Adjoint or goal-based error norms for adaptive mesh ocean modelling

P.W. Power [a,*], C.C. Pain [a], M.D. Piggott [a], G.J. Gorman [a], F. Fang [a], D.P. Marshall [b], A.J.H. Goddard [a] and I.M. Navon [c].

[a] *Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College, Prince Consort Road, London, SW7 2BP, UK.*
URL: `http://amcg.ese.imperial.ac.uk`

[b] *Department of Meteorology, University of Reading, PO Box 243, Reading, RG6 6BB, UK.*

[c] *Department of Mathematics, Florida State University, Tallahassee FL, 32306-4120. USA.*

Abstract

Flow in the world's oceans occurs at a wide range of spatial scales, from micro-metres to mega-metres. In particular, regions of intense flow are often highly localised, for example, western boundary currents, equatorial jets and the Antarctic Circumpolar Current. Conventional numerical ocean models generally use static meshes. The use of dynamically-adaptive meshes has many potential advantages but needs to be guided by an error measure reflecting the underlying physics. A method of defining an error measure to guide an adaptive meshing algorithm for unstructured tetrahedral finite elements, utilizing an adjoint or goal-based method is described here. This method is based upon a functional, encompassing important features of the flow structure. The sensitivity of this functional, with respect to the solution variables, is used as the basis from which an error measure is derived. This error measure acts to predict those areas of the domain where resolution should be changed. A barotropic wind driven gyre problem is used to demonstrate the capabilities of the method. The overall objective of this work is to develop robust error measures for use in an oceanographic context which will ensure areas of fine mesh resolution are used only where and when they are required.

* Corresponding author
Email address: `philip.power@imperial.ac.uk` (P.W. Power).

1 Introduction

Among the smaller scale flows in the world's oceans are those associated with abrupt topographic features, such as the continental shelf break, which has a scale of the order of 10km, and tall isolated seamounts, which have a horizontal order of 20km. Western boundary currents have a typical width of 50km and are strongly influenced by topography, whereas convective plumes have a scale of the order of 100m. On the other hand, horizontal flow in the ocean basins is of the order of 1000km. These differences in scale produce challenges in numerical modelling. At present, models exist for the large-scale circulation and for oceanic flows and processes at smaller scales, but none which can simultaneously resolve the full range of pertinent scales.

Conventional numerical ocean models generally use uniformly spaced finite-differences to discretize the governing equations. The finite-element model described in Ford et al. (2004) has been generalised to allow the mesh to evolve to better resolve regions of intense flow where increased resolution is required. Coarser resolution can be used in other areas of the flow domain. The eventual aim of the work is: to exploit the use of unstructured dynamically adapting meshes in both the horizontal and vertical; to utilize the geometric flexibility of the mesh to conform well to both topography and coastlines; and to allow the variable mesh resolution to capture and optimally resolve the fluid flow at an intermediate point in time between consecutive 'adaptations' of the mesh.

Use of a dynamically adapting, variable resolution unstructured mesh allows a faithful representation of isolated local features as well as their relation to larger scale processes. An objective of implementing an adaptive mesh approach is to reduce the overall computational cost and increase accuracy of solutions; ensuring areas of fine resolution are used only when and where they are required. Although these techniques require additional computations, there is a point at which the additional overhead incurred from the use of a uniform grid, composed of elements of the finest resolution required, is greater than the overhead associated with the use of finite elements coupled to a mesh adaptivity algorithm.

Any mesh adaptation algorithm requires the derivation of an appropriate error measure. This error measure 'guides' the adaptivity algorithm, or in other words decides how the mesh is to be modified. Numerous error measures to serve this purpose have been presented in the literature. For example, interpolation based methods (Peraire et al., 1987; Wu et al., 1990; Lohner et al., 1985; Piggott et al., 2005) calculate an *a priori* measure of the error based on both the local mesh size and some higher order derivative of the exact solution, which must typically be obtained in practice using recovery from the numerical solution; there also exist various types of explicit and implicit *a posteriori* error measures (Strouboulis and Oden, 1990; Ainsworth and Oden, 1997). An error measure, for use in an oceanographic context, should encompass criteria which are important to the problem in question. For example, it should be able to identify sharp temperature gradients or regions of high vorticity. Ocean flows also occur at a very wide variety of scales and a truly rigorous error norm should reflect these issues. It should also be possible for the user to have the freedom to build into the measure tailored information or criteria that

only the expert in the application of local geographic area would possess.

The Imperial College Ocean Model (ICOM) (Ford et al., 2004; Pain et al., 2005) utilizes dynamic adaptation of a fully unstructured tetrahedral mesh in three-dimensions (3-D), as presented in Pain et al. (2001). This technique uses a form of *h-refinement* (or mesh optimization) to adapt the mesh, changing the size, shape and location of tetrahedral elements to optimize the mesh according to specific criteria, as defined by an error measure. The algorithm is based on a series of mesh connectivity and node position searches, defining the mesh quality. A Riemannian metric tensor reflecting the error measure is used to calculate the desired element size and, importantly, shape. A functional is used to gauge the mesh quality, this functional embodies both element size and shape with respect to the metric tensor. A local based search strategy is adopted to carry out the adaptation operations: node smoothing; edge and face-edge swapping; and edge splitting and collapsing, to minimize the functional. The algorithm is robust, produces high quality anisotropic meshes, and has a time complexity which varies linearly with the number of elements, see Pain et al. (2001). The anisotropy of the method offers both substantial computational improvements over fixed mesh methods and feature-following opportunities.

The error measure utilized in this work is one based upon both the curvature of the solution (which provides directional information) and a required specification of an appropriate interpolation error, derived from a goal-based method. The required interpolation error varies in time and space through the simulation. Scaling of the resulting metric tensor (Pain et al., 2001; Castro-Diaz et al., 1997) allows phenomena acting at various scales to be resolved. The overall approach effectively uses the current solution of the problem in hand to adapt the mesh to reflect future 'activity' in the flow.

The motivation for the incorporation of sensitivity analysis in error measure design stems from the assumption that it is ineffective and undesirable (computationally speaking) to adapt the mesh every time step (or few time steps); so in a dynamically evolving flow the mesh could be said to be 'behind the flow', and therefore not in an optimal form to resolve features and provide the best solution. The method presented here is a first step toward being able to 'predict' the flow's future movement (at least to highlight possibly important areas), and adapt the mesh accordingly.

Sensitivity analysis deals with the calculation of the gradients of a model forecast with respect to model parameters, where these parameters might be initial conditions, boundary conditions or other physical inputs. Use of an adjoint model, first introduced in Cacuci (1981), can identify regions where changes to variables or parameters has the largest impact. Put simply, an area with high 'sensitivity' is one within which small perturbations can strongly influence the growth of errors in the overall solution. Sensitivity analysis has been applied in a variety of fields, including the control of water through irrigation channels (Sanders and Katopodes, 1999) and contaminant releases in rivers (Piasecki and Katopodes, 1997), applications to the shallow water equations (Sanders and Katopodes, 2000), as well as various meteorological applications (Zhang et al., 2000; Gunzburger, 1999; Lang-

land et al., 1995). Theoretical considerations have been presented by (Ilioescu and Navon, 2003; Cacuci, 1988), with later applications to optimal control, 3/4D-VAR data assimilation and error estimation (Alekseev and Navon, 2005).

The aim of goal-based error measure design is to take a measure of what is 'important' in a problem and design an error measure, and consequently mesh adaptation scheme, to optimize the accuracy of this quantity. A bound on the required accuracy for this quantity (goal) can be set, and the method presented in this paper yields a mesh which achieves this level of accuracy with minimal computational resources. A method of this type is particularly important for problems with a number of solution variables where it is typically unclear what priority to put on resolving each of the solution fields. The method developed here provides a systematic way of doing this. The error contribution to each of the nodal solution variables can be determined and used to substantially improve the accuracy of the goal functional, see Pierce and Giles (2000); Müller and Giles (2001). In addition, sensitivity information may also be used to provide a bound for the error in this functional, which can be invaluable to any model, see Paraschivoiu and Patera (1998). An example of an application area of this type of technique is in the adaptation of a mesh to optimize a quantity of interest in a fluid simulation such as the drag or lift past an aerofoil (Pierce and Giles, 2000). The particular interest here is for an ocean model, as described by Ford et al. (2004); Pain et al. (2005). The goal function in such a model could be an observation or some measure of the dynamics of the system, for example some integral of vorticity or the strength of the thermohaline circulation.

What distinguishes this work from previous work on goal-based error measures and adaptivity, for example in (Oden and Prudhomme, 2001; Prudhomme and Oden, 1999; Cirak and Ramm, 1998; Venditti and Darmofal, 2000), is the use of readily accessible (in simulation codes) discretized equations, the application to transient problems and the use of a metric tensor obtained from sensitivity analysis to adapt three-dimensional meshes of unstructured tetrahedral elements. The metric tensor is used to gauge both shape and size quality of each tetrahedral element which forms the basis of a mesh adaptivity/optimization procedure. Since the adjoint solution is typically calculated in data assimilation, control, and optimization problems, then the method presented in this paper provides an additional use for the adjoint information. See Hide (1989, 2002) for a number of important diagnostics which may be used to generate error measures.

The remainder of this paper is organised as follows. Section 2 gives a brief overview of the anisotropic adaptivity method employed here. Section 3 outlines the methodology for the goal-based method and the creation of a metric tensor. Section 4 outlines how the method is implemented for ocean modelling problems. Section 5 briefly outlines the model equations which are then utilized to solve a wind driven rotating gyre problem, using the goal-based technology, in section 6. In section 7 some conclusions are drawn.

2 Anisotropic Mesh Adaptivity Method

For completeness a brief overview of the mesh optimization algorithm employed in this work is given in this section. A more detailed description of the method is given by Pain et al. (2001).

