

Development of a Simplified Adaptive Finite Element Model of the Gulf Stream

Nicola Stone

August 20, 2006

Abstract

The goal of this dissertation is to look at the feasibility of incorporating a moving adaptive mesh refinement procedure into an existing finite element model of the shallow water equations. The Stommel model simulates simple oceanic flows including western boundary currents like the Gulf Stream. This dissertation describes the finite element formulation of the Stommel equations, a modified form of the shallow water equations. The feasibility

Contents

1	Introduction	1
2	The Stommel Model	5
2.1	The Shallow Water Equations	5
2.2	The Stommel Model	6
2.3	Scale Analysis	8
3	Numerical Solution to the Stommel Problem	11
3.1	Finite Element Formulation	11
3.2	Temporal Discretisation	13
3.3	Computation	14
3.3.1	Mass Matrix Entries	14
3.3.2	Computing the solution	15
3.4	Initial Conditions	16
3.5	Stommel Solution	17
4	Moving Mesh Adaptation	19
4.1	Moving Mesh Methods	19
4.2	Mesh Velocity Principle	20

5	Solving for the Mesh Velocities	23
5.1	Finite Element Formulation	23
5.1.1	Stiffness Matrix Entries	24
5.1.2	Load Vector Entries	26
5.1.3	Mesh Velocities	27
6	Implementation of the Mesh Velocities	29
6.1	Interpolating the Mesh Velocities onto the P_1 nodes . .	29
6.2	Moving the Mesh	30
6.3	Interpolating the Mesh Velocities onto the P_1^{NC} nodes .	30
6.4	Moving the Solution	31
7	Numerical Simulations and Results	33
7.1	Simple westward mesh movement	34
7.1.1	Results for simple westward mesh movement . .	34
7.2	Conservation based mesh movement	36
7.2.1	Results for conservation based mesh move . .	36
7.2.1	Results for conservation based mesh monervationv7.-22.317Td(.)-5ci3C	
6.3	Interpolating the Mesh Vion basedbased 3vtm.1 for fsh	

List of Figures

2.1	Depth of the fluid (1-D)	6
3.1	P_1 and P_1^{NC} basis functions	12
3.2	Initial Mesh	16
3.3	Stommel solution on a square domain	17
5.1	Dimensions of an element in the mesh	25
6.1	Averaging the element velocities to obtain the mesh velocities on the nodes	30
6.2	Averaging the element velocities to get the mesh velocities on the mid segments	31
7.1	Solution after 1000 time steps with $\alpha = 0.5$	35
7.2	Solution after 50,000 time steps with $\alpha = 0.01$	35
7.3	Solution after 1000 time steps using conservation principle	37
7.4	Solution after 25000 time steps using conservation principle	38
7.5	Solution after 3000 time steps using flow based mesh movement	39

7.6	Solution after 9000 time steps using combined conserva-	
	tion/flow based mesh movement	41

List of Tables

2.1	Scales for each parameter in the Stommel equations . .	9
-----	--	---

Chapter 1

Introduction

Traditionally, ocean models used structured grids, reflecting the computational resources that were available at the time. Now ocean modellers are starting to use unstructured and adapting meshes more.

Unstructured meshes have many advantages over structured meshes. The coastline and basin geometry can be better represented. Also, the mesh can be easily adapted to give an increased local mesh resolution, or a dynamically adaptive mesh [4]. In the first case, a high resolution grid can be placed in an area of interest with lower resolution in less important areas, to give optimal use of computer power. In a North Atlantic model, for example, we might place a higher resolution grid in the region of the Gulf Stream. In the second case, we can have a mesh which moves to follow certain aspects of the solution, for example eddies moving across a basin.

Adaptive meshes are useful as we may not know the positions of developing flow features before the model is run. The resolution of the model can be changed to optimally resolve the flow. Adaptive methods can also be incorporated into existing models and may give increased

efficiency and accuracy of the solution.

This dissertation will investigate the feasibility of using adaptive mesh movement in the Stommel model. The mesh movement is determined by following a certain feature (fluid volume) of the solution in Stommel's equations of ocean circulation with a prescribed vorticity. We will be looking at the possibility of including mesh movement into an existing two-dimensional finite element model. The existing model uses an implicit time discretisation with a black box solver to solve the equations. To avoid having to invert a matrix, this has been changed to an explicit scheme for this dissertation. The mesh movement is based on the idea of having a monitor function to determine how the mesh will move. The monitor function we have chosen is the depth of the ocean $H = h + \eta$, where h is the resting depth of the fluid and η is the surface elevation, and the prescribed vorticity of the mesh is taken to be zero.

We start by stating the shallow water equations in Chapter 2. Then we briefly describe how Stommel simplified these equations to create a model which describes the creation of western boundary layers in areas of the ocean such as the Gulf Stream. Scale analysis is performed on these equations to determine the dominant terms, which allow us to make assumptions of the nature of the solution without first having to solve the equation.

In Chapter 3 we look at how the governing equations of Stommel's model are formulated using finite elements, and then discretised in time [1]. We then describe how the solution is computed. The initial conditions of the model are stated, along with the initial mesh. The elevation solution on a fixed mesh is then described as a motivation to

explore the possibility of moving the mesh to follow a flow.

The idea of generating a mesh velocity from the conservation of a monitor function [8] is introduced in Chapter 4. We have chosen one particular monitor function which ensures that the volume of fluid in each element remains constant over all time. This is one of many monitor functions that could be used to move the mesh. Likewise, the choice of zero mesh vorticity is only one of many choices that can be made.

Then an equation for the mesh velocity induced by this monitor function is formulated using finite element methods in Chapter 5. The entries of the resulting stiffness matrix and load vector are then derived, and a mesh velocity calculated for each element.

Chapter 6 describes how this element velocity is interpolated onto the nodes, and the mid-segments of the elements in the mesh. The mesh node velocity is then used in a mesh movement equation to move the mesh. As the true solution should not change when the mesh is modified, the mid-segment velocity is used in advection terms included in the governing equations to compensate for the mesh movement.

