Available online at www.sciencedirect.com

ScienceDirect

Ocean Modelling 15 (2006) 3–38

Ocean Modelling

www.elsevier.com/locate/ocemod

Adjoint goal-based error norms for adaptive mesh ocean modelling

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

[a] *Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College London, Prince Consort Road, London SW7 2BP, UK*

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

[c] *School of Computational Science and Department of Mathematics, Florida State University, Tallahassee 32306-4120, FL, USA*

Received 4 July 2005; received in revised form 12 May 2006; accepted 15 May 2006
Available online 16 June 2006

Abstract

Flow in the world's oceans occurs at a wide range of spatial scales, from a fraction of a metre up to many thousands of kilometers. In particular, regions of intense flow are often highly localised, for example, western boundary currents, equatorial jets, overflows and convective plumes. 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.

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 horizontal scale of order 10 km, and tall isolated seamounts, which have a horizontal scale of order 20 km. Western boundary currents have a typical width of 50 km and are strongly influenced by topography, whereas convective plumes have a scale of the order of 100 m. On the other hand, basin-scale gyre circulations typically occur on scales of the order of 1000 km. These differences in

[*] Corresponding author. Tel.: +44 20 75949322; fax: +44 20 75947444.
E-mail address: m.d.piggott@imperial.ac.uk (M.D. Piggott).
URL: http://amcg.ese.imperial.ac.uk (M.D. Piggott).

1463-5003/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ocemod.2006.05.001

scale produce challenges in numerical modelling. At present, no numerical ocean models are able to routinely resolve simultaneously the full range of pertinent spatial scales. Conventional numerical ocean models generally use uniformly spaced finite-differences to discretise the governing equations. The finite-element model described in Ford et al. (2004a) 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 utilise 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 the 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; Löhner et al., 1985; Piggott et al., 2005) calculate an 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 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 or local geographic area would possess.

The Imperial College Ocean Model (ICOM) (Ford et al., 2004a,b; Pain et al., 2005) utilises 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 optimisation) to adapt the mesh, changing the size, shape and location of tetrahedral elements to optimise 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, its 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 minimise 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 error measure utilised 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; Langland et al., 1995). Theoretical considerations have been presented by Homescu and Navon (2003) and 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 optimise 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 optimise 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. (2004a) and 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, 1999b,a; Cirak and Ramm, 1998; Venditti and Darmofal, 2000) is the use of readily accessible (in simulation codes) discretised 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/optimisation procedure. Since the adjoint solution is typically calculated in data assimilation, control, and optimisation 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. Sections 3 and 4 outline the methodology for the goal-based method and the creation of a metric tensor. Section 5 outlines how the method is implemented for ocean modelling problems. Section 6 briefly outlines the model equations which are then utilised to solve a wind driven rotating gyre problem, using the goal-based technology, in Section 7. In Section 8, some conclusions are drawn.

2. An anisotropic mesh adaptivity method

For completeness a brief overview of the mesh optimisation 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 optimisation 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 ith 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 optimisation via $\|v\| = v^{\mathrm{T}} \boldsymbol{M}_v v$, where $\boldsymbol{M}_v$ is the average metric tensor along vector v. This approach may be viewed as a discretisation 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 optimisation 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 minimisation problem being defined the aim is to find a local configuration that reduces the functional value. Defining the mesh functional as

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

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

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

Here r_ℓ is the length, with respect to the edge centred metric tensor, $\boldsymbol{M}_\ell$, of edge ℓ; $\mathscr{L}_e$ 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 as an equilateral tetrahedron with sides of unit length when measured in metric space (and hence may still be highly anisotropic when viewed in physical space).

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).

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 the calculation of a new metric tensor $\boldsymbol{M}$, based upon both forward and adjoint calculations. Calculating $\boldsymbol{M}$ is the focus of this paper and while the theoretical development is presented in this section, the overall construction of a practical and usable metric will be expanded upon in this and the following three sections. The overall approach has five main steps which may be summarised as follows:

- take a single large time step forward with the model to anticipate the forward solution (see Sections 5 and 6);
- calculate the differential of the functional encoding what is important in the simulation and place this on the right-hand side of the adjoint problem (see Section 6);
- take a large time step backwards (half the forward time step) by solving the associated adjoint problem (see Section 5);
- calculate the residual of the forward and/or the adjoint equations using a multi-grid like method (see Sections 3.2.3 and 4.2);
- convolve this residual with the Hessians of the forward and/or adjoint solution fields to obtain the final metric (Sections 3.2.2 and 3.2.3) which yield the error measures to be used by the mesh optimisation algorithm.

Following the introduction of a differential problem (3) and associated continuous (5) and discrete residuals (6), the following subsections construct a measure of the error in a functional or goal (7). This functional would typically be chosen such that it encodes in some sense what is important in the simulation being undertaken. The associated adjoint problem is introduced in Section 3.1.2 and is used to estimate the error in the numerical estimate of the functional. This then leads to the two expressions for the error, (18) and (20), which are used in the remainder of this work. Section 3.2 assumes that the unknown quantities in these expressions have been estimated (with details in Section 4) and subsequently constructs two new error metric tensors (28) and (31).

3.1. A goal-based error measure

Suppose a differential equation to be solved is

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

for source s, linear operator $\mathscr{L}$ (the extension to non-linear operators is relatively straightforward although it can involve considerable algebra), dependent solution variable ψ (in this work typically momenta, temperature, etc.), and the exact solution is $\psi_{\text{exact}} \equiv \psi_{\text{exact}}(\mathbf{x})$. This solution is approximated with a finite element numerical expansion

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

with $\mathbf{x}$ being the spatial coordinates, $\boldsymbol{\Psi} \equiv (\Psi_1, \Psi_2, \ldots, \Psi_{\mathscr{N}})^{\mathrm{T}}$ the discrete solution vector, $N_j(\mathbf{x})$, $j = 1, \ldots, \mathscr{N}$ the (piecewise linear in this work) finite element basis functions, and $\mathscr{N}$ 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 (5) by the weighting functions and integrating over the domain. In the Bubnov–Galerkin method these weighting functions are chosen to be the basis functions $N_i(\mathbf{x})$, $i = 1, \ldots, \mathcal{N}$. Whatever method is chosen to discretise the equations a matrix equation and residual vector result

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

for matrix $\mathbf{A}$ and discretised 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 derivations. In addition, note that in the analysis it is assumed that $r(\psi)$ is in some sense a discretised representation of the residual $\mathcal{R}(\psi)$ 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 optimised is represented as $F \equiv F(\psi)$, and

$$F(\psi) = \int_{\Omega} f(\psi)\,\mathrm{d}V, \tag{7}$$

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 $\partial f/\partial\psi$ near the exact solution ψ_{exact} can be obtained from

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

or in discrete form,

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

in which $\Psi_{\mathrm{exact}} \equiv (\Psi_{\mathrm{exact}_1}, \Psi_{\mathrm{exact}_2}, \ldots, \Psi_{\mathrm{exact}_{\mathcal{N}}})^{\mathrm{T}}$ is a vector containing the exact solution at the $\mathcal{N}$ finite element nodes (or control volume cells) and

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

That is, $\widetilde{\psi}_{\mathrm{exact}}$ 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 $\mathbf{A}_{\mathrm{exact}}$ exists such that

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

where Ψ_{exact} is the exact solution at the nodes for a given discretised source vector S. Note that via a probing method (taking S to be the unit vectors) the columns of $\mathbf{A}^{-1}_{\mathrm{exact}}$ may be constructed directly assuming the corresponding exact solution Ψ_{exact} is known. In practice of course $\mathbf{A}_{\mathrm{exact}}$ and Ψ_{exact} are not available, the procedures for their estimation are given in Section 4. Now define the adjoint solutions Ψ^* and Ψ^*_{exact} in terms of the adjoint problems

$$\mathbf{A}^{\mathrm{T}}\Psi^* - \frac{\partial F}{\partial \Psi} = 0,$$

and

$$\mathbf{A}^{\mathrm{T}}_{\mathrm{exact}}\Psi^*_{\mathrm{exact}} - \frac{\partial F}{\partial \Psi} = 0. \tag{12}$$

Again, via a first order Taylor series expansion,

$$\frac{\partial(r_{\mathrm{exact}}(\widetilde{\psi}_{\mathrm{exact}}))}{\partial \Psi}(\Psi_{\mathrm{exact}} - \Psi) \approx r_{\mathrm{exact}}(\widetilde{\psi}_{\mathrm{exact}}) - r_{\mathrm{exact}}(\psi) \tag{13}$$

$$\approx r(\widetilde{\psi}_{\mathrm{exact}}) - r(\psi), \tag{14}$$

which with (11) and (6) yields

$$(\boldsymbol{\Psi}_{\text{exact}} - \boldsymbol{\Psi}) \approx (\mathbf{A}_{\text{exact}})^{-1}(r(\widetilde{\psi}_{\text{exact}}) - r(\psi)). \tag{15}$$

This can then be combined with (9) and (12) to obtain

$$\begin{aligned} F(\widetilde{\psi}_{\text{exact}}) - F(\psi) &\approx \left(\frac{\partial F}{\partial \boldsymbol{\Psi}}\right)^{\mathrm{T}} (\mathbf{A}_{\text{exact}})^{-1}(r(\widetilde{\psi}_{\text{exact}}) - r(\psi)) = (r(\widetilde{\psi}_{\text{exact}}) - r(\psi))^{\mathrm{T}}(\mathbf{A}_{\text{exact}})^{-\mathrm{T}}\left(\frac{\partial F}{\partial \boldsymbol{\Psi}}\right) \\ &= (r(\widetilde{\psi}_{\text{exact}}) - r(\psi))^{\mathrm{T}}\boldsymbol{\Psi}^*_{\text{exact}}. \end{aligned} \tag{16}$$

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

3.1.3. Discrete functional correction

Although the error estimates are obtained from (16), the product $(r(\widetilde{\psi}_{\text{exact}}) - r(\psi))^{\mathrm{T}}\boldsymbol{\Psi}^*_{\text{exact}}$ is not known and cannot be used directly to correct the functional F. To do this the following relation is employed:

$$F(\widetilde{\psi}_{\text{exact}}) - F(\psi) \approx (r(\widetilde{\psi}_{\text{exact}}) - r(\psi))^{\mathrm{T}}\boldsymbol{\Psi}^* \approx (\hat{r}(\psi) - r(\psi))^{\mathrm{T}}\boldsymbol{\Psi}^*. \tag{17}$$

$\hat{r}(\psi)$ is approximately equal to $r(\widetilde{\psi}_{\text{exact}})$ and is obtained from the residual calculation method outlined in Section 4. Eq. (17) 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 (16),

$$\begin{aligned} F(\psi_{\text{exact}}) - F(\psi) &\approx (r(\widetilde{\psi}_{\text{exact}}) - r(\psi))^{\mathrm{T}}\boldsymbol{\Psi}^*_{\text{exact}} = ((\mathbf{A}\boldsymbol{\Psi}_{\text{exact}} - S) - (\mathbf{A}\boldsymbol{\Psi} - S))^{\mathrm{T}}\boldsymbol{\Psi}^*_{\text{exact}} \\ &= (\mathbf{A}(\boldsymbol{\Psi}_{\text{exact}} - \boldsymbol{\Psi}))^{\mathrm{T}}\boldsymbol{\Psi}^*_{\text{exact}} = \left(\mathbf{A}^{\mathrm{T}}\boldsymbol{\Psi}^*_{\text{exact}}\right)^{\mathrm{T}}(\boldsymbol{\Psi}_{\text{exact}} - \boldsymbol{\Psi}). \end{aligned} \tag{18}$$

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

3.1.5. Alternative error measures

The dual error measure derived in Section 3.1.4 can be written, taking into account (12) and (15) and assuming $r(\psi) = 0$, as

$$F(\psi_{\text{exact}}) - F(\psi) \approx r(\widetilde{\psi}_{\text{exact}})^{\mathrm{T}}\boldsymbol{\Psi}^*_{\text{exact}} = r(\widetilde{\psi}_{\text{exact}})^{\mathrm{T}}\boldsymbol{\Psi}^* + r(\widetilde{\psi}_{\text{exact}})^{\mathrm{T}}\left(\boldsymbol{\Psi}^*_{\text{exact}} - \boldsymbol{\Psi}^*\right). \tag{19}$$