2.1 Metric tensor

The mesh optimization method presented requires as a precursor an error measure in the form of a nodally defined metric tensor (George, 1998). This positive definite matrix defines, anisotropically, the desired mesh edge lengths at each node. The desired edge length, h_i, in the direction of the i^{th} eigenvector, $\mathbf{e}_i$, of the metric tensor $\boldsymbol{M}$, is defined as $h_i = 1/\sqrt{\lambda_i}$, where λ_i is the eigenvalue associated with $\mathbf{e}_i$. In general the metric tensor is used to calculate distances during mesh optimization via $\|v\| = v^T \boldsymbol{M}_v v$ where $\boldsymbol{M}_v$ is the average metric tensor along vector v. This approach may be viewed as a discretization of a Riemannian geometry constructed from an error norm. As the directionality of the solution is encoded in the metric tensor, anisotropic solutions will generally lead to anisotropic meshes that balance accuracy with computational efficiency For example, the fact that the method seeks a specific error means that the procedure will remove elements where there is a glut in accuracy, thus improving computation efficiency.

When forming a suitable metric tensor a number of generic operations are regularly applied. These include: limiting the maximum and minimum desired edge lengths so as not to specify unrealistic modelling goals; limiting the aspect ratio of principal directions of an element; combining several metric tensors from different solution fields; global scaling of the metric tensor in order to limit the computational resources the model may request (i.e. limit the number of degrees of freedom in the model); and metric tensor gradation control. Most of these operations are discussed extensively in the literature (e.g. Pain et al. (2001); George (1998)) and so will not be elaborated here.

2.2 Element functional

To define a mesh optimization problem an element functional is defined in terms of the metric tensor and of the properties a good mesh should exhibit for modelling. Trials are then performed on the local mesh connectivity and node position — in the case of a minimization problem being defined the aim is to find a local configuration that reduces the functional value. Defining the mesh functional as

$$\mathcal{F} = ||\boldsymbol{F}||_\infty, \tag{1}$$

where $\boldsymbol{F}$ is the vector of element functionals for the whole mesh, the process terminates when $\mathcal{F}$ falls below some tolerance. There are many possible choices for the definition of the local functional $\mathcal{F}$ (see Knupp (2000) for a review). The element

functional used here is geometrical based.

$$F_e = \frac{1}{2} \sum_{\ell \in \mathcal{L}_e} (r_\ell - 1)^2 + \mu \left(\frac{\alpha}{\rho_e} - 1 \right)^2 . \tag{2}$$

Here r_ℓ is the length, with respect to the edge centred metric tensor, $\boldsymbol{M}_\ell$, of edge ℓ; $\mathcal{L}_e$ is the set of edges of element e; ρ_e is the radius, with respect to $\boldsymbol{M}$, of the inscribed sphere (insphere) of element e; and α is the radius of the inscribed sphere of an ideal element. In 3-D the ideal element is defined in metric space as an equilateral tetrahedron with sides of unit length.

The trade-off between size and shape is controlled by the parameter μ, which has been chosen to be unity for the work here. The first term in the functional becomes zero as all of the edge lengths approach unity (measured with respect to the metric tensor $\boldsymbol{M}$), and the second term becomes zero as the in-sphere radius approaches α (measured with respect to the metric tensor $\boldsymbol{M}$). In this way the functional gives a measure of the quality of an element in terms of both its size and shape.

2.3 Mesh adaptations

For a given error metric and objective functional, the tetrahedral mesh is adapted through a combination of:

- node insertion/deletion via edge splitting and collapsing,
- face-to-edge swapping and edge swapping (Joe, 1989),
- laplacian smoothing (in metric space),
- optimization-based smoothing.

It is worth noting that: edge collapsing reduces the number of elements and nodes thereby coarsening the mesh; edge splitting increases the number of elements and nodes, thereby refining the mesh; face and edge swapping, and mesh smoothing primarily serve to modify the shape of elements. This multi-pronged approach is similar to that of Freitag and Ollivier-Gooch (1997) except that the objective function described above is used rather than purely Euclidean measures of element quality.

3 An Adjoint Based Sensitivity Measure

This section outlines the basis for calculation of an alternative metric tensor $\boldsymbol{M}$, based upon both forward and adjoint calculations.

3.1 A goal-based error measure

In this section two equivalent error measures are derived, based on the forward and adjoint solutions. Suppose a differential equation to be solved is

$$\mathscr{L} \psi_{exact} - s = 0, \tag{3}$$

for source s, linear operator $\mathscr{L}$ (the extension to non-linear operators is relatively straightforward although it can involve considerable algebra), and the exact solution

is $\psi_{exact} = \psi_{exact}(\mathbf{x})$. This solution is approximated with a finite element scheme as

$$\psi \equiv \psi(\mathbf{x}) = \sum_{j=1}^{\mathcal{N}} N_j(\mathbf{x}) \Psi_j, \tag{4}$$

with $\mathbf{x}$ being the spatial coordinates; discrete solution vector $\Psi \equiv (\Psi_1, \Psi_2, \ldots, \Psi_{\mathcal{N}})^T$; basis function $N_j(\mathbf{x})$; and $\mathcal{N}$ the number of nodes in the finite element mesh. Then the equation residual is

$$\mathscr{R}(\psi) \equiv \mathscr{L}\psi - s. \tag{5}$$

The aim is to make this residual small in some sense. This is done by multiplying equation 5 by a weighting function. In the Bubnov-Galerkin method this weighting function is chosen to be the basis function $N_i(\mathbf{x})$. Whatever method is chosen to discretize the equations a matrix equation and residual vector result:

$$r(\psi) = \mathbf{A}\Psi - S = 0, \tag{6}$$

for matrix $\mathbf{A}$ and discretized source S. In a practical implementation with inexact arithmetic and possibly the use of iterative solution methods, $r(\psi)$ may not be identically zero. It is assumed insignificantly small in this work. This assumption may be relaxed by retaining $r(\psi)$ in the following equations. In addition, note that in the analysis it is assumed that $r(\psi)$ is in some sense a discretized representation of the residual $\mathscr{R}(\psi)$ multiplied by a representative volume of each cell or node, as achieved in typical finite element Petrov-Galerkin or control volume methods.

3.1.1 The functional or goal

Suppose that the functional whose accuracy is to be optimized is represented as $F \equiv F(\psi)$, and

$$F(\psi) = \int_\Omega f(\psi)\, dV, \tag{7}$$

where Ω is the solution domain. $F(\psi)$ may be any derived quantity of the solution ψ. Applying a first order Taylor series analysis, the gradient $\frac{\partial f}{\partial \psi}$ near the exact solution ψ_{exact} can be obtained from

$$\frac{\partial f}{\partial \psi}(\psi_{exact} - \psi) \approx f(\psi_{exact}) - f(\psi), \tag{8}$$

or in discrete form,

$$\left(\frac{\partial F}{\partial \Psi}\right)^T (\Psi_{exact} - \Psi) \approx F\left(\widetilde{\psi}_{exact}\right) - F(\psi), \tag{9}$$

in which $\Psi_{exact} \equiv (\Psi_{exact_1}, \Psi_{exact_2}, \ldots, \Psi_{exact_{\mathcal{N}}})^T$ is a vector containing the exact solution at the $\mathcal{N}$ finite element nodes (or control volume cells) and

$$\widetilde{\psi}_{exact} = \sum_{j=1}^{\mathcal{N}} N_j(\mathbf{x}) \Psi_{exact_j}. \tag{10}$$

That is, $\widetilde{\psi}_{exact}$ is the finite element (or other numerical) interpolant of the exact solution ψ_{exact}.

3.1.2 Discrete error measure

Suppose a non-singular matrix $\mathbf{A}_{exact}$ exists such that

$$r_{exact}\left(\widetilde{\psi}_{exact}\right) = \mathbf{A}_{exact}\Psi_{exact} - S = 0, \tag{11}$$

and

$$r^*_{exact}\left(\widetilde{\psi}^*_{exact}\right) = \left(\mathbf{A}_{exact}\right)^T \Psi^*_{exact} - \frac{\partial F}{\partial \Psi} = 0. \tag{12}$$

Define $\Psi^*_{exact} \equiv \left(\Psi^*_{exact_1}, \Psi^*_{exact_2}, \ldots, \Psi^*_{exact_{\mathcal{N}}}\right)^T$ which is associated with the finite element representation

$$\widetilde{\psi}^*_{exact} = \sum_{j=1}^{\mathcal{N}} N_j\left(\mathbf{x}\right)\Psi^*_{exact_j}. \tag{13}$$

Therefore,

$$\frac{\partial\left(r_{exact}\left(\widetilde{\psi}_{exact}\right)\right)}{\partial\Psi}\left(\Psi_{exact} - \Psi\right) \approx r_{exact}\left(\widetilde{\psi}_{exact}\right) - r_{exact}\left(\psi\right) \tag{14}$$

$$\approx r\left(\widetilde{\psi}_{exact}\right) - r\left(\psi\right). \tag{15}$$

Then using the differential of equation 11,

$$\left(\Psi_{exact} - \Psi\right) \approx \left(\mathbf{A}_{exact}\right)^{-1}\left(r\left(\widetilde{\psi}_{exact}\right) - r\left(\psi\right)\right). \tag{16}$$

This can then be combined with equations 9 and 12 to obtain

$$\begin{aligned} F\left(\widetilde{\psi}_{exact}\right) - F\left(\psi\right) &\approx \left(\frac{\partial F}{\partial \Psi}\right)^T \left(\mathbf{A}_{exact}\right)^{-1}\left(r\left(\widetilde{\psi}_{exact}\right) - r\left(\psi\right)\right) \\ &= \left(r\left(\widetilde{\psi}_{exact}\right) - r\left(\psi\right)\right)^T \left(\mathbf{A}_{exact}\right)^{-T}\left(\frac{\partial F}{\partial \Psi}\right) \\ &= \left(r\left(\widetilde{\psi}_{exact}\right) - r\left(\psi\right)\right)^T \Psi^*_{exact}. \end{aligned} \tag{17}$$

This expression can also be used to obtain an improved prediction of F, that is to find $F\left(\widetilde{\psi}_{exact}\right)$, once estimates of $r\left(\widetilde{\psi}_{exact}\right)$ and $\widetilde{\Psi}^*_{exact}$ have been obtained.

