Monge-Ampére based moving mesh methods for numerical weather prediction, with applications to the Eady Problem.

C.J. Budd (1), M.J.P. Cullen (2) and E.J. Walsh (3)

(1) Centre for Nonlinear Mechanics, University of Bath, UK, BA2 7AY,
(2) Met Office, FitzRoy Rd., Exeter, UK, EX1 3PB,
(3) Simon Fraser University, Burnaby, BC, Canada, V5A 1S6

November 6, 2012

Abstract

We derive a moving mesh method based upon ideas from optimal transport theory which is suited to solving PDE problems in meteorology. In particular we show how the Parabolic Monge-Ampere method for constructing a moving mesh in two-dimensions can be coupled successfully to a pressure correction method for the solution of incompressible flows with significant convection and subject to Coriolis forces. This method can be used to resolve evolving small scale features in the flow. In this paper the method is then applied to the computation of the solution to the Eady Problem which is observed to develop large gradients in a finite time. The moving mesh method is shown to work and be stable, and to give significantly better resolution of the evolving singularity than a fixed, uniform mesh.

Keywords: Moving mesh method, Monge Ampere, Numerical Weather Prediction, Eady Problem

1 Introduction

Successful atmospheric numerical modelling requires solving a set of partial differential equations (PDEs) to obtain an objective forecast of the future state of the atmosphere. These equations describe the evolution of variables such as temperature, wind speed, humidity and pressure, which together define the state of the atmosphere. However the physical processes that occur within the atmosphere are at a physical scale which is too small to be directly resolved by current weather forecasting or climate models. At present the finest resolution weather forecast models used operationally have a grid length of 16 km globally and about 1.5 km over the UK, and will only accurately resolve features larger than about five grid lengths. Hence, features smaller than around 10 km are not well resolved. Given that major thunderstorms and other meteorological phenomena may often be 5-10km, or less, across there is a strong incentive to develop adaptive numerical methods in order to improve the accuracy and duration of future meteorological forecasting.

Adaptive methods for numerical weather prediction usually take the form of h-adaptive procedures for progressively refining the computational mesh in the physical domain. In [2], [1] such methods are described in the context of finite element calculations of atmospheric flows and an a-posteriori error estimate is used as an indicator for where mesh refinement is necessary. Whilst such h-adaptive methods have a good track record for computation, they suffer from certain implementation issues, such as the need for complex data structures to allow for the additional mesh points as the grid is refined, and also the constantly changing sparsity structure of the various matrices used in the computation. These issues are especially important for the very large computations typically encountered in meteorology. In contrast, moving mesh (r-adaptive) methods take a *fixed* number of mesh points and concentrate them in regions where the underlying problem has structures evolving on small length scales [24]. Typically such methods use logically rectangular meshes leading to much simpler data structures and constant matrix sparsity structures. Such methods have been considered by Tang and his co-workers [32], [33], in which the moving mesh method has been combined with a finite volume method and all computations performed in the physical domain. Another interesting work by [26] combines moving mesh methods with a semi-implicit non-oscillatory forward-in-time solver to simulate multiscale atmospheric flows. An attractive feature of moving mesh methods is that as they typically map a uniform (rectangular) mesh in a computational domain into the logically rectangular mesh in the physical domain, all computations on solving the underlying PDE posed in the physical domain can be performed on the uniform mesh in the *computational domain* by using explicit information on the derivatives of the coordinate change from the computational to the physical domain. This allows the possibility of using fast (eg. spectral) methods and consistent linear algebra methods in the computational domains. This approach was used with success by Ceniceros and Hou [11] and by [33] in computations of evolving structures in the Bousinnesq equations. In both cases a variational approach, closely related to the method originally developed by Winslow [36], was used to move the mesh points by solving an associated coupled system of (moving mesh) partial differential equations describing the map from the computational to the physical domain. In this paper we consider applying moving mesh methods to meteorological problems. These have special features which make the use of such methods challenging. In particular: near incompressibility conditions, the existence of conservation and balance laws and the effects of Coriolis forces. We will consider how to meet these challenges for a general class of problems, and then will look at the detailed example of using a moving mesh method to find an approximate solution of the Eady Equations.

In a series of earlier papers [7],[6],[8], we have considered the use of moving mesh methods based on the method of *Optimal Transport*. These methods compute a scalar *mesh potential* $Q(\xi, \eta)$ from a computational to a physical domain, and then determine the mapping $(x(\xi, \eta), z(\xi, \eta))$ from the computational to the physical domain, by taking the gradient (with respect to (ξ, η)) of Q. The mesh potential Q satisfies a fully nonlinear *Monge-Ampere* equation. In the papers above, Q is determined by a relaxation method, solving a *Parabolic Monge-Ampère* (PMA) type equation, the solution of which evolves towards that of the Monge-Ampere equation. A similar approach is used by Finn et. al. [17] in which the function Q is computed by using a Newton-Raphson based algorithm coupled to a multi-grid solver in the computational domain.

The Optimal Transport based procedures to find a computational mesh change the grid globally at any refinement, and have a number of advantages over other (more local) mesh relocation methods. The meshes derived from such methods can be shown to have good regularity properties [8], they are relatively easy to compute (as only a scalar problem need ever be solved regardless of the physical dimension of the problem), and unlike many other methods, as a direct consequence of the regularity properties of the solutions of the Monge-Ampere equation, the grid avoids problems associated with large distortions and possible tangling. Thus Optimal Transport based methods for mesh generation have many potential advantages as robust and reliable mesh generators for CFD and meteorological applications.

In the papers cited above, the equation for the mesh potential is coupled to (for example) a system of parabolic PDEs in order to resolve the small scales associated with blow-up type phenomena. However, the optimal transport methods have so far not been tried on large scale convective equations and conservation laws associated with the almost incompressible flow models encountered in large scale meteorology. The purpose of this paper is to do precisely this, and we show that such an extension is possible, though by no means straightforward. Throughout this paper the terms 'convection' and 'convective' will be used to refer to transport flow rather than buoyancy-driven vertical motion, as is the convention in meteorological literature. We also note that although fully compressible equations are now favoured in Numerical Weather Prediction, they are solved using a semi-implicit method to ensure that the solutions remain almost incompressible. Therefore use of a simplified incompressible problem is a suitable testbed for methods that will be suitable for the semi-implicit treatment of compressible flows.

When solving a PDE on an evolving mesh two distinct processes are involved. One is the solution of the PDE and the other the evolution of the mesh. These two are typically coupled through an interpolation procedure which maps the solution of the PDE onto the evolving mesh. Special care has to be taken when implementing such a moving mesh method in the context of meteorological problems so that the evolution and interpolation procedures preserve both the incompressibility and balance conditions associated with large scale atmospheric flows. We will show in this paper that this can be done successfully in the context of a general meteorological computation for an incompressible flow by carefully coupling the interpolation procedure with a pressure correction method. Indeed, we show that yields an effective and efficient adaptive moving mesh method for a potentially wide class of meteorological problems. As the potential class for problems for which this method is applicable is large, all with different details of implementation, we do not go into great detail for the general class. However, we will consider an important example which we then describe in more detail, namely we will apply the moving mesh method to the Eady equations discretised on a Charney-Phillips mesh. The Eady equations are a system derived from the Euler equations with Coriolis forces in a vertical cross-section. The Eady equations have a natural asymptotic limit which is given by the semi-geostrophic equations, and in analogy with these equations it is known [12]) that the Eady equations can develop strongly localised features (fronts) in a finite time. These features are poorly resolved on a fixed mesh and such computations can also lead to instabilities. We show in contrast, that the moving mesh method derived in this paper is very effective for these problems and correctly resolve the evolving front both up to, and potentially beyond, the formation of the singularity.

The layout of the remainder of this paper is as follows. In Section 2 we describe the main features of the PMA based moving mesh methods based on ideas from the theory of Optimal Transport. In Section 3 we consider coupling these methods to the general case of a convection dominated incompressible flow of the form which arises in meteorological applications. This procedure will use a combination of the PMA method with the conservative interpolation methods proposed by [33]. In Section 4 we introduce the Eady equations and consider solution methods for these based on fixed Charney-Phillips meshes and demonstrate that these poorly resolve the solution. In Section 5 we derive a moving mesh method for the same problem, also based on a (moving) Charney-Phillips mesh and give the results of the computations on this, showing the major improvement over those on a fixed mesh. Finally in Section 6 we draw some conclusions from this work.

2 Optimal Transport based moving mesh methods

In this section we summarise some of the results presented in earlier papers on the construction of moving meshes appropriate for solving time-evolving partial differential equations by using optimal transport (Parabolic Monge-Ampére) based methods. We then give a new application of these mesh generation methods to a meteorological problem, namely the problem of the collapse of a large bubble of cold air. It is assumed that the solution of the underlying PDE is *known*. In the next section we consider the question of coupling the mesh to the solution of the PDE.

2.1 Moving mesh geometry

We consider the situation of having a *physical domain* Ω_P in which we pose the partial differential equations describing the meteorological phenomena. For convenience in this paper we will take Ω_P to be a two-dimensional domain with physical coordinates (x, z). Typically we will be considering 2D vertical slice models of the atmosphere in which x is related to Longitude, with the solutions and meshes satisfying a periodic boundary condition, and z represents height. (The methods and applications of the techniques described in this paper also extend naturally, and easily to three dimensional problems, but we will not consider these here).

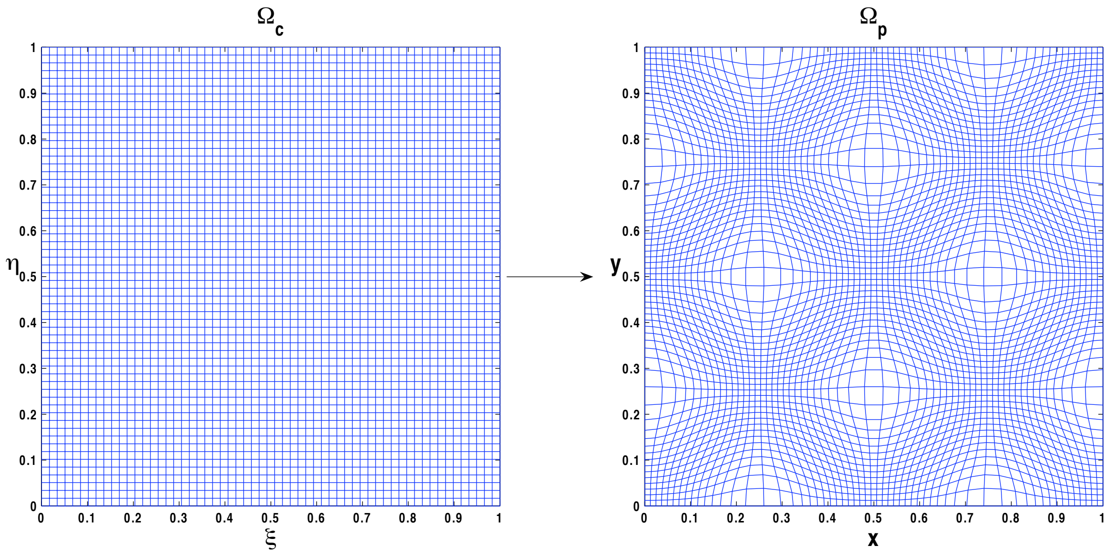

Figure 1: An example of a mapping from the computational to the physical domain in which a uniform mesh is mapped to a non-uniform one.

2.2 Optimal transport and moving mesh equations

The basis of the moving mesh approach is that a fixed set of mesh points, typically arranged in a uniform rectangular mesh in a *computational domain* Ω_C, with computational co-ordinates $\boldsymbol{\xi} = (\xi, \eta)$, is continuously mapped to the physical domain Ω_P, with physical co-ordinates $\mathbf{x} = (x(\boldsymbol{\xi}, t), z(\boldsymbol{\xi}, t)) \equiv \mathbf{F}(\boldsymbol{\xi}, t)$. This is illustrated in Figure 1. Furthermore, for all t,

$$\mathbf{F} : \partial\Omega_C \rightarrow \partial\Omega_P.$$

We shall assume that the map $\mathbf{F}$ is invertible, smooth and periodic (in x). Accordingly, we may define the Jacobian matrix of the map and its determinant (simply referred to as the Jacobian) by

$$\mathbf{J}(\mathbf{x}) = \frac{\partial(x, z)}{\partial(\xi, \eta)} = \begin{bmatrix} x_\xi & x_\eta \\ z_\xi & z_\eta, \end{bmatrix}, \quad \text{with} \quad J = \det(\mathbf{J}),$$

and assume that $\mathbf{J}$ is non-singular. The *local* stretching of the coordinate map is given by $J > 0$ and typically we seek to control this, so that the mesh points in Ω_P are *dense* when (for example) the solution of the PDE has small length scales (such as in an evolving front). Typically we can control this via a scalar *monitor function* $M(\mathbf{x}, t) > 0$ so that the mesh equidistributes M in the manner

$$M(\mathbf{x}, t)J(\mathbf{x}) = \theta(t), \quad \text{where} \quad \int_{\Omega_P} M(\mathbf{x}, t)\, d\mathbf{x} = \theta(t). \tag{1}$$

The precise choice of the monitor function is important to the success of this method and is very problem dependent [24]. We will discuss choices of M appropriate to the computation of the solutions of the Eady Problem in Section 5.

In one spatial dimension the equidistribution principle (1) together with the boundary conditions, uniquely defines the coordinate transformation and this leads to a set of algorithms and software for mesh movement and mesh generation described, for example in [24]. However, in higher dimensions equidistribution does not define a unique mapping. Thus in addition to the equidistribution condition (1), additional constraints on the mesh need to be imposed to specify it exactly. Such constraints allow some additional control over the form and regularity of the mesh and there is a wide literature on the use of different forms of constraint, from Harmonic maps to tensor-valued monitor functions. However, we will adopt a particular approach which is simple to implement and leads to meshes with good regularity. A strong degree of control over the form of the transfomation is given by requiring that the map $\mathbf{F}$ minimises the average deviation I of the computed mesh from the original regular mesh where

$$I = \int_{\Omega_C} |\mathbf{x}(\boldsymbol{\xi}) - \boldsymbol{\xi}|^2 \, d\boldsymbol{\xi},$$

with the Jacobian determinant J of the map specified. It is shown in [3], that this constrained minimisation problem has a *unique* solution. It can be shown further [17] that for such a map, $\mathbf{J}$ is *symmetric* so that $\mathbf{J} = \mathbf{J}^T$, and the map is *irrotational* so that

$$x_\eta = z_\xi.$$

Such transformations are called *optimally transported maps* and the fact that they minimise I immediately leads to good regularity properties [9]. Two immediate results follow from the symmetry of $\mathbf{J}$. Firstly, we have that

$$\mathbf{J} = O^T \Lambda O$$

where O is orthogonal and Λ is diagonal, so that such a map can be regarded as a rotation, followed by a stretch, followed by a rotation. A feature of such maps [34] is that the rotation O allows them to align coordinate lines parallel and orthogonal to developing fronts, which is a useful feature for meteorological computations. We see examples of this presently. Secondly, it follows immediately from Poincaré's lemma that such map is the *gradient of a convex scalar function* $Q(\boldsymbol{\xi}, t)$ (which we shall refer to as the *mesh potential*), such that

$$\mathbf{x} = \nabla_\xi Q \tag{2}$$

where $\nabla_\xi = (\partial\xi, \partial\eta)^T$. It follows that the Jacobian J is equal to the Hessian of Q, defined by:

$$H(Q) \equiv \det(\mathbf{H}(Q)), \;\; \text{where} \;\; \mathbf{H}(Q) = \begin{bmatrix} Q_{\xi\xi} & Q_{\xi\eta} \\ Q_{\xi\eta} & Q_{\eta\eta} \end{bmatrix}. \tag{3}$$

