University of Reading

Preconditioners for Inhomogeneous Anisotropic Problems with Spherical Geometry in Ocean Modelling

David Brown

This thesis is submitted for the degree of
Doctor of philosophy

Department of Mathematics

August 2004

Abstract

Elliptic equations arising in free-surface ocean models are typically solved using iterative methods. Mesh anisotropy associated with standard co-ordinate systems causes the convergence of the iterative methods to be slow, particularly in polar regions. This is demonstrated here using a preconditioned conjugate gradient (PCG) iterative method with diagonal preconditioning.

Numerical evidence is presented, using a 2D spherical domain model and a standard five-point discretisation scheme, to show that the polar convergence problem is caused by the increased importance, with increased mesh anisotropy, of eigenmodes with strong polar signals. Block diagonal, alternating direction implicit (ADI) and Binormalization preconditioners are considered here as alternative preconditioners for the PCG method. Their impact on reducing the computing time and improving the polar convergence issue is investigated. Theoretical estimates for the rates of convergence for the different preconditioners are derived and tested using numerical experiments with varying mesh anisotropy. The ADI preconditioner is found to provide the fastest convergence with the most improvement to the polar convergence problem. Block diagonal and Binormalization preconditioning also offer some improvements.

The work is extended to include a varying topography operator within the elliptic equations and the use of a particular nine-point discretisation scheme. The applicability of the proposed preconditioners to the problems in these extended cases is confirmed. ADI is again found to provide the fastest convergence. The results highlight the importance of using the correct stopping criteria to obtain the most accurate results.

Acknowledgements

I firstly thank my family for their continual love, encouragement and finance over the years of my education. This thesis is dedicated to them.

I would like to thank my academic supervisor Prof. Nancy Nichols for her boundless enthusiasm and invaluable assistance during my research. I also thank my industrial supervisor Dr. Mike Bell for the support and interest he has shown throughout.

My appreciation also to Dr. Beatrice Pelloni and Dr. Amos Lawless for their additional support and supervision and to Dr. Dave Storkey and Dr. Graham Rickard for their help in understanding the Met.Office models.

To my girlfriend, Cara : your patience and understanding during the more stressful moments were a great comfort.

To my friends in many spheres, from my colleagues in the Mathematics department, to those others I have met at Reading University, to my fellow supporters from Chelsea football club, to my oldest friends from home : you have all played a vital role in keeping me grounded in reality (and sanity!) and have been a great source of strength and support.

To those whose excellent teaching got to me to where are I am now, both from Reading University and Bristol Grammar School; you provided the sound basis on which this research could be done.

Finally I acknowledge the financial support of the Engineering and Physical Sciences Research Council (E.P.S.R.C.) and the Met.Office.

Declaration

I confirm that this is my own work and the use of all materials from other sources has been fully and properly acknowledged.

Contents

List of Figures

List of Tables

List of Parameters

Parameter	Description	Parameter	Description
$\mathbf{U}$	General solution vector	α', γ'	Used to time-centre Coriolis terms
H	Depth of ocean	α, γ	Used to time-centre pressure gradient
ψ	Streamfunction	θ	Used to time-centre divergence terms
η	Free surface height	β	Helmholtz term
k	Helmholtz parameter	c	Wave speed
λ	Longitude	r_D	Radius of deformation
ϕ	Latitude	$\mathbf{s}$	Source vector
p	Pressure	A	System matrix
ρ	Density	n_λ	No. of grid points in λ direction
S	Salinity	n_ϕ	No. of grid points in ϕ direction
T	Temperature	ϕ_{NB}	Latitude of northern boundary
u, v	Horizontal velocities	P	Preconditioner
$\bar{u}$, $\bar{v}$	Barotropic velocities	$\mathbf{r}$	Residual
a	Radius of the Earth	$\mathbf{d}$	Conjugate gradient search vector
g	Gravitational acceleration	m	Iteration number
f	Coriolis parameter	N	Size of A
w	Vertical velocity	G_P	Iteration matrix (preconditioner P)
L	Advection operator	μ_i	i^{th} eigenvalue
D	Diffusion operator	$\mathbf{w_i}$	i^{th} eigenvector
u', v'	Baroclinic velocities	a_{ij}	Element of A (i^{th} column, j^{th} row)
p_s	Surface pressure	$\rho()$	Spectral radii
p_h	Hydrostatic pressure	$\kappa()$	Condition number
G^x, G^y	Baroclinic forcing	$\mathbf{b}$	Source vector
H_Υ	'Horizontal' matrix for ADI	V_Υ	'Vertical' matrix for ADI
Υ	Parameter for ADI	$\mathbf{e}$	Error vector
δt	Time step	ω_m	Chebyshev parameter at iteration m

Chapter 1

Introduction

The world ocean is a key component in the climate system of the Earth. This is due to its size (covering 70% of the Earth's surface) and its high heat capacity. It is also the largest store of carbon dioxide in the Earth's carbon cycle. The study of physical oceanography, however, is a comparatively recent research area, starting with the theoretical understanding of western boundary currents, such as the Gulf Stream, in the 1940's. Numerical modelling of the ocean began even more recently, in the late 1960's, and has expanded greatly in the last 15-20 years with the introduction of high performance, massively parallel, supercomputers.

Most ocean models in use today are based on integrating the incompressible primitive equations on a sphere [36]. Complex topography is used at the ocean bottom, and the ocean surface is either fixed or free to move with time. The ocean basins themselves typically contain irregularly shaped coastlines and islands which require the inclusion of specific boundary conditions into any solution algorithm.

The solution of elliptic problems are commonly required in many ocean models. Various excellent references exist which discuss the known properties of general elliptic operators and the resulting matrix forms of the discretised equations ([3], [6], [33], [38], [40] [62], [75]). The problem we consider is that of a modified Helmholtz equation on a sphere with appropriate boundary conditions using a latitude-longitude co-ordinate framework. This is an isotropic problem which is rendered mesh anisotropic by the choice of coordinates. The coordinate framework also introduces singularities at the poles of the domain. The singularity at the South Pole is not a problem as it falls on land (Antarctica). However, the North Pole singularity does not fall on land and must be addressed. Methods for doing this are described in Chapter 2. An operator is anisotropic if its local properties vary with direction (i.e. the operator is not invariant with respect to direction). As an example consider the constant coefficient partial differential equation

$$-\frac{\partial}{\partial x}\left(L_x\frac{\partial U}{\partial x}\right)-\frac{\partial}{\partial y}\left(L_y\frac{\partial U}{\partial y}\right)=\gamma(x,y), \tag{1.1}$$

where L_x and L_y are taken here to be constant. Note that the case $L_x = L_y = 1$ is just the Laplacian operator which, when discretised on a regular Cartesian grid, is known to be relatively easy to model. If we alter the coefficients though, making L_x much larger than L_y, the operator becomes poorly conditioned. This is an example of strong anisotropy, where the ratio between the maximum and minimum ranges (the anisotropy ratio) is large.

In the ocean models we consider, the effects of mesh anisotropy are seen in the latitudinally varying rates of convergence of the iterative methods used to solve the elliptic problems. Poor rates of convergence are observed in polar regions compared to equatorial and mid-latitude regions. In these cases L_x and L_y are non constant, with $L_x \approx L_y$ in equatorial regions, but $L_x >> L_y$ in polar regions. This is an example of inhomogeneous anisotropy. In these cases an operator is inhomogeneous if its properties, when measured in a particular direction, change with location. An inhomogeneous anisotropic operator is therefore an operator whose ratio of anisotropy varies spatially. The ratio is large in polar regions, whereas in equatorial regions it is close to 1 (isotropy). A typical convergence result encountered with this type of operator is shown in Figure 1.1. This shows the variation in the residual errors with latitude for a "Northern Hemisphere only" numerical experiment with the free surface formulation of the Met.Office ocean model. This is shown 'at convergence' i.e. when the convergence criteria for the numerical iterative method has been reached. Significantly higher residual error values are observed in the polar region compared to the equatorial and mid-latitude regions.

One of the main aims of this thesis is discover how the mesh anisotropy causes the polar convergence issue shown in Figure 1.1. Also we aim to investigate the effect that the mesh anisotropy has on the numerics and conditioning of the discrete elliptic problem. It is often very time-consuming for an ocean group to investigate and implement a new co-ordinate system. We will therefore investigate ways to improve the existing iterative methods that are used to solve the discrete problems, obviating the need for a change in co-ordinate system. In particular we will devise preconditioners to improve the existing iterative methods, with particular focus on addressing the polar convergence issue.

In Chapter 2 we discuss the history behind the various models that are used in ocean modelling today and highlight differences in their formulations. We discuss the Bryan-Cox-Semtner model ([13], [18], [66], [67]), on which most current operational models are based. We summarise the two main formulations of the Bryan-Cox-Semtner model used to calculate the barotropic (depth-averaged) flow: rigid-lid and implicit free surface. The numerical solution for the free-surface height field, calculated in the latter,

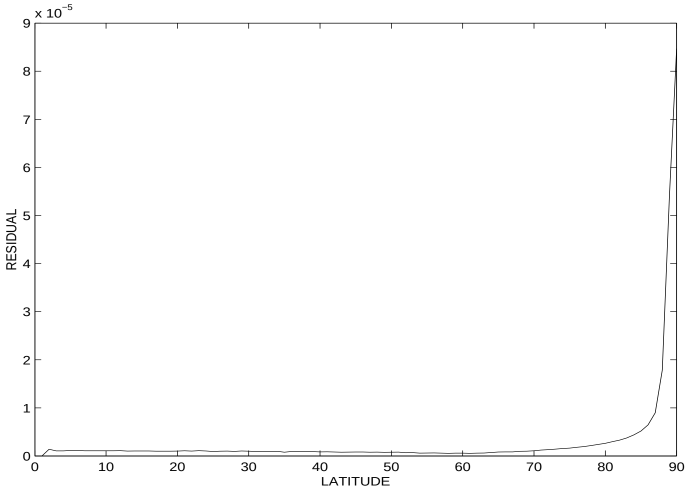

Figure 1.1: Latitudinal variance in residual errors at convergence, for experiment with free surface formulation of Met.Office ocean model (Northern Hemisphere only)

suffers from a growing chequerboard (+−) mode ([24],[46]). The reasons for the appearance of this mode as well as the filter most commonly used to remove it are described ([46]). We also discuss the grids used most commonly to position the model variables in the computational mesh ([4]). In addition we discuss the spatial discretisation schemes used in the model formulations. In particular we describe how islands are treated with regards to the applications of boundary conditions with the nine-point discretisation operator used in the free surface formulation. In addition we summarise the various ways in which the singularity at the North Pole is resolved ([65],[54]). Finally we briefly summarise the state of global ocean modelling today. We highlight ocean groups worldwide and describe the model formulations and discrete grids they are using in their models.

Chapter 3 highlights the mathematical theory which is crucial to the study of this problem. The key theorems and definitions that are used to confirm the validity of the methods and preconditioners that are most commonly used in the rigid-lid and free surface formulations are given, as well as proposed extensions. The rigid-lid formulation is typically solved using a Chebyshev Semi-Iterative method, whilst the free-surface formulation is normally solved using a Preconditioned Conjugate Gradient (PCG) Method with a diagonal preconditioner. These methods are introduced along with alternative preconditioners that we will consider. These include a new Alternating-Direction-Implicit (ADI) preconditioner with spatially varying free parameter.

In Chapter 4 we illustrate the inherent problems by examining an idealised spherical domain model of the ocean, with a five-point discretisation scheme, and obtain theoretical results demonstrating the convergence of the various methods and preconditioners considered. The Gerschgorin circle theorem [75] is applied to our matrix problems in order to obtain qualitative information on the relative speeds of convergence of the preconditioned methods. We also use truncation error analysis to confirm the consistency of the discretisation scheme used to the continuous problems. Our theoretical findings are then tested in a set of numerical experiments, with varying mesh anisotropy, in Chapter 5. These demonstrate how the eigenstructure of the preconditioned methods relates to the polar convergence issue. Also we assess the computational efficiency of our preconditioned methods.

Our idealised spherical domain model is extended in Chapter 6 to include a varying ocean depth operator, H. We firstly consider cases where the elliptic operator is of the form $-\nabla \cdot (H \nabla)$, which is analogous to the operator used in the free-surface formulation. We then consider an elliptic operator of the form $-\nabla \cdot (\frac{1}{H} \nabla)$, which is the generic operator encountered in the rigid-lid formulation. The numerical implications of the form of the operator, and hence the efficiency of the formulation used, are investigated. We also re-examine, both theoretically and experimentally, the convergence of our preconditioners in this varying topography problem.

In Chapter 7 we further extend our idealised spherical domain model to include a nine-point discretisation operator analogous to that used in the free surface ocean model. The applicability of the preconditioned methods to the discrete problems using the nine-point scheme is demonstrated using convergence analysis. Numerical experiments are performed to test the efficiency of our preconditioned methods in this extended case. The importance of the eigenstructure of the preconditioned methods with regards to the polar convergence issue is again confirmed. Also the importance of using the correct stopping criteria in the iteration process is particularly highlighted by the numerical results.

Finally in Chapter 8 we present conclusions to this work and consider some possible future expansions to this research area.

Chapter 2

Overview of ocean modelling

2.1 Introduction

Since the main ideas of this thesis are motivated by the properties of inhomogeneous anisotropic elliptic problems in the context of Ocean General Circulation models (OGCM) we include a chapter discussing the current situation with regard to ocean modelling worldwide, and describe the origin and theory behind the models in which our anisotropic elliptic operators appear.

Most of the OGCMs in use today are based on the work, in the late 1960's, of Bryan [13]. He produced a finite-difference formulation of the Navier-Stokes equations and the nonlinear equation of state. After being programmed in FORTRAN by Cox [18] the model came to be known as the Bryan-Cox model. In the mid-1970's the model was renamed the Bryan - Cox - Semtner model (Henceforth BCS model) when the code was updated, for use with vector processors, by Semtner [66]. This was used until the mid-1980's when the code was further updated by Cox and Semtner [67] using both Cray and Cyber coding.

The BCS model solves the primitive equations, derived from the Navier-

Stokes equations, in a spherical coordinate system, using hydrostatic and Boussinesq approximations. The hydrostatic approximation assumes that the Lagrangian derivative of the vertical velocities is small compared to pressure variations with height and gravitational terms i.e. that the vertical pressure gradient is only due to density variations ([13]). This uses the fact that the ratio of vertical to horizontal scales of motion (aspect ratio) is usually small in the large scale ocean. The Boussinesq approximation assumes that density terms in the primitive equations are replaced by a constant mean value for density, ρ_0, everywhere except in the buoyancy term in the vertical momentum equation. This has the advantage of eliminating shock and sound waves. For a detailed overview of the Primitive equations and the approximations typically used with them see Gill [36].

The oceanic response to surface forcing may be split into two parts : a barotropic response in the form of external Kelvin and gravity waves (on the surface) and a baroclinic response in the form of internal gravity, planetary Rossby, Kelvin and other similar types of wave (see Gill [36] for more details). From the point of view of modelling long (climate level) timescales it is the internal waves that are the more important whilst the external waves are damped if possible. This is because the speeds of surface gravity waves are typically around $200ms^{-1}$ which severely limits the permissible time step of the numerical scheme. The timesteps of the model are limited by a Courant-Friedrichs-Levy criterion which states that diffusive, advective and wave processes may not move more than one discrete grid spacing per time step ([8]).

Bryan [13] eliminated surface gravity waves by introducing a rigid-lid approximation. This means that the surface of the ocean is assumed to be fixed (i.e. does not go up and down) an assumption which effectively makes the phase speeds of all surface gravity waves infinite. This removes the strict constraint on the time step. Rigid lid models are still used extensively today; there is, however, a recent tendency towards reverting to a free surface model (The Met.Office (MO) ocean group is one group considering this). There are a number of advantages of such a formulation over a rigid-lid assumption which we discuss in Section 2.2. Dukowicz and Smith [23], [24] explain fully the theory behind both formulations.

The rigid-lid formulation incorporates a rigid-lid approximation of the ocean model. The problem is decoupled into barotropic and baroclinic velocity components. The barotropic components are vertical averages of the velocities whereas the baroclinic components represent deviations from those vertical averages. In order to obtain the 2D barotropic velocity field it is necessary to solve an elliptic boundary value problem of Poisson type i.e. $-\nabla^2\psi = g(x, y)$, where $g(x, y)$ is a source term. Dirichlet boundary conditions are taken on the boundary of the domain. This problem is solved in terms of the volume transport streamfunction, ψ.

For the free-surface formulation the barotropic equations are rewritten in terms of the free surface height, η, instead of the volume transport streamfunction, i.e. the rigid-lid assumption has been replaced by an implicit free surface condition. The implicit free surface formulation leads to greater computational efficiency due to the adding of a diagonal term to the matrix representation of the discrete elliptic operator. The resulting equation is known as a Modified Helmholtz equation and is of the general form $-\nabla^2\eta + k\eta = g(x, y)$, where $k > 0$. The additional diagonal term increases the diagonal dominance of the associated matrix system of equations hence yielding a more efficient algorithm. The importance of the diagonal dominance of a matrix to the speed of numerical algorithms is discussed in greater detail in later chapters.

In the rigid-lid formulation the barotropic problem uses about 33 percent of the computational time on the Cray machine ([23]). This is mainly due to the inefficiencies introduced by the use of islands in this formulation (discussed in greater detail in Section 2.2.1). With the free-surface formulation this is reduced to less than 25 percent ([24]). However, it is only this large due to the polar convergence issue we have highlighted. If this could be remedied, via the use of a more efficient preconditioner in the iterative method for example, this factor could be reduced still further.

In Section 2.2 we summarise the BCS Model formulation, in particular the barotropic solution in the rigid-lid and free surface cases. We also describe the various possible grid formulations for the arrangement of the model variables. In Section 2.3 we introduce the spatial discretisation schemes commonly used in the BCS model formulations. A standard five-point discretisation scheme is used in the rigid-lid formulation whereas a particular nine-point discretisation scheme is used in the free-surface formulation. We examine the use, with the nine-point scheme, of implicit boundary conditions at the island boundaries as referred to by Dukowicz and Smith [24]. We also highlight issues of a particular type of computational noise which commonly arises when using the nine-point scheme in the free-surface formulation. The filter typically used to deal with this noise is also described. In Section 2.4 we discuss the key role that polar regions play in ocean modelling and describe the methods used today to discretise those regions and resolve the issue of the co-ordinate singularity at the North Pole. Finally in Section 2.5 we provide an overview of the current state of ocean modelling by considering the models used by ocean groups worldwide and describe particularly the numerics and formulations they use.

2.2 BCS model formulation

A full description of the Bryan - Cox - Semtner model may be found in Bryan [13] and Semtner [67]. Here we summarise how the barotropic problem may be formulated in both rigid-lid and free surface forms. We adopt the notation of Semtner [67], with λ, ϕ, and z representing longitude, latitude and depth respectively. The Earth is assumed to be spherical with radius a and to be rotating with angular velocity Ω. The horizontal velocity components are u in the longitudinal direction and v in the latitudinal direction. The vertical velocity component is denoted by w. The state of the ocean is also described by the pressure, p, the density, ρ, the potential temperature, T, and the salinity, S. The model is simplified by various assumptions, namely the Hydrostatic and Boussinesq approximations described in Section 2.1. The following are the primitive equations for a stratified fluid using the Hydrostatic and Boussinesq approximations :

$$\frac{\partial u}{\partial t} + L(u) - fv = -\frac{1}{\rho_0 acos\phi}\frac{\partial p}{\partial \lambda} + F^{\lambda}, \tag{2.1}$$

$$\frac{\partial v}{\partial t} + L(v) + fu = -\frac{1}{\rho_0 a}\frac{\partial p}{\partial \phi} + F^{\phi}, \tag{2.2}$$

$$\frac{\partial p}{\partial z} = -\rho g, \tag{2.3}$$

$$\frac{1}{acos\phi}\left[\frac{\partial u}{\partial \lambda} + \frac{\partial (vcos\phi)}{\partial \phi}\right] + \frac{\partial w}{\partial z} = 0, \tag{2.4}$$

where g is the gravitational acceleration, ρ_0 is the mean density of the ocean, $f = 2\Omega sin\phi$ is the Coriolis parameter and

$$L(q) = \frac{1}{acos\phi}\left[\frac{\partial u(q)}{\partial \lambda} + \frac{\partial vcos\phi(q)}{\partial \phi}\right] + \frac{\partial w}{\partial z}$$

is an advection operator. The terms F^{λ} and F^{ϕ} include metric and viscous terms that are treated explicitly in the discretisation of the horizontal momentum equations. The system of primitive equations is completed by the inclusion of equations for temperature and salinity transport, and the equation of state :

$$\frac{\partial T}{\partial t} + L(T) = D(T), \tag{2.5}$$

$$\frac{\partial S}{\partial t} + L(S) = D(S), \tag{2.6}$$

$$\rho = \rho(T, S, z), \tag{2.7}$$

where $D()$ is a diffusion operator. This system of equations is to be solved with appropriate boundary and initial conditions. The main difference between the two formulations, rigid-lid and free surface, are the boundary conditions for the vertical velocity at the surface. We summarise the formulations as described by Dukowicz and Smith ([23], [24]) in the next two sections.