3.1.3 Discrete functional correction

Although the error estimates are obtained from equation 17, the product $\left(r\left(\widetilde{\psi}_{exact}\right) - r\left(\psi\right)\right)^T \Psi^*_{exact}$ is not known and cannot be used directly to correct the functional F. To do this the following relation is employed

$$F\left(\widetilde{\psi}_{exact}\right) - F\left(\psi\right) \approx \left(r\left(\widetilde{\psi}_{exact}\right) - r\left(\psi\right)\right)^T \Psi^* \approx \left(\hat{r}\left(\psi\right) - r\left(\psi\right)\right)^T \Psi^*. \tag{18}$$

$r(\psi)$ is approximately equal to $r\left(\psi_{exact}\right)$ and is obtained from the residual calculation method outlined in section 3.3. Equation 18 can also be used to re-normalise the error measures to correctly obtain the desired error in F.

3.1.4 Dual discrete error measure

Another error measure, in a discrete sense can be defined from equation 17,

$$\begin{aligned} F\left(\psi_{exact}\right) - F\left(\psi\right) &\approx \left(r\left(\widetilde{\psi}_{exact}\right) - r\left(\psi\right)\right)^T \Psi^*_{exact} \\ &= \left(\left(\mathbf{A}\Psi_{exact} - S\right) - \left(\mathbf{A}\Psi - S\right)\right)^T \Psi^*_{exact} \\ &= \left(\mathbf{A}\left(\Psi_{exact} - \Psi\right)\right)^T \Psi^*_{exact} = \left(\Psi_{exact} - \Psi\right)^T \mathbf{A}^T \Psi^*_{exact} \\ &= \left(\mathbf{A}^T \Psi^*_{exact}\right)^T \left(\Psi_{exact} - \Psi\right). \end{aligned} \tag{19}$$

This is used to form a directional error measure based on the forward solution.

3.1.5 Alternative error measures

The dual error measure measures derived in section 3.1.4 can be written, assuming $r\left(\psi\right) = 0$, as

$$\begin{aligned} F\left(\psi_{exact}\right) - F\left(\psi\right) &\approx r\left(\widetilde{\psi}_{exact}\right)^T \Psi^*_{exact} \\ &= r\left(\widetilde{\psi}_{exact}\right)^T \Psi^* + r\left(\widetilde{\psi}_{exact}\right)^T \left(\Psi^*_{exact} - \Psi^*\right) \\ &= r\left(\widetilde{\psi}_{exact}\right)^T \Psi^*_{exact} + r^*\left(\widetilde{\psi}^*_{exact}\right)^T \left(\Psi_{exact} - \Psi\right). \end{aligned} \tag{20}$$

From equation 20, a useful relationship which has an analagous continuous form is

$$F\left(\psi_{exact}\right) - F\left(\psi\right) \approx \left(\frac{\partial F}{\partial \psi}\right)^T \left(\Psi_{exact} - \Psi\right) + r\left(\widetilde{\psi}_{exact}\right)^T \left(\Psi^*_{exact} - \Psi^*\right). \tag{21}$$

Equation 21 is used to form a directional error measure based on the adjoint solution. Similar results to equation 20 have been used by Venditti and Darmofal (2000), in which the latter part of equation 21, that is $r\left(\widetilde{\psi}_{exact}\right)^T \left(\Psi_{exact} - \Psi\right)$ is used to gauge mesh adaptivity isotropically (Venditti and Darmofal, 2002) and anisotropically (Venditti and Darmofal, 2003). In Pierce and Giles (2000) a defect correction method, similar in form, is presented to increase the accuracy of the estimation. Müller and Giles (2001) use a similar approach to that described here, but using an isotropic error measure. Note that equation 20 can also be written for linear operators as

$$F(\psi_{exact}) - F(\psi) \approx \left(\mathbf{A}^T \left(\Psi^*_{exact} - \Psi^* + \mathbf{A}^{-T} \frac{\partial F}{\partial \psi} \right) \right) \quad (\Psi_{exact} - \Psi)$$

$$= \left(\mathbf{A}^T (\Psi^*_{exact} - \Psi^*) + \frac{\partial F}{\partial \psi} \right)^T (\Psi_{exact} - \Psi)$$

$$= (\Psi^*_{exact} - \Psi^*)^T \mathbf{A} (\Psi_{exact} - \Psi) + \left(\frac{\partial F}{\partial \psi} \right)^T (\Psi_{exact} - \Psi) . \quad (22)$$

Estimates of the solution errors $(\Psi^* - \Psi^*_{exact})$ and $(\Psi - \Psi_{exact})$ can readily be obtained using interpolation theory as outlined in section 3.2.

3.2 Derivation of a Metric Tensor field to guide anisotropic mesh adaptivity

In section 3.1 the error measures (equations 19 and 21) were derived. In this section these error measures are modified slightly by approximating the unknown quantities $\mathbf{A}^T \Psi^*_{exact}$ and $r\left(\widetilde{\psi}_{exact}\right)$ with known quantities $\widehat{\mathbf{A}^T \Psi^*_{exact}}$ and $\hat{r}\left(\widetilde{\psi}_{exact}\right)$ respectively. Full details of how this is implemented are outlined in section 3.3. The error measures (equations 19 and 21) then become,

$$F(\psi_{exact}) - F(\psi) \approx \left(\widehat{\mathbf{A}^T \Psi^*_{exact}} \right)^T (\Psi_{exact} - \Psi) , \quad (23)$$

$$F(\psi_{exact}) - F(\psi) \approx \left(\frac{\partial F}{\partial \psi} \right)^T (\Psi_{exact} - \Psi) + \hat{r}\left(\widetilde{\psi}_{exact}\right)^T (\Psi^*_{exact} - \Psi^*) . \quad (24)$$

These equations require a measure of the error in the forward solution (equation 23) and both forward and adjoint solutions (equation 24).

3.2.1 Defining a Metric Tensor

A metric tensor (see section 2) can be defined:

$$\bar{\boldsymbol{M}} = \frac{\gamma}{|\epsilon|} |\boldsymbol{H}| .$$

Here $\boldsymbol{H}$ is the hessian matrix, ϵ is the required level of error and γ a scalar constant. Ideal elements then have sides of length unity when measured with respect to this metric tensor. The discrete (nodal) form of ϵ, from equation 19 is

$$\epsilon_i = \frac{\widetilde{\delta F}}{\left| \left(\widehat{\mathbf{A}^T \Psi^*_{exact}} \right)_i \right|} . \quad (25)$$

Here $\left(\widehat{\mathbf{A}^T \Psi^*_{exact}} \right)_i$ is the i^{th} entry of the vector $\widehat{\mathbf{A}^T \Psi^*_{exact}}$ and $\epsilon = (\psi_{exact} - \psi) = \sum_{j=1}^{\mathcal{N}} N_j(\mathbf{x}) \epsilon_j$. Suppose δF is the acceptable error in F; then assuming the error contribution to F is the same for each node, then we can define

$$\widetilde{\delta F} = \frac{\delta F}{\mathcal{N}} . \quad (26)$$

5.2.2 Application to multiple field problems

For each solution variable l and each node i one can define two Hessians: $\boldsymbol{H}_i^l$ associated with the forward solution ψ^l and $\boldsymbol{H}^{*l}_i$ associated with the adjoint solution ψ^{*l}. The Hessian matrices are defined as

$$\boldsymbol{H}_i^l \equiv \left(\nabla^T \nabla \psi^l\right)_i, \quad \boldsymbol{H}^{*l}_i = \left(\nabla^T \nabla \psi^{*l}\right)_i.$$

To calculate the Hessians, the method presented in Pain et al. (2001) is followed. Galerkin methods are repeatedly applied to calculate the first derivatives. Let us just consider the forward Hessian constructed from the forward solution ψ^l, consequently at node i:

$$\left.\frac{\partial \psi^l}{\partial x}\right|_i \approx q_{x_i} = \mathsf{M_L}_i^{-1} \int N_i \left(\frac{\partial \psi^l}{\partial x}\right) dV$$

and in a similar way for $q^l_{y_i}$ and $q^l_{z_i}$. Therefore,

$$\left(\frac{\partial \psi^l}{\partial x}\right) = q_x = \sum_{j=1}^{\mathcal{N}} \frac{\partial N_j}{\partial x} \psi_j,$$

where N_j are the finite element basis functions and $\mathsf{M_L}$ is the row summed lumped mass matrix, see Zienkiewicz and Taylor (1991). The second order terms which form the Hessian for an equation solution variable l centred on node i,

$$\boldsymbol{H}_i^l = \begin{pmatrix} q^l_{xx_i} & q^l_{xy_i} & q^l_{xz_i} \\ q^l_{yx_i} & q^l_{yy_i} & q^l_{yz_i} \\ q^l_{zx_i} & q^l_{zy_i} & q^l_{zz_i} \end{pmatrix},$$

are calculated in a similar way, for example

$$q^l_{xx_i} = \mathsf{M_L}_i^{-1} \int N_i \frac{\partial q^l_x}{\partial x} dV, \qquad q^l_{xy_i} = \mathsf{M_L}_i^{-1} \int N_i \frac{\partial q^l_x}{\partial y} dV.$$

If there are $\mathcal{M}$ solution variables per node then an averaged Hessian $\bar{\boldsymbol{H}}_i$ associated with node i can be defined as

$$\bar{\boldsymbol{H}}_i = \frac{1}{\sum_{l=1}^{\mathcal{M}} \left|\lambda_i^l\right|} \sum_{l=1}^{\mathcal{M}} \left|\lambda_i^l\right| \left|\boldsymbol{H}_i^l\right|, \tag{27}$$

where

$$\lambda_i^l = \left(\widehat{\mathbf{A}^T \Psi^*_{exact}}\right)^l_i.$$

This technique uses equation 23 as the basis for this average. Using the error measure defined by equation 23 the metric tensor field can be found. An interpolation error $\bar{\epsilon}_i$ at node i can be defined as

$$\bar{\epsilon}_i = \frac{\widetilde{\delta F}}{\sum_{l=1}^{\mathcal{M}} \left|\lambda_i^l\right|}, \tag{28}$$

then the nodal metric tensor $\boldsymbol{M}_i$ is obtained from

$$\bar{\boldsymbol{M}}_i = \frac{\gamma}{|\bar{\epsilon}_i|} \left| \bar{\boldsymbol{H}}_i \right|. \tag{29}$$