The results of numerical simulations carried out to move the mesh are shown in Chapter 7. These test situations look at the suitability of different ideas for mesh movement. The first test carried out is a simple prescribed westerly mesh movement, given that we know the solution travels to the west. Then the results for the conservation of volume principle are discussed. The next test involves the Lagrangian idea of moving the mesh according to the flow velocity which is a special case of the computed velocity when the vorticity of the mesh is equal to the vorticity of the fluid. The last test is a linear combination of

Chapter 2

The Stommel Model

This chapter will describe the Stommel model, looking at the governing equations and how the model is represented using finite element methods. Then the time discretisation will be described, along with how the results are computed. This is followed by results of a fixed mesh finite element model of the Stommel model in a square domain.

2.1 The Shallow Water Equations

The shallow water equations on domain Ω with boundary $d\Omega$

$$\frac{\partial \eta}{\partial t} + \nabla \cdot (H \mathbf{u}) = 0, \quad (2.1)$$

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + f \mathbf{e}_z \times \mathbf{u} = & -g \nabla \eta + \frac{\nu}{H} \nabla \cdot (H \nabla \mathbf{u}) \\ & + \frac{\tau^\eta - \tau^b}{\rho_0 H}, \end{aligned} \quad (2.2)$$

are solved for $\eta(\mathbf{x}, t)$, the surface elevation, and $\mathbf{u}(\mathbf{x}, t)$, the depth-averaged velocity. In these equations $H = h + \eta$ is the depth of the fluid, where $h(\mathbf{x})$ is the resting depth of the fluid, as shown in Figure 2.1, $\mathbf{e}_z$ is a unit vector in the vertical direction, g is the gravitational

acceleration, ν is the eddy viscosity, τ^η and τ^b are the surface and bottom stresses respectively and ρ_0 is the homogeneous density of the fluid.

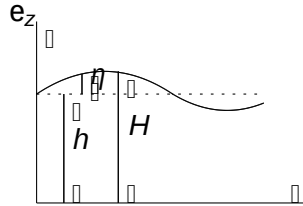


Figure 2.1: Depth of the fluid (1-D)

Equations (2.1) and (2.2) describe the development of an incompressible flow of constant density on the surface of the sphere. They are only relevant when the scale of the region of flow is much larger than the depth of the fluid, as in the oceans where the depth is much smaller in proportion to the size of the ocean. Equation (2.1) describes conservation of mass over the domain, and (2.2) describes the momentum balance.

2.2 The Stommel Model

The Stommel model is a simplification of these shallow water equa-

The shallow water equations of the previous section describe the motion of fluid on the surface of a sphere. The Stommel model simplifies this by considering the motion of fluid in Cartesian coordinates on an f -plane, where to first order the Coriolis parameter is constant.

Stommel [5] tried to understand the cause of the intensification of these boundary currents, by proposing the following

- considering the Coriolis parameter changing with latitude at a constant rate by letting $f = f_0 + \beta y$ (β -plane approximation.)
- including a bottom friction term, so keeping the higher order terms and the ability to satisfy conditions of no normal flow over the boundaries.

Then, also neglecting non-linear and diffusion terms, the governing equations of the Stommel model become

$$\frac{\partial \eta}{\partial t} + h \nabla \cdot \mathbf{u} = 0, \quad (2.3)$$

$$\frac{\partial \mathbf{u}}{\partial t} + (f_0 + \beta y) \mathbf{e}_z \times \mathbf{u} = -g \nabla \eta - \gamma \mathbf{u} + \frac{\boldsymbol{\tau}^\eta}{\rho h}. \quad (2.4)$$

In these equations, $\mathbf{x} = (x, y)$ is the spatial coordinate, h is the resting depth of the fluid which is assumed constant, f_0 is the reference value of the Coriolis parameter, β is the reference value of the Coriolis parameter first derivative in the y direction, $\mathbf{e}_z$ is a unit vector in the vertical direction, g is the gravitational acceleration, γ is a linear friction coefficient, ρ is the homogeneous density of the fluid, $\boldsymbol{\tau}^\eta$ is the wind stress acting on the surface of the fluid and ∇ is the two dimensional gradient operator. The domain Ω is the interval $[0, L] \times [0, L]$. Stommel assumed a simple form for the wind stress

$$\tau_x = -\tau_0 \cos \frac{\pi y}{y_n}, \quad 0 \leq y \leq L,$$

$$\tau_y = 0.$$

These equations are solved subject to no-normal flow boundary conditions, $\mathbf{u} \cdot \mathbf{n} = 0$ on $\partial\Omega$.

We note that there is an analytic solution to the Stommel equation, so we know that the solution moves westward to create a boundary layer at the western boundary. We might then think that we could just prescribe the mesh movement to be to the west for all time. The results for this idea are shown in Chapter 7.

2.3 Scale Analysis

We now use scale analysis to compare the important terms in the Stommel equations in relation to the other terms. This gives us an insight into the flow of the fluid before we solve the equations. It may also help us to determine possibilities for monitor functions that could be used to move the mesh. The parameters we will use for scaling are shown in Table 2.1.

The existing code that we are using solves the dimensionless shallow water equations. The dimensional variables are multiplied by a scale factor to make them dimensionless. If we choose scale factors carefully, we end up with a set of equations where all variables and coordinates are of order 1. Scaling is as follows:

$$x' = L_h x,$$

$$u' = U u,$$

$$h' = L_v h,$$

Parameter	Description	Scale
E	vertical surface elevation	1 m
U	horizontal velocity	0.1 m s ⁻¹
L_h		

We want these two terms to be of equal importance as it would not be sensible to neglect one of the two terms in the equation. Therefore in the code we choose $T = L_h E$

Chapter 3

Numerical Solution to the Stommel Problem

where the divergence term has been integrated by parts and the resulting boundary integral can therefore be removed as $\mathbf{u} \cdot \mathbf{n} = 0$ on $\partial\Omega$.

To make the finite element approximation, we use a mixture of conforming (P_1) and nonconforming basis functions (P_1^{NC}). The conforming nodes are on the vertices of the triangulation, whilst the nonconforming nodes are on the mid segment of the triangulation (Figure 3.1). This P_1^{NC} - P_1 finite element pair is very good for solving the shallow water equations [3]. This is due to the orthogonality of the shape functions, which allows us to reduce the computational cost of the scheme. The nonconforming elements are discontinuous everywhere, except at the mid segments. They are a compromise between continuous and discontinuous approximations.

