf{x})$$

$$\eta\approx\sum_{j=1}^{N}\boldsymbol{\eta}_j(t)\phi_j(\mathbf{x}),\quad T\approx\sum_{j=1}^{N}\mathbf{T}_j(t)\phi_j(\mathbf{x})$$

This form for $\mathbf{u}$, p, η and T is substituted into (1). Galerkin projection in space and theta differencing in time ($\theta=\frac{1}{2}$, or Crank–Nicolson is employed here to ensure second-order temporal accuracy) results in the discrete momentum equation which in acceleration form is

$$(M_L+\theta\Delta t A)\left(\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t}\right)=-A\mathbf{u}^n+C\mathbf{p}^{n+1/2}+gC_\eta\boldsymbol{\eta}^{n+1/2}-G\boldsymbol{\rho}'^{n+1/2}+\mathbf{f}^{n+1/2}\tag{2}$$

the discrete continuity equation

$$C^{\mathrm{T}}\mathbf{u}^{n+1}=\mathbf{0}\tag{3}$$

and the discrete temperature (and other scalars) equation

$$(M'_L+\theta\Delta t H)\mathbf{T}^{n+1}=(M'_L-(1-\theta)\Delta t H)\mathbf{T}^n$$

For simplicity, homogeneous Neumann boundary conditions $\boldsymbol{\tau}\cdot\mathbf{n}-p\mathbf{n}=\mathbf{0}$, $\mathbf{n}\cdot(\boldsymbol{\kappa}\nabla T)=0$ (i.e. free stress and zero heat flux) have been assumed. Superscripts refer to the time level, Δt is the time step size and $A\equiv A(\tilde{\mathbf{u}})=N(\tilde{\mathbf{u}})+L-D$, which is defined in more detail below. M'_L and M_L are the diagonal lumped mass matrices obtained by row summing M' and M given by

$$M\equiv\mathrm{diag}(M',M',M')\quad\text{where } M'_{ij}=\int_V\phi_i(\mathbf{x})\phi_j(\mathbf{x})\,\mathrm{d}V\quad\text{for } 1\leqslant i,j\leqslant N$$

The Coriolis and buoyancy terms are represented *via*

$$L\equiv 2\Omega_z\begin{pmatrix}0&-M'&0\\M'&0&0\\0&0&0\end{pmatrix},\quad G\equiv g\begin{pmatrix}0&0&0\\0&0&0\\0&0&M'\end{pmatrix},\quad \boldsymbol{\rho}'\equiv\begin{pmatrix}0\\0\\\boldsymbol{\rho}\end{pmatrix}$$

DOI: 10.1002/fld

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

$$N(\tilde{\mathbf{u}}) \equiv \mathrm{diag}(N', N', N') \quad \text{where } N'_{ij}(\tilde{\mathbf{u}}) = \int_V \phi_i(\mathbf{x})\tilde{\mathbf{u}} \cdot \nabla \phi_j(\mathbf{x})\, \mathrm{d}V \quad \text{for } 1 \leqslant i, j \leqslant N$$

which is linearized using $\tilde{\mathbf{u}} = \mathbf{u}^n$ during the first Picard iteration used to solve the nonlinear system and $\tilde{\mathbf{u}} = \theta \mathbf{u}^{n+1} + (1-\theta)\mathbf{u}^n$ in subsequent iterations. In the results to be presented two iterations have been used as this is sufficient to preserve second-order temporal accuracy. The matrix D represents a discretization of the viscous terms that are considered in stress form in this framework and are hence too lengthy to detail here. It does, however, follow all the above conventions, for further background see [5]. The vector $\mathbf{f}$ contains any additional external forcing. The matrix $-C$ represents a discretization of the gradient operator where

$$C \equiv (C^1, C^2, C^3)^{\mathrm{T}}, \quad C^q_{ij} = \int_V \phi_j(\mathbf{x}) \frac{\partial \phi_i(\mathbf{x})}{\partial x_q}\, \mathrm{d}V, \quad 1 \leqslant i, j \leqslant N, \;\; q = 1, 2, 3$$

C_η is defined similarly and C^{T} is a representation of the discrete divergence operator. Finally the matrix H is defined by

$$H_{ij} = \int_V \left(\phi_i \tilde{\mathbf{u}}^n \cdot \nabla \phi_j + \boldsymbol{\kappa}_{lm} \frac{\partial \phi_j}{\partial x_m} \frac{\partial \phi_i}{\partial x_l} \right) \mathrm{d}V \quad \text{for } 1 \leqslant i, j \leqslant N$$

Although a Galerkin (i.e. non-stabilized, non-dissipative) method is used for momentum here, a flux-limited control volume finite element extension to the above is used for tracer fields in the results to be presented in order to guarantee boundedness and stability. Details of the discretization method for advancing $\boldsymbol{\eta}$ are not reported here due to space limitations; it will, however, be the subject of future publications. Further details may be found in [6–8].