Substituting (2) and (3) into the equidistribution equation (1) we obtain the fully non-linear *Monge-Ampere equation* satisfied by the mesh potential

$$M(\mathbf{x}, t) H(Q) = \theta(t), \quad \mathbf{x} = \nabla_\xi Q. \tag{4}$$

It follows from results of [3] that this equation, together with the boundary condition

$$\mathbf{F} : \partial\Omega_C \to \partial\Omega_P,$$

has a unique solution Q (up to an added constant), and therefore, by taking the gradient of Q, it defines a unique mesh. Computation of such an optimally transported mesh in any dimension, thus reduces to the solution of an appropriate *scalar partial differential equation*. However the partial differential equation (4) is fully nonlinear and difficult to solve directly. Various strategies for solving it have been proposed. In [17] a Newton method coupled with a multi-grid approach is used. In [18] a variational algorithm based upon the optimal transport is considered. However, in this paper we will instead we consider a relaxed form of (4) in which we *freeze the*

physical time t and evolve the mesh potential towards an equidistributed state in a pseudo-time τ via the solution of a $\tau-$dependent perturbation of (4). The actual time is then increased and the process repeated to generate a new mesh etc. (Note. This implementation works well for the partial differentlal equations encountered in describing incompressible flows in which an advection equation is coupled to a pressure correction step. In other PDES, such as those with blow-up [5], it may be advantageous to evolve the mesh and the solution simultaneously. In this case we take the pseudo time to equal a multiple of the real time.) A temporal relaxation of the Monge-Ampere equation in two-dimensions which admits a solution $Q(\boldsymbol{\xi}, \tau)$, such that $\nabla_\xi Q(., \tau)$ evolves towards the solution of (4) as the (pseudo-time) $\tau \to \infty$ is described in [7]
This system has the form

$$\epsilon(I - \gamma \triangle_\xi) Q_\tau = (M(\mathbf{x}) H(Q))^{1/2}, \tag{5}$$

$$\mathbf{x} = \nabla_\xi Q,$$

where it is referred to as the *Parabolic Monge Ampère (PMA) Equation*. For this section we initially consider the true time t to be fixed and M to be independent of the pseudo-time (other than through its explicit dependence on τ through $\mathbf{x}$), though we will extend this case presently. The scaling power $1/2$ in (5) is necessary for global existence of the solution and ensures that the right hand side of this equation scales linearly with Q. The operator $(I - \gamma \triangle_\xi)^{-1}$ is a smoothing operator, similar to that used by Ceniceros & Hou [11], which reduces the stiffness of the system when it is discretised. The parameter γ controls the amount of spatial smoothing and ϵ the relaxation rate, and we discuss the choice of these values presently.

Boundary Conditions.

One of the advantages of the Optimal Transport Methods over other moving mesh methods is the way that they treat boundary conditions in a natural way and automatically assign points along boundaries. (Other moving mesh methods [24] require additional equations for the boundary points). In the case of a square boundary $\Omega_C = \{(\xi, \eta) \in [0,1]\}$mapped to a square boundary $\Omega_P = \{(x, z) \in [0,1]\}$ then the natural boundary conditions are Neumann in the ξ and η directions so that

$$Q_\xi|_{\xi=0} = Q_\eta|_{\eta=0} = 0, \quad Q_\xi|_{\xi=1} = Q_\eta|_{\eta=1} = 1. \tag{6}$$

It is possible in the example discussed in this paper to consider periodic boundary conditions in the ξ and x directions. As $x = Q_\xi$ then a uniform mesh in the x-direction is given by $Q = \xi^2/2$ and we can enforce periodicity by taking

$$(Q_\xi - \xi)|_{\xi=0} = (Q_\xi - \xi)|_{\xi=1}, \quad (Q - \frac{1}{2}\xi^2)|_{\xi=0} = (Q - \frac{1}{2}\xi^2)|_{\xi=1}. \tag{7}$$

Initial Conditions.

In a time stepping algorithm it is natural to take an initial mesh when $t = \tau = 0$ to be uniform such that

$$Q(\boldsymbol{\xi}, 0) = \frac{1}{2}|\boldsymbol{\xi}|^2.$$

However, when implementing this method more generally as part of the procedure to solve a PDE it is assumed that M is time varying. In such a context we can assume that Q is known at some time t. If t is then advanced to $t + \Delta t$ then we may use the function $Q(t)$ as the initialization of the pseudo-timestepping algorithm to find the new mesh at the time $t + \Delta t$. Throughout its evolution, the regularity of the mesh is controlled by certain convexity properties of the potential Q which can be estimated from the regularity properties of the PMA equation [9]. Indeed, it is shown in [7] that throughout the evolution Q remains convex and as a result the Jacobian matrix $\mathbf{J}$ is always *non-singular*, which prevents any mesh tangling if the equation (5) is solved sufficiently accurately. This is an important feature of meshes generated by using Optimal Transport methods.

2.3 Discretisation of the PMA equation in two-dimensions

We assume at this stage that the monitor function $M(\mathbf{x}, t)$ is a prescribed function at the time t at all mesh points $\mathbf{x}$. (In subsequent sections M will be given in terms of the solution of the underlying PDE.) To discretise (5) we assume that the computational domain Ω_C has a square logically rectangular mesh with points $\boldsymbol{\xi}_{i,j} = (i\Delta\xi, j\Delta\eta)$, where

$$\Delta\xi = \Delta\eta = \frac{1}{n-1}.$$

Associated with each such mesh point in Ω_C is a point $\mathbf{X}_{i,j}(\tau) = (X_{i,j}(\tau), Z_{i,j}(\tau)) \in \Omega_P$, $i = 1 \ldots n$, $j = 1 \ldots m$, and a corresponding value of the monitor function given by

$$M_{i,j}(\tau) \equiv M(\mathbf{X}_{i,j}(\tau)).$$

It is often found [24], that to obtain more regular meshes the monitor function has to be smoothed by applying a low pass filter . This spreads large local variations in the point values of the monitor function over several mesh points. A typical low pass filter has

$$\frac{1}{4}M_{i,j} + \frac{1}{8}(M_{i+1,j} + M_{i-1,j} + M_{i,j-1} + M_{i,j+1}) + \frac{1}{16}(M_{i+1,j+1} + M_{i-1,j-1} + M_{i+1,j-1} + M_{i-1,j+1}) \to M_{i,j}. \tag{8}$$

The evolving mesh potential $Q_{i,j}(\tau)$ will be defined at the same points in this mesh and once Q is determined the mesh (and hence M) is obtained via

$$X_{i,j} = \frac{Q_{i+1,j} - Q_{i-1,j}}{2\Delta\xi}, \quad Z_{i,j} = \frac{Q_{i,j+1} - Q_{i,j-1}}{2\Delta\eta}. \tag{9}$$

To discretise the Hessian operator $H(Q)$ at the interior points of the domain Ω_C we use a 9-point stencil, so that for $i = 2, \ldots n-1$, $j = 2, \ldots m-1$ we take

$$(Q_{\xi\xi})_{i,j} \approx \frac{Q_{i+1,j} - 2Q_{i,j} + Q_{i-1,j}}{\Delta\xi^2}, \quad (Q_{\eta\eta})_{i,j} \approx \frac{Q_{i,j+1} - 2Q_{i,j} + Q_{i,j-1}}{\Delta\eta^2}$$

and

$$(Q_{\xi\eta})_{i,j} \approx \frac{Q_{i+1,j+1} - Q_{i+1,j-1} - Q_{i-1,j+1} + Q_{i-1,j-1}}{4\Delta\xi\Delta\eta}. \tag{10}$$

The Hessian

$$H(Q) = Q_{\xi\xi}Q_{\eta\eta} - Q_{\xi\eta}^2$$

is then constructed by using the approximations above.

At the boundaries it follows that $Q_{\xi\eta} = 0$. To approximate $Q_{\xi\xi}$ at $x = 0$ we use the following expansions

$$\begin{aligned} Q_{2,j} &= Q_{1,j} + \Delta\xi Q_{1,j}^{'} + \frac{\Delta\xi^2}{2} Q_{1,j}^{''} + \frac{\Delta\xi^3}{6} Q_{1,j}^{'''} + \mathrm{O}(\Delta\xi^4), & (11) \\ Q_{3,j} &= Q_{1,j} + 2\Delta\xi Q_{1,j}^{'} + \frac{4\Delta\xi^2}{2} Q_{1,j}^{''} + \frac{8\Delta\xi^3}{6} Q_{1,j}^{'''} + \mathrm{O}(\Delta\xi^4). & (12) \end{aligned}$$

N.B. The symbol prime denotes $\partial/\partial\xi$. By the Neumann boundary condition (6) $Q_{1,j}^{'} = 0$, and so combining (11) and (12), and neglecting terms of $\mathrm{O}(\Delta\xi^4)$, we obtain

$$(Q_{\xi\xi})_{1,j} \approx \frac{-7Q_{1,j} + 8Q_{2,j} - Q_{3,j}}{2\Delta\xi^2}. \qquad (13)$$

Similarly, at the boundary $x = 1$, we use the approximation

$$(Q_{\xi\xi})_{n,j} \approx \frac{-7Q_{n,j} + 8Q_{n-1,j} - Q_{n-2,j}}{2\Delta\xi^2}. \qquad (14)$$

Similar approximations are used for the term $Q_{\eta\eta}$ at $\eta = 0, 1$. In the case of periodic boundary conditions (7) we simply exploit periodicity in ξ to evaluate ghost nodes as follows

$$\begin{aligned} Q(0, j) &= Q(n-1, j), \ \text{ for } j = 1, ..., m-1, \\ Q(n, j) &= Q(2, j), \ \text{ for } j = 1, ..., m-1. \end{aligned}$$

With the discretisation of the Hessian and the monitor function as above we can formulate the right hand side F, of the PMA equation (5). The left hand side involving the operator $A \equiv (I - \gamma\Delta_\xi)$ is constructed using the standard five point stencil representation of the Laplacian. As the equation (5) is posed in a rectangular domain with a uniform mesh, the operator A can be rapidly inverted through the use of the discrete cosine transformation (see [8]) to give the system

$$Q_\tau = A^{-1}F. \qquad (15)$$

2.4 Solving the PMA equation

As great accuracy for finding Q is not required to produce adequate meshes for solving a PDE, it is easy and cheap to solve (15) by using an explicit forward Euler method so that

$$Q_{i,j}^{n+1} = Q_{i,j}^{n} + \Delta\tau(A^{-1}F)^n. \qquad (16)$$

The choice of ϵ and γ is important to ensure that we can take a time step $\Delta\tau$ that is not too small for practical computations. Note that if too large a value of $\Delta\tau$ is taken then we may observe mesh tangling. If the mesh is initially far from equidistribution (as in the first stage of solving the PDE) then we take $\epsilon = 1$ and $\gamma = \sqrt{\max(M)}$. For meshes which are close to being equidistributed (as will occur in later stages of any computation) we can take $\epsilon = 0.01$. The length of the pseudo-time interval T_f is determined by the following stopping criteria

$$\|\nabla Q^{n+1} - \nabla Q^n\|_2 \leqslant \texttt{tol}$$

for some specified tolerance `tol` chosen to be between 10^{-6} and 10^{-9}.

2.5 Time varying monitor function

In practical applications of this method for generating a mesh to solve a PDE we cannot always consider M to be independent of time. This occurs when M is a function of a variable (such as the potential temperature θ) which in turn depends upon (x, z) and on time. Suppose that a mesh $(X_{i,j}(t), Z_{i,j}(t))$ is computed at a (real time) t which equidistributes $M(\theta(x, z, t))$ and that using this mesh the variable θ is computed at a new time $t + \Delta t$. Typically we have point values $\Theta_{i,j}$ of $\theta(x, z, t + \Delta t)$ at the 'old' mesh points. A new mesh must now be computed to equidistribute the function $M(x, z, t + \Delta t)$ using these point values of Θ. To do this we must *interpolate* the values of $\Theta_{i,j}$ from these original mesh points onto the evolving points which are created as part of the pseudo time-stepping procedure. A procedure for doing this based on [33] is given in the next section.

2.6 Example of a mesh for a collapsing bubble blow

An example of an application of this mesh generation procedure in the context of a meteorological problem we consider the collapsing bubble flow problem described in [29]. This problem relates to the evolution of cold bubble of air collapsing onto the ground over 12 minutes. The data for the calculation described in [29] was obtained by solving the 2D dry Euler equations in cartesian coordinates over the region $x \in [-L, L]$, $z \in [0, H]$ with $L = 25600\text{m}$ and $H = 6400\text{m}$, using the Met Office 2D vertical slice model (with periodic boundary conditions in the $x-$ direction). The computations are made on a uniform square mesh of size 200m. Using this data for the *potential temperature variation* $\Delta\theta$ on the uniform mesh it is then possible to construct a mesh to more correctly resolve the evolving atmospheric phenomena. In particular, as the bubble collapses, it is useful to have a mesh which is refined in the bubble region and coarser elsewhere. As the solution is only given at certain data points within the domain the potential temperature variation must be interpolated onto the whole domain in order to construct the adapted mesh. This is done at each value of the pseudo-timestepping routine by using a conservative interpolation method proposed in [33] (see [6]). To generate the mesh the monitor function $M = 1 + |\Delta\theta|$ was used as this combined simplicity with high resolution. In the PMA mesh equation we take

$$\Delta t = 1, \quad \epsilon = 1 \quad \text{and} \quad \gamma = \sqrt{\max(M)}.$$

This monitor function naturally concentrates points where the absolute potential temperature variation is larger. When solving (5) at each pseudo time-step the value of the potential temperature variation had to be interpolated onto the new mesh in order to recompute the monitor function. The resulting meshes at two times are given in Figures 2 and 3. In each of the figures the upper panel shows the variation in the potential temperature field. The contour intervals, ranging from red to blue, represent the minimum to maximum variation in absolute potential temperature. The lower panel shows the adapted mesh. The meshes contain 33×256 grid points. The resulting resolution in the region of the bubble in the adapted mesh is approximately 30m in the horizontal and 130m in the vertical. The resulting mesh can be seen to have good regularity both close to the bubble and further away. Indeed, we see no evidence of severe mesh non-uniformity or the presence of long thin regions. A partial explanation for this good degree of mesh regularity is given in [6]. The limitations of the results are likely to be because the adapted mesh was not used to solve the flow equations. Clearly in future work it would be of interest to develop a solver for the compressible equations.

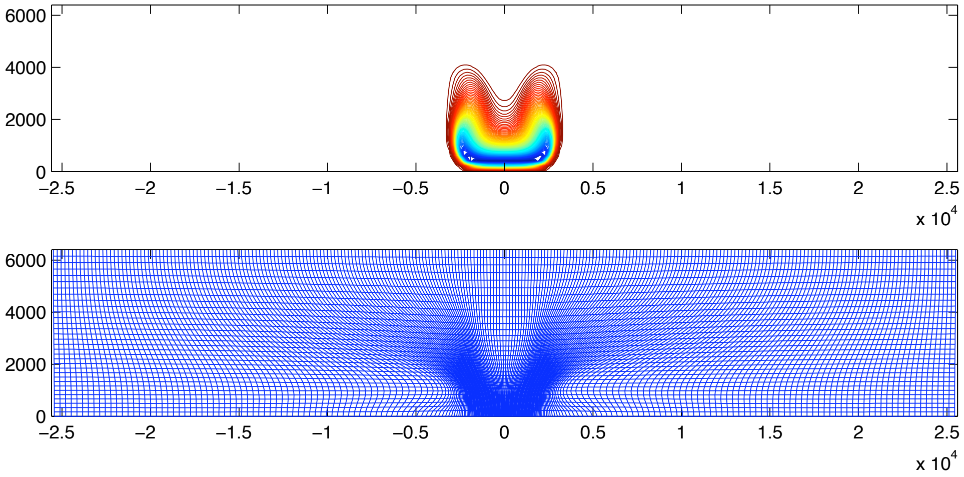

Figure 2: Mesh generated for a collapsing bubble flow at time t = 4 minutes with $M = 1 + |\theta|$.