This is the metric tensor used in the applications (see section 6). Using the adjoint error measure defined by equation 24 a second metric tensor can be obtained. To this end suppose that

$$\lambda^{+l}_{\ i} = \left(\frac{\partial F}{\partial \psi} \right)^l_i, \quad \lambda^{*l}_{\ i} = \hat{r} \left(\psi \right)^l_i,$$

then a combined averaged Hessian $\bar{\boldsymbol{H}}^*_i$ associated with node i can be defined as

$$\bar{\boldsymbol{H}}^*_i = \frac{1}{\sum_{l=1}^{\mathcal{M}} \left(\left| \lambda^{+l}_{\ i} \right| + \left| \lambda^{*l}_{\ i} \right| \right)} \sum_{l=1}^{\mathcal{M}} \left(\left| \lambda^{+l}_{\ i} \right| \left| \boldsymbol{H}^l_i \right| + \left| \lambda^{*l}_{\ i} \right| \left| \boldsymbol{H}^{*l}_{\ i} \right| \right). \tag{30}$$

An adjoint based interpolation error $\bar{\epsilon}^*_i$ can be defined as

$$\bar{\epsilon}^*_i = \frac{\widetilde{\delta F}}{\sum_{l=1}^{\mathcal{M}} \left(\left| \lambda^{+l}_{\ i} \right| + \left| \lambda^{*l}_{\ i} \right| \right)}. \tag{31}$$

So a new adjoint based metric tensor $\bar{\boldsymbol{M}}^*_i$ at node i is obtained from

$$\bar{\boldsymbol{M}}^*_i = \frac{\gamma}{|\bar{\epsilon}^*_i|} \left| \bar{\boldsymbol{H}}^*_i \right|. \tag{32}$$

3.2.3 Superimposition of Metric Tensors

In practise, one metric tensor $\boldsymbol{M}$ is required to guide mesh adaptivity. A single metric tensor field can be obtained by either ignoring the metric tensor $\bar{\boldsymbol{M}}^*$ and using only $\bar{\boldsymbol{M}}$ as the basis of mesh adaptivity, thus the overall Metric Tensor $\boldsymbol{M}_i$ at node i is obtained from

$$\boldsymbol{M}_i = \bar{\boldsymbol{M}}_i, \tag{33}$$

or alternatively by including both metric tensors $\bar{\boldsymbol{M}}$ and $\bar{\boldsymbol{M}}^*$ where

$$\boldsymbol{M}_i = \mathcal{G} \left(\bar{\boldsymbol{M}}_i, \bar{\boldsymbol{M}}^*_i \right). \tag{34}$$

In this case the operator $\mathcal{G}\left(\mathbf{A}, \mathbf{B}\right)$ superimposes the two metric tensors $\mathbf{A}$ and $\mathbf{B}$ by the method presented in Pain et al. (2001).

3.3 Determining equation residuals

In this section approximations $\widehat{\mathbf{A}^T \Psi^*_{exact}}$ and $\hat{r}\left(\tilde{\psi}_{exact}\right)$ to $\mathbf{A}^T \Psi^*_{exact}$ and $r\left(\tilde{\psi}_{exact}\right)$ respectively, are obtained for a particular node i. This is achieved by finding a coarse grid representation of the discrete matrix equations at each node i; the residual is obtained when an approximation to the adjoint solution Ψ^* is evaluated at the coarse grid nodes and used in conjunction with the resulting coarse grid matrix. The coarse grid residuals are obtained algebrically, so that no access to the governing equations

is necessary. This takes a step towards the possibility of developing a method (and an associated computer implementation) that is independent of the governing equations. With knowledge of how the residual varies with the grid resolution one can determine an estimate of the residual. A fine grid stencil is constructed from the nodes directly connected to the node i; these nodes form the set $\mathcal{S}_i$ and are shown as filled circles in figure 1. The fine grid stencil is indicated by solid bold lines in figure 1. The fine grid stencil is extended by a factor of two in each direction to give the location of coarse grid nodes. The coordinates of the coarse nodes $\mathbf{x}_c$, related to the nodes $j \in \mathcal{S}_i$ are defined as

$$\mathbf{x}_c = \mathbf{x}_i + 2\left(\mathbf{x}_j - \mathbf{x}_i\right). \tag{35}$$

These points are shown with solid open circles in figure 1, the extension of the stencil is shown with a bold dashed line. To apply quadratic interpolation a further point must be defined, $\mathbf{x}_d$ say. Similarly, the coordinates of the coarse nodes $\mathbf{x}_d$, related to the nodes $j \in \mathcal{S}_i$ are defined as

$$\mathbf{x}_d = \mathbf{x}_i - 2\left(\mathbf{x}_j - \mathbf{x}_i\right). \tag{36}$$

These points are shown with dashed open circles in figure 1, the extension of the stencil shown with a standard weight dashed line. This process results in points $\mathbf{x}_c$ and $\mathbf{x}_d$, related to each node $j \in \mathcal{S}_i$. These points generally lie inside elements of the fine (original) mesh. The elements are identified by means of a local vicinity search, initiated from one of the elements surrounding node i. The values of a field, for the purposes of this work the solution to the adjoint problem, Ψ^{*l} at points $\mathbf{x}_c$ and $\mathbf{x}_d$ can then be found by interpolating from the known values of Ψ^{*l} at the nodes of the element in which the point lies, thus yielding ${\Psi^*}_c^l$ and ${\Psi^*}_d^l$. The objective is to find an interpolated value of Ψ^{*l} at the fine grid node j. Computing this is now a simple matter of applying a quadratic interpolation scheme, such that

$${\Psi^*}_{exact\,j}^{\;\;l} = \frac{-{\Psi^*}_d^l + 6{\Psi^*}_i^l + 3{\Psi^*}_c^l}{8}. \tag{37}$$

This process is repeated for each of the nodes j surrounding node i, the interpolated value at node i, the centre of the descretization, is taken to be ${\Psi^*}_i^l$. Figure 1 shows the stencil extensions required for a node i with three surrounding nodes j.

Generally the points $\mathbf{x}_c$ and $\mathbf{x}_d$ lie within elements of the fine (original) mesh; there are however some special cases to consider when the nodes lie close to the boundaries of the domain. As demonstrated in figure 2. It is likely that the stencil will be extended beyond the boundaries of the domain. In the case that only $\mathbf{x}_c$ lies outside the domain or both $\mathbf{x}_c$ and $\mathbf{x}_d$ lie outside the domain then

$${\Psi^*}_{exact\,j}^{\;\;l} = {\Psi^*}_j^l. \tag{38}$$

Alternatively if $\mathbf{x}_d$ lies outside the domain then linear interpolation is applied such that

$${\Psi^*}_{exact\,j}^{\;\;l} = \frac{{\Psi^*}_j^l + {\Psi^*}_c^l}{2}. \tag{39}$$

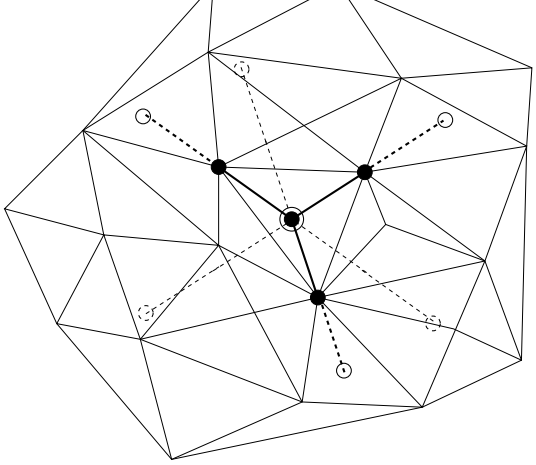

Fig. 1. Diagram showing how the coarse grid is obtained for an unstructured mesh. Filled circles represent fine grid nodes (Ψ_j); solid open circles indicate coarse grid nodes (Ψ_c) and dashed open circles represent the extra points required for interpolation (Ψ_d).

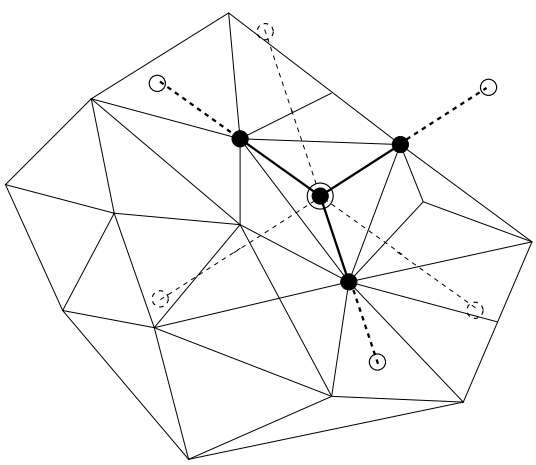

Fig. 2. Diagram showing how the coarse grid is obtained for an unstructured grid near to the domain boundary. Definitions identical to those given by figure 1. Note that it is possible to extend the stencil beyond the domain boundary, the methodology for deadling with this is detailed in section 3.3.

The interpolated values of Ψ^* at the nodes surrounding node i are enough to compute the i^{th} row of $\mathbf{A}\widehat{^T\Psi^*_{exact}}$, required by equation 25. It is sufficient to take the interpolated values of Ψ^* in this case as the values of Ψ^*_{exact}.