2.3. A projection-based solution procedure for velocity and pressure

For computational efficiency a projection method is used to decouple the solution procedure for velocity and pressure. The method proceeds by first solving (2) for a velocity prediction $\mathbf{u}_*^{n+1}$ using the old pressure $\mathbf{p}^{n-1/2}$ in place of $\mathbf{p}^{n+1/2}$. In general, $\mathbf{u}_*^{n+1}$ will not satisfy the discrete continuity equation (3) and, hence, velocity and pressure are modified to ensure a divergence-free velocity field at the end of every time step. A discrete irrotational correction of the form

$$M_L \mathbf{u}^{n+1} = M_L \mathbf{u}_*^{n+1} + \Delta t C \Delta \mathbf{p} \tag{4}$$

is sought to ensure $C^{\mathrm{T}} \mathbf{u}^{n+1} = \mathbf{0}$. Multiplying (4) by $C^{\mathrm{T}} M_L^{-1}$, and applying (3), yields the following discretized elliptic problem for $\Delta \mathbf{p}$:

$$C^{\mathrm{T}} M_L^{-1} C \Delta \mathbf{p} = -\frac{C^{\mathrm{T}} \mathbf{u}_*^{n+1}}{\Delta t}$$

At this point a pressure filter is included to remove singularities in this discretization method, whereby a matrix K representing a discretization of a fourth-order operator is applied. There are several options for stabilizing equal-order element pairs such as the piecewise-linear elements

DOI: 10.1002/fld

used here. For a discussion, including a second-order version of the approach taken here, see [5] (p. 535). The matrix K takes the form $K = Q^{\mathrm{T}} M_L^{-1} Q$, where the matrix Q is given by

$$Q_{ij} = \frac{1}{2} \int_V (\nabla \phi_i)^{\mathrm{T}} \overline{\overline{h}} \nabla \phi_j \, \mathrm{d}V$$

and $\overline{\overline{h}}$ is a tensor representing the anisotropic local size of elements. The discrete equation for the pressure update then becomes

$$(C^{\mathrm{T}} M_L^{-1} C + K) \Delta \mathbf{p} = -\frac{C^{\mathrm{T}} \mathbf{u}_*^{n+1}}{\Delta t} - K \mathbf{p}_*^{n+1/2} \tag{5}$$

where $\mathbf{p}_*^{n+1/2}$ is the pressure at the new time level from the previous nonlinear iteration. This is equivalent to replacing the discrete continuity equation with $C^{\mathrm{T}} \mathbf{u}^{n+1} + \Delta t K \mathbf{p}^{n+1/2} = \mathbf{0}$. This stabilization approach is attractive since it does not violate global mass conservation. Using the solution $\Delta \mathbf{p}$ from (5), $\mathbf{u}^{n+1}$ is obtained from (4), and pressure updated *via* $\mathbf{p}^{n+1/2} = \mathbf{p}_*^{n+1/2} + \Delta \mathbf{p}$. During the first nonlinear iteration $\mathbf{p}_*^{n+1/2} = \mathbf{p}^{n-1/2}$.

3. ANISOTROPIC MESH ADAPTIVITY IN PARALLEL

3.1. Estimates of discretization error

Once the discretization underlying the modelling framework is in place a robust method to optimize the computational mesh is required. Mesh optimization is used dynamically to best represent evolving solution structures and reduce discretization errors. The starting point is a solution-dependent error measure that varies in space and time and yields the desired mesh resolution. The approach is motivated by the need to identify anisotropy in the solution structures; hence, these can be built into the eventual adapted mesh. Anisotropy is crucial in this application area since structures such as boundary layers are prevalent and exert enormous influence on both large and small-scale dynamics, e.g. the Gulf Stream.

This is achieved here using interpolation error-based measures that are cheap to compute, yield directional information and give a good guide to problem complexity. The approach boils down to the construction of a metric tensor field that is used to measure element edge lengths, and returns unit lengths for elements that are of the appropriate shape and size in metric space. The metric is built upon the Hessian matrix field for each solution component and user-defined definitions of (spatially varying) desired errors and incorporates maximum and minimum allowed edge lengths. See [9] for a more thorough description of the algorithm.

Since the form of this error measure includes a user-defined weight and a measure of anisotropy, the anisotropy can be preserved while building extra isotropic but inhomogeneous information into the weight. For example, in addition to including regional preference, this is a simple means to combine anisotropic *a priori* and isotropic *a posteriori* information, e.g. [10].

