

Numerical Solution of the Optimal Transport Problem for Mesh Generation on the Sphere

Hilary Weller^a, Philip Browne^a, Chris Budd^b, Mike Cullen^c

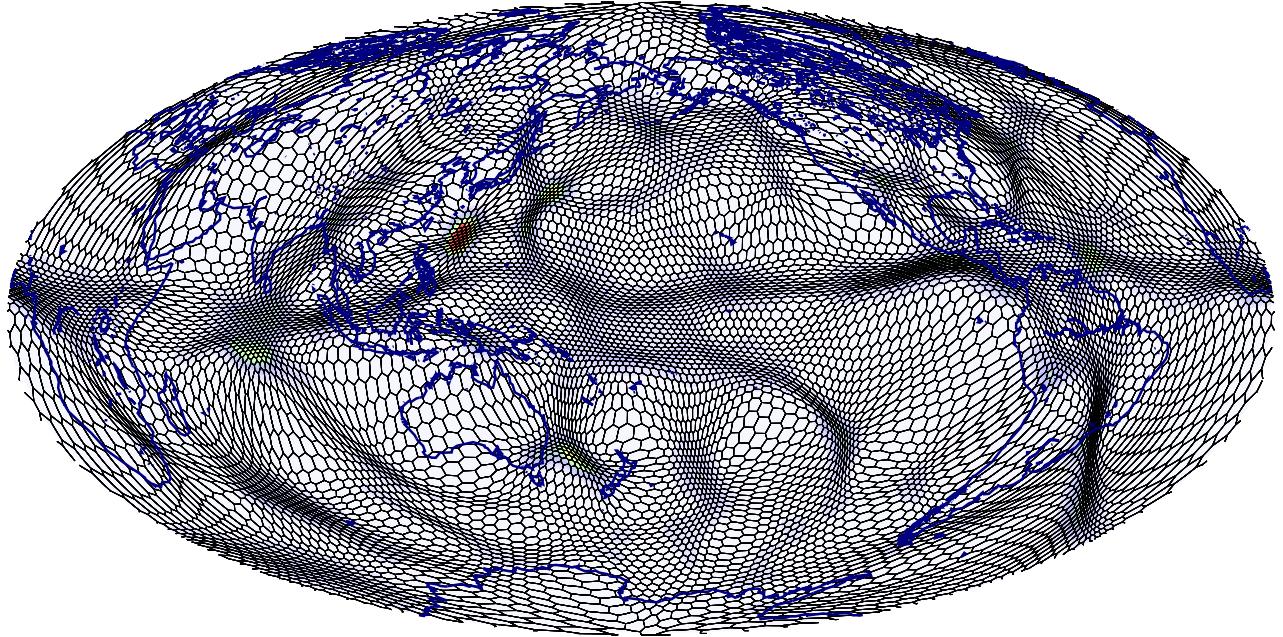
^a*Meteorology, University of Reading, UK*

^b*University of Bath, UK*

^c*Met Office, UK*

Abstract

Solved Monge-Ampére equation on the sphere for generating polygonal meshes ...



Keywords: Optimal Transport, Adaptive, Mesh, Refinement, Mesh generation, Monge-Ampére, Atmosphere, Modelling

Email address: h.weller@reading.ac.uk (Hilary Weller)

1. Introduction

The need to represent scale interactions in weather and climate prediction models has, for many decades, motivated research into the use of adaptive meshes [2, 25, 28]. R-adaptivity - mesh redistribution - involves deforming a mesh in order to vary local resolution and was first considered for atmospheric modelling more than twenty years ago by Dietachmayer and Droegeemeier [12]. It is an attractive form of adaptivity since it does not involve altering the mesh connectivity, does not create load balancing problems because points are never created or destroyed, does not require mapping of solutions between meshes [19], does not lead to sudden changes in resolution and can be retro-fitted into existing models. The solution of the Monge-Ampère equation to generate an optimally transported mesh based on a monitor function is a useful form of r-adaptive mesh generation because it generates a mesh equi-distributed with respect to a monitor function and does not lead to mesh tangling [4]. However the Monge-Ampère equation has not before been solved numerically on the surface of a sphere, which would be necessary for weather and climate prediction.

At first glance, r-adaptivity does not look ideal for adaptive meshing of the global atmosphere; Dietachmayer and Droegeemeier [12] pointed out that the resulting meshes can be quite distorted which leads to truncation errors and it is not possible to control the resolution in individual directions, just the total cell size (area or volume). With r-adaptivity, it is not possible, for example, to increase the total number of points around the equator, just re-distribute them [14]. However, if the mesh redistribution starts from a mesh with enough points around the equator, then these points can be redistributed according to transient features of the flow.

Models of the global atmosphere are being developed with accurate treatment of non-orthogonality and which allow arbitrary grid structures [9, 16, 20, 21, 29]. The time may therefore be right to reconsider r-adaptive modelling of the global atmosphere.

An interesting form of adaptivity that, like r-adaptivity, retains the same total number of points and is consequently suitable for good load balancing on parallel computers, is Voronoi meshing using a non-uniform density (or monitor) function to control the mesh spacing, using Lloyd's algorithm [24]. Lloyd's algorithm generates smoothly varying, orthogonal isotropic meshes suitable for finite-volume models and is being used by the Model for Prediction Across Scales (MPAS, 26). Lloyd's algorithm alters the mesh connectivity meaning that, if it is used

in conjunction with dynamic mesh adaptivity, mapping between old and new solutions would be needed and there is an additional layer of complexity involved with changing the data structures and moving information between parallel processors. Also, Lloyd’s algorithm is extremely expensive [18]. We therefore propose r-adaptivity which uses cheaper mesh generation and fixed data structures associated with the mesh.

In section 2 we review numerical methods of solving the Monge-Ampére equation in Euclidean geometry. In section 3 we describe the new numerical method for solving the Monge-Ampére equation both on a Euclidean plane and on the sphere. In order to address issues of mesh distortion, a range of diagnostics of mesh quality are presented. These diagnostics, along with the diagnostics of solution convergence, are described in section 4 and the diagnostics are presented in section 5. The meshes generated, both on the plane and on the sphere, are shown and described in section 5 and the meshes on the sphere are compared with Voronoi meshes generated using Lloyd’s algorithm [24] with the same monitor function. In order to demonstrate the performance of the mesh generation using real data as a monitor function, meshes are generated using a monitor function derived from reanalysis precipitation in section 5. Final conclusions are drawn in section 6.

2. A Review of Numerical Methods for solving the Monge-Ampére Equation

The fully non-linear, second-order, elliptic Monge-Ampére equation is:

$$|H(\phi(\xi))| = f(\xi, \phi) \quad (1)$$

for independent variable $\xi \in \Omega$ and $\Omega \subset \mathbb{R}^d$ where ϕ is the (scalar) dependent variable, f is a known scalar function of ξ and ϕ , $H = \nabla \nabla$ is the Hessian (the tensorial gradient of the gradient) and $|H|$ denotes the determinant of the Hessian. Froese and Oberman [15] give an excellent review of some numerical methods for solving this equation and this review draws from their review and includes some other techniques and insights.

There are two challenging parts to solving the Monge-Ampére equation. Firstly we need a spatial discretisation methods both for the Hessian, H and for the source term, f (although f is a known function, it can be a function of ϕ or $\nabla \phi$ so numerical approximations are necessary). The spatial discretisation leads to a set of non-linear algebraic equations. Secondly,

the algebraic equations require a numerical algorithm to find solutions. We will start by considering the spatial discretisation of the Hessian, H .

Budd et al. [4] used simple finite differences on a structured, Cartesian grid to discretise the Hessian, a technique that was extended to three dimensions by Browne et al. [3]. Oberman [22] describe a finite difference method that uses a wide stencil to calculate the Hessian on a structured Cartesian grid. This was extended to three dimensions by Froese and Oberman [15]. The wide stencil was needed to ensure monotonicity of the iterative solution. Feng and Neilan [13] approximate the Monge-Ampère equation by a fourth-order quasi-linear equation in order to used mixed finite elements for the spatial discretisation. Dean and Glowinski [10, 11] also use mixed finite-elements on triangulations of the unit plane. All of these techniques have been used on either 2D or 3D Euclidean geometry.

There has been less discussion of the discretisation methods needed to calculate f .

Once the Monge-Ampère equation is discretised in space, it is necessary to solve the resulting non-linear algebraic equations, the part of the method that we describe as the “algorithm”. Budd et al. [4] introduced a parabolic version of the Monge-Ampère equation which is solved by time-stepping, including an implicit relaxation term to smooth the transient solution and to speed up convergence. This effectively creates fixed-point iterations but is is not clear if these fixed-point iterations are optimally convergent. Benamou et al. [1] also used fixed-point iterations by linearising the two-dimensional Hessian term with a Laplacian, $\nabla^2\phi$:

$$|H(\phi)| = \frac{1}{2} (\nabla^2\phi)^2 - \frac{\phi_{xx}^2 + \phi_{yy}^2}{2} - \phi_{xy}^2 |.$$

After some manipulation, this results in a Poisson equation which can be solved implicitly with the non-linear terms on the right hand side. Froese and Oberman [15] describe this as a semi-implicit method and use it to find the starting point for a Newton method. A Newton method is a common approach for sovling the algebraic equations [8, 10, 15]. However the cost and complexity of the Newton method may not be necessary for mesh generation. In this paper we focus on fixed-point iterations, although the new linearisation proposed may also be beneficial for calculating the first guess for a Newton method.

The Monge-Ampère equation has not before been solved numerically on the sphere. The description of the optimally transported mesh problem using the Monge-Ampère equation relies on properties of Euclidean geometry [4]. Therefore the primary challenge is to write

down the optimal transport problem for mesh generation on the surface of the sphere.

These problems and proposed solutions will be described in section 3 where we will describe mesh generation by optimal transport.

3. Mesh Generation by Optimal Transport

3.1. Equations of Mesh Transport in Euclidean Space

This subject is covered in detail in Budd et al. [4]. A summary is given here, using the same notation as is used in following sections of this paper.

Optimally transported meshes in Euclidean space use a mesh potential, ϕ , to map from the original mesh (or computational space) to the new mesh (or physical space) so that the new mesh locations are given by:

$$\mathbf{x} = \boldsymbol{\xi} + \nabla\phi \quad (2)$$

where $\boldsymbol{\xi}$ are the locations of the old mesh points [4]. A mesh is equidistributed with respect to a monitor function when the product of the cell volumes and the monitor function is constant across all mesh cells. The equation for creating an equi-distributed mesh will use $|\nabla\mathbf{x}|$, the determinant of the gradient of \mathbf{x} with respect to $\boldsymbol{\xi}$. If $|\nabla\mathbf{x}|$ can be calculated exactly, then it is equal to the ratio of the cell volume for a cell with vertices at the positions \mathbf{x} to the cell volume with vertices at $\boldsymbol{\xi}$ (ie the change in cell volume under the map). Therefore, for the mesh to be equi-distributed with respect to a monitor function, m , the new mesh locations should satisfy

$$|\nabla\mathbf{x}| m(\mathbf{x}) = c \quad (3)$$

where c is a constant, uniform over space, which will be determined once the numerical method is defined. $|\nabla\mathbf{x}|$ can also be interpreted as the Jacobian of the map from computational space, with positions $\boldsymbol{\xi}$, to physical space, \mathbf{x} . Taking the determinant of the gradient of eqn. (2), we can see that $|\nabla\mathbf{x}| = |I + \nabla\nabla\phi| = |I + H(\phi)|$ where I is the identity tensor and H is the Hessian. Consequently, for the mesh to be equidistributed, the mesh potential, ϕ , must satisfy a Monge-Ampére equation:

$$|I + H(\phi)| m(\mathbf{x}) = c. \quad (4)$$

The presence of the identity tensor in this Monge-Ampére equation will be exploited in the linearisation to create a novel numerical algorithm.

The equidistribution principle leads directly to the optimal transport constraint that the transport map has to push forward a uniform measure to a prescribed non-uniform measure (ref). The map is then found by minimising a cost function, Euclidean distance squared or Riemannian distance squared according to the geometry. This minimises the mesh distortion (ref). In both cases a unique solution exists (McCann's paper and Villani's 2008 book need to be cited) and can be expressed in terms of a Monge-Ampère equation. They both satisfy monotonicity requirements which prevent mesh tangling. The optimal transport procedure also acts as a proof of existence of solutions for the Monge-Ampère equation.

3.2. Equations of Mesh Transport on the Sphere

On the surface of the sphere, the gradient of the mesh potential, $\nabla\phi$, at ξ lies on the tangent of the sphere and so does not map to another point on the sphere. Instead, we can use the exponential map

$$\mathbf{x} = \exp_{\xi} \nabla\phi$$

to take us from position ξ along the geodesic parallel with $\nabla\phi$ for a distance $|\nabla\phi|$ to \mathbf{x} . On the surface of the sphere, the Jacobian determinant term ($|\nabla\mathbf{x}|$ of eqn [3]) cannot be represented using local gradients at ξ because changes in volume need to take the curvature of the domain into account. Rather than calculating the appropriate metric terms, for our iterative solution, we simply use the ratio of new to old cell volumes under the map, denoted $r = V_\xi/V_x$ where V_ξ are the volumes of the original cells with vertices at positions ξ and V_x are the volumes of the transformed cells will vertices at positions \mathbf{x} . So the map, ϕ , must satisfy:

$$r(\phi) m(\mathbf{x}) = c \tag{5}$$

and this geometric equation must be solved on the surface of a sphere.

