



Parallel Geometric Multigrid for Global Weather Prediction

S.D. Buckeridge and R. Scheichl

Bath Institute For Complex Systems

Preprint 10/09 (2009)

<http://www.bath.ac.uk/math-sci/BICS>

Parallel Geometric Multigrid for Global Weather Prediction

Sean Buckeridge* and Robert Scheichl

Dept. Mathematical Sciences, University of Bath, Claverton Down, Bath BA2 7AY, UK

S.D.Buckeridge@bath.ac.uk and R.Scheichl@bath.ac.uk

Abstract

The subject of this work is an optimal and scalable parallel geometric multigrid solver for elliptic problems on the sphere. The use of fast elliptic solvers is crucial to the forecasting and data assimilation tools used at the UK Met Office, and the preconditioned Krylov subspace solvers currently used do not perform well for large problem sizes due to the ill-conditioned nature of the problems. The optimality of multilevel techniques for elliptic problems therefore makes them a suitable choice for these applications. The Met Office uses spherical polar grids which, although structured, have the drawback of creating strong anisotropies near the poles where the grid lines converge. Moreover, a higher resolution of mesh points in the radial direction poses further anisotropies, and so modifications to the standard multigrid relaxation and coarsening procedures are necessary in order to retain optimal efficiency. Since the strength of anisotropy varies between the equator and the poles, we propose a non-uniform coarsening strategy, where the grid is coarsened only in regions that are sufficiently isotropic. This is combined with line relaxation in the radial direction. The success of non-uniform coarsening strategies has been demonstrated with Algebraic Multigrid (AMG) methods. Without the large setup costs required by these methods, however, we aim to surpass them with the geometric approach outlined above. We demonstrate the advantages of the method with experiments on model problems, both sequentially and in parallel. Our experiments show robustness and optimal efficiency of this method with constant V-cycle convergence factors of less than 0.1. It substantially outperforms the Krylov subspace methods with one-level preconditioners and the BoomerAMG implementation of AMG by factors of 10 and 5, respectively, on typical grid resolutions. The parallel implementation scales almost optimally on up to 256 processors, so that a global solve of the quasi-geostrophic ω -equation with a resolution of 10km at the equator and 3×10^9 unknowns takes about 60 seconds.

Keywords. Quasi-geostrophic ω -equation, anisotropy, spherical polar grid, geometric multigrid, conditional semi-coarsening, line relaxation.

1 Introduction

In this paper we propose an optimal and scalable parallel iterative solver for the following three dimensional elliptic problem (in spherical polar coordinates) that plays a key role in data assimilation for numerical weather forecasting, i.e.

$$-N^2(r) \nabla_r^2 \omega - f_0^2 \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial \omega}{\partial r} \right) = g, \quad (1)$$

*Supported by Great Western Research (GWR) and the UK MET Office under grant GWR-151.

where ∇_r^2 is the 2D-Laplacian in spherical coordinates at constant height r . It is known as the *quasi-geostrophic omega (QG- ω) equation* and describes the vertical motion in scales important for weather forecasting in the atmosphere. We are interested in solving it in the spherical shell representing the (entire) Earth’s atmosphere. The unknown function ω is the vertical motion. f_0 is the Coriolis parameter, which in the quasi-geostrophic regime is assumed constant. Note however that from an algorithmic point of view, changing this to a (more realistic) variable parameter $f(\phi)$ poses no additional difficulties. $N^2(r)$ is related to the frequency of vertical buoyancy oscillations, which depends on the temperature gradient and varies smoothly with r . The right hand side term g encompasses all the sources of quasi-geostrophic forcing for vertical motion, such as temperature gradients, quasi-geostrophic wind, quasi-geostrophic vorticity, and latent heat release. Details of the equation, including its derivation and the asymptotic regimes in which it is valid, can be found in [4, 9]. By *optimal* we mean that the time for solving the discretised problem is proportional to the (discrete) problem size. Similarly, we say that an algorithm has *optimal parallel scalability*, if the solution time remains constant when the problem size and the number of processors are increased proportionally.

Many of the standard meteorological codes, in particular at the UK MET Office, use spherical polar grids, which lead to strong grid anisotropies near the poles where the grid lines converge. Therefore, several alternative grids that avoid the “pole problem” such as Yin–Yang or icosahedral grids, are becoming increasingly popular in numerical weather prediction, as discussed in [3, 23]. Nevertheless, for all the negative things the spherical polar grids might entail, these grids are very structured. This greatly simplifies the discretisation of (1) and the coding, which is why they are still widely used. We will show in this article that from a solver point of view the bad reputation of spherical polar grids (e.g. in [23, §3.2b]) is unjustified, provided the solvers are suitably adapted. Before we expand a bit further on this let us note that the grid spacings in the radial direction are in general much smaller than in the horizontal ones, since the thickness of the atmosphere is two orders of magnitude smaller than the circumference of the Earth. This creates a further source of anisotropy. Additionally, the grid is usually strongly graded in the radial direction with smaller grid spacings near the surface of the Earth to obtain a better resolution in the regions of most interest.

A standard finite volume discretisation of (1) on this anisotropic mesh leads to a system of equations

$$A\omega = \mathbf{b}, \tag{2}$$

where A is a large, sparse, symmetric positive definite (SPD) matrix. The discretisation which we use is basically identical to that given in [2] for Poisson’s equation on a spherical polar grid. The matrix A contains a 7-point stencil for each node on the grid, with non-zero entries only for the node itself and for its immediate neighbours. Typical grid resolutions used in data assimilation at the Met Office are 216, 163 and 50 nodes in the latitudinal, longitudinal and radial directions, respectively. This leads to a large problem size of over a million degrees of freedom and a highly ill-conditioned system matrix A , making (2) very difficult to solve efficiently. The solver currently used at the MET Office, i.e. a Krylov subspace method preconditioned with simple r -line relaxation or ADI-type methods, is not optimal and restricts the grid resolutions that are currently feasible for global simulations, as highlighted in [23, §3.2b].¹

It is well known that it is necessary to resort to multilevel techniques to obtain optimality (of iterative methods for large elliptic problems). For isotropic problems with smoothly varying coefficients standard geometric multigrid with simple point-wise smoothing and uniform coarsening is the most efficient method. As outlined in [7, 22] a simple relaxation method (the smoother) eliminates the high

¹Note that the mean radius of the earth is about 6370km, and so the horizontal grid size in the latitudinal direction for 216 nodes is about 185km near the equator.

frequency components of the error of an initial approximation, which is then approximated on a coarse grid (coarse grid correction). For anisotropic problems, this standard approach does unfortunately not lead to an optimal method. However, if the anisotropies are aligned with the grid then simple modifications achieve optimality even to strong anisotropies. These modifications are line smoothing and/or semi-coarsening. Line smoothing involves collectively relaxing all unknowns on an entire grid line by solving a tridiagonal system corresponding to the unknowns on that line. Semi-coarsening uses a family of coarse grids that are only coarsened in the direction of the larger coefficient. In (2) there are two sources of anisotropy: one due to the large aspect ratio between the radial and horizontal grid spacings; the second due to the spherical polar grid. In this paper we propose a robust geometric multigrid method that is able to deal with both these problems by applying a simple non-uniform partial coarsening (inspired from algebraic multigrid methods) combined with an r -line smoother.

The robustness of the non-uniform coarsening strategy is first demonstrated on a two-dimensional model problem: Poisson’s equation on the unit sphere. The idea is extremely simple. The spherical polar grid introduces anisotropy near the poles but not near the equator, so the grid is semi-coarsened near the poles but fully coarsened near the equator. We compare the off-diagonal matrix entries in the latitudinal and longitudinal direction at each line of latitude, and the grid line is fully coarsened only if the coefficients in both directions are of similar magnitude. This will be true near the equator where we coarsen in both directions, but not near the poles where we only coarsen in the longitudinal direction, thus leading to coarse grids that are better and better adapted to the anisotropy.