2.2.1 Rigid-lid formulation

In the rigid-lid formulation the boundary conditions for the vertical velocity are derived from the rigid-lid approximation at the surface, and the requirement that the flow follows the topography at the ocean floor :

$$w = 0 \mid_{z=0}, \tag{2.8}$$

$$w = -\frac{1}{acos\phi}\left(u\frac{\partial H}{\partial \lambda} + vcos\phi\frac{\partial H}{\partial \phi} \right) \mid_{z=-H(\lambda,\phi)}, \tag{2.9}$$

where $H(\lambda, \phi)$ is the (positive) ocean depth.

Decomposing the horizontal velocity field into a baroclinic (internal) and barotropic (external) mode we obtain:

$$(u, v) = (u', v') + (\bar{u}, \bar{v}), \tag{2.10}$$

where

$$\begin{aligned} \bar{u} &= \tfrac{1}{H}\int_{-H}^{0} udz, \\ \bar{v} &= \tfrac{1}{H}\int_{-H}^{0} vdz, \end{aligned} \tag{2.11}$$

are the barotropic (vertically averaged) velocities, and (u', v') are the baroclinic (deviations from vertically averaged) velocities. By integrating (2.3) the pressure may also be decomposed into the surface pressure, p_s (corresponding to the pressure at the rigid lid boundary) and a hydrostatic pressure, p_h where

$$p_h = \int_{z_h}^{0} \rho(z') g dz'. \tag{2.12}$$

We now take vertical averages of (2.1), (2.2) and (2.4) and combine them with (2.8) and (2.9) to obtain

$$\frac{\partial \bar{u}}{\partial t} - f\bar{v} = -\frac{1}{\rho_0 acos\phi}\frac{\partial \overline{(p_s + p_h)}}{\partial \lambda} + \tilde{F^{\lambda}}, \tag{2.13}$$

$$\frac{\partial \bar{v}}{\partial t} - f\bar{u} = -\frac{1}{\rho_0 a}\frac{\partial \overline{(p_s + p_h)}}{\partial \lambda} + \tilde{F^{\phi}}, \tag{2.14}$$

$$\frac{1}{acos\phi}\left[\frac{\partial H\bar{u}}{\partial \lambda} + \frac{\partial (H\bar{v}cos\phi)}{\partial \phi}\right] = 0. \tag{2.15}$$

where

$$\tilde{F^{\lambda}} = -\frac{1}{\rho_0 acos\phi}\frac{\partial \bar{p_h}}{\partial \lambda} + \bar{F^{\lambda}}, \tag{2.16}$$

$$\tilde{F^{\phi}} = -\frac{1}{\rho_0 a}\frac{\partial \bar{p_h}}{\partial \phi} + \bar{F^{\phi}}. \tag{2.17}$$

We now summarise the elimination of the surface pressure p_s by the introduction of a volume transport streamfunction, ψ. We discretise the partial derivatives which involve time in (2.13) - (2.15), using a Leapfrog scheme, and use vector notation to rewrite them in the following form:

$$(1 - \tau\alpha f\mathbf{k}\times)\bar{\mathbf{u}}^{\mathbf{n+1}} + \frac{\tau}{\rho_0}\nabla p_s = \mathbf{s}, \tag{2.18}$$

$$\nabla \cdot H\bar{\mathbf{u}}^{\mathbf{n+1}} = 0, \tag{2.19}$$

where

$$\begin{aligned}
\mathbf{\bar{u}^{n+1}} &= \begin{pmatrix} \bar{u}^{n+1} \\ \bar{v}^{n+1} \end{pmatrix}, \\
\nabla &= \tfrac{1}{acos\phi} \begin{pmatrix} \frac{\partial}{\partial \lambda} \\ cos\phi \frac{\partial}{\partial \phi} \end{pmatrix}, \\
\nabla \cdot &= \tfrac{1}{acos\phi} \left(\tfrac{\partial}{\partial \lambda}, \tfrac{\partial cos\phi}{\partial \phi} \right), \\
\mathbf{s} &= \left[1 + \tau(1-\alpha) f \mathbf{k} \times\right] \mathbf{\bar{u}^{n-1}} + \tau \begin{pmatrix} -\frac{1}{\rho_0 acos\phi} \frac{\partial \bar{p_h}}{\partial \lambda} + \tilde{F^\lambda} \\ -\frac{1}{\rho_0 a} \frac{\partial \bar{p_h}}{\partial \lambda} + \tilde{F^\phi} \end{pmatrix}.
\end{aligned} \tag{2.20}$$

Here $\mathbf{k}$ is a unit vector in the z direction, $\tau = 2\delta t$ (where δt is the size of a timestep), α is a parameter which is used to vary the time-centering of the Coriolis term, and $\mathbf{s}$ is a source function containing information from a previous timestep.

The velocity may be written in terms of the volume transport streamfunction as

$$\mathbf{\bar{u}}^{n+1} = \frac{1}{H} \nabla \psi \times \mathbf{k} = \frac{1}{Hacos\phi} \begin{pmatrix} -cos\phi \frac{\partial \psi}{\partial \phi} \\ \frac{\partial \psi}{\partial \lambda} \end{pmatrix}. \tag{2.21}$$

Using the definition of the curl operator $\nabla \times$ we may eliminate the surface pressure, p_s from (2.18). This gives the following equation for the streamfunction, ψ :

$$\nabla \times \left[(1 - \tau \alpha f \mathbf{k} \times) \left(\frac{1}{H} \nabla \psi \times \mathbf{k} \right) \right] = \nabla \times \mathbf{s}. \tag{2.22}$$

From (2.21) we observe that the barotropic velocity component normal to the boundaries is proportional to the tangential derivative of the streamfunction at the boundaries. The normal component of velocity is required by the boundary condition to be zero. Therefore the streamfunction is required to be spatially constant along all coastlines.

The elliptic equations for the rigid-lid formulation are solved iteratively at every time step in terms of the difference in streamfunction, ψ, between two consecutive timesteps, $n+1$ and n, of the overall rigid-lid ocean model. We denote ψ' to be the change in ψ between timesteps. The equation to be solved is then given by

$$\frac{1}{acos\phi}\left[\frac{\partial}{\partial\lambda}\left(\frac{1}{Hacos\phi}\frac{\partial\psi'}{\partial\lambda}\right)+\frac{\partial}{\partial\phi}\left(\frac{cos\phi}{Ha}\frac{\partial\psi'}{\partial\phi}\right)\right]=\mathbf{S}(\lambda,\phi). \tag{2.23}$$

where $\mathbf{S}$ may be regarded as a source term, containing information from previous timesteps and forcing terms.

2.2.2 Free surface formulation

In the free surface formulation Dukowicz [24] again decomposes the pressure into the surface pressure and a hydrostatic pressure (after integrating the hydrostatic equation (2.3)). As before we have

$$p_h=\int_z^0\rho(\zeta)gd\zeta, \tag{2.24}$$

giving the hydrostatic pressure. This time, however, we have the surface pressure, p_s given by

$$\begin{aligned} p_s &= \textstyle\int_o^\eta \rho(\zeta)gd\zeta \\ &\approx \rho_0 g\eta, \end{aligned} \tag{2.25}$$

where η is the height of the free surface above mean sea level. The equations for a set of variable depth horizontal layers are then derived and applied to a model with a variable top layer only. In this analysis the constant ρ_0 is incorporated into the terms for pressure or density. The resulting layer equations are given by

$$\frac{\partial u_k}{\partial t}+L^*(u_k)-fv_k=-\frac{1}{acos\phi}\frac{\partial(p_s+p_{h,k})}{\partial\lambda}+F_k^\lambda, \tag{2.26}$$

$$\frac{\partial v_k}{\partial t} + L^*(v_k) + fu_k = -\frac{1}{a}\frac{\partial(p_s + p_{h,k})}{\partial\phi} + F_k^\phi, \tag{2.27}$$

where k is the index label for the vertical layers (top layer at $k = 1$) and L^* is the free-surface advection operator. In the top level L^* is given by

$$L^*(q_1) = \frac{q_1}{h_1}\frac{\partial\eta}{\partial t} + \frac{1}{acos\phi}\left[\frac{\partial(u_1 q_1)}{\partial\lambda} + \frac{\partial(v_1 q_1 cos\phi)}{\partial\phi}\right] - \frac{w_{\frac{3}{2}} q_{\frac{3}{2}}}{h_1}, \tag{2.28}$$

and at all other levels it is given by

$$L^*(q_k) = \frac{q_k}{h_k}\frac{\partial\eta}{\partial t} + \frac{1}{acos\phi}\left[\frac{\partial(u_k q_k)}{\partial\lambda} + \frac{\partial(v_k q_k cos\phi)}{\partial\phi}\right] + \frac{w_{k-\frac{1}{2}} q_{k-\frac{1}{2}} - w_{k+\frac{1}{2}} q_{k+\frac{1}{2}}}{h_k}, \tag{2.29}$$

where h_k and q_k are the constant thickness and an advected quantity (u, v, T or S) respectively at layer k. Note the difference between $L^*()$ and $L()$ is due to the extra free surface term on the right hand side of (2.28).

The resulting barotropic, or vertically averaged equations are given by

$$\begin{aligned} &\tfrac{\partial\bar{u}}{\partial t} - f\bar{v} = -g\tfrac{1}{acos\phi}\tfrac{\partial\eta}{\partial\lambda} + G^\lambda, \\ &\tfrac{\partial\bar{v}}{\partial t} + f\bar{u} = -g\tfrac{1}{a}\tfrac{\partial\eta}{\partial\phi} + G^\phi, \\ &\tfrac{\partial\eta}{\partial t} + \tfrac{1}{acos\phi}\left[\tfrac{\partial H\bar{u}}{\partial\lambda} + \tfrac{\partial H\bar{v}cos\phi}{\partial\phi}\right] = 0, \end{aligned} \tag{2.30}$$

where $H = H(\lambda, \phi)$ is the total depth of the ocean, and (u, v) are the barotropic velocity components. The terms G^λ and G^ϕ represent baroclinic forcing. The barotropic equations differ from their rigid-lid counterparts by the inclusion of a term involving the time derivative of the surface elevation, η, in the continuity equation. This equation accounts for the change in volume due to a change in the surface elevation, and for the associated change in the surface pressure. We consider only the barotropic velocities from now on and drop the bar superscript over the (u, v).

Dukowicz [24] considered the following general time discretisation of equation (2.30) :

$$\begin{aligned}
&\tfrac{u^{n+1}-u^{n-1}}{2\delta t} - fv^{\alpha'} = -g\tfrac{1}{acos\phi}\tfrac{\partial \eta^{\alpha}}{\partial \lambda} + G^{\lambda,n},\\
&\tfrac{v^{n+1}-v^{n-1}}{2\delta t} + fu^{\alpha'} = -g\tfrac{1}{a}\tfrac{\partial \eta^{\alpha}}{\partial \phi} + G^{\phi,n},\\
&\tfrac{\eta^{n+1}-\eta^{n}}{\delta t} + \tfrac{1}{acos\phi}\left[\tfrac{\partial H u^{\theta}}{\partial \lambda} + \tfrac{\partial H v^{\theta} cos\phi}{\partial \phi}\right] = 0,
\end{aligned} \tag{2.31}$$

where

$$\begin{aligned}
u^{\alpha'} &= \alpha' u^{n+1} + (1-\alpha'-\gamma')u^{n} + \gamma' u^{n-1},\\
v^{\alpha'} &= \alpha' v^{n+1} + (1-\alpha'-\gamma')v^{n} + \gamma' v^{n-1},\\
\eta^{\alpha} &= \alpha \eta^{n+1} + (1-\alpha-\gamma)\eta^{n} + \gamma \eta^{n-1},\\
u^{\theta} &= \theta u^{n+1} + (1-\theta)u^{n}\\
v^{\theta} &= \theta v^{n+1} + (1-\theta)v^{n}.
\end{aligned} \tag{2.32}$$

δt is the time step, n is the current time level, and α, α', γ, γ' and θ are coefficients used to parameterise the time centering of the pressure gradient, Coriolis, and divergence terms. In general the parameters

$$\gamma = \gamma' = \alpha = \alpha' = \frac{1}{2}, \tag{2.33}$$

are taken in the momentum equations to centre the equations in time, and eliminate temporal truncation errors in the leading-order geostrophic balance. This increases accuracy and reduces damping of the physical modes. Trapezoidal discretisation is taken for the continuity equation to reduce the number of computational modes. If leapfrog differencing had been chosen for this equation as well, then the solutions on alternate time steps would completely decouple, and for each of the three physical modes (one Rossby wave and two gravity waves) there would be a computational mode, resulting in three physical and three computational modes. By using a two time-level discretisation one of these computational modes is eliminated. The remaining two computational modes are a divergence oscillation associated with the gravity waves, and a vorticity oscillation associated with the Rossby wave.

The gravity wave mode is a more serious problem since high-frequency instabilities tend to grow more quickly. This mode is damped most effectively by choosing $\theta = 1$.

Eliminating u^{n+1} and v^{n+1} in (2.31) we obtain an implicit equation for η^{n+1}. The details for this calculation may be found in Appendix A. In the free surface solver this implicit equation is solved by calculating η' which represents the change in the free surface height η between two consecutive timesteps of the overall ocean model(namely $\eta' = \eta(t_{n+1}) - \eta(t_n)$ where $\eta(t_n)$ is known). This is done using the following equation:

$$\frac{1}{acos\phi}\left[\frac{\partial}{\partial\lambda}\left(\frac{H}{acos\phi}\frac{\partial\eta'}{\partial\lambda}\right)+\frac{\partial}{\partial\phi}\left(\frac{Hcos\phi}{a}\frac{\partial\eta'}{\partial\phi}\right)\right]-\beta\eta' = \mathbf{S}(\lambda,\phi). \tag{2.34}$$

The term $\mathbf{S}$ is again a source term containing information from previous timesteps and forcing terms. The Helmholtz parameter β is given by

$$\beta = \frac{1}{2\alpha\theta g\tau^2}, \tag{2.35}$$

where τ is the fixed timestep.

One advantage of this free surface formulation is that the elliptic equation for the surface pressure (and hence the surface height) involves local boundary conditions (discussed in the next section) and allows the use of as many islands as are consistent with the horizontal resolution without computational penalty. Another distinct advantage of this free surface formulation over the rigid lid formulation is that the elliptic operator involves $\nabla(H)$ rather than $\nabla(\frac{1}{H})$. This means that the operator is much less sensitive to sharp changes in the ocean depth (particularly in shallow ocean areas where H is small) and hence avoids the need for any topographic smoothing. Finally large scale Rossby waves are not distorted in this formulation.

2.2.3 Grid staggering of variables

Arakawa and Lamb [4] investigated the dispersion errors arising from the finite difference approximation of the shallow-water equations with regards to simulating geostrophic adjustment (where geostrophic balance is found by the dispersion of inertia-gravity waves). This was done on five different grids which have come to be known as the Arakawa $A-E$ grids. These grids arise from the possible arrangement of the dependent variables of the primitive equations (u, v and η (or ψ)). In this section we examine the dynamical advantages and disadvantages of these grids and highlight the reasons for the extensive use of two of them (B and C) in modern ocean modelling. We summarise the excellent description of this area given by Randall [64] and referred to by Bell [8] and Kantha et al [44] among others. Figures 2.2 - 2.7 show the distribution of the variables in the different grids. Note that the E grid is a 45^o rotation of the B grid. The stepsize is assumed to be the same in both directions with $\delta\lambda = \delta\phi = h$.

An important length scale, when discussing these grids, is the Rossby Radius of Deformation. This is given by

$$r_D = \frac{c}{f},$$

where $c^2 = gH$ for external waves. The ratio of this length scale to the grid size is also a vital issue.

One of the difficulties of choosing a numerical scheme in these cases is ensuring that spatial gradients involved in the primitive equations can be calculated at a given grid resolution. When choosing grids it is important, where possible, to avoid having to calculate tendency terms by averaging values from the surrounding grid points.

From appearance the A-grid seems the simplest. It is unstaggered and allows the Coriolis terms of the momentum equations to be easily calculated (as u and v are defined at the same points). However the pressure gradient terms in the momentum equations and the divergence terms in the continuity equation require averaging. The averaging process actually alters the pressure gradient (since the λ derivative is averaged in the λ direction and the same with the ϕ derivative) which is vital when calculating the geostrophic adjustment. Because of this the solutions are very noisy and require smoothing. As a result the A-grid is hardly ever used in modern ocean models.

As with the A-grid, the Coriolis terms arising from the momentum equations are readily calculated on the B-grid. On the other hand the pressure gradient and divergence terms must still be averaged. However on the B-grid the calculation of the pressure gradient in one direction involves averaging in the other direction and therefore the averaging does not alter solutions in that direction. Hence Arakawa and Lamb [4] concluded that the B-grid is useful for simulating geostrophic adjustment.

On the C-grid the pressure gradient and convergence terms are easily calculated since the η (or ψ) values are defined east/west of u points and north/south of v points as required. However, since the u and v values are defined at different points in this grid averaging is required to calculate the Coriolis values in the momentum equations. It follows that waves for which the Coriolis force is negligible (such as small-scale inertia-gravity waves) are well resolved by the grid. Further this means that if the grid resolution is small enough that the smallest waves resolved on the grid are insensitive to the Coriolis force, then the grid will perform well.

The D-grid permits a ready evaluation of the geostrophic wind which is a clear benefit to geostrophic adjustment models such as that considered by [4]. However it requires averaging to calculate the Coriolis, pressure gradient and divergence terms and as a result is not often used in modern ocean models.

The E-grid appears to be a useful compromise of all the grids as it allows the calculation of the Coriolis, pressure gradient and divergence terms without averaging. However there is a problem when considering solutions that are uniform in one direction. In these (not unlikely) 1D cases the grid is effectively an A-grid with all its inherent problems.

Most modern ocean models use either the Arakawa B or Arakawa C grid to stagger the variables. We have noted that the C grid performs well if the resolution of the model is smaller than the Rossby radius of deformation, a. In the early days of numerical ocean models the resolutions available were too coarse to resolve all but the first internal Rossby radius. Hence Bryan's initial model was formulated on a B grid. As we shall see in Section 2.4 many models in use today still use a B grid. However, C grids are increasingly being used now that finer grid resolutions are attainable. Both of the MO ocean model formulations use a B grid. In the case of the free surface formulation implicit boundary conditions are used at island boundaries which are described in the Section 2.2. Also the combination of the B-grid and nine-point operator used in the free-surface formulation cause a particular form of computational noise which is described in Section 2.3.3.

For completeness we have included the Z grid advocated by Randall [64]. This uses divergence and vorticity as the solution variables and as we can deduce from Figure 2.6 requires no averaging to obtain those variables. Averaging would, however, be needed to calculate the barotropic velocities we require.