3.3. Outline of Numerical Algorithm

3.3.1. In Euclidian Space

A convergent, fixed-point iteration sequence to solve eqn. (4) can be found by observing that the linear terms of $|I + H(\phi)|$ are in fact $1 + \nabla^2\phi$ where ∇^2 is the Laplacian operator. Eqn. (4) can then be written as fixed-point iterations:

$$1 + \nabla^2\phi^{n+1} = 1 + \nabla^2\phi^n - |I + H(\phi^n)| + \frac{c^n}{m(\mathbf{x}^n)} \tag{6}$$

where n is the iteration number. This is simpler than the fixed-point iterations used by Benamou et al. [1], Feng and Neilan [13] because of the presence of the identity tensor in our Monge-Ampère equation which simplifies the linearisation.

Given suitable spatial discretisations, eqn. (6) can be solved for ϕ^{n+1} given known values ϕ^n . Assuming periodic boundary conditions, for the Poisson equation (6) to have a solution, c^n must take the value

$$c^n = \frac{\sum |I + H(\phi^n)| V_\xi}{\sum \frac{V_\xi}{m(\mathbf{x}^n)}}$$

where V_ξ are the volumes of the original, computational mesh cells and the summations are over all cells of the computational mesh.

In order to improve stability of the fixed-point iteration sequence, the Laplacian terms of eqn. (6) can be multiplied by a factor, $1 + \alpha$, where $\alpha \geq 0$:

$$(1 + \alpha) \nabla^2 \phi^{n+1} = (1 + \alpha) \nabla^2 \phi^n - \left| I + H(\phi^n) \right| + \frac{c^n}{m(\mathbf{x}^n)} \quad (7)$$

which clearly has no affect on a converged solution but will alter the convergence of the fixed-point iterations used to find ϕ . This is a form of under-relaxation and the value of α will be defined in section 3.5.1.

3.3.2. On the Surface of the Sphere

In order to define convergent fixed-point iterations, we need to find a linearisation of eqn. (5). Assuming that eqn. (5) can be approximated by eqn. (4) on a tangent plane to the sphere, we use the same linearisation as in section 3.3.1 and the same fixed-point iteration sequence:

$$(1 + \alpha) \nabla \phi^{n+1} = (1 + \alpha) \nabla^2 \phi^n - r(\phi^n) + \frac{c^n}{m(\mathbf{x}^n)} \quad (8)$$

since $r = V_\xi/V_x$ can be computed explicitly at every iteration. The linearisation will now be less accurate than in the Euclidean case due to the curvature of the sphere, so it may be necessary to increase α further to avoid divergence.

The fixed-point iteration sequences defined in eqns. (7) and (8) are preferred to the solution of the parabolic Monge-Ampère equation as used by Browne et al. [3], Budd et al. [5] since the Laplacian term used above is a linearisation of the Hessian and so will accelerate convergence to the given monitor function whereas the Laplacian used by Browne et al. [3], Budd et al. [5]

smoothed the solution away from equidistribution at each iteration. However there may be further scope for improvement by combining the best aspects of the two approaches.

3.4. Spatial Discretisation

For the solution of eqns. (7,8) to be smooth and convergent, there are a number of desirable properties of the spatial discretisation:

1. The discretisation of $|I + H(\phi^n)|$ should be consistent with the discretisation of $1 + \nabla^2\phi$ otherwise the linearisation will not be good and the iterative solution will not converge quickly. In this context, consistent means that the trace of the discretised $H(\phi)$ must be equal to $\nabla^2\phi$, as occurs analytically. This is only possible when solving eqn (7), not eqn. (8) since the relationship between r and $1 + \nabla^2\phi$ is not known numerically.
2. The spatial discretisation should be at least second-order accurate and the errors should be smooth. If we have rough truncation errors or truncation errors which are proportional to the mesh size, Δx , and we are deforming mesh cells of size Δx to size, say $0.1\Delta x$, then truncation errors will lead to mesh tangling.
3. To avoid grid-scale oscillations in the solution of ϕ , the spatial discretisation should be as compact as possible so that grid-scale oscillations of ϕ are not hidden in the discretisations of $|I + H(\phi^n)|$ and $m(\mathbf{x})$.
4. If the solution, ϕ , is convex or locally convex, then convex cells in the initial mesh should remain convex in the mapped mesh. This implies that $\nabla\phi$ should be sufficiently smooth but the analysis has not been done to find how smooth $\nabla\phi$ should be.¹

We are considering a finite-volume discretisation on initially orthogonal nearly uniform polygonal prisms. This and the above requirements suggests the following spatial discretisation:

¹We postulate that, following the TRiSK discretisation on polygons [27], the divergence of the mesh map on the initial (primal) mesh should be a convex combination of the divergence of the mesh map if it were calculated on a dual mesh (eg a triangulation).

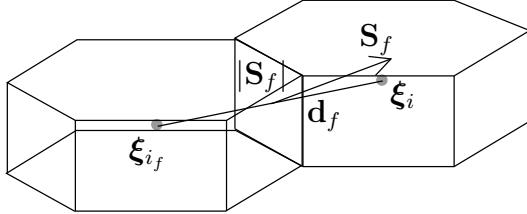


Figure 1: Cell centre locations, ξ_i and ξ_{i_f} of cells i and i_f , face area vector \mathbf{S}_f and vector between cell centres, \mathbf{d}_f .

3.4.1. Discretisation of the Laplacian

For cell i with faces $f \in i$ and neighbour cells i_f , the simplest, most compact discretisation of the Laplacian, suitable for an orthogonal grid, using Gauss's divergence theorem, is:

$$\nabla_i^2 \phi \approx \frac{1}{V_i} \sum_{f \in i} \nabla_{nf} (\phi) |\mathbf{S}_f|$$

where cell i has volume V_i , \mathbf{S}_f is the outward pointing normal vector to cell i at face f with area equal to the face area so that $|\mathbf{S}_f|$ is the face area and the normal gradient at each face is:

$$\nabla_{nf} \phi = \frac{\phi_{i_f} - \phi_i}{|\mathbf{d}_f|} \quad (9)$$

where $|\mathbf{d}_f|$ is the (geodesic) distance between cell centre i and i_f . This simple form ensures curl free pressure gradients (assuming that the curl is calculated using Stokes circulation theorem around every edge of the 3D mesh). If cell i has centre ξ_i then $\mathbf{d}_f = \xi_i - \xi_{i_f}$ in Euclidean geometry. On the surface of the sphere, $|\mathbf{d}_f|$ is the great circle distance between ξ_i and ξ_{i_f} . Locations ξ_i and ξ_{i_f} , vector \mathbf{S}_f and \mathbf{d}_f for cells i and i_f are shown in fig. 1.

3.4.2. Discretisation of the Hessian

Two approaches are taken to calculate the Hessian. The first we call a finite-difference approach (which uses both finite volume and finite difference approximations). The second uses the fact that, in solving the Monge-Ampère equation for mesh generation, we are approximating the change in cell volume by the determinant of the Hessian. Therefore, rather than calculating a discretised Hessian, we can simply use the change in cell volume, r . This is the geometric approach.

3.4.2.1. Finite Difference Discretisation of the Hessian. For a discretisation of the Hessian consistent with the discretisation of the Laplacian, we use Gauss's theorem:

$$H(\phi)_i = \nabla \nabla_i \phi = \frac{1}{V_i} \sum_{f \in i} \nabla_f(\phi) \mathbf{S}_f \quad (10)$$

where $\nabla_f \phi$ is the vector gradient of ϕ located at face f . The vector gradient, $\nabla_f \phi$, is reconstructed from normal components, $\nabla_{nf} \phi$ using a least-squares fit which is derived by assuming that $\nabla \phi$ is uniform so that it is first-order accurate on non-uniform meshes. This approach starts by reconstructing a cell-centred gradient from surrounding normal gradients using a least squares fit:

$$\nabla_i \phi = \left(\sum_{f \in i} \hat{\mathbf{S}}_f \mathbf{S}_f^T \right)^{-1} \sum_{f \in i} \nabla_{nf} \phi \mathbf{S}_f. \quad (11)$$

Next, a temporary value of the vector valued gradient at each face is calculated:

$$\nabla'_f \phi = \lambda_f \nabla_i \phi + (1 - \lambda_f) \nabla_{i_f} \phi$$

where λ_f is the coefficient for linear interpolation. For consistency with the Laplacian, we must have $\nabla_f \phi \cdot \mathbf{S}_f = \nabla_{nf} \phi |\mathbf{S}_f|$ which can be enforced with an explicit correction:

$$\nabla_f \phi = \nabla'_f \phi + \left(\nabla_{nf} \phi - \nabla'_f \phi \cdot \hat{\mathbf{S}}_f \right) \hat{\mathbf{S}}_f. \quad (12)$$

The Hessian calculated using eqn. (10) is not symmetric, as the analytic version would be.

3.4.2.2. Geometric approach to calculating the Hessian. A numerical approximation to calculating H will introduce truncation errors so instead we can simply use the change in cell volume:

$$r_i = |I + H_i(\phi)| = \frac{V_i(\mathbf{x})}{V_i(\boldsymbol{\xi})}$$

where $V_i(\mathbf{x})$ is the volume of the transported mesh cell i and $V_i(\boldsymbol{\xi})$ is the volume of the original cell. Volumes are calculated on the surface of the sphere by decomposing every polyhedron (on the original and new meshes) into tetrahedra with curved surfaces which are flat in spherical geometry. The volumes of these tetrahedra are found using the formula for the area of a spherical triangle.

3.4.3. The Gradient at the Vertices

In order to calculate the mesh map and consequently to calculate m , we must calculate $\nabla\phi$ at the mesh vertices. Ideally, this operation should not produce any non-convex cells and it turns out to be particularly sensitive to the numerical approximation and its stencil. If the stencil is too large, grid-scale oscillations in ϕ can grow which are not seen in $\nabla\phi$ and convergence is slow. If the stencil is too small, on a hexagonal mesh, grid-scale oscillations of the approximation of the gradient at vertices can grow since the calculation of the gradient is not smooth. This does not slow down convergence but the resulting meshes are locally distorted. So the approximation and the stencil need to be just right - we need the Goldilocks gradient.

3.4.3.1. Vertex Gradient using a Large Stencil. In order to calculate $\nabla_v\phi$ at the vertices, $\nabla_f\phi$ at the faces is calculated using eqn. (12). These values are then interpolated onto the vertices using linear interpolation. On a mesh of squares, four values of $\nabla_f\phi$ are averaged to calculate each $\nabla_v\phi$ at a vertex and on a mesh of hexagons, three values of $\nabla_f\phi$ are averaged to calculate each $\nabla_v\phi$. Including the calculation of $\nabla_f\phi$, the reconstruction of $\nabla_v\phi$ from ϕ uses a stencil of 10 hexagons on a hexagonal mesh and 12 squares on a mesh of squares. Due to these large stencils, the gradients calculated are smooth even if the ϕ field is not smooth. Due to the averaging (interpolation) of the gradient from the cell centres to the vertices there will be some consistency between gradients at different vertices and so cells may remain convex.

3.4.3.2. Vertex Gradient using a Small Stencil. The vector gradient at each vertex, $\nabla_v\phi$, can be reconstructed directly from the normal component of the gradient at the surrounding faces using a least squares fit

$$\nabla_v\phi = \left(\sum_{f \in v} \mathbf{d}_f \mathbf{d}_f^T \right)^{-1} \sum_{f \in v} (\mathbf{d}_f \nabla_{nf}\phi) \quad (13)$$

where $f \in v$ is the set of faces which share vertex v . This approximation is exact for a uniform vector field, $\nabla\phi$, and is consequently first order accurate on an arbitrary mesh. However on a hexagonal mesh, eqn. (13) only uses information from three surrounding faces and three surrounding hexagons and the resulting gradients are prone to grid-scale oscillations. The small amount of information used at every vertex means that neighbouring vertices can have

very different gradients. We therefore need a larger stencil, but not as large as the stencil used in section 3.4.3.1.

On a mesh of squares, ϕ at four squares is sufficient to reconstruct a smooth $\nabla_v \phi$ to second order.

3.4.3.3. Vertex Gradient using the Goldilocks Stencil. The vertex gradient reconstruction described in section 3.4.3.1 has a very large stencil (10 hexagons or 12 squares), includes interpolation and calculates a smooth gradient field. Consequently it could let grid-scale oscillations in ϕ grow since grid scale oscillations in ϕ may be hidden in the gradient field. The vertex gradient using a stencil of 3 hexagons (section 3.4.3.2) is only first-order accurate, even on a uniform mesh of hexagons since only 3 hexagons are used to reconstruct the gradient. When this small stencil is used, a noisy gradient field is calculated (which includes grid-scale oscillations) even when the ϕ field is smooth. Therefore we need the Goldilocks stencil, large enough to calculate a smooth gradient but without including averaging and thus hiding grid-scale oscillations in ϕ . This stencil includes the faces which share vertex v and the face neighbours of those faces. Alternatively, considering the dual mesh of triangles (in 2D), the stencil consists of the edges of the triangle around the vertex v and the edges of the edge neighbour triangles, as shown in fig. 2. The vertex gradient is then reconstructed using a least squares fit:

$$\nabla_v \phi = \left(\sum_{f \in v \in f \in v} \mathbf{d}_f \mathbf{d}_f^T \right)^{-1} \sum_{f \in v \in f \in v} (\mathbf{d}_f \nabla_{nf} \phi) \quad (14)$$

where $f \in v \in f \in v$ is the set of faces shown by dashed lines in fig. 2. In the least squares fit in eqn. (14), the central faces are counted three times (making the fit more accurate near the centre, following Weller et al. [30]).