To deal with the strong anisotropy in the radial direction in three dimensions we use r -line relaxation and do not coarsen at all in the radial direction. This is then combined with the nonuniform coarsening strategy in the longitudinal and latitudinal directions. Although this partial coarsening only leads to a coarsening factor of about 3 from one grid to the next (instead of 8 for uniform coarsening), it guarantees that the method is fully robust to the anisotropies induced by the geometry and the grid and leads to an optimal method with an average V-cycle convergence factor of less than 0.1, as our numerical tests show.

Geometric multigrid methods with line and plane smoothers, but with “uniform” semi-coarsening, have already been studied in [2]. Theoretical results for planar polar coordinates, line smoothers and (uniform) semi-coarsening can be found in [5] (see also [6]). PDEs of the type (1) from meteorological applications have already been solved with geometric multigrid methods, but only on cube-like domains with doubly-periodic boundary conditions and not on the entire globe (cf. [13, 1, 24, 25, 26, 19]). The most closely related paper is [26], where r -line relaxation and partial coarsening (i.e. uniform coarsening in the horizontal directions and no coarsening in the radial direction) was already studied extensively for the quasi-geostrophic equations. However, since the domain was not the entire atmosphere the additional complication of the anisotropy at the poles played no role. Multigrid algorithms have also already been proposed for alternative grids on the sphere, such as the icosahedral or the Yin–Yang grids, in [3, 17]

The idea of “conditional” semi-coarsening in the longitudinal direction proposed here has only been explored for edge and corner singularities so far (cf. [14, 18, 27]) but not for spherical polar grids (even in two dimensions). To the best of our knowledge, it seems to be a novel approach. It is clearly inspired by algebraic multigrid (AMG) ideas (see e.g. [7, 21]). AMG methods are fully automatic and only based on algebraic information in the matrix A . Coarse grid unknowns are chosen based on the relative size of the off-diagonal entries in the matrix which in the application here will lead to very similar coarse grids. However, AMG methods are known to require a large setup cost to design these coarse grids and the operator-dependent interpolation and restriction operators, especially in three dimensions. Our geometric method on the other hand, requires almost no setup cost to obtain the same robustness, which is why it easily outperforms established AMG methods. Numerical tests (cf. Section 4) for a variety of problem sizes confirm this. In that section, we also give a comparison to preconditioned Krylov solvers

as currently used by the MET Office [10] and their collaborators [8]. As expected, Krylov methods are only optimal when preconditioned with a robust multigrid method, such as AMG or the non-uniform geometric method proposed in this paper. With standard preconditioners used at the MET Office, such as r -line relaxation (on a single grid) or ADI-type preconditioners, the number of iterations grows with the problem size.

All (sequential) computations are carried out using the Fortran95 compiler `ifort` on a single processor of a Dual dual-core 64bit AMD Opteron 2210 processor with clock speed of 1.8GHz, cache size 1.0MB and 2GB memory. The initial guess for each iterative scheme is always taken to be zero. The stopping criterion is a relative residual reduction of 10^{-8} .

The rest of this paper is organised as follows. In Section 2, we describe the discretisation of (1) in more detail. In Section 3 we describe our nonuniform geometric multigrid method that is adapted to the particular anisotropies induced by the geometry and by the grid, and highlight some similarities with AMG. Numerical results for the sequential solver, as well as comparisons with AMG and preconditioned Krylov methods, will be given in Section 4. In the final section, we outline how we parallelised our method and demonstrate its optimal parallel scalability for up to 64 processors, as well as comparing the speedup to the speedup of parallel versions of the other solvers.

Acknowledgements. We would like to thank Mike Cullen (MET Office) for bringing this problem to our attention and for many helpful discussions.

2 Model Problem and Discretisation

Let Ω be a non-dimensionalised spherical shell representing the earth's atmosphere², i.e.

$$\Omega = \{ (x, y, z) : 0.99 \leq x^2 + y^2 + z^2 \leq 1 \}.$$

Re-parameterising into spherical polar coordinates gives

$$\hat{\Omega} = \{ (\theta, \phi, r) : 0.99 \leq r \leq 1, 0 \leq \phi \leq \pi, 0 \leq \theta \leq 2\pi \}, \quad (3)$$

where ϕ and θ are the polar and azimuthal angles, respectively. The parameterisation is given by

$$\mathcal{A}(\theta, \phi, r) = r \sin(\phi) \cos(\theta) \mathbf{i} + r \sin(\phi) \sin(\theta) \mathbf{j} + r \cos(\phi) \mathbf{k}$$

In order to discretise (1) on this domain, let us subdivide $\hat{\Omega}$ into $n_\theta \times n_\phi \times n_r$ cubes with cell centres

$$\{ (\theta_i, \phi_j, r_k) : i = 1, \dots, n_\theta, j = 1, \dots, n_\phi, k = 1, \dots, n_r \},$$

and edge lengths $h_\theta = 2\pi/n_\theta$, $h_\phi = \pi/(n_\phi + 1)$ and $h_{r,k}$, $k = 1, \dots, n_r$, as well as into $2 \times n_r$ cells at the poles with edge lengths 2π , $\pi/(2n_\phi + 2)$ and $h_{r,k}$. The computational grid in the $\theta - \phi$ plane can be seen in Figure 1(a), where the top and bottom boundary represent the North and South pole, respectively. As we can see, the grid on the $\theta - \phi$ plane is uniform (except at the poles). The nodes are located at the cell centres, where $\theta_i = (i - \frac{1}{2})h_\theta$ and $\phi_j = jh_\phi$. At the poles we use half cells, so that the poles themselves are located at the centres of the cells in the physical domain Ω , and so that a discrete equation can be derived at these points in the same fashion as at the other points. In the radial direction, the mesh is graded as shown in Figure 1(b) with the cell centres located at $r_k = \sum_{i=1}^{k-1} h_{r,i} + \frac{1}{2}h_{r,k}$, and with the mesh widths $h_{r,k}$ increasing with k . Thus, the total number of unknowns (including the unknowns at the poles) is $(n_\theta \times n_\phi + 2) \times n_r$.

²The mean radius of the Earth is about 6,370km and the height of the atmosphere of interest for meteorology is about 63km.

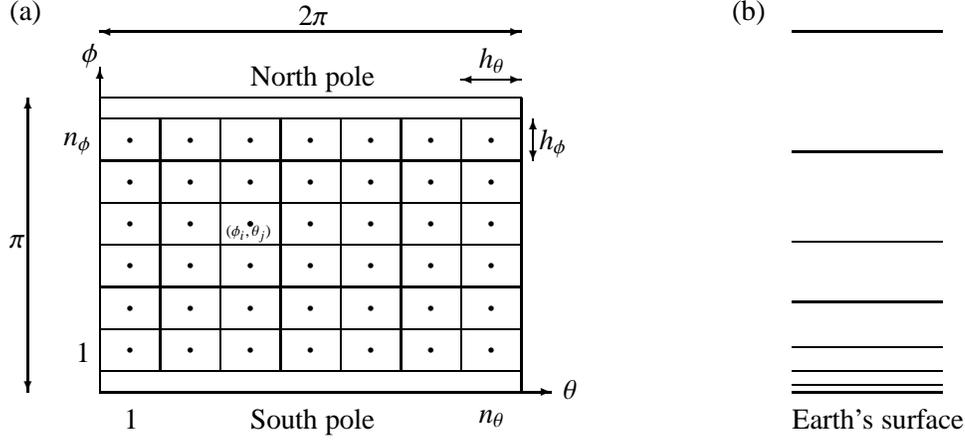


Figure 1: (a) The computational grid in the θ - ϕ plane, and (b) the graded mesh in the r -direction.