3.2. Mesh optimization

Following the construction of the metric, a functional is formed whose minimization ensures a mesh whose elements are appropriately shaped and sized, that is, they are uniform equilateral

DOI: 10.1002/fld

elements in metric space. An optimization procedure that visits elements in the mesh and performs topological operations on local connectivity then drives down this functional, and hence adapts the mesh. The operations permitted on the mesh are *edge split*—a node is inserted at the centre of an edge and surrounding elements created; *edge collapse*—all elements belonging to the edge are deleted and the two nodes of the edge collapsed to its centre, in cases where one of the nodes is used to define the domain geometry or internal structures the nodes are collapsed to that point, if both nodes define some geometrical structure then this operation is not permitted; *edge–face swap*—if two tetrahedra share a common face, and provided their combined interior is convex, the face is deleted and a new edge introduced between the two nodes not shared thus producing three tetrahedra with different alignments, the inverse operation where an edge is replaced by a face is also allowed; *edge–edge swap*—elements that lie around an edge are replaced by a new edge resulting in elements with different alignments; *node movement*—the local topology of the mesh is preserved, but mesh quality is improved by visiting each node and applying a hill climbing algorithm to locate the node in the position that minimizes the maximum functional value of the surrounding elements.

3.3. A load-balanced model in parallel with changing resolution and computational requirements

In order to parallelize the model, dynamic load balancing needs to be given primary importance. Large load imbalances are to be expected following mesh optimization which will be highly detrimental to parallel computational efficiency. Thus, the mesh needs to be repartitioned among the processors involved in the computation, here this is achieved using the graph partitioner ParMetis.[§] The number of nodes assigned to a partition/processor is assumed to be a good proxy for the expected computational load on that processor and, hence, one aim of the algorithm is to equidistribute nodes. Other requirements include a minimization of edge cut and data migration. The parallel algorithm first applies the serial mesh optimization method within each sub-domain independently while disallowing any modification to be made to the sub-domain interfaces. Following this, the graph partitioner produces a new decomposition of the mesh which re-balances the number of nodes. In addition to this, weights are applied to the graph edges such that edges associated with poor elements have high weights and are therefore less likely to be split by a partition. This choice of edge weight has the effect of perturbing partitions away from elements that require optimization. This may include elements that were previously fixed due to their location on sub-domain interfaces. Diffusion repartitioning is used by default to minimize data migration, scratch remapping is also used periodically to remove any islands that can occur with diffusion alone. For further details see [11].

3.4. Example 1: The backward-facing step computed with mesh adaptivity on 16 processors

The backward-facing step is a classical computational fluid dynamics example and one of the most frequently selected problems for simulating the separation and reattachment of turbulent flows. It is also often used as a test problem for validating numerical codes. In the context of ocean modelling flow separation is important in large-scale western boundary currents such as the Gulf Stream. Accurate experimental results for a wide range of flow regimes exist (e.g. see [12]) and the problem has several important flow characteristics, such as the reattachment length (which

[§] ParMetis: http://glaros.dtc.umn.edu/gkhome/views/metis.

DOI: 10.1002/fld

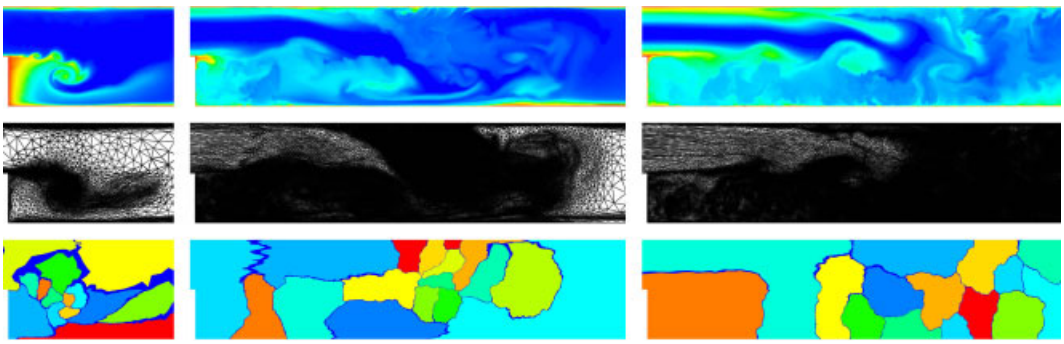

Figure 1. Simulation of flow past a backward-facing step, the domain is $55 \times 0.1 \times 2$ units in size—only subsets of which are shown. The step is of unit height and located 5 units from the left-hand boundary where a parabolic inflow of unit magnitude is applied. A passive tracer field, set to unity on upper and lower boundaries, is shown in the upper panels to aid visualization of the mixing processes. The adapted mesh on one lateral boundary (the problem is set up to be quasi-2D) is shown in the middle panels and the adapting domain decomposition shown in the lower panels where the extended dark blue region is the halo. A 'do-nothing' outflow condition is applied at the right-hand boundary, with no slip conditions on the upper and lower boundaries and free slip on lateral boundaries. The time step is 0.02 and results are shown at times 5, 25 and 45 where there are 39 726, 585 717 and 727 974 nodes in the mesh, respectively. Viscosity is set to 10^{-4}, and the interpolation error weights are set to 0.005 for u and w and are effectively infinite for v and the tracer field.

occurs at approximately 8 step heights length downstream of the step at high Reynolds numbers). See Figure 1 for an example solution where the evolution of a passive tracer, the adapted mesh and the load-balanced domain decomposition are displayed. The results are consistent with the expected reattachment length.