3.4.4. Linear equation solver and outer iterations

The Laplacian is solved implicitly in eqns (7,8) using the OpenFOAM GAMG solver (geometric algebraic multi-grid, [23]) using diagonal incomplete Cholesky smoothing with 50 cells in the coarsest level. At each outer iteration (ie each value of n in eqn. (7)) the Laplacian is solved once with a relative tolerance of 0.001, so that the final residual is 0.001 times the initial residual. Then the right hand side of eqn. (7) is updated, incrementing n and the Laplacian is solved again until the initial residual is less than 10^{-8} . A weaker tolerance is probably

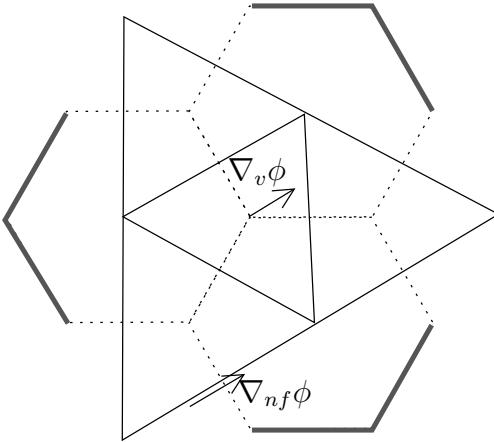


Figure 2: The Goldilocks stencil (dashed lines or triangle edges) for reconstructing the full gradient of the mesh potential, $\nabla_v\phi$, at the central vertex, v , from the component of the gradient along each of the triangle edges, $\nabla_{nf}\phi$. This results in a stencil of six hexagons to calculate ϕ at a vertex.

acceptable for mesh generation but we are using a tight tolerance to have more confidence that the numerical method is convergent.

3.4.5. Moving Voronoi Generating Points

If the initial mesh is Voronoi and it is required that the transported mesh is also Voronoi, then the Voronoi generating points can be moved using eqn. (2) using the cell centre gradient, reconstructed from the face gradient using volume weighting. However, a transported Voronoi mesh may not have exactly the same connectivity due to edge swapping in the Delaunay algorithm. This technique therefore may not be so suitable for r-adaptivity.

3.4.6. Calculating the Monitor Function

When using r-adaptivity, the mesh monitor function (that controls the mesh density) will need to be mapped from the previous mesh onto the new transported mesh so that it can be evaluated when solving eqns. (7) or (8). In idealised situations, the monitor function may take an analytic form so its evaluation at the new transported positions is straightforward. In this work, the monitor function is calculated analytically at the transported mesh cell centres.

What about for precipitation ...

3.5. Enforcing Stability

3.5.1. Under-relaxation

Here we describe how α is calculated. We start by defining the source terms of these equations to be $s^n = \left| I + H(\phi^n) \right| - \frac{c}{m(\mathbf{x}^n)}$ and $s^n = r(\phi^n) - \frac{c}{m(\mathbf{x}^n)}$ respectively. For convergence to occur, we would like the source term to decrease relative to the Laplacian term, $\nabla^2\phi$. Initially, the source term has order 1. In the tests undertaken, both on the plane and on a sphere, it has been sufficient to keep the ratio of the Laplacian to the source term greater than four and to always ensure that α increases with iteration number, n . So α is set to be:

$$1 + \alpha^{n+1} = \max \left(1 + \alpha^n, 4 \max \left(\frac{1}{4}, \max(|s^n|) \right) \right)$$

3.5.2. Smoothing the Monitor Function

Following Browne et al. [3], we experimented with smoothing the monitor function and this smoothing certainly improved convergence and generated meshes with smoother grading and hence lower anisotropy and skewness and better orthogonality. However the purpose of this work is to describe a robust solution of the Monge-Ampère equation on the sphere for any monitor function. So smoothing of the monitor function will not be considered further in this work.

What about for precipitation ...

4. Diagnostics of Convergence and of Mesh Quality

Convergence is measured in two ways. Firstly by plotting the initial residual of the matrix solution at every iteration as a function of iteration number. For a solution of the matrix equation $A\mathbf{x} = \mathbf{b}$, the initial residual is given by

$$\frac{\sum |\mathbf{b} - A\mathbf{x}^0|}{\sum |\mathbf{b}| + |A\mathbf{x}^0|}$$

where the sum is over all cells of the mesh (ie over all elements of the vectors \mathbf{b} and $A\mathbf{x}^0$) and \mathbf{x}^0 is the initial value in the iterative solution. The convergence of the final solution is assessed by plotting the change in cell area at the final iteration for each cell as a function of distance to the axis of symmetry in comparison to $\frac{c}{m(\mathbf{x}^n)}$. This tells us where the solution is not converging to the required mesh density and also, for solutions using $\left| I + H(\phi^n) \right|$ instead of $r(\phi^n)$ (ie

using the finite difference Hessian rather than the geometric Hessian) in the Monge-Ampére equation, it tells us how well $|I + H(\phi^n)|$ approximates $r(\phi^n)$.

A few diagnostics of mesh quality will be used. Firstly, the mesh spacing, $|\mathbf{d}|$, for each cell face as a function of distance to the axis of symmetry in comparison to that prescribed by the monitor function. This informs us about the aspect ratios of the cells. Other mesh distortions are measured by the skewness and the non-orthogonality. The non-orthogonality for each cell face is measured as

$$\cos^{-1} \frac{\mathbf{S}_x \cdot \mathbf{d}_x}{|\mathbf{S}_x||\mathbf{d}_x|}$$

where \mathbf{S}_x and \mathbf{d}_x are the cell face area vector and vector between cell centres respectively for the deformed mesh (see fig 1). The skewness of each cell face is measured as the distance between the face centre and the crossing point between the vector \mathbf{d}_x with the face, normalised by $|\mathbf{d}_x|$.

5. Results

Optimally transported meshes are generated in two-dimensional planar geometry to compare with those generated by numerical solution of the parabolic Monge-Ampére equation by Budd et al. [6]. Meshes are generated on the surface of the sphere in order to compare with the meshes generated by Ringler et al. [24] using Lloyd's algorithm.

5.1. Optimally Transported Meshes in Euclidean Geometry

Meshes are generated on a finite plane using the radially symmetric monitor function used by Budd et al. [6] defined for each location \mathbf{x}_i :

$$m(\mathbf{x}_i) = 1 + \alpha_1 \operatorname{sech}^2(\alpha_2(R^2 - a^2)) \quad (15)$$

where \mathbf{x}_c is the centre of the refined region (the origin for these results), R is the distance of \mathbf{x}_i to \mathbf{x}_c and α_1 , α_2 and a control the variations of the density function. Following Budd et al. [6] we generate two types of mesh with this monitor function, the first we call the ring mesh using $a = 0.25$, $\alpha_1 = 10$ and $\alpha_2 = 200$ and the second the bell mesh using $a = 0$, $\alpha_1 = 50$ and $\alpha_2 = 100$, both using periodic boundary conditions for ϕ .

The ring and bell meshes generated using both the finite difference and the geometric Hessian on the plane are shown in figure 3. The convergence diagnostics are shown in figures

4 and 5. Mesh quality for these meshes was analysed by Budd et al. [6] and this is not repeated here.

The meshes in fig. 3 calculated using both Hessian techniques are similar to each other and they are also similar to the meshes generated by Budd et al. [6]. However, the meshes calculated using the geometric Hessian conform more closely to the monitor function.

5.2. Convergence of the Monge-Ampère Solution in Euclidean Geometry

Figure 4 on the left shows the initial residual of the matrix solution as a function of iteration number for the calculation of all of the meshes of the sphere. Using the finite difference Hessian, the solution converges rapidly but convergence stalls when using the geometric Hessian. There are two possible reasons for the stalling. Firstly, the Laplacian is no longer a good linearisation of the geometric Hessian and secondly, a solution at this resolution may not exist. Smoothing the monitor function removes the stall in convergence and speeds convergence of all solutions (not shown). However this is not the topic of this paper.

The underrelaxation factor, $1 + \alpha$, is shown in the right of fig. 4. It never rises above the initial value because the source term never increases above its initial value. The initial value of $1 + \alpha$ is simply $4 \max |1 - \frac{c}{m}|$ and so $1 + \alpha$ is independent of the Hessian calculation method.

In order to diagnose how closely the final mesh equidistributes the monitor function, we plot the cell area as a scatter plot for every cell in the mesh as a function of the distance from the axis of symmetry in fig. 5 in comparison with $\frac{c}{m}$. Using the finite difference Hessian, there are discrepancies between $\frac{c}{m}$ and the cell area where the second derivative of $\frac{c}{m}$ is high. This is because the discrete calculation of the Hessian is not a good approximation of the cell area in these regions, where the derivatives of ϕ are varying rapidly. However, for the purpose of mesh generation, these discrepancies do not look problematic. If the optimally transported mesh is smoother than that specified by the monitor function then it could be beneficial. The meshes generated using the geometric Hessian fit the monitor function very closely, implying that the convergence is closer using the geometric Hessian, despite the lack of convergence of the initial residual.

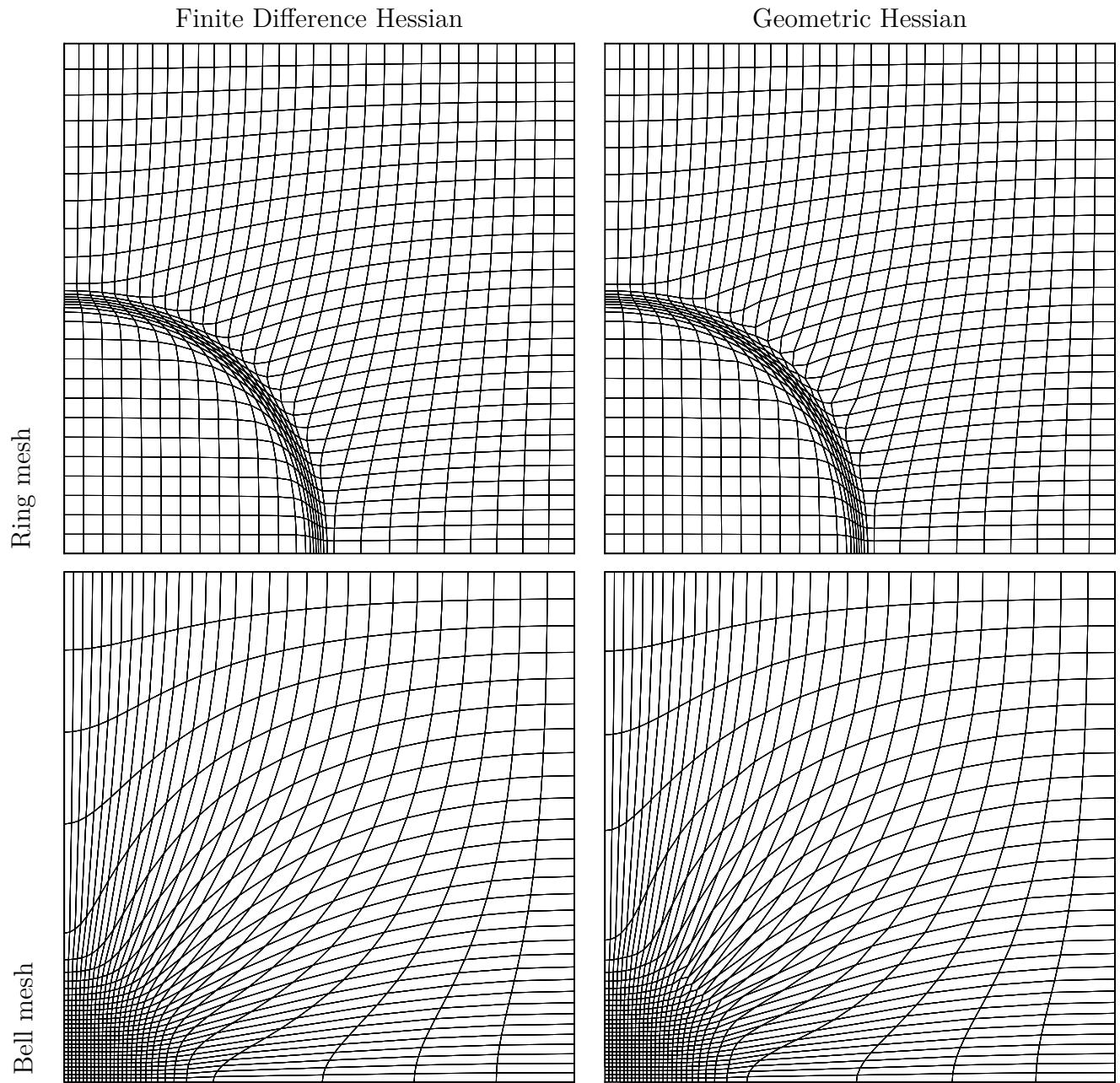


Figure 3: A quarter of the ring and bell meshes using the finite difference and volume Hessians.