We discretise (1) using the finite volume method, which is done by integrating the PDE over each mesh cell (or control volume) corresponding to a grid point (θ_i, ϕ_j, r_k) , i.e.

$$\hat{\Omega}_{i,j,k} = \left[\theta_{i-\frac{1}{2}}, \theta_{i+\frac{1}{2}} \right] \times \left[\phi_{j-\frac{1}{2}}, \phi_{j+\frac{1}{2}} \right] \times \left[r_{k-\frac{1}{2}}, r_{k+\frac{1}{2}} \right],$$

where $\theta_{i\pm\frac{1}{2}} = \theta_i \pm \frac{h_\theta}{2}$, $\phi_{j\pm\frac{1}{2}} = \phi_j \pm \frac{h_\phi}{2}$ and $r_{k\pm\frac{1}{2}} = r_k \pm \frac{h_{r,k}}{2}$.

Except at the poles, the boundary of each control volume consists of six faces, i.e.

$$\partial\hat{\Omega}_{i,j,k} = \Gamma_{i-\frac{1}{2},j,k} \cup \Gamma_{i+\frac{1}{2},j,k} \cup \Gamma_{i,j-\frac{1}{2},k} \cup \Gamma_{i,j+\frac{1}{2},k} \cup \Gamma_{i,j,k-\frac{1}{2}} \cup \Gamma_{i,j,k+\frac{1}{2}}$$

and the cell faces are denoted by $\Gamma_{i\pm\frac{1}{2},j,k} = \{\theta_{i\pm\frac{1}{2}}\} \times [\phi_{j-\frac{1}{2}}, \phi_{j+\frac{1}{2}}] \times [r_{k-\frac{1}{2}}, r_{k+\frac{1}{2}}]$, with analogous definitions for $\Gamma_{i,j\pm\frac{1}{2},k}$ and $\Gamma_{i,j,k\pm\frac{1}{2}}$ and suitable modifications on the boundaries.

Since the problem is discretised on a sphere, it is necessary to impose periodic boundary conditions on the lateral boundary, i.e.

$$\begin{aligned} \omega(0, \phi, r) &= \omega(2\pi, \phi, r) \\ \frac{\partial\omega}{\partial\theta}(0, \phi, r) &= \frac{\partial\omega}{\partial\theta}(2\pi, \phi, r) \quad \forall \phi \in [0, \pi], \forall r \in [0.99, 1]. \end{aligned} \quad (4)$$

In addition, we impose for expositional purposes homogeneous Dirichlet boundary conditions on the upper and lower boundaries of the atmosphere, i.e.

$$\omega(\theta, \phi, 0.99) = \omega(\theta, \phi, 1) = 0, \quad (5)$$

but this is not essential. The finite volume discretisation is now obtained by integrating (1) over each control volume $\hat{\Omega}_{i,j,k}$, i.e.

$$\int_{\hat{\Omega}_{i,j,k}} \left\{ -N^2(r) \nabla^2 \omega - f_0^2 \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial \omega}{\partial r} \right) \right\} dV = \int_{\hat{\Omega}_{i,j,k}} g dV. \quad (6)$$

Where dV is the volume element in spherical coordinates, i.e.

$$dV = \left| \left(\frac{\partial \mathcal{A}}{\partial r} \times \frac{\partial \mathcal{A}}{\partial \phi} \right) \cdot \frac{\partial \mathcal{A}}{\partial \theta} \right| d\theta d\phi dr = r^2 \sin(\phi) d\theta d\phi dr.$$

Substituting dV into (6), applying the Divergence Theorem, and using central differences to approximate derivatives in the integrals over the cell faces, we obtain the following 7-point stencil³ for the interior nodes of $\hat{\Omega}$:

$$-f_0^2 \frac{h_\theta h_\phi}{h_{r,k}} r_{k-\frac{1}{2}}^2 \sin(\phi_j) \left[\begin{array}{ccc} & -N^2(r_k) \frac{h_\theta h_{r,k}}{h_\phi} \sin(\phi_{j+\frac{1}{2}}) & \\ -N^2(r_k) \frac{h_\theta h_{r,k}}{h_\theta} \sin(\phi_j)^{-1} & -\sum & -N^2(r_k) \frac{h_\theta h_{r,k}}{h_\theta} \sin(\phi_j)^{-1} \\ & -N^2(r_k) \frac{h_\theta h_{r,k}}{h_\phi} \sin(\phi_{j-\frac{1}{2}}) & \end{array} \right] - f_0^2 \frac{h_\theta h_\phi}{h_{r,k}} r_{k+\frac{1}{2}}^2 \sin(\phi_j) \quad (7)$$

Similar stencils are obtained at the lateral and vertical boundaries. At the poles we can proceed in a similar fashion, but each pole cell has $n_\theta + 2$ neighbours (with the obvious changes at the top and bottom of the atmosphere). After integrating over the pole cap and using the Divergence Theorem and central differences as before, we find that the n_θ off-diagonal entries in the θ -direction are $-N^2(r_k) \frac{h_r h_\theta}{h_\phi} \sin(h_\phi/2)$, whereas the entries (e.g. at the south pole) in the radial direction are $-f_0^2 \frac{\pi h_\phi}{2h_{r,k}} \sin(h_\phi/2) r_{k\pm\frac{1}{2}}^2$. This results in a system of linear equations of the form

$$A\omega = \mathbf{b},$$

where $A \in \mathbb{R}^{n \times n}$, and $n = (n_\theta \times n_\phi + 2) \times n_r$ is the dimension of the problem. $\omega \in \mathbb{R}^n$ is the unknown solution vector corresponding to the values of the unknown function ω at the cell centres, and $\mathbf{b} \in \mathbb{R}^n$ is the right-hand side containing the source terms.

3 Non-Uniform Geometric Multigrid for Spherical Polar Grids

In this section we describe a novel geometric multigrid method for solving (2). Standard geometric multigrid methods for simple isotropic problems use full coarsening (i.e. coarsening in all directions) and point relaxation smoothers (see [7] for details), and the optimal convergence of this method has been proven both experimentally and theoretically (cf. [15]). There are several variants of the method, but here we focus on the V-cycle, which can be described as follows.

Iterate the following routine with $A_F = A$, $\mathbf{b}_F = \mathbf{b}$ and initial guess $\mathbf{u}_F = \mathbf{0}$, until a certain stopping criterion is satisfied:

```

subroutine VCycle( $A_\ell, \mathbf{b}_\ell, \mathbf{u}_\ell$ )
  if ( $\ell = 1$ ) then
     $\mathbf{u}_1 = A_1^{-1} \mathbf{b}_1$  (solve on coarsest grid)
  else
     $\mathbf{u}_\ell = \mathcal{S}_\ell^{v_1}(\mathbf{u}_\ell, \mathbf{b}_\ell)$  ( $v_1$  pre-smoothing steps)
     $\mathbf{r}_{\ell-1} = R_\ell(\mathbf{b}_\ell - A_\ell \mathbf{u}_\ell)$  (calculate residual and restrict onto next coarser grid)
     $\mathbf{e}_{\ell-1} = \mathbf{0}$ 
    VCycle( $A_{\ell-1}, \mathbf{r}_{\ell-1}, \mathbf{e}_{\ell-1}$ ) (recursively apply VCycle for coarse grid correction)
     $\mathbf{u}_\ell = \mathbf{u}_\ell + P_\ell \mathbf{e}_{\ell-1}$  (interpolate error and update solution)
     $\mathbf{u}_\ell = \mathcal{S}_\ell^{v_2}(\mathbf{u}_\ell, \mathbf{b}_\ell)$  ( $v_2$  post-smoothing steps)
  end if

```

³Note that we use a similar notation as in [2] to present the 7-point stencil. The numbers in square brackets give the 5-point stencil in the $\theta - \phi$ plane in the usual way. The numbers outside the brackets denote the entries corresponding to the upwards and downwards neighbours, respectively. \sum denotes the sum of the off-diagonal entries corresponding to the six neighbours.

This routine requires a sequence of matrices A_ℓ , $\ell = 1, \dots, F$, corresponding to the PDE (1) discretised on a sequence of grids, where usually the grid on level ℓ is a uniform refinement of the grid on level $\ell - 1$. It requires a smoother S_ℓ on each grid (which is commonly a simple relaxation scheme like Gauss–Seidel), as well as prolongation and restriction matrices P_ℓ and R_ℓ , e.g. linear interpolation and full-weighting restriction. The number of pre- and post-smoothing steps is denoted by ν_1 and ν_2 . An alternative approach to define the matrices A_ℓ on the coarser grids is via the Galerkin approach, i.e. $A_{\ell-1} = R_\ell A_\ell P_\ell$, but this is more costly to set up and creates denser matrices on the coarser levels.