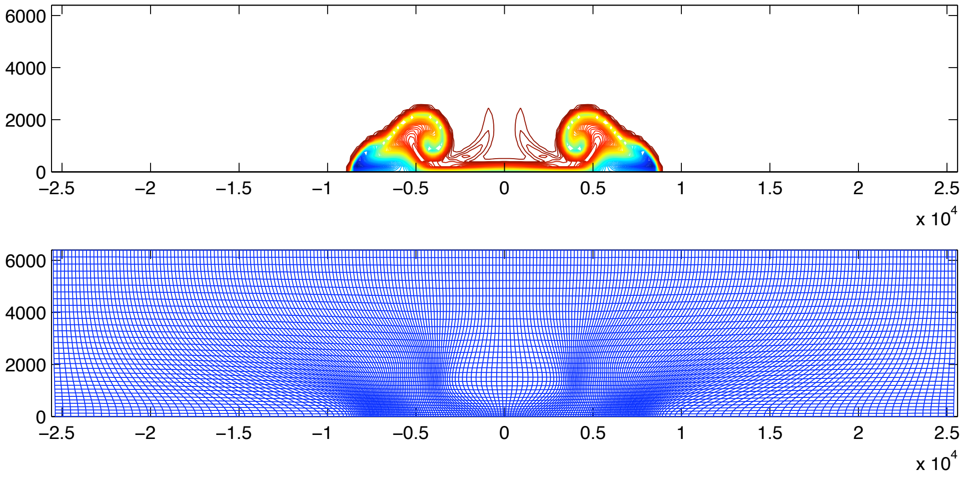

Figure 3: Mesh generated for a collapsing bubble flow at time t = 12 minutes.

3 Coupling moving mesh methods to convection dominated incompressible flows.

The purpose of generating an adaptive mesh in the context of this paper, is to determine accurate solutions of time dependent partial differential equations derived from meteorological applications. Accordingly it is necessary to couple the mesh generation procedure (5) described in the previous section to the solution of the underlying partial differential equation. There are a number of different issues to consider to perform this coupling, see [6], or [24] , [32], [33] for a summary and there are certain aspects of solving meteorological problems which dictate our choice. The first issue is that of whether to discretise the PDE in either the *computational domain* Ω_C (so that we always work in the computational variables (ξ, η), or in the *physical domain* Ω_P (discretising in terms of the physical variables (x, z). In the former case we always work on a uniform, logically rectangular mesh and this has considerable computational advantages. However, we have to rescale the various differential operators to allow for the change of variables from (ξ, η) to (x, z). In general this is difficult, however, an advantage of using the PMA method is that the various rescalings are computed automatically as part of the process of finding the mesh. Accordingly, it is advantageous to use the computational domain for all calculations, and we do this here. The second issue is how often to update the mesh. A class of moving mesh methods called *simultaneous methods* solve the mesh equations such as (5) simultaneously with the underlying PDE (so that the pseudo-time is simply a scaling of the real time), so that the mesh and PDE combine to give an extended system of equations. As the mesh points are moving with (though not necessarily at the same speed as) the solution they are a form of Lagrangian integration. Such methods are implemented, for example, in the package MOVCOL. The great advantage of such methods is that they require no interpolation from the solution onto the mesh (indeed the interpolation is done automatically by incorporating additional, Lagrangian, terms into the underlying PDE to allow for the mesh motion). In the context of the solution of blow-up problems in parabolic PDEs they have proved very effective [5],[7]. A disadvantage of such methods is that the extended system of the mesh and the PDE can be stiff, and time marching schemes may have problems when integrating it especially when the underlying solution has a strongly advective nature [27], [28]. An alternative to such simultaneous schemes are *alternating* or *re-zoning methods* [32] which alternate between finding the mesh for a solution at a particular time level, using the pseudo-timestepping algorithm, interpolating the existing solution onto this new mesh and then freezing the mesh and finding the solution of the PDE on the next time level. Such methods are more robust than simultaneous methods (having much less issues with stiffness), and can also use standard software to perform the time-marching step. However, they require careful interpolation of the solution from one mesh to the next, and this becomes a significant part of the overall computation. Further references to coupling PDES with mesh generation algorithms, such as Winslow's variable diffusion method can be found in the references quoted above. In the context of *meteorological problems* there are a number of special issues that must be considered which help to dictate the choice between these two approaches. Firstly, meteorological systems are described by *near-incompressible flows*. Typically these problems are solved by using a semi-implicit method in which the dynamical variables such as the velocity and temperature are first time marched to new values, and in a second step these values, and that of the pressure, are then corrected to ensure that the velocity is divergence free. This two step process is most easily coupled to a (locally) stationary rather than a moving mesh. Secondly, associated with the meteorological flows describing the dynamics and thermodynamics of the atmosphere are conserved quantities such as mass, total energy, angular momentum and entropy. Conserving such quantities in a simultaneous solution is difficult. In contrast, in an alternating method a careful interpolation procedure can ensure that these quantities are conserved. Accordingly, in this section we will describe a general *computational domain based alternating method* in the context of solving such incompressible flows law using the PMA adaptive moving mesh method coupled to a Pressure-Correction method. The method we describe is, in part, motivated by the Eady Problem which we describe in detail in the next section, however we derive it in more generality here. Accordingly, we consider a PDE describing a 2 dimensional incompressible fluid flow of velocity (u, w) and pressure/geopotential φ in the region $(x, z) \in [-L, L] \times [0, H]$, with x a horizontal and z a vertical coordinate. This flow is then coupled to a set of advection equations for the variables (v, θ) and is described by the (constrained) momentum equations

$$\mathbf{u}_t + \mathbf{u}.\nabla\mathbf{u} = -\nabla\varphi + \mathbf{F}(\mathbf{u}, \boldsymbol{\theta}, \mathbf{x}, t), \tag{17}$$

and

$$\nabla.\mathbf{u} \equiv u_x + w_z = 0, \tag{18}$$

coupled to the (unconstrained) advection equations

$$\boldsymbol{\theta}_t + \mathbf{u}.\nabla\boldsymbol{\theta} = \mathbf{G}(\mathbf{u}, \boldsymbol{\theta}, \mathbf{x}, t). \tag{19}$$

Here $\mathbf{F} = [F1, F2]$, and $\mathbf{G} = [G1, G2]$. In this system the function $\mathbf{u}$ must exactly satisfy the continuity equation (18) in discrete form on the current computational mesh. This places a severe restriction on both the solution and the mesh.

For boundary conditions, we assume that all variables are periodic with respect to x. We also assume a zero flux condition at the upper and lower boundaries, so that $w = 0$ when $z = 0$ and $z = H$.

The equations (17)-(19) are then solved subject to the initial data

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u_0}, \quad \boldsymbol{\theta}(\mathbf{x}, 0) = \boldsymbol{\theta_0}. \tag{20}$$

The natural asymptotic limit of (17) is a *balanced state* [4] for which the initial pressure gradient satisfies the balance relation

$$\nabla\varphi(\mathbf{x}, 0) = \mathbf{F}(\mathbf{u_0}, \boldsymbol{\theta_0}, \mathbf{x}, 0). \tag{21}$$

The adaptive procedure used comprises *four* steps. We assume that at time-level t^n we have a discrete approximation to the solution of (17) given by

$$(\mathbf{U}^n, \boldsymbol{\Theta}^n, \Phi^n) \approx (\mathbf{u}(t^n), \boldsymbol{\theta}(t^n), \varphi(t^n)),$$

and both a mesh (X^n, Z^n) and explicit estimates for the Jacobian derivative of the mesh transformation function.

- **Step 1**: Keeping (X^n, Z^n) and Φ^n *fixed*, advance $\mathbf{U}^n$ to an intermediate velocity field $\tilde{\mathbf{U}}^{n+1}$ using the momentum equations (17), and advance $\boldsymbol{\Theta}^n$ to $\hat{\boldsymbol{\Theta}}^{n+1}$ using the advective equations (19) posed (using the mesh transformation function and its derivatives) in the computational domain. Keeping (X^n, Z^n) fixed, use a Pressure-Correction method, solving a Poisson equation to advance Φ^n to $\hat{\Phi}^n$ and $\tilde{\mathbf{U}}^{n+1}$ to $\hat{\mathbf{U}}^{n+1}$ so that (in a discrete setting) $\nabla.\hat{\mathbf{U}}^{n+1} = 0$.

- **Step 2**: Using pseudo-time stepping applied to the PMA equation of Section 2, advance the mesh (in pseudo-time) to (X^{n+1}, Z^{n+1}) to equidistribute an appropriate monitor function $M(\hat{\mathbf{U}}^{n+1}, \hat{\mathbf{\Theta}}^{n+1})$.
- **Step 3**: Using a careful interpolation scheme, interpolate $\hat{\mathbf{U}}^{n+1}, \hat{\mathbf{\Theta}}^{n+1}$, and $\hat{\Phi}^{n+1}$, onto the new mesh to give $\mathbf{U}^{n+1}, \mathbf{\Theta}^{n+1}$, and Φ^{n+1}, ensuring through a further pressure correction step that $\nabla.\mathbf{U}^{n+1} = 0$, in a discrete sense, on the new mesh.

In this process, Step 1 which evolves the solution on a fixed (non-uniform) mesh follows usual practices in CFD, although the fact that we do all computations for evolving the PDE in the computational domain introduces some additional complexity to the system which we describe below. The pseudo-time stepping procedure to adapt the mesh in Step 2, is as described in the previous section. In Step 3 we update the new solution onto the new mesh. This step has to be done with care to preserve the solution structure, in particular the divergence free constraint.

3.1 The Pressure-Correction Method and its discretisation

We now give more details about Step 1. To perform all of the computations, we employ a finite-difference, incremental and alternating, pressure-correction method on a Charney-Phillips mesh to find an approximate solution of (17)-(19). In the first stage of this method, the momentum equations for (17) are used to advance the velocity field over one time-step Δt_n to give an approximate velocity field which is not generally divergence-free. In the second stage, we solve a Poisson equation for the pressure/geopotential with Neumann boundary conditions in the z-direction, and periodic boundary conditions in the x-direction. The pressure and the velocity fields are corrected to satisfy the continuity equation.

A. Temporal discretisation

- **Step 1A**: Compute an intermediate discrete velocity field $\tilde{\mathbf{U}}^{n+1}$ via

$$\tilde{\mathbf{U}}^{n+1} = \mathbf{U}^n - \Delta t_n \left((\mathbf{U}.\nabla_h \mathbf{U} + \nabla \Phi - \mathbf{F} \right)^n \qquad (22)$$

and advance the advective variables via

$$\mathbf{\Theta}^{n+1} = \mathbf{\Theta}^n - \Delta t_n \left(\mathbf{U}.\nabla_h \mathbf{\Theta} - \mathbf{G} \right)^n. \qquad (23)$$

Here ∇_h is a *discrete spatial gradient operator* which we define presently.

- **Step 1B**: Let

$$\hat{\Phi}^{n+1} \quad = \quad \tilde{\Phi}^n + \Phi^n, \qquad (24)$$

where $\tilde{\Phi}^n$ is a pressure correction term to be determined, such that the updated velocity field is given by

$$\hat{\mathbf{U}}^{n+1} \quad = \quad \tilde{\mathbf{U}}^{n+1} - \Delta t_n \nabla_h \tilde{\Phi}^n. \qquad (25)$$

We require the updated velocity field to be divergence-free in the discrete sense, such that

$$\nabla_h.\hat{\mathbf{U}}^{n+1} = 0$$

where now $\nabla_h.$ is the *discrete divergence operator*. This leads to a discrete Poisson equation for $\tilde{\Phi}^n$ in the form

$$\nabla_h.\nabla_h\tilde{\Phi}^n \quad = \quad \Delta t_n^{-1}\nabla_h.\tilde{\mathbf{U}}^{n+1}. \tag{26}$$

We discretise, and then solve, this Poisson equation in the computational space using a careful discretisation of the scaled Laplacian operator to obtain the pressure correction $\tilde{\Phi}^n$, which we describe presently. Once the pressure correction term is obtained , both the pressure/geopotential and velocity field are updated using (24) and (25).

In implementing this method, the magnitude of the variable time step Δt_n is determined by the CFL condition [25]. In practice, due to the nonlinear transport terms, a predictor corrector method [25] is used to ensure stability and accuracy in implementing Steps 1A and 1B. This requires two iterations of Steps 1A and 1B at each time-step in which the right hand side of the expressions (22), (23) is replaced by the mean of the values of this expression at the original time level n and the most recently updated pressure and velocity terms calculated from step 1B.

B. Spatial Discretisation of the momentum and Poisson equations

Now we discuss the spatial discretisation of the Pressure-Correction method. This is done by using a finite difference method in the computational domain Ω_C, using a uniform mesh in this domain and introducing scaling factors to account for the transformation from the computational to the physical coordinates.

In the computational domain we use a uniform semi-staggered grid. In particular we define the discrete pressure/geopotential Φ at the cell centre and adopt the convention that all vectors $\mathbf{U}$, $\mathbf{\Theta}$, $\mathbf{F}$, and $\mathbf{G}$, in equations (17)-(19) have their first component located at the centre of the vertical cell edge and their second component at the centre of the horizontal edge. If $\mathbf{U} = (U, W)^T$ and $\mathbf{\Theta} = (V, \Theta)^T$ then we have a Charney-Phillips mesh. This is specifically relevant to the Eady problem discussed in the next section, but can apply to any other problem of similar structure. In Figure 4 we show the locations of $\mathbf{U} = (U, W)^T$, and Φ in a typical cell for such a grid. Following the convention adopted in the previous section, the mesh is defined by two-dimensional grid-points $\mathbf{X}_{i,j} = (X_{i,j}, Z_{i,j})$, and these are defined at the cell corners. The centre of the mesh

$$\mathbf{X}_{i+\frac{1}{2},j+\frac{1}{2}} \equiv (X_{i+\frac{1}{2},j+\frac{1}{2}}, Z_{i+\frac{1}{2},j+\frac{1}{2}}),$$

is defined as the average of the coordinates of its four corners. Similar averaging can also be used to estimate the value of variables at locations other than where they are defined. All spatial derivatives in the physical domain must now be expressed in terms of the computational co-ordinates, and doing this carefully is an essential part of the computational procedure. To do this, we note that if f is an arbitrary scalar function (eg. u, v, w, θ and φ) then in the computational space the physical derivatives take the form

$$\nabla f \equiv (f_x, f_z)^T = \begin{bmatrix} J^{-1}[z_\eta f_\xi - z_\xi f_\eta] \\ J^{-1}[-x_\eta f_\xi + x_\xi f_\eta] \end{bmatrix} \tag{27}$$

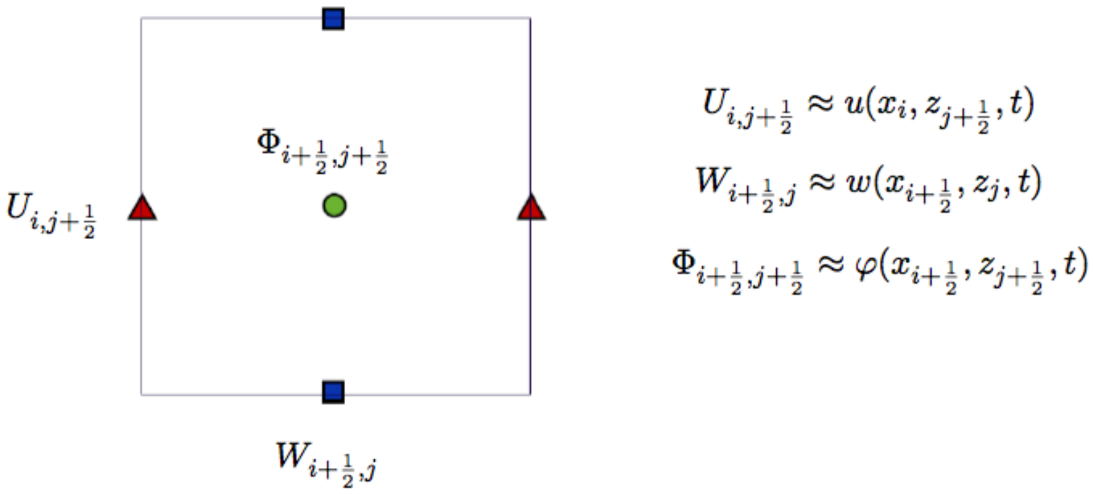
$\Phi_{i+\frac{1}{2},j+\frac{1}{2}}$
$U_{i,j+\frac{1}{2}}$
$W_{i+\frac{1}{2},j}$
$U_{i,j+\frac{1}{2}} \approx u(x_i, z_{j+\frac{1}{2}}, t)$
$W_{i+\frac{1}{2},j} \approx w(x_{i+\frac{1}{2}}, z_j, t)$
$\Phi_{i+\frac{1}{2},j+\frac{1}{2}} \approx \varphi(x_{i+\frac{1}{2}}, z_{j+\frac{1}{2}}, t)$

Figure 4: Locations of the discrete variables U, W, and Φ on the staggered grid.

where

$$J = x_\xi z_\eta - x_\eta z_\xi = H(Q). \tag{28}$$

An equivalent representation of the rescaled gradient operator in *conservation form* is given by

$$\nabla f = \begin{bmatrix} J^{-1}[(z_\eta f)_\xi - (z_\xi f)_\eta] \\ J^{-1}[-(x_\eta f)_\xi + (x_\xi f)_\eta] \end{bmatrix}. \tag{29}$$

For ease of implementation we use the representation given in (27) to discretise $\nabla\varphi$ as this gives rise to a narrower stencil when discretising the left hand side of the Poisson equation (26). For all other variables we use the conservation form given in (29). We will also consistently use the rescaled divergence operator in the form

$$\nabla \cdot \mathbf{f} = \frac{1}{J}\left[(z_\eta f_1)_\xi - (z_\xi f_1)_\eta - (x_\eta f_2)_\xi + (x_\xi f_2)_\eta\right] \tag{30}$$

where $\mathbf{f} = [f_1, f_2]^T$.