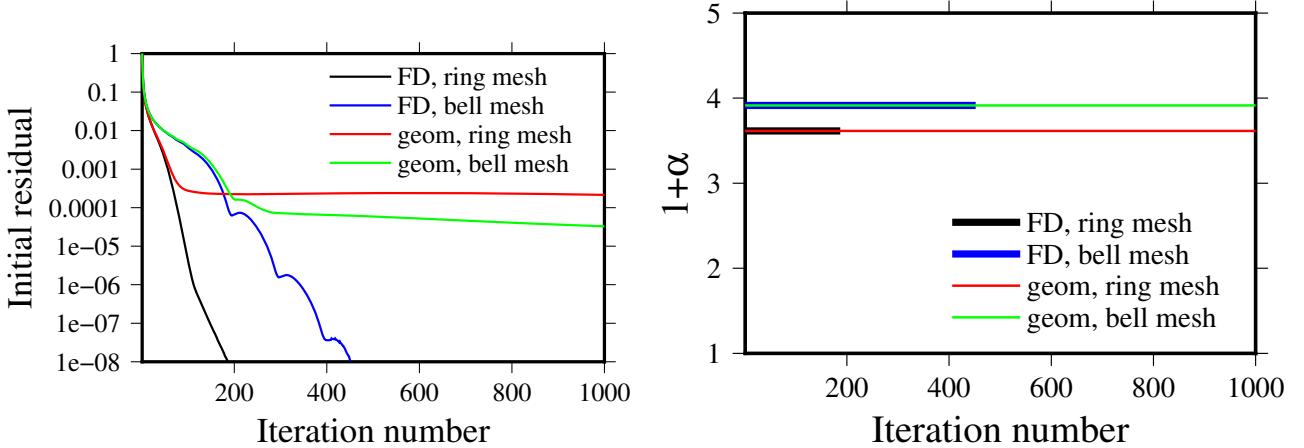


Figure 4: Convergence of initial residual and $1 + \alpha$ as a function of iteration number for the planar meshes generated with the monitor function defined in eqn. (15)..

5.3. Spherical Meshes

Meshes are generated using the geometric Hessian in order to compare with the locally refined meshes generated using Lloyd's algorithm by Ringler et al. [24]. We use a monitor function given by the square root of the density function of the corrected eqn. (4) of Ringler et al. [24]:

$$m(\mathbf{x}_i) = \sqrt{\frac{1}{2(1+\gamma)} \left(\tanh \frac{\beta - \|\mathbf{x}_c - \mathbf{x}_i\|}{\alpha} + 1 \right) + \gamma}$$

where \mathbf{x}_c is the centre of the refined region which has a latitude of 30° and a longitude of 90° . $\|\mathbf{x}_c - \mathbf{x}_i\|$ is the geodesic distance between the points and is computed as $\cos^{-1}(\mathbf{x}_c \cdot \mathbf{x}_i)$. α and β are in radians and they control the size of the refined region and the distance over which the mesh changes from fine to coarse resolution. We follow Ringler et al. [24] and use $\alpha = \pi/20$ and $\beta = \pi/6$. γ controls the ratio between the finest and coarsest resolution and we use $\gamma = (\frac{1}{2})^4$, $\gamma = (\frac{1}{4})^4$, $\gamma = (\frac{1}{8})^4$ and $\gamma = (\frac{1}{16})^4$ for meshes with finest mesh spacing factors of 2, 4, 8 and 16 times smaller than that of the coarsest. Following Ringler et al. [24], these meshes are referred to as X2, X4, X8 and X16.

The X1 meshes are not shown but the X1 meshes for Lloyd's algorithm and for the Monge-Ampére meshes are slightly different. The X1 Lloyd's algorithm meshes are generated using Lloyd's algorithm which guarantees that the X1 meshes are centroidal (the Voronoi generating point is co-located with the cell centre) whereas the X1 meshes for the Monge-Ampére equation are the Heikes and Randall [17] version of the hexagonal icosahedron, optimised

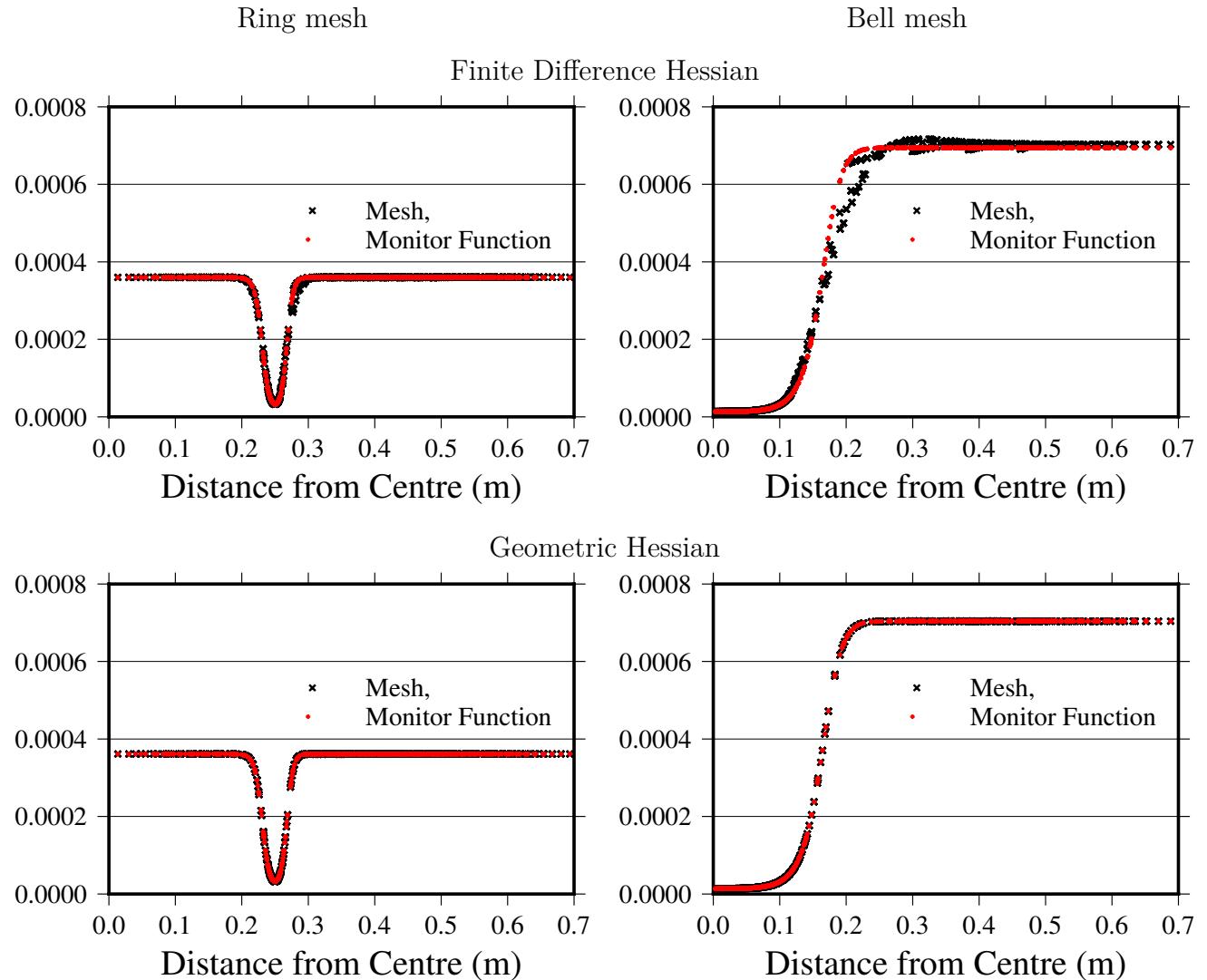


Figure 5: Cell area as a function of distance from the axis of symmetry in comparison to that implied by the monitor function for the planar meshes generated with the monitor function defined in eqn. (15)..

to reduce skewness. The X2, X4, X8 and X16 meshes are shown in figure 6 with the ratio between the cell area and the average cell area coloured.

The Lloyd's algorithm meshes in fig. 6 are orthogonal, close to centroidal and the mesh topology (connectivity) is different for all the refinement levels. (Lloyd's algorithm generates meshes that are centroidal relative to a density function which means that they are not centroidal when the centroid is simply the centre of mass.) The optimally transported meshes all have the same connectivity and they are centroidal but not orthogonal. (Orthogonality could be achieved by Voronoi tesselating the meshes, at the expense of centroidality, see section 5.6.)

All of the Monge-Ampére meshes in fig. 6 have regions of anisotropy in between the fine and coarse regions whereas the Lloyd algorithm meshes remain isotropic and the mesh topology changes between resolutions. The anisotropy will be investigated further in section 5.5. Before looking in more detail at the mesh quality in section 5.5, we will examine diagnostics of convergence in section 5.4.

5.4. Convergence of the Monge-Ampére Solution on the Sphere

Convergence of the initial residual is shown in fig. 5.4 for X2-X16 meshes of various resolutions. Convergence is rapid for the X2 and X4 meshes but slows after around 100 iterations, once the non-linearities have grown and the Laplacian is no longer a good approximation for the Hessian and once the exact solutions becomes difficult to achieve at finite resolution.

The underrelaxation factor, $1+\alpha$, is shown in fig. 8. Unlike in the Euclidean case, $1+\alpha$ does rise after initialisation. This implies that the maximum of the source term increases before it decreases. However the initial residual monotonically decreasing during these early iterations. This is because the initial residual is a mean over the whole domain whereas $1+\alpha$ is set from the maximum of the source term.

The convergence of the cell area with the monitor function is shown in fig. 9 as scatter plots of cell area change as a function of distance to the centre of the refined region. As occurred in Euclidean geometry, the geometric Hessian gives accurate equidistribution.

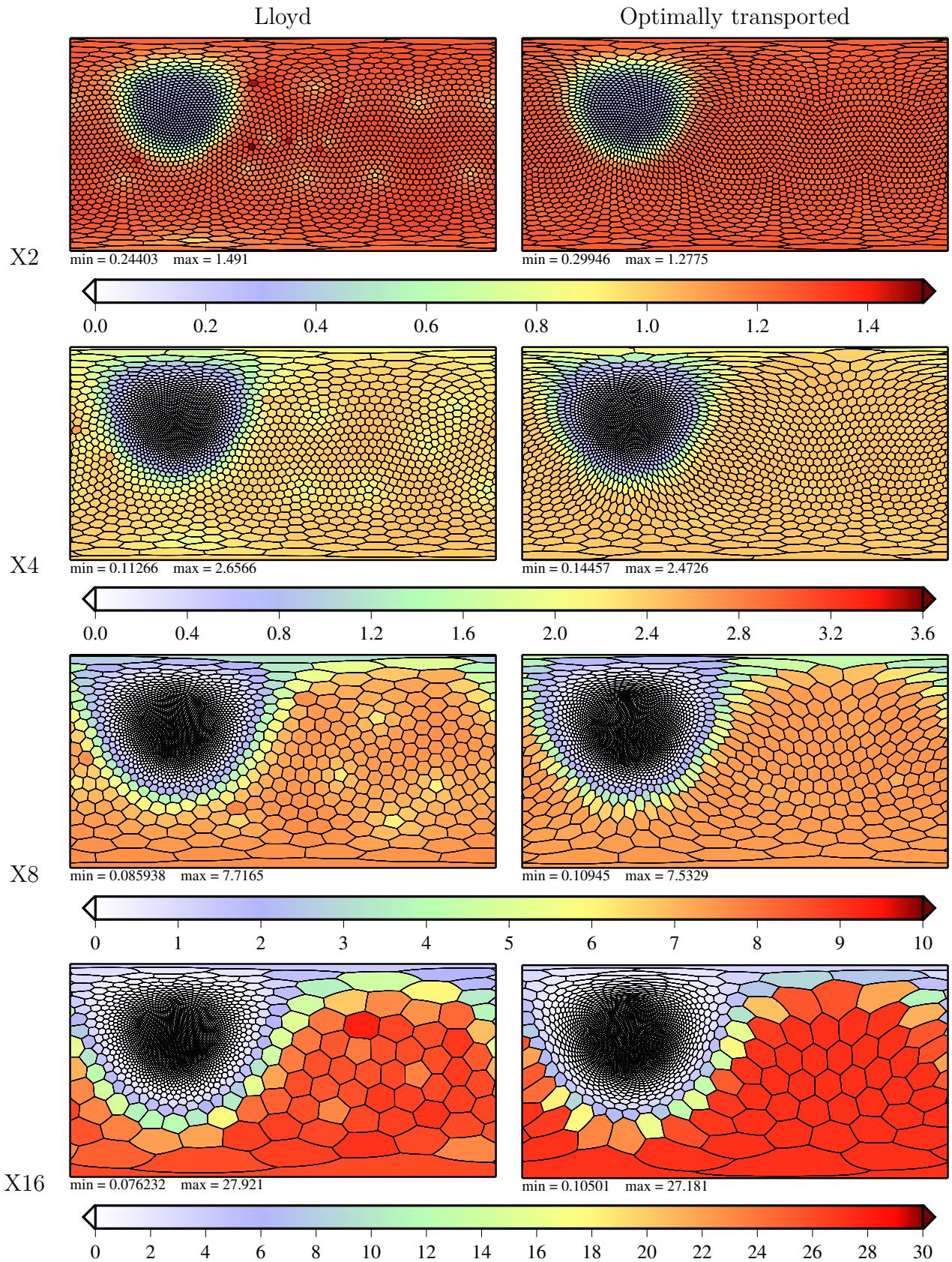


Figure 6: X2, X4, X8 and X16 meshes of the sphere of 2,562 cells and the cell volumes relative to the average cell volume in colour.