4. OPTIMAL DOMAIN REPRESENTATION AND MESH GENERATION

Since it exerts a critical influence over the dynamics of the ocean, a high-quality discrete representation of the physical domain in question is vital. In practice, there is a trade-off between how close the discretized domain is to reality, and how appropriate it is for numerical modelling with finite computational resources. Hence, the constructed computational mesh representing the geometry should be optimized to attain a specified error when compared with 'reality' while using a minimum number of nodes. To achieve this, methods similar to those presented in Section 3 are employed.

The first stage in the construction of a computational domain is an approximation of the shoreline for the region of interest. The simplest option is to take and optimize the zero depth contour from the bathymetry data set. Here a distance to edge criteria are used where the shoreline data at highest resolution are iteratively simplified/coarsened so as to remain within a user-defined distance to edge criterion. Figure 2 shows the result of this operation starting from the zero depth 2 min GEBCO data set and varying the distance to edge tolerance. Figure 3 then shows the resulting unstructured mesh which again starts from the highest resolution raw data and coarsens to achieve a user-defined interpolation error everywhere. Anisotropic estimates of the error are used which results in anisotropic meshes conforming to ridges, shelf breaks, seamounts, etc. Two approaches

DOI: 10.1002/fld

Figure 2. Representation of the shoreline in a section of the Mediterranean Sea. The figures show the results with distance to edge tolerances of 40, 10 and 2.5 km. The entire region includes all shoreline in the longitude–latitude box $[-10, 30] \times [28, 48]$. The number of points representing the shoreline is, respectively, 155, 670 and 3007. The raw data are the zero contours from the GEBCO bathymetric data set.

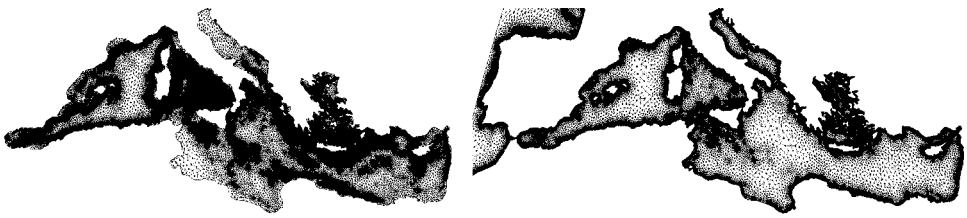

Figure 3. Representation of the bathymetry in the Mediterranean Sea, optimized from GEBCO bathymetric data subsampled initially to 2 min. The image on the left shows the result using a bathymetry interpolation error tolerance of 200 m and comprises 59 748 nodes and 172 041 elements. The image on the right shows the result using a relative bathymetry interpolation error tolerance of 3.5 and comprises 70 638 nodes and 197 679 elements, but this includes many outside of the Mediterranean in the longitude–latitude box $[-10, 30] \times [28, 48]$, not all of which is displayed. The improved representation of shallows can be seen in the mesh based on the relative error; in particular, the Straits of Gibraltar have not been artificially closed off. Also note the improved representation of other shallow region such as the Adriatic and Aegean Seas and the Gulf of Gabes.

are compared where the user-defined interpolation error takes (1) a constant absolute value and (2) a relative (to the depth of the ocean) interpolation error. For further details and examples see [13, 14].

5. AN EXTENSION TO THE BASIC DISCRETIZATION FOR BALANCE IN LARGE-SCALE GEOPHYSICAL PROBLEMS

An interesting problem occurs with standard velocity–pressure finite element pairs for large scale geophysical problems (i.e. problems where strong stratification and rapid rotation are present). This relates to the magnitude of the Coriolis and buoyancy terms in comparison with the other terms in the momentum equations. The dominant Coriolis and buoyancy terms are largely balanced