3.3.1 Residual Smoothing

It should also be noted that the equation residual can be quite oscillatory, see (Pierce and Giles, 2000) and has been smoothed in (Venditti and Darmofal, 2000) to obtain residuals that can be used in the equations above. As indicated in the previous section, smoothing is applied to the residuals calculated using the above method. The smoothing method employed uses a weighted average of residual values at the node in question as well as those nodes directly connected to it. For simplicity, let $\hat{r}$ be the approximation to the residual that has calculated; $\mathbf{A}\widehat{^T\Psi^*_{exact}}$ or $\hat{r}\left(\widetilde{\psi}_{exact}\right)$. This is defined at the nodes. Then a smoothed value of this, $\hat{r}_i^{\mathsf{S}}$ say, at a node i, is defined as,

$$\hat{r}_i^{\mathsf{S}} = \frac{1}{2}\hat{r}_i + \frac{1}{2\mathsf{s}_i}\sum_{j\in\mathcal{S}_i}\hat{r}_j \tag{40}$$

where $\mathcal{S}_i$ is the set of nodes directly connected to node i, excluding node i itself and s_i is the number of nodes in set $\mathcal{S}_i$. Applying this proceedure to all the nodes in turn creates a new smoothed field, the process is repeated a number of times (in the examples presented here 5 times where we amend $\mathbf{A}\widehat{^T\Psi^*_{exact}}$, in equation 23 using this approach), producing increasingly smooth fields.

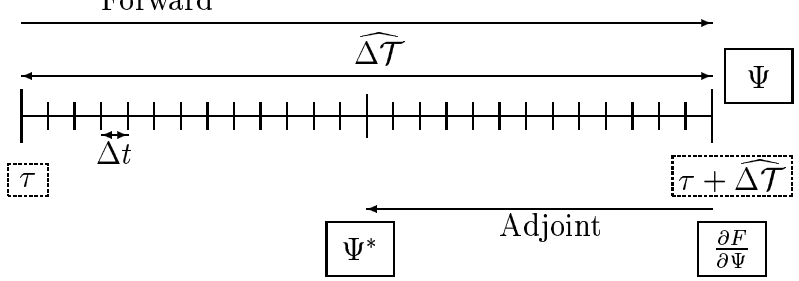

Fig. 3. Pictorial representation of time domain.

4 Outline Implementation

At a point $t = \tau$ in the computation, when one wishes to adapt the mesh, the forward problem is solved from this time level ($t = \tau$) using a time step of size $\widehat{\Delta\mathcal{T}}$. The solution of this problem provides the forward solution Ψ. The time step $\widehat{\Delta\mathcal{T}}$ is the size of a multiple of time steps Δt (where Δt is the time step size for the overall problem being solved) and is equal to the pre-set time between adaptations of the mesh.

The adjoint problem is then initialized and solved backward from this point ($t = \tau + \widehat{\Delta\mathcal{T}}$), again using a time step of size $\widehat{\Delta\mathcal{T}}$, giving the adjoint solution Ψ^*. The source term $\frac{\partial F}{\partial \Psi}$ for the adjoint problem is constructed from the solution to the forward problem Ψ (at time $t = \tau + \widehat{\Delta\mathcal{T}}$).

This procedure is best illustrated by figure 3 which shows the process for a computation between two mesh adaptations. Once Ψ and Ψ^* have been found the error measures and metric tensor $\boldsymbol{M}$ can be computed, and the mesh adapted. An important point to note is that once the error measures and metric tensor have been formed and the mesh adapted, the simulation continues, using time steps of size Δt from the time level $t = \tau$, until the next scheduled adapt (at $t = \tau + \widehat{\Delta\mathcal{T}}$), at which point the process is repeated. The initial conditions, boundary conditions and solution procedure for this forward problem (with a large time step $\widehat{\Delta\mathcal{T}}$) are identical to those for the problem being solved. Once the end of the time step $\widehat{\Delta\mathcal{T}}$ has been reached the adjoint problem is initialized; the initial conditions are set to the values produced by the forward problem at time $t = \tau + \widehat{\Delta\mathcal{T}}$. More details on the construction of the adjoint and boundary conditions may be found in Fang et al. (2005).

5 Numerical basis of model

The method outlined in this paper is used in conjunction with the Imperial College Ocean Model, the numerical basis of which is briefly outlined in this section and more fully considered in (Ford et al., 2004; Pain et al., 2005). In the model the three-dimensional nonhydrostatic Boussinesq equations are discretized in the domain

$\Omega \in \mathbb{R}^3$. The equations are as follows:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + 2\boldsymbol{\Omega} \times \boldsymbol{u} = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \mathbf{S}, \tag{41}$$

$$\nabla \cdot \boldsymbol{u} = 0. \tag{42}$$

Where $\boldsymbol{u} = (u, v, w)$ represents the 3D velocity and p is the perturbation pressure. Here u, v and w are the velocity components in the x-, y- and z-directions respectively. The rotation vector is $\boldsymbol{\Omega}$ and takes the form $\boldsymbol{\Omega} = \left(0, 0, \frac{f}{2}\right)^T$, where f is the Coriolis parameter. The viscous terms are represented by the stress tensor $\boldsymbol{\tau}$. More details of the numerical basis of the model can be found in Ford et al. (2004). A functional can be defined in terms of the vorticity, as the problem is two dimensional only one component of the vorticity,

$$\zeta = \left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right), \tag{43}$$

is non-zero. The rationale for basing functionals around vorticity stems from the ability to represent the vorticity numerically. Thus an appropriate functional can be defined as,

$$f(u, v) = \frac{1}{2}\zeta^2. \tag{44}$$

To initialize the adjoint problem (see section 4) the source terms $\frac{\partial F}{\partial \Psi}$ must be calculated. When time stepping is used the functional from section 3 can be defined as total enstrophy at time level $n + L$:

$$F = \frac{1}{2}\int \left(\zeta^{(n+L)}\right)^2 dV = \frac{1}{2}\int \left(\left(\frac{\partial v}{\partial x}\right)^{(n+L)} - \left(\frac{\partial u}{\partial y}\right)^{(n+L)}\right)^2 dV, \tag{45}$$

in which L is the number of time steps used to form the error measure. The aim of using the functional defined by equation 45 is to optimize the accuracy of the functional at the end of time step $\widehat{\Delta \mathcal{T}}$ (time interval between two consecutive mesh adaptations), or taking into account all time levels by using

$$f = \frac{1}{2}\sum_{k=n+1}^{n+L} \left(\zeta^{(k)}\right)^2. \tag{46}$$

Thus

$$\frac{\partial F}{\partial u_i^k} = -\int \frac{\partial N_i}{\partial y}\zeta^k dV, \quad \frac{\partial F}{\partial v_i^k} = \int \frac{\partial N_i}{\partial x}\zeta^k dV, \quad \frac{\partial F}{\partial p_i^k} = 0.$$

Then applying the procedure previously described the solution at each time level between two mesh adaptations from n to $n + L$ is calculated. Then the multi-field

formulation defined previously can be applied with a forward model of:

$$\begin{pmatrix} \mathsf{A}^{(n+1)} & 0 & \cdots & 0 \\ \mathsf{B}^{(n+2)} & \mathsf{A}^{(n+2)} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \mathsf{B}^{(n+L)} & \mathsf{A}^{(n+L)} \end{pmatrix} \begin{pmatrix} \phi^{(n+1)} \\ \phi^{(n+2)} \\ \vdots \\ \phi^{(n+L)} \end{pmatrix} = \begin{pmatrix} f^{(n+1)} - \mathsf{B}^{(n+1)}\phi^{(n)} \\ f^{(n+2)} \\ \vdots \\ f^{(n+L)} \end{pmatrix} \tag{47}$$

in which $\phi^{(n)} = \begin{pmatrix} \underline{u}^{(n)} \\ \underline{v}^{(n)} \\ \underline{p}^{(n)} \end{pmatrix}$ with $\underline{u}^{(n)} = \left(u_1^{(n)}, u_2^{(n)}, \ldots, u_{\mathcal{N}}^{(n)}\right)$, $\underline{v}^{(n)} = \left(v_1^{(n)}, v_2^{(n)}, \ldots, v_{\mathcal{N}}^{(n)}\right)$ and $\underline{p}^{(n)} = \left(p_1^{(n)}, p_2^{(n)}, \ldots, p_{\mathcal{N}}^{(n)}\right)$ contain the nodal velocities and pressures at time level n. $\phi^{*(n)}$ can be defined in a similar way for the adjoint problem where $\phi^{*(n)} = \begin{pmatrix} \underline{u}^{*(n)} \\ \underline{v}^{*(n)} \\ \underline{p}^{*(n)} \end{pmatrix}$. Now assuming the problem is linear, the adjoint equation becomes

$$\begin{pmatrix} \left(\mathsf{A}^{(n+1)}\right)^T & \left(\mathsf{B}^{(n+2)}\right)^T & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \left(\mathsf{A}^{(n+L-1)}\right)^T & \left(\mathsf{B}^{(n+L)}\right)^T \\ 0 & \cdots & 0 & \left(\mathsf{A}^{(n+L)}\right)^T \end{pmatrix} \begin{pmatrix} \phi^{*(n+1)} \\ \vdots \\ \phi^{*(n+L-1)} \\ \phi^{*(n+L)} \end{pmatrix} = \begin{pmatrix} \frac{\partial F}{\partial \phi^{(n+1)}} \\ \frac{\partial F}{\partial \phi^{(n+2)}} \\ \vdots \\ \frac{\partial F}{\partial \phi^{(n+L)}} \end{pmatrix}, \tag{48}$$

where

$$\begin{pmatrix} \frac{\partial F}{\partial \phi^{(n+1)}} \\ \frac{\partial F}{\partial \phi^{(n+2)}} \\ \vdots \\ \frac{\partial F}{\partial \phi^{(n+L)}} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \frac{\partial F}{\partial \phi^{(n+L)}} \end{pmatrix}.$$