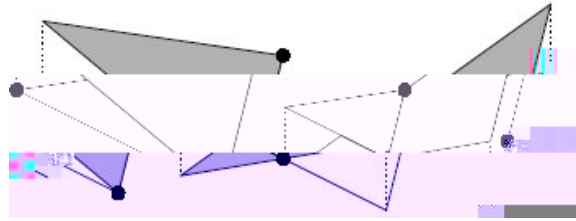


Figure 3.1: P_1 and P_1^{NC} basis functions

We replace $\mathbf{u}$ and η in (3.1) and (3.2) by their discrete approximations $\mathbf{u}^h$ and η^h as follows. Let

$$\eta \approx \eta^h = \sum_{i=1}^{N_V} \eta_i \psi_i$$

and N_S are the number of vertices and segments in the mesh. The nodal values are then calculated using the Galerkin method; replacing $\hat{\eta}$ by φ_i ($1 \leq i \leq N_V$), and $\hat{u}$ by $(\psi_j, 0)$ and $(0, \psi_j)$ ($1 \leq j \leq N_S$) in (3.1) and (3.2). This results in

$$\begin{aligned} \frac{d\eta_j}{dt} \int_{\Omega} \varphi_i \varphi_j d\Omega - h u_j \int_{\Omega} \varphi_i \psi_j d\Omega &= 0, \quad (3.3) \\ \frac{du_j}{dt} \int_{\Omega} \psi_i \psi_j d\Omega + (f_0 + \beta y_j)(\mathbf{e}_z \times \mathbf{u}_j) \int_{\Omega} \psi_i \psi_j d\Omega \\ + g \eta_j \int_{\Omega} \psi_i \varphi_j d\Omega + \gamma u_j d\Omega \\ + g \varphi_j d\Omega \end{aligned}$$

We now use the third order Adams Bashforth scheme to solve these equations for η and $\mathbf{u}$. This gives us two matrix systems

$$A_\eta \eta^{n+1} = \mathbf{b}_\eta,$$

$$A_{\mathbf{u}} \mathbf{u}^{n+1} = \mathbf{b}_{\mathbf{u}},$$

where

$$A_\eta^{i,j} = \int_\Omega \varphi_i \varphi_j d\Omega,$$

$$A_{\mathbf{u}}^{i,j} = \int_\Omega \psi_i \psi_j d\Omega,$$

and

$$\mathbf{b}_\eta^i = \int_j \int_\Omega \eta_j^n \varphi_i \varphi_j d\Omega + \frac{\Delta t}{12} \int_\Omega (23f_\eta^n - 16f_\eta^{n-1} + 5f_\eta^{n-2}) \varphi_i d\Omega,$$

$$\mathbf{b}_{\mathbf{u}}^i = \int_j \int_\Omega \mathbf{u}_j^n \psi_i \psi_j d\Omega + \frac{\Delta t}{12} \int_\Omega (23f_{\mathbf{u}}^n - 16f_{\mathbf{u}}^{n-1} + 5f_{\mathbf{u}}^{n-2}) \psi_i d\Omega.$$

These can be solved by inverting the mass matrices A_η and $A_{\mathbf{u}}$.

3.3 Computation

We now discuss how the entries of the mass matrices are computed to simplify the calculation of the solution.

3.3.1 Mass Matrix Entries

The entries of the mass matrices A_η and $A_{\mathbf{u}}$ for each element are

$$A_\eta^{i,j} = \int_{\Omega_e} \varphi_i \varphi_j d\Omega,$$

$$A_{\mathbf{u}}^{i,j} = \int_{\Omega_e} \psi_i \psi_j d\Omega.$$

So we have for each element

$$A_{\eta}^{i,j} = \frac{\Delta_{\text{area}}}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}, \quad (3.7)$$

$$A_{\mathbf{u}}^{i,j} = \frac{\Delta_{\text{area}}}{3} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

These are then assembled to form mass matrices for the whole domain.

An alternative to inverting these large matrices is to note that $A_{\mathbf{u}}$ is a diagonal matrix, so when computing the solution we can think of it as a vector $A_{\mathbf{u}}^{diag}$ containing all of the diagonal entries of $A_{\mathbf{u}}$. A_{η} however is not diagonal so it is not so simple to create a vector in this way. However, if we assume the mass of each element is concentrated at its nodes then the matrix can be converted into a diagonal matrix by lumping the off-diagonal entries in A_{η} onto the diagonal. Then this diagonal matrix can be represented by the vector A_{η}^{diag} .

3.3.2 Computing the solution

To compute the solution it is now easier to represent the two systems as one system

$$A^{diag} \begin{bmatrix} \vdots \\ \eta_j \\ \vdots \\ \mathbf{u}_j \\ \vdots \end{bmatrix} = \mathbf{b},$$

$\vdots$ $n+1$
 η_j
 $\mathbf{u}_j$

where A^{diag} contains all of the entries from A_{η}^{diag} and $A_{\mathbf{u}}^{diag}$, and $\mathbf{b}$ contains all of the entries from $\mathbf{b}_{\eta}$ and $\mathbf{b}_{\mathbf{u}}$. We can then compute the

solution for η_j and u_j simply by dividing b by A^{diag} .

3.4 Initial Conditions

Initially the solution is zero everywhere in Ω for the elevation and velocity fields. The movement of the solution is determined by the wind stress function which is set to

$$\tau^\eta = \begin{pmatrix} 0.2 \sin \left(y - \frac{L}{2} \right) \\ 0 \end{pmatrix}.$$

The initial mesh shown in Figure 3.2 is a structured mesh with equal sized elements

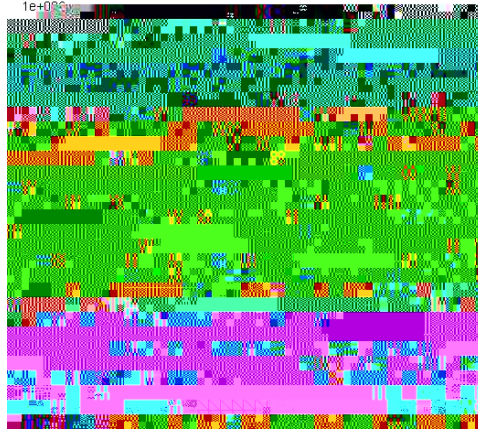


Figure 3.2: Initial Mesh

When using the third order Adams Bashforth scheme for the temporal discretisation of the Stommel equations, the time step that is required to keep the solution stable is $\Delta_t = 20$ seconds. We then run the model with this time step and let the model converge to a steady solution.