DOI: 10.1002/fld

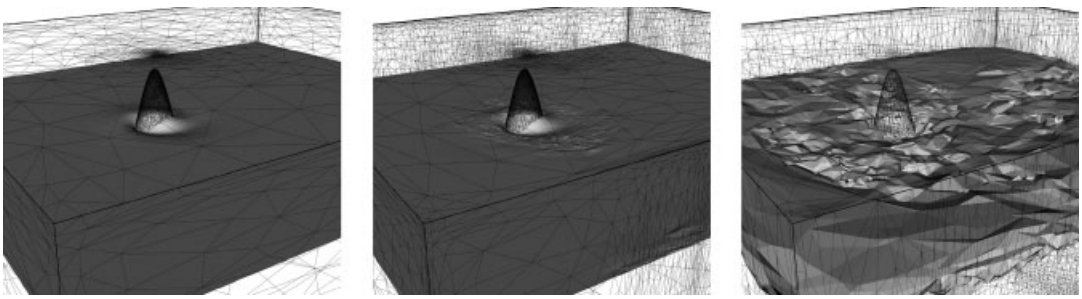

Figure 4. Example 2: Density isosurface and mesh on the surface of the domain after 60 h in stratified flow past an isolated Gaussian seamount. The image on the left shows the results when the 'balanced' pressure solver is employed to treat both Coriolis and buoyancy (31 669 nodes); the middle image show the results when only the hydrostatic pressure solver was employed (65 658 nodes); the image on the right shows the result when no special treatment of either Coriolis or buoyancy forces is used (79 915 nodes). See also [2].

(in states of hydrostatic and geostrophic balance this relationship is exact) by the gradient of a pressure, i.e.

$$2\boldsymbol{\Omega}\times\mathbf{u}+\rho\mathbf{g}\approx\nabla p_{\mathrm{hg}} \tag{6}$$

where p_{hg} is a component of the full dynamic pressure. With any standard finite element discretization a problem is immediately evident. Velocity and density are typically represented with piecewise polynomials of the same order (here linear) and pressure to the same or often lower order (here also linear). Equation (6) can therefore be subject to numerical errors that can result in large spurious accelerations. Within the modelling framework described here, two techniques have been successful in reducing this spurious behaviour to levels where meaningful simulations may be conducted.

The first approach, which deals with buoyancy only, splits full pressure into a 'hydrostatic pressure' p_{h} and a 'non-hydrostatic pressure' p_{nh}. p_{h} is assumed to balance the buoyancy force in the momentum equations: $\partial p_{\mathrm{h}}/\partial z=-\rho g$. The problematic buoyancy term and vertical derivative of p_{h} can then be dropped from the vertical momentum equation but the horizontal derivatives of p_{h} must be included in the horizontal momentum equations. At first sight nothing has been achieved, but through this decomposition of pressure p_{h} can be calculated more accurately (e.g. with higher-order discontinuous elements), while p_{nh} which still plays the role of ensuring a divergence-free velocity field can be calculated using a standard and (e.g. LBB) stable finite element pair. The results presented in Section 6.3 used this method as does the central image in Figure 4.

An alternative but related method can be derived by first decomposing the problematic forcing terms into divergence and curl-free parts:

$$2\Omega\times\mathbf{u}+\rho\mathbf{g}=\nabla\times\mathbf{A}-\nabla p_{\mathrm{hg}} \tag{7}$$

where p_{hg} is the 'hydrostatic–geostrophic' component of the full pressure. In states close to balance $\nabla\times\mathbf{A}$ is a small residual term, the difference of two large terms, which for standard finite element pairs could exhibit large spurious errors for the reasons outlined above. As before, an improved method results from calculating $\nabla\times\mathbf{A}$ more accurately and then substituting this into the momentum equations instead of the problematic Coriolis and buoyancy terms. There are

DOI: 10.1002/fld

two obvious ways to proceed, either by taking the curl or divergence of (7) and both have been pursued. The method that takes the divergence of (7) ultimately solves an elliptic equation for p_{hg}. Following the same philosophy as above, p_{hg} is calculated using piecewise polynomials of higher order than those used to represent velocity and density, here quadratics are used. The new source term $\nabla \times \mathbf{A}$ is then recovered from (7) by taking the difference of two known terms. The first image in Figure 4 uses this approach; it can be seen to be the most stable and hence it uses fewer nodes since the adaptive algorithm does not waste resources trying to resolve spurious noise.

6. FURTHER EXAMPLES

6.1. Example 3: differentially heated rotating annulus

Sloping convection in a differentially heated rotating annulus represents an ideal test problem with which to help validate a new ocean model. It encompasses non-hydrostatic dynamics and the effects of rotation, buoyancy, external forcing and closed non-trivial geometry. The problem set-up and parameters match [15] (their Figure 4 with rotation rate $\Omega_z = 3.91\,\mathrm{rad\,s^{-1}}$). Figure 5 shows results at two time levels. The first at the time when the maximum number of nodes is used, which corresponds to the maximum of the two norms of the velocity field; the second shows a later time when the system has settled into the expected mode five structure.