In equation 48 it has effectively been assumed that $\phi^{*(n+L+1)} = 0$. There are a number of issues arising from the formulation of these matrices, for example the use of the same projection method to approximately solve the adjoint equations (48) as that used to solve the forward equations (47). This modifies the equations above slightly, but is an approximation to them. This has been the motivation for not necessarily assuming that $r(\psi) = 0$ in the derivation of equation 19. If the time steps for the forward and adjoint calculations used here are large (bearing in mind that the time step will typically be larger than that used to solve the equations once an error measure has been found and the mesh adapted) then a single Picard iteration in time is used for non-linear stability. The equations themselves can become very poorly conditioned for these large time steps if the same discretization method were used

as as that to solve the forward equations ie. Galerkin spatial discretization with Crank-Nicholson time stepping, thus the SUPG method for spatial discretization with backward Euler time-stepping is employed to evaluate equations 47 and 48. For $L = 1$ equations 47 and 48 become (assuming ϕ^n is known from before model is adapted):

$$\mathsf{A}^{(n+L)}\phi^{(n+L)} = -\mathsf{B}^{(n+L)} + f^{(n+L)}, \tag{49}$$

$$\left(\mathsf{A}^{(n+L)}\right)^T \phi^{*(n+L)} = \frac{\partial F}{\partial \phi^{(n+L)}}, \tag{50}$$

and are evaluated in this order. To recover some of the accuracy lost by using only one time step to calculate the error measure the trapezoidal rule is used to calculate ϕ via $\phi = \frac{1}{2}\left(\phi^{(n)} + \phi^{(n+L)}\right)$, and similarly evaluate ϕ^* between mesh adaptations. Our aim is to obtain a mesh which can define the fields at each time step between adapts and thus optimizing the mesh at the mid time level. Thus the equations become,

$$\mathsf{A}^{(n+L)}\phi^{(n+L)} = -\mathsf{B}^{(n+L)}\phi^n + f^{(n+L)}, \tag{51}$$

$$\left(\mathsf{A}^{\left(n+\frac{1}{2}\right)}\right)^T \phi^{*\left(n+\frac{1}{2}\right)} = \frac{\partial F}{\partial \phi^{(n+L)}}, \tag{52}$$

with

$$\Psi = \frac{1}{2}\left(\phi^{(n)} + \phi^{(n+1)}\right) \quad \text{and} \quad \mathsf{A}^T\Psi^* = \left(\mathsf{A}^{\left(n+\frac{1}{2}\right)}\right)^T \phi^{*\left(n+\frac{1}{2}\right)}.$$

These values are used to evaluate the metric tensor, given by equation 29 and the process is shown diagramatically in figure 3.

6 Non-linear 'Munk' problem

Wind driven barotropic circulation provides a good test of this technology. Extending the considerations from the linear problem considered in Munk (1950) to the non-linear version of the problem examined by Ierley (1987) gives a test not only applicable in an 'oceanographic' context, but one where the flow structure is less predictable and exists at a variety of scales. The domain is taken to be a square box of size $1000\,km$ which is $500m$ deep. A maximum zonal wind stress of $\tau_0 = 0.1\,Nm^{-2}$ is applied in a cosine of latitude profile, as described in Marshall et al. (2003). The Coriolis terms are taken into account by $\beta = 1.8 \times 10^{-11} m^{-1}s^{-1}$ and the reference density is $\rho_0 = 1000\,kg\,m^{-3}$.

The problem is non-dimensionalized, so that the domain is a box with sides 1, and depth 0.0005. The domain is maintained at 1 element in the vertical. Incorporating the beta-plane approximation gives a non-dimensional rotation vector of $\mathbf{\Omega}^* = (0, 257.143, 0)$ and non-dimensional wind stress of $\tau_0^* = 163.2653$ The wind stress is applied in a cosine of latitude profile (see Marshall et al. (2003)) and is averaged over the depth of the domain. The non-dimensional time step is taken to be 3.78×10^{-2}, equivalent of 3 hours. No-slip conditions are applied to the lateral

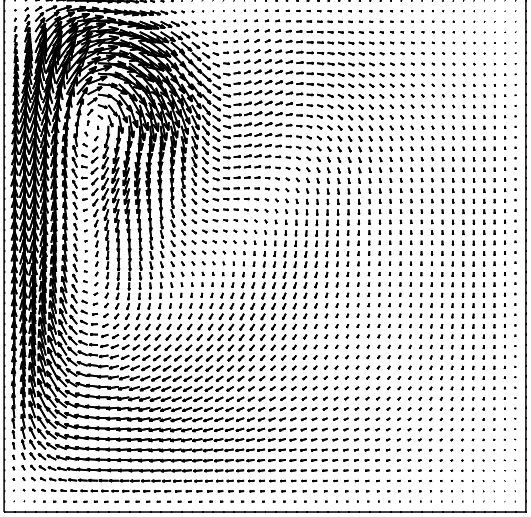

Fig. 4. Initial flow vectors, at day 100.

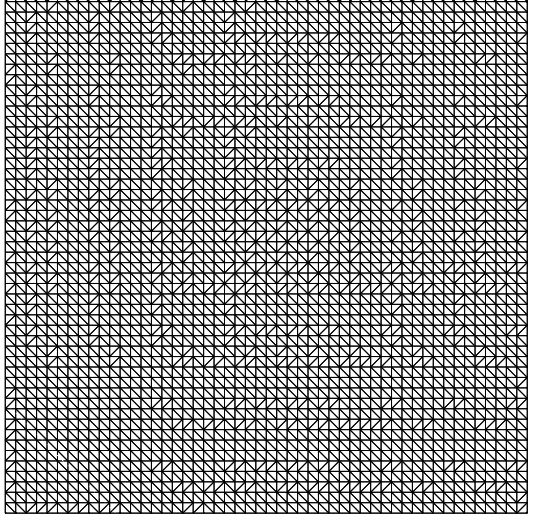

Fig. 5. Initial mesh, at day 100.

boundaries. Two sets of results are presented with Reynolds numbers of 20, ensuring a reasonably non-chaotic regime (section 6.1) and 1000, providing a reasonably chaotic regime (section 6.3).

The functional and source terms are calculated using the solution to the forward problem Ψ, (see section 4 for details), at time level $t = \tau + \widehat{\Delta\mathcal{T}}$, where $\widehat{\Delta\mathcal{T}}$ is taken to be 1.51×10^{-2} or 5 days, which is the time between adapts of the mesh, as discussed in section 4. δF is taken to be 5.0×10^{-1}, with the minimum and maximum edge lengths being 0.001 ($1km$) and 0.1 ($100km$) respectively. The number of nodes is capped at $10,000$.

6.1 Results - $Re = 20$

The adaptive method is employed on this problem from 100 days into the simulation, after most spin-up effects have taken place. The simulation is conducted up to this point on a set mesh of 50×50 elements; the initial velocity vectors, mesh and velocity components are shown in figures 4 to 7 respectively. The forward and adjoint time steps are preformed as detailed in section 4, the adjoint problem being initialized from source terms derived from the differential of the functional F. The u and v components of the forward solution Ψ are shown in figures 8 and 9; with the components u^* and v^* of the adjoint solution Ψ^* shown in figures 10 and 11. The residuals are estimated from Ψ^* by the method described in section 3, resulting in two *residual fields* relating to u^* and v^*. These are shown in figures 12 and 13. The Hessians of the forward solution fields are calculated and averaged and used to form a metric tensor $\boldsymbol{M}$ with which to adapt the mesh. The results shown in figures 8 to 13 relate to the first adaption of the mesh, i.e. the initial large forward time step is between day 100 and day 105 of the simulation, the aim being to optimize the accuracy of the functional at half way through day 102. The adapted mesh, used to compute the actual solution, using time steps of 3 hours between days 100 and 105 is shown in figure 14. The process is repeated when the simulation reaches day 105,

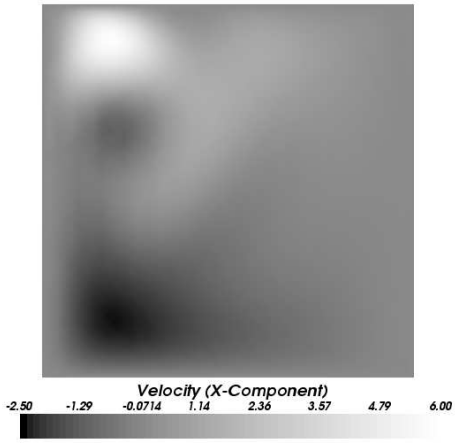

Fig. 6. Initial x-component of velocity.

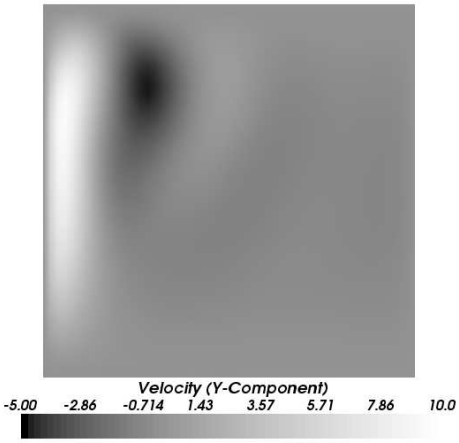

Fig. 7. Initial y-component of velocity.

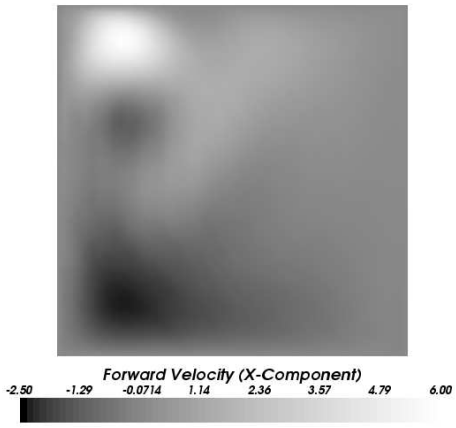

Fig. 8. x-component of forward velocity at end of large timestep $\widehat{\Delta\mathcal{T}}$.

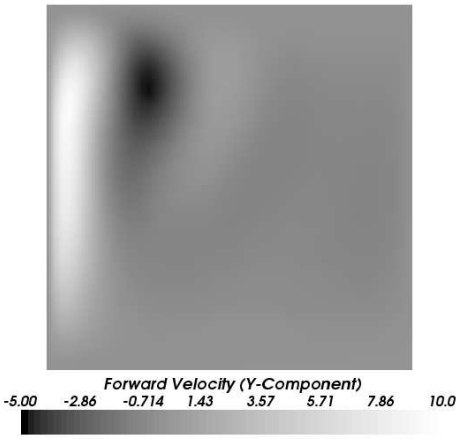

Fig. 9. y-component of forward velocity at end of large timestep $\widehat{\Delta\mathcal{T}}$.