We now describe the discretisation of the momentum and advective equations. As a first calculation we must determine discrete values for the derivatives x_ξ, z_η, J etc. as used in (29). In the mesh generated by the PMA equation, the values of $X_{i,j}$ and $Z_{i,j}$ are given at the cell corners in the computational domain. Accordingly we use the approximations

$$[X_\xi]_{i+\frac{1}{2},j} = \frac{X_{i,j+1} - X_{i,j}}{\Delta\xi}, \quad [X_\eta]_{i,j+1/2} = \frac{X_{i,j+1} - X_{i,j}}{\Delta\eta}, \quad [Z_\xi]_{i+\frac{1}{2},j} = \frac{Z_{i,j+1} - Z_{i,j}}{\Delta\xi}, \quad \text{etc.} \tag{31}$$

We also need to determine approximations to the derivatives at other points in the mesh. This is done by averaging so that

$$[X_\xi]_{i,j+\frac{1}{2}} = \frac{[X_\xi]_{i+\frac{1}{2},j} + [X_\xi]_{i+\frac{1}{2},j+1} + [X_\xi]_{i-\frac{1}{2},j} + [X_\xi]_{i-\frac{1}{2},j+1}}{4} \tag{32}$$

$$[X_\eta]_{i+\frac{1}{2},j} = \frac{[X_\eta]_{i,j+\frac{1}{2}} + [X_\eta]_{i+1,j+\frac{1}{2}} + [X_\eta]_{i,j-\frac{1}{2}} + [X_\eta]_{i+1,j-\frac{1}{2}}}{4}, \quad \text{etc.} \tag{33}$$

Finally we set

$$J_{i,j+\frac{1}{2}} = [X_\xi]_{i,j+\frac{1}{2}}[Z_\eta]_{i,j+\frac{1}{2}} - [X_\eta]_{i,j+\frac{1}{2}}[Z_\xi]_{i,j+\frac{1}{2}}$$

.

The first component U of the discretised velocity field is then advanced through the explicit expression

$$[U_t]_{i,j+\frac{1}{2}} = -U_{i,j+\frac{1}{2}}[U_x]_{i,j+\frac{1}{2}} - \frac{1}{4}\Big((W_{i-\frac{1}{2},j} + W_{i+\frac{1}{2},j})[U_z]_{i,j} + (W_{i-\frac{1}{2},j+1} + W_{i+\frac{1}{2},j+1})[U_z]_{i,j+1}\Big)$$

$$+F1_{i,j+\frac{1}{2}} - [\Phi_x]_{i,j+\frac{1}{2}}. \tag{34}$$

The various terms in this expression are evaluated as follows: For $j = 2, \ldots, m-2$, the value of u_x is approximated by using a discrete form of (29) to give

$$[U_x]_{i,j+\frac{1}{2}} = \frac{1}{J_{i,j+\frac{1}{2}}}\left[\frac{[Z_\eta]_{i+1,j+\frac{1}{2}}U_{i+1,j+\frac{1}{2}} - [Z_\eta]_{i-1,j+\frac{1}{2}}U_{i-1,j+\frac{1}{2}}}{2\Delta\xi} - \frac{[Z_\xi]_{i,j+\frac{3}{2}}U_{i,j+\frac{3}{2}} - [Z_\xi]_{i,j-\frac{1}{2}}U_{i,j-\frac{1}{2}}}{2\Delta\eta}\right].$$

At the bottom boundary, for $j = 1$ this is replaced by

$$[U_x]_{i,j+\frac{1}{2}} = \frac{1}{J_{i,j+\frac{1}{2}}}\left[\frac{[Z_\eta]_{i+1,j+\frac{1}{2}}U_{i+1,j+\frac{1}{2}} - [Z_\eta]_{i-1,j+\frac{1}{2}}U_{i-1,j+\frac{1}{2}}}{2\Delta\xi} - \frac{[Z_\xi]_{i,j+\frac{3}{2}}U_{i,j+\frac{3}{2}} - [Z_\xi]_{i,j+\frac{1}{2}}U_{i,j+\frac{1}{2}}}{\Delta\eta}\right],$$

and at the top boundary, for $j = m-1$ by

$$[U_x]_{i,j+\frac{1}{2}} = \frac{1}{J_{i,j+\frac{1}{2}}}\left[\frac{[Z_\eta]_{i+1,j+\frac{1}{2}}U_{i+1,j+\frac{1}{2}} - [Z_\eta]_{i-1,j+\frac{1}{2}}U_{i-1,j+\frac{1}{2}}}{2\Delta\xi} - \frac{[Z_\xi]_{i,j+\frac{1}{2}}U_{i,j+\frac{1}{2}} - [Z_\xi]_{i,j-\frac{1}{2}}U_{i,j-\frac{1}{2}}}{\Delta\eta}\right].$$

Similarly, for $j = 2, \ldots, m-1$, we approximate u_z via

$$[U_z]_{i,j} = \frac{1}{J_{i,j}}\left[-\frac{[X_\eta]_{i+\frac{1}{2},j}U_{i+\frac{1}{2},j} - [X_\eta]_{i-\frac{1}{2},j}U_{i-\frac{1}{2},j}}{\Delta\xi} + \frac{[X_\xi]_{i,j+\frac{1}{2}}U_{i,j+\frac{1}{2}} - [X_\xi]_{i,j-\frac{1}{2}}U_{i,j-\frac{1}{2}}}{\Delta\eta}\right].$$

Notice that we need to evaluate U at grid points where W, and θ, are located (highlighted in blue). We approximate the value of U at these points by taking an average of its value at neighbouring points where it is defined as follows:

$$U_{i+\frac{1}{2},j} \approx \frac{(U_{i,j+\frac{1}{2}} + U_{i,j-\frac{1}{2}} + U_{i+1,j+\frac{1}{2}} + U_{i+1,j-\frac{1}{2}})}{4}.$$

The term $[U_z]_{i,j}$ does not have to be evaluated for $j = 1$, and $j = m$, since the term $[X_\eta]_{i+\frac{1}{2},j} = 0$ at the boundary. The term $[X_\xi U]_\eta$ is discretised using a forward and backward difference at the boundary such that

$$[U_z]_{i,1} = \frac{1}{J_{i,1}}\left[\frac{(X_\xi)_{i,\frac{3}{2}}U_{i,\frac{3}{2}} - (X_\xi)_{i,1}U_{i,1}}{\Delta\eta}\right], \quad [U_z]_{i,m} = \frac{1}{J_{i,m}}\left[\frac{(X_\xi)_{i,m-\frac{1}{2}}U_{i,m-\frac{1}{2}} - (X_\xi)_{i,m}U_{i,m}}{\Delta\eta}\right].$$

The periodic boundary condition on U are imposed via

$$U_{0,k+\frac{1}{2}} = U_{n-1,k+\frac{1}{2}}, \quad U_{n,k+\frac{1}{2}} = U_{1,k+\frac{1}{2}}.$$

We use equation (27) to discretise the value of φ_x as follows

$$[\Phi_x]_{i,j+\frac{1}{2}} \;=\; \frac{1}{[J]_{i,j+\frac{1}{2}}}([Z_\eta]_{i,j+\frac{1}{2}}[\Phi_\xi]_{i,j+\frac{1}{2}} - [Z_\xi]_{i,j+\frac{1}{2}}[\Phi_\eta]_{i,j+\frac{1}{2}}), \tag{35}$$

where

$$[\Phi_\xi]_{i,j+\frac{1}{2}} = \frac{\Phi_{i+\frac{1}{2},j+\frac{1}{2}} - \Phi_{i-\frac{1}{2},j+\frac{1}{2}}}{\Delta\xi}, \quad [\Phi_\eta]_{i,j+\frac{1}{2}} = \frac{1}{2}\left[\frac{\Phi_{i+\frac{1}{2},j+\frac{3}{2}} - \Phi_{i+\frac{1}{2},j-\frac{1}{2}}}{2\Delta\eta} + \frac{\Phi_{i-\frac{1}{2},j+\frac{3}{2}} - \Phi_{i-\frac{1}{2},j-\frac{1}{2}}}{2\Delta\eta}\right]. \tag{36}$$

This invokes the values $\Phi_{\frac{1}{2},j+\frac{1}{2}}$, $\Phi_{i+\frac{1}{2},\frac{1}{2}}$, and $\Phi_{i+\frac{1}{2},m+\frac{1}{2}}$, which are exterior to the domain. However since Φ is periodic with respect to ξ then it follows that

$$\Phi_{\frac{1}{2},j+\frac{1}{2}} = \Phi_{n-\frac{1}{2},j+\frac{1}{2}}. \tag{37}$$

The other components W, V, Θ are advanced using very similar expressions, taking care at all boundaries. See [34] for more details.

3.2 Solving the Pressure-Poisson equation

This is a substantial part of the overall calculation and has to be treated with great care. The pressure correction satisfies the elliptic PDE

$$\nabla_h.\nabla_h\tilde{\Phi}^n = \Delta t^{-1}\nabla_h.\tilde{\mathbf{U}}^{n+1}. \tag{38}$$

For ease of notation we now drop the superscript notation which indicates the time level. Since the pressure update satisfies $\hat{W} = \tilde{W} - \Delta t[\tilde{\Phi}_z]$, and $\hat{W} = \tilde{W} = 0$ at $z = 0, H$, it follows immediately that the solution of (38) satisfies the mixed Neumann and Periodic boundary conditions

$$[\tilde{\Phi}_z] = 0, \quad z = 0, H \quad \text{and} \quad \tilde{\Phi}(x,z) = \tilde{\Phi}(x+L,z).$$

We now discretise (38) on the same semi-staggered grid as before, using the gradient operator for φ given in (27), and in discrete form in (35), and the divergence operator in (30). It is important that the divergence operator is discretised consistently on both sides of the Poisson equation for the divergence free constraint is to be enforced to machine precision. Note that this requires the x and z derivatives of $\tilde{\Phi}^n$ to be located in the same position as the discrete velocities $\tilde{U}^{n+1}$ and $\tilde{W}^{n+1}$. The divergence operator (30) for the general vector $\mathbf{F} = [F1, F2]^T$ is discretised as follows, for $i = 1, ..., n-1$, and $j = 2, ..., m-2$,

$$\begin{aligned}[\nabla\cdot\mathbf{F}]_{i+\frac{1}{2},j+\frac{1}{2}} \;=\; & \frac{1}{[J]_{i+\frac{1}{2},j+\frac{1}{2}}}\left[\frac{[Z_\eta F1]_{i+1,j+\frac{1}{2}} - [Z_\eta F1]_{i,j+\frac{1}{2}}}{\Delta\xi} - \right.\\ & \frac{[Z_\xi F1]_{i+\frac{1}{2},j+1} - [Z_\xi F1]_{i+\frac{1}{2},j}}{\Delta\eta} - \frac{[X_\eta F2]_{i+1,j+\frac{1}{2}} - [X_\eta F2]_{i,j+\frac{1}{2}}}{\Delta\xi}\\ & \left. + \frac{[X_\xi F2]_{i+\frac{1}{2},j+1} - [X_\xi F2]_{i+\frac{1}{2},j}}{\Delta\eta}\right],\end{aligned}$$

noting that due to the periodic boundary conditions $[Z_\eta F1]_{n,j+\frac{1}{2}} = [Z_\eta F1]_{1,j+\frac{1}{2}}$ and $[X_\xi F2]_{n,j+\frac{1}{2}} = [X_\xi F2]_{1,j+\frac{1}{2}}$. Note that for $\mathbf{F} = \tilde{\mathbf{U}}, [\nabla_h\tilde{\Phi}]$, due to the boundary conditions of the mesh and the momentum equations, we do not need to evaluate the terms $[Z_\xi F1]$, and $[X_\eta F2]$ at the boundaries $z = 0$ and $z = H$. The terms $[J]$,$[Z_\eta F1]$, $[Z_\xi F1]$, $[X_\eta F2]$, and $[X_\xi F2]$, are then evaluated as follows: For $i = 1, ..., n-1$, $j = 1, ..., m-1$, by the expressions

$$[Z_\eta F1]_{i,j+\frac{1}{2}} = [Z_\eta]_{i,j+\frac{1}{2}}\, F1_{i,j+\frac{1}{2}}, \quad [X_\xi F2]_{i,j+\frac{1}{2}} = [X_\xi]_{i,j+\frac{1}{2}}\, \frac{\left(F2_{i+\frac{1}{2},j} + F2_{i+\frac{1}{2},j+1} + F2_{i-\frac{1}{2},j} + F2_{i-\frac{1}{2},j+1}\right)}{4},$$

and for $i = 1, ..., n-1$, and $j = 2, ..., m-2$,, by the expressions

$$[X_\eta F2]_{i+\frac{1}{2},j} = [X_\eta]_{i+\frac{1}{2},j}\, F2_{i+\frac{1}{2},j}, \quad [Z_\xi F1]_{i+\frac{1}{2},j} = [Z_\xi]_{i+\frac{1}{2},j}\, \frac{\left(F1_{i,j+\frac{1}{2}} + F1_{i,j-\frac{1}{2}} + F1_{i+1,j+\frac{1}{2}} + F1_{i+1,j-\frac{1}{2}}\right)}{4}.$$

Again due to the periodic boundary conditions we use $F1_{n,j-\frac{1}{2}} = F1_{1,j-\frac{1}{2}}$ and $F2_{\frac{1}{2},j} = F2_{n-\frac{1}{2},j}$.