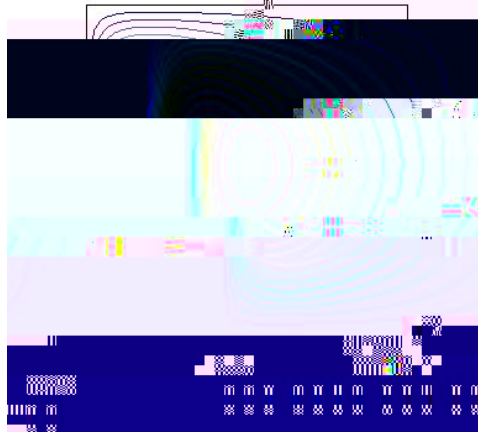


Figure 3.3: Stommel solution on a square domain

3.5 Stommel Solution

The results for the Stommel problem show a strong western intensification (Figure 3.3). The flow created by the wind stress is initially in the same direction as the wind stresses, from the west to the east in the top half of the region, and from the east to the west in the bottom half of the region. The flow starts to rotate around the centre of the region. Then because of the change in Coriolis parameter with latitude, the solution moves west creating the boundary current. The flow is southward for most of the region, and then returns in a swift northward jet on the western boundary. This motion idealises the Gulf Stream in which a narrow western boundary current returns the southward interior flow to the north. If Stommel had not included the Coriolis term in his equations, the solution would be centred in the middle of the domain with no westerly movement.

This solution leads us to explore a mesh that moves with the flow.

As the flow moves towards the west, we wish to move the mesh with it to increase the resolution in the developing boundary layer.

Chapter 4

Moving Mesh Adaptation

To adapt the mesh as the solution changes, we aim to find a velocity for each node in the mesh. This section will look at the principles behind the moving mesh, and how the mesh adaptation principle is formulated to calculate the mesh velocities.

4.1 Moving Mesh Methods

Moving mesh methods, or r-adaptive methods, move the positions of a fixed number of nodes in the mesh so that the nodes follow a feature or become concentrated in certain areas of the solution. In principle these methods are easy to implement and fixed mesh models can be easily adapted to incorporate moving adaptive meshes. As there is no addition and deletion of nodes, difficulties in restarting the time integration procedure are avoided. Another benefit is the ability of the mesh to move in a quasi-Lagrangian manner. If the mesh points were to move with the flow velocity, this would lead to reduced transport velocity in relation to the mesh, with the possibility of using large time

steps.

There are also drawbacks to these methods. The number of nodes is constant, which may not always be optimal because the level of dynamic behaviour may vary during the simulation. There is also the possibility that inappropriately shaped or tangled meshes may occur.

Using the Reynolds Transport Theorem we can change this Lagrangian form of the conservation principle into an Eulerian form. This Theorem states that the rate of change of the integral of a property, N , over a moving control volume is equal to the integral of the rate of change of N over the control volume plus the net flux of N through the control surface. So we can rewrite (4.2) as

$$\int_{\Omega_c} \frac{\partial(h + \eta)}{\partial t} d\Omega + \int_{\partial\Omega_c} (h + \eta) \dot{\mathbf{x}} \cdot \mathbf{n} d\Gamma = 0,$$

where $\dot{\mathbf{x}}$ is the boundary velocity. As h is constant it can be removed from the first integral. Also, as η is small compared with h , we replace $h + \eta$ by h in the second integral. Then using Gauss' Theorem on the second integral we obtain

$$\int_{\Omega_c} \frac{\partial \eta}{\partial t} d\Omega + \int_{\Omega_c} \nabla \cdot (h \dot{\mathbf{x}}) d\Omega = 0, \quad (4.3)$$

where $\dot{\mathbf{x}}$ is any velocity consistent with the boundary velocity. If we use the continuity equation (2.3) from the shallow water equations, we can replace $\frac{\partial \eta}{\partial t}$ in (4.3) to give

$$\int_{\Omega_c} \nabla \cdot (-hu + h\dot{\mathbf{x}}) d\Omega = 0.$$

It follows that

$$\nabla \cdot (-hu + h\dot{\mathbf{x}}) = 0,$$

and so

$$\nabla \cdot (h\dot{\mathbf{x}}) = \nabla \cdot (hu).$$

This equation is not sufficient to determine $\dot{\mathbf{x}}$ but if we use Helmholtz decomposition Theorem, and let $\nabla \times \dot{\mathbf{x}} = 0$, we have that

$$\dot{\mathbf{x}} = \nabla p, \quad (4.4)$$

where p is the mesh velocity potential. We then have

$$\nabla \cdot (h \nabla p) = \nabla \cdot (hu), \quad (4.5)$$

which has a unique solution, given p on the boundary or given $\frac{\partial p}{\partial n}$ on the boundary and one prescribed value of p .

Chapter 5

Solving for the Mesh Velocities

5.1 Finite Element Formulation

We need to formulate a weak form of (4.5) so it can be solved using the finite element method. First we introduce a test function w , a partition of unity, into (4.1) giving

$$\int_{\Omega_c} w(h + \eta) d\Omega = C.$$

Then using a generalisation of the Reynolds Transport Theorem, and setting $\dot{x} = \eta p$ as before,

$$\int_{\Omega} w \eta \cdot (h \eta p) d\Omega = \int_{\Omega} w \eta \cdot (hu) d\Omega.$$

Integration by parts on both sides gives

$$-\int_{\partial\Omega} wh \eta p \cdot n d\Gamma + \int_{\Omega} h \eta w \cdot \eta p d\Omega = -\int_{\partial\Omega} wh u \cdot n d\Gamma + \int_{\Omega} h \eta w \cdot u d\Omega. \quad (5.1)$$

The boundary terms then cancel as they are either zero on the inner boundaries, or $\eta p \cdot n = u \cdot n = 0$ on the outer boundary. This condition

means that the mesh velocity has no component in the outward normal direction to the boundary, and so none of the mesh or the fluid leaves the boundary. We can also divide both sides by the constant h . The weak form (5.1) can then be expressed in finite element form by letting the continuous functions w and p be represented by the piecewise functions φ and P respectively, where φ is the P_1 basis function described in section 2.2. We then represent $P = \sum_j P_j \varphi_j$. The finite element form is therefore

$$\sum_{\Omega} \varphi_i \cdot \sum_{\Omega} \varphi_j d\Omega P_j = \sum_{\Omega} w \cdot u d\Omega,$$

which can be written in matrix vector form

$$K P = f, \quad (5.2)$$

where K is a stiffness matrix and f is a load vector.