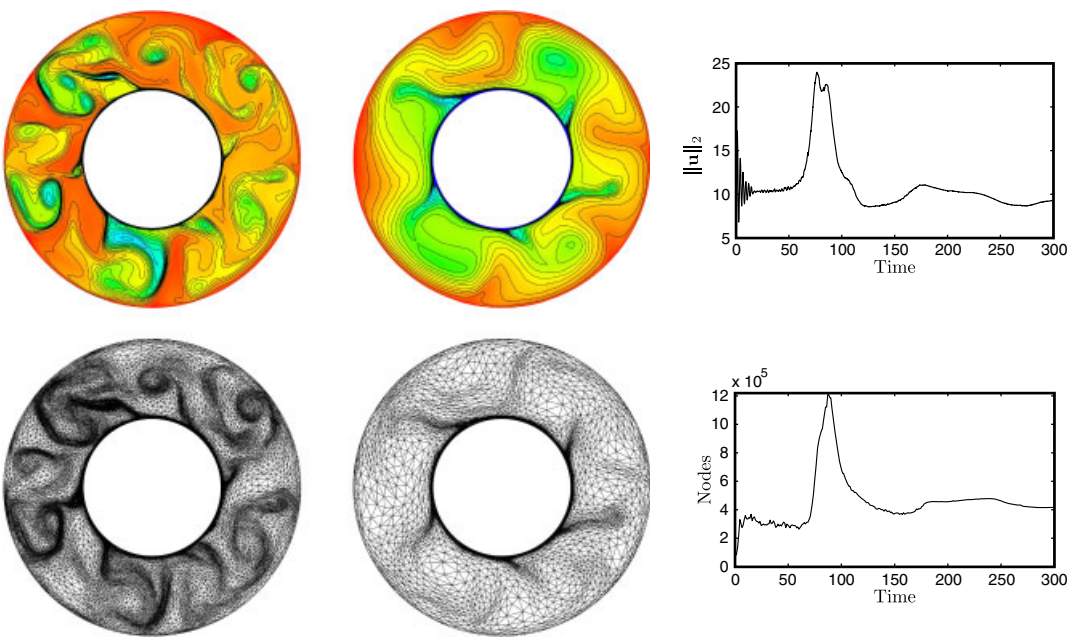

Figure 5. Temperature field and mesh at the upper surface for the rotating annulus problem at times 87 and 200 s. Also shown are the two norm of the velocity field and total number of nodes.

DOI: 10.1002/fld

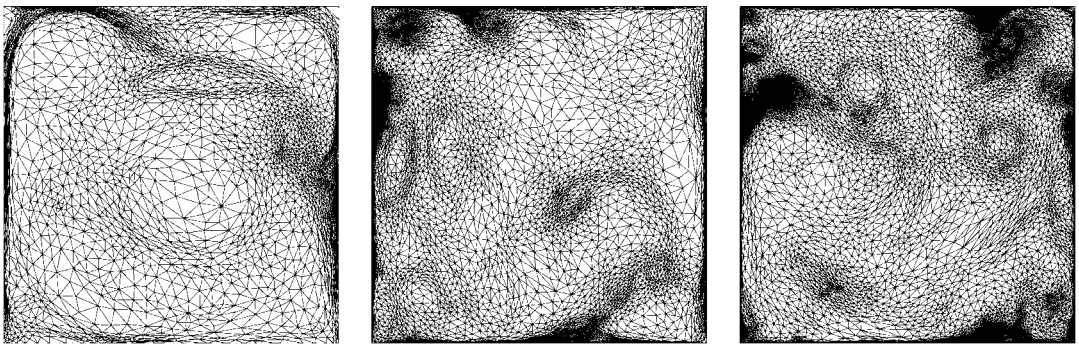

Figure 6. Western boundary current simulation at Reynolds numbers 625, 2500 and 10 000. Meshes are shown after approximately one year of simulation time and contain 5609, 14 817 and 36 848 nodes.

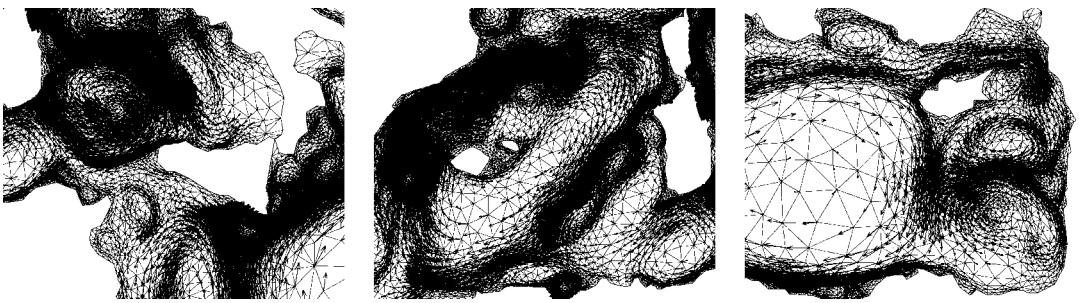

Figure 7. Mesh and velocity vectors in a flat-bottomed domain with a shoreline from the Mediterranean Sea. The simulation comprises 54 255 nodes at this time level.