From (19) a useful relationship which has an analogous continuous form is

$$F(\psi_{\text{exact}}) - F(\psi) \approx \left(\frac{\partial F}{\partial \psi}\right)^{\mathrm{T}}(\boldsymbol{\Psi}_{\text{exact}} - \boldsymbol{\Psi}) + r(\widetilde{\psi}_{\text{exact}})^{\mathrm{T}}\left(\boldsymbol{\Psi}^*_{\text{exact}} - \boldsymbol{\Psi}^*\right). \tag{20}$$

Eq. (20) is used to form a directional error measure based on the adjoint solution. Similar results to (19) have been used by Venditti and Darmofal (2000), in which the latter part of (20), that is $r(\widetilde{\psi}_{\text{exact}})^{\mathrm{T}}(\boldsymbol{\Psi}_{\text{exact}} - \boldsymbol{\Psi})$, 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 (19) can also be written for linear operators as

$$\begin{aligned} F(\psi_{\text{exact}}) - F(\psi) &\approx \left(\mathbf{A}^{\mathrm{T}}\left(\boldsymbol{\Psi}^*_{\text{exact}} - \boldsymbol{\Psi}^* + \mathbf{A}^{-\mathrm{T}}\frac{\partial F}{\partial \psi}\right)\right)^{\mathrm{T}}(\boldsymbol{\Psi}_{\text{exact}} - \boldsymbol{\Psi}) \\ &= \left(\mathbf{A}^{\mathrm{T}}\left(\boldsymbol{\Psi}^*_{\text{exact}} - \boldsymbol{\Psi}^*\right) + \frac{\partial F}{\partial \psi}\right)^{\mathrm{T}}(\boldsymbol{\Psi}_{\text{exact}} - \boldsymbol{\Psi}) \\ &= \left(\boldsymbol{\Psi}^*_{\text{exact}} - \boldsymbol{\Psi}^*\right)^{\mathrm{T}}\mathbf{A}(\boldsymbol{\Psi}_{\text{exact}} - \boldsymbol{\Psi}) + \left(\frac{\partial F}{\partial \psi}\right)^{\mathrm{T}}(\boldsymbol{\Psi}_{\text{exact}} - \boldsymbol{\Psi}). \end{aligned} \tag{21}$$

Estimates of the solution errors $(\Psi^{*} - \Psi^{*}_{\text{exact}})$ and $(\Psi - \Psi_{\text{exact}})$ 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 (18) and (20) were derived. In this section these error measures are modified slightly by approximating the unknown quantities $\mathbf{A}^{\mathrm{T}}\Psi^{*}_{\text{exact}}$ and $r(\widetilde{\psi}_{\text{exact}})$ with known quantities $\widehat{\mathbf{A}^{\mathrm{T}}\Psi^{*}_{\text{exact}}}$ and $\hat{r}(\widetilde{\psi}_{\text{exact}})$ respectively. Full details of how this is implemented are outlined in Section 4. The error measures (18) and (20) then become,

$$F(\psi_{\text{exact}}) - F(\psi) \approx (\widehat{\mathbf{A}^{\mathrm{T}}\Psi^{*}_{\text{exact}}})^{\mathrm{T}}(\Psi_{\text{exact}} - \Psi), \tag{22}$$

$$F(\psi_{\text{exact}}) - F(\psi) \approx \left(\frac{\partial F}{\partial \psi}\right)^{\mathrm{T}}(\Psi_{\text{exact}} - \Psi) + \hat{r}(\widetilde{\psi}_{\text{exact}})^{\mathrm{T}}(\Psi^{*}_{\text{exact}} - \Psi^{*}). \tag{23}$$

These equations require a measure of the error in the forward solution (22) and both forward and adjoint solutions (23).

3.2.1. Defining a metric tensor

A metric tensor (see Section 2) can be defined

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

Here $\boldsymbol{H}$ is the Hessian matrix with $|\boldsymbol{H}|$ its symmetric positive-definite absolute value, ϵ is the required level of error and γ is a scalar constant. The Hessian here is defined in terms of both the forward solution ψ and the adjoint solution ψ^{*}. The metric tensor is defined such that an element size (length) is unity if it has the desired interpolation error ϵ. The discrete (nodal) form of ϵ, from (22) is

$$\epsilon_i = \frac{\widetilde{\delta F}}{\left|\left(\widehat{\mathbf{A}^{\mathrm{T}}\Psi^{*}_{\text{exact}}}\right)_i\right|}. \tag{24}$$

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

$$\widetilde{\delta F} = \frac{\delta F}{\mathcal{N}}. \tag{25}$$

3.2.2. Application to multiple field problems

For each solution variable l and each node i one can define two Hessians: $\boldsymbol{H}^{l}_{i}$ 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

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

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

$$\left.\frac{\partial \psi^{l}}{\partial x}\right|_i \approx q_{x_i} = \mathsf{M}_{\mathsf{L}}^{-1}{}_{i}\int N_i\left(\frac{\partial \psi^{l}}{\partial x}\right)\mathrm{d}V,$$

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 M_{L} is the row summed lumped mass matrix (see Zienkiewicz and Taylor, 2000). 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

$$q^l_{xx_i} = \mathsf{M}^{-1}_{\mathsf{L}_i} \int N_i \frac{\partial q^l_x}{\partial x}\,\mathrm{d}V, \quad q^l_{xy_i} = \mathsf{M}^{-1}_{\mathsf{L}_i} \int N_i \frac{\partial q^l_x}{\partial y}\,\mathrm{d}V.$$

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

$$\overline{\boldsymbol{H}}_i = \frac{1}{\sum_{l=1}^{\mathscr{M}} |\lambda_i^l|} \sum_{l=1}^{\mathscr{M}} |\lambda_i^l||\boldsymbol{H}_i^l|, \tag{26}$$

where

$$\lambda_i^l = (\mathbf{A}^{\mathrm{T}}\widehat{\boldsymbol{\Psi}^*_{\mathrm{exact}}})_i^l.$$

This technique uses (22) as the basis for this average. Using the error measure defined by (22) the metric tensor field can be found. An interpolation error $\bar{\epsilon}_i$ at node i can be defined as

$$\bar{\epsilon}_i = \frac{\widetilde{\delta F}}{\sum_{l=1}^{\mathscr{M}} |\lambda_i^l|}, \tag{27}$$

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

$$\overline{\boldsymbol{M}}_i = \frac{\gamma}{|\bar{\epsilon}_i|}|\overline{\boldsymbol{H}}_i|. \tag{28}$$

This is the metric tensor used in the applications (see Section 7). Using the adjoint error measure defined by (23) a second metric tensor can be obtained. To this end suppose that

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

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

$$\overline{\boldsymbol{H}}_i^* = \frac{1}{\sum_{l=1}^{\mathscr{M}} (|\lambda^{+l}_{\ i}| + |\lambda^{*l}_{\ i}|)} \sum_{l=1}^{\mathscr{M}} (|\lambda^{+l}_{\ i}||\boldsymbol{H}_i^l| + |\lambda^{*l}_{\ i}||\boldsymbol{H}^{*l}_{\ i}|). \tag{29}$$

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

$$\bar{\epsilon}_i^* = \frac{\widetilde{\delta F}}{\sum_{l=1}^{\mathscr{M}} (|\lambda^{+l}_{\ i}| + |\lambda^{*l}_{\ i}|)}. \tag{30}$$

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

$$\overline{\boldsymbol{M}}_i^* = \frac{\gamma}{|\bar{\epsilon}_i^*|}|\overline{\boldsymbol{H}}_i^*|. \tag{31}$$

3.2.3. Superposition of metric tensors

In practice, 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 $\overline{\boldsymbol{M}}^{*}$ and using only $\overline{\boldsymbol{M}}$ as the basis of mesh adaptivity, thus the overall metric tensor $\boldsymbol{M}_i$ at node i is obtained from

$$\boldsymbol{M}_i = \overline{\boldsymbol{M}}_i, \tag{32}$$

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

$$\boldsymbol{M}_i = \mathscr{G}(\overline{\boldsymbol{M}}_i, \overline{\boldsymbol{M}}_i^{*}). \tag{33}$$

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

4. Equation residuals

4.1. Determining the residual

In this section approximations $\widehat{\mathbf{A}^{\mathrm{T}}\Psi^{*}_{\mathrm{exact}}}$ and $\hat{r}(\widetilde{\psi}_{\mathrm{exact}})$ to $\mathbf{A}^{\mathrm{T}}\Psi^{*}_{\mathrm{exact}}$ and $r(\widetilde{\psi}_{\mathrm{exact}})$ 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 algebraically, 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 node i; these nodes form the set $\mathscr{S}_i$ and are shown as filled circles in Fig. 1. The fine grid stencil is indicated by solid bold lines in Fig. 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 $j \in \mathscr{S}_i$ are defined as

$$\mathbf{x}_c = \mathbf{x}_i + 2(\mathbf{x}_j - \mathbf{x}_i). \tag{34}$$

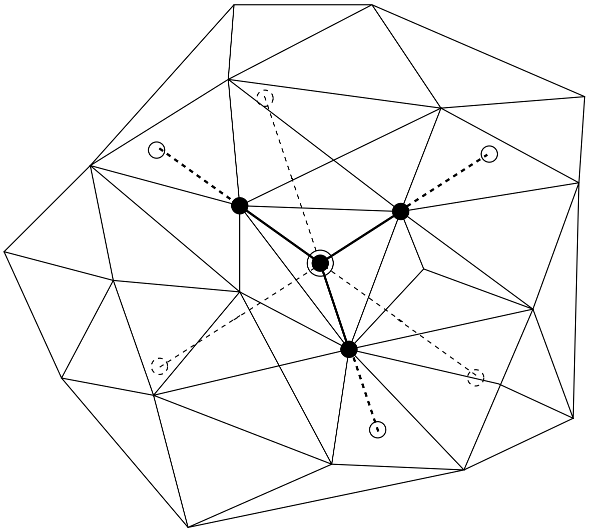

Fig. 1. Diagram showing how the coarse grid is obtained for an unstructured mesh. Filled circles represent fine grid nodes (Ψ_j); solid open circles indicate coarse grid nodes (Ψ_c) and dashed open circles represent the extra points required for interpolation (Ψ_d).

These points are shown with solid open circles in Fig. 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 $j \in \mathscr{S}_i$ are defined as

$$\mathbf{x}_d = \mathbf{x}_i - 2(\mathbf{x}_j - \mathbf{x}_i). \tag{35}$$

These points are shown with dashed open circles in Fig. 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 $j \in \mathscr{S}_i$. 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 Ψ^{*l}_c and Ψ^{*l}_d. 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

$$\Psi^{*}_{\mathrm{exact}}{}^{l}_{j} = \frac{-\Psi^{*l}{}_d + 6\Psi^{*l}{}_i + 3\Psi^{*l}{}_c}{8}. \tag{36}$$

This process is repeated for each of the nodes j surrounding node i; the value at node i, the centre of the discretisation, is taken to be Ψ^{*l}_i. Fig. 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 Fig. 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

$$\Psi^{*}_{\mathrm{exact}}{}^{l}_{j} = \Psi^{*l}_{j}. \tag{37}$$

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

$$\Psi^{*}_{\mathrm{exact}}{}^{l}_{j} = \frac{\Psi^{*l}_j + \Psi^{*l}_c}{2}. \tag{38}$$

The interpolated values of Ψ^* at the nodes surrounding node i are enough to compute the ith row of $\mathbf{A}^{\mathrm{T}}\widehat{\Psi^*_{\mathrm{exact}}}$, required by (24). It is sufficient to take the interpolated values of Ψ^* in this case as the values of Ψ^*_{exact}.

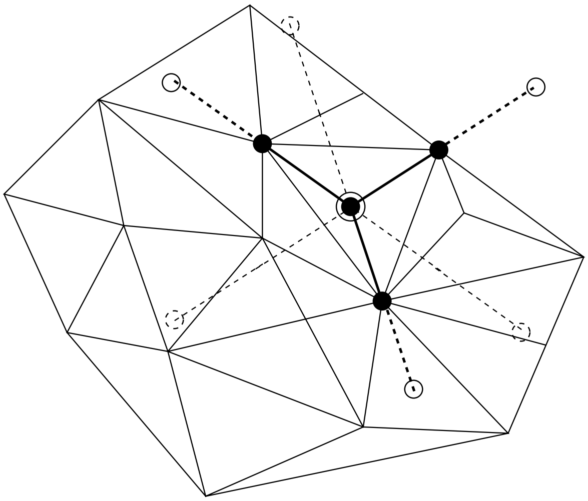

Fig. 2. Diagram showing how the coarse grid is obtained for an unstructured grid near to the domain boundary. Definitions identical to those given by Fig. 1. Note that it is possible to extend the stencil beyond the domain boundary, the methodology for dealing with this is detailed in Section 4.