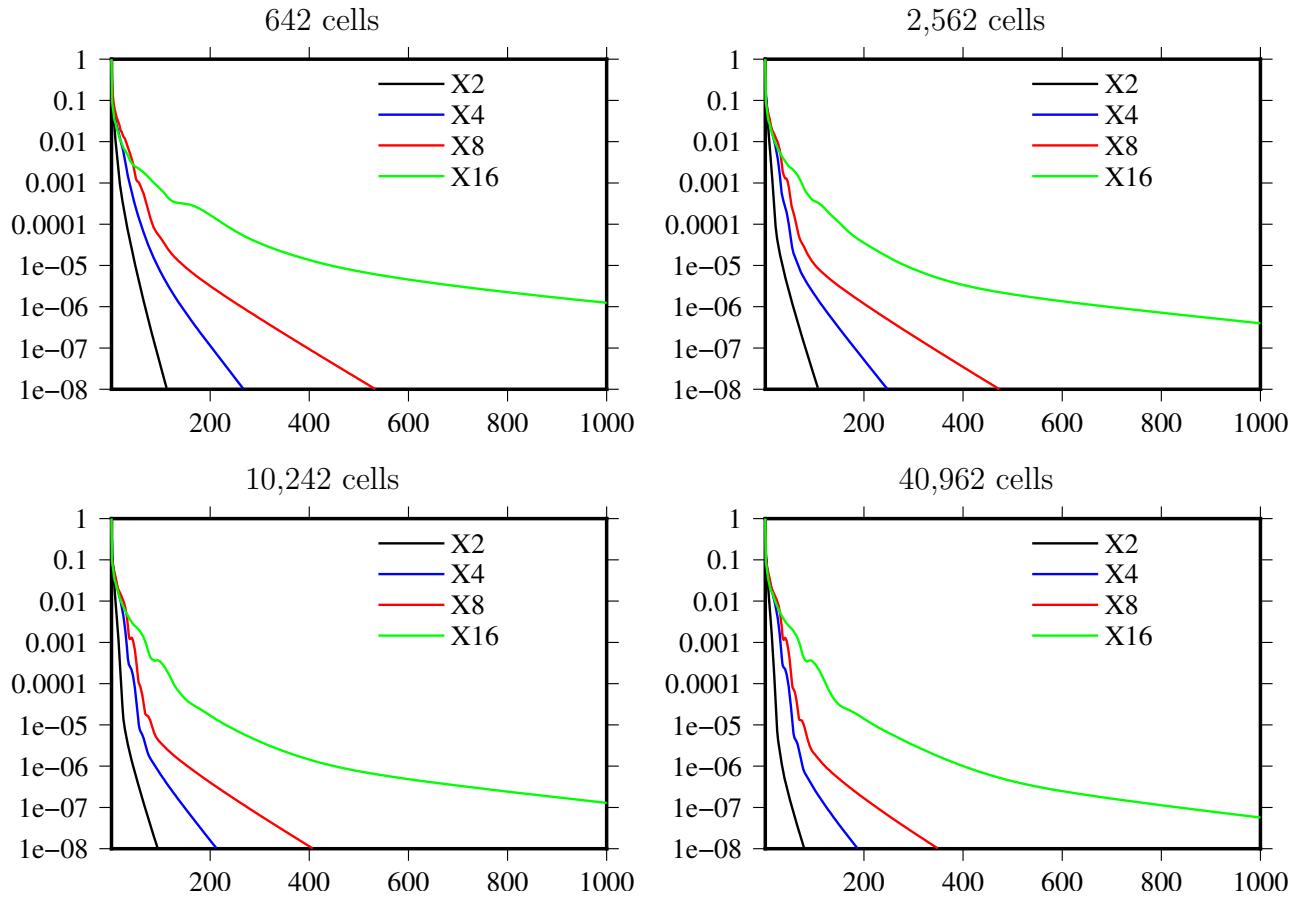


Figure 7: Convergence of initial residuals with iteration number for all of the optimally transported meshes on the sphere

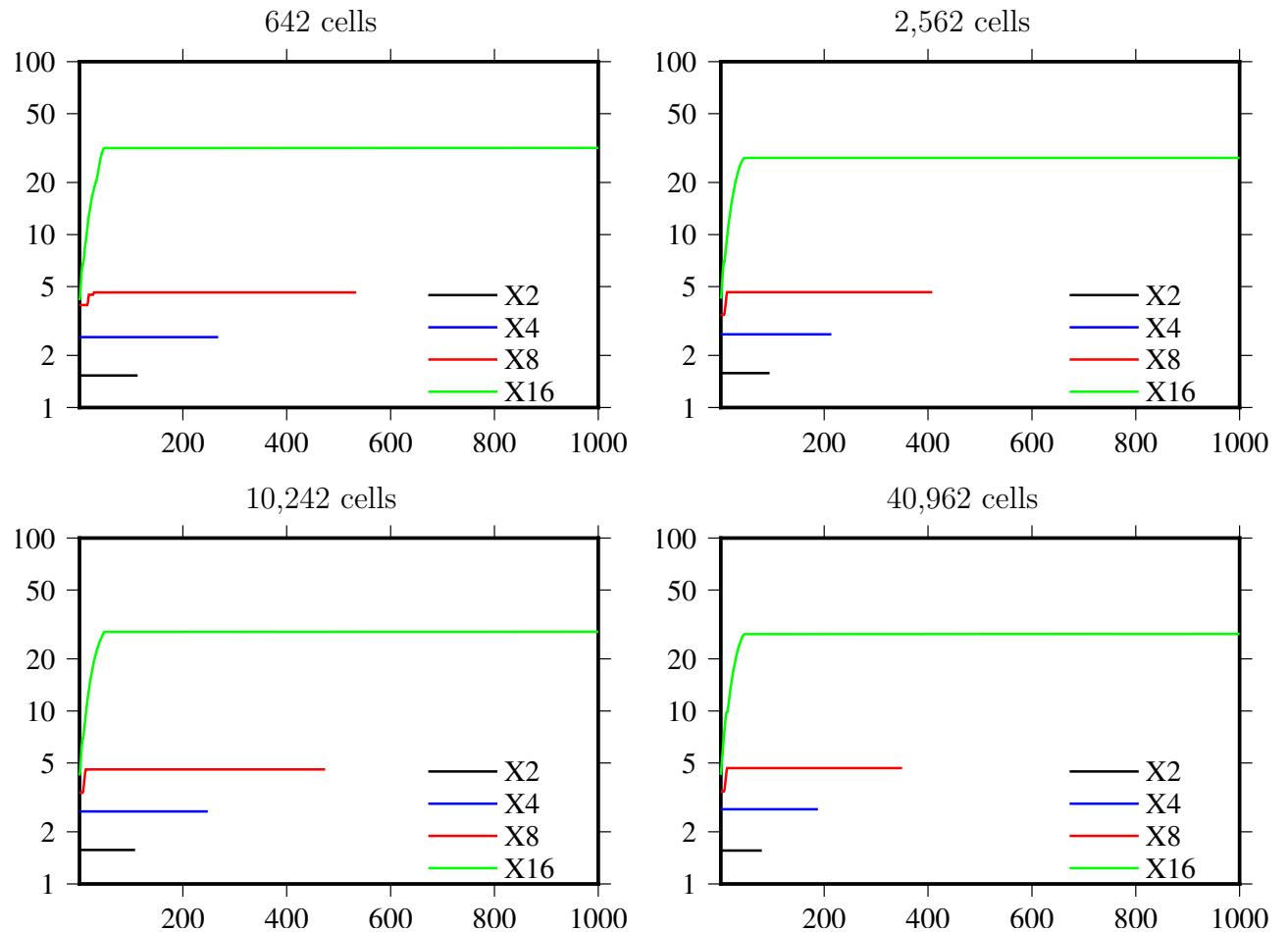


Figure 8: Under-relaxation factor, $1 + \alpha$, for with iteration number for all of the optimally transported meshes on the sphere

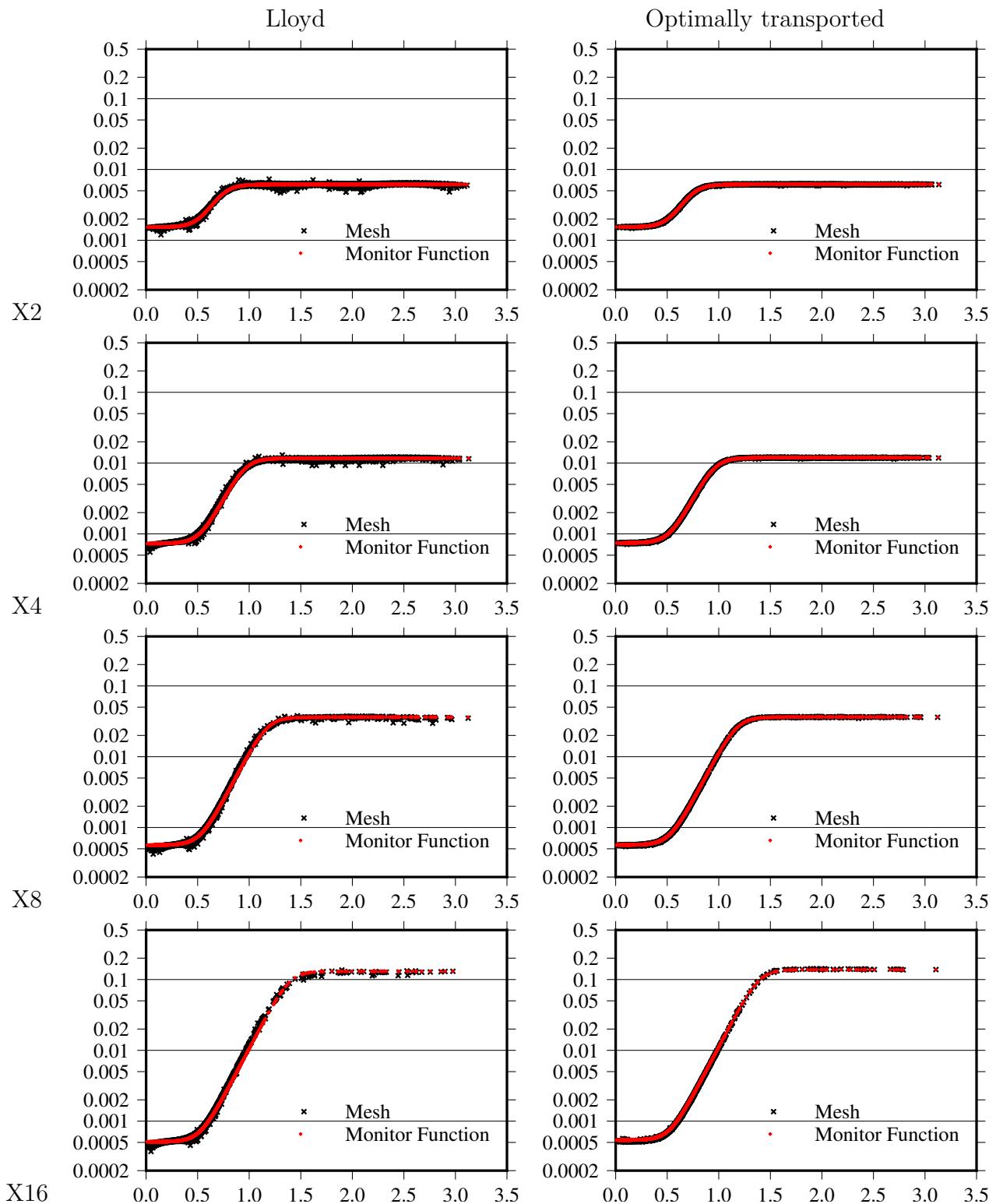


Figure 9: Cell area as a function of geodesic distance to the centre (in red) of the refined region in comparison to that implied by the monitor function for the X2, X4, X8 and X16 meshes of 2,562 cells