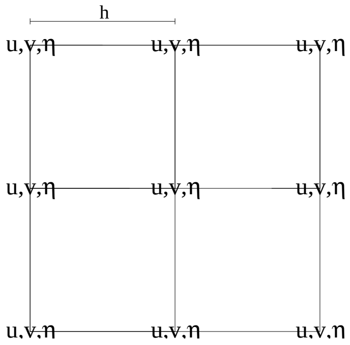

Figure 2.1: Arakawa A Grid

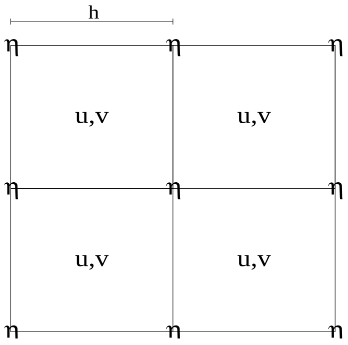

Figure 2.2: Arakawa B Grid

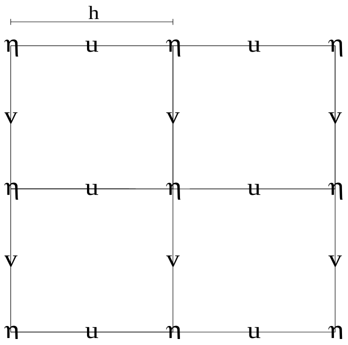

Figure 2.3: Arakawa C Grid

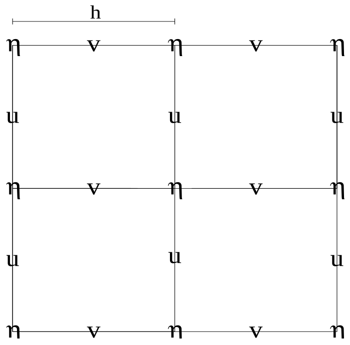

Figure 2.4: Arakawa D Grid

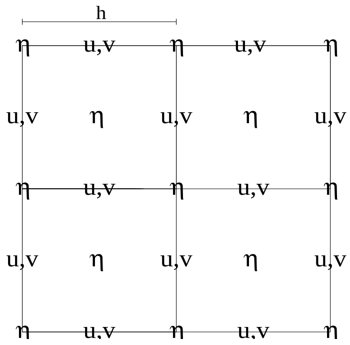

Figure 2.5: Arakawa E Grid

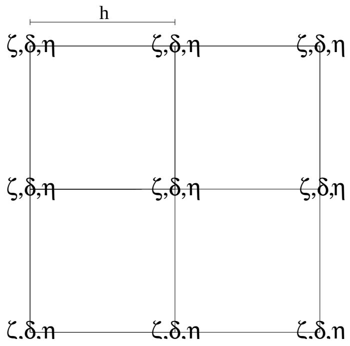

Figure 2.6: Randalls Z Grid

2.3 Spatial discretisations

Throughout this work we will use finite-differences to discretise our model domain. Finite-differences are almost exclusively used in barotropic ocean modelling to generate the discrete approximations to the elliptic model equations (as we shall discuss in Section 2.5). The MO rigid-lid formulation uses a standard five-point discretisation scheme ([6],[38],[40],[75]). The MO free-surface formulation uses a nine-point discretisation scheme. A five-point scheme cannot be used with this formulation for reasons of energy consistency (discussed in detail in Dukowicz and Smith [23]). The nine-point scheme does suffer, though, from a particular type of computational noise which we discuss in Section 2.3.3. Both types of scheme lead to matrix equations with particular properties that may be exploited in order to solve them. The particular properties will also depend on the exact form of the equation being solved, the choice of boundary conditions for the domain, and the ordering of the grid-points in the mesh. In this study we consider two of the most commonly used orderings : the natural and red-black orderings. A detailed study of the use of these, and many other grid point orderings, may be found in Duff and Meurant [22]. We mostly use the natural ordering which is normally used in the free-surface formulation. The rigid-lid formulation typically employs the red-black ordering. To illustrate the orderings consider the Figures 2.7 and 2.8 which show the numbering of the grid points for a simple 5×5 grid. The forms and properties of the matrices generated by these orderings are discussed generally in Chapter 3 and for the particular cases considered in Chapters 4 to 7. These will impact on the choice of numerical methods used to solve the matrix equations.

In the rest of this section we briefly describe some common issues arising from the use of the nine-point discretisation scheme in the free-surface

21	22	23	24	25
16	17	18	19	20
11	12	13	14	15
6	7	8	9	10
1	2	3	4	5

Figure 2.7: Natural ordering

11	24	12	25	13
21	9	22	10	23
6	19	7	20	8
16	4	17	5	18
1	14	2	15	3

Figure 2.8: Red-Black ordering

models. We will examine the use, with the nine-point scheme, of implicit boundary conditions at the island boundaries as referred to by Dukowicz et al [24]. We also highlight issues of a particular type of computational noise which commonly arises when using the nine-point scheme in the free-surface formulation. The filter typically used to deal with this noise is also described.

2.3.1 Nine-point operator

In this section we briefly illustrate how the nine-point discrete operator used in the free-surface formulation is constructed. This will allow us to discuss the implicit boundary conditions in the next section. To illustrate the 9-pt stencil we consider the discretisation scheme about a general solution point $\eta_{i,j}$ (where $\eta_{i,j} = \eta(\lambda_i, \phi_j)$). The general form of the elliptic equation that is solved in the free-surface formulation is of the form $\nabla\cdot(\nabla\eta) - \beta\eta$ (c.f equation (2.34)). The divergence $\nabla\cdot$ is differenced using the four surrounding velocity points $(\eta(i+\frac{1}{2}, j+\frac{1}{2}), \eta_{i-\frac{1}{2},j+\frac{1}{2}}, \eta_{i+\frac{1}{2},j-\frac{1}{2}}, \eta_{i-\frac{1}{2},j-\frac{1}{2}})$, which leaves (∇U) to be resolved at these velocity points. The $\nabla\eta$ is resolved by using the four solution points surrounding each velocity point (i.e $\eta_{i+1,j+1}, \eta_{i-1,j+1}, \eta_{i+1,j-1}, \eta_{i-1,j-1}$) : components of $\nabla\eta$ are differenced and then averaged. For example $\frac{\partial\eta}{\partial\lambda}$ on velocity row $\eta_{i,\frac{3}{2}}$ is found from,

$$\frac{1}{2}\left(\frac{(\eta_{i+1,1} - \eta_{i,1})}{\delta\lambda} + \frac{(\eta_{i+1,2} - \eta_{i,2})}{\delta\lambda}.\right)$$

Repeating this at each velocity point results in the nine-point stencil.

2.3.2 Boundary conditions for islands

Dukowicz [24], when describing the free surface formulation, refers to 'implicit boundary conditions' when discussing island boundary conditions. Note that the free surface height formulation is analogous with an elliptic problem that requires Neumann conditions of

$$\frac{\partial \eta}{\partial \hat{n}} = 0,$$

at island boundaries (where $\hat{n}$ is the direction normal to the boundary). In this section we show that this boundary condition is 'implied' by the setting of the ocean depth, H, to be zero at land points. This means that the condition does not have to be specifically included in the matrix equations. Consequently there is little extra computational expense in using islands with the free-surface formulation, allowing the use of as many as desired.

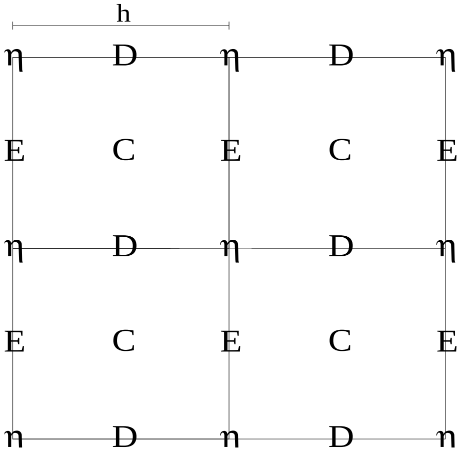

Figure 2.9: Illustration of boundary condition in free surface model

Figure 2.9 shows the arrangement of variables in the computational B-grid. The free surface height, η, is at the η-points shown on the grid ,with the velocity and depth information at the C-points in the middle of each box (grid staggered one half grid spacing in each direction). The generic operator of interest is

$$\frac{\partial}{\partial \lambda}(H\frac{\partial}{\partial \lambda}(\eta)) + \frac{\partial}{\partial \phi}(H\frac{\partial}{\partial \phi}(\eta)).$$

This is constructed at grid point (i,j) from

$$(H\frac{\partial}{\partial\lambda})(\eta_{i+\frac{1}{2},j}) - (H\frac{\partial}{\partial\lambda})(\eta_{i-\frac{1}{2},j}),$$

at the D-points and

$$(H\frac{\partial}{\partial\phi})(\eta_{i,j+\frac{1}{2}}) - (H\frac{\partial}{\partial\phi})(\eta_{i,j-\frac{1}{2}}),$$

at the E-points.

It suffices to simply consider one of these, so take $H\frac{\partial\eta}{\partial\lambda}(i+\frac{1}{2},j)$ to get the nine-point operator. This is taken to be the average of the operator at the two C-points, $C(i+\frac{1}{2},j+\frac{1}{2})$ and $C(i+\frac{1}{2},j-\frac{1}{2})$ i.e.

$$(H\frac{\partial}{\partial\lambda})(\eta_{i+\frac{1}{2},j+\frac{1}{2}}) = \frac{1}{2}((H\frac{\partial}{\partial\lambda})(\eta_{i+\frac{1}{2},j+\frac{1}{2}}) + (H\frac{\partial}{\partial\lambda})(\eta_{i+\frac{1}{2},j-\frac{1}{2}})).$$

On the right hand side of this equation, the H value is defined at that point, whereas the gradient of η is now taken to be the average of the values above and below, so e.g.

$$\begin{aligned}&(H\tfrac{\partial}{\partial\lambda})(\eta_{i+\frac{1}{2},j}) =\\ &H(i+\tfrac{1}{2},j+\tfrac{1}{2})\tfrac{1}{2}(\eta_{i+1,j+1} - \eta(i,j+1) + \eta(i+1,j) - \eta_{i,j}),\end{aligned}$$

so we can now see that the matrix coefficients are simply geometric products always multiplied by the H value on the velocity grid points. Note that

$$H(C-point) = min(4\ \ surrounding\ \ H(\eta - points)).$$

So if any C-point is on the boundary of a η point which is land, then $H(C-point)$ will be zero.

For example suppose $(i+1,j+1)$ is a land point. Then $H(i+\frac{1}{2},j+\frac{1}{2}) = 0$. This implies that the local normal components at $(i+\frac{1}{2},j+\frac{1}{2})$ will always contribute 0 to the matrix, thereby implying local normal gradients of zero at that point. NB This does not imply that the finite normal gradient at

$D(i+\frac{1}{2},j)$ contributes zero; this will of course contribute a finite amount via the averaging to obtain the contribution at $C(i+\frac{1}{2},j-\frac{1}{2})$. In this way the imposition of normal components of η across land interfaces equates to zero, and the boundary condition has been implicitly applied. Note that it is possible to substitute the 'correct' boundary conditions into the equations : upon eliminating the land values for the free surface, the land values of the source terms, $S(\lambda, \phi)$, are also eliminated.

2.3.3 Chequerboard null Space

Using a nine-point operator with the B-grid in the Poisson equation (rigid-lid) case generates a non-empty null space (as described by Dukowicz et al [23], [24]). This corresponds to two null eigenvectors which are not removed by the discrete operator : a global chequerboard (+/-) field and a constant field. In one sense the free surface nine-point operator on a B-grid does not have this null space, because the extra diagonal term moves the eigenvalues of the null space fields away from zero, hence damping the constant and chequerboard fields. However, if the solution to the Helmholtz equation is in near steady state, this extra term cancels with the corresponding term on the right hand side yielding a Poisson type problem with null space issues we have highlighted. This commonly occurs in isolated coastal regions where the solution is only weakly coupled to the interior. The chequerboard noise only appears in the surface height fields. It does not appear in the velocity fields as the chequerboard mode does not appear in the gradient operators which are used in the barotropic momentum equation.

On a C-grid the global chequerboard is not a null eigenvector and hence grid-scale chequerboard noise is damped. The C-grid is therefore the better option in order to avoid chequerboarding. It does, however, have other disadvantages, as noted in Section 2.2.3, which lead to preference for keeping the B grid in the MO ocean models.

2.3.4 Killworth Delplus-Delcross filter

In this section we summarise the Killworth Delplus-Delcross filter, used to address the chequerboard null mode described in Section 2.3.3, as discussed in Killworth et al [46]. We briefly describe the form of the filter and how it deals with boundary effects. As noted by Killworth [46] solutions to the chequerboarding problem are standard in the Atmospheric literature (e.g. Jancic [43]). Solution methods typically evaluate the divergence terms with components from both chequerboard grids, thus re-coupling the grids and smoothing out the (+/−) mode [46]. The Killworth delplus-delcross (∇_+-$\nabla_\times$) filter, which we summarise here, uses the difference between the standard five-point ∇^2 operator, delplus :

$$\nabla_+^2(i,j) = \frac{-\eta_{i+1j} + 2\eta_{ij} - \eta_{i-1j}}{\delta\lambda^2} + \frac{-\eta_{ij+1} + 2\eta_{ij} - \eta_{ij-1}}{\delta\phi^2}, \tag{2.36}$$

and the five-point operator acting along the NW-SE and NE-SW axes, delcross :

$$\nabla_\times^2(i,j) = \frac{-\eta_{i+1j+1} + 2\eta_{ij} - \eta_{i-1j-1}}{\delta\lambda^2 + \delta\phi^2} + \frac{-\eta_{i-1j+1} + 2\eta_{ij} - \eta_{i+1j-1}}{\delta\lambda^2 + \delta\phi^2}. \tag{2.37}$$

Due to the spherical grid the difference between the delplus and delcross operators is multiplied by a term of the form

$$\frac{gH_{max}\delta t}{a^2 cos\phi\delta\lambda\delta\phi}, \tag{2.38}$$

where H_{max} is the largest depth in the model basin. The form of the multiplication was carefully chosen (using experimentation), by Killworth et al [46], to maintain mass conservation, in order that the integrated effects of the filter are zero. The filter in (2.38) does this as mass conservation involves a local metric of precisely the denominator of (2.38) ([46]).

There are also boundary effects to consider. The main problem is that, with a standard Laplacian diffusion operator, η values outside a boundary (i.e. on land) would be set such that $\frac{\partial \eta}{\partial \hat{n}}$ vanished (where $\hat{n}$ is a co-ordinate normal to the land boundary). However, land solution points could be involved in evaluations of (2.36) and (2.37) depending on the shape of the land boundary. It is required for mass conservation that all land solution point contributions to (2.36) and (2.37) must sum to zero. Also (2.38) must still be a smoothing operator everywhere, with all Fourier components of perturbations being damped near land boundaries as well as in the interior of the ocean domain. The solution Killworth et al [46] implemented was to define the value of a land η point as the average of any surrounding ocean solution points which access that point, in order to calculate (2.36). If (2.37) needs to use a land point, the contribution of that pair is neglected.

2.4 Discretising polar regions

Polar regions, in particular the Arctic Ocean, play a key role in the ocean circulation system via sea ice and deep water formation. Unfortunately, the traditional latitude-longitude coordinate system used to solve the Primitive equations in OGCMs possesses singularities at the North and South Poles. The problem of the South Pole singularity is resolved by the fact that it is on Antarctica and hence not part of the ocean domain. The North Pole singularity does not fall on land, however. This impacts on the computational stability of finite difference schemes by drastically reducing permitted time step lengths. Various solutions exist for countering this problem. One such method is the use of a North Polar island [65]. This is currently used in the rigid-lid ocean model at the MO. The topmost ψ row is taken to be land. At the topmost velocity row (half a grid spacing away latitudinally to the south) the prescribed Dirichlet boundary conditions of zero flow are assumed to hold.

One alternative to the Polar island is the use of a North Polar point, as discussed by Rickard and Cresswell [65]. This method locates a solution (ψ or η) point at the Pole itself. The solution values at the polar point are updated by calculating the fluxes on the velocity points to the south. The updates to the value of the solution at the pole from these fluxes are then averaged to obtain the unique new value of the solution point at the pole. Obtaining the fluxes, in a consistent manner, is therefore vital to this method. This scheme is currently used in the free surface ocean model of the MO (amongst others).

Another approach is that suggested by Madec and Imbard [54]. The coordinates of the North Pole are shifted onto land (Siberia is often used) and an orthogonal curvilinear ocean mesh is used to discretise the North Hemisphere region. This has the obvious benefit of removing the North Polar singularity entirely. However this does have the drawback of grid non-uniformity as well as the extra expense of altering Topography and Bathymetry settings in the code.

2.5 Ocean modelling today

In this section we will briefly summarise the models currently being used by ocean modelling groups worldwide. In particular we will highlight what formulations the ocean groups are using and which co-ordinate systems and

Group	Full name/location
CERFACS	European Centre for Research and Advanced Training in Scientific Computation, France
GFDL	Geophysical Fluid Laboratory, USA
IPSL	Institut Pierre Simon Laplace, France
LANL	Los Alamos National Laboratory, USA
MIT	Massachusetts Institute of Technology, USA
MPI	Max Planck Institute, Germany
NCAR	National Center for Atmospheric Research, USA
NASA	National Aeronautics and Space Administration
NRL	US Naval Research Laboratory
SOES	School of Ocean and Earth Science, UK
UM	University of Miami, USA
UP	University of Princeton, USA

Table 2.1: Ocean groups worldwide

grids they are using them with. There are ocean models that use alternative methods for discretising the Primitive equations such as finite element or spectral methods, although these are seldom used operationally. Table 2.1 is a list of major modelling institutions worldwide.

Table 2.2 gives details on the models and numerics used by the groups listed in Table 2.1. FS and RL refer to free-surface and rigid-lid formulations respectively. All of the groups/models listed use a Primitive equation model with second order finite differences and leap-frog time-stepping in the numerics with the exception of MITgcm. This uses a finite-volume, non-hydrostatic formulation with horizontal orthogonal curvilinear co-ordinates. The OPA, MPI-OM, POM and ROMS models also use this co-ordinate sys-

Group	Model	RL/FS	Grid	Vert.Co-ord.
CERFACS	OPA [53]	FS	C	z
GFDL	MOM [63]	Exp FS	B	z
IPSL	OPA [53]	FS	C	z
MIT	MITgcm [2]	-	-	z
MO	MOM [63]	RL(FS)	B	z
MPI	MPI-OM [57]	FS	C	z
NASA	ROMS [71]	FS	C	σ
NCAR	CSM [35]	RL	B	z
SOES	OCCAM [78]	Exp FS	B	z
UM,LANL,NRL	HYCOM [42]	FS	C	hybrid
UP	POM [30]	FS	C	σ

Table 2.2: Details of ocean groups worldwide and details of the models they use for solving the barotropic problem

tem which overcomes the North Pole singularity by using two poles in the Northern Hemisphere which are placed over land (One over North America and the other over Siberia). A slightly different approach is used by OCCAM. It uses a regular latitude-longitude grid for the Pacific, Indian and South Atlantic oceans whilst incorporating a rotated latitude-longitude grid in the Arctic and North Atlantic oceans with two poles placed on the equator in the Indian and Pacific. The GFDL explicit free-surface method is based on the work of Killworth et al [46]. It differs from the free-surface method of Dukowicz and Smith [24] by the allowance of external gravity waves. These require the use of a smaller timestep when resolving the linear terms of the barotropic equations.