This standard method, however, is not robust for problems with anisotropy. Problem (1) discretised on the grid described in Section 2 contains two sources of anisotropy as outlined in Section 1, thus alternative ingredients are needed to solve it optimally. If the anisotropy has the convenient feature of being grid-aligned, i.e. aligned with the coordinate directions, two standard ways to retain optimality of multigrid are to use semi-coarsening and/or line relaxation. Line relaxation involves collectively relaxing all unknowns on an entire grid line by solving a tridiagonal system corresponding to the unknowns on that line. Semi-coarsening uses a family of coarse grids that are only coarsened in the direction of the larger coefficient, thus reducing the strength of the anisotropy on the coarser grids.

Conditional semi-coarsening in the $\theta - \phi$ plane

In two dimensions, problems with grid aligned anisotropy can be written in the form

$$-\nabla \cdot \left(\begin{pmatrix} \alpha_1(x_1, x_2) & 0 \\ 0 & \alpha_2(x_1, x_2) \end{pmatrix} \nabla u \right) = g, \quad (8)$$

with α_1, α_2 uniformly positive almost everywhere. The simplest model problem with grid aligned anisotropy is $\alpha_1 \equiv \epsilon \ll 1$ constant and $\alpha_2 \equiv 1$. For this model problem, it is shown theoretically in [20] that x -line relaxation with full coarsening leads to an optimal multigrid convergence. In the more general case of varying coefficients α_1, α_2 it seems that is necessary to combine line relaxation with semi-coarsening to still get an optimal method at least theoretical (cf. [5]).

If we ignore for a moment the r -dependency in our problem (1) and restrict to the two dimensional Poisson equation in spherical polar coordinates in each r -layer of the domain, we see that our problem is exactly of the type (8), i.e.

$$-\frac{\partial}{\partial \phi} \left(\sin(\phi) \frac{\partial u}{\partial \phi} \right) - \frac{\partial}{\partial \theta} \left(\frac{1}{\sin(\phi)} \frac{\partial u}{\partial \theta} \right) = g_{2D} \sin(\phi). \quad (9)$$

We will use this problem now to motivate the key idea of this paper and to show that in practice line relaxation is not necessary for problems of the type (9) provided a conditional semi-coarsening strategy is used. A theoretical proof of this is still missing. Note however, that even in the case of problem (8) with constant α_1, α_2 it has not yet been possible to obtain such a proof.

The finite volume discretisation of problem (9) on the grid introduced in Section 2 results in a singular system of linear equations. We require a compatibility condition on g_{2D} and we need to regularise the problem, e.g. by projecting the right-hand side vector onto the range of the operator or by fixing the solution at one of the poles. Now assuming that we use a quasi-uniform grid such that $h_\theta \approx h_\phi$, then the stencil at the interior nodes is:

$$\left[\begin{array}{ccc} & -\frac{h_\theta}{h_\phi} \sin(\phi_{j+\frac{1}{2}}) & \\ -\frac{h_\phi}{h_\theta} \frac{1}{\sin(\phi_j)} & -\Sigma & -\frac{h_\phi}{h_\theta} \frac{1}{\sin(\phi_j)} \\ & -\frac{h_\theta}{h_\phi} \sin(\phi_{j-\frac{1}{2}}) & \end{array} \right]. \quad (10)$$

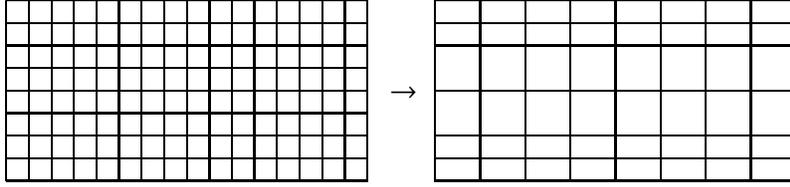


Figure 2: Conditional semi-coarsening on a 16×8 grid

Since $\phi_j \in [0, \pi]$, we observe a strong anisotropy near the poles caused by the spherical polar grid, where $\frac{1}{\sin \phi_j} \rightarrow \infty$ and $\sin \phi_j \rightarrow 0$. Thus the entries in the θ -direction are significantly larger than the entries in the ϕ -direction at the poles. Near the equator, on the other hand, the problem is close to isotropic. So while semi-coarsening would be effective near the poles, it would not work near the equator. This motivates the key idea which we propose, i.e. to introduce conditional semi-coarsening, where full uniform coarsening is performed near the equator and semi-coarsening (in θ -direction only) near the poles. More specifically, we compare the ratio of the ϕ and θ off-diagonal entries at each line of latitude. We fully coarsen that line only if the ratio is sufficiently close to 1. We observe from (10) that on a uniform mesh with $h_\theta \approx h_\phi$ the ratio is about $\sin^2(\phi_j)$. On subsequent grids this gets compensated by the factor $(h_\theta/h_\phi)^2$. In the actual computations, since $0 \leq \sin^2(\phi_j) \leq 1$, we fully coarsen only if $(h_\theta/h_\phi)^2 \sin^2(\phi_j)$ is greater than $\frac{1}{2}$ which in numerical experiments proved to be the optimal value. Figure 2 shows the non-uniform coarsening strategy applied to a 16×8 uniform grid.

The idea of the non-uniform coarsening strategy is to make the problem on the coarser grids more isotropic. Looking at stencil (10), we observe that an isotropic problem is obtained if

$$-\frac{h_{\phi,j}}{h_\theta} \frac{1}{\sin \phi_j} \approx -\frac{h_\theta}{h_{\phi,j}} \sin \phi_j, \quad (11)$$

where h_θ is constant on each grid whilst $h_{\phi,j}$ varies with ϕ on the coarser grids. Equality in (11) is achieved if $\frac{h_{\phi,j}}{h_\theta} \rightarrow \sin(\phi_j)$ as the grid is coarsened. Figures 3(a) to 3(d) monitor the aspect ratio $\frac{h_{\phi,j}}{h_\theta}$ obtained by our algorithm on progressively coarser grids. The stars represent the aspect ratio at each value of ϕ_j , and it becomes clear that this ratio does indeed converge to $\sin(\phi_j)$ as the grid is coarsened. Hence this coarsening strategy yields an isotropic problem on the coarser grids, which is a heuristic explanation of the optimal convergence. We confirm this claim with a simple test for the two dimensional problem. The Poisson equation (9) on the unit sphere is solved using a standard multigrid V-cycle with pointwise Gauss–Seidel smoother combined with the conditional semi-coarsening described above. The stopping criterion is the relative reduction of the residual norm by a factor 10^{-8} . In addition to CPU times and to the numbers of iterations N_{its} , we also give the (geometric) average of the V-cycle convergence factor (excluding the first cycle), i.e.

$$\mu_{\text{avg}} = \left(\|\mathbf{r}_F^{(N_{\text{its}})}\| / \|\mathbf{r}_F^{(1)}\| \right)^{1/(N_{\text{its}}-1)} \quad (12)$$

Table 1 shows that the time taken to solve (9) increases linearly with problem size, and that the number of iterations as well as the V-cycle convergence factor remain constant, which shows that the method is robust and performs optimally. Note that the coarsening factor from grid level to grid level is about 3.

In contrast, Table 2 demonstrates that neither full coarsening nor semi-coarsening (on all latitudes) are optimal in conjunction with point relaxation. In both cases the number of iterations grows strongly with problem size.