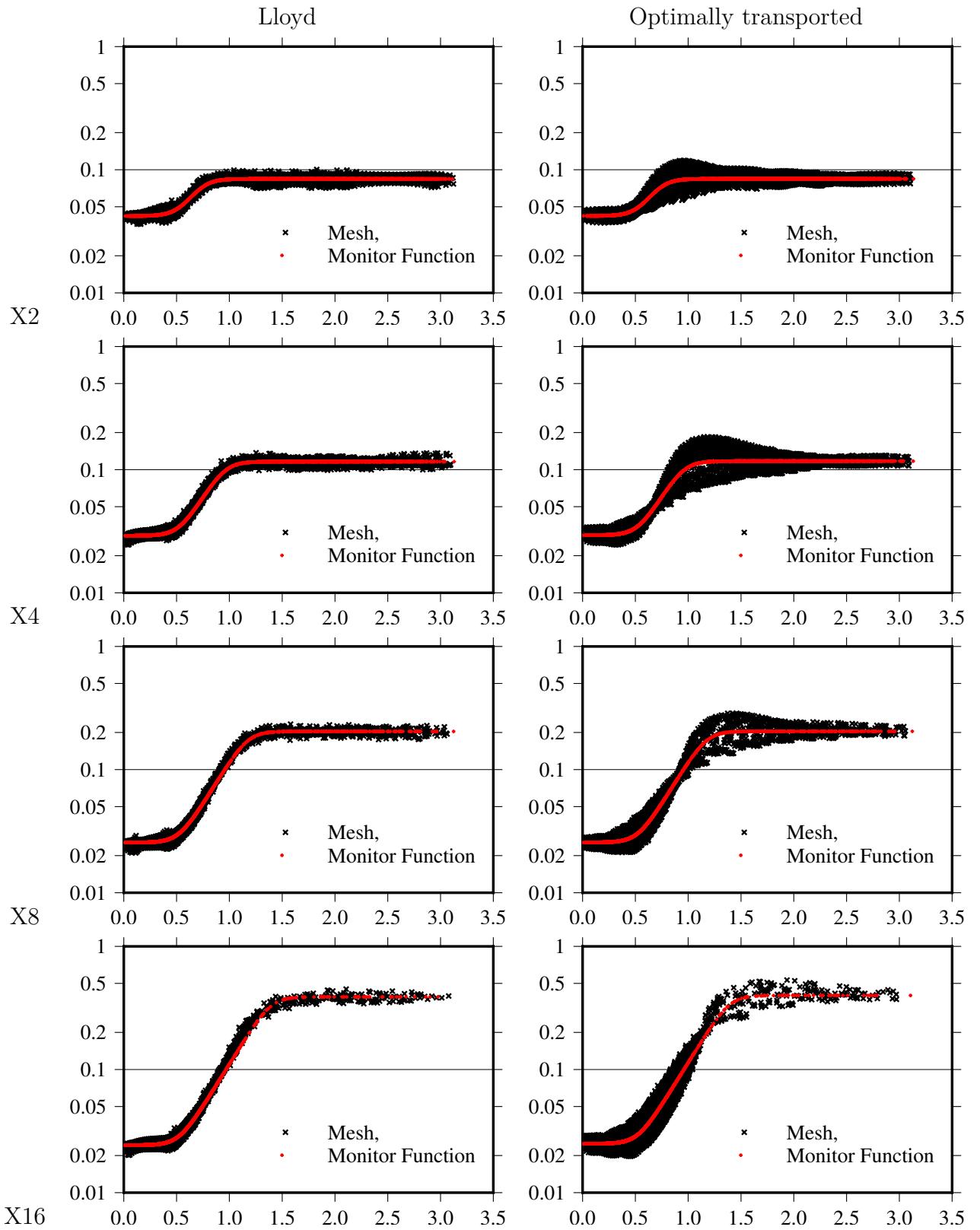


Figure 10: Cell-centre to cell-centre geodesic distance as a function of geodesic distance to the centre of the refined region in comparison to that implied by the monitor function for the X2, X4, X8 and X16 meshes of 2,562 cells

5.5. Mesh Quality on the Sphere

Scatter plots of the cell-centre to cell-centre distance, Δx , as a function of distance to the centre of the refined region are shown in fig. 10 for the X2-X16 meshes of 2,562 cells on the sphere. This shows that the Lloyd meshes are close to isotropic whereas the optimally transported meshes have high anisotropy where the second derivative of the monitor function is high. This could be reduced by smoothing the monitor function.

Unlike the meshes on the plane, the meshes on the sphere are isotropic in the uniformly coarse region, due to the isotropy of the domain relative to the centre of refinement. This could be an advantage of using r-adaptivity on the sphere over its use in Euclidean geometry with corners. However the meshes on the sphere still have a bulge in Δx on the edge of the coarse region. This is not ideal for atmospheric simulations since global errors are often proportional to the largest Δx [24].

The orthogonality and skewness of the faces of the optimally transported X2-X16 meshes on the sphere are shown in figs. 11 and 12 in comparison to the Lloyd meshes. Lloyd's algorithm with a non-uniform monitor function generates exactly orthogonal, non-centroidal meshes and so for comparison with the optimally transported meshes, the Lloyd's algorithm meshes are made centroidal at the expense of orthogonality by using the cell centroid as the cell centre rather than using the Voronoi generating point. Even so, they remain very close to orthogonal in comparison to the optimally transported meshes which have dramatically high non-orthogonality where the second derivative of the monitor function is high. In fact the non-orthogonality reaches 90 degrees for some cells implying that the cell-centre to cell-centre line is parallel to one of the edges of a face. This is clearly an unacceptable mesh for simulation. This problem will be investigated further.

5.6. Improving Mesh Quality

The optimally transported X16 meshes presented in sections 5.3-5.5 have unacceptably large non-orthogonality at regions with the resolution is changing rapidly (fig 11). The reason for this can be seen more clearly in a zoomed regions of the meshes in the second row of fig. 13. The double zoomed plot shows that some of the cells are not convex. This implies that the calculation of $\nabla_v \phi$ has in fact not yielded a smooth vector field, despite the development of the Goldilocks stencil with the aim of achieving a smooth $\nabla_v \phi$ on the smallest possible

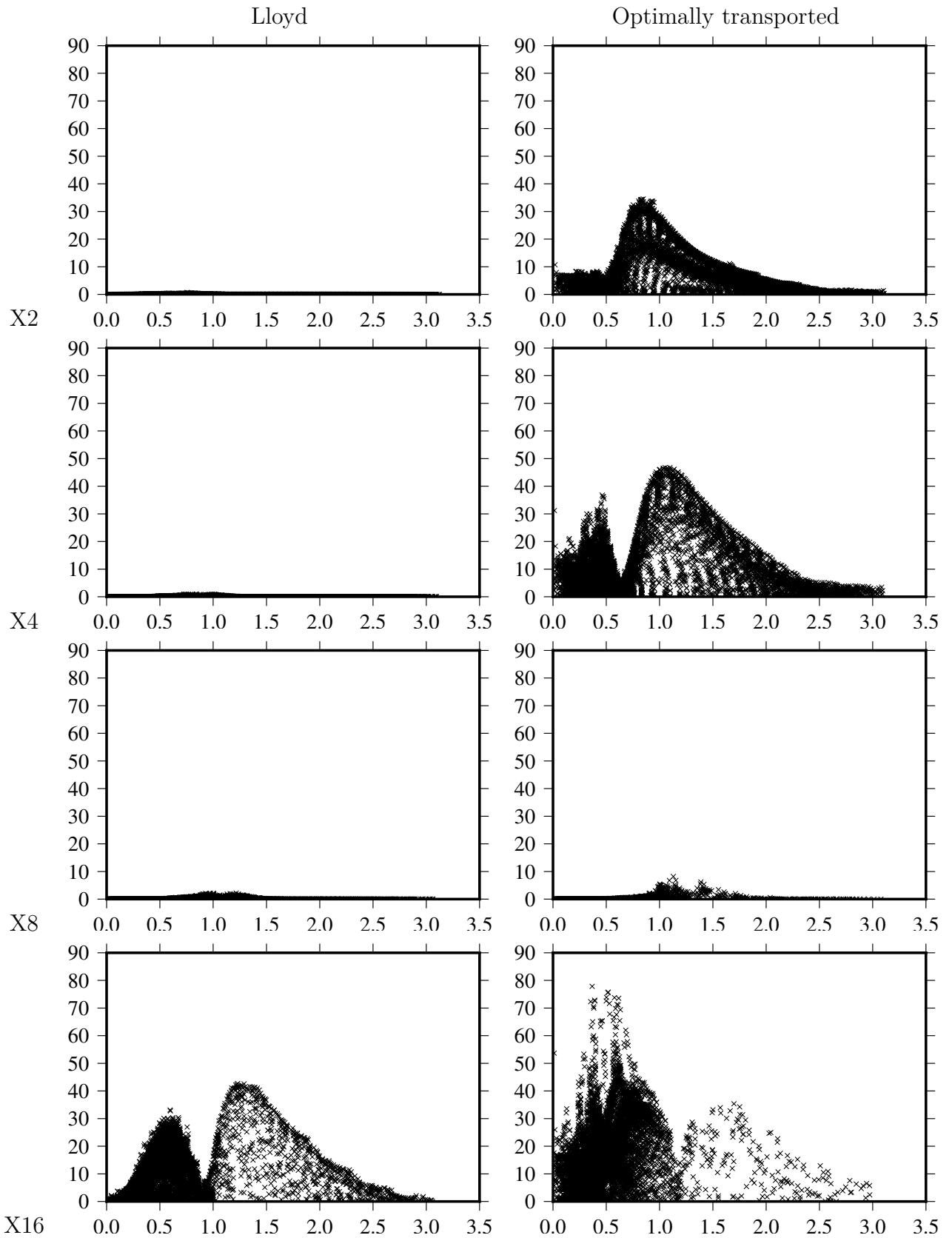


Figure 11: Orthogonality of the meshes on the sphere as a function of geodesic distance to the centre of the refined region for the X2-X16 meshes of 2,562 cells

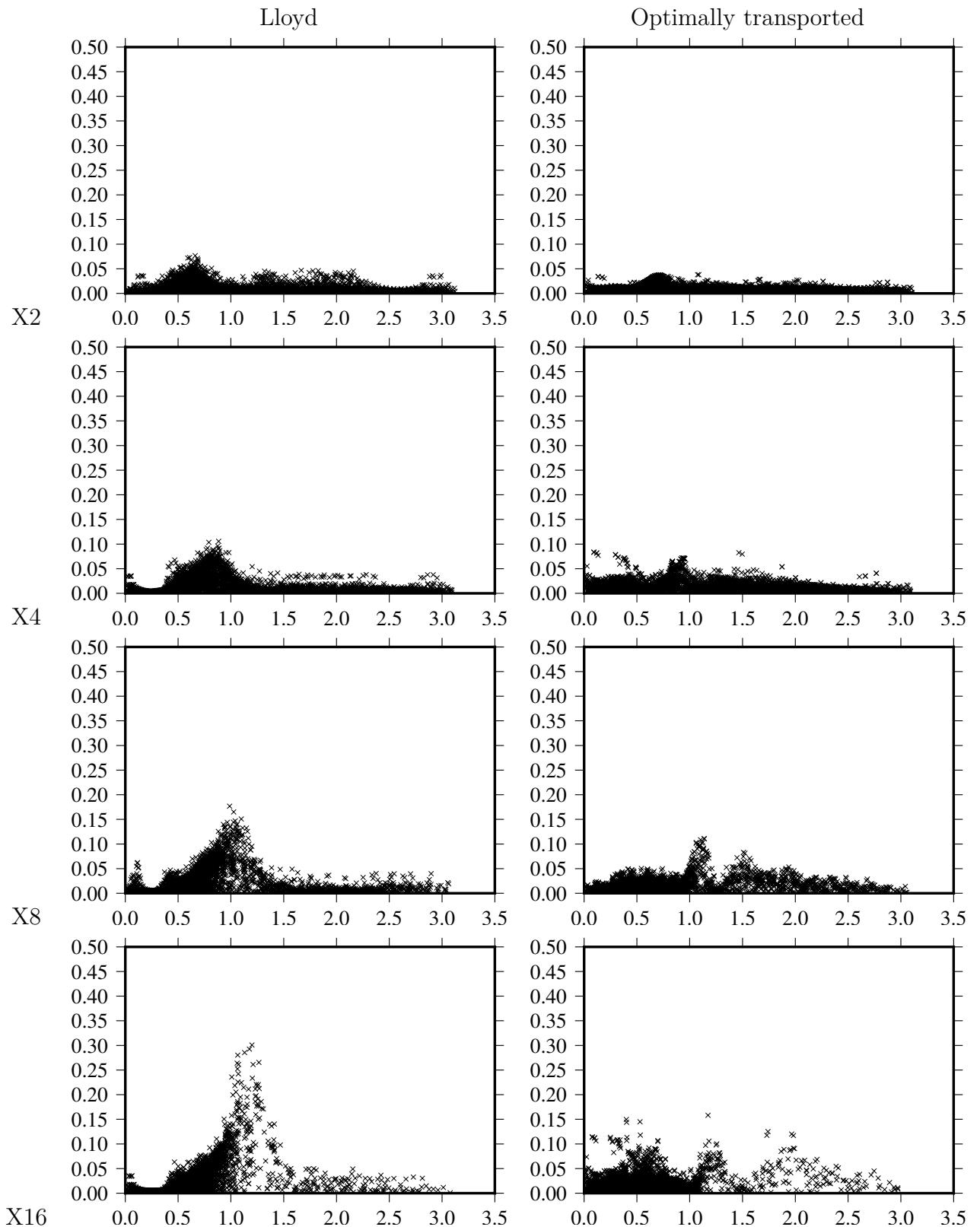


Figure 12: Mesh skewness as a function of geodesic distance to the centre of the refined region for the X2 and X4 meshes of 2,562 cells

stencil. The Goldilocks stencil does give a much smoother $\nabla_v\phi$ than the small stencil (first row of fig 13). If instead we interpolate $\nabla\phi$ from faces onto vertices which entails the use of a larger stencil (secn 3.4.3.1), the non-convex cells are not generated (third row of fig 13). Alternatively, a Voronoi tessellation can be created using the cell centres of the Goldilocks stencil mesh as generating points. This also eliminates non-convex cells and the mesh is shown at the bottom of fig. 13.