6.2. Example 4: barotropic wind-driven circulation and western boundary currents

In this problem, a flat-bottomed domain of square horizontal extent 1000 km and 500 m depth on a beta plane with $\beta = 1.8 \times 10^{-11}\,\mathrm{m^{-1}\,s^{-1}}$ is forced with a wind stress of strength $\tau_0 = 0.1\,\mathrm{N\,m^{-2}}$ which varies as a sine of latitude and sets up an anti-cyclonic gyre. Owing to the variable strength of rotation an intense western boundary is formed. Figure 6 presents results after approximately one year of simulated time. Figure 7 shows results from a similar simulation with Mediterranean shoreline.

6.3. Example 5: tidally forced internal wave breaking at a shelf break of critical slope

Adaptive simulations of the interaction of an internal gravity wave with a shelf break of critical slope have been conducted following [16]. Parameters and model domain have been set up to closely match their 'highly nonlinear' case. The problem is forced at the left-hand boundary with a first-mode internal wave which interacts with an idealized shelf break that is of critical slope. Figure 8 shows the wave breaking and a bolus of dense water propagating along the shelf.

DOI: 10.1002/fld

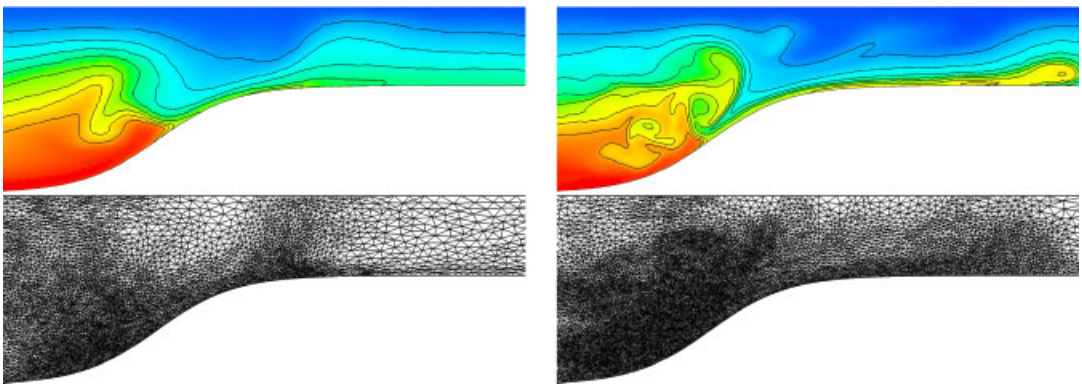

Figure 8. Density field and mesh for internal wave breaking at a shelf break of critical slope at 4.6 and 5.6 times the wave period T. Overturning of the isopycnals and 'bolus' propagation can be seen.

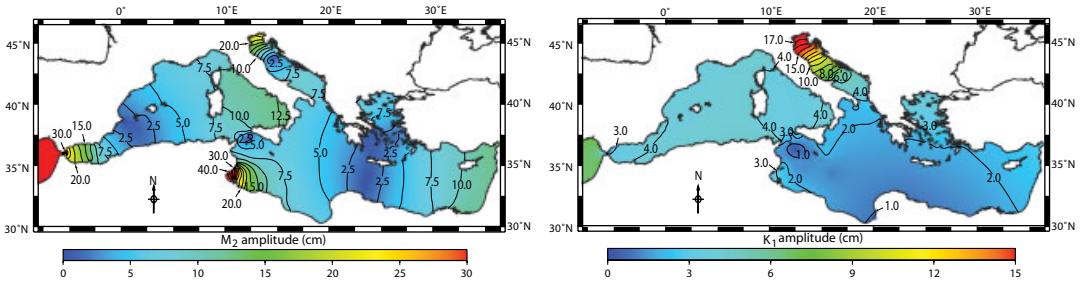

Figure 9. Amplitude from $M2$ (left) and $K1$ (right) Mediterranean tidal simulations. The mesh is constructed on the surface of an assumed sphere representing the surface of the Earth. The fixed mesh contains 179 726 and 518 955 nodes and is constructed following the methodology of Section 4.

6.4. Example 6: tides in the Mediterranean Sea

Figure 9 shows results from tidal simulations in the Mediterranean Sea. The simulation is forced with the astronomical tide, and a free surface height taken from the FES2004 global tide model [17] is imposed at the open boundary to the West. The main patterns of the M2 (principal lunar semi-diurnal) tidal constituent match those of [18] well. Amphidromic points are present in the same parts of the sea as [18], i.e. the north-central Adriatic, between Crete and Lybia, Sicilian Channel, southwest of the Balearic Islands. Areas of tidal amplification are also correctly located, i.e. northern Adriatic, northern Aegean, Gulf of Gabes and the eastern end of the Mediterranean. Similarly, promising results are obtained for the K1 and other constituents tested. For further details and examples, see [8].