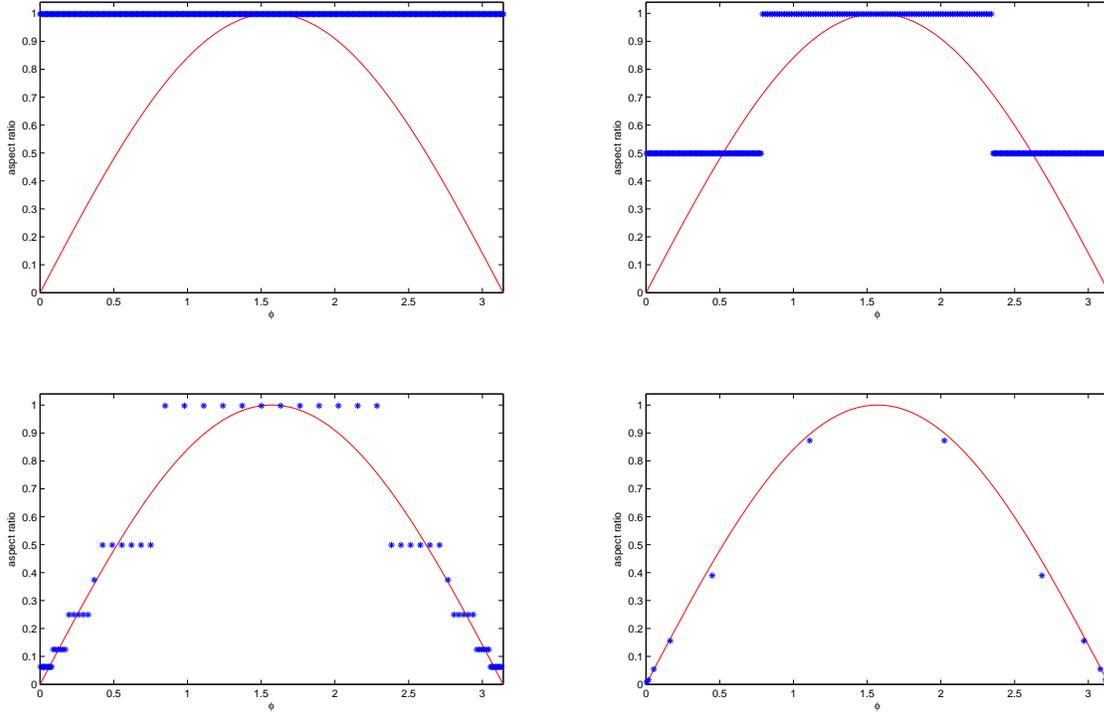


Figure 3: Aspect ratio $\frac{h_\phi}{h_\theta}$ for (top left) zero, (top right) one, (bottom left) four and (bottom right) seven refinements

Problem size	# Coarse grids	Setup time (s)	Solve time (s)	# Iterations	μ_{avg}
32x16	2	1.63E-3	2.27E-3	9	0.102
64x32	3	5.12E-3	8.63E-3	9	0.114
128x64	4	1.80E-2	3.61E-2	9	0.118
256x128	5	6.86E-2	1.70E-1	9	0.118
512x256	6	2.67E-1	7.86E-1	9	0.119

Table 1: Two dimensional Poisson's equation on the unit sphere solved using non-uniform MG (with a projection onto the range of A in each iteration). CPU time in seconds.

Line smoothing and no vertical coarsening

Let us come back to the original three dimensional problem. The anisotropy introduced by the spherical polar grid is of the exactly same type as in two dimensions and so we adopt the same coarsening strategy in the $\theta - \phi$ plane. However, there is a second source of anisotropy in three dimensions due to the large grid aspect ratio between the radial direction and the horizontal directions. In typical computations at the MET Office, the mesh widths h_θ and h_ϕ are $O(10^{-2})$, whereas in the radial direction we have

$$h_{r,1} = O(10^{-6}) \leq h_{r,k} \leq h_{r,n_r} = O(10^{-3}).$$

Therefore, we have $h_{r,k} \ll h_\theta$ and $h_{r,k} \ll h_\phi$ for all k .

		Full Coarsening		Semi-Coarsening	
Problem size	# Coarse Grids	Solve time (s)	# Iterations	Solve time (s)	# Iterations
32x16	2	7.68E-3	34	2.56E-3	8
64x32	3	8.61E-2	103	1.21E-2	10
128x64	4	1.66	471	9.34E-2	18
256x128	5	40.36	2396	1.07	42
512x256	6	875.70	11139	14.88	125

Table 2: Two dimensional Poisson’s equation on the unit sphere solved using full coarsening and (uni-form) semi coarsening (with a projection onto the range of A in each iteration).

Let us first consider $f_0^2 = 1$ and $N^2(r) \equiv 1$, i.e. Poisson equation in three dimensions in the spherical shell. The solution of this problem is also of great importance in numerical weather forecasting, but the traditional solvers, employed at the UK MET Office for example, are coping better with this problem, as we shall see below. The small mesh widths in the radial direction mean that the behaviour of the off-diagonal entries in the stencil at each grid point is dominated by the $h_{r,k}$ dependency, i.e.

$$O\left(\frac{h_0 h_\phi}{h_{r,k}}\right) \begin{bmatrix} O(h_{r,k}) & O(h_{r,k}) \\ O(h_{r,k}) & -\sum O(h_{r,k}) \\ O(h_{r,k}) & O(h_{r,k}) \end{bmatrix} O\left(\frac{h_0 h_\phi}{h_{r,k}}\right)$$

where $h_{r,k} \ll \frac{h_0 h_\phi}{h_{r,k}}$, for all k . The anisotropy is therefore very large and we deal with this by modifying our multigrid method in the usual way, i.e. by using r -line relaxation, namely r -line Gauss–Seidel. We see in Table 3 that in combination with the non-uniform coarsening strategy in the $\theta - \phi$ plane this leads to an extremely efficient method with a V-cycle convergence factor of about 10^{-8} . Note that the anisotropy is in fact so large that r -line Gauss–Seidel on its own and conjugate gradients (CG) preconditioned with r -line Jacobi – which is essentially the same as the method currently employed by the MET Office [10] – work very well too (cf. Table 3). However, neither of these two other methods is robust to grid refinement and the number of iterations grows as the grid resolution is increased, such that for large (typical) problem sizes the multigrid method outperforms both of them.

Problem size	Non-uniform Multigrid		r -line Gauss-Seidel		Preconditioned CG	
	# Iterations	Total time	# Iterations	Total time	# Iterations	Total time
32x16x8	1	6.57E-3	3	2.00E-3	3	3.77E-3
64x32x16	1	5.89E-2	6	3.11E-2	5	2.82E-2
128x64x32	2	4.99E-1	26	1.08	10	4.38E-1
192x120x48	2	3.83	136	23.72	28	4.97

Table 3: Three dimensional Poisson’s equation in the spherical shell. Stopping criterion: Relative residual reduction of 10^{-8} . (CPU time in seconds.)

Now let us consider the QG- ω equation (1). In this case $f_0^2 = O(10^{-8})$ and $N^2(r) = O(10^{-4})$, which complicates the situation drastically, since it largely reduces the strength of the anisotropy in the radial direction, such that the matrix entry in the radial direction is not always the largest at each grid point

anymore. The stencil (7) for the QG- ω equation is now dominated by the following terms

$$\mathcal{O}\left(f_0^2 \frac{h_\theta h_\phi}{h_{r,k}}\right) \left[\begin{array}{ccc} \mathcal{O}\left(N^2(r_k)h_{r,k}\right) & \mathcal{O}\left(N^2(r_k)h_{r,k}\right) & \\ & - \Sigma & \\ \mathcal{O}\left(N^2(r_k)h_{r,k}\right) & \mathcal{O}\left(N^2(r_k)h_{r,k}\right) & \end{array} \right] \mathcal{O}\left(f_0^2 \frac{h_\theta h_\phi}{h_{r,k}}\right)$$

and so the ratio of the radial entry to the horizontal entries has been reduced by a factor $\frac{f_0^2}{N^2(r_k)} = \mathcal{O}(10^{-4})$ (w.r.t. Poisson's equation above). This means that for common grid resolutions, towards the bottom of the atmosphere the aspect ratio of the radial entry to the horizontal entries is $\mathcal{O}(10^{-4})$, whereas towards the top of the atmosphere the aspect ratio is $\mathcal{O}(10^{+2})$, i.e. the radial entry is actually the smaller one. This reduces the smoothing properties of r -line relaxation (as shown in [5]) and it is necessary to combine it with semi-coarsening, i.e. carry out no coarsening in radial direction. This fact has already been noted in [26], where they solved the QG- ω equation (1) in some part of the atmosphere with doubly periodic boundary conditions using multigrid methods. It is again related to the theory for (8) in [5].