4.2. Smoothing the residual

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 been calculated, $\widehat{\mathbf{A}^{\mathrm{T}}\boldsymbol{\Psi}^{*}_{\text{exact}}}$ or $\hat{r}(\widetilde{\psi}_{\text{exact}})$. 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,

$$\hat{r}_i^{\mathsf{S}} = \frac{1}{2}\hat{r}_i + \frac{1}{2\mathsf{s}_i}\sum_{j\in\mathscr{S}_i}\hat{r}_j, \tag{39}$$

where $\mathscr{S}_i$ is the set of nodes directly connected to node i, excluding node i itself and s_i is the number of nodes in set $\mathscr{S}_i$. Applying this procedure to all of the nodes in turn creates a new smoothed field, the process is repeated a number of times (in the examples presented here 5 iterations are used when $\widehat{\mathbf{A}^{\mathrm{T}}\boldsymbol{\Psi}^{*}_{\text{exact}}}$ is amended in (22) using this approach), producing increasingly smooth fields.

5. 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 $\widehat{\Delta\mathscr{T}}$. The solution of this problem provides the forward solution $\boldsymbol{\Psi}$. The time step $\widehat{\Delta\mathscr{T}}$ 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 initialised and solved backward from this point ($t = \tau + \widehat{\Delta\mathscr{T}}$), using a time step of size $\widehat{\Delta\mathscr{T}}/2$, giving the adjoint solution $\boldsymbol{\Psi}^*$. The source term $\partial F/\partial\boldsymbol{\Psi}$ for the adjoint problem is constructed from the solution to the forward problem $\boldsymbol{\Psi}$ (at time $t = \tau + \widehat{\Delta\mathscr{T}}$).

This procedure is best illustrated by Fig. 3 which shows the process for a computation between two mesh adaptations. Once $\boldsymbol{\Psi}$ and $\boldsymbol{\Psi}^*$ 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 $t = \tau + \widehat{\Delta\mathscr{T}}$), at which point the process is repeated. The initial conditions, boundary conditions and solution procedure for this forward problem (with a large time step $\widehat{\Delta\mathscr{T}}$) are identical to those for the problem being solved. Once the end of the time step $\widehat{\Delta\mathscr{T}}$ has been reached the adjoint problem is initialised; the initial conditions are set to the values produced by the forward problem at time $t = \tau + \widehat{\Delta\mathscr{T}}$. More details on the construction of the adjoint and boundary conditions may be found in Fang et al. (this issue).

6. 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. (2004a)

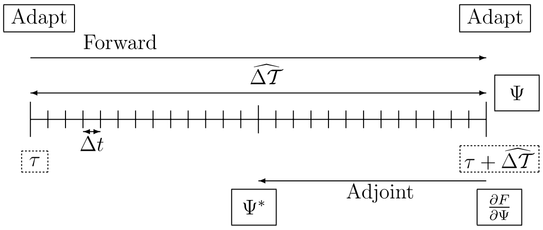

Fig. 3. Pictorial representation of time domain.

and Pain et al. (2005). In the model the three-dimensional (3-D) non-hydrostatic Boussinesq equations are discretised in the domain $\Omega \subset \mathbb{R}^3$. The equations are as follows:

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + 2\Omega \times \boldsymbol{u} = -\nabla p - \rho g \boldsymbol{k} + \nabla \cdot \boldsymbol{\tau}, \tag{40}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{41}$$

where $\boldsymbol{u} = (u, v, w)^{\mathrm{T}}$ represents the 3-D 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 $\boldsymbol{\Omega} = \left(0, 0, \frac{f}{2}\right)^{\mathrm{T}}$, where f is the Coriolis parameter. ρ is the perturbation density, g is the acceleration due to

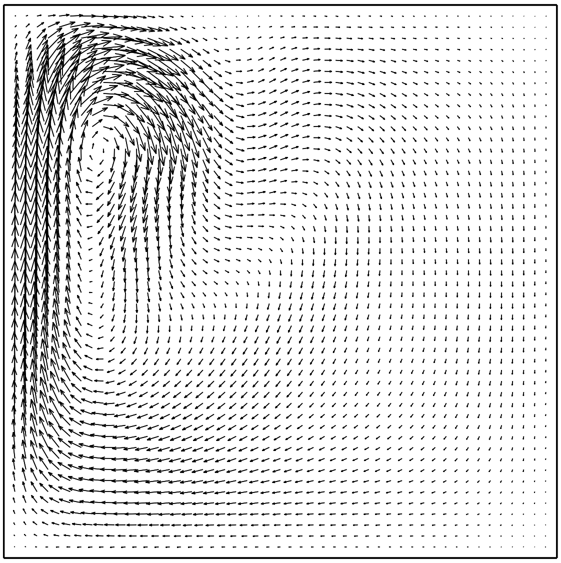

Fig. 4. Initial flow vectors (maximum velocity = 9.87) at day 100, $Re = 20$.

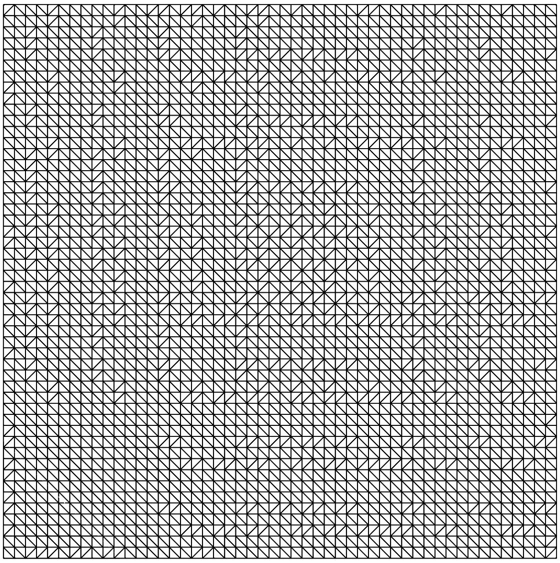

Fig. 5. Initial mesh, at day 100. 3362 nodes, 9600 elements, $Re = 20$.

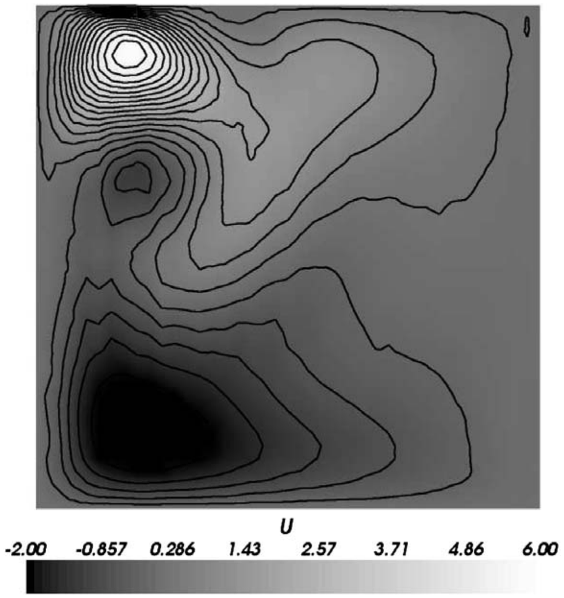

Fig. 6. Initial x-component of velocity, at day 100, $Re = 20$.

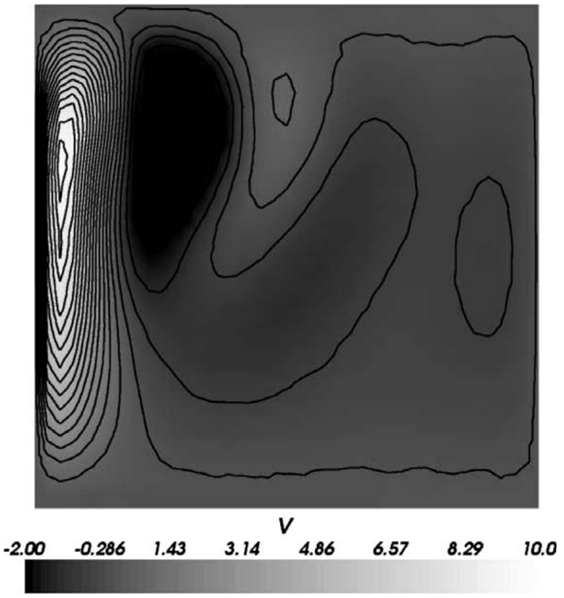

Fig. 7. Initial y-component of velocity, at day 100, $Re = 20$.

gravity and $\boldsymbol{k} = (0, 0, 1)^{\mathrm{T}}$. The viscous terms are represented by the stress tensor $\boldsymbol{\tau}$. Additional details of the numerical basis of the model can be found in Ford et al. (2004a).

A candidate functional used to derive the error metric and thus guide mesh adaptivity can be defined in terms of the vorticity of the flow dynamics. As the application of the model in this paper is 2-D only one component of the vorticity,

$$\zeta = \left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right), \tag{42}$$

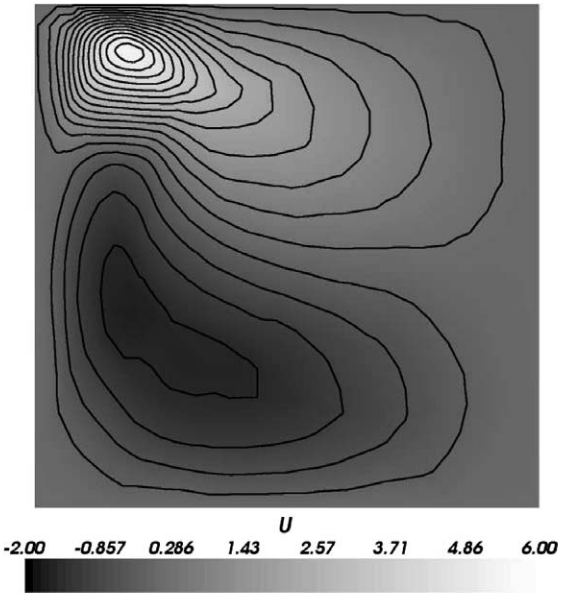

Fig. 8. x-component of forward velocity at end of large timestep $\widehat{\Delta\mathscr{T}}$ (equivalent of day 105), $Re = 20$.

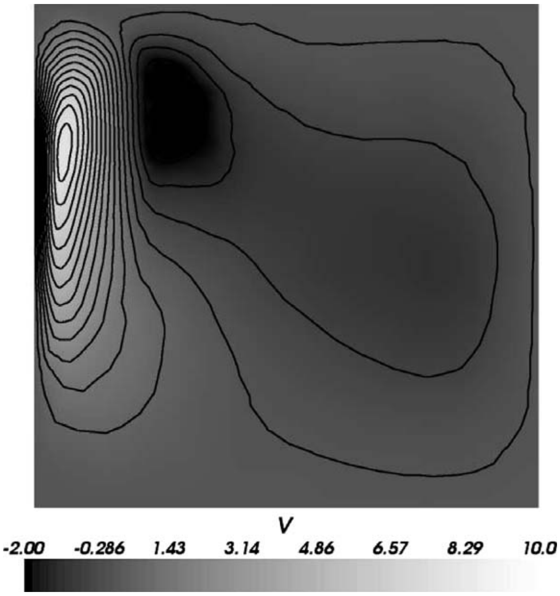

Fig. 9. y-component of forward velocity at end of large timestep $\widehat{\Delta\mathscr{T}}$ (equivalent of day 105), $Re = 20$.

is non-zero. The rationale for basing functionals around vorticity stems from both the physics but also the ability to represent the vorticity numerically. Thus an appropriate functional can be defined in terms of

$$f(u,v) = \frac{1}{2}\zeta^2. \tag{43}$$

The functional F (7) being optimised is then simply the enstrophy of the flow.