5.1.1 Stiffness Matrix Entries

On the left hand side of (5.2), we have a stiffness matrix K . The entries of this matrix for each of the elements Ω_e are given by

$$K_{ij}^e = \sum_{\Omega_e} \varphi_i \cdot \varphi_j d\Omega.$$

As φ_i and φ_j are constants, we can take these out of the integral.

This then gives

$$K_{ij}^e = \varphi_i \cdot \varphi_j \sum_{\Omega_e} d\Omega$$

where

$$\sum_{\Omega_e} d\Omega = \text{area of element} \equiv \Delta_{\text{area}}.$$

We also have that $\varphi_i = |\varphi_i| \mathbf{n}_i$, where $\mathbf{n}_i$ is the normal vector in the direction of the node i .

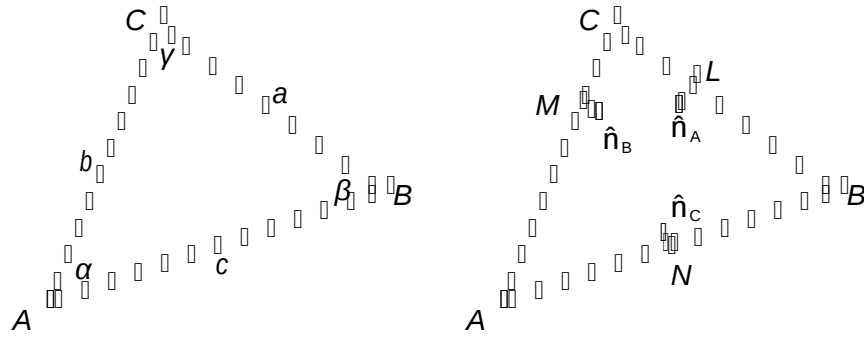


Figure 5.1: Dimensions of an element in the mesh

Looking at the element as in Figure 5.1, if we consider φ_B and φ_C , then

$$\begin{aligned} \varphi_B \cdot \varphi_C &= |\varphi_B| |\varphi_C| \mathbf{n}_B \cdot \mathbf{n}_C \\ &= 1.3130 \text{Td}[(n)] \text{TJC} \end{aligned}$$

and

$$\Delta_{\text{area}} = \frac{ba \sin \gamma}{2}.$$

Therefore,

$$\begin{aligned} f_B^e &= \frac{ba \sin \gamma}{2a^2 \sin^2 \gamma} (x_A - x_L, y_A - y_L) \cdot \sum_{k=1}^3 \mathbf{u}_k \psi_k, \\ &= \frac{b}{2a \sin \gamma} (x_A - x_L, y_A - y_L) \cdot \sum_{k=1}^3 \mathbf{u}_k \psi_k. \end{aligned}$$

The other entries in $\mathbf{f}$ are calculated similarly and so we have

$$f_i^e = \Delta_{\text{area}} |\boldsymbol{\varphi}_i| \hat{\mathbf{n}}_i \cdot (\mathbf{u}_1 + \mathbf{u}_2 + \mathbf{u}_3),$$

as $\mathbf{u}_i \psi_i = \mathbf{u}_i$ for all i . This is then assembled over the whole domain with other load vector which are calculated similarly.

5.1.3 Mesh Velocities

Once the stiffness matrix and load vector have been computed, the system (5.2) can then be solved to find the mesh velocity potential, p , on each node using a conjugate gradient solver. Then we can find the mesh velocities. To get the mesh velocities, we need to find the gradient of the potential velocity at the nodes of the mesh as in (4.4).

This may be computed using a finite element formulation as follows. Multiplying by a test function w and integrating over Ω we get

$$\int_{\Omega} w \dot{\mathbf{x}} d\Omega =$$

which can then be solved for $\dot{X}_j$ using a conjugate gradient solver.

However, we note that the integral on the left hand side of the equation gives the same mass matrix as in (3.7). To simplify the calculations this could again be lumped to give $\frac{\Delta_{\text{area}}}{3} I$ for each element, where I is the identity matrix. On the right hand side the integral is equal to $\frac{\Delta_{\text{area}}}{3}$.

Therefore we have that

$$\sum_j \dot{X}_j = \sum_j P_j \varphi_j.$$

To simplify this further, we consider the velocity computed on each element, which we will call the element velocity,

$$\dot{x}_{\Omega_e} = \sum_{j=1}^3 P_j \varphi_j.$$

However, this way of simplifying the calculation of the mesh velocity leads to a lot of averaging which may compromise the values of the mesh velocities. This is not as serious as it seems, since the main solver remains the Stommel equations.

elements as follows:

$$\dot{\mathbf{x}}_i = \frac{\sum_{e=1}^E |\Omega_e| \dot{\mathbf{x}}_{\Omega_e}}{\sum_{e=1}^E |\Omega_e|}.$$

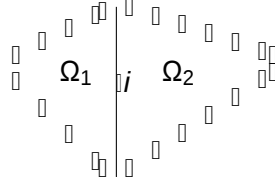


Figure 6.2: Averaging the element velocities to get the mesh velocities on the mid segments

These velocities can now be used to modify the solution.

6.4 Moving the Solution

When the mesh has been moved, we need to modify the solution so that it relates to the correct position in the mesh. After including the mesh adaptation procedure an extra term is required in each of the governing equations of the Stommel model as shown below.

By the chain rule for the conservation equation (2.3),

$$\frac{\partial \eta}{\partial t}_m = \frac{\partial \eta}{\partial t}_s + \dot{\mathbf{x}} \cdot \nabla \eta, \quad (6.2)$$

and similarly for the momentum equation (2.4),

$$\frac{\partial \mathbf{u}}{\partial t}_m = \frac{\partial \mathbf{u}}{\partial t}_s - \dot{\mathbf{x}} \cdot \nabla \mathbf{u}. \quad (6.3)$$

where m denotes the moving solution and s denotes the stationary solution.