Finally, to summarise the multigrid method we propose for (2): We employ the standard V-cycle with linear interpolation and full weighting restriction with r -line relaxation smoother (Gauss–Seidel) and a non-uniform coarsening strategy. We employ no coarsening in the radial direction and conditional semi-coarsening (as described above) in the $\theta - \phi$ plane. For the coarse grid solve we use the smoother and iterate until the relative residual is reduced by 10^{-2} . In the next section we will study the robustness and efficiency of this method numerically and compare it to some other methods.

Similarities with algebraic multigrid methods

The success of non-uniform (conditional) coarsening strategies for anisotropic elliptic problems has already been demonstrated with the highly successful algebraic multigrid (AMG) methods. As stated in [21], the advantage of AMG is its robustness and applicability in any complex geometric situations which are out of reach of geometric multigrid methods. In addition, AMG methods usually use matrix-dependent prolongation operators which further enhance its robustness, particularly for problems with large (non-smooth) coefficient variation.

Thus for many complex problems, AMG is the only approach that can be used. However, the flexibility of AMG comes at a price: its setup cost. The selection of coarse nodes, the construction of interpolation operators and the construction of coarse level operators is slower for AMG than for geometric methods (especially in 3D), since everything has to be deduced algebraically from the system matrix A via graph theoretical techniques. Also, the coarse grid operators generally become very dense and expensive to apply. Therefore, AMG is usually less efficient than geometric multigrid on problems for which geometric multigrid can be suitably adapted. The QG- ω equation is a highly anisotropic problem, but as discussed above it can be dealt with efficiently and robustly using geometric approaches by exploiting the particular structure of the grid anisotropy. Because of the reduced setup cost and the sparser coarse grid matrices it should outperform standard AMG methods, such as the popular `BoomerAMG` from the Hypre library [16, 12], comfortably. In fact, in Section 4 we will compare our geometric method with `BoomerAMG`. Note however, that there are also AMG codes that are more tailored to anisotropic problems (e.g. [14]) which may be cheaper than `BoomerAMG` for (2), but they are still unlikely to outperform a robust geometric method like the one presented here.

4 Numerical Results – Robustness and Comparisons

All tests in this section are carried out on a single 1.8GHz processor of a Dual dual-core 64bit AMD Opteron 2210 (Cache size 1.0MB and 2GB memory) using the Fortran95 compiler `ifort`. The non-uniform geometric multigrid algorithm (NUMG) for solving (2) uses the V-cycle scheme [7]. The components of the algorithm have been described in detail in Section 3. The initial guess for the iteration is taken to be zero, with a relative residual reduction of $\epsilon = 10^{-8}$ as the stopping criterion. The number of pre- and post-smoothing steps is $\nu_1 = 3$ and $\nu_2 = 2$.

We compare the method with BoomerAMG, an AMG preconditioner from the Hypra library [16, 12]. We use the default settings for BoomerAMG, i.e. a symmetric-SOR/Jacobi smoother, Falgout coarsening, classical Ruge-Stüben interpolation and Gaussian Elimination as the coarse grid solver. Some experiments with other settings have not led to a significant improvement of the method. Both NUMG and AMG are tested stand-alone and as preconditioners for the conjugate gradient (CG) method (one V-cycle per iteration). We also use r -line Jacobi as a preconditioner for CG. This is essentially the solver currently used by the MET Office. The actual Krylov subspace method used at the MET Office is the *generalised conjugate residual* (GCR) method [10]. This method is also applicable to nonsymmetric matrices, but (in exact arithmetic) it reduces to CG for symmetric positive definite problems such as (2). Since it requires more floating point operations and the storage of all previous search directions, it is substantially more expensive (see Table 7), and so we also use CG to get a fairer comparison. We do however choose exactly the same preconditioner as that employed at the MET Office, i.e. r -line Jacobi relaxation.

Table 4 gives the performance for solving (2) using NUMG (as a solver and as a preconditioner for CG) with $f_0^2 = 10^{-8}$ and $N^2(r) = \mathcal{O}(10^{-4})$ as provided by the MET Office. The number of iterations and the convergence rate per iteration (cf. (12)) is constant (asymptotically) suggesting full robustness of the solver with respect to an increase in problem size. The CPU times for the setup and for the solve phase are both scaling linearly with the problem size and so the method is optimal. There is hardly any difference in the performance of NUMG as a solver or as a preconditioner for CG.

Problem size	Setup time	NUMG (V-cycle only)			CG + NUMG		
		Solve time	# Iterations	μ_{avg}	Solve time	# Iterations	μ_{avg}
32x16x8	1.06E-2	4.16E-2	7	0.050	4.43E-2	7	0.057
64x32x16	7.97E-2	3.42E-1	7	0.072	4.10E-1	8	0.078
128x64x32	6.29E-1	3.54	8	0.093	3.72	8	0.080
256x128x64	4.94	29.94	8	0.101	34.24	8	0.081

Table 4: Non-Uniform geometric multigrid applied to the QG- ω equation in 3D. (CPU times in seconds.)

In Table 5 we see that all the modifications to the standard multigrid V-cycle which we applied were necessary, and that without the modifications geometric multigrid is not robust to grid refinement.

Table 6 gives the performance for solving (2) using AMG as a preconditioner for CG and stand-alone. We observe from this table that, although CG + AMG is also robust with respect to grid refinement, the solve time is higher than that of NUMG by a factor of about 3.5 and the difference gets larger as the problem size is increased. As for the setup time, NUMG is approximately an order of magnitude faster than AMG, so that in total NUMG is almost 5 times faster than CG + AMG. When used as a stand-alone solver AMG is not fully robust. The number of iterations and the average convergence factor per iteration grow slightly with the problem size.

Table 7 shows that without a multilevel preconditioner, Krylov subspace methods such as CG or

Coarsening:	(a) θ & ϕ direction		(b) θ direction only		(c) r & conditional θ - ϕ	
Problem size	Solve time	# Iterations	Solve time	# Iterations	Solve time	# Iterations
32x16x8	0.11E-3	23	5.04E-2	6	4.39E-2	7
64x32x16	3.44E-2	84	0.74	9	0.66	16
128x64x32	174.54	441	15.26	16	30.69	86
256x128x64	***	>1000	332.81	36	271.37	228

Table 5: Geometric multigrid with r -line Gauss-Seidel smoother applied to the QG- ω equation in 3D, using (a) full coarsening on the $\theta - \phi$ plane and no coarsening in r , (b) semi-coarsening on the $\theta - \phi$ plane and no coarsening in r , or (c) non-uniform coarsening on the $\theta - \phi$ plane and full coarsening in r . (CPU times in seconds.)

Problem size	Setup time	BoomerAMG (V-cycle only)			CG + AMG		
		Solve time	# Iterations	μ_{avg}	Solve time	# Iterations	μ_{avg}
32x16x8	4.05E-2	9.37E-2	7	0.059	8.05E-2	5	0.024
64x32x16	6.11E-1	1.37	8	0.083	1.12	5	0.029
128x64x32	6.76	15.62	10	0.134	12.37	6	0.049
256x128x64	57.64	133.80	11	0.163	108.55	7	0.058

Table 6: BoomerAMG applied to the QG- ω equation in 3D. (CPU times in seconds.)

GCR (the method currently employed by the MET Office) are not robust to grid refinement. The number of iterations grows and therefore the CPU time is not proportional to the problem size. At the typical grid sizes currently used at the MET Office, NUMG is more than 10 times faster than CG with r -line Gauss-Seidel preconditioner.