7. CONCLUDING REMARKS

A new approach to ocean modelling in 3D which is applicable to both the large and small scales simultaneously has been presented. Anisotropic mesh optimization and parallel methods are used together to efficiently achieve high spatial resolution where and when appropriate. Preference for

DOI: 10.1002/fld

the vertical direction can be included, and should emerge naturally by the use of appropriate error measures that are currently under development. Other algorithmic developments include data assimilation capabilities *via* an adjoint to the model which takes into account the evolving mesh, and the use of mesh movement to track structures such as density levels. Work is also ongoing on further validation and additional applications, especially those where small-scale processes have important impacts on the larger scale and *vice versa*, these will be the subject of future publications.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the funding of NERC under grants NE/C52101X/ and NE/C51829X/1, and the support of colleagues at Imperial College London, the University of Oxford, the National Oceanography Centre Southampton and Proudman Oceanographic Laboratory.

REFERENCES

1. Haidvogel DB, Beckmann A. *Numerical Ocean Circulation Modeling*. Imperial College Press: London, U.K., 1999.
2. Ford R, Pain CC, Piggott MD, Goddard AJH, de Oliveira CRE, Umpleby AP. A non-hydrostatic finite element model for three-dimensional stratified oceanic flows. Part I: model formulation and Part II: model validation. *Monthly Weather Review* 2004; **132**:2816–2844.
3. Pain CC, Piggott MD, Goddard AJH, Fang F, Gorman GJ, Marshall DP, Eaton MD, Power PW, de Oliveira CRE. Three-dimensional unstructured mesh ocean modelling. *Ocean Modelling* 2005; **10**:5–33.
4. Piggott MD, Pain CC, Gorman GJ, Marshall DP, Killworth PD. Unstructured adaptive meshes for ocean modeling. In *Eddy-Resolving Ocean Modeling*, Hecht M, Hasumi H (eds). AGU, 2007; under review.
5. Gresho PM, Sani RL. *Incompressible Flow and the Finite Element Method*. Wiley: New York, 1998.
6. Pain CC, Liu H, Piggott MD, Gorman GJ, Wells MR, Mitchell A, Allison PA, Ham D, Goddard AJH, Killworth PD, Marshall DP. Implicit free surface algorithms for global ocean modelling. *Ocean Modelling* 2007; submitted.
7. Piggott MD, Pain CC, Gorman GJ, Power PW, Goddard AJH. h, r, and hr adaptivity with applications in numerical ocean modelling. *Ocean Modelling* 2005; **10**:95–113.
8. Wells MR, Allison PA, Piggott MD, Gorman GJ, Hampson GJ, Pain CC, Fang F. Numerical modelling of tides in the Late Pennsylvanian Midcontinent Seaway of North America with implications for hydrography and sedimentation. *Journal of Sedimentary Research* 2007; **77**:843–865.
9. Pain CC, Umpleby AP, de Oliveira CRE, Goddard AJH. Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:3771–3796.
10. Power PW, Piggott MD, Fang F, Gorman GJ, Pain CC, Marshall DP, Goddard AJH, Navon IM. Adjoint goal-based error norms for adaptive mesh ocean modelling. *Ocean Modelling* 2006; **15**:3–38.
11. Gorman GJ, Pain CC, Umpleby AP, Piggott MD. Combined parallel tetrahedral mesh optimisation and dynamic load-balancing. *Computer Methods in Applied Mechanics and Engineering* 2007; submitted.
12. Armaly BF, Durst F, Pereira JCF, Schönung B. Experimental and theoretical investigation of backward-facing step flow. *Journal of Fluid Mechanics* 1983; **127**:473–496.
13. Gorman GJ, Piggott MD, Pain CC, de Oliveira CRE, Umpleby AP, Goddard AJH. Optimal bathymetric representation through constrained unstructured mesh adaptivity. *Ocean Modelling* 2006; **12**:436–452.
14. Gorman GJ, Piggott MD, Pain CC. Shoreline approximation for unstructured mesh generation. *Computational Geosciences* 2007; **33**:666–677.
15. Hide R, Mason PJ. Sloping convection in a rotating fluid. *Advances in Physics* 1975; **24**:47–100.
16. Venayagamoorthy SK, Fringer OB. On the formation and propagation of nonlinear internal boluses across a shelf break. *Journal of Fluid Mechanics* 2007; **577**:137–159.
17. Lyard F, Lefevre F, Letellier T, Francis O. Modelling the global ocean tides: modern insights from FES2004. *Ocean Dynamics* 2006; **56**:394–415.
18. Tsimplis MN, Proctor R, Flather RA. A two-dimensional tidal model for the Mediterranean Sea. *Journal of Geophysical Research* 1995; **100**(C8):16223–16239.

DOI: 10.1002/fld