2.6 Summary

This chapter has provided the motivation for our study of anisotropic elliptic problems by discussing their use in the context of Ocean General Circulation models, and in particular the barotropic problem. We have described some of of the history of OGCMs up to the development of the Bryan-Cox-Semtner model which is used extensively today in various guises. We have focussed on the two formulations of the BCS model most commonly used today : rigid-lid and implicit free surface. We have also summarised the form of the various 'Arakawa' grids used to position the model variables in those formulations and highlighted the reasons for the extensive use of the 'B' and 'C' Arakawa grids. In addition we introduced the spatial discretisation schemes used with the BCS model formulations. The appearance of a chequerboard null mode in the solution with the free-surface nine point operator is highlighted and the use of the Killworth delplus-delcross filter to resolve the problem is discussed. The work is placed in a more global context by the discussion of the characteristics of ocean models used by research and industrial organisations worldwide.

Chapter 3

Numerical methods and preconditioners

3.1 Introduction

In this chapter we provide the theoretical background to the numerical methods that are commonly used in the ocean models for solving the discrete approximations to the elliptic model equations we introduced in Chapter 2. We also provide the theoretical background to extensions we will consider. The discrete approximation of any of the elliptic problems we consider in this study gives rise to a matrix equation of the general form :

$$A\mathbf{U} = \mathbf{b}, \tag{3.1}$$

where the variable $\mathbf{U}$ is a (unknown) column vector of the grid points of the model variable U, and $\mathbf{b}$ is a (known) column vector representing boundary values and source terms. The system matrix A is a real $N \times N$ matrix representing the discretised model equations, where N is the number of grid points in the discrete grid. The size of N is determined by $N = n_\lambda \times n_\phi$ where

n_λ, n_ϕ are the number of grid points in the longitudinal and latitudinal directions respectively. The system matrix A is also square and sparse in general. The exact form of A will depend on the type of elliptic equation being discretised, the boundary conditions that are required, and the ordering of the grid points across the domain of the problem.

There are two main types of methods used to solve a matrix equation of the form (3.1) : direct and iterative. Direct methods are those that would compute the exact answer in a finite number of steps if there was no round off error. Examples include Cholesky Decomposition and Gaussian Elimination (discussion of which may be found in Axelsson [6] and Golub and Van Loan [38]). Iterative methods on the other hand use a repeated application of a simple algorithm, but yield the exact solution only in the limit of a sequence. The latter are more commonly used in recent times as they have advantages in storage and operation, particularly with large sparse matrices. The system matrices we consider in this study are indeed large and sparse, making iterative methods practical to use. Also, as noted by Forsythe and Wasow [33], iterative methods are self-correcting and useful in minimizing problems with round-off errors. We will use only iterative methods in this study. A basic iterative method has the form

$$P\mathbf{U}^{m+1} = (P - A)\mathbf{U}^m + \mathbf{b} \quad m = 0, 1, 2, \cdots, \tag{3.2}$$

where m is the iteration number. An alternative form to (3.2) is

$$\begin{aligned} \mathbf{d}^{m+1} &= -P^{-1}\mathbf{r}^m, \\ \mathbf{U}^{m+1} &= \mathbf{U}^m + \mathbf{d}^{m+1}, \end{aligned} \tag{3.3}$$

where $\mathbf{r}^m$ is the residual given by $\mathbf{r}^m = A\mathbf{U}^m - \mathbf{b}$ and $\mathbf{d}^{m+1}$ is a correction at iteration m. Axelsson [6] shows that the basic iterative method (3.3) may be improved by introducing parameters into the iteration scheme which vary through the iteration process. An iterative method is referred to as stationary if the parameters used remain constant throughout the iterative process and nonstationary if the parameters vary.

The form in (3.3) is the basis for the Preconditioned Conjugate Gradient (PCG) method. The matrix P is often referred to as the preconditioning matrix or preconditioner. Various options exist for preconditioners which will be considered in Section 3.4. The other iterative method we consider in this study is the Chebyshev Semi-Iterative Method. Both this and the PCG method may be thought of as acceleration procedures for preconditioned stationary methods.

In Section 3.2 we give key definitions and theorems which are used to confirm the validity of the numerical methods used in the models. We then discuss the relevant theory for the Chebyshev Semi-Iterative method in Section 3.3. Also in Section 3.3 we describe the theory for the PCG method. Finally in Section 3.4 we discuss the type of preconditioner currently used in the free-surface formulation, and describe alternatives which we intend to test.

3.2 Theorems and definitions

This section highlights the key definitions and theorems required to prove the validity, and assess the convergence speeds, of the iterative methods and preconditioners that we introduce in Sections 3.3 and 3.4. With iterative methods the eigenstructure of the matrices we consider is of vital importance. Therefore we begin with some basic definitions ([75]) which will be used throughout.

3.2.1 General definitions

Definition 3.1 *(Spectral radius) : Let $A \in \mathrm{IR}^{N\times N}$ have eigenvalues $\mu_i, 1 \leq i \leq N$. Then*

$$\rho(A) = max \mid \mu_i \mid, \quad 1 \leq i \leq N,$$

is the spectral radius of the matrix A.

Definition 3.2 *(Reducible) : $A \in \mathrm{IR}^{N\times N}$ is reducible if there exists a $N\times N$ permutation matrix P such that*

$$PAP^T = \begin{pmatrix} A_{1,1} & A_{1,2} \\ 0 & A_{2,2} \end{pmatrix},$$

where $A_{1,1} \in \mathrm{IR}^{r\times r}$ and $A_{2,2} \in \mathrm{IR}^{N-r\times N-r}$ with $A_{1,1}$ and $A_{2,2}$ being submatrices of A. If no such permutation exists then the matrix A is irreducible.

Definition 3.3 *(Positive) : $A \in \mathrm{IR}^{N\times N}$ non-negative, denoted by $A \geq 0$, if $a_{ij} \geq 0 \; \forall \; i,j \in [1,N]$. A is positive if the inequality is strict $\forall \; i,j$.*

Definition 3.4 *(Positive-Definite) : $A \in \mathrm{IR}^{N\times N}$ is positive definite if $x^T Ax > 0$, $x \neq 0, x \in \mathrm{IR}^N$.*

Definition 3.5 *(Cyclic) : Let $A \in \mathrm{IR}^{N\times N}$ be irreducible and let k be the number of eigenvalues of A of magnitude $\rho(A)$. If $k = 1$, then the matrix is primitive. If $k > 1$, then A is cyclic of index k.*

The concept of the irreducibility of a matrix is equivalent to the consideration of the strongly connected directed graph of a matrix :

Definition 3.6 *(Directed graph) : Let $A \in \mathrm{IR}^{N\times N}$ and consider N distinct points P_1, P_2, $\cdots$, P_n in the plane. For every non-zero entry a_{ij} of the matrix A we connect the point P_i to the point P_j by means of a path $\vec{P_iP_j}$ directed from P_i to P_j. In this way every $N \times N$ matrix A can be associated with a finite directed graph $G(A)$.*

Definition 3.7 *(Strongly connected) : A directed path is strongly connected if, for any pair of points P_i and P_j there exists a directed path*

$$P_i\vec{P}_{l1}P_{l1}\vec{P}_{l2}\cdots P_{\overset{\rightarrow}{l_{r-1}}}P_j,$$

connecting P_i to P_j.

Varga [75] links the two concepts via the following theorem :

Theorem 3.1 *([75], p20) An $N \times N$ matrix A is irreducible if and only if its directed graph is strongly connected.*

The concept of diagonally dominant matrices also plays a key role in iterative method theory. These may be defined by ([75]) :

Definition 3.8 *(Diagonally Dominant) : $A \in \mathrm{I\!R}^{N\times N}$ is diagonally dominant if $\mid a_{ii} \mid\geq \sum_{j=1,j\neq i}^{n} \mid a_{ij} \mid, \quad 1 \leq i \leq N$. The matrix A is strictly diagonally dominant if strict inequality holds for all $1 \leq i \leq N$.*

Combining Definition 3.8 with the concept of irreducibility we have :

Definition 3.9 *(Irreducibly Diagonally Dominant) : $A \in \mathrm{I\!R}^{N\times N}$ is irreducibly diagonally dominant if it is irreducible and $\mid a_{ii} \mid\geq \sum_{j=1,j\neq i}^{N} \mid a_{ij} \mid$ $1 \leq i \leq N$, with strict inequality for at least one value of $i \in [1, n]$.*

From the Definitions 3.8 and 3.9 it is possible to show the non-singularity of the matrix A via the following theorem :

Theorem 3.2 *([75], p23) Let $A \in \mathrm{I\!R}^{N\times N}$ be strictly or irreducibly diagonally dominant. Then the matrix A is nonsingular. If all the diagonal entries of A are, in addition, positive real numbers, then the eigenvalues μ_i of A satisfy*

$$Re(\mu_i) > 0, \quad 1 \leq i \leq N,$$

and the matrix A is positive-definite.

Theorem 3.3 *([75], p85) If $A \in \mathrm{I\!R}^{N \times N}$ is irreducibly diagonally dominant with $a_{ij} \leq 0 \quad \forall i \neq j$ and $a_{ii} > 0 \quad \forall 1 \leq i \leq N$ then $A^{-1} > 0$.*

As Axelsson [6] notes, the following two definitions are important as they (particularly the latter) appear frequently in practice. Stieltjes matrices are also important when considering Alternating Direction Implicit (ADI) preconditioned methods (see Section 3.4 for an introduction to these).

Definition 3.10 *(Stieltjes matrix) : $A \in \mathrm{I\!R}^{N \times N}$ with $a_{ij} \leq 0 \quad i \neq j$, is a Stieltjes matrix if A is symmetric and positive definite.*

Definition 3.11 *(M-matrix) : $A \in \mathrm{I\!R}^{N \times N}$ with $a_{ij} \leq 0 \;\; i \neq j$, is an M-matrix if A is nonsingular and $A^{-1} \geq 0$.*

Axelsson [6] further states that it can be shown that even if a matrix is not an M-matrix it can, if it is symmetric positive-definite, be reduced to a Stieltjes matrix by using a method of diagonal compression of reduced positive entries. Further, using these definitions we may obtain the following theorem

Theorem 3.4 *([75], p85) If A is a Stieltjes matrix then it is also an M-matrix. If A is, in addition, irreducible then $A^{-1} > 0$.*

3.2.2 Convergence of basic iterative methods

We now consider the convergence of our basic iterative methods (3.2), and introduce the concept of error modes in relation to eigenvectors, which will be vital in our discussion of how the mesh anisotropy affects the convergence of our methods. The following theorem is vital to the former :

Theorem 3.5 *([6], p163) The sequence of vectors $\mathbf{U^m}$ in (3.2) converges to the solution of $A\mathbf{U} = \mathbf{b}$ for any 'initial guess' vector, $\mathbf{U}^0 \iff \rho(P^{-1}(P - A)) = \rho(I - P^{-1}A) < 1$.*

We will refer to the matrix $G_P = I - P^{-1}A$ as the iteration matrix of the preconditioned method. One of the oldest classical stationary preconditioned iterative methods is the Jacobi method ([75]). This is uses the diagonal elements of A as the preconditioner i.e. $P = D = diag(A)$, where $G_D = I - D^{-1}A$.

Let the error at iteration m be denoted by $\mathbf{e}^m = U - \mathbf{U}^m$. It is well known ([6], [38]) that classical stationary iterative methods yield the following expression for the error at iteration m

$$\mathbf{e}^m = G_P^m \mathbf{e}^0. \tag{3.4}$$

Now assume that the matrix G_P is symmetrizable. This follows if $P^{-1}A$ is symmetrizable which, in turn, may be deduced by showing that P is symmetric positive-definite and that A is symmetric ([6],[38]). The condition that G_P is symmetrizable implies that the eigenvalues of G_P are real and that the matrix possesses a full set of independent eigenvectors. In addition assume that $\rho(G_P) < 1$. Let $\mathbf{w}_i$ for $i = 1, \cdots, N$ be the set of eigenvectors of G_P with corresponding eigenvalues μ_i, and let us expand the initial error $\mathbf{e}^0$ in terms of this basis

$$\mathbf{e}^0 = \sum_{i=1}^{N} \gamma_i \mathbf{w}_i, \tag{3.5}$$

where γ_i are constants. At the m^{th} iteration we obtain

$$\mathbf{e}^m = \sum_{i=1}^{N} \gamma_i G^m \mathbf{w}_i = \sum_{i=1}^{N} \gamma_i \mu_i^m \mathbf{w}_i. \tag{3.6}$$

After m iteration steps the magnitude of the i^{th} mode of initial error has been reduced by a factor of μ_i^m. It is possible to expand the initial error in terms of any orthogonal basis e.g. Fourier modes if applicable. These concepts will be used extensively in Chapters 5 to 7.

3.2.3 Gerschgorin circle theorem

The theorems we have given so far allow us to prove the convergence of the iterative methods we use in this study. We now consider the following theorem which we shall use to obtain information on the rates of convergence of some of our iterative methods.

Theorem 3.6 *(Gerschgorin Circle Theorem) ([75], p17) Let* $A \in \mathbb{R}^{N\times N}$ *and let*

$$R_i = \sum_{j=1, j\neq i}^{N} \mid a_{ij} \mid, \quad 1 \leq i \leq N.$$

Then, all the eigenvalues μ *of* A *lie in the union of the disks*

$$\mid z - a_{ii} \mid \leq R_i \quad 1 \leq \ i \leq N.$$

From this we have the following theorem

Theorem 3.7 *([75], p17) Let* $A \in \mathbb{R}^{N\times N}$ *and let*

$$\nu = max \sum_{j=1}^{N} \mid a_{ij} \mid, \quad 1 \leq i \leq N,$$

then $\rho(A) \leq \nu$*.*

Thus, the maximum of the row sums of the moduli of the entries of the matrix A *gives an upper bound for the spectral radius* $\rho(A)$ *of the matrix* A*.*

Both theorems of this section will be used to derive bounds on the spectral radii and conditioning of our preconditioned matrices.

3.2.4 Block partitioning of A

Typically the system matrices we consider in this study have a particular 'block' structure which it often useful to take advantage of. One of the most classical stationary iterative methods which makes use of this structure is the block Jacobi method ([75]). This uses the diagonal 'blocks' of the system matrix A to form the preconditioned method i.e. $P = D = blockdiag(A)$, with $G_{Block} = I - D^{-1}A$. This will be explained in more detail in Section 3.4.2. What we mean by 'block structure' is highlighted in the following definitions, from Feingold and Varga [32], where we assume that we can write our system matrix A in the following block partitioned form:

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,n_\phi} \\ A_{2,1} & A_{2,2} & \cdots & A_{2,n_\phi} \\ \vdots & & & \vdots \\ A_{n_\phi,1} & A_{n_\phi,2} & \cdots & A_{n_\phi,n_\phi} \end{pmatrix}, \tag{3.7}$$

where the diagonal submatrices, $A_{i,i}$, for $1 \leq i \leq n_\phi$, are square of size $n_\lambda \times n_\lambda$.

Definition 3.12 *(Block diagonally dominant) : Let $A \in \mathbb{R}^{N\times N}$ be partitioned as in (3.7). If the diagonal submatrices $A_{j,j}$ are nonsingular, and if*

$$\left(\| A_{j,j}^{-1} \|\right)^{-1} \geq \sum_{l=1,l\neq j}^{n_\phi} \| A_{j,l} \|, \quad \forall\ 1 \leq j \leq n_\phi, \tag{3.8}$$

then A is block diagonally dominant, relative to the partitioned matrix A. If strict inequality in (3.8) is valid $\forall i$, then A is block strictly diagonally dominant, relative to the partitioned matrix A.

Definition 3.13 *(Block irreducible) : Let $A \in \mathbb{R}^{N\times N}$ be partitioned as in (3.7). A is block irreducible if the $n_\phi \times n_\phi$ matrix $\hat{A} = \{\hat{a_{i,j}} \equiv \| A_{i,j} \|\}$, $1 \leq i,j \leq n_\phi$, is irreducible (i.e. the directed graph of $\hat{A}$ is strongly connected.*

Unless otherwise specified, the norms in this section are all assumed to be the 2-norm. Feingold and Varga [32] also derive the equivalent block theorems for positive-definiteness, non-singularity and bounding of eigenvalues :

Theorem 3.8 *([32]) If the matrix $A \in \mathbb{R}^{N \times N}$, partitioned as in (3.7) is block strictly diagonally dominant, or if A is block irreducible and block diagonally dominant with inequality holding in (3.8) for at least one i, then A is nonsingular.*

Theorem 3.9 *(Block Gerschgorin Circle Theorem) ([32]) For the partitioned matrix A of (3.7), each eigenvalue, μ, of A satisfies*

$$\left(|| \left(A_{j,j} - \mu I_j \right)^{-1} || \right)^{-1} \leq \sum_{i=1, i \neq j}^{n_\phi} || A_{j,i} ||,$$

for at least one j, $1 \leq j \leq n_\phi$.

Theorem 3.10 *([32]) Let $A \in \mathbb{R}^{N \times N}$ be partitioned as in (3.7) and let A be block strictly diagonally dominant (or block irreducible and block diagonally dominant with strict inequality in (3.8) for at least one i). Further assume that each submatrix $A_{i,i}$ is an M-matrix. If μ is any eigenvalue of A, then*

$$Re(\mu) > 0.$$

3.3 Nonstationary iterative methods

We now introduce, in the next two sections, two iterative methods used in ocean models : Preconditioned Conjugate Gradient and firstly the Chebyshev Semi-Iterative method. These are nonstationary methods which act as acceleration procedures for stationary iterative methods.

3.3.1 Chebyshev semi-iterative method

The Chebyshev Semi-Iterative method is used in the rigid-lid formulation to solve a Poisson equation (2.23), for the change in streamfunction, ψ'. The theory for this method is discussed by Axelsson [6] and Varga [75], and is summarised here. The classical Chebyshev Semi-Iterative method is given by

$$\mathbf{U}^{m+1} = \omega_{m+1} \left\{ G_P \mathbf{U}^m + \mathbf{b} - \mathbf{U}^{m-1} \right\} + y^{m-1}, \quad m \geq 0, \tag{3.9}$$

where G_P is the preconditioned iteration matrix for the method being accelerated. $\rho(G_P) < 1$ is required for the method to converge. For $m = 0$ (3.9) reduces to $\mathbf{U}^1 = \mathbf{U}^0 + \mathbf{b}$. The parameter ω_{m+1} is determined by the following algorithm :

$$\begin{aligned} &\omega_1 = 1, \\ &\omega_2 = \tfrac{1}{1-\frac{1}{2}\rho(G_P)^2}, \\ &\omega_{m+1} = \tfrac{1}{1-\frac{1}{4}\rho(G_P)^2\omega_m} \quad m \geq 2, \end{aligned}$$

where $\rho(G_P)$ is the spectral radius of the iterative method.

As Axelsson [6] notes, the Chebyshev method may be applied to a preconditioned iteration method if the preconditioned system matrix, $P^{-1}A$, has positive eigenvalues. It is further noted by Axelsson [6] that the method, therefore, is applicable if both P and A are symmetric positive definite. Finally it is remarked by Axelsson [6] that the number of iterations of the Chebyshev method varies at most as the square root of the condition number, κ, of the preconditioned system, $P^{-1}A$ where

$$\kappa_p(P^{-1}A) = \| P^{-1}A \|_p \cdot \| (P^{-1}A)^{-1} \|_p, \tag{3.10}$$

with p representing the norm used. In practice the 2 norm is typically used for theoretical work whilst the ∞ norm is normally used for numerical experimentation. With the 2 norm we have the useful results

$$\begin{aligned} \rho(P^{-1}A) &= \| P^{-1}A \|_2, \\ \frac{1}{\mu_{min}}(P^{-1}A) &= \| (P^{-1}A)^{-1} \|_2, \end{aligned}$$

$$\Longrightarrow \kappa_2(P^{-1}A) \;=\; || P^{-1}A ||_2 \cdot || (P^{-1}A)^{-1} ||_2 = \frac{\rho(P^{-1}A)}{\mu_{min}(P^{-1}A)}. \tag{3.11}$$

where μ_{min} is the smallest eigenvalue in magnitude.