To initialise the adjoint problem (see Section 5) the source terms $\partial F/\partial \Psi$ must be calculated. When time stepping is used the functional from Section 3 can be defined as total enstrophy at time level $n + L$

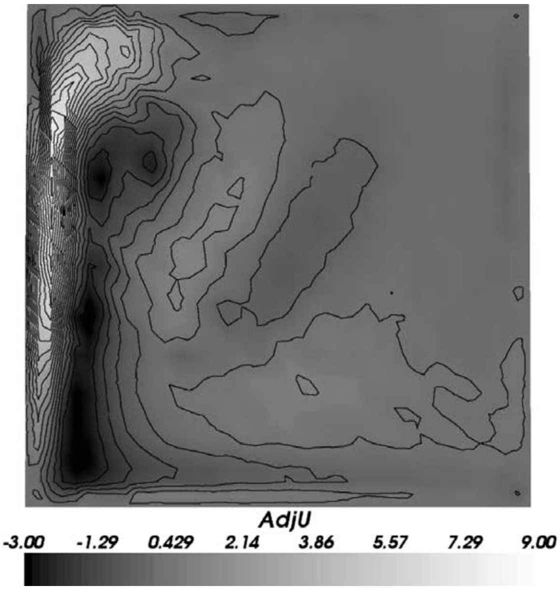

Fig. 10. x-component of adjoint velocity, after adjoint timestep of $\frac{1}{2}\widehat{\Delta\mathscr{T}}$ (equivalent of day 102.5), $Re = 20$.

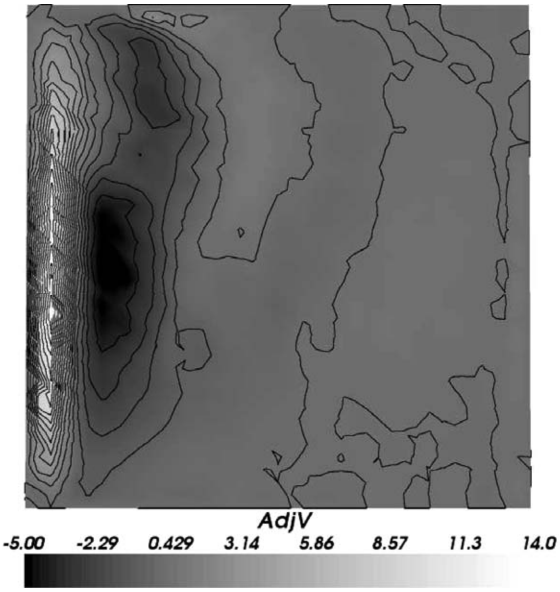

Fig. 11. y-component of adjoint velocity, after adjoint timestep of $\frac{1}{2}\widehat{\Delta\mathscr{T}}$ (equivalent of day 102.5), $Re = 20$.

$$F = \frac{1}{2}\int(\zeta^{(n+L)})^2\,\mathrm{d}V = \frac{1}{2}\int\left(\left(\frac{\partial v}{\partial x}\right)^{(n+L)} - \left(\frac{\partial u}{\partial y}\right)^{(n+L)}\right)^2\mathrm{d}V, \tag{44}$$

in which L is the number of time steps used to form the error measure. The aim of using the functional defined by (44) is to optimise the accuracy of the functional at the end of time step $\widehat{\Delta\mathscr{T}}$ (time interval between two consecutive mesh adaptations), or taking into account all time levels using

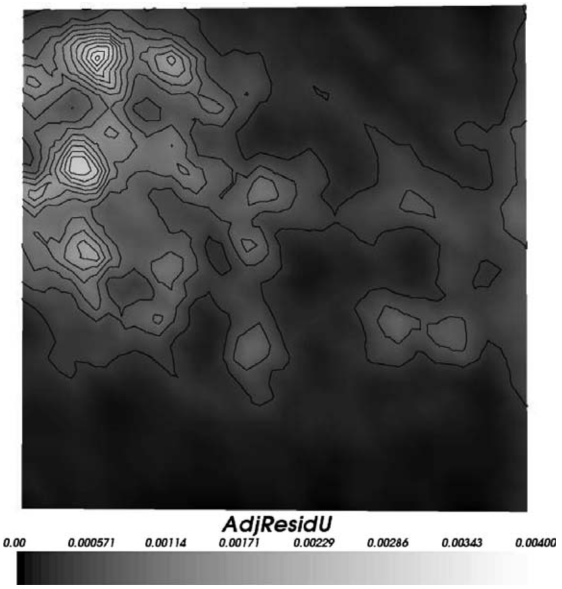

Fig. 12. The residual of the adjoint equation associated with u^*, at day 102.5, $Re = 20$ (after 5 smoothing iterations).

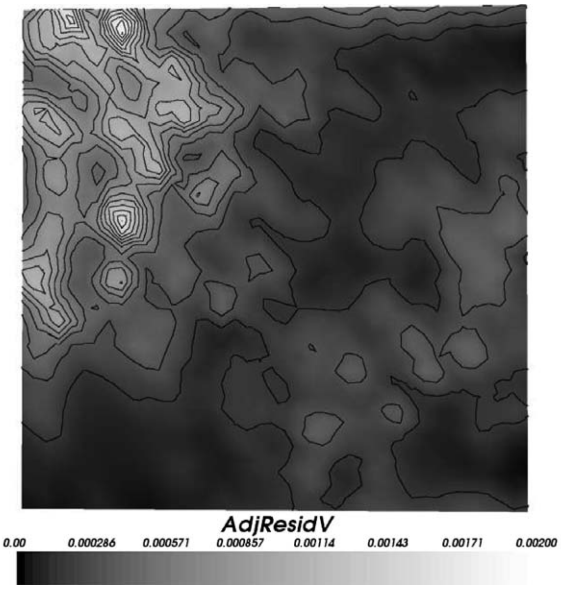

Fig. 13. The residual of the adjoint equation associated with v^*, at day 102.5, $Re = 20$ (after 5 smoothing iterations).

$$f = \frac{1}{2} \sum_{k=n+1}^{n+L} (\zeta^{(k)})^2. \tag{45}$$

Thus

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

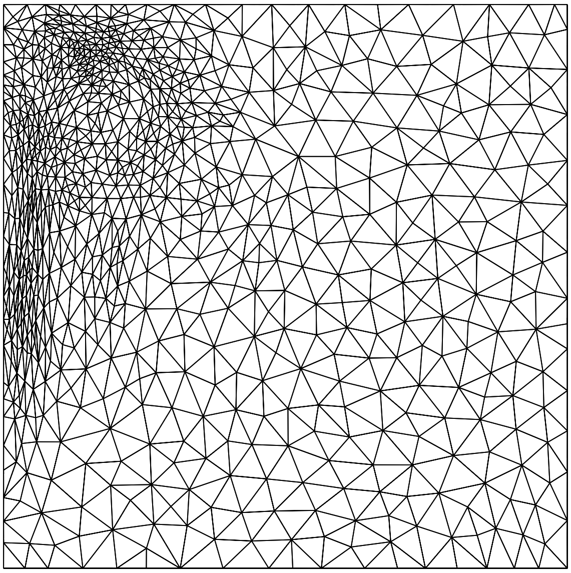

Fig. 14. Mesh adapted to optimise accuracy of functional F at day 105. 1716 nodes, $Re = 20$.

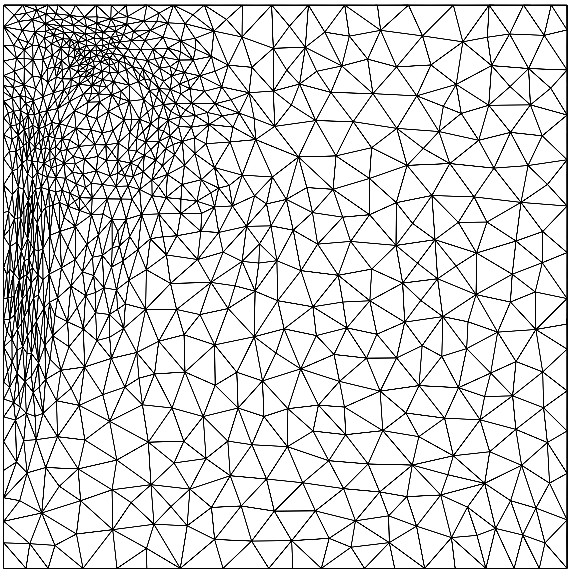

Fig. 15. Mesh adapted to optimise accuracy of functional F at day 110. 1730 nodes, $Re = 20$.

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

$$\begin{pmatrix} \mathsf{A}^{(n+1)} & 0 & \cdots & 0 \\ \mathsf{B}^{(n+2)} & \mathsf{A}^{(n+2)} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \mathsf{B}^{(n+L)} & \mathsf{A}^{(n+L)} \end{pmatrix} \begin{pmatrix} \phi^{(n+1)} \\ \phi^{(n+2)} \\ \vdots \\ \phi^{(n+L)} \end{pmatrix} = \begin{pmatrix} f^{(n+1)} - \mathsf{B}^{(n+1)}\phi^{(n)} \\ f^{(n+2)} \\ \vdots \\ f^{(n+L)} \end{pmatrix}, \tag{46}$$

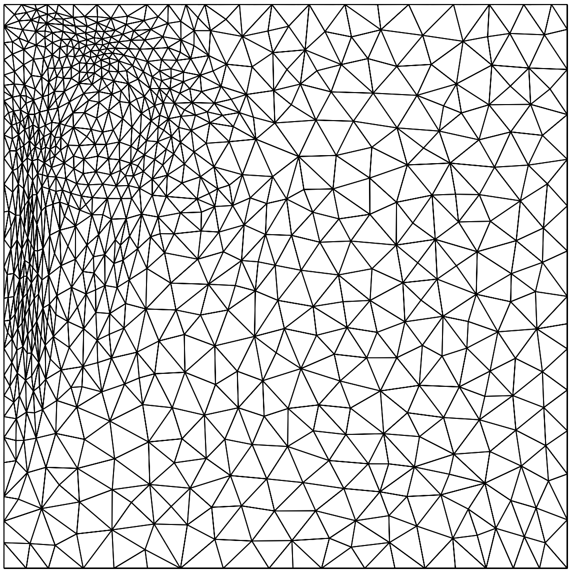

Fig. 16. Mesh adapted to optimise accuracy of functional F at day 125. 1596 nodes, $Re = 20$.

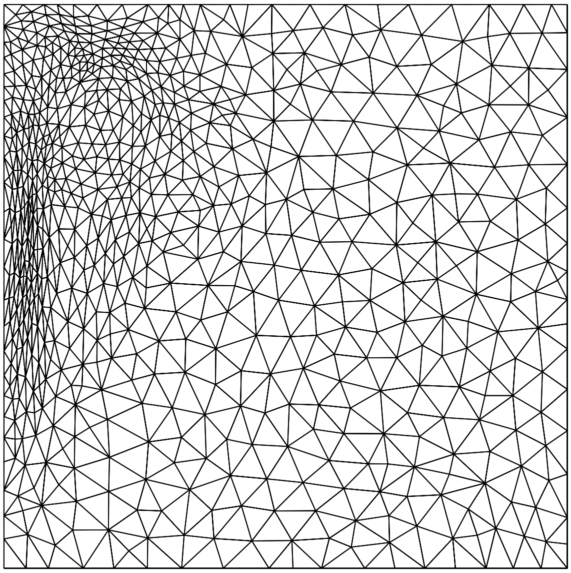

Fig. 17. Mesh adapted to optimise accuracy of functional F at day 150. 1433 nodes, $Re = 20$.

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

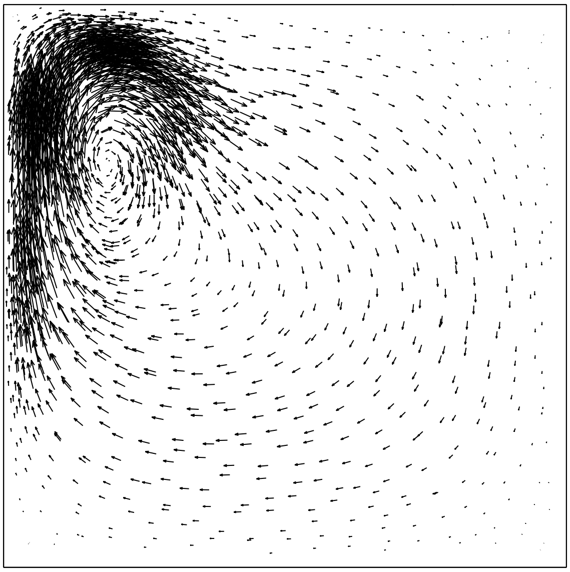

Fig. 18. Forward flow vectors at day 105 (maximum velocity =7.82), $Re = 20$.