The problem with the large stencil calculation of $\nabla_v\phi$ is that convergence is slowed and orthogonality is not reduced enough (fig 14). Therefore it is necessary to consider the Voronoi tessellation of the cell centres (final row of fig. 13). This modification does not affect the convergence since the Voronoi tessellation is calculated after convergence of the Monge-Ampere solution. This mesh is insensitive to the calculation of $\nabla_v\phi$ but is no longer exactly equidistributed because the cell areas change a little (locally) when the Voronoi tessellation is calculated (fig 15). However these area changes are very small and simply smooth out the curve where the monitor function flattens out into the coarse region. Fig 15 also shows that the Voronoi version is more orthogonal than the large stencil version, the anisotropy is similar and the skewness is increased. However, the connectivity may be changed slightly.

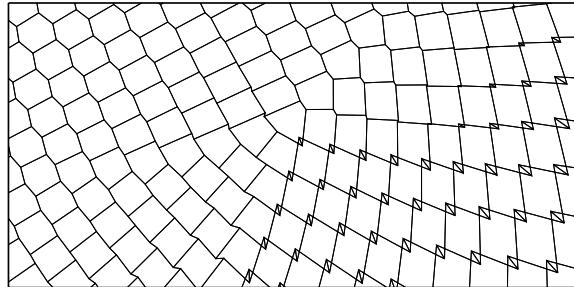
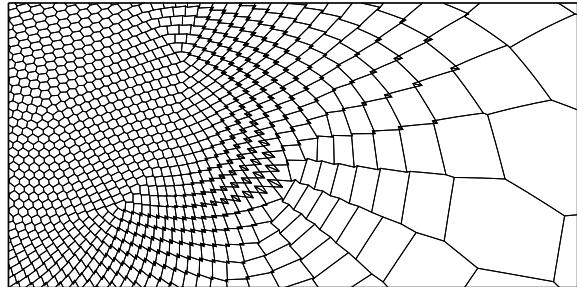
5.7. Optimally Transported Meshes based on a Precipitation Monitor Function

In order to demonstrate the numerical solution of the Monge-Ampère equation using realistic data as a monitor function, a mesh is generated based on the daily average precipitation rate from the NOAA-CIRES 20th Century Reanalysis version 2 (7, http://www.esrl.noaa.gov/psd/data/gridded/data.20thC_ReanV2.html) on 11 Oct 2012. The numerical solution of the Monge-Ampère equation uses two near uniform hexagonal-icosahedral meshes of 2,562 and 10,242 cells. The re-analysis precipitation ranges from zero to $p_{\max} = 8.73 \times 10^{-4} \text{kgm}^{-2}\text{s}^{-1}$. A strictly positive, non-dimensional monitor function, m , is defined from the precipitation rate, p using:

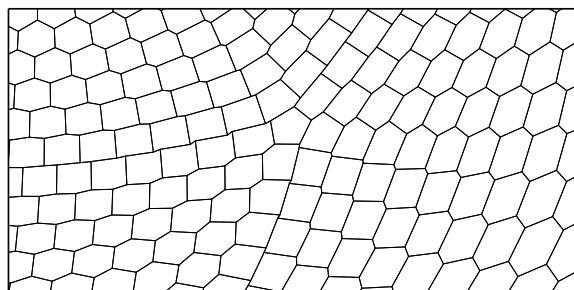
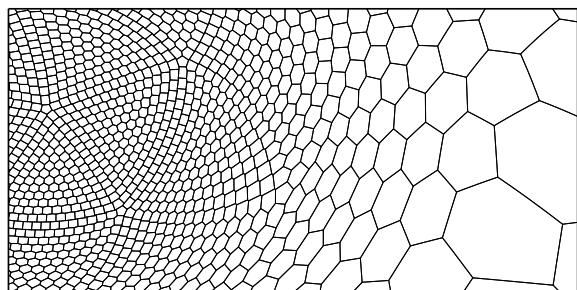
$$m = \frac{p + p_{\min}}{p_{\max} + p_{\min}}$$

where the minimum allowable values is set to $p_{\min} = 10^{-5} \text{kgm}^{-2}\text{s}^{-1}$. The resulting meshes are shown in fig. 16. Precipitation clearly could not be used as a monitor function for a dynamically adapting simulation of the global atmosphere since it is strongly resolution dependent. Instead, monitor functions with less resolution dependency should be developed.

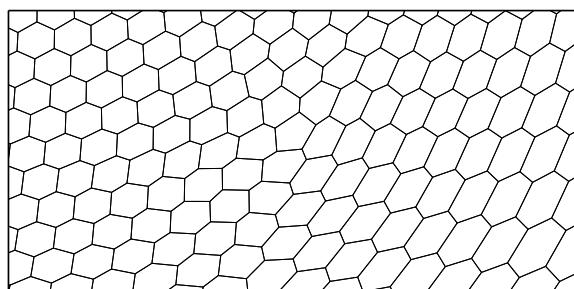
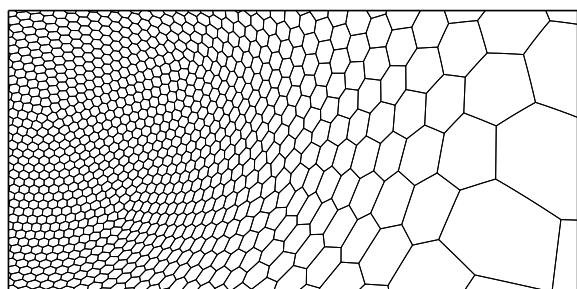
\downarrow Geometric Hessian with small stencil for $\nabla_v\phi$



\downarrow Geometric Hessian with Goldilocks stencil for $\nabla_v\phi$



\downarrow Geometric Hessian with large stencil for $\nabla_v\phi$



\downarrow Voronoi version of Geometric Hessian with Goldilocks stencil for $\nabla_v\phi$

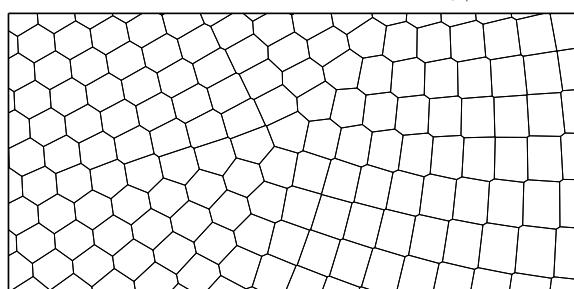
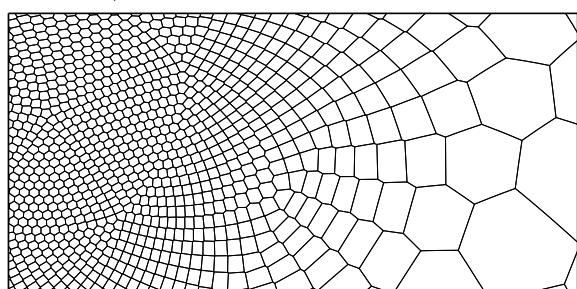


Figure 13: Zoomed and double zoomed X16 meshes with 2,562 cells using different calculations of $\nabla_v\phi$ and Voronoi versions.

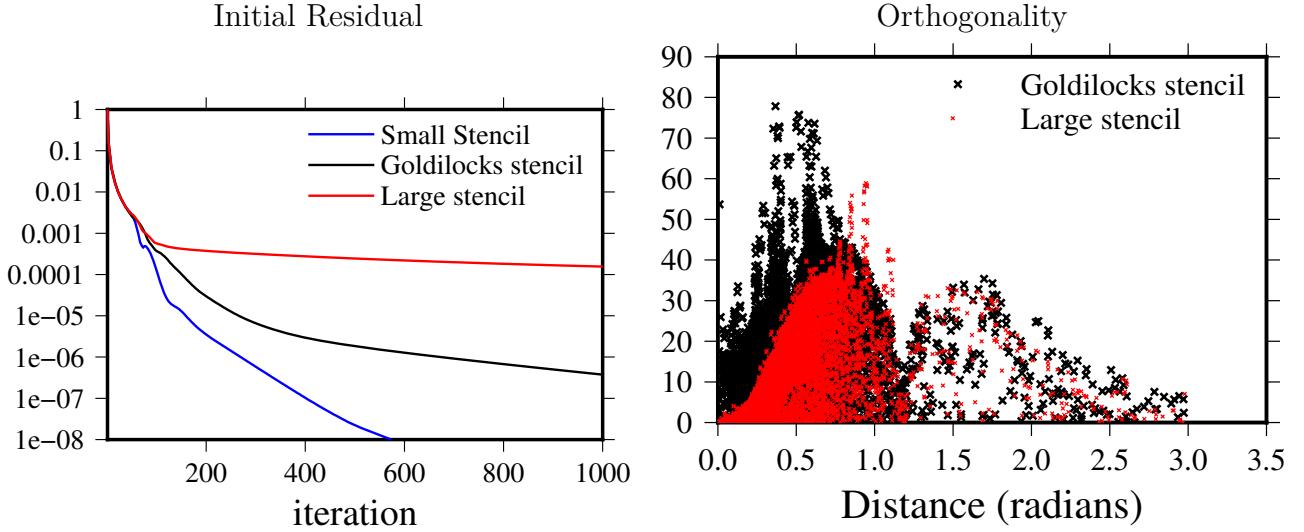


Figure 14: Comparisons of convergence of initial residuals and orthogonality for the X16 meshes of 2,562 cells calculated using the geometric Hessian but with different calculations of $\nabla_v \phi$.

The meshes resolved based on precipitation show excellent refinement along fronts, particularly looking at South America. The ITCZ is also refined in the latitudinal direction. However, based on the limitations of r-adaptivity, the ITCZ cannot be refined everywhere around the equator in the longitudinal direction. If this were a requirement, a mesh starting with more points around the equator should be used. This is the subject of future research.

6. Conclusions

A technique for generating optimally transported meshes on the surface of the sphere has been developed and some meshes are generated which have advantages and disadvantages relative to Voronoi meshes generated using Lloyd’s algorithm. The advantages are that optimally transported meshes can be generated in linear time and the connectivity never changes so these meshes can be used in r-adaptive simulations. The disadvantages are the increased non-orthogonality, skewness and anisotropy relative to Voronoi meshes. However the optimal transport technique can also be used to generate Voronoi meshes so some of these disadvantages can be ameliorated.

We have described, for the first time, a numerical method for solving the Monge Ampére equation on the surface of a sphere. Exponential maps are used to create the map from the old to the new mesh and we use a geometric interpretation of the Hessian rather than a numerical

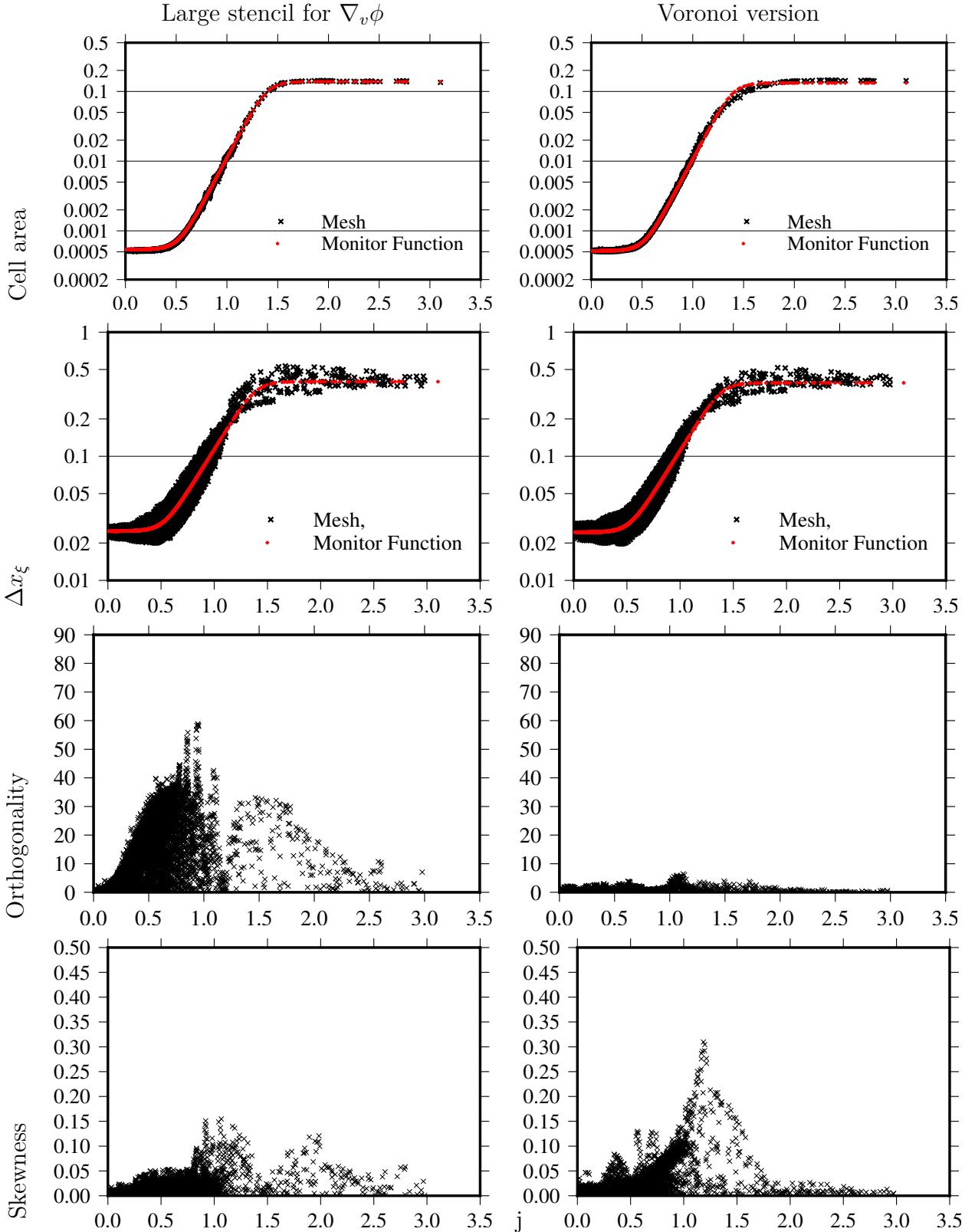


Figure 15: Mesh diagnostics as a function of distance from the centre of the refined region for the X16 meshes of 2,562 cells using the large stencil for $\nabla_p \phi$ on the left and using the Voronoi tesselation on the right.