3.3.2 Preconditioned conjugate gradient method

Gradient methods are a group of iterative methods that are commonly considered as acceleration schemes for linear, stationary, iterative methods. Hageman and Young [41] indicate that conjugate gradient possesses a number of desirable features such as not requiring any parameter estimates and converging, in finite iterations, to the true solution of the linear system in the absence of rounding errors. The number of iterations required to converge (to machine accuracy) is at most equal to the number of distinct eigenvalues of the system matrix, A ([6],[38],[41]). Conjugate gradient is often used in minimization problems in meteorological studies ([60]). A good summary of the theory of CG is given by Axelsson [6], Concus et al [17], Golub and Van Loan [38], Hackbusch [40] and Hageman and Young [41], and is repeated here.

Typically the conjugate gradient method is applied to a preconditioned system of the form

$$P^{-1}A\mathbf{U} = P^{-1}\mathbf{b}, \tag{3.12}$$

where P is a non-singular matrix. Ideally P should be easy to invert and should also approximate A in some sense. The aim of the preconditioner with CG is to reduce the interval in which the eigenvalues of the system are found and cluster those eigenvalues where possible; CG converges faster under these conditions. The effectiveness of a preconditioner is therefore determined by the amount of clustering of the eigenvalues and by the condition number, κ.

In general the PCG method performs worst when the eigenvalues of $P^{-1}A$ are evenly spread out ([6],[38],[41]). As Hackbusch [40] succinctly notes, although the asymptotical convergence rate of the PCG method depends on the condition number, and therefore on the size of the extreme eigenvalues, the convergence of the PCG method is influenced by the whole spectrum.

The PCG method calculates solutions to the discrete problem at each iteration using the iteration of the form

$$\mathbf{U}^{m+1} = \mathbf{U}^m + \alpha_m \mathbf{d}^m, \tag{3.13}$$

where vector $\mathbf{d}^m$ is a search direction and α_m a step length. The PCG scheme chooses the search directions $\mathbf{d}^m$ such that they are orthogonal in the A-inner product

$$\left(\mathbf{d}^l, A\mathbf{d}^k\right) = 0, \qquad l \neq k. \tag{3.14}$$

The norm induced by this inner product is known as the energy norm, or A-norm ($\| \mathbf{U} \|_A = (\mathbf{U}^T A \mathbf{U})^{\frac{1}{2}}$). The PCG method then updates the residuals ($\mathbf{r}^m = \mathbf{b} - A\mathbf{U}^m$) by

$$\mathbf{r}^m = \mathbf{r}^{m-1} - \alpha^m \mathbf{q}^m, \tag{3.15}$$

where

$$\mathbf{q}^m = A\mathbf{d}^m, \tag{3.16}$$

and

$$\alpha^m = \frac{(\mathbf{r}^{m-1})^T \mathbf{z}^{m-1}}{(\mathbf{d}^m)^T \mathbf{q}^m}, \tag{3.17}$$

where the vector $\mathbf{z}^m$ is the preconditioned residual computed by solving

$$P\mathbf{z}^m = \mathbf{r}^m. \tag{3.18}$$

The search directions are updated at each iteration using the residuals

$$\mathbf{d}^m = \mathbf{z}^{m-1} + \beta^{i-1}\mathbf{d}^{m-1} \tag{3.19}$$

where the parameter β^m is given by

$$\beta^m = \frac{(\mathbf{r}^m)^T \mathbf{z}^m}{(\mathbf{r}^{m-1})^T \mathbf{z}^{m-1}}, \tag{3.20}$$

which ensures that $\mathbf{d}^m$ and $q^{m-1} = A\mathbf{d}^{m-1}$ are A-orthogonal.

The A-norm may used with the condition number, κ_2 to put a theoretical bound on the speed of convergence relative to the initial error. Golub and Van Loan [38] showed that the difference between the solution at iteration m, $\mathbf{U}^m$, and the real solution, $\mathbf{U}$, may be bounded by

$$\frac{|| \mathbf{U}^m - \mathbf{U} ||_A}{|| \mathbf{U}^0 - \mathbf{U} ||_A} \leq 2 \left(\frac{\kappa_2(P^{-1}A) - 1}{\kappa_2(P^{-1}A) + 1} \right)^m \quad m = 0, 1, \cdots. \tag{3.21}$$

Note the presence of a term involving the initial iterate. Although the choice of initial 'guess' vector has no effect on whether the method will converge, it will affect the number of iterations required to reach a given accuracy. The spectrum of eigenvalues of the iteration matrix (and hence the spectrum of eigenvalues of the preconditioned system) is the vital issue. If the initial error $\mathbf{e}_0 = \mathbf{U}^0 - \mathbf{U}$, when expressed as a linear combination of the eigenvectors, contains a component in the direction of an eigenvector associated with an eigenvalue close to the spectral radius, that component will converge more slowly and affect the speed of convergence of the overall method. On the other hand even if the spectral radius of G is close to 1, if the initial error does not contain any components from eigenvectors associated with large eigenvalues, the method will converge more quickly.

As Axelsson [6] and Golub and Van Loan [38] note, having a symmetric positive-definite system matrix, A, and a symmetric positive-definite preconditioner, P is a sufficient condition for the PCG method to converge. This is noted to be equivalent to $P^{-1}A$ being similar to a symmetric positive-definite matrix. Also it is noted that $\rho(G_P) = \rho(I - P^{-1}A) < 1$ is a necessary condition for convergence.

A wide range of preconditioners exist for attaining the required conditions, some of which we summarise in Section 3.4.

3.4 Preconditioners

We now introduce the preconditioners which we intend to use in our investigations. These could be used in conjunction with the PCG or Chebyshev methods. The preconditioners we consider are suitable for latitude-longitude grid co-ordinate systems, as used in ocean models. Both methods assume that A is positive-definite ([6]). We will assume that this is the case in this section and prove it for our particular problems in Chapters 4 to 7. At that time we consider the other conditions for the convergence of those methods; the sufficient condition that the preconditioners used with them are symmetric positive-definite as well and the necessary condition that $\rho(G_P) = \rho(I - P^{-1}A) < 1$). As an alternative to the former we could show that $P^{-1}A$ is similar to a symmetric positive-definite matrix. We will discuss the preconditioners generally with reference to the stationary iterative method they are related to and give some indication how the splittings used could yield the required properties of symmetric positive-definiteness.

3.4.1 Diagonal preconditioner

One of the main strategies for preconditioning a matrix involves considering approximate inverses. Arguably the simplest of these is a preconditioner, D, containing only the diagonal elements, a_{ii}, of A. In this case the positive-definiteness of the preconditioner follows by simply showing that all of the entries of the diagonal preconditioner are positive.

This is the preconditioner used in the MO free surface formulation. It is also used as a preconditioning step in the preparation of the preconditioned system matrix in the MO rigid-lid formulation. PCG with diagonal preconditioning may be regarded as an acceleration method for the pointwise Jacobi iteration matrix, $G_D = I - D^{-1}A$, where $D = diag[A]$. In order for the diagonally preconditioned conjugate gradient method to be convergent we require that P and A are symmetric positive-definite and that $\rho(G_D) < 1$. The latter may be proved using the following theorem

Theorem 3.11 *([75], p73) Let $A \in \mathbb{R}^{N \times N}$ be a strictly or irreducibly diagonally dominant matrix. Then, the associated point Jacobi matrix is convergent and the iterative method (3.2), with $P = D = diag(A)$ for the matrix problem (3.1) is convergent for any initial vector approximation, $\mathbf{U}^0$.*

3.4.2 Block diagonal preconditioner

The theory for a pointwise Jacobi iteration matrix may be extended to allow the consideration of a block Jacobi iteration matrix. If A is of the partitioned form (3.7) then we may consider a Block Diagonal preconditioner of the form

$$P = \begin{pmatrix} A_{1,1} & & & & & \\ & A_{2,2} & & & & \\ & & A_{3,3} & & & \\ & & & \ddots & & \\ & & & & A_{n_\phi-1,n_\phi-1} & \\ & & & & & A_{n_\phi,n_\phi} \end{pmatrix}.$$

From Hackbusch [40] we know that for

$$G_{Block} = I - P^{-1}A,$$

if μ_i^D is an eigenvalue of P and μ_i is an eigenvalue of A, then the spectrum σ of the eigenvalues of G_{Block} is given by the set

$$\left\{ \tfrac{\mu_i^D - \mu_i}{\mu_i^D} \qquad 1 \leq i \leq n, \right.$$

and therefore the spectral radius is given by

$$\rho(G_{Block}) = max \left\{ \mid \frac{\mu_i^D - \mu_i}{\mu_i^D} \mid \right\} \quad 1 \leq i \leq n.$$

Also from Hackbusch [40] we have the following theorem

Theorem 3.12 *([40], p162) Let A be an M-matrix. Then the pointwise as well as the block Jacobi methods converge, where the latter, however, is faster :*

$$\rho(G_{Block}) \leq \rho(G_D) < 1, \tag{3.22}$$

In (3.22) the strict inequality $0 < \rho(G_{Block}) < \rho(G_D) < 1$ *holds if* $A^{-1} > 0$ *and* $D^{ptw} \neq D^{Block} \neq A$ *with strict inequality if* $A^{-1} > 0$.

There is another way to prove that the block diagonal (and indeed diagonal) preconditioned methods converge. It makes use of the following property (given in Young [81])

Definition 3.14 *(Property A) : A matrix* $A \in \mathbb{R}^{N \times N}$ *has "Property A" if there exist two disjoint subsets* S_1 *and* S_2 *of* W*, the set of the first* N *positive integers, such that* $S_1 + S_2 = W$ *and such that if* $i \neq j$ *and if* $aij \neq 0$ *or* $a_{ji} \neq 0$*, then* $i \in S_1$ *and* $j \in S_2$ *or else* $i \in S_2$ *and* $j \in S_1$.

This property is linked to the property of consistent ordering (also given in Young [81])

Definition 3.15 *(Consistent ordering) : A matrix* $A \in \mathbb{R}^{N \times N}$ *is consistently ordered if for some* t *there exist disjoint subsets* $S_1, S_2, \cdots, S_t$ *of* $W = \{1, 2, \cdots, N\}$ *such that* $\sum_{k=1}^{t} S_k = W$ *and such that if* $a_{ij} \neq 0$ *or* $a_{ji} \neq 0$ *then* $j \in S_{k+1}$ *if* $j > i$ *and* $j \in S_{k-1}$ *if* $j < i$*, where* S_k *is the subset containing* i.

via the theorem

Theorem 3.13 *([81], p145) If $A \in \mathrm{IR}^{N\times N}$ is consistently ordered then A has "property A".*

Hageman and Young [41] note the following connection of Property A with the convergence of diagonal and block diagonal preconditioned methods

Theorem 3.14 *([41],p25) If $A \in \mathrm{IR}^{N\times N}$ is symmetric positive-definite and has "Property A" then the diagonal and block diagonal preconditioned methods converge with $G_{Block} < 1$ and $G_D < 1$. Further if μ_{min} is the algebraically smallest eigenvalue, and ρ is the spectral radius, of the iteration matrices then if A has property A we have $\mu_{min}(G) = -\rho(G)$.*

Young [81] shows that this means that the eigenvalues of G_{Block} and G_D occur in ± pairs if the matrix A has property A.

We can prove that if A has a particular structure then it will be consistently ordered and have property A. A matrix, A, is said to be block tri-diagonal if it can be partitioned into the form

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} & 0 & \cdots & 0 \\ A_{2,1} & A_{2,2} & A_{2,3} & 0 & \cdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & A_{n_\phi,n_\phi-1} & A_{n_\phi,n_\phi} \end{pmatrix}, \tag{3.23}$$

The relevance to property A is highlighted by Young [81] in the theorem

Theorem 3.15 *([81],p145) If $A \in \mathrm{IR}^{N\times N}$ is a block-tridiagonal matrix of the form (3.23) with non-vanishing diagonal blocks, $A_{j,j}$, $1 \leq j \leq n_\phi$, then A is consistently ordered.*

3.4.3 ADI preconditioner

Linewise iterative methods such as Jacobi and block Jacobi iterate along all the lines of a mesh in the same direction. Rates of convergence can be improved by doing a double sweep firstly in the horizontal (row by row) direction, and then in the vertical (column by column) direction. Methods which use this technique are referred to as Alternating Direction Implicit (ADI) methods. The theory of ADI is discussed in numerous references ([3], [5], [9], [40],[55], [73], [75] among others). We summarise the method here.

We split the system matrix A into

$$A = H_\Upsilon + V_\Upsilon.$$

The matrices defined in (3.4.3) are required to have the following properties : H_Υ and V_Υ are symmetric positive-definite and have strictly positive diagonal entries and non-positive off-diagonal entries. They are therefore Stieltjes matrices. For the purposes of preconditioning we assume that H_Υ and V_Υ are relatively easy to invert.

The ADI method is formed by writing the matrix equation (4.6) as a pair of matrix equations

$$\begin{aligned}(H_\Upsilon + \Upsilon I)\mathbf{U} &= (\Upsilon I - V_\Upsilon)\mathbf{U} + \mathbf{b},\\ (V_\Upsilon + \Upsilon I)\mathbf{U} &= (\Upsilon I - H_\Upsilon)\mathbf{U} + \mathbf{b},\end{aligned} \tag{3.24}$$

for any positive scalar, Υ. The iterative matrix of the ADI method, G_{ADI} is similar to

$$(\Upsilon I - H_\Upsilon)(\Upsilon I + H_\Upsilon)^{-1}(\Upsilon I - V_\Upsilon)(\Upsilon I + V_\Upsilon)^{-1}, \tag{3.25}$$

if H_Υ and V_Υ are Stieltjes matrices. An ADI method is stationary if Υ is constant throughout the iteration process and nonstationary if it varies during the iteration process. Skamarock et al [68] discuss a '1D' ADI preconditioner for use with a Helmholtz problem in a non-hydrostatic model. Such a preconditioner would be analogous with only using one of the equations in (3.24) in the preconditioning.

Axelsson [6] summarises how the free parameter, Υ, is calculated for the ADI scheme. Parameters are chosen which are upper and lower bounds of the eigenvalues of H_Υ and V_Υ such that

$$\begin{gathered} 0 < \alpha \leq \mu_i^H, \mu_i^V \leq \beta \\ 1 \leq i \leq N. \end{gathered}$$

The free parameter, Υ, is then chosen in such a way that the spectral radii of the iteration matrix (3.25) is minimized. This is accomplished by taking $\Upsilon = \sqrt{\alpha\beta}$. In addition Varga [75] states that

Theorem 3.16 *Let H_Υ and V_Υ be non-negative definite matrices, where at least one of the matrices H_Υ and V_Υ is positive-definite. Then, for any $\Upsilon > 0$ the ADI iteration matrix (3.25) is convergent i.e. $\rho(G_{ADI}) < 1$. $\rho(G_{ADI}) < 1$ for $\Upsilon > 0$.*

i.e. we can guarantee the convergence of the ADI preconditioned method by the use of strictly positive parameter values, Υ.

Axelsson [6] notes that commutativity of H_Υ and V_Υ (i.e. $H_\Upsilon V_\Upsilon = V_\Upsilon H_\Upsilon$) is not a condition for the convergence of the stationary ADI method. However, it is noted to be a necessary condition when considering nonstationary ADI methods; Axelsson [6] shows that commutativity is required in order to use a sequence of parameters through the iteration process. A new alternative to this, which does not require commutativity, is discussed in the next section. For more details on ADI theory see Axelsson [6], Ames [3] and Varga [75].

3.4.4 ADI preconditioner with spatially varying parameter

In this section we propose a new variant on the basic ADI method for solving $A\mathbf{U} = \mathbf{b}$. Typically the ADI method is extended by using a set of parameters sequentially, varying iteration by iteration ([19],[31],[76],[77]). As an alternative we investigate the possibility of using a set of parameters spatially. We assume that our system matrix, A, may be written in block partitioned form (3.7), and that Υ is now a vector of parameter values. We further assume that there are as many entries in Υ as there are rows of submatrices, n_ϕ, in the matrix A. The motivation for doing this is to address the mesh anisotropy in the elliptic operators we consider. It is hoped that by selecting different parameters, Υ_j, $1 \leq j \leq n_\phi$, we may be able to differentially treat the anisotropy. Note that this still may be regarded as a stationary method as the parameters do not change through the iteration process.

We want to compute the optimal values of Υ_j such that $\rho(G_{ADI})$ is minimized. In the constant Υ case parameters, α and β were used to bound the eigenvalues of H_Υ and V_Υ. We assume that we now have n_ϕ such bounds, α_j and β_j, $1 \leq j \leq n_\phi$. We also assume a more specific structure to one of the matrices used (H_Υ say, without loss of generality). We assume that H_Υ is a block-diagonal matrix of the form

$$H_\Upsilon = \begin{pmatrix} D_1^H & & & \\ & D_2^H & & \\ & & \ddots & \\ & & & D_{n_\phi}^H \end{pmatrix}. \qquad (3.26)$$

We replace ΥI in the ADI preconditioner by D^Υ where D^Υ is a block diagonal matrix with diagonal blocks given by $D_j^{\Upsilon} = \Upsilon_j I_{n_\lambda}$ where I_{n_λ} is the $n_\lambda \times n_\lambda$ identity matrix. Each lower bound, α_j, is a lower bound on the eigenvalues of each block D_j^H of H_Υ as well as those of V_Υ, whilst each upper bound, β_j is an upper bound on the eigenvalues of each block D_j^H of H_Υ and as well as those of V_Υ. A similar method is used to determine the 'optimal' values of α_j and β_j to that used by Axelsson [6] for the constant parameter case. This is summarised by the following theorem

Theorem 3.17 *Let $A \in \mathbb{R}^{N \times N}$ be symmetric positive definite and of block partitioned form (3.7) and assume that A may be split into $A = H_\Upsilon + V_\Upsilon$ where H_Υ and V_Υ are $N \times N$ Stieltjes matrices. Assume that H_Υ may be written in block tridiagonal form (3.26). Also assume that the ADI parameter from the stationary method has been replaced by a vector of n_ϕ parameters, Υ_j, $1 \leq j \leq n_\phi$. Let D^Υ be a block diagonal matrix with diagonal blocks given by $D_j^\Upsilon = \Upsilon_j I_{n_\lambda}$. Further assume that the eigenvalues of each block D_j^H of H_Υ and those of V_Υ may be bounded below by j bounds, β_j, and above by j bounds, α_j. Then the 'optimal' parameters to use in the spatially varying ADI scheme are given by*

$$\Upsilon_j = \sqrt{\alpha_j \beta_j}. \tag{3.27}$$

Proof

Since H_Υ and V_Υ are symmetric positive-definite, with strictly positive diagonal entries, their eigenvalues are positive (by Theorem 3.2) and

$$|| \, (\Upsilon_j I_{n_\lambda} - D_j^H)(\Upsilon_j I_{n_\lambda} + D_j^H)^{-1}) \, ||_2 = \rho((\Upsilon_j I_{n_\lambda} - D_j^H)(\Upsilon_j I_{n_\lambda} + D_j^H)^{-1})$$
$$= max \left| \frac{\Upsilon_j - \mu_{ij}^H}{\Upsilon_j + \mu_{ij}^H} \right| \quad 1 \leq j \leq n_\phi, 1 \leq i \leq n_\lambda,$$

where μ_{ij}^H are the eigenvalues of D_j^H and

$$\begin{aligned} &\| (D^\Upsilon - V_\Upsilon)(D^\Upsilon + V_\Upsilon)^{-1}) \|_2 = \rho((D^\Upsilon - V_\Upsilon)(D^\Upsilon + V_\Upsilon)^{-1}) \\ &= max \left| \tfrac{\Upsilon_j - \mu_i^V}{\Upsilon_j + \mu_i^V} \right| \qquad 1 \leq j \leq n_\phi, 1 \leq i \leq N. \end{aligned}$$

where μ_i^V are the eigenvalues of D^V. Therefore

$$\begin{aligned} &\rho(G_{ADIV}) = \rho((\Upsilon_j I_{n_\lambda} - D_j^H)(\Upsilon_j I_{n_\lambda} + D_j^H)^{-1}(D^\Upsilon - V_\Upsilon)(D^\Upsilon + V_\Upsilon)^{-1})) \\ &\leq \| (\Upsilon_j I_{n_\lambda} - D_j^H)(\Upsilon_j I_{n_\lambda} + D_j^H)^{-1}(D^\Upsilon - V_\Upsilon)(D^\Upsilon + V_\Upsilon)^{-1}) \|_2 \\ &\leq \| (\Upsilon_j I_{n_\lambda} - D_j^H)(\Upsilon_j I_{n_\lambda} + D_j^H)^{-1} \|_2 \| (D^\Upsilon - V_\Upsilon)(D^\Upsilon + V_\Upsilon)^{-1}) \|_2 \end{aligned}$$

i.e.

$$\rho(G_{ADIV}) \leq max \left| \frac{\Upsilon_j - \mu_{ij}^H}{\Upsilon_j + \mu_{ij}^H} \right| .max \left| \frac{\Upsilon_j - \mu_i^V}{\Upsilon_j + \mu_l^V} \right|, \qquad 1 \leq i \leq n_\lambda, 1 \leq j \leq n_\phi, 1 \leq l \leq N.$$

Applying our bounds gives

$$\rho(G_{ADIV}) \leq max \left\{ \left| \frac{\Upsilon_j - \alpha_j}{\Upsilon_j + \alpha_j} \right|, \left| \frac{\Upsilon_j - \beta_j}{\Upsilon_j + \beta_j} \right| \right\} .max \left\{ \left| \frac{\Upsilon_j - \alpha_j}{\Upsilon_j + \alpha_j} \right|, \left| \frac{\Upsilon_j - \beta_j}{\Upsilon_j + \beta_j} \right| \right\}, \qquad 1 \leq j \leq n_\phi. \tag{3.28}$$

We want to chose the Υ_j such that the bound in (3.28) is as small as possible. Firstly note that $\rho(G_{ADIV}) < 1$ for $\Upsilon_j > 0$. For $\Upsilon_j > 0$ we must have

$$\begin{aligned} \Upsilon_j - \alpha_j > 0, \\ \Upsilon_j - \beta_j < 0. \end{aligned}$$

Also note that each factor in the bound (3.28) is minimized when

$$\frac{\Upsilon_j - \alpha_j}{\Upsilon_j + \alpha_j} = \frac{\Upsilon_j - \beta_j}{\Upsilon_j + \beta_j},$$

i.e. when

$$\frac{1}{\Upsilon_j^2} + \frac{\alpha_j + \beta_j}{2\alpha_j \beta_j} \left(\frac{1}{\Upsilon_j} - \frac{1}{\Upsilon_j} \right) - \frac{1}{\alpha_j \beta_j} = 0$$

$\implies \Upsilon_j = \sqrt{\alpha_j \beta_j}$.