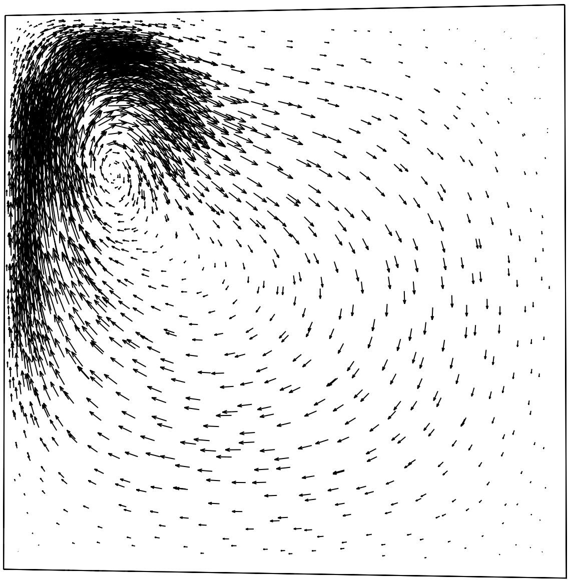

Fig. 19. Forward flow vectors at day 110 (maximum velocity =6.85), $Re = 20$.

$$\begin{pmatrix} (\mathsf{A}^{(n+1)})^{\mathrm{T}} & (\mathsf{B}^{(n+2)})^{\mathrm{T}} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & (\mathsf{A}^{(n+L-1)})^{\mathrm{T}} & (\mathsf{B}^{(n+L)})^{\mathrm{T}} \\ 0 & \cdots & 0 & (\mathsf{A}^{(n+L)})^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} \phi^{*(n+1)} \\ \vdots \\ \phi^{*(n+L-1)} \\ \phi^{*(n+L)} \end{pmatrix} = \begin{pmatrix} \frac{\partial F}{\partial \phi^{(n+1)}} \\ \frac{\partial F}{\partial \phi^{(n+2)}} \\ \vdots \\ \frac{\partial F}{\partial \phi^{(n+L)}} \end{pmatrix}, \tag{47}$$

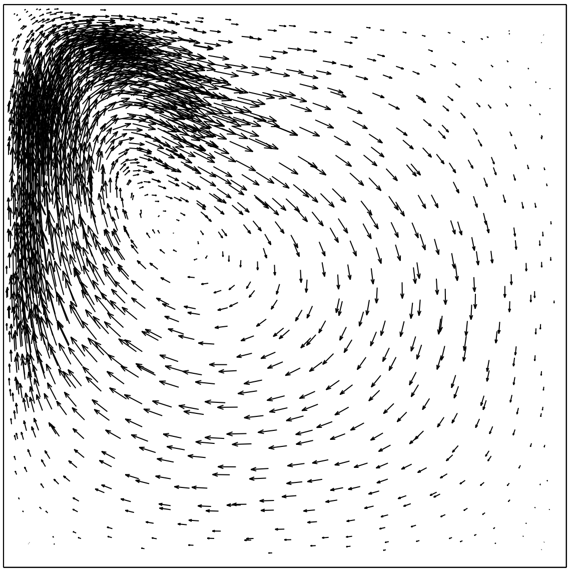

Fig. 20. Forward flow vectors at day 120 (maximum velocity =6.66), $Re = 20$.

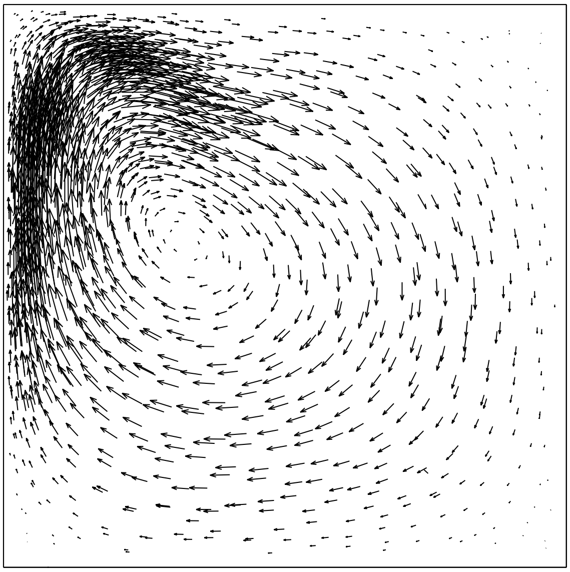

Fig. 21. Forward flow vectors at day 145 (maximum velocity = 7.05), $Re = 20$.

where

$$\left(\frac{\partial F}{\partial \phi^{(n+1)}}, \frac{\partial F}{\partial \phi^{(n+2)}}, \ldots, \frac{\partial F}{\partial \phi^{(n+L)}}\right)^{\mathrm{T}} = \left(0, \ldots, 0, \frac{\partial F}{\partial \phi^{(n+L)}}\right)^{\mathrm{T}}.$$

In (47) 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 equation (47) as that used to solve the forward equation (46). This modifies the equations above

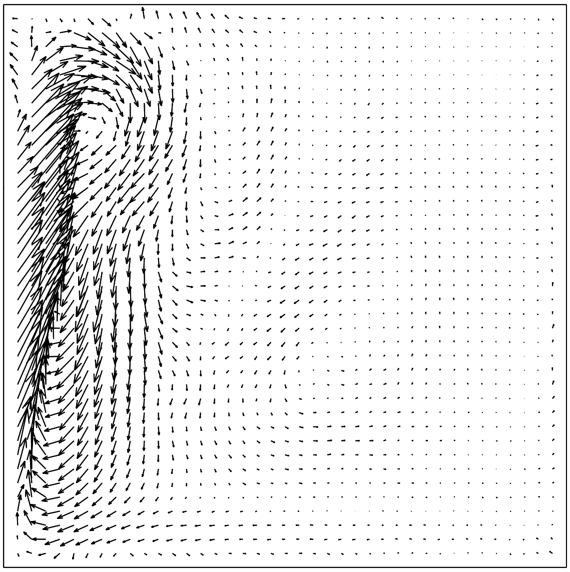

Fig. 22. Adjoint flow vectors at day 102.5 (maximum velocity = 15.1), $Re = 20$.

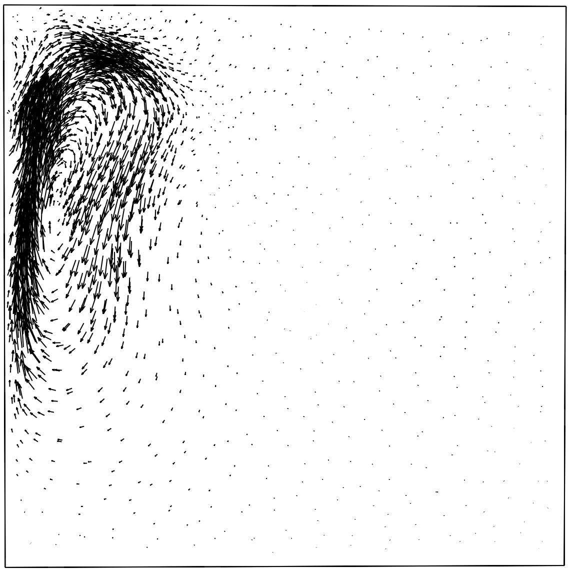

Fig. 23. Adjoint flow vectors at day 107.5 (maximum velocity = 11.5), $Re = 20$.

slightly, but is an approximation to them. This has been the motivation for not necessarily assuming that $r(\psi) = 0$ in the derivation of (18). 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 discretisation method is used as that to solve the forward equations, i.e. Galerkin spatial discretisation with Crank–Nicolson time stepping. Thus the SUPG (Streamline Upwind Petrov–Galerkin) method for spatial discretisation with backward Euler time-stepping is employed to evaluate (46) and (47). For $L = 1$, Eqs. (46) and (47) become (assuming ϕ^n is known from before the model is adapted)

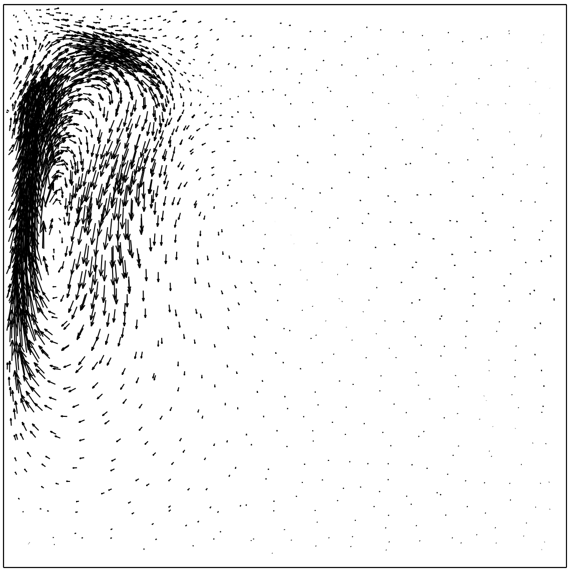

Fig. 24. Adjoint flow vectors at day 117.5 (maximum velocity = 9.94), *Re* = 20.

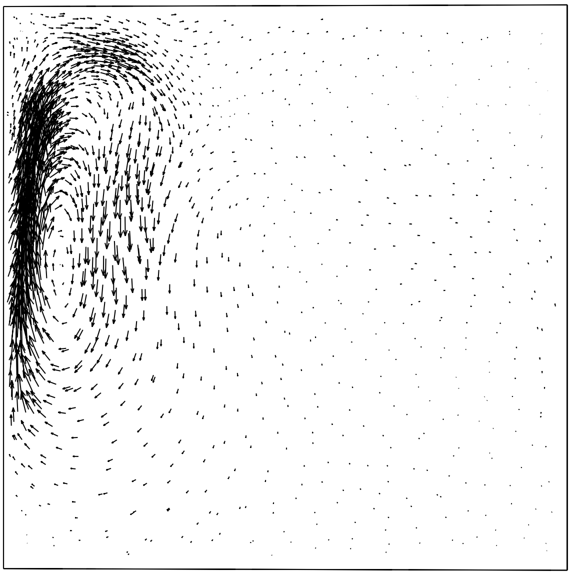

Fig. 25. Adjoint flow vectors at day 147.5 (maximum velocity = 10.6), *Re* = 20.

$$\mathsf{A}^{(n+L)}\phi^{(n+L)} = -\mathsf{B}^{(n+L)}\phi^{n} + f^{(n+L)}, \tag{48}$$

$$(\mathsf{A}^{(n+L)})^{\mathrm{T}}\phi^{*(n+L)} = \frac{\partial F}{\partial\phi^{(n+L)}}, \tag{49}$$

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 $\phi = \frac{1}{2}(\phi^{(n)} + \phi^{(n+L)})$, and similarly to evaluate ϕ^{*} between mesh adaptations. The aim is to obtain a mesh which can define the fields at each time step between adapts and thus optimising the mesh at the mid time level. Thus the equations become

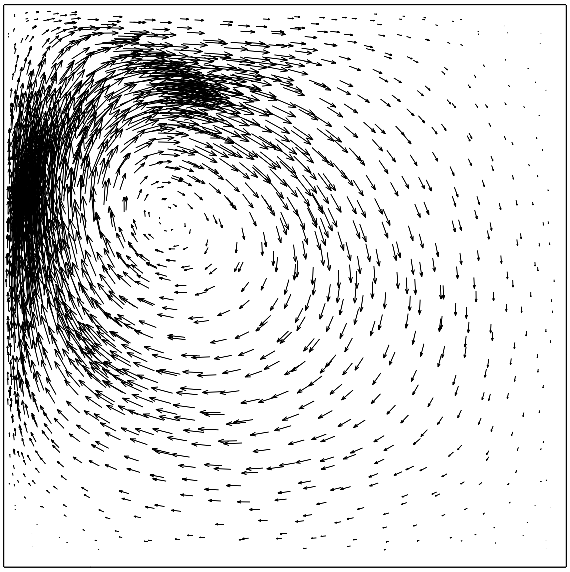

Fig. 26. Forward flow vectors after one year (maximum velocity = 11.2), $Re = 20$.