Finally the term $[J]_{i+\frac{1}{2},j+\frac{1}{2}}$ is evaluated as follows

$$[J]_{i+\frac{1}{2},j+\frac{1}{2}} \quad = \quad [X_\xi]_{i+\frac{1}{2},j+\frac{1}{2}}\, [Z_\eta]_{i+\frac{1}{2},j+\frac{1}{2}} - [X_\eta]_{i+\frac{1}{2},j+\frac{1}{2}}\, [Z_\xi]_{i+\frac{1}{2},j+\frac{1}{2}}\,, \tag{39}$$

where

$$[X_\xi]_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{([X_\xi]_{i+\frac{1}{2},j} + [X_\xi]_{i+\frac{1}{2},j+1})}{2}, \quad [Z_\eta]_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{([Z_\eta]_{i,j+\frac{1}{2}} + [Z_\eta]_{i+1,j\frac{1}{2}})}{2}, \quad \text{etc.}$$

When $\mathbf{F} = [\nabla_h \tilde{\Phi}]$ we evaluate $\left[\tilde{\Phi}_z\right]_{i,j+\frac{1}{2}}$, and $\left[\tilde{\Phi}_x\right]_{i+\frac{1}{2},j}$, as follows

$$\left[\tilde{\Phi}_z\right]_{i,j+\frac{1}{2}} \quad = \quad \frac{\left([\tilde{\Phi}_z]_{i+\frac{1}{2},j} + [\tilde{\Phi}_z]_{i+\frac{1}{2},j+1} + [\tilde{\Phi}_z]_{i-\frac{1}{2},j} + [\tilde{\Phi}_z]_{i-\frac{1}{2},j+1}\right)}{4},$$

and for $i = 1, ..., n-1$, and $j = 2, ..., m-2$,

$$\left[\tilde{\Phi}_x\right]_{i+\frac{1}{2},j} \quad = \quad \frac{\left([\tilde{\Phi}_x]_{i,j+\frac{1}{2}} + [\tilde{\Phi}_x]_{i,j-\frac{1}{2}} + [\tilde{\Phi}_x]_{i+1,j+\frac{1}{2}} + [\tilde{\Phi}_x]_{i+1,j-\frac{1}{2}}\right)}{4}.$$

where the relevant gradient terms for $\tilde{\varphi}$ are given by the expression (35). Given $\tilde{\Phi}_z = 0$ at the boundaries $z = 0$ and $z = H$, and the Neumann boundary condition for the mesh equation, it follows that

$$[\tilde{\Phi}_\eta] = 0, \qquad \text{at} \qquad z = 0, H. \tag{40}$$

Discretising equation (40) we obtain $\tilde{\Phi}_{i+\frac{1}{2},\frac{1}{2}} = \tilde{\Phi}_{i+\frac{1}{2},\frac{3}{2}}, \quad \tilde{\Phi}_{i+\frac{1}{2},m+\frac{1}{2}} = \tilde{\Phi}_{i+\frac{1}{2},m-\frac{1}{2}}$. Similarly, we can discretise the term $\left[\tilde{\Phi}_z\right]_{i+\frac{1}{2},j}$ so that for $i = 1, ..., n-1$, and $j = 2, ..., m-2$,

$$\left[\tilde{\Phi}_z\right]_{i+\frac{1}{2},j} = \frac{1}{[J]_{i+\frac{1}{2},j}}(-[X_\eta]_{i+\frac{1}{2},j}[\tilde{\Phi}_\xi]_{i+\frac{1}{2},j} + [X_\xi]_{i+\frac{1}{2},j}[\tilde{\Phi}_\eta]_{i+\frac{1}{2},j}) \tag{41}$$

where

$$\left[\tilde{\Phi}_\xi\right]_{i+\frac{1}{2},j} \quad = \quad \frac{1}{2}\left[\frac{\Phi_{i+\frac{3}{2},j+\frac{1}{2}} - \tilde{\Phi}_{i-\frac{1}{2},j+\frac{1}{2}}}{2\Delta\xi} + \frac{\tilde{\Phi}_{i+\frac{3}{2},j-\frac{1}{2}} - \tilde{\Phi}_{i-\frac{1}{2},j-\frac{1}{2}}}{2\Delta\xi}\right], \tag{42}$$

and

$$\left[\tilde{\Phi}_\eta\right]_{i+\frac{1}{2},j} = \frac{\tilde{\Phi}_{i+\frac{1}{2},j+\frac{1}{2}} - \tilde{\Phi}_{i+\frac{1}{2},j-\frac{1}{2}}}{\Delta\eta}. \tag{43}$$

This discretisation of the Poisson equation leads to a linear system of the form

$$A\tilde{\mathbf{\Phi}} = \mathbf{d} \tag{44}$$

where A is a block diagonal symmetric sparse matrix. Due to the Neumann and Periodic boundary conditions the matrix A is singular. This system can still be solved for $\tilde{\mathbf{\Phi}}$ up to a constant by removing the null space, since for any solution $\tilde{\mathbf{\Phi}}$, $\tilde{\mathbf{\Phi}} + c[1, ..., 1]^T$ is also a solution. This can be done by fixing the pressure at one point, whereby an arbitrary row k, in the matrix A, is set to to the identity and the kth element in the vector on the right hand side to zero. Typically the error in the solution will be larger at the point that we choose to fix. To average this error out across all pressure points we instead solve the extended system

$$\begin{bmatrix} A & e \\ e^T & 1 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{\Phi}} \\ \varphi \end{bmatrix} = \begin{bmatrix} \mathbf{d} \\ 1 \end{bmatrix}. \tag{45}$$

Further details about this can be found in [19]. For the relatively small systems (3600 × 3600) that were used in this calculation (in anticipation of the small number of mesh points used for the moving mesh) it was straightforward to solve (45) using a sparsely implemented Gaussian elimination based solver with pivoting.

3.3 Conservative and Divergence Free Interpolation

Following an implementation of Steps 1, and 2, we have the variables $\hat{U}^{n+1}_{i,j+\frac{1}{2}}, \hat{V}^{n+1}_{i,j+\frac{1}{2}}, \hat{W}^{n+1}_{i+\frac{1}{2},j}, \hat{\Theta}^{n+1}_{i+\frac{1}{2},j}$ and $\hat{\Phi}^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}}$, with the velocity field $\hat{\mathbf{U}}^{\mathbf{n+1}}$ divergence free, on the mesh $(X^n_{i,j}, Z^n_{i,j})$. The mesh has evolved (through the PMA equations) to the mesh $(X^{n+1}_{i,j}, Z^{n+1}_{i,j})$. As a final step we must now interpolate the physical variables onto this new mesh in such a manner that the interpolant, $\mathbf{U}^{n+1}$, is divergence free. In order to ensure this we extend the second-order conservative interpolation method of [33], based on flux limiters, with a second Pressure-Correction step. For the pressure term, in the context of dealing with a balanced system, we use Tang's method to interpolate the residuals

$$\hat{R1}^{n+1}_{i,j+\frac{1}{2}} = [\hat{\Phi}_x]^{n+1}_{i,j+\frac{1}{2}} - \hat{F1}^{n+1}_{i,j+\frac{1}{2}}, \quad \text{and} \quad \hat{R2}^{n+1}_{i+\frac{1}{2},j} = [\hat{\Phi}_z]^{n+1}_{i+\frac{1}{2},j} - \hat{F2}^{n+1}_{i+\frac{1}{2},j},$$

which are the deviations from the initial balance relations given in (21), and then use a Pressure-Correction step using the interpolated residuals. We describe this in the context of the PMA method, and then extend it further in Section 5 to also deal with the balance laws for the Eady equation problem.

Interpolation stage 1

Let a logically rectangular *mesh cell* $A^n_{i+1/2,j+1/2}$, in the physical domain Ω_P, be defined by the four points

$$A^n_{i+1/2,j+1/2} = \{\mathbf{X}^n_{i,j}, \mathbf{X}^n_{i+1,j}, \mathbf{X}^n_{i+1,j+1}, \mathbf{X}^n_{i,j+1}\},$$

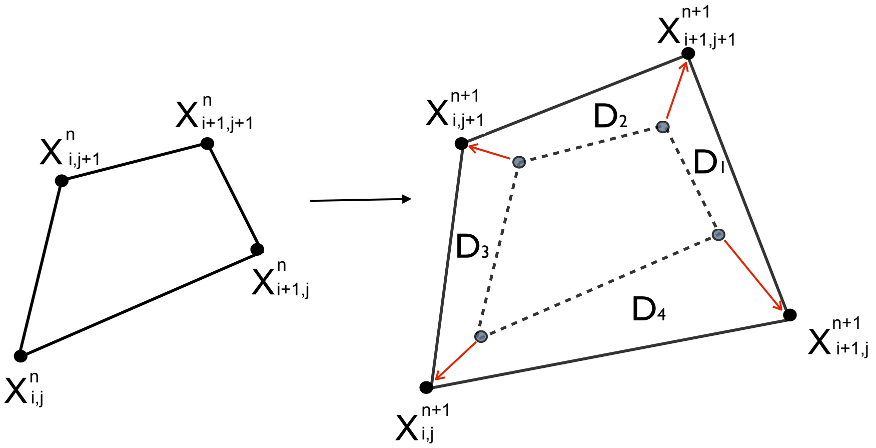

Figure 5: The mesh cell before and after moving the mesh

where $\mathbf{X}_{i,j} \equiv (X_{i,j}, Z_{i,j})$, and let $\hat{S}^{n+1}_{i+1/2,j+1/2}$ be the average of $\hat{S}^{n+1}$ over this new cell so that for $S = U, V, R1, F1$

$$\hat{S}^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{2}(\hat{S}^{n+1}_{i+1,j+\frac{1}{2}} + \hat{S}^{n+1}_{i,j+\frac{1}{2}}),$$

and for $S = W, \Theta, R2, F2$

$$\hat{S}^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{2}(\hat{S}^{n+1}_{i+\frac{1}{2},j+1} + \hat{S}^{n+1}_{i+\frac{1}{2},j}).$$

We define the signed area of a set A by $|A|$. Following the update of the mesh, this mesh cell evolves to a new cell $A^{n+1}_{i+1/2,j+1/2}$ creating four new regions D_1, D_2, D_3, D_4 as illustrated in Figure 3.3 On this new cell we now compute a new cell average $S^*_{i+1/2,j+1/2}$ by including certain integrals, H_k, of $\hat{S}^{n+1}$ over the domains D_k through the formula

$$|A^{n+1}_{i+1/2,j+1/2}|S^*_{i+1/2,j+1/2} = |A^n_{i+1/2,j+1/2}|\hat{S}^{n+1}_{i+1/2,j+1/2} + \sum_{k=1}^{4} H_k. \tag{46}$$

A degree of care has to be taken in evaluating H_k to avoid large numerical dissipation, and to do this we use a second order approximation for each H_k as described in [33]. The signed area of each of the regions D_k can be calculated explicitly, with,

$$|D_1| = \frac{1}{2}((X^{n+1}_{i+1,j+1} - X^n_{i+1,j})(Z^n_{i+1,j+1} - Z^{n+1}_{i+1,j}) - (Z^{n+1}_{i+1,j+1} - Z^n_{i+1,j})(X^n_{i+1,j+1} - X^{n+1}_{i+1,j})), \tag{47}$$

and similar expressions for the other $D_i, i = 2, 3, 4$. We then use the approximation

$$H_1 = |D_1|S_L, \quad \text{if} \quad |D_1| > 0, \quad H_1 = |D_1|S_R \quad \text{if} \quad |D_1| \leqslant 0, \tag{48}$$

with

$$S_L = \hat{S}^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} + \frac{1}{2}U^{\xi}_{i+\frac{1}{2},j+\frac{1}{2}}, \quad S_R = \hat{S}^{n+1}_{i+\frac{3}{2},j+\frac{1}{2}} - \frac{1}{2}S^{\xi}_{j+\frac{1}{2},k+\frac{1}{2}}. \tag{49}$$

Here S^{ξ} is an approximation of the flux given by

$$S^{\xi} = L(\hat{S}^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} - \hat{S}^{n+1}_{i-\frac{1}{2},j+\frac{1}{2}}, \hat{S}^{n+1}_{i+\frac{3}{2},j+\frac{1}{2}} - \hat{S}^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}})$$

where $L(a, b)$ is the van Leer flux-limiter function [33] defined by

$$L(a, b) = (sgn(a) + sgn(b))\frac{|ab|}{|a| + |b| + \epsilon} \quad \text{for some small parameter} \quad 0 < \epsilon \ll 1.$$

Using these values of H_i we compute the new cell averages $S^*_{i+\frac{1}{2},j+\frac{1}{2}}$. For $S = U, V$, we then compute these values at the midpoint of the vertical cell edges using

$$S^*_{i,j+\frac{1}{2}} = \frac{1}{2}(S^*_{i,j+\frac{1}{2}} + S^*_{i+1,j+\frac{1}{2}}).$$

For $S = W, \Theta$, we obtain values at the midpoint of the horizontal cell edges as follows

$$S^*_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{2}(S^*_{i+\frac{1}{2},j} + S^*_{i+\frac{1}{2},j+1}).$$

Interpolation stage 2

The final step of the interpolation procedure is to ensure that the interpolated velocity satisfies the divergence free condition, so that on the new mesh

$$[\nabla_h.\mathbf{U}^{n+1}]_{i+1/2,j+1/2} = 0$$

to machine precision. We achieve this by a second Pressure-Correction step on the new mesh. If we assume that

$$\mathbf{U}^{n+1} = \hat{\mathbf{U}}^{n+1} - \nabla_h \varphi^{n+1},$$

where φ^{n+1} is a correction term that we can obtain by solving the Poisson equation

$$[\nabla_h.\nabla_h \varphi^{n+1}]_{i+1/2,j+1/2} = [\nabla_h.\hat{\mathbf{U}}^{n+1}]_{i+1/2,j+1/2}.$$

Having solved for φ^{n+1} we then correct $\hat{\mathbf{U}}^{n+1}$ on the new mesh to ensure it is divergence free.

Having interpolated the residuals $\hat{R1}^{n+1}$ and $\hat{R2}^{n+1}$ the pressure gradient on the new mesh will satisfy the following

$$R1^{n+1}_{i,j+\frac{1}{2}} = [\Phi^{n+1}_x]_{i,j+\frac{1}{2}} - F1^{n+1}_{i,j+\frac{1}{2}}, \quad \text{and} \quad R2^{n+1}_{i+\frac{1}{2},j} = [\Phi^{n+1}_z]_{i+\frac{1}{2},j} - F2^{n+1}_{i+\frac{1}{2},j}.$$

Therefore by solving

$$[\nabla_h.\nabla_h \Phi^{n+1}]_{i+1/2,j+1/2} = [\nabla_h.(\mathbf{F} + \mathbf{R})^{n+1}]_{i+1/2,j+1/2},$$

we obtain a balanced pressure field Φ^{n+1} on the new mesh. Note that we also use this pressure correction method on the initial adaptive grid to enforce the divergence condition and to satisfy the initial pressure balance relation.

4 The Eady problem and its discretisation

We now put the methods of the previous section into context by applying them to solve the Eady problem. The Eady problem is described by a simplified system of partial differential equations describing the evolution of a vertical cross-section of an incompressible Boussinesq fluid, see [14] Section 4.3. They present a challenge to both fixed mesh and moving mesh methods and thus it is appropriate to extend, in more detail, the methods described in the previous section to this system of equations. They have an asymptotic limit given by the Semi-Geostrophic solution which is known to be singular [13],[15]. Indeed, it is known that the Semi-Geostrophic limit can develop solutions with contact discontinuities in a finite time, and that these are possible models of frontogenesis. It is believed [15] that the Eady equations behave in a similar fashion. In this section we will firstly describe the Eady equations and their associated boundary conditions. We then look at discretisations of them on a Charney-Phillips mesh in the computational plane, following the methods described in the previous section (taking special care to avoid instability through Coriolis terms). To map into a moving mesh the physical plane we consider three cases (i) a constant monitor function, in which case we are simply solving the equations on a fixed, uniform mesh in the physical plane (ii) an arc-length based monitor function and (iii) a potential-vorticity based monitor function. We show that the best results are given for the vorticity based monitor function.