The extra terms are advection terms, which counterbalance the movement of the mesh, and make sure that the change in mesh does not also change the solution.

Formulating the new terms using the finite element method with the same test functions as described in Chapter 3, gives, for the extra term in (6.2)

$$-\int_{\Omega} \varphi_i \dot{\mathbf{x}} \cdot \nabla \eta d\Omega, \quad (6.4)$$

and for (6.3)

$$-\int_{\Omega} \varphi_i \dot{\mathbf{x}} \cdot \nabla \mathbf{u} d\Omega. \quad (6.5)$$

We can then include (6.4) in f_{η} and (6.4) in $f_{\mathbf{u}}$, in equations (3.5) and (3.6) respectively, at the time step when the mesh is moved.

Chapter 7

Numerical Simulations and Results

As mentioned before, there are many methods for moving the mesh. We now look at several tests for moving the mesh. We set the mesh velocities:

- in a westward direction, given that we know the solution moves to the west.
- according to the conservation principle described in Chapter 4,
- equal to the flow velocities,
- a mixture of the two above tests.

For each case, we put these velocities into equation (6.1) to move the mesh, and also move the solution as described in Chapter 6.

7.1 Simple westward mesh movement

We start with a special example of setting the mesh velocities in the same direction as we know the solution is moving. We use the following mesh velocity equation

$$\dot{x} = \begin{bmatrix} -\alpha x(1-x) \\ 0 \end{bmatrix},$$

with different values of α . This is a simplified mesh movement, as we already know the direction that the solution will take. In other applications, we may not know how the solution evolves over time, and therefore would not be able to make this kind of assumption over which direction the mesh velocities should be travelling in.

7.1.1 Results for simple westward mesh movement

For all values of $\alpha > 0$ the mesh moves towards the west with varying speeds. When $\alpha = 0.5$, the mesh moves westwards very rapidly, even before the solution starts to move. This leads to instability in the mesh due to very narrow elements in the western region of the mesh and after 1693 time steps the solution blows up. Before this happens, however, the solution is not distorted by the mesh movement.

This leads us to look at smaller values of α to see if the solution remains stable as the mesh changes. Using $\alpha = 0.01$ gives better results in that, as well as there being no distortion in the solution, the solution remains stable up to maximum of 50,000 time steps that the solution is run for. This is not to say that it will always be stable for this value of α . The solution will eventually blow up as the elements in the western side of the mesh become very narrow, but takes longer to do

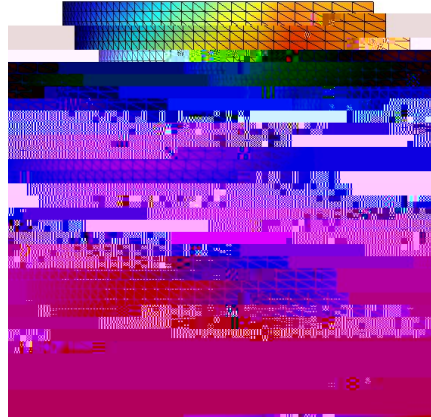


Figure 7.1: Solution after 1000 time steps with $\alpha = 0.5$.

so as $\alpha \rightarrow 0$.

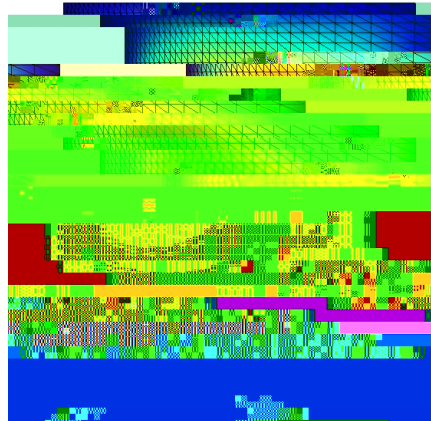


Figure 7.2: Solution after 50,000 time steps with $\alpha = 0.01$.

This simplified mesh movement is very easy to implement, but is a little crude. It is also not straightforward to move the mesh at a similar speed as the solution, and some sort of guess work is needed to determine the value of α to use to gain the best results. We know that the solution moves westward and a boundary layer forms on the

diagonal band has highest elevation in the centre of the diagonal, with decreasing elevation on either side.

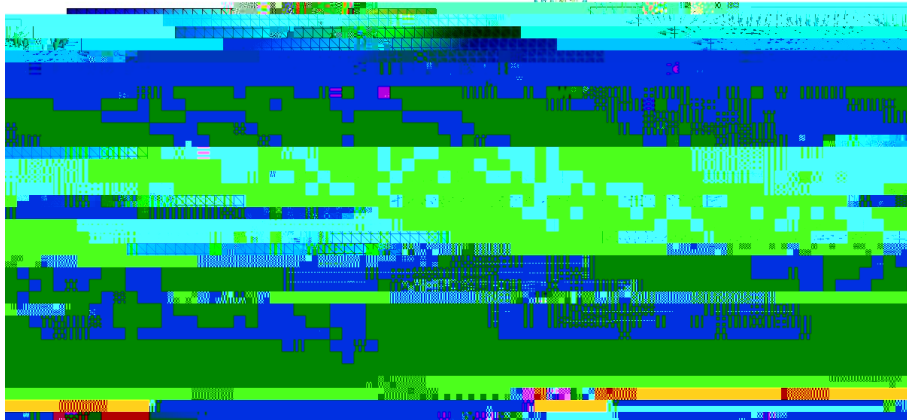


Figure 7.3: Solution after 1000 time steps using conservation principle

Note that in Figure 7.3 the velocities are not actually causing the mesh to leave the region. The length of the arrows has been exaggerated to show the direction of the mesh movement.

Using the conservation principle, we would expect patches of elements to be smaller in the area of high elevation, and larger in the area of low elevation. The results show mesh velocities directed into the high elevation in the north east and south west of the region. This suggests that the elements are being pushed together, and therefore reducing in size. In the low elevation areas, we see the mesh velocities moving away from these regions, suggesting that the elements are being pulled apart, or increasing in size. This is what we expected to happen when we used this principle.