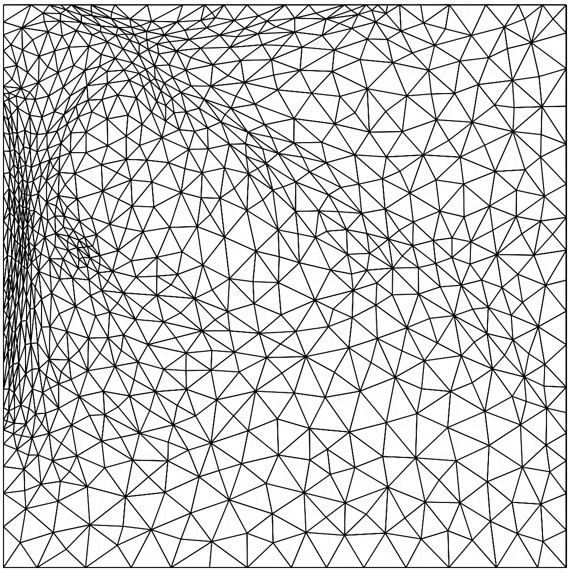

Fig. 27. Mesh adapted to optimise accuracy of functional F after one year. 1656 nodes, $Re = 20$.

$$\mathsf{A}^{(n+L)}\phi^{(n+L)} = -\mathsf{B}^{(n+L)}\phi^{n} + f^{(n+L)}, \tag{50}$$

$$\left(\mathsf{A}^{\left(n+\frac{1}{2}\right)}\right)^{\mathrm{T}}\phi^{*\left(n+\frac{1}{2}\right)} = \frac{\partial F}{\partial\phi^{(n+L)}}, \tag{51}$$

with

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

These values are used to evaluate the metric tensor, given by (28) and the process is shown schematically in Fig. 3.

7. Application: non-linear wind-driven gyre

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 $L = 1000$ km which is $H = 500$ m deep. A maximum zonal wind stress of $\tau_0 = 0.1$ N m^{-2} is applied in a cosine of latitude profile. The Coriolis terms are taken into

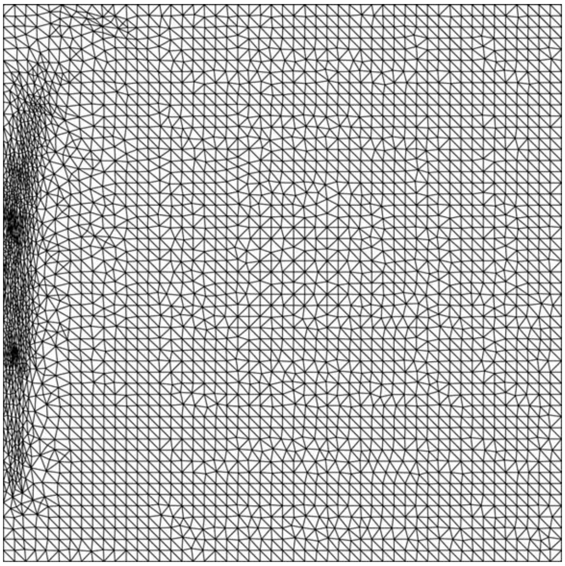

Fig. 28. Mesh adapted to optimise accuracy of functional F at day 105. 6894 nodes, $Re = 1000$.

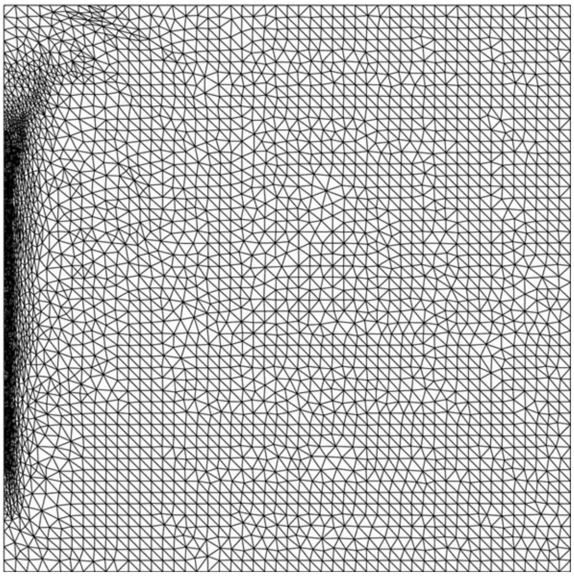

Fig. 29. Mesh adapted to optimise accuracy of functional F at day 110. 7057 nodes, $Re = 1000$.

account with the beta-plane approximation ($f = \beta y$) where $\beta = 1.8 \times 10^{-11}\ \mathrm{m}^{-1}\ \mathrm{s}^{-1}$, and the reference density is taken to be $\rho_0 = 1000\ \mathrm{kg\ m}^{-3}$.

The problem is non-dimensionalised with the maximum Sverdrup balance velocity

$$\beta H \rho_0 v = \frac{\partial \tau}{\partial y} \leqslant \frac{\tau_0 \pi}{L} \Rightarrow v \leqslant 3.5 \times 10^{-2}\ \mathrm{m\ s}^{-1},$$

(and so the velocity scale $U = 3.5 \times 10^{-2}\ \mathrm{m\ s}^{-1}$ is used here), and the horizontal length-scale ($L = 10^6$ m) so that the domain is a box with a horizontal size of *1* and depth of 0.0005. The domain is maintained as a single layer, i.e. one element in the vertical. Time is non-dimensionalised with $T = L/U$. Incorporating the beta-plane

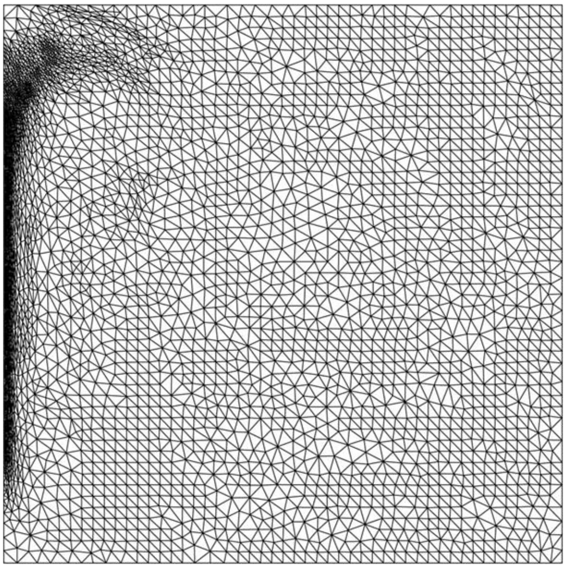

Fig. 30. Mesh adapted to optimise accuracy of functional *F* at day 115. 6869 nodes, *Re* = 1000.

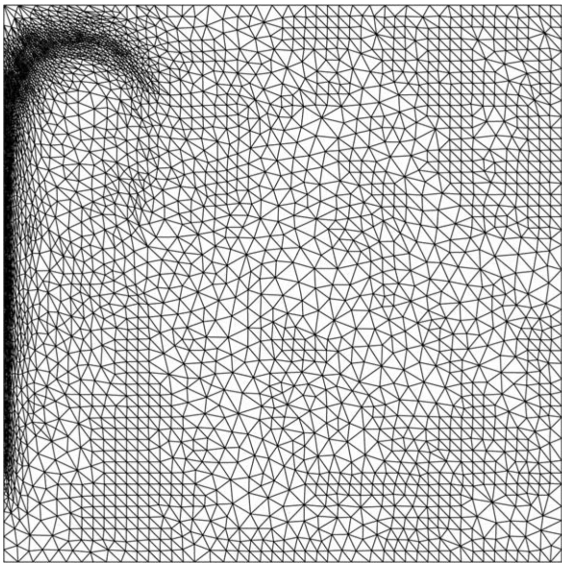

Fig. 31. Mesh adapted to optimise accuracy of functional *F* at day 120. 6678 nodes, *Re* = 1000.

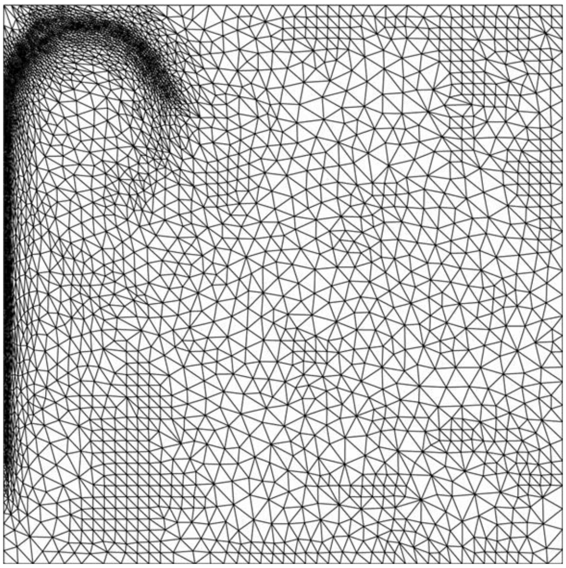

Fig. 32. Mesh adapted to optimise accuracy of functional F at day 125. 6324 nodes, $Re = 1000$.

approximation gives a non-dimensional $\beta^* = L^2\beta/U = 514.286$. The non-dimensional wind stress (applied as a body force here averaged over the depth of the domain) takes the same cosine of latitude profile with $\tau_0^* = \tau_0 L/(U^2\rho_0 H) = 163.2653$. The non-dimensional time step is taken to be 3.78×10^{-4}, equivalent to 3 hours. No-slip conditions are applied to the lateral boundaries.

Two sets of results are presented with Reynolds numbers of 20, ensuring a non-chaotic regime (Section 7.1) and 1000, providing a reasonably chaotic regime (Section 7.2). The Reynolds number is defined as $Re = UL/\nu$ and hence the values 20 and 1000 correspond to kinematic viscosities of 35 and 1750 $\mathrm{m^2\,s^{-1}}$, respectively. The Munk layer thickness $(\nu/\beta)^{1/3}$ in each case is then approximately 12.5 and 46 km, respectively.

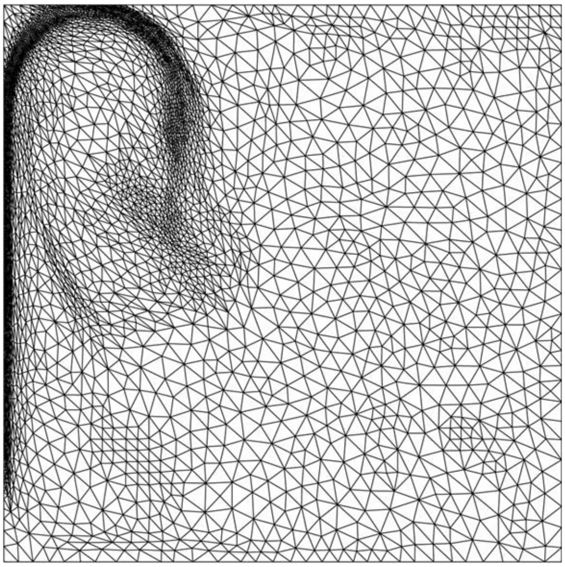

Fig. 33. Mesh adapted to optimise accuracy of functional F at day 150. 6248 nodes, $Re = 1000$.

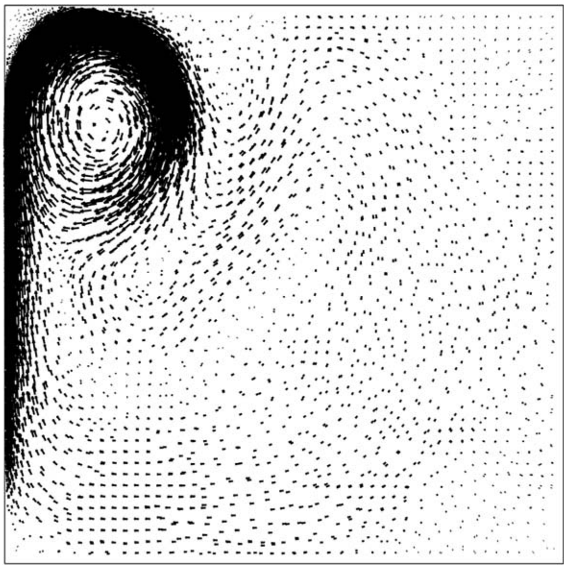

Fig. 34. Flow vectors at day 125, $Re = 1000$.

The functional and source terms are calculated using the solution to the forward problem Ψ (see Section 5 for details), at time level $t = \tau + \widehat{\Delta\mathscr{T}}$, where $\widehat{\Delta\mathscr{T}}$ is taken to be 1.51×10^{-2} or 5 days, which is the time between adaptations of the mesh, as discussed in Section 5. δF is taken to be for 7.85×10^{-1} for $Re = 20$ and 5.0×10^{-1} for $Re = 1000$, with the minimum and maximum edge lengths being 0.001 (1 km) and 0.1 (100 km), respectively. The number of nodes is capped at 10,000. Note that the number of nodes quoted in the figures is the value for the entire one element deep three-dimensional mesh, to compare with the resolution (but not cost) of a two-dimensional code this number would need to be halved.