4.1 The 2-D Eady problem

To describe the initial-value Eady problem, we take the physical coordinates (x, y, z), to represent longitude, latitude, and the vertical height at which a given pressure is reached in the reference atmosphere. All variables are assumed to be independent of y, apart from the presence of a prescribed gradient of the hydrostatically balanced basic state pressure and temperature in the y direction. The Boussinesq approximation is made allowing such a solution to be consistent with the full 3D equations (unlike [15]). We solve for the five unknown variables $(u, v, w, \varphi, \theta)$ which represent respectively, the velocity in the x, y and z directions (respectively the zonal, meridional, and vertical velocity), the geo-potential/pressure, and the potential temperature. We define the vector velocity $\mathbf{u} = (u, w)$ and set $\nabla\cdot$ to be the divergence operator in (x, z) coordinates. The Eady problem is then formulated [14] as follows:

$$\begin{aligned}
\frac{\mathrm{D}u}{\mathrm{D}t} - fv + \frac{\partial\varphi}{\partial x} &= 0, \\
\frac{\mathrm{D}w}{\mathrm{D}t} + \frac{\partial\varphi}{\partial z} - \frac{g\theta}{\theta_0} &= 0, \\
\nabla.\mathbf{u} &= 0, \\
\frac{\mathrm{D}v}{\mathrm{D}t} + fu - \frac{Cg}{\theta_0}(z - H/2) &= 0, \\
\frac{\mathrm{D}\theta}{\mathrm{D}t} - Cv &= 0.
\end{aligned} \tag{50}$$

Here θ is the change in potential temperature from the reference state θ_0, the (assumed) constant C is the pole-equator temperature gradient, f is the (constant) Coriolis parameter and g is the acceleration due to gravity. We see that this system takes exactly the form of the more general problem discussed in Section 3, with $\theta = (v, \tilde{\theta})$ corresponding to the advected variable and with

$$\mathbf{F} = (fv, g\tilde{\theta}/\theta_0)^T, \quad \mathbf{G} = (-fu + Cg(z - H/2)/\theta_0, Cv)^T. \tag{51}$$

The Eady equations are solved in a periodic channel $-L < x < L$ and $0 < z < H$, with $H \ll L$. For boundary conditions we apply the rigid lid boundary conditions so that we set

$$w = 0 \quad \text{at} \quad z = 0 \quad \text{and} \quad z = H.$$

At $x = -L$, and $x = L$, we apply periodic boundary conditions to all variables to admit wavelike solutions. The initial data used are taken from [30], where it is assumed that the initial conditions are a reference state plus a perturbation in geostrophic and hydrostatic balance (54). This gives

$$\mathbf{u}(\mathbf{x},0) = \left(\begin{array}{c} \frac{Cg}{f\theta_0}(z - H/2) \\ 0 \end{array} \right), \tag{52}$$

$$\theta(\mathbf{x},0) = N_0^2 \frac{\theta_0 z}{g} + B \sin\left(\pi\left(\frac{x}{L} + \frac{z}{H}\right)\right), \quad v(\mathbf{x},0) = \frac{gBH}{f\theta_s L}\left(\sin\left[\pi\left(\frac{x}{L} + \frac{z}{H}\right)\right] - \cos\left[\pi\left(\frac{x}{L}\right)\right]\right), \tag{53}$$

and

$$\nabla\Phi(\mathbf{x},0) = \left(\begin{array}{c} fv(\mathbf{x},0) \\ \frac{g\theta(\mathbf{x},0)}{\theta_0} \end{array} \right). \tag{54}$$

This data corresponds to an unstable mode of the equations (50) linearised about the state with $B = 0$. If the isentropes have negative slope dx/dz, then v will increase with z, and the evolution equation for θ will increase the vertical gradient of θ, giving a positive feedback. This process represents a conversion of potential energy from the infinite reservoir implied by the imposed basic state, into kinetic energy. It has been demonstrated in [30] and in [16] that the asymptotic limit solution as $\epsilon \equiv f^{-1}D/Dt \to 0$, is semi-geostrophic. The semi-geostrophic limit equations can be obtained from (50) by omitting the terms $\frac{\mathrm{D}u}{\mathrm{D}t}$, and $\frac{\mathrm{D}w}{\mathrm{D}t}$. This limit solution gives a discontinuity in finite time, and the solution for small Rosby number evolves very strong gradients. Hence the need for a fine resolution of the developing front. For our calculations we took $L = 1000\mathrm{km}$, $H = 10\mathrm{km}$, $N_0^2 = 2.5 \times 10^{-5}\mathrm{s}^{-2}$, $f = 10^{-4}\mathrm{s}^{-1}$, $g = 10\mathrm{ms}^{-1}$, $\theta_0 = 300\mathrm{K}$, $C = 3 \times 10^{-6}\mathrm{m}^{-1}\mathrm{K}$, and B is a positive constant.

4.2 Discretisation of the Eady problem

To discretise the Eady problem in the context of the moving mesh we implement the semi-explicit semi-staggered incremental pressure-correction method described in Section 3 with the velocity $\mathbf{u} = (u, w)$, advected components $\theta = (v, \tilde{\theta})$ and pressure φ taking values on the sides and centre of the Charney-Phillips mesh . See similar approaches in [6] and [11] for discretisations of the Navier-Stokes equations. For all of the computations a regular $n \times m$ rectangular mesh was used in the computational domain with $n = m = 60$ and also with $n = 120$, $m = 60$. As initial conditions we took a velocity and pressure field satisfying (52)-(54). In the initial stages of the adaptive computation an initially uniform mesh is evolved to an adapted mesh (X^0, Z^0) by solving the PMA equation in pseudo-time until steady state. (An important assumption is that the *initial state* of the atmosphere is in both hydrostatic and geostrophic balance as expressed in equation (54).) Given the following analytical forms for the initial potential temperature in (52)-(53) we initialised Φ so that the discrete form of (54) is satisfied exactly on the initial adapted mesh, by using the Pressure-Correction method described in Section 3. This Pressure-Correction method is also used to ensure that the initial velocity field satisfies the divergence free condition discretely on the initial adapted mesh.

5 Fixed and Moving mesh computations of the Eady problem

Following the analytic theory, it is expected that the Eady problem admits a solution which becomes increasingly singular in a finite time. Computing this is a significant challenge for any numerical scheme. In this section we describe three computations of the solutions of the Eady problem on (i) a fixed mesh corresponding to a constant monitor function (ii) a moving mesh generated by an arc-length monitor function (iii) a moving mesh generated by a potential vorticity based monitor function. In these calculations we will see both a loss of resolution and an instability when a uniform mesh is used to compute the front. However, using a moving mesh method to concentrate points in the frontal region both reduces the numerical error and avoids spurious oscillations close to the front.

5.1 Computations using a fixed, uniform mesh

With the monitor function M set to a constant, the grid in the physical domain is a simple rescaling of the grid in the computational domain. In all calculations a time step of $\Delta t = 10$ minutes was used to advance the momentum equations and we compared the cases of $n = m = 60$ and $n = 120$, $m = 60$. In the former case the scaling gives physical spacings of $\Delta x \approx 30$km horizontally and $\Delta z \approx 160$m vertically. The calculation using the uniform mesh was allowed to proceed for just over seven days. In this period a front appeared to develop and intensify, visible as an extended approximately vertical linear feature extending from the base to the lid of the atmosphere, with a small cross section horizontally. As might be expected, the resolution of the front was much better when $n = 120$ than when $n = 60$. Full figures showing the results of these numerical computations will be presented in the subsection on results. In these computations, the calculation of the solution to the Eady problem proceeded stably with no sign of intrinsic oscillations (due to the careful discretisation on the Charney-Phillips mesh) for days 1 to 6, until the front started to form. For the case of $n = 60$ the front was poorly resolved, although the computation was stable. In contrast, however, after day 7 the front became well developed when $n = 120$, but the numerical model developed severe grid point noise (significant oscillations) close to the top and bottom boundaries. This occurred because the resolution of the grid (with the fixed mesh) was inadequate for resolving the diminishing length scale of the frontal region, which is most acute on the lower and upper boundaries. To show both the low resolution when $n = 60$ and the oscillations when $n = 120$ we took a cross-section of u in the $x-$direction of the front (setting $z = 0$) at time $t = 6$ days. These results are presented in Figure 6. Similar results can be found in the earlier calculations of Williams [35] who also reported spurious oscillations due to inadequate resolution of the front.

5.2 Computations using a moving mesh

5.2.1 Overview

Before we can implement the moving method described in the previous sections we must determine a suitable monitor function to determine the evolution of the mesh. Provided that such a function is computed in a separate part of the program, then changing it proves simple in general. The choice of a good monitor function is in general both important and difficult for many different problems, and the Eady Equations prove no exception to this. Accordingly we will assess a number of different monitor functions by looking at the form of meshes that they generate, considered both for their capacity to resolve the developing singularity and also for the regularity of the resulting mesh. Having done this we are then able to compute the evolving singular solutions of the

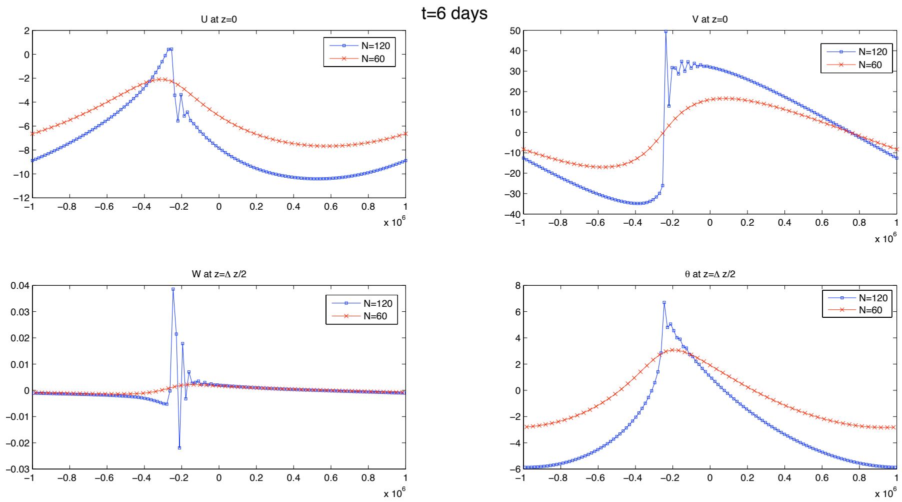

Figure 6: A cross-section in the $x-$direction of the Eady solution on a uniform mesh after $t = 6$ days with $n = 60$, and $n = 120$ mesh points in the horizontal. This shows low resolution when $n = 60$ and grid point oscillations when $n = 120$.

Eady problem and compare them with the computations obtained using a fixed uniform mesh, as reported in the previous subsection. As the front forms it is necessary to take increasingly smaller time-steps, and accordingly we take

$$\Delta t_n = \kappa \Delta x_{\min}$$

where $\Delta x_{\min}$ is the smallest horizontal mesh spacing on the adapted mesh, and κ takes the value of $\Delta t / \Delta x$ as evaluated on the earlier uniform mesh. For all of the moving mesh computations we took $n = m = 60$. Following the initialisation the mesh is updated every 6 real time steps (that is once every hour) and then more rapidly (once every 5 minutes) as the singularity starts to form. The mesh was generated by solving the PMA equation *to steady state* with a pseudo-timestep of $\Delta \tau = 1$ and with $\epsilon = 1$ and $\gamma = \|M\|_\infty$. Typically this took about 10-15 iterations to reach the required tolerance for the mesh equidistribution. No problems with stability were encountered at any stage of the computation.

5.2.2 Arc-length Monitor Function

Arc-length is widely used as a monitor function [32]. It is easy to compute, and has a good track record for placing mesh points close to an evolving interface, and avoiding mesh tangling problems. It is widely used in readily available software [24]. Accordingly we consider this as a first choice of a monitor function for the Eady problem. As all of the physical variables in the Eady problem are tightly coupled, we can justify the monitor function to be a function of just one variable, with the spatial variation appropriately scaled. Accordingly, we consider the arc-length of the *meridional velocity* v so that

$$M = \sqrt{1 + |S \nabla v|^2} \tag{55}$$

where $v = v(x, z, t)$, $\nabla v = (v_x, v_z)^T$ and S is a normalisation given by

$$S = \begin{bmatrix} 2L & 0 \\ 0 & H \end{bmatrix}^{-1}.$$

This monitor function is discretised on the computational domain using central differences so that

$$M_{i+1/2,j+1/2} = \sqrt{1 + \frac{1}{4L^2}\left([V_x]_{i+1/2,j+1/2}\right)^2 + \frac{1}{H^2}\left([V_z]_{i+1/2,j+1/2}\right)^2}. \tag{56}$$

Here (as in the discretisation of the momentum equations), the mean spatial derivative $[V_x]$ is evaluated in the computational space via the rescaled expression

$$\begin{aligned}[V_x]_{i+1/2,j+1/2} = \frac{1}{J_{i+1/2,j+1/2}} \Bigg[& \frac{(Z_\eta)_{i+3/2,j+1/2}V_{i+3/2,j+1/2} - (Z_\eta)_{i-1/2,j+1/2}V_{i-1/2,j+1/2}}{2\Delta\xi} \\ & \frac{(Z_\xi)_{i+1/2,j+3/2}V_{i+1/2,j+3/2} - (Z_\xi)_{i+1/2,j-1/2}V_{i+1/2,j-1/2}}{2\Delta\eta}\Bigg]\end{aligned} \tag{57}$$

with a similar expression for $[V_z]$. This monitor function was then implemented in the moving mesh method described in the previous section using the parameter values $\Delta\tau = 1, \gamma = \|M\|_\infty$ and $\epsilon = 1$ in the moving mesh equation, over the same seven day period as in the uniform mesh computation. In this period a severe front was seen to form which was well resolved by the evolving mesh, with the monitor function correctly concentrating the mesh points. We will not present the figures for this as they are very similar to those for the potential vorticity based monitor function which we consider next. The resulting mesh is presented Figure 7 and shows both a good degree of local uniformity and away from the front location the mesh is very close to being uniform. (This is a direct consequence of the mesh regularity properties of meshes derived from the Monge-Ampere equation). There is no evidence of extreme skewness or a significant departure from orthogonality. The mesh compression achieved with this monitor function is approximately 10:1 when compared to the uniform grid. The mesh generated is very regular away from the front and exhibits smooth grading into the frontal region. No mesh tangling is visible in any stage of the calculation. An alternative to using the arc-length of meridonal velocity for the monitor function, is to instead consider the arc-length of the *potential temperature* so that

$$M = \sqrt{1 + |S\nabla\theta|^2}. \tag{58}$$

This gives very similar results to those obtained above with a slightly lower resolution of the front region.