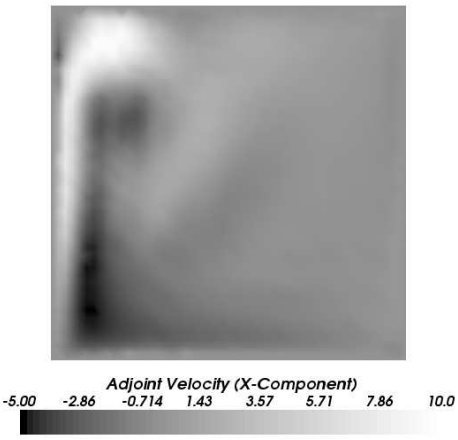

Fig. 10. x-component of adjoint velocity, after adjoint timestep of $\frac{1}{2}\widehat{\Delta\mathcal{T}}$.

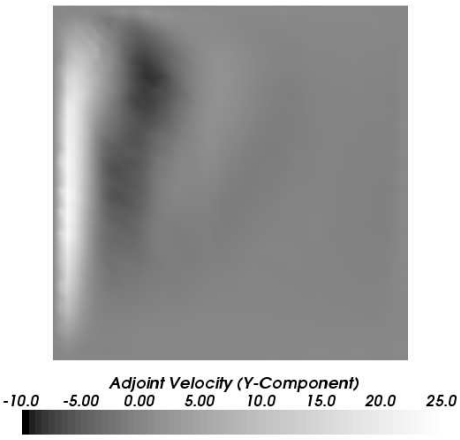

Fig. 11. y-component of adjoint velocity, after adjoint timestep of $\frac{1}{2}\widehat{\Delta\mathcal{T}}$.

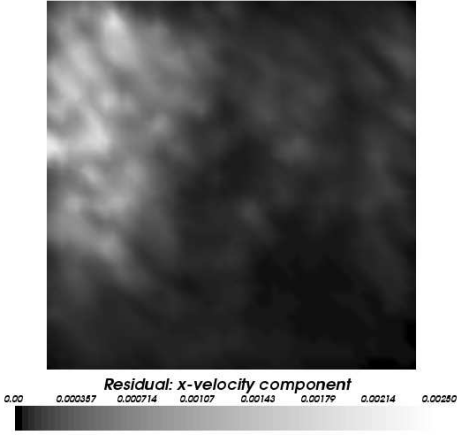

Fig. 12. x-velocity component of residual.

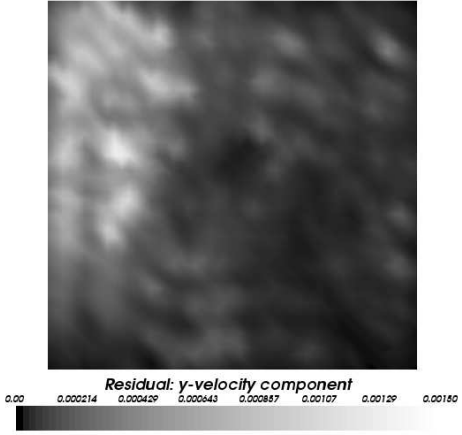

Fig. 13. y-velocity component of residual.

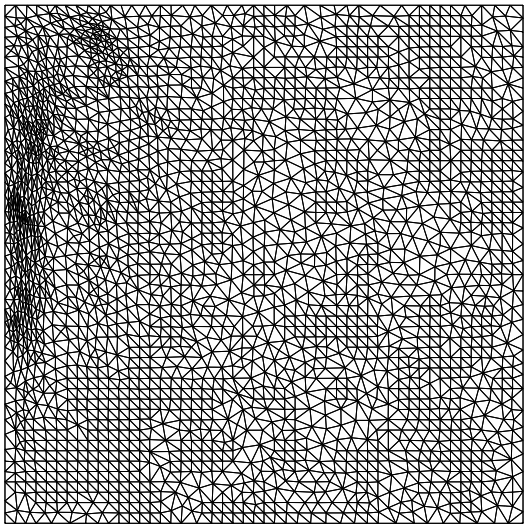

Fig. 14. Mesh adapted to optimize accuracy of functional F at day 102.5. 5023 nodes.

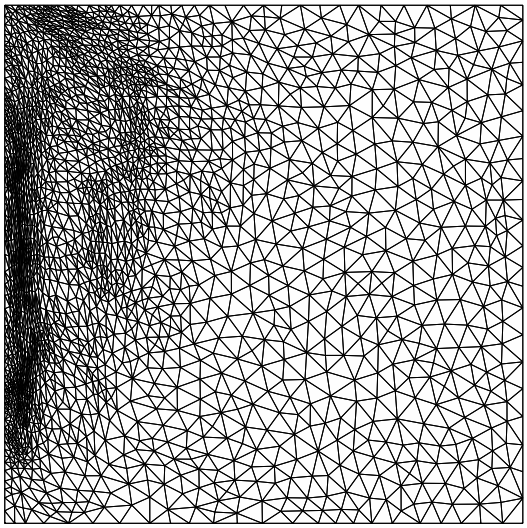

Fig. 15. Mesh adapted to optimize accuracy of functional F at day 107.5. 4701 nodes.

taking a large time step to day 110 and finding a mesh to optimize the accuracy of the functional at half way through day 107. Subsequent adapted meshes are shown in figures 15 to 19.

6.2 Results - $Re = 20$, Old Method

The results in section 6.1 are compared to results obtained by the method described in Pain et al. (2001), where the metric tensor $\boldsymbol{M}$ is defined as:

$$\bar{\boldsymbol{M}} = \frac{\gamma}{\epsilon} |\boldsymbol{H}| \,. \tag{53}$$

Here ϵ is a globally defined value of the interpolation error. Metric tensors based on the x- and y-components of velocity are constructed and then superimposed as previously discussed to create one metric tensor $\boldsymbol{M}$. In the cases presented here

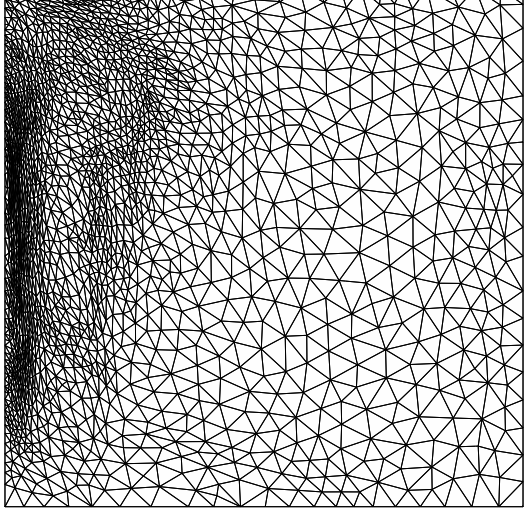

Fig. 16. Mesh adapted to optimize accuracy of functional F at day 112.5. 3701 nodes.

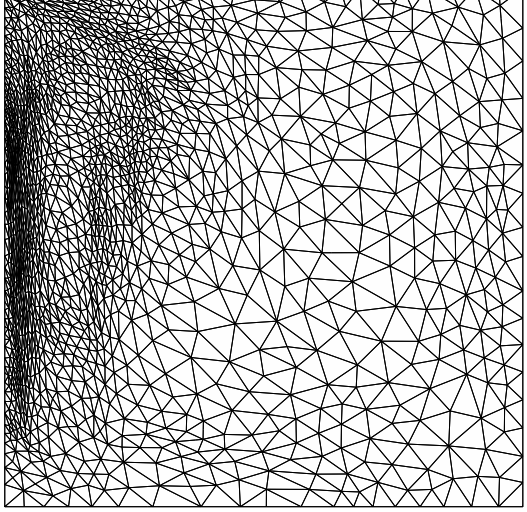

Fig. 17. Mesh adapted to optimize accuracy of functional F at day 117.5. 3004 nodes.

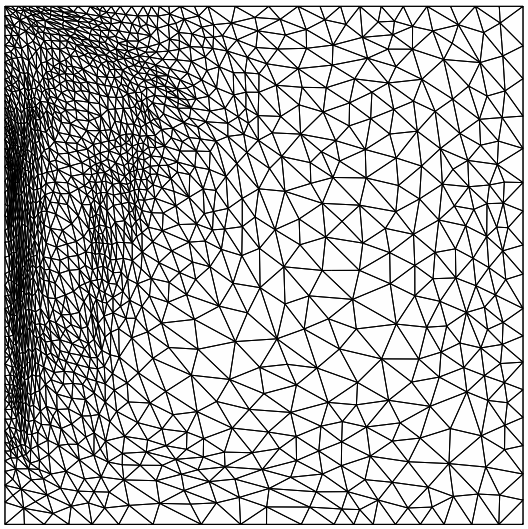

Fig. 18. Mesh adapted to optimize accuracy of functional F at day 122.5. 2543 nodes.

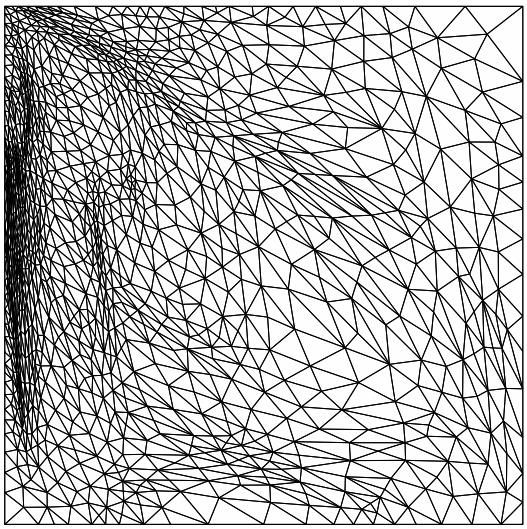

Fig. 19. Mesh adapted to optimize accuracy of functional F at day 147.5. 2133 nodes.

$\epsilon = 0.1$ is used when constructing metric tensors based on u or v; all other parameters are identical to those used in section 6.1.