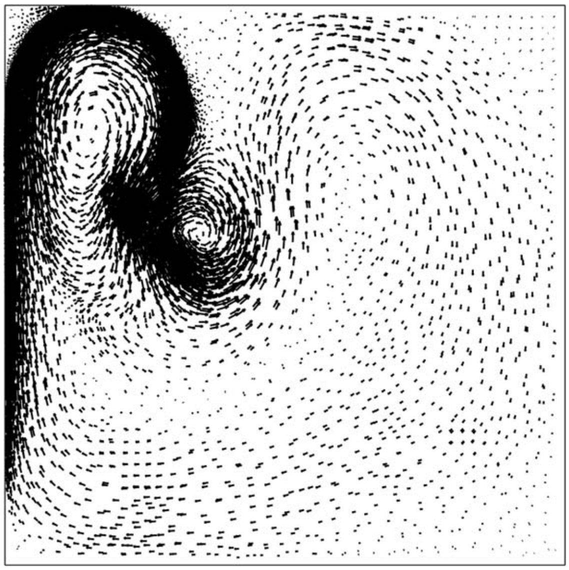

Fig. 35. Flow vectors at day 150, $Re = 1000$.

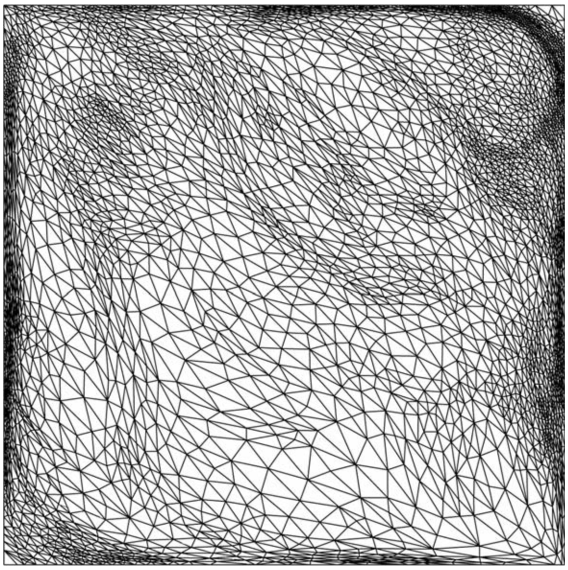

Fig. 36. Mesh adapted to optimise the functional F after 2 years. 6651 nodes, $Re = 1000$.

7.1. Results—$Re = 20$

The forward metric tensor is used in this simulation. The mesh adapts every 41 time steps (5 days). The adaptive method is employed on this problem from 100 days into the simulation, after the initial spin-up of the gyre is complete. The simulation is conducted up to this point on a set mesh of 40×40 elements; the initial velocity vectors, mesh and velocity components are shown in Figs. 4–7. The forward and adjoint time steps are preformed as detailed in Section 5, the adjoint problem being initialised from source terms derived from the differential of the functional F defined in terms of the vorticity. The u and v velocity

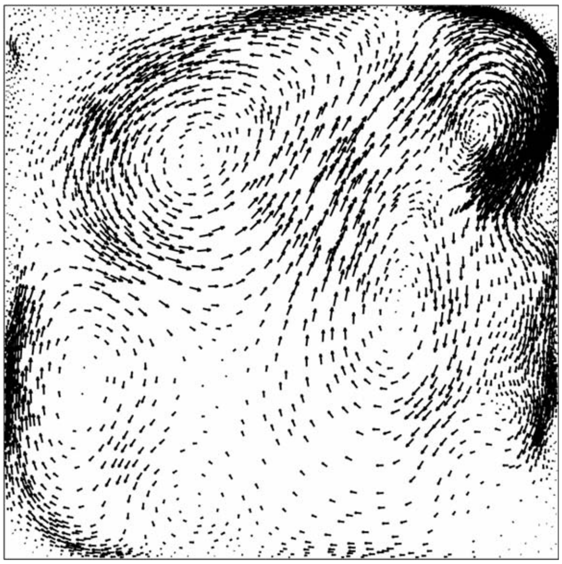

Fig. 37. Flow vectors after 2 years, $Re = 1000$.

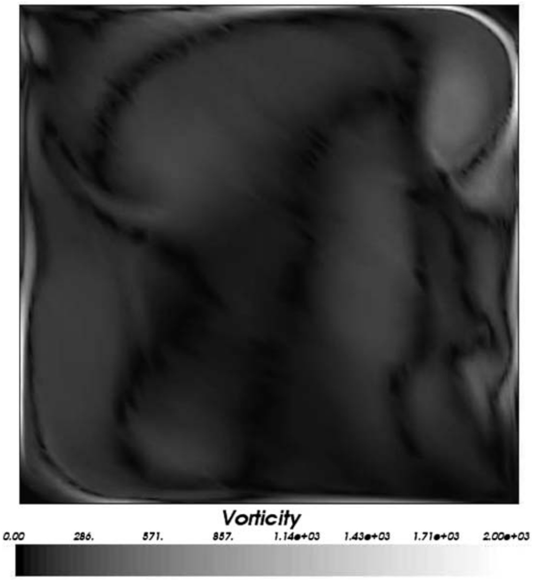

Fig. 38. Vorticity after 2 years, $Re = 1000$.

components of the forward solution Ψ are shown in Figs. 8 and 9; with the components u^* and v^* of the adjoint solution Ψ^* shown in Figs. 10 and 11. Once the forward and adjoint solutions have been calculated the residuals can be obtained. As the metric tensor (28) is employed in adapting the mesh here it is useful to look at the adjoint residuals. The residuals associated with u^* and v^* are estimated by the method described in Section 4. These are shown in Figs. 12 and 13. The Hessians of the forward solution fields are calculated and averaged, and used to form the metric tensor $\boldsymbol{M}$ with which the mesh is adapted. The results shown in Figs. 8–13 relate to the first adaptation 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 optimise the accuracy of the functional F at day 105. The adapted

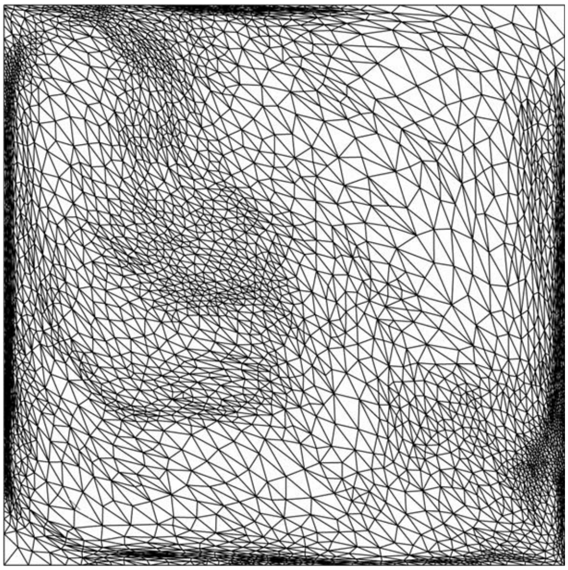

Fig. 39. Mesh adapted to optimise the functional F after 4 years. 6288 nodes, $Re = 1000$.

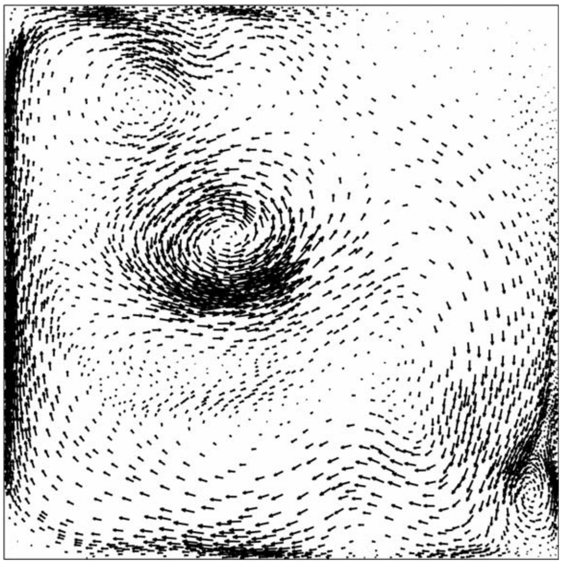

Fig. 40. Flow vectors after 4 years, $Re = 1000$.

mesh, used to compute the actual solution, using time steps of 3 h between days 100 and 105 is shown in Fig. 14. The process is repeated when the simulation reaches day 105, taking a large time step to day 110 and finding a mesh to optimise the accuracy of the functional F at day 110. Subsequent adapted meshes are shown in Figs. 15–17. Areas highlighted by both the Hessian of the forward solution and the residual can be seen to be receiving mesh refinement, whilst in other areas the mesh is coarsened. It can be seen that increased mesh resolution is concentrated around the western boundary. The finest resolution normal to the boundary at the western side of the domain is approximately 15 km (for the mesh shown in Fig. 17). As described above, the mesh adapts every 40 time steps (5 days). When the mesh adapts, the CPU time overhead

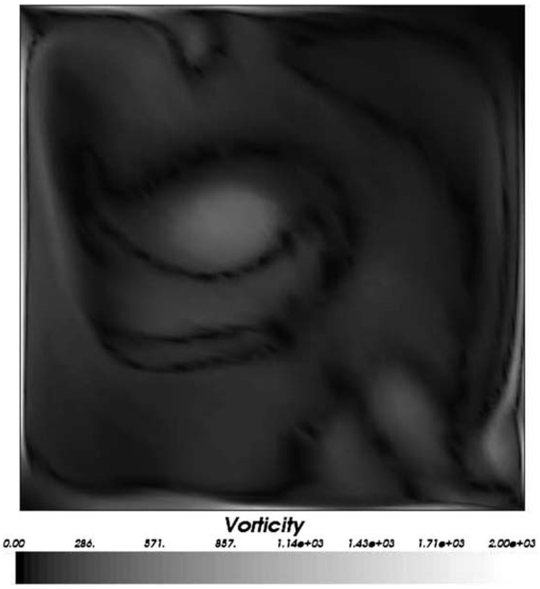

Fig. 41. Vorticity after 4 years, $Re = 1000$.

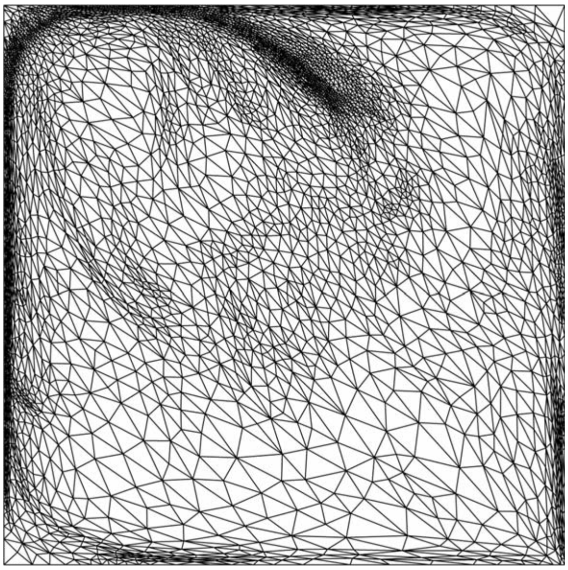

Fig. 42. Mesh adapted to optimise the functional F after 6 years. 6400 nodes, $Re = 1000$.

associated with the mesh optimisation algorithm is equivalent to approximately two standard time steps of the forward model. The forward and adjoint calculations used to compute the error measures and metric have a CPU cost also approximately equivalent to the cost of two time steps. However, after the mesh is first adapted the number of nodes not only decreases to half of its original value, it also reaches a state where it is using less than a quarter of the number of nodes a uniform mesh achieving the resolution of 15 km everywhere would require. Therefore, although the calculation of the metric and the mesh adaptation requires additional computations, these costs are offset by the reduction in the total number of nodes used in the calculation.