5.2.3 Potential vorticity based monitor function

Potential Vorticity (PV) is widely recognised as having a central importance in understanding the dynamical behaviour of the atmosphere and of the oceans, including the balanced flows which can lead to cyclonic storms [20]. The formal definition of PV for the semi-geostrophic limit of the Eady equations is given in [14] as

$$PV = \begin{vmatrix} f^2 + fv_x & fv_z \\ g\theta_0^{-1}\theta_x & g\theta_0^{-1}\theta_z \end{vmatrix} = |R| \tag{59}$$

PV is conserved by smooth solutions of the flow and hence remains almost uniform for the initial stages of the evolution. On its own PV therefore makes a poor choice of monitor function. However, as the front is approached all of the terms of the matrix R become large, in particular the largest eigenvalue which provides a measure in the local distortion in the fluid particles. A monitor function which measures this is given by

$$M = 1 + a\rho_{\max} \tag{60}$$

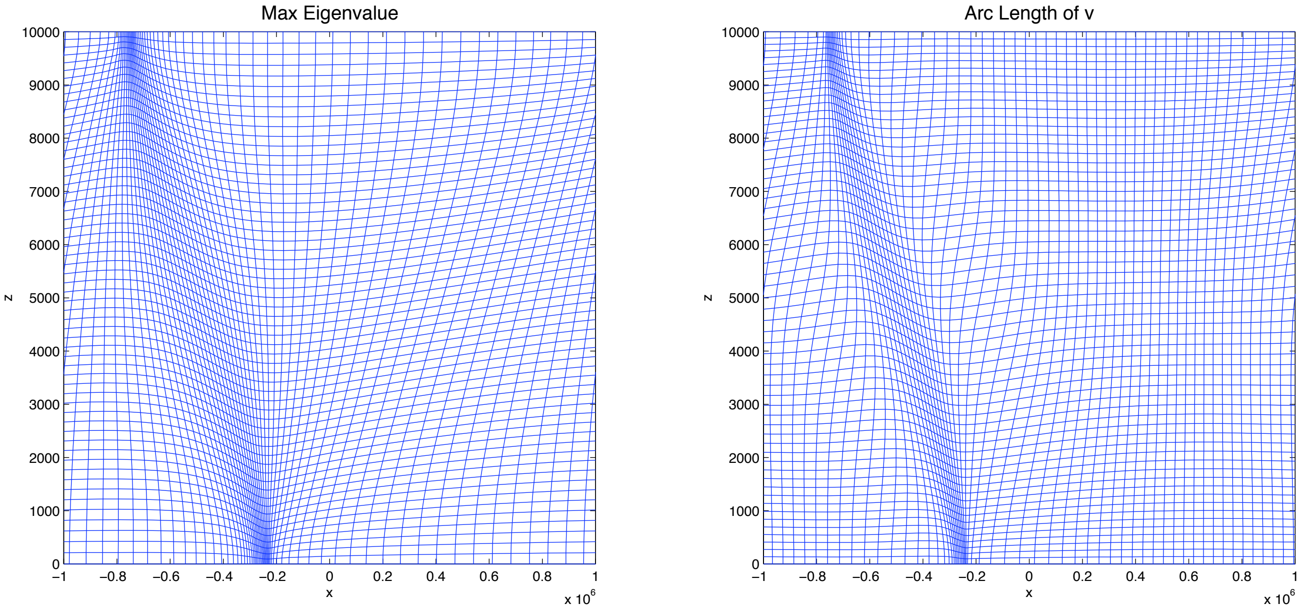

Figure 7: The meshes for the developing front of the Eady problem, obtained using PV based (left) and arc-length based (right) monitor functions.

where $\rho_{\max}$ is the maximum eigenvalue of the matrix R and $a = 1/f$ where f is the Coriolis constant (this allows M to scale correctly). This monitor function leads to a similar 10:1 compression as with the arc-length monitor. Two meshes, computed for the solution of the Eady problem using the maximum eigenvalue of R (which we now refer to as the PV based monitor function) and the arc-length monitor function are presented in Figure 7 with a blow-up in the front region presented in Figure 8.

As can be seen, the two meshes are very similar. However, the PV based monitor appears to represent the solution structure much more appropriately, generating a slightly more regular mesh with a greater compression of mesh points in the frontal region. This leads us to recommend its use in similar future calculations and we describe the performance of the moving mesh method using this in more detail in the next section.

5.3 Results

5.3.1 Computed solution

The results of computations of the solutions of the Eady problem with $n = m = 60$ and using both a uniform mesh and a moving mesh with a PV based monitor function up to the formation of the front are presented in Figures 9, 10, and 11. In these figures we show the contours of u, v, w and θ at times 5 days, 6 days and 6.3 days.

Two features of these figures are clear. **Firstly** that the moving mesh produces a much finer resolution of the evolving front and that much more detail can be resolved. **Secondly**, none of the moving mesh calculations show the oscillatory instability observed when using the uniform mesh. To investigate the resolution of the front in more detail we present in Figure 12 a cross section at the lower boundary of the computed vertical velocity

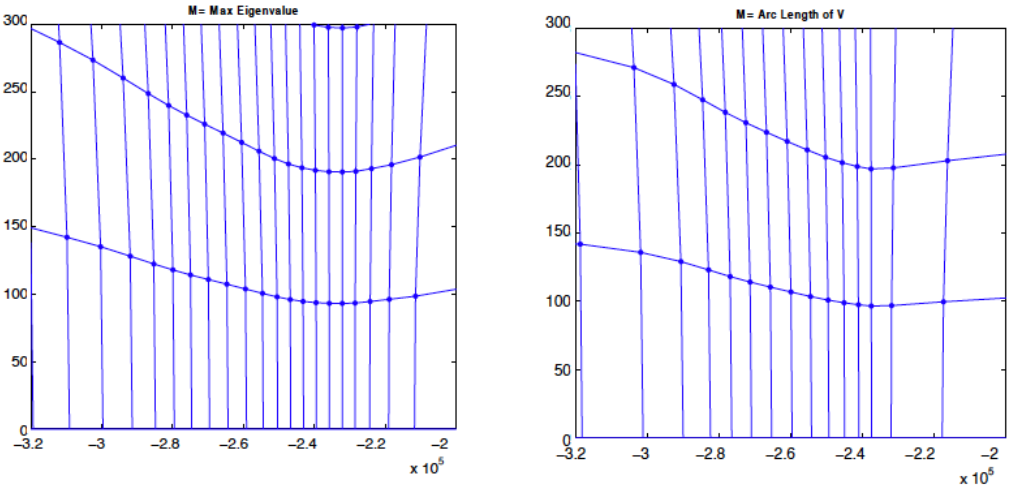

Figure 8: A blow up of the meshes generated using the PV based (left) and-arc length based (right) monitor functions.

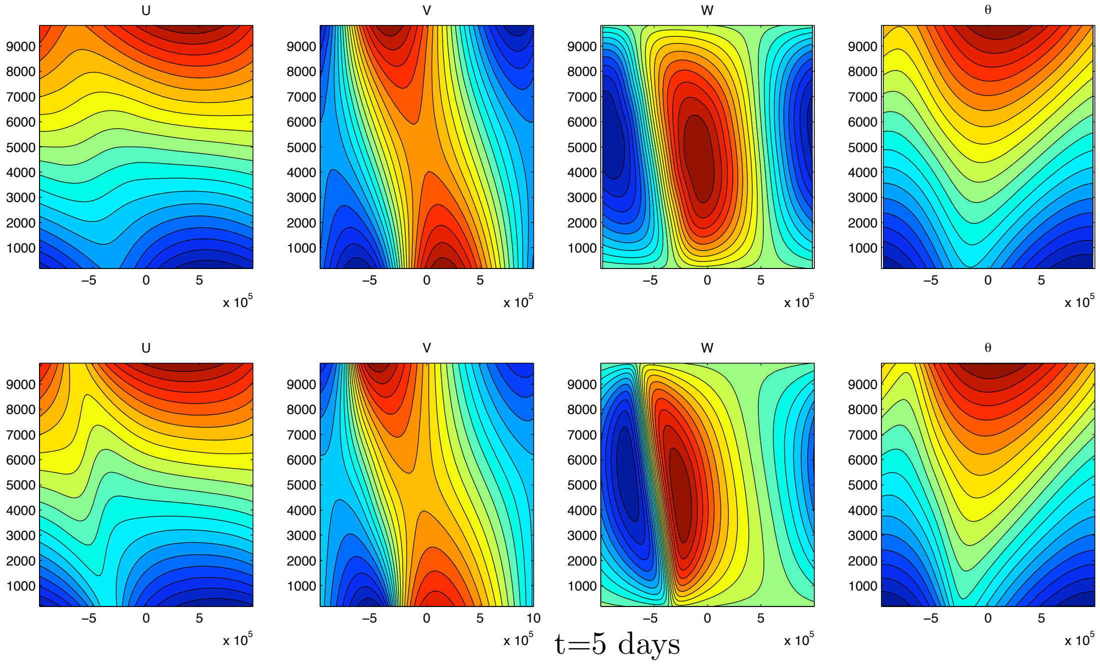

Figure 9: The front of the Eady problem at 5 days, with the uniform solution above and the moving mesh solution (obtained using the PV based monitor function) below. In both cases $n = m = 60$.

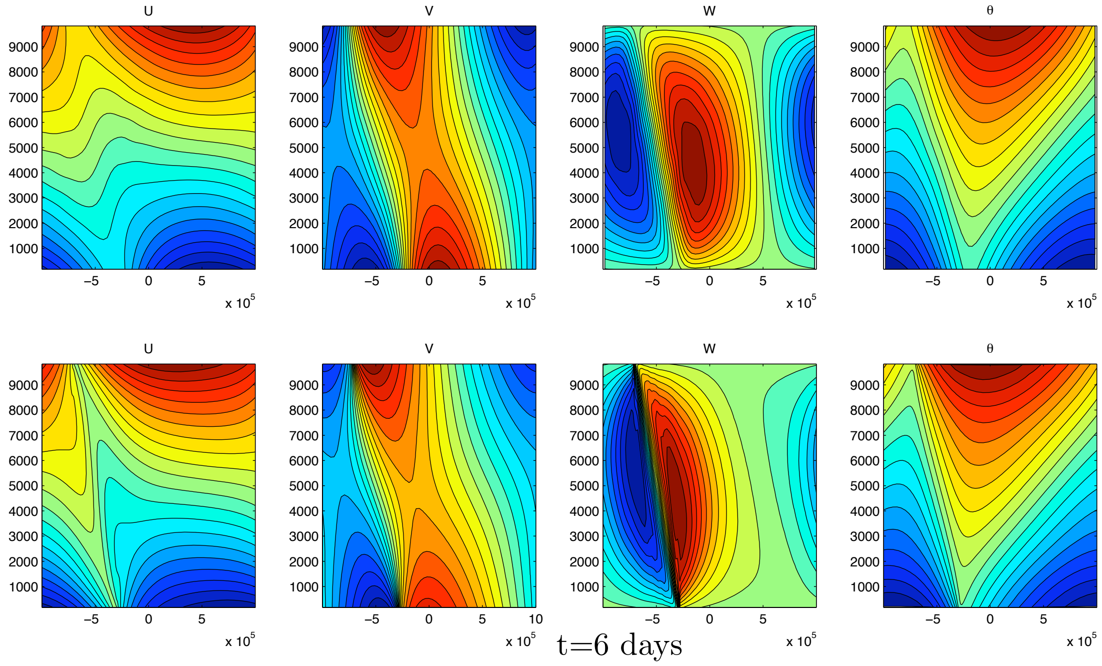

Figure 10: The front of the Eady problem at 6 days, with the uniform solution above and the moving mesh solution (obtained using the PV based monitor function) below. In both cases $n = m = 60$.

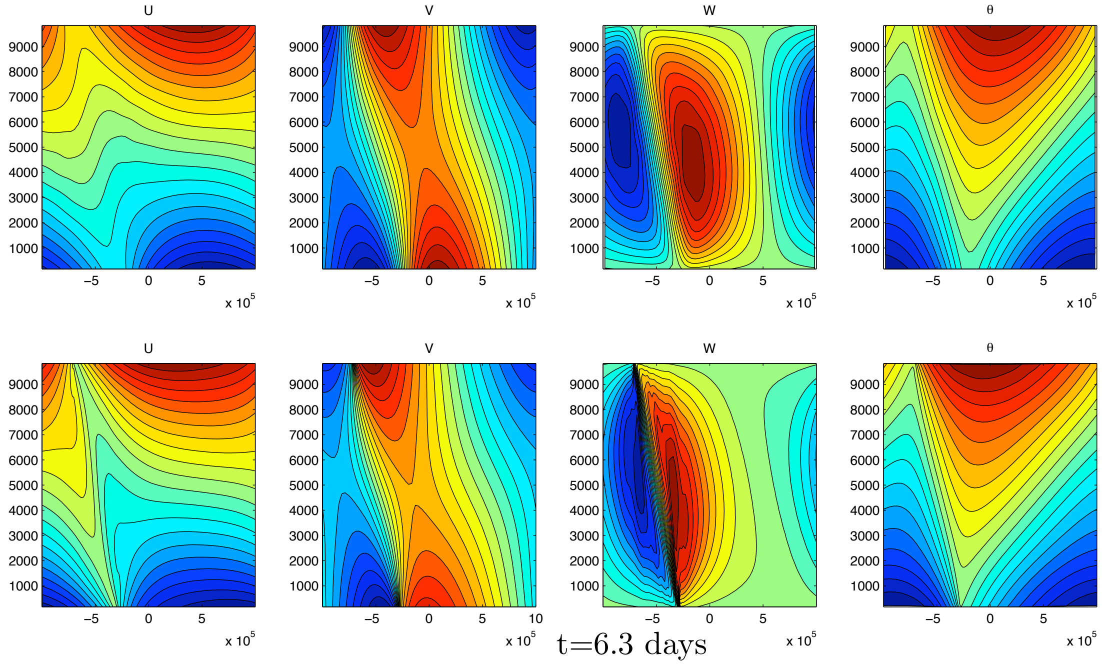

Figure 11: The front of the Eady problem at 6.3 days, with the uniform solution above and the moving mesh solution (obtained using the PV based monitor function) below. In both cases $n = m = 60$.

w and potential temperature θ. The singularity is forming at $x \approx -2700\mathrm{km}$. In these figures the results of the computations on the uniform mesh (in red) are compared with those of the moving mesh (blue). The results for W are most revealing. Observe that both the uniform and adapted mesh solutions have similar behaviour away from the singularity. This is a reassuring result as it indicates that the use of the adaptive mesh has not introduced global errors into the solution. Close to the singularity we see, in contrast, that the two solutions differ significantly. Indeed the resolution of the front is far sharper in the adaptive case, and we see a much greater variation in the solution as a result.

5.3.2 The computed mesh

We now consider in more detail the quality of the mesh generated to solve the Eady problem. Key questions here are the compression factor of the mesh, the scaling of neighbouring mesh elements and issues of orthogonality and skewness. In Figure 13 we present the complete mesh in the case of $t = 6$ days and $t = 6.3$ days, computed using the PV monitor function. (The meshes generated by the arc-length monitor function are almost identical.) Both figures show excellent mesh regularity. The mesh distant from the front is close to being a uniform regular rectangular grid. The front (which lies close to x =- 2500 km when $z = 0$ and $x = -6000$km when $z = 10$km) is well resolved, with a gradual mesh grading as the front is approached. There is excellent compression in the immediate vicinity of the front. At this macroscopic scale there is no evidence of mesh skewness or any significant loss of orthogonality. Observe both the good compression and the regularity of this mesh. The skewness of the mesh can be calculated using the following measure

$$s = \frac{1}{2}\left(\frac{\lambda_1}{\lambda_2} + \frac{\lambda_2}{\lambda_1}\right)$$

where λ_1 and λ_2 are the eigenvalues of the Jacobian matrix. In Figure 14 we can see that this measure is close to 1 in the vicinity of the front and away from the front, indicating that the mesh is very regular. In particular we see a gradual grading (in the x direction) of the mesh both into and away from the front. There is no grading in the z direction, showing that the mesh has correctly aligned with the front. Moreover, the elements in the physical domain are all close to being rectangles, with no evidence at all at this scale of mesh skewness or any sign of significant loss of orthogonality. As a comparison of the scaling factors for the meshes generated with the arc-length and PV based monitor functions respectively, we present a table showing the smallest and largest mesh sizes in the x and z directions respectively. These results are very similar, but show a slightly increased compression in the x-direction for the arc-length monitor function.