Problem size	Conjugate Gradients		Generalised Conjugate Residuals	
	Solve time (s)	# Iterations	Solve time (s)	# Iterations
32x16x8	2.83E-2	57	5.19E-2	62
64x32x16	0.70	132	1.21	170
128x64x32	20.03	370	40.06	525
256x128x64	397.24	879	***	>1000

Table 7: Preconditioned Krylov subspace methods applied to the QG- ω equation in 3D (with r -line Gauss-Seidel preconditioner).

5 Parallel Tests

5.1 Parallelisation Strategy

Because of the r -line smoother it would not be good in the parallelisation of the method to partition the domain in the radial direction. The tridiagonal solves along each r -grid line would lead to too much unnecessary communication and data dependencies between the processors. Therefore we partition only in the longitudinal and latitudinal directions. This is also the case in the MET Office codes. However,

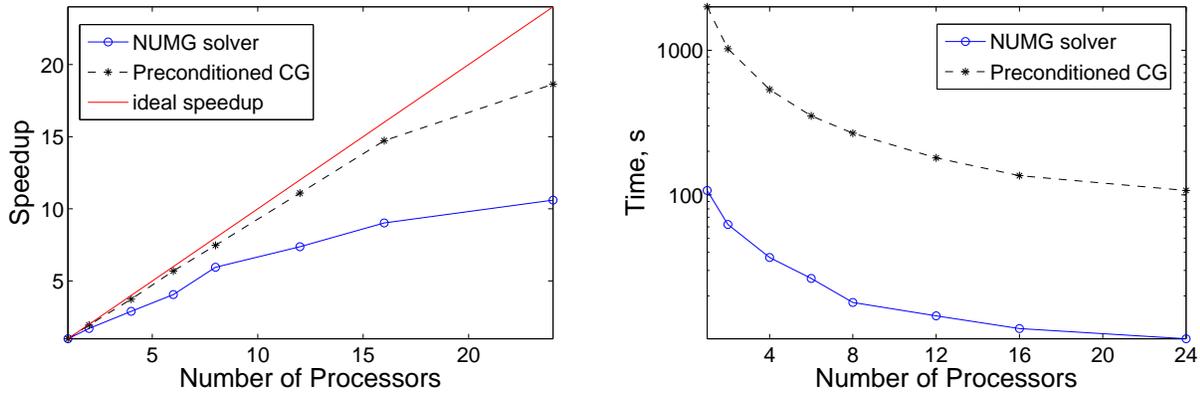


Figure 4: Speedup test on `wolf`: Global problem size $360 \times 180 \times 100$

any number of partitionings in these directions should be admissible.

A ghost point (or halo) strategy [11] is used for communication between processors, which is implemented using MPI (message passing interface). Note that we must handle the communication at the boundaries of the domain differently to account for the poles and the periodic boundary conditions. The discretisation scheme, the choice of coarse level matrices and the choice of restriction and prolongation R_ℓ and P_ℓ , are chosen such that communication only occurs with adjacent processors. Thus, we have a communication topology which resembles a 5-point stencil. The conversion of a distributed vector to an accumulated vector (cf. [11]) therefore requires communication only with four neighbouring processors.

To reduce inter-processor communication, the smoother is modified to a hybrid Jacobi/Gauss–Seidel smoother, only making use of the most up-to-date values of the solution vector if the corresponding node is associated with the same processor (within one relaxation step). If the node is a ghost point and belongs to another processor then the value from the previous relaxation sweep is used (as in the Jacobi method). Thus all the communication during the smoothing iteration is concentrated at the start of each sweep. This may lead to a slight increase in the number of iterations on large numbers of processors, but it avoids unnecessary communication and data dependencies.

Finally, we modify the coarsening strategy slightly to ensure that all coarse grids are partitioned along the same planes to avoid extra communication when using the grid transfer operators R_ℓ and P_ℓ . This leads to minor modifications at processor boundaries.

Note, that in our partitioning strategy, subdomain blocks near the equator will have a four-fold reduction in the number of grid points from one grid level to the next as a result of full coarsening, while blocks near the poles will only have a two-fold reduction. This leads to a load imbalance on coarser grids. However, the majority of the work is done on the finest grid and so it is sufficient to make sure the load is distributed evenly on that level. Our numerical tests confirmed this. The amount of time certain processors are idle is minimal.

5.2 Parallel Numerical Results – Speedup and Scaling

We test the parallel non-uniform multigrid code on two different clusters, a 64-bit AMD Opteron 2210 cluster (`wolf`) with a total of 24 processors (the same ones as in Sections 3 and 4) and a 64-bit Intel Xeon E5462 cluster (`aquila`) with 2GB memory and 3MB Cache per processor. Both clusters use an

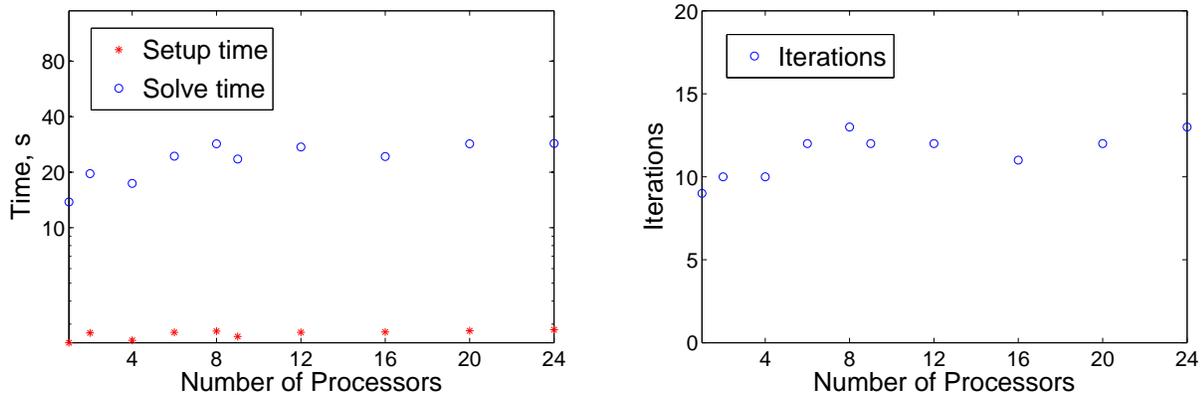


Figure 5: Scaled efficiency test on `wolf`: Problem size $200 \times 100 \times 50$ on each processor.

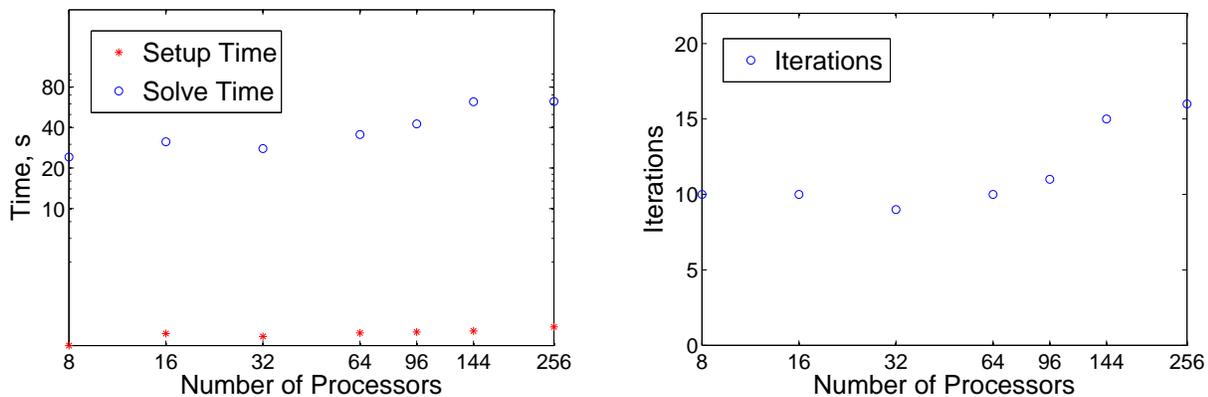


Figure 6: Scaled efficiency test on `aquila`: Problem size $192 \times 120 \times 50$ on each processor.