Figures 20 to 25 show the meshes used for computations at similar times to those in figures 14 to 19, the difference being in the construction of the metric tensors used to adapt the mesh. In this section the metric tensors are based solely on u and v at a single point in time. For example, figure 20 shows the mesh used for computing the solution between days 100 and 105 and is constructed from the values of u and v at day 100; in contrast the mesh shown in 14 (used for computing the solution between the same points in time) is adpated to optimize the accuracy of the functional F half way between the two adapts of the mesh - in this case half way through day

Fig. 20. Mesh used for computing solution between days 100 and 105, with metric tensor as defined by equation 53. ?? nodes.

Fig. 21. Mesh used for computing solution between days 105 and 110, with metric tensor as defined by equation 53. ?? nodes.

Fig. 22. Mesh used for computing solution between days 110 and 115, with metric tensor as defined by equation 53. ?? nodes.

Fig. 23. Mesh used for computing solution between days 115 and 120, with metric tensor as defined by equation 53. ?? nodes.

Fig. 24. Mesh used for computing solution between days 120 and 125, with metric tensor as defined by equation 53. ?? nodes.

Fig. 25. Mesh used for computing solution between days 145 and 150, with metric tensor as defined by equation 53. ?? nodes.

Fig. 26. Mesh adapted to optimize accuracy of functional F at day 102.5. ?? nodes.

Fig. 27. Mesh adapted to optimize accuracy of functional F at day 107.5. ?? nodes.

Fig. 28. Mesh adapted to optimize accuracy of functional F at day 112.5. ?? nodes.

Fig. 29. Mesh adapted to optimize accuracy of functional F at day 117.5. ?? nodes.

102.

6.3 Results - $Re = 1000$

In this section the method is applied to a case with a Reynolds number of 1000, to explore a more chaotic regime. Identical conditions are used as to those in section 6.1. Meshes at equivalent times to those in the two previous sections are shown in figures 26 to 31.

7 Conclusions

A novel method of defining a metric tensor field for giving the size and shape of elements in adaptive mesh modelling of oceanographic phenomena has been presented. A goal-based technique has been used to highlight areas of the domain where increased (or decreased) mesh resolution is desirable. The goal and measure of the dynamics of the two-dimensional gyre problem presented here is to optimize the accuracy of the total enstrophy (integral over the solution domain of vorticity squared). The method employs both forward and adjoint computations to calculate a sensitivity, thereby highlighting locations of the domain that warrant increased resolution. The specified level of acceptable error for each variable is set by the method, rather than a constant global value being defined by the user. This allows the specified level of acceptable error to be varied automatically in time and space, according to the requirements of the problem.

Fig. 30. Mesh adapted to optimize accuracy of functional F at day 122.5. ?? nodes.

Fig. 31. Mesh adapted to optimize accuracy of functional F at day 147.5. ?? nodes.

The methodology behind this work will become particularly important when considering turbulent flows, where very fine resolution may be desired in a majority of the domain, but where a restriction in computational overheads exists. In cases such as these, a method which highlights more important areas to be considered will be an important tool in resolving the problem accurately with the given resources. Future work will involve the testing of the method on an extended set of benchmarks; the application of a variety of functionals or goals.

Acknowledgements

NERC David's sponsors

References

Ainsworth, M., Oden, J. T., 1997. A posteriori error estimation in finite element analysis. Comput. Methods Appl. Mech. Engrg. 142, 1–88.

Alekseev, A., Navon, I., 2005. A *posteriori* pointwise error estimation for compressible fluid flows using adjoint parameters and lagrange remainder. Int. J. Num. Meth. Fluids 47, 45–74.

Cacuci, D., 1981. Sensitivty theory for non-linear systems. I: Nonlinear function analysis approach. J. Math. Phys. 22, 2794–2802.

Cacuci, D., 1988. The foward and adjoint methods of sensitivity analysis. CRC Press, Inc., Ch. 3, pp. 71–144.

Castro-Diaz, M., Hecht, F., Mohammadi, B., Pironneau, O., 1997. Anisotropic unstructured mesh adaption for flow simulations. Int. J. Num. Meth. Fluids 25, 475–491.

Cirak, F., Ramm, E., 1998. a-posteriori error estimation and adaptivity for linear elasticity using the reciprocal theorem. Comput. Meth. Appl. Mech. Eng. 156, 351–362.

Fang, F., Piggott, M., Pain, C., Gorman, G., Goddard, A., 2005. An adaptive mesh adjoint data assimilation method for coastal flows, in preparation for Ocean Modelling.

Ford, R., Pain, C., Piggott, M., Goddard, A., de Oliveira, C. R., Umpleby, A. P., 2004. A non-hydrostatic finite element model for three-dimensional stratified ocean flows. part i: Model formulation. Monthly Weather Rev. 132, 2816–2831.

Freitag, L., Ollivier-Gooch, C., 1997. Tetrahedral mesh improvement using swapping and smoothing. Int. J. Num. Meth. Eng. 40, 3979–4002.

George, P., 1998. Delaunay Triangulation and Meshing: Application to Finite Elements. HERMES, Paris.

Gunzburger, M., 1999. Sensitivities, adjoints and flow optimization. Int. J. Num. Meth. Fluids 31, 53–78.

Hide, R., 1989. Superhelicity, helicity and potential vorticity. Geophys. Astrophys. Fluid Dyn. 48, 69–79.

Hide, R., 2002. Helicity, supermelicity and weighted relative potential vorticity: Useful disgnostic pseudoscalars? Q. J. R. Meteorol. Soc. 128, 1759–1762.

Homescu, C., Navon, I., 2003. Numerical and theoretical considerations for sensitivity calculation of discontinuous flow. Systems and Control Letters 48, 253–260.

Ierley, G., 1987. On the onset of recirculation in barotropic general circulation models. J. Phys. Oceanogr. 17, 2366–2374.

Joe, B., 1989. Three-dimensional triangulations from local transformations. SIAM J. Sci. Stat. Comput. 10, 718–741.

Knupp, P., 2000. Achieving finite element mesh quality via optimization of the jacobian matrix norm and associated quantities. part ii - a framework for volume mesh optimisation and the condition number of the jacovian matrix. Int. J. Num. Meth. Eng. 48, 1165–1185.

Langland, R., Elsberry, R., Errico, R., 1995. Evaluation of physical processes in an idealized extratropical cyclone using adjoint sensitivity. Q. J. R. Meteorol. Soc. 121, 1349–1386.

Lohner, R., Morgan, K., Zienkiewicz, O., 1985. An adaptive finite element procedure for compressible high speed flows. Comput. Meth. Appl. Mech. Eng. 51, 441–465.

Marshall, D., Pain, C., Piggott, M., Power, P., Gorman, G., Umpleby, A., de Oliveira, C., 2003. Application of unstructured adaptive finite element methods to the wind driven circulation in a barotropic ocean basin., in preparation.

Müller, J.-D., Giles, M., June 2001. Solution adaptive mesh refinement using adjoint error analysis. 15th Computational Fluid Dynamics Conference, American Institute of Aeronautics and Astronautics.

Munk, W., 1950. On the wind-driven ocean circulation. J. Meteorol. 7, 79–93.

Oden, J., Prudhomme, S., 2001. Goal-oriented error estimation and adaptivity for the finite element method. Comp. & Math. with Applic. 41, 735–756.

Pain, C., Piggott, M., Goddard, A., Fang, F., Gorman, G., Marshall, D., Eaton, M., Power, P., de Oliveira, C., 2005. Three-dimensional unstructured mesh ocean modelling. Ocean Modelling 10, 5–33.

Pain, C., Umpleby, A., de Oliveira, C., Goddard, A., 2001. Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations. Comput. Meth. Appl. Mech. Eng. 190, 3771–3796.

Paraschivoiu, M., Patera, A., 1998. A hierarchical duality approach to bounds for the outputs of partial differential equations. Comput. Meth. Appl. Mech. Eng. 158, 389–407.

Peraire, J., Vahdati, M., Morgan, K., Zienkiewicz, O., 1987. Adaptive remeshing for compressible flow computations. J. Comput. Phys. 72, 449–466.

Piasecki, M., Katopodes, N., 1997. Control of contaminant releases in rivers. I: adjoint sensitivity analysis. J. Hydr. Engrg. 123, 488–492.

Pierce, N., Giles, M., 2000. Adjoint recovery of superconvergent functionals from pde approximations. SIAM Review 42, 247–264.

Piggott, M., Pain, C. C., Gorman, G., Power, P., Goddard, A., 2005. *h, r* and *hr* adaptivity with applications in numerical ocean modelling. Ocean Modelling 10, 95–113.

Prudhomme, S., Oden, J., 1999. On goal-oriented error estimation for elliptic prob-

lems. Comput. Meth. Appl. Mech. Eng. 179, 313–351.
Sanders, B., Katopodes, N., 1999. Control of canal flow by adjoint sensitivity method. J. Irrigation Drainage Eng. 125, 287–297.
Sanders, B., Katopodes, N., 2000. Adjoint sensitivity analysis for shallow water wave control. J. Eng. Mech. 126, 909–919.
Strouboulis, T., Oden, J. T., 1990. A posteriori error estimation of finite element approximations in fluid mechanics. Comput. Meth. Appl. Mech. Eng. 78, 201–242.
Venditti, D., Darmofal, D., 2000. Adjoint error estimation and grid adaption for functional outputs: Application to quasi-one-dimensional flow. J. Comput. Phys. 164, 204.
Venditti, D., Darmofal, D., 2002. Grid adaption for functional outputs: Application to two-dimensional inviscid flows. J. Comput. Phys. 176, 40–69.
Venditti, D., Darmofal, D., 2003. Anisotropic grid adaption for functional outputs: application to two dimensional viscous flows. J. Comput. Phys. 187, 22–46.
Wu, J., Zhu, J., Szmelter, J., Zienkiewicz, O., 1990. Error estimation and adaptivity in navier-stokes incompressible flows. Comput. Mech. 6, 259–270.
Zhang, S., Zou, X., Ahlquist, J., Navon, I., Sela, J., 2000. Use of differentiable and non-differentiable optimization algorithms for variational data assimilation with discontinuous cost functions. Monthly Weather Rev. 128, 4031–4044.
Zienkiewicz, O., Taylor, R., 1991. The finite element method, 4th Edition. Vol. 2. McGraw-Hill, New York.