Monitor function	max Δx	min Δx	max Δz	min Δz
Arc length v	3E-02	1.8E-03	2.3E-02	9.7E-03
Max eigenvalue	3.5E-02	1.8E-03	2.3E-02	9.3E-03

6 Conclusions

We have shown in this paper that the moving mesh method based upon the Parabolic Monge Ampere equation can be successfully coupled to the solution of a convection dominated flow with a conservation law. In particular,

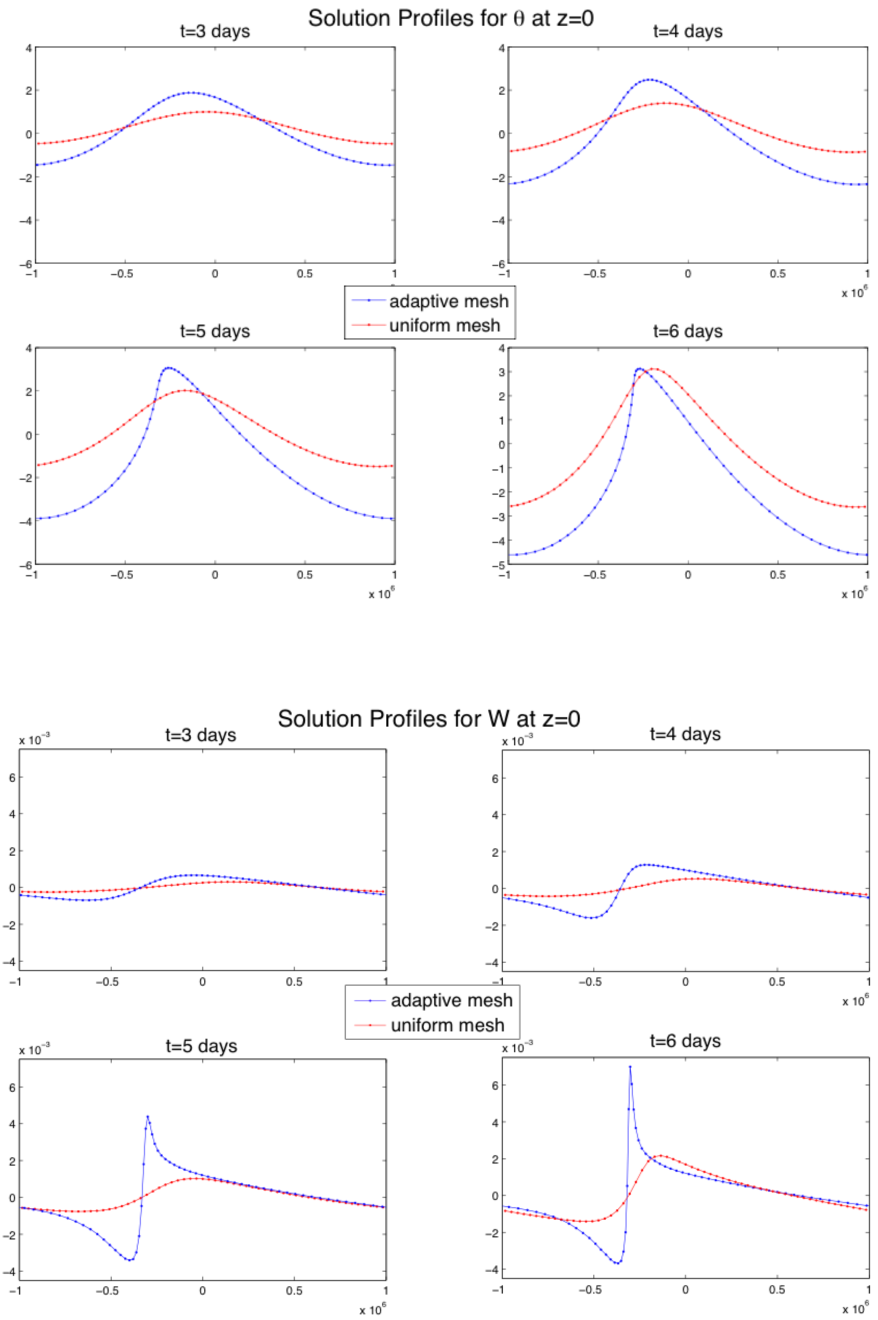

Figure 12: A cross section through the front of the Eady problem at 3,4,5 and 6 days, with the solution on the uniform mesh (with $n = m = 60$) in red, and the solution on the PV based moving mesh in blue. Observe the much greater resolution given by the moving mesh and also the absence of grid point oscillations in the solution despite the higher resolution.

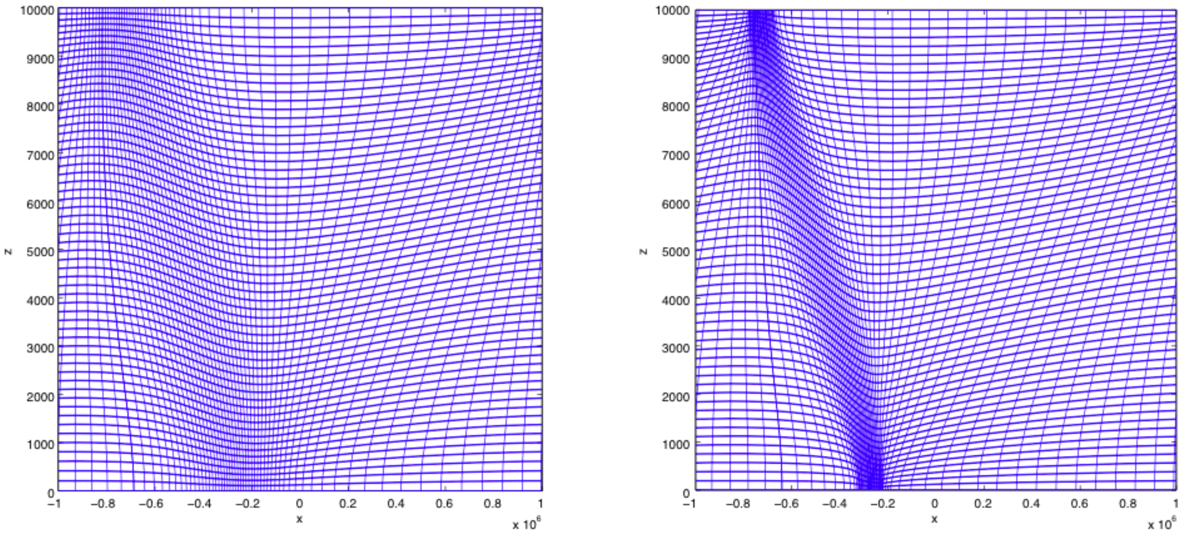

Figure 13: The evolving mesh used to compute the Eady problem with the PV monitor function at t=5 days (left) and $t = 6.3$ days (right).

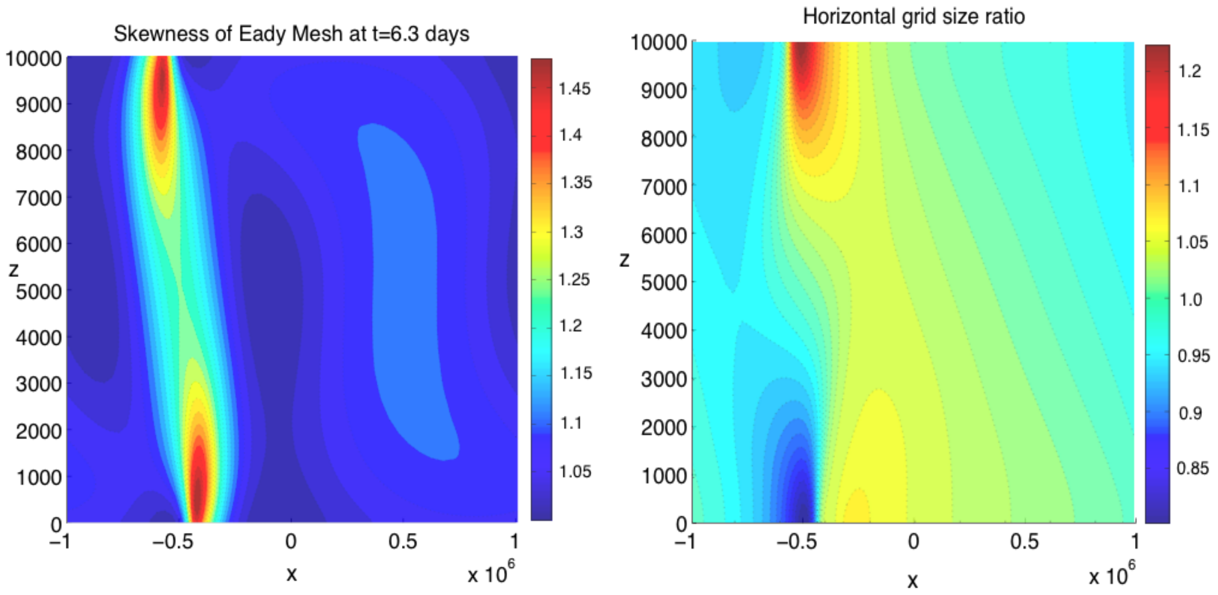

Figure 14: A figure to show the skewness measure s (left), and the horizontal grid ratio (right), for the Eady mesh at $t = 6.3$ days. The mesh has strong regularity in the vicinity of the front and a gradual graded scaling of Δx as the front is approached.

it has performed well in resolving the evolving front of the Eady problem. Indeed, the resolution of the adaptive mesh in the region of the weather front is 10 times greater than that on a uniform mesh with an equivalent number of grid points. This means there is the potential to resolve small scale features which are currently computationally too expensive to capture in an operational weather forecast. Ultimately this could lead to the location and magnitude of developing weather features being forecast much more accurately, and for longer periods of time. The Eady problem considered in this paper is of particular importance because satisfactory solutions have not been obtained by standard methods. It would now also be of interest to try other idealised meteorological problems where solutions are available in the literature, particular ones where standard methods appear not to work very well. The choice of appropriate monitor function and the coupling of the mesh equation to the underlying PDE are both challenging problems that require further exploration.

References

[1] J. Behrens, *Atmospheric and ocean modeling with an adaptive finite element solver for the shallow water equations*, Applied Numerical Mathematics 26, (1998), 217–226.

[2] J. Behrens, *Adaptive atmospheric modeling, key techniques in grid generation, data structures, and numerical operations with applications*, Springer Verlag, (2006).

[3] Y. Brenier, *Polar factorization and monotone rearrangement of vector- valued functions,* , Communications on Pure and Applied mathematics, **44** (1991), 375–417.

[4] Y. Brenier and M.J.P. Cullen, *Rigorous derivation of the $X - Z$ semigeostrophic equations*, Commun. Math. Sci., **7**, (2009), 779–784.

[5] C. J. Budd, W. Z. Huang, and R. D. Russell , *Moving mesh methods for problems with blow-up*, SIAM J. Sci. Comput., **17**,(1996), 305–327.

[6] C. J. Budd, W. Z. Huang, and R. D. Russell , *Adaptivity with moving grids*, Acta Numerica, **18**,(2009), 111–241.

[7] C. J. Budd and J. F. Williams, *Parabolic Monge-Ampère methods for blow-up problems in several spatial dimensions*, J. of Physics A, **39**, (2006) 5425–5444.

[8] C. J. Budd and J. F. Williams, *Moving mesh generation using the parabolic Monge-Ampère equation*, SIAM J. Sci. Comput., **31** (2009), 3438–3465

[9] L. Caffarelli, *Interior $W^{2,p}$ estimates for solutions of the Monge-Ampère equation* , Annals of Mathematics, **131** (1990), 135–150.

[10] H. D. Ceniceros, *A Semi-implicit moving mesh method for the focusing nonlinear Schrödinger equation*, Communications on pure and applied analysis, **4**, (2002) 1–14.

[11] H. D. Ceniceros and T. Y. Hou, *An efficient dynamically adaptive mesh for potentially singular solutions*, J. Comput. Phys., **172**, (2001), 609–639.

[12] M.J.P. Cullen and M. Feldman, *Lagrangian solutions of semigeostrophic equations in physical space*, SIAM J. Math. Anal., **37** (2006), 1371–1395.

[13] M.J.P. Cullen, *A Mathematical Theory of Large-Scale Atmosphere/Ocean Flow*, Imperial College Press, (2006).

[14] M.J.P. Cullen, *Modelling atmospheric flows*, Acta Numerica., **16**, (2007), 67–154.

[15] M.J. P. Cullen *A comparison of numerical solutions to the eady frontogenesis problem*, Q. J. Roy. Meteorol. Soc., **134**, (2008), 2143–2155.

[16] M.J.P. Cullen and I. Roulstone, *A geometric model of the nonlinear equilibrium of two-dimensional Eady waves*, J. Atmos. Sci. , **50**, (1993), 328–322

[17] G. Delzanno, L. Chacón, J. Finn, Y. Chung and G. Lapenta, *An optimal robust equidistribution method for two-dimensional grid adaptation* based on Monge-Kantorovich optimization', J. Comput. Phys., **227**(23), (2008), 9841 – 9864.

[18] R. Giddings and M.J.P. Cullen, *Mesh movement in Legendre transformations*, Int. J. Numerical Methods in Fluids, **56**, (2008), 1255–1260.

[19] W. Hackbusch, *Elliptic differential equations: theory and numerical treat- ment,* Springer, 1992.

[20] B.J. Hoskins, M.E. McIntyre and A.W. Robertson, *On the use and significance of isentropic potential vorticity maps*, Q. J. Roy. Meteorol. Soc., **111**, (1985), 877–946.

[21] W. Huang, Y. Ren, and R. D. Russell, *Moving mesh partial differential equations (MMPDES) based on the equidistribution principle*, SIAM J. Numer. Anal., **31**, (1994), 709–730.

[22] W. Huang, Y. Ren, and R. D. Russell, *Moving Mesh Methods Based on Moving Mesh Partial Differential Equations*, J. Comput. Phys., 113,279-290 (1994).

[23] W. Huang, Y. Ren, and R. D. Russell, *Moving Mesh Partial Differential Equations (MMPDES) based on the Equidistribution Principle*, SIAM J. Numer. Anal., Vol 31, No 3, 709-730 (1994).

[24] W. Huang and R. D. Russell, *Adaptive Moving Mesh Methods*, Springer, (2011).

[25] I. N. James, *Introduction to Circulating Atmospheres*, Cambridge University Press, **20**, (1995).

[26] Kühnlein, C. and Smolarkiewicz, P. K. and Dörnbrack, A. *Modelling atmospheric flows with adaptive moving meshes*, J. Comput. Phys., Volume 231, Issue 7, April 2012, 2741-2763.

[27] S.T. Li and L.R. Petzold, *Moving mesh methods with upwinding schemes for time dependent PDEs*, J. Comput. Phys., **131**, (1997), 368–377.

[28] S.T. Li, L.R. Petzold and Y. Ren, *Stability of moving mesh systems of partial differential equations*, SIAM J. Sci. Comput., **20**, (1998), 719–738.

[29] T. Melvin, M.Dubal, N. Wood, A. Staniforth, and M. Zerroukat, *An inherently mass-conserving iterative semi-implicit semi-lagrangian discretization of the non-hydrostatic vertical-slice equations*, Q. J. Roy. Meteorol. Soc., **136**, (2010), 799–814.

[30] N. Nakamura and I. Held, *Nonlinear equilibration of two-dimensional eady waves*, J. Atmos. Sci., **46**, (1989), 3055–3064.

[31] T. Tang, *Moving Mesh Methods for Computational Fluid Dynamics*, Recent Advances in Adaptive Computation (Z. Shi, Z. Chen, T. Tang, and D. Yu, eds.), Contemporary Mathematics, vol. 383, American Mathematical Society, 2005, Proceedings of the International Conference on Recent Advances in Adaptive Computation, May 2004, Hangzhou, China, pp. 141-173.

[32] T. Tang (2005), *Moving mesh methods for computational fluid dynamics*, Contemporary mathematics, **383**, (2005) 141–173.

[33] H. Tang and T. Tang (2003), *Adaptive mesh methods for one- and two- dimensional hyperbolic conservation laws*, SIAM J. Numer. Anal., **41**, (2003) 487–515.

[34] E. J. Walsh, *Moving Mesh Methods for Problems in Meteorology*, University of Bath Thesis, (2010).

[35] R. Williams, *Atmospheric frontogenesis: A numerical experiment* , J. Atmos. Sci., **24**, (1967), 627–641.

[36] A. Winslow, *Numerical solution of the quasilinear poisson equation in a nonuniform triangle mesh*, J. Comput. Phys., **1**, (1966) 149–172.

[37] Z. -R. Zhang and Tao Tang, *An Adaptive Mesh Redistribution Algorithm For Convection-Dominated Problems*, Communications on Pure and Applied Analysis, Volume 1, Number 3, September 2002.