Infinipath network.

We firstly perform a speedup test, which is a measure of the performance gain of a parallel code running on N processors over the sequential code where the global problem size remains unchanged. We test this on a problem of size $360 \times 180 \times 100$, with the results shown in Figure 4. The speedup is very good (almost optimal) on up to about 8 processors. However, for larger numbers of processors the amount of work that each of the processors has to do becomes too small and so the speedup drops off slightly from the optimal (linear) growth.

A better test for how well the implementation scales on larger numbers of processors is a scaled efficiency test, where the problem size per processor is kept fixed as the number of processors is increased. In this test the CPU time of a method that scales optimally should remain constant as the number of processors is increased. Figures 5 and 6 show how the method scales on the two different clusters with respect to CPU time and the number of iterations. In Figure 5 we fix the problem size per processor to $200 \times 100 \times 50$ on `wolf`, and observe that the method scales almost optimally, particularly beyond four processors. On `aquila` we fix the problem size per processor to $192 \times 120 \times 50$ and observe also

very good parallel scaling with a slight increase of the CPU time for more than 96 processors. This is because the number of iterations jumps from about 10 to about 15 when using more than 96 processors, potentially because of the hybrid smoother or because the anisotropy in the r -direction is getting weaker. Nevertheless, we see that on 256 processors with the new non-uniform multigrid method, it is possible to solve the QG- ω equation on the entire globe with a resolution of 10km at the equator and 3×10^9 unknowns in about 60 seconds. On larger clusters, potentially even finer resolutions could be solved within the same time.

Finally let us compare the parallel NUMG solver with parallelisations of the other methods. Krylov subspace methods with r -line Jacobi preconditioners (such as the ones used at the Met Office) are extremely well suited to an efficient parallelisation, and the numerical results in Figure 4 show this clearly. The speedup is almost optimal (linear) on any number of processors. However, as we saw in Section 4, the method is not robust, i.e. the number of iterations grows with the problem size, and with the typical grid resolution of $360 \times 180 \times 100$ used at the Met Office, the multigrid method is about 10 times faster than r -line Jacobi preconditioned CG on 24 processors. For finer grid resolutions, such as the ones used in the scaling tests in Figures 5 and 6, the preconditioned CG method becomes increasingly inferior to NUMG. As for BoomerAMG, this does not parallelise as well as the geometric multigrid, particularly in the setup phase. The speedup and the scaled efficiency are substantially smaller than for NUMG.

References

- [1] C.F. Baillie, J.C. McWilliams, J.B. Weiss, and I. Yavneh, Implementation and performance of a grand challenge 3D quasi-geostrophic multi-grid code on the Cray T3D and IBM SP2, Technical Report CU-CS-771-95, University of Colorado at Boulder, Department of Computer Science, 1995.
- [2] S.R.M. Barros, Multigrid methods for two and three dimensional Poisson-type equations on the sphere, *J. Comp. Phys.* **92**, 313-348, 1991.
- [3] J.R. Baumgardner, P.O. Frederickson, Icosahedral Discretisation on the Two-Sphere, *SIAM J. Numer. Anal.* **22**(6), 1107-1115, 1985.
- [4] H. Bluestein, *Synoptic-Dynamic Meteorology in Midlatitudes. Volume I: Principles of Kinematics and Dynamics*, Oxford University Press, Oxford, 1992.
- [5] S. Börm, R. Hiptmair, Analysis of tensor product multigrid, *Numerical Algorithms*, **26**, 219-234, 2001.
- [6] J.H. Bramble and X. Zhang, Uniform convergence of the multigrid V-Cycle for an anisotropic problem, *Math. Comp.* **70**(234), 453-470, 2001.
- [7] W.L. Briggs, V.E. Hanson, S.F. McCormick, *A Multigrid Tutorial*, SIAM, 2000.
- [8] D.E. Brown, N.K. Nichols and M.J. Bell, Preconditioners for inhomogeneous anisotropic problems in spherical geometry, *Int. J. Numer. Meth. Fluids* **47**, 1213-1219, 2005.
- [9] M.J. Cullen, *A Mathematical Theory of Large-Scale Atmospheric/Ocean Flow*, Imperial College Press, 2006.
- [10] T. Davies, M.J. Cullen, A.J. Malcolm, M.H. Mawson, A. Staniforth, A.A. White and N. Wood, A new dynamical core for the MET Office's global and regional modelling, *Q. J. R. Meteorol. Soc.* **131**, 1759-1782, 2005.
- [11] C. Douglas, G. Haase, U. Langer *A Tutorial on Elliptic Solvers and Their Parallelisation*, SIAM, 2003.

- [12] R. Falgout, V.E. Henson, U.M. Yang et al, Hypre: High Performance Preconditioners, Software Library, User's Manual (Version 2.4.0b), Lawrence Livermore National Laboratory, Livermore, CA, 2008. <https://computation.llnl.gov/casc/linear_solvers/sls_hypre.html>.
- [13] S.R. Fulton, P.E. Ciesielski and W.H. Schubert, Multigrid methods for elliptic problems: A review, *Mon. Wea. Rev.* **114**, 943-959, 1986.
- [14] M.W. Gee, J.J. Hu and R.S. Tuminaro, A new smoothed aggregation multigrid method for anisotropic problems, *Numer. Lin. Alg. Appl.* **16**, 19-37, 2009.
- [15] W. Hackbusch, *Multigrid Methods and Applications*, Springer-Verlag, 1985
- [16] V.E. Henson and U.M. Yang, BoomerAMG: A parallel algebraic multigrid solver and preconditioner, *Appl. Numer. Math.* **41**, 155-177, 2002.
- [17] M. Kameyama, A. Kageyama, T. Sato, Multigrid-based simulation code for mantle convection in spherical shell using YinYang grid, *Physics of the Earth and Planetary Interiors* **171**, 19-32, 2008.
- [18] J. Larsson, F.S. Lien and E. Yee, Conditional semi-coarsening multigrid algorithm for the Poisson equation on anisotropic grids, *J. Comput. Phys.* **208**, 368-383, 2005.
- [19] J.W. Ruge, Y. Li, S. McCormick, A. Brandt and J.R. Bates, A nonlinear multigrid solver for a semi-Lagrangian potential vorticity-based shallow-water model on the sphere, *SIAM J. Sci. Comput.* **21**, 2381-2395, 2000.
- [20] R. Stevenson, Robustness of multi-grid applied to anisotropic equations on convex domains and on domains with re-entrant corners, *Numer. Math.* **66**, 373-398, 1993.
- [21] K Stüben, *Algebraic Multigrid (AMG): An Introduction with Applications*, GMD-Studien NR. 53, 1999.
- [22] U. Trottenberg, C. Oosterlee, A. Schüller, *Multigrid*, Academic Press, 2001.
- [23] D.L. Williamson, The Evolution of Dynamical Cores for Global Atmospheric Models, *J. Meteorol. Soc. Jpn.* **85B**, 241-269, 2007.
- [24] I. Yavneh, A.F. Shchepetkin, J.C. McWilliams and L.P. Graves, Multigrid solution of rotating, stably stratified flows: The balance equations and their turbulent dynamics, *J. Comput. Phys.* **136**, 245-262, 1997.
- [25] I. Yavneh and J.C. McWilliams, Robust multigrid solution of the shallow-water balance equations, *J. Comput. Phys.* **119**, 1-25, 1995.
- [26] I. Yavneh and J.C. McWilliams, Multigrid Solution of stably stratified flows: The quasigeostrophic equations, *J. Sci. Comput.* **11**, 47-69, 1996.
- [27] H. bin Zubair, S.P. MacLachlan and C.W. Oosterlee, A geometric multigrid method based on L-shaped coarsening for PDEs on stretched grids, Technical Report, Delft University of Technology, Report 08-15, 2008.