During the proof it was demonstrated that in order to obtain the necessary condition $\rho(G_{ADIV}) < 1$ for the convergence of the spatially varying ADI preconditioned method, we are required to use parameter values $\Upsilon_j > 0$.

3.4.5 Scaling by Binormalization

In this section we summarise the binormalization scaling, used most recently by Livne and Golub [52], for scaling all of the rows and columns of a real symmetric matrix to unit 2-norm. The system matrix A is diagonally scaled by $F = diag(f_1, \cdots, f_n)$ such that the matrix obtained, $\hat{A} = FAF$, is symmetric and satisfies

$$\sum_j \hat{A_{ij}}^2 = 1 \quad i = 1, \cdots, n. \tag{3.29}$$

As Livne and Golub [52] comment, this binormalization scaling may be regarded as a form of preconditioning i.e. $\kappa(\hat{A}) < \kappa(A)$. A key issue is how this compares with straightforward diagonal (Jacobi) preconditioning. It is noted by Livne and Golub [52] that the diagonal preconditioner is "optimal" (i.e. the condition number of the preconditioned system is smaller than that with Binormalization scaling) when the system matrix, A, has "Property A".

In the numerical experiments of Chapter 5 we will investigate if the eigenvalue distribution of a Binormalization scaled preconditioned system matrix has a distribution which a PCG method could take advantage of (clustered eigenvalues, for example). The following theorem is highlighted by Livne and Golub to [52] illustrate the relation between diagonal and binormalization preconditioning.

Theorem 3.18 *For any symmetric positive-definite matrix A*

$$\kappa(P_D^{-1}A) \leq n\kappa(\hat{A}).$$

The factor n can be replaced by p when there are at most p non-zero elements per row of A.

We will check the validity of this bound experimentally in our numerical experiments of Chapter 5.

3.5 Implementation details

In this section we highlight some ways in which the algorithms for our preconditioned conjugate gradient routines may be implemented more efficiently. In particular we consider the form of the preconditioners and demonstrate that certain parts of the preconditioning step in the PCG algorithm may be performed 'once and once only' or 'off-line'. This refers to parameters and vectors which are only calculated once and therefore ought to be calculated outside of the main iteration sweeps. Doing the calculations outside the main iteration sweeps will lead to large savings in computation time. This is especially true for the time-varying barotropic models. In those models the system matrix does not change with time and therefore any 'off-line' calculations should be performed outside the main time-varying calculations (as well outside the main iterations sweeps). If the parameters and vectors could be stored then a model run could be repeated many times using the saved vectors with even more computational savings.

Various parts of algorithms used to implement the preconditioners we have introduced in this chapter could be classed as 'off-line' calculations. For the block preconditioner the diagonal blocks are factored in order to make the algorithm more efficient. This could be done 'off-line'. Depending on the structure of the matrices chosen, a similar thing could be done for the ADI preconditioners. The parameters for the ADI preconditioner we use in our numerical experiments of Chapters 5, 6, and 7, were calculated using the EIGS subroutine in MATLAB which estimates bounds on the eigenvalues of the matrices H_Υ and V_Υ. This can also be done 'off-line'. Finally the binormalization scaling which calculates the scaled matrix need only be done once as the matrix A is constant co-efficient. In the numerical experiments of Chapters 5, 6, and 7, any CPU times we give are for the main iteration process only. Any 'off-line' calculations are not included in the timings for the reasons we have given.

3.6 Summary

This chapter has summarised the important definitions and theorems that underpin the numerical methods and preconditioners used in this study. These will be applied in later chapters to confirm the convergence properties of the numerical methods and preconditioners with the specific problems that we investigate. We have introduced Diagonal, Block Diagonal and Alternating-Direction-Implicit preconditioners, as well as Binormalization scaling. We have briefly described techniques for efficiently implementing these preconditioners in a solution algorithm. We have also summarised the theory of the preconditioned conjugate gradient and Chebyshev semi-iterative methods and their use an nonstationary methods for accelerating preconditioned stationary methods.

Chapter 4

Spherical coordinate model : constant depth problem

4.1 Introduction

We now introduce the basic spherical coordinate model we shall use to investigate how mesh anisotropy affects the convergence of iterative solutions to elliptic problems. In this chapter we introduce the constant depth problem (i.e the basic Helmholtz problem) using a standard five-point discretisation operator. We prove the convergence of the numerical methods, we use in the numerical experiments of Chapter 5, using the general theory discussed in Chapter 3. Section 4.2 describes the formulation for the problem, in both limited area and periodic domain cases, and the discretisation scheme used. The consistency of the discretisation scheme is checked using Truncation Error analysis in Section 4.3. Theoretical analysis is performed in Sections 4.4 confirming the convergence of the methods and preconditioners considered. Finally in Section 4.5 the Gerschgorin Circle Theorem 3.6 is used to derive approximate bounds on the eigenvalues and conditioning of the preconditioned matrices. Qualitative assessments will then be made of the likely speeds of convergence of the preconditioned methods.

4.2 Problem formulation and discretisation

Recall that the Laplacian operator in 2D-spherical coordinates is

$$\nabla^2_{\lambda,\phi}U = \frac{1}{cos\phi}\left[\frac{\partial}{\partial\lambda}\left(\frac{1}{cos\phi}\frac{\partial U}{\partial\lambda}\right) + \frac{\partial}{\partial\phi}\left(cos\phi\frac{\partial U}{\partial\phi}\right)\right]. \tag{4.1}$$

We consider a modified Helmholtz equation of the form $-\nabla^2_{\lambda,\phi}U + kU = \gamma(\lambda,\phi)$ (where $k \geq 0$ and $\gamma(\lambda,\phi)$ is a source function). For our numerical experiments we use a fixed mesh of $n_\lambda \times n_\phi$ grid points. We concentrate on a theoretical segment of Northern Hemisphere ocean from 10^oN to between 40^o and 89.5^oN in the latitudinal (ϕ) direction with Dirichlet boundary conditions of $U = 0$ in that direction. The position of the northern boundary is given by the parameter ϕ_{NB} (where $\phi_{NB} \in [40^o, 89.5^o]$).In the longitudinal ($\lambda$) direction we consider either a limited area model from 0^oE to 30^oE with Dirichlet boundary conditions of $U = 0$, yielding the problem

$$\begin{cases} -\frac{1}{cos\phi}\left[\frac{\partial}{\partial\lambda}\left(\frac{1}{cos\phi}\frac{\partial U}{\partial\lambda}\right) + \frac{\partial}{\partial\phi}\left(cos\phi\frac{\partial U}{\partial\phi}\right)\right] + kU = \gamma(\lambda,\phi) \\ \lambda \in (0^oE, 30^oE) \quad \phi \in (10^oN, \phi_{NB}) \\ U(0^oE,\phi) = 0, U(30^oE,\phi) = 0 \\ U(\lambda, 10^oN) = 0, U(\lambda,\phi_{NB}) = 0 \\ \phi_{NB} \in (40^oN, 89.5^oN), \end{cases} \tag{4.2}$$

or we allow λ to vary through an entire hemispheric revolution and take periodic boundary conditions, thus yielding the problem

$$\begin{cases} -\frac{1}{cos\phi}\left[\frac{\partial}{\partial\lambda}\left(\frac{1}{cos\phi}\frac{\partial U}{\partial\lambda}\right)+\frac{\partial}{\partial\phi}\left(cos\phi\frac{\partial U}{\partial\phi}\right)\right]+kU=\gamma(\lambda,\phi) \\ \lambda\in(0^oW,0^oE) \quad \phi\in(10^oN,\phi_{NB}) \\ U(0^oW,\phi)=U(0^oE,\phi) \\ \frac{\partial U(0^oW,\phi)}{\partial\lambda}=\frac{\partial U(0^oE,\phi)}{\partial\lambda} \\ U(\lambda,10^oN)=0, U(\lambda,\phi_{NB})=0 \\ \phi_{NB}\in(40^oN,89.5^oN). \end{cases} \tag{4.3}$$

These problems are intended to be analogous to the ocean models discussed in Chapter 2 that solve for the increments of the quantities of interest. The solutions at each time step in those time-varying problems should satisfy the same boundary conditions at each time step. Therefore our choice of taking Dirichlet conditions of $U = 0$ everywhere (except in the longitudinal, λ, direction in the periodic problem) is reasonable.

In the following discretisation and convergence theory we assume that we are considering the limited area case. The few changes that are required to the numerical scheme, in the periodic case, are discussed in the next section. We use the following five-point discretisation scheme for our problems:

$$\begin{aligned} -\nabla^2U_{ij}+kU_{ij} \approx -\frac{1}{\delta\lambda\delta\phi}\left[\frac{1}{cos^2\phi_j}\left(\frac{(U_{i+1j}-U_{ij})\delta\phi}{\delta\lambda}-\frac{(U_{ij}-U_{i-1j})\delta\phi}{\delta\lambda}\right)\right. \\ \left.+\frac{1}{cos\phi_j}\left(\frac{cos\phi_{j+\frac{1}{2}}(U_{ij+1}-U_{ij})\delta\lambda}{\delta\phi}-\frac{cos\phi_{j-\frac{1}{2}}(U_{ij}-U_{ij-1})\delta\lambda}{\delta\phi}\right)\right]+kU_{ij}=\gamma(\lambda_i,\phi_j) \end{aligned} \tag{4.4}$$

which is equivalent to

$$\begin{aligned} -\left[\frac{1}{cos^2\phi_j}\left(\frac{U_{i+1j}-2U_{ij}+U_{i-1j}}{\delta\lambda^2}\right)\right. \\ \left.+\frac{1}{cos\phi_j}\left(\frac{cos\phi_{j+\frac{1}{2}}(U_{ij+1}-U_{ij})-cos\phi_{j-\frac{1}{2}}(U_{ij}-U_{ij-1})}{\delta\phi^2}\right)\right]+kU_{ij}=\gamma(\lambda_i,\phi_j). \end{aligned} \tag{4.5}$$

The system of equations may then be written in the classical matrix form

$$A\mathbf{U}=\mathbf{b}, \tag{4.6}$$

where the variable **U** is a (unknown) column vector of the grid points of the model variable U, and **b** is a (known) column vector representing boundary values and source terms. The system matrix A is a real $N \times N$ matrix representing the discretised model equations, where N is the number of grid points in the discrete grid. The size of N is determined by $N = n_\lambda \times n_\phi$ where n_λ, n_ϕ are the number of grid points in the longitudinal and latitudinal directions respectively. The system matrix A is also square and sparse. The exact form of A will depend on the type of elliptic equation being discretised, the boundary conditions that are required, and the ordering of the grid points across the domain of the problem. The system matrix A is non-symmetric but it can be symmetrised. To do this we multiply each equation through by $cos\phi_j$ and combine this into the source term $\gamma(\lambda_i, \phi_j)$. This gives

$$\begin{aligned} &cos\phi_j \left(-\nabla^2 U_{ij} + kU_{ij}\right) \approx -\left[\frac{1}{cos\phi_j}\left(\frac{U_{i+1j}-2U_{ij}+U_{i-1j}}{\delta\lambda^2}\right)\right. \\ &\left.+\left(\frac{cos\phi_{j+\frac{1}{2}}(U_{ij+1}-U_{ij})-cos\phi_{j-\frac{1}{2}}(U_{ij}-U_{ij-1})}{\delta\phi^2}\right)\right] + kcos\phi_j U_{ij} = \gamma(\lambda_i, \phi_j)cos(\phi_j). \end{aligned} \tag{4.7}$$

We order the equations for our grid points using the natural ordering. In this case, therefore, our system matrix A has the block-tridiagonal structure

$$A = \begin{pmatrix} D_1 & C_1 & & & & \\ B_2 & D_2 & C_2 & & & \\ & B_3 & D_3 & C_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & & D_{n_\phi-1} & C_{n_\phi-1} \\ & & & & B_{n_\phi} & D_{n_\phi} \end{pmatrix}, \tag{4.8}$$

where

$$D_j = tridiag\left[\; -\frac{1}{cos\phi_j\delta\lambda^2}, \quad \frac{2}{cos\phi_j\delta\lambda^2} + \frac{cos\phi_{j+\frac{1}{2}}+cos\phi_{j-\frac{1}{2}}}{\delta\phi^2} + kcos\phi_j, \quad -\frac{1}{cos\phi_j\delta\lambda^2} \;\right], \tag{4.9}$$

$$B_j = diag\left[-\frac{cos\phi_{j-\frac{1}{2}}}{\delta\phi^2}\right] \quad 2 \leq j \leq n_\phi, \tag{4.10}$$

$$C_j = diag\left[-\frac{cos\phi_{j+\frac{1}{2}}}{\delta\phi^2}\right] \quad 1 \leq j \leq n_\phi - 1. \tag{4.11}$$

From the definitions (4.9) to (4.11) it is straightforward to observe that each block D_j is symmetric and this, combined with the fact that

$$B_j = diag\left[-\frac{cos\phi_{j-\frac{1}{2}}}{\delta\phi^2}\right] = C_{j-1}, \quad 2 \leq j \leq n_\phi, \tag{4.12}$$

is enough for us to conclude that the matrix A is symmetric. Further properties of A are discussed in Section 4.4.1.

4.2.1 Periodic boundary conditions

If we extend our spherical region in the longitudinal direction (λ) to include a whole hemisphere it becomes necessary for us to consider periodic boundary conditions for our domain (We retain our Dirichlet boundary conditions in the latitudinal (ϕ) direction). We obtain our periodic boundary conditions by setting

$$U(1, j) = U(n_\lambda, j), \tag{4.13}$$

$$U(n_\lambda, j) = U(1, j). \tag{4.14}$$

This changes the structure of the first and last rows of each block D_j. The first row of each block D_j is of the form

$$\left(\frac{2}{cos\phi_j\delta\lambda^2} + \frac{cos\phi_{j+\frac{1}{2}} + cos\phi_{j-\frac{1}{2}}}{\delta\phi^2} + kcos\phi_j, \quad -\frac{1}{cos\phi_j\delta\lambda^2} \quad \cdots \quad -\frac{1}{cos\phi_j\delta\lambda^2} \right), \tag{4.15}$$

whilst the last (n_λ^{th}) row of each block D_j

$$\left(-\frac{1}{cos\phi_j\delta\lambda^2} \quad \cdots \quad -\frac{1}{cos\phi_j\delta\lambda^2}, \quad \frac{2}{cos\phi_j\delta\lambda^2} + \frac{cos\phi_{j+\frac{1}{2}} + cos\phi_{j-\frac{1}{2}}}{\delta\phi^2} + kcos\phi_j \right). \tag{4.16}$$

Note that A is still symmetric with this formulation. The B_j and C_j sub-matrices are unchanged and retain the property (4.12). The extra terms in the D_j's are added to the top right and bottom left hand corners of each matrix D_j. They are also identical to each other within each D_j. Thus each sub-matrix D_j is still symmetric and therefore the system matrix A is as well.