As the elevation solution settles down into a rotating movement, the regions of high elevation are in the centre of the rotation, and the

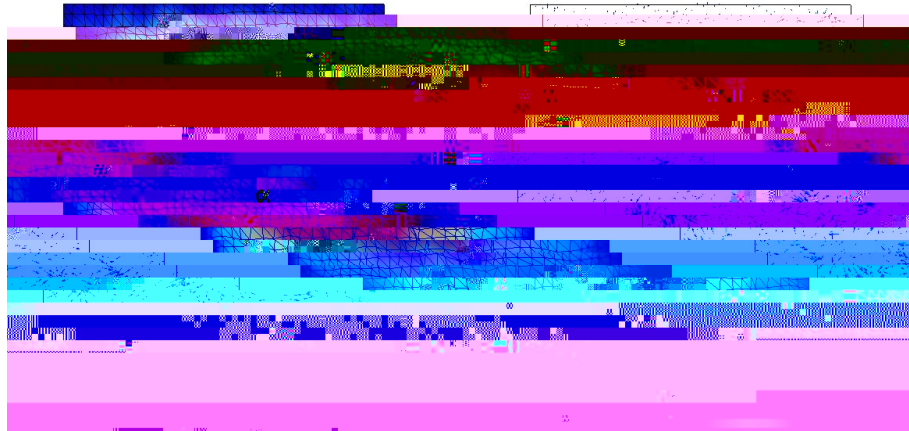


Figure 7.4: Solution after 25000 time steps using conservation principle

low elevation areas are near the boundaries. Because of the volume preservation, we would expect the patches of elements to be larger around the edges, and smaller in the centre of the rotation. Then as the solution moves westwards, we would expect the patches of elements in the western region to get smaller with the increased elevation, and patches of elements in the eastern region to grow. The results we get show that in the centre of the rotation, and around the boundaries, the patches are well preserved over time see Figure 7.4 .

However, in the rest of the domain the conservation principle does not seem to have the same effect. This may be due to the number of nodes in the mesh being constant. This means that the patches that

narrow elements that are generated in parts of the mesh lead to the solution blowing up.

7.3 Flow based mesh movement

A purely Lagrangian method is to use the mesh velocity to be equal to the flow velocity. This would be the solution to the conservation principle described in Chapter 4 if we had imposed that $\nabla \times \dot{\mathbf{x}} = \nabla \times \mathbf{u}$. We have that

$$\dot{\mathbf{x}} = \mathbf{u}.$$

We expect the mesh to move round with the flow in a clockwise direction. This may not be ideal as the flow velocities on the boundary regions, particularly the western boundary, are much larger than the flow velocities on the central region. This may lead to an inappropriate tangled mesh.

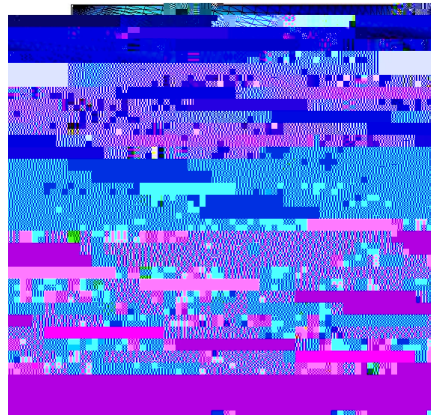


Figure 7.5: Solution after 3000 time steps using flow based mesh movement

whereas the flow velocity in the centre of the domain is comparable to the mesh velocities. We obtain the same narrow elements on the boundaries as before which cause the solution to blow up. Putting $\alpha = 0.1$ gives slightly better results, but the flow velocities are still overpowering the conservation velocities, and again the solution blows up.

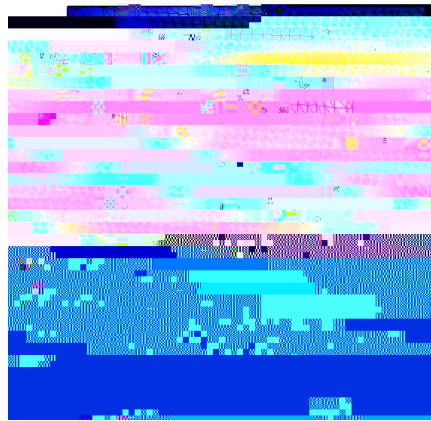


Figure 7.6: Solution after 9000 time steps using combined conservation/flow based mesh movement

By decreasing α the solution tends towards that of the conservation principle solution as we would expect. The solution also becomes more stable with the reduction in α .

Chapter 8

Conclusion

This dissertation has considered the possibility of incorporating a moving mesh method into an existing model of a special form of the shallow water equations, the Stommel model.

First we looked at how Stommel formed equations that describe the formation of a strong western boundary current such as the Gulf Stream. Then scale analysis was performed on these equations to determine the dominant terms. The result of this was that the rotational property of these velocities is an important part of the solution. This led us to investigate whether equating the mesh velocity to the flow velocities was a good strategy for moving the mesh. We saw in Chapter 7 that this resulted in a rotational mesh which became unstable very quickly. This is a result of the boundary nodes being restricted to remain on the boundary, as there is only flow velocity tangential to the boundary. Therefore the boundary nodes were able to move along the boundary which led to a bunching of nodes in the four corners of the domain.

We then discussed how the Stommel model was formulated using

finite elements following [1]. These finite element equations were then discretised in time using an explicit Adams Bashforth method of order 3. The resulting mass matrix on the left hand side of the equation was lumped to enable us to avoid inversion of a matrix. Explicit methods, however, are not as stable as implicit methods, and a small time step of $\Delta t = 20$ seconds was required to keep a stable solution over the simulation. The initial conditions and mesh were then stated, followed by a description of how the solution evolves to form a western boundary layer over time. This formation of the boundary layer is one motivation for including a moving mesh procedure into the model.