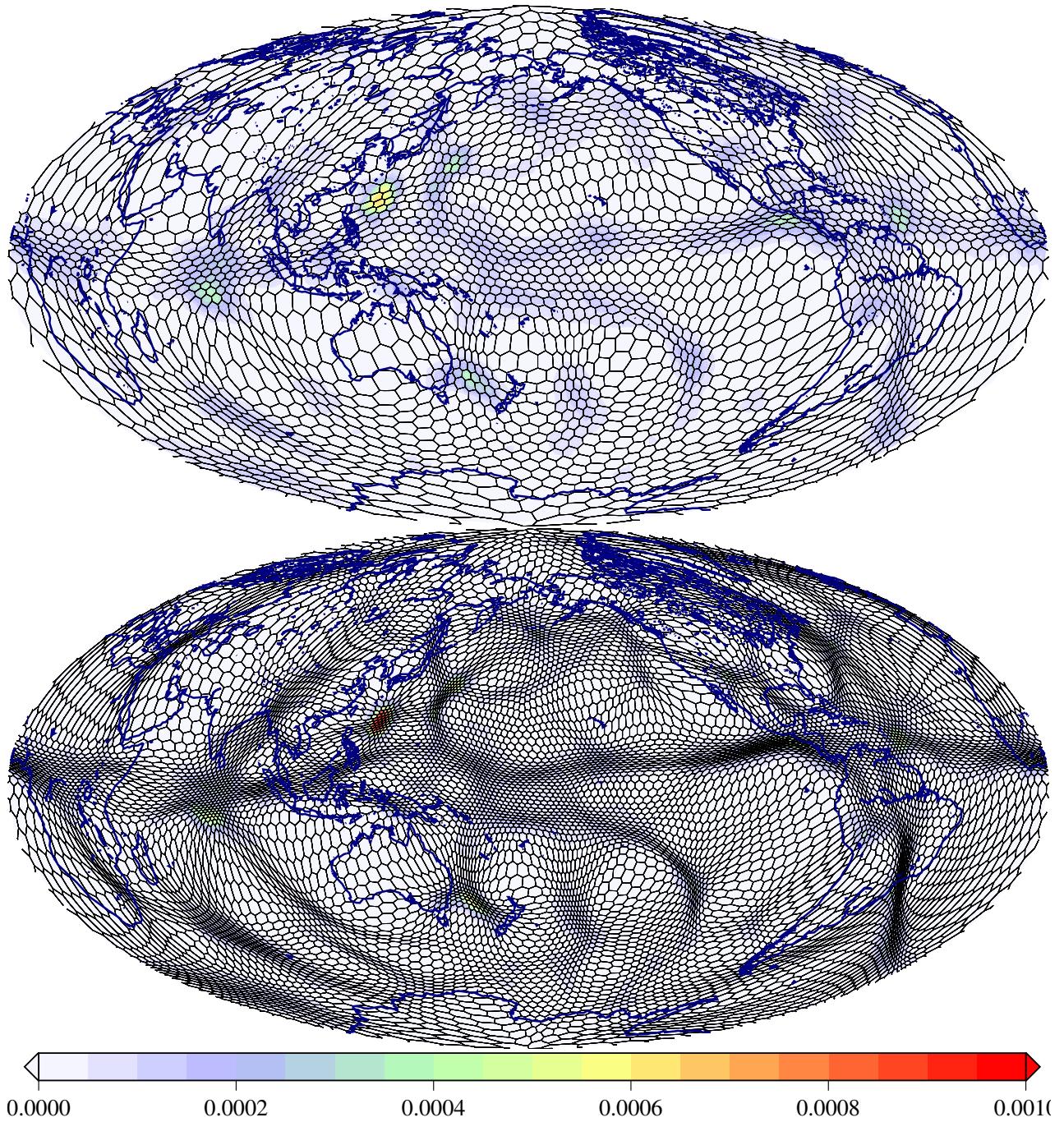


Figure 16: Meshes of 2,562 and 10,242 cells generated based on the monitor function of precipitation on 11 Oct 2012. The colours show the precipitation rate in $\text{kgm}^{-2}\text{s}^{-1}$.

approximation which is accurate on the surface of the sphere. In order to create a semi-implicit algorithm, a new linearisation of the Monge-Ampère equation is proposed which includes a Laplacian term and the resulting Poisson equation is solved at each fixed-point iteration.

To validate the novel aspects of the numerical method we first reproduce some known solutions of the Monge-Ampère equation on a two dimensional plane. We also generate meshes of polygons on the sphere to compare with the Voronoi meshes generated by Ringler et al. [24]. The use of the geometric version of the Hessian leads to meshes which conform well to the monitor function (they are equidistributed with respect to the monitor function). The algorithm is found to be sensitive to the numerical method used to calculate the gradient of the mesh potential (the map to the new mesh) with a compact stencil leading to mesh tangling and a large stencil leading to very slow convergence. The mesh tangling can be eliminated by creating a Voronoi tessellation of the cell centres of the final mesh.

Finally we generate a mesh using a monitor function based on reanalysis precipitation. This mesh refines smoothly along atmospheric fronts and convergence zones and provides inspiration for using r-adaptivity for global atmospheric modelling.

- [1] J.-D. Benamou, B. Froese, and A. Oberman. Two numerical methods for the elliptic Monge-Ampère equation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 44(4):737–758, 2010.
- [2] M. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, 53:484–512, 1984.
- [3] P. Browne, C. Budd, C. Piccolo, and M. Cullen. Fast three dimensional r-adaptive mesh redistribution. *J. Comput. Phys.*, 2014.
- [4] C. Budd, W. Huang, and R. Russell. Adaptivity with moving grids. *Acta Numerica*, 18:111–241, 2009.
- [5] C. Budd, M. Cullen, and E. Walsh. Monge-Ampère based moving mesh methods for numerical weather prediction, with applications to the eady problem. *J. Comput. Phys.*, 236(247-270), 2013.

- [6] C. Budd, R. Russell, and E. Walsh. The geometry of R-adaptive meshes generated using optimal transport methods. *J. Comput. Phys.*, 282:113–137, 2015.
- [7] G. Compo, J. Whitaker, P. Sardeshmukh, N. Matsui, R. Allan, X. Yin, B. Gleason, R. Vose, G. Rutledge, P. Bessemoulin, S. Bronnimann, M. Brunet, R. Crouthamel, A. Grant, P. Groisman, P. Jones, M. Kruk, A. Kruger, G. Marshall, M. Maugeri, H. Mok, O. Nordli, T. Ross, R. Trigo, X. Wang, S. Woodruff, and S. Worley. The twentieth century reanalysis project. *Quart. J. Roy. Meteor. Soc.*, 137:1–28, 2011.
- [8] J.-F. Cossette, P. Smolarkiewicz, and P. Charbonneau. The Monge-Ampére trajectory correction for semi-Lagrangian schemes. *J. Comput. Phys.*, 274:208–229, 2014.
- [9] C. Cotter and J. Shipton. Mixed finite elements for numerical weather prediction. *J. Comput. Phys.*, 231(21):7076–7091, Aug 2012.
- [10] E. Dean and R. Glowinski. Numerical methods for fully nonlinear elliptic equations of the Monge-Ampére type. *Computer methods in applied mechanics and engineering*, 195(13):1344–1386, 2006.
- [11] E. Dean and R. Glowinski. An augmented Lagrangian approach to the numerical solution of the Dirichlet problem for the elliptic Monge-Ampére equation in two dimensions. *Electronic Transactions on Numerical Analysis*, 22:71–96, 2006.
- [12] G. Dietachmayer and K. Droege. Application of continuous dynamic grid adaption techniques to meteorological modeling. Part I: Basic formulation and accuracy. *Mon. Wea. Rev.*, 120(8):1675–1706, 1992.
- [13] X. Feng and M. Neilan. Mixed finite element methods for the fully nonlinear Monge-Ampére equation based on the vanishing moment method. *SIAM Journal on Numerical Analysis*, 47(2):1226–1250, 2009.
- [14] B. Fiedler and R. J. Trapp. A fast dynamic grid adaption scheme for meteorological flows. *Mon. Wea. Rev.*, 121(10):2879–2888, 1993.

- [15] B. Froese and A. Oberman. Convergent finite difference solvers for viscosity solutions of the elliptic Monge-Ampère equation in dimensions two and higher. *SIAM Journal on Numerical Analysis*, 49(4):1692–1714, 2011.
- [16] F. Giraldo and T. Rosmond. A scalable spectral element Eulerian atmospheric model (SEE-AM) for NWP: Dynamical core tests. *Mon. Wea. Rev.*, 132(1):133–153, 2004.
- [17] R. Heikes and D. Randall. Numerical integration of the shallow-water equations on a twisted icosahedral grid. Part II: A detailed description of the grid and an analysis of numerical accuracy. *Mon. Wea. Rev.*, 123:1881–1997, June 1995.
- [18] D. Jacobsen, M. Gunzburger, T. Ringler, J. Burkardt, and J. Peterson. Parallel algorithms for planar and spherical Delaunay construction with an application to centroidal Voronoi tessellations. *Geosci. Model Dev.*, 6:1353–1365, 2013.
- [19] R. Li, T. Tang, and P. Zhang. Moving mesh methods in multiple dimensions based on harmonic maps. *J. Comput. Phys.*, 170:562–588, 2001.
- [20] T. Melvin, A. Staniforth, and C. Cotter. A two-dimensional mixed finite-element pair on rectangles. *Quart. J. Roy. Meteor. Soc.*, 140(680):930–942, 2014.
- [21] S. Mishra, M. Taylor, R. Nair, P. Lauritzen, H. Tufo, and J. Tribbia. Evaluation of the HOMME dynamical core in the aquaplanet configuration of NCAR CAM4: Rainfall. *J. Climate*, 24(15):4037–4055, 2011.
- [22] A. Oberman. Wide stencil finite difference schemes for the elliptic Monge-Ampère equation and functions of the eigenvalues of the Hessian. *Discrete Contin. Dyn. Syst. Ser. B*, 10(1):221–238, 2008.
- [23] OpenFOAM. The opensource CFD toolbox. [Available online at <http://www.openfoam.org>], cited 2014. The OpenCFD Foundation.
- [24] T. Ringler, D. Jacobsen, M. Gunzburger, L. Ju, D. M., and W. Skamarock. Exploring a multi-resolution modeling approach within the shallow-water equations. *Mon. Wea. Rev.*, 139(11):3348–3368, 2011.

- [25] W. Skamarock and J. Klemp. Adaptive grid refinement for two-dimensional and three-dimensional nonhydrostatic atmospheric flow. *Mon. Wea. Rev.*, 121(3):788–804, 1993.
- [26] W. Skamarock, J. Klemp, M. Duda, L. Fowler, S.-H. Park, and T. Ringler. A multi-scale nonhydrostatic atmospheric model using centroidal Voronoi tesselations and C-grid staggering. *Mon. Wea. Rev.*, 140(9):3090–3105, 2012.
- [27] J. Thuburn, T. Ringler, W. Skamarock, and J. Klemp. Numerical representation of geostrophic modes on arbitrarily structured C-grids. *J. Comput. Phys.*, 228(22):8321–8335, 2009.
- [28] Y. Wang. An explicit simulation of tropical cyclones with a triply nested movable mesh primitive equation model: TCM3. Part I: Model description and control experiment. *Mon. Wea. Rev.*, 129:1370–1394, 2001.
- [29] H. Weller. Non-orthogonal version of the arbitrary polygonal C-grid and a new diamond grid. *Geosci. Model Dev.*, 7:779–797, 2014.
- [30] H. Weller, H. Weller, and A. Fournier. Voronoi, Delaunay and block structured mesh refinement for solution of the shallow water equations on the sphere. *Mon. Wea. Rev.*, 137(12):4208–4224, 2009.