4.3 Truncation error analysis

In this section we shall check the properties of the discrete, five-point, differential operator we have used to discretise our spherical elliptic problem, by using Truncation error analysis. The discretisation scheme we have used to approximate our modified Helmholtz problems, in unsymmetric form, is given by :

$$\begin{aligned}-\nabla^2_{\lambda,\phi}U_{ij} + kcos\phi_j U_{ij} \approx -\Bigg[&\frac{1}{cos(\phi_j)^2}\left(\frac{U_{i+1j}-2U_{ij}+U_{i-1j}}{\delta\lambda^2}\right) \\ &+\frac{1}{cos(\phi_j)}\left(\frac{cos(\phi_{j+\frac{1}{2}})(U_{ij+1}-U_{ij})-cos(\phi_{j-\frac{1}{2}})(U_{ij}-U_{ij-1})}{\delta\phi^2}\right)\Bigg] + kU_{ij}.\end{aligned} \tag{4.17}$$

evaluated at $(\lambda_i,\, \phi_j)$. This is equivalent to

$$\begin{aligned}-\nabla^2_{\lambda,\phi}U_{ij} + kcos\phi_j U_{ij} \approx -\Bigg[&\frac{1}{cos(\phi^2)}\left(\frac{U(\lambda+\delta\lambda,\phi)-2U(\lambda,\phi)+U(\lambda-\delta\lambda,\phi)}{\delta\lambda^2}\right) \\ &+\frac{1}{cos(\phi)}\left(\frac{cos(\phi+\frac{\delta\phi}{2})(U(\lambda,\phi+\delta\phi)-U(\phi,r))-cos(\phi-\frac{\delta\phi}{2})(U(\lambda,\phi)-U(\lambda,\phi-\delta\phi))}{\delta\phi^2}\right)\Bigg] + kU(\lambda,\phi).\end{aligned} \tag{4.18}$$

Using Taylor series expansions about (λ, ϕ) this is approximately given by

$$\begin{aligned}-\Bigg[&\frac{1}{cos\phi^2}\left(\frac{-2U+U+\delta\lambda U_\lambda+\frac{\delta\lambda^2}{2!}U_{\lambda\lambda}+\frac{\delta\lambda^3}{3!}U_{\lambda\lambda\lambda}+U-\delta\lambda U_\lambda+\frac{\delta\lambda^2}{2!}U_{\lambda\lambda}-\frac{\delta\lambda^3}{3!}U_{\lambda\lambda\lambda}+O(\delta\lambda^4)}{\delta\lambda^2}\right) \\ &+\frac{1}{cos\phi}\left(\frac{cos(\phi+\frac{\delta\phi}{2})(-U+U+\delta\phi U_\phi+\frac{\delta\phi^2}{2!}U_{\phi\phi}+\frac{\delta\phi^3}{3!}U_{\phi\phi\phi}+O(\delta\phi^4)}{\delta\phi^2}\right) \\ &-\frac{1}{cos\phi}\left(\frac{cos(\phi-\frac{\delta\phi}{2})(-U+U+\delta\phi U_\phi-\frac{\delta\phi^2}{2!}U_{\phi\phi}+\frac{\delta\phi^3}{3!}U_{\phi\phi\phi}+O(\delta\phi^4)}{\delta\phi^2}\right)\Bigg] + kU\end{aligned}$$

$$= -\Big[\tfrac{1}{cos\phi^2}U_{\lambda\lambda} + O(\delta\lambda^2)$$

$$+\tfrac{1}{cos\phi}\left(\tfrac{(cos\phi - \frac{\delta\phi}{2}sin_\phi - \frac{\delta\phi^2}{4}cos\phi + O(\delta\phi^3))(\delta\phi U_\phi + \frac{\delta\phi^2}{2!}U_{\phi\phi} + \frac{\delta\phi^3}{3!}U_{\phi\phi\phi} + O(\delta\phi^4))}{\delta\phi^2}\right)$$

$$-\tfrac{1}{cos\phi}\left(\tfrac{(-cos\phi - \frac{\delta\phi}{2}sin_\phi + \frac{\delta\phi^2}{4}cos\phi + O(\delta\phi^3))(\delta\phi U_\phi - \frac{\delta\phi^2}{2!}U_{\phi\phi} + \frac{\delta\phi^3}{3!}U_{\phi\phi\phi} + O(\delta\phi^4))}{\delta\phi^2}\right)\Big] + kU$$

$$= -\Big[\tfrac{1}{cos\phi^2}U_{\lambda\lambda} + \tfrac{1}{\delta\phi}U_\phi - \tfrac{sin_\phi}{2cos_\phi}U_\phi - \tfrac{\delta\phi}{4}U_\phi + \tfrac{1}{2!}U_{\phi\phi} - \tfrac{\delta\phi sin\phi}{4cos\phi}U_{\phi\phi} + \tfrac{\delta\phi}{3!}U_{\phi\phi\phi}$$

$$-\tfrac{1}{\delta\phi}U_\phi - \tfrac{sin_\phi}{2cos_\phi} + \tfrac{\delta\phi}{4}U_\phi + \tfrac{1}{2!}U_{\phi\phi} + \tfrac{\delta\phi sin\phi}{4cos\phi}U_{\phi\phi} - \tfrac{\delta\phi}{3!}U_{\phi\phi\phi}\Big] + kU + O(\delta\lambda^2)$$

$$+O(\delta\phi^2)$$

$$= -\left[\frac{1}{cos\phi^2}U_{\lambda\lambda} + U_{\phi\phi} - \frac{sin\phi}{cos\phi}U_\phi\right] + kU + O(\delta\lambda^2) + O(\delta\phi^2). \qquad (4.19)$$

We note here that

$$\frac{1}{cos\phi}\frac{\partial}{\partial\phi}\left(cos\phi\frac{\partial U}{\partial\phi}\right) = -\frac{sin\phi}{cos\phi}\frac{\partial U}{\partial\phi} + \frac{\partial^2 U}{\partial\phi^2}.$$

Hence the last line of equation (4.19) is equal to the differential equation apart from the higher order terms. Therefore the truncation error of our scheme is order $(\delta\lambda^2 + \delta\phi^2)$, and our scheme is therefore consistent with the differential equation. This property is needed to ensure the convergence of the discrete solution to that of the continuous problem as the step sizes ($\delta\lambda$ and $\delta\phi$) go to zero.

4.4 Convergence analysis

In this section we use the theorems and definitions we introduced in Chapter 3 to establish the convergence properties of the preconditioned methods we use in the numerical experiments of Chapter 5. Although we will be referring to the PCG method throughout this section, as that is what we shall exclusively in Chapter 5, the analysis follows in a similar manner for the Chebyshev semi-iterative method. We begin by confirming properties for the system matrix A for the four sub-cases we could have ($k > 0$ or $k = 0$ with Dirichlet or Periodic boundary conditions). We then consider the properties of our preconditioned methods.

4.4.1 Properties of A

We will firstly confirm some of the general properties of A which hold irrespective of the value of k (within the limit $k \geq 0$ considered in this study) or the choice of boundary conditions in the longitudinal direction. We have already demonstrated the symmetry of the matrix A generally in Sections 4.2 and 4.2.1. Note that, by definition, we cannot have non-positive discrete horizontal stepsizes. Therefore $\delta\lambda > 0$ and $\delta\phi > 0$. Also we are only considering $k \geq 0$. In addition on the domain we are considering $cos\phi \in (0, 1)$. Therefore by considering (4.9), (4.10), (4.11), (4.15) and(4.16) we find that we have $a_{ll} > 0$, and $a_{lm} \leq 0$ for $l \neq m$, $1 \leq l, m \leq N$. We now consider the four specific cases :

- (1) $k > 0$, Dirichlet bcs. : We have already demonstrated that the diagonal entries of the system matrix A are strictly positive, irrespective of k or the boundary conditions in the λ direction. In addition we note that, due to the conditions stated for $\delta\lambda$, $\delta\phi$, k and $cos\phi$, the quantities stated in (4.9), (4.10), and (4.11) are strictly non-zero across the whole domain (for all j). Therefore we may deduce that the connected graph of our matrix A is strongly connected. Hence, via Theorem 3.1, our matrix A is irreducible.

  We now consider the conditions under which the system matrix A is strictly/irreducibly diagonally dominant. For a general line of the matrix A we have that

$$a_{ll} = \frac{2}{cos\phi_j \delta\lambda^2} + \frac{cos\phi_{j+\frac{1}{2}} + cos\phi_{j-\frac{1}{2}}}{\delta\phi^2} + kcos\phi_j, \qquad (4.20)$$

whilst where $l = (j-1)n_\lambda + i$

$$\sum_{m=1 m\neq l}^{N} \mid a_{lm} \mid \leq \frac{2}{cos\phi_j \delta\lambda^2} + \frac{cos\phi_{j+\frac{1}{2}} + cos\phi_{j-\frac{1}{2}}}{\delta\phi^2}. \tag{4.21}$$

Since $k > 0$ we observe that we have

$$a_{ll} > \sum_{m=1, m\neq l}^{N} \mid a_{lm} \mid, \tag{4.22}$$

for every row. We therefore have a matrix that is strictly diagonally dominant. Since it is irreducible it is also irreducibly diagonally dominant. In addition we have already shown that $a_{ll} > 0$, and $a_{lm} \leq 0$ for $i \neq j$, $1 \leq l, m \leq N$. Therefore we may deduce via Theorem 3.2 that A is nonsingular with strictly positive eigenvalues and is positive definite. We may also deduce by definition that A is a Stieltjes matrix and therefore, via Theorems 3.3 and 3.4, that A is an M-matrix with $A^{-1} > 0$.

- (2) $k = 0$, Dirichlet bcs. : With $k = 0$ we still have $a_{ll} > 0$ for $1 \leq l \leq N$. We therefore do not lose any of the connectedness of A by taking $k = 0$. As all other factors remain the same as case(1) we can deduce that the connected graph of our matrix A is still strongly connected and hence, via Theorem 3.1, that our matrix A is irreducible. We no longer have a matrix which is strictly diagonally dominant though. It is strictly diagonally dominant in certain rows (the first and last n_λ rows, and the first and last rows of each block row) and diagonally dominant in all of the others. Since the matrix is irreducible it is therefore still irreducibly diagonally dominant. In addition $a_{ii} > 0$, and $a_{ij} \leq 0$ for $i \neq j$. Therefore we may again deduce via Theorem 3.2 that A is nonsingular with strictly positive eigenvalues and is positive definite.

Again we may also deduce by definition that A is a Stieltjes matrix and therefore, via Theorems 3.3 and 3.4, that A is an M-matrix with $A^{-1} > 0$.

- (3) $k > 0$, Periodic bcs. : The addition of extra terms to the matrix, via the use of periodic boundary conditions, adds to the connectedness of A. Since the matrix A in case(1) was shown to possess a strongly connected graph, it follows that the connected graph of the matrix A is also strongly connected. Hence, via Theorem 3.1, our matrix A is irreducible. Also, despite the extra terms, the matrix A is strictly diagonally dominant when $k > 0$ (as in case(1)). Since it is irreducible it is also still irreducibly diagonally dominant. In addition $a_{ii} > 0$, and $a_{ij} \leq 0$ for $i \neq j$. Therefore we may again deduce via Theorem 3.2 that A is nonsingular with strictly positive eigenvalues and is positive definite. Again we may also deduce by definition that A is a Stieltjes matrix and therefore, via Theorems 3.3 and 3.4, that A is an M-matrix with $A^{-1} > 0$.

- (4) $k = 0$, Periodic bcs. : With $k = 0$ we still have $a_{ll} > 0$ for $1 \leq l \leq N$. We therefore do not lose any of the connectedness of A we had in case(3) by taking $k = 0$. As all other factors remain the same as case(3) we can deduce that the connected graph of our matrix A is still strongly connected and hence, via Theorem 3.1, that our matrix A is irreducible. We no longer have a matrix which is strictly diagonally dominant though. It is strictly diagonally dominant in certain rows (the first and last n_λ rows) and diagonally dominant in all of the others. Since the matrix is irreducible it is therefore still irreducibly diagonally dominant. In addition $a_{ii} > 0$, and $a_{ij} \leq 0$ for

$i \neq j$. Therefore we may again deduce via Theorem 3.2 that A is nonsingular with strictly positive eigenvalues and is positive definite. Again we may also deduce by definition that A is a Stieltjes matrix and therefore, via Theorems 3.3 and 3.4, that A is an M-matrix with $A^{-1} > 0$.

In all of the above cases we are considering a system matrix A that is of block-tridiagonal form. Therefore by Theorem 3.15 the matrix A is consistently ordered and hence by Theorem 3.13 has property A.

4.4.2 Diagonal preconditioner

This preconditioner was introduced in Section 3.4.1. The preconditioner $P = D$ contains only the diagonal elements of A, a_{ll} where $1 \leq l \leq N$. It corresponds to the Jacobi stationary iterative method, $G_D = I - D^{-1}A$, introduced in Section 3.2.2. As stated in Section 3.4.1, in order to show the positive-definiteness of P we are simply required to show that all of the entries of the diagonal preconditioner are strictly positive. This is indeed the case as shown in Section 4.2. Also in order for the PCG and Chebyshev semi-iterative methods with diagonal preconditioning to be convergent we also require that $\rho(G_D) < 1$. Since A is irreducibly or strictly diagonally dominant this follows from Theorem 3.11. Therefore we can guarantee the convergence of the PCG and Chebyshev semi-iterative methods with diagonal preconditioning. This could also have been deduced from the fact that the system matrix A has property A. From this we know that the eigenvalues of G_D will occur in ± pairs via the remarks of Section 3.4.1.

4.4.3 Block diagonal preconditioner

This preconditioner was introduced in Section 3.4.2. We may consider a Block Diagonal preconditioner of the form $P = D = blockdiag(A)$ where

$$P = \begin{pmatrix} D_1 & & & & & \\ & D_2 & & & & \\ & & D_3 & & & \\ & & & \ddots & & \\ & & & & D_{n_\phi - 1} & \\ & & & & & D_{n_\phi} \end{pmatrix}. \tag{4.23}$$

with the $D_j's$ of the form given in (4.9). The block Jacobi iteration matrix is given by $G_{Block} = I - P^{-1}A$. From our work in Section 4.4.1 we know that the system matrix, A, is an M-matrix for all the cases we consider. Therefore, by Theorem 3.12, the Block Jacobi method converges with $\rho(G_{Block}) < 1$. Since, by the definition (4.9) each D_j is symmetric it follows that P is symmetric. Also P is strictly diagonally dominant. Hence, by Theorem 3.2 it is positive-definite. Therefore P and A are symmetric positive-definite and, with $G_{Block} < 1$ as well, we can guarantee the convergence of the block preconditioned conjugate gradient method.

In order to demonstrate the (strict) block diagonal dominance of A we need to show that

$$\begin{aligned} &\left(\| D_j^{-1} \|\right)^{-1} > \| B_j \| + \| C_j \| \\ &\Longrightarrow 1 > \| D_j^{-1} \| \left(\| B_j \| + \| C_j \|\right), \end{aligned} \tag{4.24}$$

in some norm. The $B_j's$ and $C_j's$ are of the forms given in (4.10) and (4.11) respectively. We use the L_2-norm with

$$\| A \|_2 = max \mid \mu_i \left(A^T A\right) \mid^{\frac{1}{2}}, \tag{4.25}$$

where the μ_i are eigenvalues of A. We observe that the D_j are symmetric and strictly diagonally dominant with $d^j_{ii} > 0$, $d^j_{ik} \leq 0$ $(i, k \in [1, n_\lambda], i \neq k)$ where $D_j = \left\{d^j_{ik}\right\}$. Hence D is positive definite and its eigenvalues are strictly positive by Theorem 3.2. In addition since the system matrix A has property A we expect the eigenvalues of G_{Block} to occur in ± pairs.

In order to get bounds on the norms we use the Gerschgorin Circle Theorem 3.6 and the Block Gerschgorin circle theorem 3.9. The eigenvalues, μ^D of D_j satisfy

$$\mid \mu^D - d^j_{ii} \mid \leq \frac{2}{cos\phi_j \delta\lambda^2}, \tag{4.26}$$

where

$$d^j_{ii} = \frac{2}{cos\phi_j \delta\lambda^2} + \frac{cos\phi_{j+\frac{1}{2}} + cos\phi_{j-\frac{1}{2}}}{\delta\phi^2} + kcos\phi_j. \tag{4.27}$$

Therefore

$$-\tfrac{2}{cos\phi_j \delta\lambda^2} + d^j_{ii} \leq \mu^D \leq d^j_{ii} + \tfrac{2}{cos\phi_j \delta\lambda^2}. \tag{4.28}$$

Hence the smallest eigenvalue of D_j satisfies

$$\mu^D_{min} \geq \frac{cos\phi_{j+\frac{1}{2}} + cos\phi_{j-\frac{1}{2}}}{\delta\phi^2} + kcos\phi_j = \delta_j. \tag{4.29}$$

Since D_j is symmetric positive-definite the eigenvalues of D_j are real and positive allowing us to obtain these bounds. Also

$$\mid\mid D_j \mid\mid_2 = \mu^D_{max}, \tag{4.30}$$

and

$$\begin{aligned} &\mid\mid D_j^{-1} \mid\mid_2 = \tfrac{1}{\mu^D_{min}} \\ &\Longrightarrow \mid\mid D_j^{-1} \mid\mid_2^{-1} = \mu^D_{min} \\ &\Longrightarrow \mid\mid D_j^{-1} \mid\mid_2^{-1} \geq \tfrac{cos\phi_{j+\frac{1}{2}} + cos\phi_{j-\frac{1}{2}}}{\delta\phi^2} + kcos\phi_j. \end{aligned} \tag{4.31}$$

We also have

$$\mid\mid B_j \mid\mid_2 + \mid\mid C_j \mid\mid_2 = \frac{cos\phi_{j+\frac{1}{2}} + cos\phi_{j-\frac{1}{2}}}{\delta\phi^2}. \tag{4.32}$$

Hence

$$\| D_j^{-1} \|_2^{-1} > \| B_j \|_2 + \| C_j \|_2 \,. \tag{4.33}$$

We also need show that the preconditioned iteration matrix, $G_{Block} = I - P^{-1}A$, is convergent (i.e. $\rho(I - P^{-1}A) < 1$). This will establish the convergence of the stationary block preconditioned method. Also we can show that the eigenvalues, $\hat{\mu}$, of $P^{-1}A$ are all clustered in the region $0 < \hat{\mu} < 2$. Let

$$\begin{aligned} &\nu = max_j(\| D_j^{-1} B_j \|_2 + \| D_j^{-1} C_j \|_2) \\ &\leq max_j \left(\| D_j^{-1} \|_2 (\| B_j \|_2 + \| C_j \|_2)\right), \end{aligned} \tag{4.34}$$

From (4.33) we note that $\nu < 1$. Then by Theorem 3.9 the eigenvalues $\hat{\mu}$ of G_{Block} satisfy

$$\| (G_{Block}^{jj} - \hat{\mu} I)^{-1} \|_2^{-1} = \| -\hat{\mu}^{-1} \|_2^{-1} = | \hat{\mu} | \leq \nu, \tag{4.35}$$

for some j. Therefore

$$\rho(I - P^{-1}A) = max \mid \hat{\mu} \mid \leq \nu < 1. \tag{4.36}$$

It follows that the preconditioned iterative method converges and hence so will the associated preconditioned conjugate gradient method. Also since the eigenvalues $\tilde{\mu}$ of $(I - P_{block}^{-1}A)$ equal $1 - \hat{\mu}$ (and since $P_{Block}^{-1}A$ has real positive eigenvalues) we find that $0 < \hat{\mu}(P_{Block}^{-1}A) < 2$.

4.4.4 ADI preconditioner

This preconditioner was introduced in Section 3.4.3. For the particular problems we are considering ((4.2) and (4.3)) we use

$$A = H_\Upsilon + V_\Upsilon, \tag{4.37}$$

with

$$H_\Upsilon = \begin{pmatrix} D_1^H & & & \\ & D_2^H & & \\ & & \ddots & \\ & & & \\ & & & D_{n_\phi}^H \end{pmatrix}, \tag{4.38}$$

where

$$D_j^H = tridiag\left(\begin{array}{ccc} -\frac{1}{cos\phi_j \delta\lambda^2} & \frac{2}{cos\phi_j\delta\lambda^2} + \frac{h^2 k cos\phi_j}{2} & -\frac{1}{cos\phi_j\delta\lambda^2} \end{array} \right) \quad 1 \le j \le n_\phi.$$

With periodic boundary conditions the first line of each D_j^H is of the form

$$\left(\begin{array}{cccccc} \frac{2}{cos\phi_j\delta\lambda^2} + \frac{h^2 k cos\phi_j}{2} & -\frac{1}{cos\phi_j\delta\lambda^2} & 0 & \cdots & 0 & -\frac{1}{cos\phi_j\delta\lambda^2} \end{array} \right),$$

whilst the last row is of the form

$$\left(\begin{array}{cccccc} -\frac{1}{cos\phi_j\delta\lambda^2} & 0 & \cdots & 0 & -\frac{1}{cos\phi_j\delta\lambda^2} & \frac{2}{cos\phi_j\delta\lambda^2} + \frac{h^2 k cos\phi_j}{2} \end{array} \right).$$

Also

$$V_\Upsilon = \begin{pmatrix} D_1^V & C_1 & & & & \\ B_2 & D_2^V & C_2 & & & \\ & B_3 & D_3^V & C_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & & & \\ & & & & D_{n_\phi-1}^V & C_{n_\phi-1} \\ & & & & B_{n_\phi} & D_{n_\phi}^V \end{pmatrix}, \tag{4.39}$$

where the $B_j's$ and $C_j's$ are as defined in (4.10) and (4.11), and

$$D_j^V = diag\left[\frac{cos\phi_{j+\frac{1}{2}} + cos\phi_{j-\frac{1}{2}}}{\delta\phi^2} + \frac{k cos\phi_j}{2} \right] \quad 1 \le j \le n_\phi. \tag{4.40}$$

From (4.38) and (4.39) we observe that the submatrices D_j^H and D_j^V are symmetric. Since we also have from (4.12) that

$$B_j = diag\left[-\frac{cos\phi_{j-\frac{1}{2}}}{\delta\phi^2} \right] = C_{j-1}, \quad 2 \le j \le n_\phi,$$

then we may deduce that H_Υ and V_Υ are symmetric. Also we observe that the connected graphs of H_Υ and V_Υ are strongly connected and therefore by Theorem 3.1 are irreducible. Also H_Υ and V_Υ are diagonally dominant. It is possible to find strict diagonal dominance in at least one row of H_Υ and V_Υ in cases (1) to (3) considered in Section 4.4.1. For these cases we may deduce that H_Υ and V_Υ are irreducibly diagonally dominant. Since we also observe that the diagonal entries of H_Υ and V_Υ are strictly positive it follows, by Theorem 3.2, that H_Υ and V_Υ are positive-definite with strictly positive eigenvalues. Therefore they are Stieltjes matrices. In addition, since we are assuming that $\Upsilon > 0$, it follows that $(H_\Upsilon + \Upsilon I)$ and $(V_\Upsilon + \Upsilon I)$ are Stieltjes matrices as well. Therefore, using Theorem 3.16, we may deduce that the ADI preconditioned method converges if and only if values of $\Upsilon > 0$ are used in the method.