In Chapter 4 a description of moving mesh methods was given, along with the benefits and drawbacks of using moving meshes. Then the use of a monitor function was introduced as in [8]. This dissertation has described the use of a conservation principle containing a monitor function together with a prescribed vorticity. This then gives rise to a conservation law from which we can obtain the mesh velocity. The conservation principle was applied to the chosen monitor function, fluid depth. In a finite element context, this prescribed that the volume of fluid in a patch of elements is constant for all time. The Reynolds Transport Theorem was then used along with Gauss' Theorem to change the equation from a Lagrangian to a Eulerian form inolds Tr 117T4phe1J0ks

and use the equation for the mesh

t

=

each element. Then the equations were solved for the velocity potential using a conjugate gradient solver. A description of how $\dot{\mathbf{x}} = -\nabla p$ could be solved using finite elements was then given. We then simplified the calculation by lumping the resulting mass matrix, so the use of a sparse matrix solver is not required. A further simplification was to find a velocity for each element which is constant over the whole element. This was then be interpolated onto the mesh nodes for mesh movement, and onto the mid-segment nodes for advecting the solution to compensate for the mesh movement. Whilst this simplification makes the computation easier, the process of calculating the mesh velocity for each element, and then having to interpolate this onto the mesh may have caused a slight loss of accuracy in the mesh velocities, but should not alter the solution.

The interpolation equations for generating nodal, and mid-segment velocities from the element velocities was described in Chapter 6. Averages were weighted according to the areas of the elements to try to obtain more accurate interpolations to the node velocities and mid-segment velocities. This averaging is not ideal as we could have calculated the mesh velocity straight from the finite element form in Section 5.2, and then used the average of the end points of a segment to determine the mid-segment velocities. However, use of computational time was an issue because of the small time step required for a stable solution. Therefore we chose the simplifications over the perhaps slight increase in accuracy. The equations of mesh movement were then stated. The mesh velocity was not calculated at every time step as the reduced time step means that the solution did not move very quickly over the simulation. The extra terms that arise in the solution

equations at the time step when the mesh movement occurs are stated, followed by the finite element formulation of these terms.

Chapter 7 discussed the results of numerical simulations for a number of test cases for different mesh movement ideas. These ideas were based on the ideas generated throughout the dissertation. The first was a special case, as the Stommel model has an analytic solution so we know that the elevation field flows westwards, considering purely westward movement of the mesh. This gave a very simple computation of the mesh velocity, but was also very crude, and the solution blew up because of the gathering of very narrow elements on the western boundary. A modified version of this idea is discussed in Chapter 9. The main idea in this dissertation was to conserve the depth function over a patch of elements with a prescribed mesh velocity. The results show that this has been a promising start to describing a moving mesh procedure for these equations. The time restriction for this dissertation means that other monitor functions or curl properties of the depth monitor function have not been able to be investigated. Some ideas for other curl properties or monitor functions are in Chapter 9. Another test was to set the mesh velocity to be equal to the flow velocity, equivalent to choosing the vorticity of the mesh to be equal to the vorticity of the fluid. This resulted in a very twisted mesh and an unstable solution. The final idea was to take a linear combination of the conservation based velocities and the flow velocity. The results showed that the flow velocity overpowered the conservation velocities, and a very small ratio of flow to conservation based mesh movement was required for the solution to remain stable for any length of time.

Chapter 9

Further Work

The scope of work in this dissertation is limited by time available to carry out the work. Given more time, there are other areas that would be interesting to investigate. Several of these are given below.

More complex equations for western mesh movement Perhaps an improved equation to use to move the mesh west would include a term that would allow for the size of the boundary layer, using the analytic solution to the Stommel equations. This could be used to make sure the size of the elements in this boundary layer do not become too small. This would of course only apply for the Stommel problem and not for general equations.

Other monitor functions The monitor function described in this dissertation is just one of many that could be used as a basis for mesh movement. Use of different monitors that may give better results could be investigated. Other monitor functions that could be used include

- vorticity or potential vorticity

- fluid surface gradient,

$$|\nabla \eta|$$

- area on the surface of the solution,

$$\frac{1}{1 + (\nabla \eta)^2}$$

The latter two would be more suitable for moving points in areas with large gradients, as in a boundary layer

Differing curl properties of mesh velocity We could investigate other curl properties using $\nabla \times \dot{\mathbf{x}} = \nabla \times \mathbf{q}$, where $\mathbf{q}$ is some variable used to determine the mesh velocity, along with a monitor function. In this case, $\dot{\mathbf{x}}$ is not only the gradient of a potential function, but

$$\dot{\mathbf{x}} = \nabla p + \mathbf{q}.$$

For the monitor function we have used, we have looked at the cases when; $\nabla \times \dot{\mathbf{x}} = 0$ taking $\mathbf{q} = 0$, $\nabla \times \dot{\mathbf{x}} = \nabla \times \mathbf{u}$ where $\mathbf{q} = \mathbf{u}$, and $\nabla \times \dot{\mathbf{x}} = \alpha \mathbf{u}$ which is equivalent to taking $\mathbf{q} = \alpha \mathbf{u}$.

Modelling tides with a moving mesh The possibility of using a moving mesh to show the rising and falling of the coastlines with tides could also be looked at. This could be done by prescribing the surface elevation on the boundaries to rise and fall over time. The boundary condition would need to be changed from $\mathbf{u} \cdot \mathbf{n} = 0$ on $\partial\Omega$ to $\mathbf{u} \cdot \mathbf{n} = \dot{\mathbf{x}} \cdot \mathbf{n}$ to enable the mesh to move in and out of the domain.

Grid generation

Bibliography

- [1] Hanert, E., 2004. Towards a Finite Element Ocean Circulation Model. PhD Thesis.
- [2] Hanert, E., Le Roux, D.Y., Legat, V., Deleersnijder, E., 2005. An efficient eulerian finite element method for the shallow water equations. *Ocean Modelling* 10, 115-136.
- [3] Hua, B.L., Thomasset, F., 1984. A noise-free finite element scheme for the two-layer shallow water equations. *Tellus* 36A, 157-165.
- [4] 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 Modelling* 10, 95-113.
- [5] Stommel, H., 1948. The westward intensification of wind-driven ocean currents. *Trans. Amer. Geophys. Union*, 29, 202-206.
- [6] Tang, T., 2005. Moving mesh methods for computational fluidns. *Amer. Gen1(n)28H*

- [8] Wells, B.V., 2004 A Moving Mesh Finite Element Method for the Numerical Solution of Partial Differential Equations and Systems. PhD Thesis.