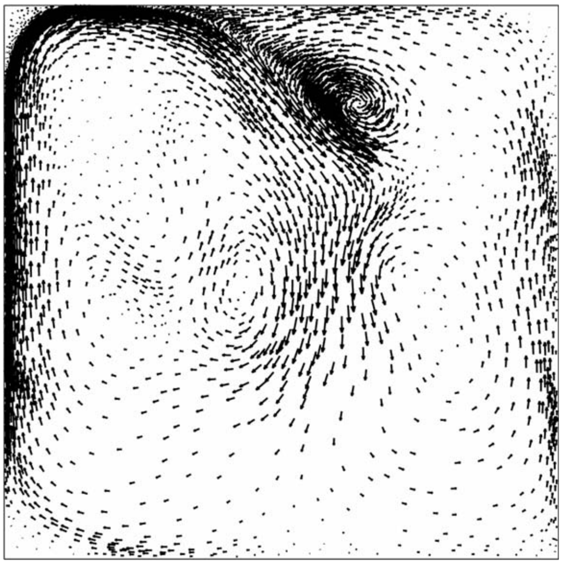

Fig. 43. Flow vectors after 6 years, $Re = 1000$.

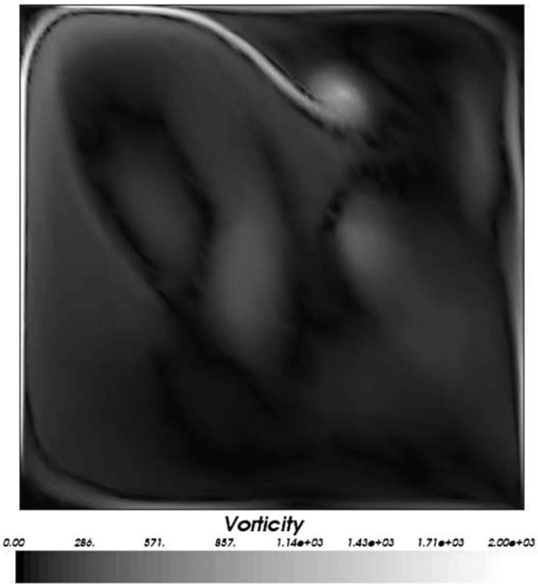

Fig. 44. Vorticity after 6 years, $Re = 1000$.

The forward vectors are given in Figs. 18–21 whilst the adjoint vectors are shown in Figs. 22–25. Figs. 26 and 27 show the flow vectors and the corresponding mesh after one year.

7.2. Results—Re = 1000

In this section the method is applied to a case with a Reynolds number of 1000; otherwise identical conditions are used as to those in Section 7.1. Meshes at equivalent times to those shown previously are shown in Figs. 28–33. Differences in the complexity of the flow can be seen. The method optimises the mesh using the

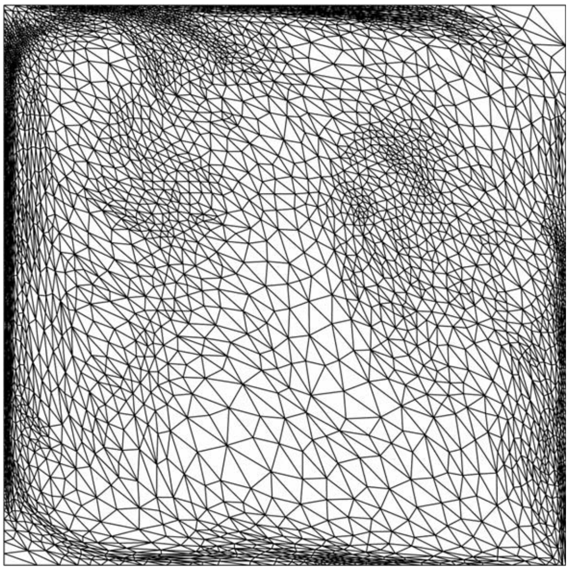

Fig. 45. Mesh adapted to optimise the functional F after 8 years. 6500 nodes, $Re = 1000$.

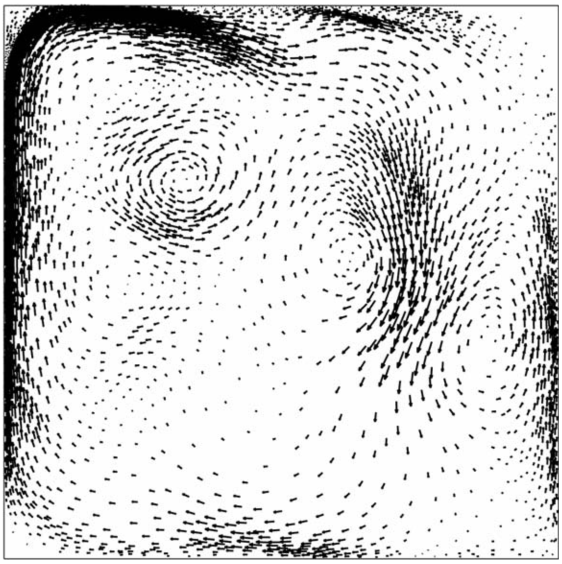

Fig. 46. Flow vectors after 8 years, $Re = 1000$.

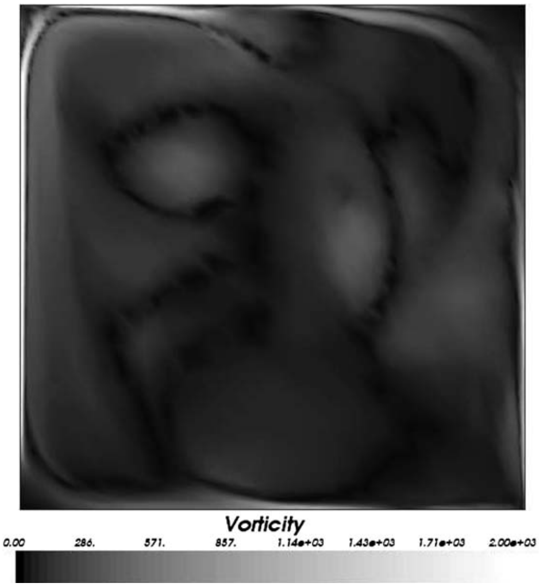

Fig. 47. Vorticity after 8 years, $Re = 1000$.

same requirements as those used in the previous section. Flow vectors at days 125 and 150 are shown in Figs. 34 and 35, respectively. The model can be seen to be resolving the intense boundary current and subsequent generation of eddies propagating into the domain interior. The finest resolution normal to the boundary at the western side of the domain is approximately 1 km (for the mesh shown in Fig. 33) and thus the model here is using a number of nodes approximately two orders of magnitude lower than would be required by a model with a uniform mesh achieving this resolution everywhere. Figs. 36–47 show results at intervals up to 8 years in simulation time. Fig. 48 shows the value of the functional F over time.

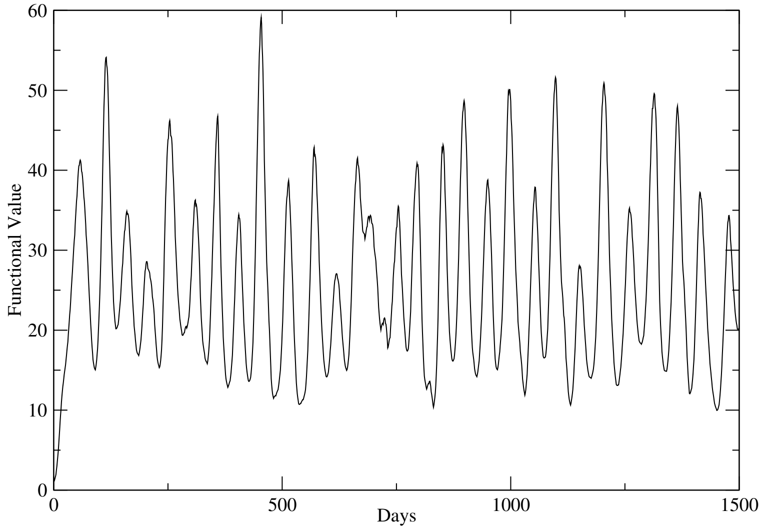

Fig. 48. Plot showing the value of the functional F over time, $Re = 1000$.

8. Concluding remarks

A novel method of constructing a metric tensor field for defining 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 wind-driven gyre problem presented here is to optimise 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 or decreased 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.

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 available computational resources. Future work will involve the testing of the method on an extended set of benchmarks and the application and comparison of a variety of functionals or goals.

Acknowledgements

The authors like to thank Prof. Raymond Hide for useful and insightful discussions related to this work. The authors would also like to acknowledge the support of the UK Natural Environment Research Council (grants NER/A/S/2000/00375, NE/C51829X/1, NE/C52101X/1), and the Leverhulme Trust. Finally the authors would like to thank the anonymous reviewers who assisted in substantially improving this paper.

References

Ainsworth, M., Oden, J.T., 1997. *A-posteriori* error estimation in finite element analysis. Comput. Meth. Appl. Mech. Eng. 142, 1–88.

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

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

Cacuci, D.G., 1988. The forward and adjoint methods of sensitivity analysis. In: Ronen, Y. (Ed.), Uncertainty Analysis, CRC Press, Boca Raton, FL, pp. 71–144 (Chapter 3).

Castro-Diaz, M.J., Hecht, F., Mohammadi, B., Pironneau, O., 1997. Anisotropic unstructured mesh adaption for flow simulations. Int. J. Numer. 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.D., Pain, C.C., Gorman, G.J., Goddard, A.J.H., this issue. An adaptive mesh adjoint data assimilation method, Ocean Modell, doi:10.1016/j.ocemod.2006.02.002.

Ford, R., Pain, C.C., Piggott, M.D., Goddard, A.J.H., de Oliveira, C.R.E., Umpleby, A.P., 2004a. A non-hydrostatic finite element model for three-dimensional stratified ocean flows. Part I: Model formulation. Mon. Wea. Rev. 132, 2816–2831.

Ford, R., Pain, C.C., Piggott, M.D., Goddard, A.J.H., de Oliveira, C.R.E., Umpleby, A.P., 2004b. A non-hydrostatic finite element model for three-dimensional stratified ocean flows. Part II: Model validation. Mon. Wea. Rev. 132, 2832–2844.

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

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

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

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

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

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

Ierley, G.R., 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.M., 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 Jacobian matrix. Int. J. Numer. Meth. Eng. 48, 1165–1185.

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

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

Müller, J.D., Giles, M.B., June 2001. Solution adaptive mesh refinement using adjoint error analysis. In: 15th Computational Fluid Dynamics Conference. American Institute of Aeronautics and Astronautics, Paper number 2001-2550.

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

Oden, J.T., Prudhomme, S., 2001. Goal-oriented error estimation and adaptivity for the finite element method. Comput. Math. Appl. 41, 735–756.

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

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

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.C., 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. Eng. 123, 488–492.

Pierce, N.A., Giles, M.B., 2000. Adjoint recovery of superconvergent functionals from PDE approximations. SIAM Rev. 42, 247–264.

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

Prudhomme, S., Oden, J.T., 1999a. *A-posteriori* error estimation and error control for finite element approximations of the time-dependent Navier–Stokes equations. Finite Elem. Anal. Des. 33, 247–262.

Prudhomme, S., Oden, J.T., 1999b. On goal-oriented error estimation for elliptic problems. Comput. Meth. Appl. Mech. Eng. 179, 313–331.

Sanders, B.F., Katopodes, N.D., 1999. Control of canal flow by adjoint sensitivity method. J. Irrigation Drainage Eng. 125, 287–297.

Sanders, B.F., Katopodes, N.D., 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.A., Darmofal, D.L., 2000. Adjoint error estimation and grid adaption for functional outputs: application to quasi-one-dimensional flow. J. Comput. Phys. 164, 204.

Venditti, D.A., Darmofal, D.L., 2002. Grid adaption for functional outputs: application to two-dimensional inviscid flows. J. Comput. Phys. 176, 40–69.

Venditti, D.A., Darmofal, D.L., 2003. Anisotropic grid adaption for functional outputs: application to two-dimensional viscous flows. J. Comput. Phys. 187, 22–46.

Wu, J., Zhu, J.Z., Szmelter, J., Zienkiewicz, O.C., 1990. Error estimation and adaptivity in Navier–Stokes incompressible flows. Comput. Mech. 6, 259–270.

Zhang, S., Zou, X., Ahlquist, J., Navon, I.M., Sela, J.G., 2000. Use of differentiable and non-differentiable optimization algorithms for variational data assimilation with discontinuous cost functions. Mon. Wea. Rev. 128, 4031–4044.

Zienkiewicz, O.C., Taylor, R.L., 2000, fifth ed. The Finite Element Method, vol. 1. Butterworth-Heinemann, Oxford.