For case (4) in Section 4.4.1, where $k = 0$ and we have periodic boundary conditions in the λ-direction, whilst we have diagonal dominance in every row of H_Υ and V_Υ we cannot show strict diagonal dominance in even one row of H_Υ (we can for V_Υ). Therefore we cannot guarantee the convergence of the ADI method in this case.

4.4.5 ADI preconditioner with spatially varying parameter

This new varying parameter preconditioner was introduced in Section 3.4.4. As stated in that section, this new ADI method differs from the constant parameter ADI method by its use of varying parameters. The matrices H_Υ and V_Υ are unchanged from Section 4.4.4. The matrix analysis performed in the previous Section 4.4.4 follows through in the same way. The spatially varying ADI stationary method converges, with $\rho(G_{ADIV}) < 1$, if and only if values of $\Upsilon_j > 0$ are used in the method.

4.5 Gerschgorin convergence analysis

In this section we use the Gerschgorin Circle Theorem 3.6 to estimate bounds on the eigenvalues and conditioning of the system matrix A and the preconditioned matrices. This analysis will provide us with some qualitative information on the relative convergence properties of our preconditioned methods. The Gerschgorin Circle Theorem 3.6 states that all eigenvalues μ of A are contained in the union of the discs

$$D_i =\mid \mu - a_{ii} \mid \leq \sum_{j=1, j\neq i}^{n} \mid a_{ij} \mid \quad 1 \leq i \leq N. \tag{4.41}$$

Our system matrix A is real, symmetric positive definite, and strictly diagonally dominant with $a_{ii} > 0$, and $a_{ij} \leq 0$ for $i \neq j$. Therefore the eigenvalues of A are real and hence we may write (4.41) in the form

$$-\sum_{j=1, j\neq i}^{n} \mid a_{ij} \mid \leq \mu - a_{ii} \leq \sum_{j=1, j\neq i}^{n} \mid a_{ij} \mid \quad 1 \leq i \leq N.$$

Adding a_{ii} to both sides gives

$$0 \leq a_{ii} - \sum_{j=1, j\neq i}^{n} \mid a_{ij} \mid \leq \mu \leq \sum_{j=1}^{N} \mid a_{ij} \mid \quad 1 \leq i \leq N.$$

The right hand bound is the upper bound we use to obtain an estimate of the spectral radius ρ of A. We use the left hand bound to obtain an estimate of the minimum eigenvalue. i.e. we assume that if μ is an eigenvalue of A then

$$\mu_{min} \geq min \left\{ a_{ii} - \sum_{j=1, j\neq i}^{n} \mid a_{ij} \mid \right\} \quad 1 \leq i \leq N. \tag{4.42}$$

We use the upper bound estimate of the spectral radius of A and the lower bound estimate of the minimum eigenvalue of A to derive an upper bound for the 2-norm condition number, $\kappa_2(A)$.

In the analysis of this section we assume that $k > 0$ and that the horizontal stepsizes are equal, with $\delta\lambda = \delta\phi = h$. The analysis is the same for Dirichlet or periodic boundary conditions in the λ-direction. Also we make use of the trigonometric identity

$$\begin{aligned} cos\phi_{j+\frac{1}{2}} + cos\phi_{j-\frac{1}{2}} &\equiv cos\phi_j cos\phi_{\frac{1}{2}} - sin\phi_j sin\phi_{\frac{1}{2}} \\ &\quad + cos\phi_j cos\phi_{\frac{1}{2}} + sin\phi_j sin\phi_{\frac{1}{2}} \\ &\equiv 2cos\phi_j cos\phi_{\frac{1}{2}} \end{aligned} \tag{4.43}$$

to simplify our equations. In addition we will assume that we can bound the $cos\phi$ values by $cos\phi \leq cos\phi_{High}$ where $cos\phi_{High} << 1$. Also we will assume that we can take low latitude $cos\phi$ values to be approximately 1 in the calculated bounds i.e $cos\phi_{\frac{1}{2}}, cos\phi_1, cos\phi_2 \approx 1$.

4.5.1 Bound on $\rho(A)$ and estimate of $\kappa_2(A)$

We firstly use the Gerschgorin Theorem 3.6 3.6 to estimate a bound for the spectral radius ρ of the system matrix A. In the problem we are considering the maximum row sum in moduli is given by

$$max \left\{ \begin{array}{ll} \frac{4}{cos\phi_1} + 2cos\phi_1 cos\phi_{\frac{1}{2}} + cos\phi_{1+\frac{1}{2}} + h^2 k cos\phi_1 & j = 1 \\ & \\ \frac{4}{cos\phi_j} + 4cos\phi_j cos\phi_{\frac{1}{2}} + h^2 k cos\phi_j & 2 \leq j \leq n_\phi - 1 \\ & \\ \frac{4}{cos\phi_{n_\phi}} + 2cos\phi_{n_\phi} cos\phi_{\frac{1}{2}} + cos\phi_{n_\phi - \frac{1}{2}} + h^2 k cos\phi_{n_\phi} & j = n_\phi \end{array} \right\}.$$

We observe that the terms involving $\frac{4}{cos\phi}$ dominate. Therefore the row sums become larger as $cos\phi$ tends to zero i.e. at higher row numbers. Therefore we take the bound

$$\rho(A) \leq \frac{4}{cos\phi_{High}} + 3cos\phi_{High} + h^2kcos\phi_{High}.$$

on the spectral radius of A. We also use equation (4.42) to estimate a bound on the minimum eigenvalue of A. This is given by

$$\mu_{min}(A) \geq min \left\{ \begin{array}{ll} cos\phi_{1-\frac{1}{2}} + h^2kcos\phi_1 & j = 1 \\ h^2kcos\phi_j & 2 \leq j \leq n_\phi - 1 \\ cos\phi_{n_\phi+\frac{1}{2}} + h^2kcos\phi_{n_\phi} & j = n_\phi \end{array} \right\}. \tag{4.44}$$

The smallest values occur as $cos\phi$ tends to zero. We therefore have

$$\mu_{min} \geq h^2kcos\phi_{High}. \tag{4.45}$$

From this we can deduce that

$$\begin{aligned} \kappa_2(A) &= \frac{\rho(A)}{\mu_{min}(A)} \\ &= \frac{\frac{4}{cos\phi_{High}} + 3cos\phi_{High} + h^2kcos\phi_{High}}{h^2kcos\phi_{High}} \\ &= \frac{4}{h^2kcos^2\phi_{High}} + \frac{3}{h^2k} + 1 \end{aligned} \tag{4.46}$$

4.5.2 Diagonal preconditioner

In this section we will estimate a bound on the spectral radius of the Jacobi iteration matrix

$$G_D = I - P^{-1}A,$$

a bound on the spectral radius of the preconditioned system matrix $P^{-1}A$, and an estimate of the condition number $\kappa(P^{-1}A)$ of the preconditioned system.

We consider a diagonal preconditioner where

$$P = diag\left[\frac{2}{cos\phi_j} + cos\phi_{j+\frac{1}{2}} + cos\phi_{j-\frac{1}{2}} + h^2 k cos\phi_j\right] \quad 1 \leq j \leq n_\phi$$

$$\Longrightarrow P^{-1} = diag\left[\frac{cos\phi_j}{2+2cos^2\phi_j cos\phi_{\frac{1}{2}} + h^2 k cos^2\phi_j}\right] \quad 1 \leq j \leq n_\phi.$$

It follows that $P^{-1}A$ is of the form

$$P^{-1}A = \begin{pmatrix} \hat{D}_1 & \hat{C}_1 & & & & \\ \hat{B}_2 & \hat{D}_2 & \hat{C}_2 & & & \\ & \hat{B}_3 & \hat{D}_3 & \hat{C}_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & & & \\ & & & & \hat{D}_{n_\phi-1} & \hat{C}_{n_\phi-1} \\ & & & & \hat{B}_{n_\phi} & \hat{D}_{n_\phi} \end{pmatrix},$$

where

$$\hat{D}_j = tridiag\left(-\frac{1}{2+2cos^2\phi_j cos\phi_{\frac{1}{2}} + h^2 k cos^2\phi_j} \quad 1 \quad -\frac{1}{2+2cos^2\phi_j cos\phi_{\frac{1}{2}} + h^2 k cos^2\phi_j} \right).$$

With periodic boundary conditions in the λ direction the first row is of the form

$$\left(1 \quad -\frac{1}{2+2cos^2\phi_j cos\phi_{\frac{1}{2}} + h^2 k cos^2\phi_j} \quad 0 \quad \cdots \quad 0 \quad -\frac{1}{2+2cos^2\phi_j cos\phi_{\frac{1}{2}} + h^2 k cos^2\phi_j} \right),$$

whilst the last row is of the form

$$\left(-\frac{1}{2+2cos^2\phi_j cos\phi_{\frac{1}{2}} + h^2 k cos^2\phi_j} \quad 0 \quad \cdots \quad 0 \quad -\frac{1}{2+2cos^2\phi_j cos\phi_{\frac{1}{2}} + h^2 k cos^2\phi_j} \quad 1 \right).$$

Also

$$\hat{B}_j = diag\left[\frac{-cos\phi_{j+\frac{1}{2}} cos\phi_j}{2+2cos^2\phi_j cos\phi_{\frac{1}{2}} + h^2 k cos^2\phi_j}\right] \quad 2 \leq j \leq n_\phi,$$

$$\hat{C}_j = diag\left[\frac{-cos\phi_{j-\frac{1}{2}} cos\phi_j}{2+2cos^2\phi_j cos\phi_{\frac{1}{2}} + h^2 k cos^2\phi_j}\right] \quad 1 \leq j \leq n_\phi - 1.$$

From this we can form the Jacobi matrix $G = I - P^{-1}A$. We may again use the maximum of the row sums of the moduli of the entries of G to find an upper bound for $\rho(G)$. This is given by

$$\max\left\{\begin{array}{ll} \frac{2+cos\phi_{1+\frac{1}{2}}cos\phi_1}{2+2cos^2\phi_1 cos\phi_{\frac{1}{2}}+h^2kcos^2\phi_1} & j=1 \\ \\ \frac{2+2cos^2\phi_j cos\phi_{\frac{1}{2}}}{2+2cos^2\phi_j cos\phi_{\frac{1}{2}}+h^2kcos^2\phi_j} & 2 \leq j \leq n_\phi - 1 \\ \\ \frac{2+cos\phi_{n_\phi-\frac{1}{2}}cos\phi_{n_\phi}}{2+2cos^2\phi_{n_\phi} cos\phi_{\frac{1}{2}}+h^2kcos^2\phi_{n_\phi}} & j = n_\phi \end{array}\right\}.$$

The maximum value occurs when the difference between the numerator and denominator is smallest. This occurs when $cos\phi$ is as small as possible. We therefore have the bound

$$\rho(G) \leq \frac{2+2cos^2\phi_{High}}{2+2cos^2\phi_{High}+h^2kcos^2\phi_{High}} \tag{4.47}$$

and hence $\rho(G) < 1$.

Next we use the Gerschgorin Theorem 3.6 to find a bound on the maximum eigenvalue of $P^{-1}A$. This is given by

$$\rho(P^{-1}A) \leq max\left\{\begin{array}{ll} \frac{4+(2cos\phi_{1+\frac{1}{2}}+cos\phi_{1-\frac{1}{2}})cos\phi_1}{2+2cos^2\phi_1 cos\phi_{\frac{1}{2}}+h^2kcos^2\phi_1} & j=1 \\ \\ \frac{4+4cos^2\phi_j cos\phi_{\frac{1}{2}}}{2+2cos^2\phi_j cos\phi_{\frac{1}{2}}+h^2kcos^2\phi_j} & 2 \leq j \leq n_\phi - 1 \\ \\ \frac{4+(cos\phi_{n_\phi+\frac{1}{2}}+2cos\phi_{n_\phi-\frac{1}{2}})cos\phi_{n_\phi}}{2+cos^2\phi_{n_\phi} cos\phi_{\frac{1}{2}}+h^2kcos^2\phi_{n_\phi}} & j = n_\phi \end{array}\right\}.$$

The largest value occurs when the difference between the numerator and denominator is smallest. This occurs when $cos\phi$ is as small as possible. We therefore have the bound

$$\rho(P^{-1}A) \leq \frac{4+4cos^2\phi_{High}}{2+2cos^2\phi_{High}+h^2kcos^2\phi_{High}}. \tag{4.48}$$

We also use the Gerschgorin Theorem 3.6 to find a bound on the minimum eigenvalue of $P^{-1}A$. In this case we require that

$$\mu_{min}(P^{-1}A) \geq min \left\{ \begin{array}{ll} \frac{(cos\phi_{1-\frac{1}{2}}+h^2kcos\phi_1)cos\phi_1}{2+2cos^2\phi_1 \cos\phi_1+h^2kcos^2\phi_1} & j=1 \\ \\ \frac{h^2kcos^2\phi_j}{2+2cos^2\phi_j cos\phi_{\frac{1}{2}}+h^2kcos^2\phi_j} & 2 \leq j \leq n_\phi - 1 \\ \\ \frac{(cos\phi_{n_\phi+\frac{1}{2}}+h^2kcos\phi_{n_\phi})cos\phi_{n_\phi}}{2+2cos\phi_{n_\phi} cos\phi_{\frac{1}{2}}+h^2kcos^2\phi_{n_\phi}} & j=n_\phi \end{array} \right\}.$$

The minimum occurs when $cos\phi$ is as small as possible. Therefore we have the bound

$$\mu_{min}(P^{-1}A) \geq \frac{h^2kcos^2\phi_{High}}{2+2cos^2\phi_{High}cos\phi_{\frac{1}{2}}+h^2kcos^2\phi_{High}}. \tag{4.49}$$

Hence

$$\begin{aligned} \kappa_2(P^{-1}A) &= \frac{\left(\frac{4+4cos^2\phi_{High}}{2+2cos^2\phi_{High}+h^2kcos^2\phi_{High}}\right)}{\left(\frac{h^2kcos^2\phi_{High}}{2+2cos^2\phi_{High}+h^2kcos^2\phi_{High}}\right)} \\ &= \frac{4}{h^2kcos^2\phi_{High}} + \frac{4cos\phi_{High}}{h^2k} \end{aligned} \tag{4.50}$$

Comparing (4.50) with (4.46) we observe that for the diagonal preconditioner

$$\kappa_2(P^{-1}A) < \kappa_2(A).$$

4.5.3 Block diagonal preconditioner

In this section we will estimate a bound on the spectral radius of the block Jacobi iteration matrix

$$G_{Block} = I - P^{-1}A,$$

a bound on the spectral radius of the preconditioned system matrix $P^{-1}A$ and an estimate of the condition number $\kappa(P^{-1}A)$ of the preconditioned system.

We consider a block preconditioner where

$$P^{-1}A = \begin{pmatrix} I & D_1^{-1}C_1 & & & & \\ D_2^{-1}B_2 & I & D_2^{-1}C_2 & & & \\ & D_3^{-1}B_3 & I & D_3^{-1}C_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & & I & D_{n_\phi-1}^{-1}C_{n_\phi-1} \\ & & & & D_{n_\phi}^{-1}B_{n_\phi} & I \end{pmatrix}.$$

From section 3.4.2 we know that for

$$G_{Block} = I - P^{-1}A,$$

if μ_i^D is an eigenvalue of P and μ_i is an eigenvalue of A, then the spectrum σ of the eigenvalues of G_{Block} is given by

$$\sigma(G_{Block}) = \left\{ \ \tfrac{\mu_i^D - \mu_i}{\mu_i^D} \qquad 1 \le i \le n, \right.$$

and therefore the spectral radius is given by

$$\rho(G_{Block}) = max \left\{ \mid \frac{\mu_i^D - \mu_i}{\mu_i^D} \mid \right\} \quad 1 \le i \le n.$$

We know from Section 4.5.1 that

$$\mu \le \rho(A) = \frac{4}{cos\phi_{n_\phi}} + 2cos\phi_{n_\phi}cos\phi_{\frac{1}{2}} + cos\phi_{n_\phi - \frac{1}{2}} + h^2kcos\phi_{n_\phi},$$

and that

$$\mu \ge \mu_{min}(A) = h^2kcos\phi_{n_\phi - 1}.$$

From the definition of D_j and using the Gerschgorin Theorem 3.6 we deduce that

$$\rho(\mu_{ij}^D) \le max \left\{ \frac{4}{cos\phi_j} + cos\phi_{j+\frac{1}{2}} + cos\phi_{j-\frac{1}{2}} + h^2kcos\phi_j \quad 1 \le i \le n_\lambda, 1 \le j \le n_\phi \right\},$$

and that

$$\mu_{ij}^{D} \geq min\left\{cos\phi_{j+\frac{1}{2}} + cos\phi_{j-\frac{1}{2}} + h^2 kcos\phi_j \quad 1 \leq i \leq n_\lambda, 1 \leq j \leq n_\phi\right\}.$$

Therefore

$$\rho(G_{Block}) = max\left\{\mid \frac{\mu_{ij}^{D} - \mu_{ij}}{\mu_{ij}^{D}} \mid\right\} \quad 1 \leq i \leq n_\lambda, 1 \leq j \leq n_\phi$$

$$= max\left\{\begin{array}{ll} \frac{cos\phi_{1+\frac{1}{2}}+cos\phi_{1-\frac{1}{2}}+h^2kcos\phi_1-cos\phi_{1-\frac{1}{2}}-h^2kcos\phi_1}{cos\phi_{1+\frac{1}{2}}+cos\phi_{1-\frac{1}{2}}+h^2kcos\phi_1} & j=1 \\ \\ \frac{cos\phi_{j+\frac{1}{2}}+cos\phi_{j-\frac{1}{2}}+h^2kcos\phi_j-h^2kcos\phi_j}{cos\phi_{j+\frac{1}{2}}+cos\phi_{j-\frac{1}{2}}+h^2kcos\phi_j} & 2 \leq j \leq n_\phi - 1 \\ \frac{cos\phi_{n_\phi+\frac{1}{2}}+cos\phi_{n_\phi-\frac{1}{2}}+h^2kcos\phi_{n_\phi}-cos\phi_{n_\phi+\frac{1}{2}}-h^2kcos\phi_{n_\phi}}{cos\phi_{n_\phi+\frac{1}{2}}+cos\phi_{n_\phi-\frac{1}{2}}+h^2kcos\phi_{n_\phi}} & j = n_\phi \end{array}\right\}$$

$$= max\left\{\begin{array}{ll} \frac{cos\phi_{1+\frac{1}{2}}}{2cos\phi_1 cos\phi_{\frac{1}{2}}+h^2kcos\phi_1} & j=1 \\ \\ \frac{2cos\phi_j cos\phi_{\frac{1}{2}}}{2cos\phi_j cos\phi_{\frac{1}{2}}+h^2kcos\phi_j} & 2 \leq j \leq n_\phi - 1 \\ \\ \frac{cos\phi_{n_\phi-\frac{1}{2}}}{2cos\phi_{n_\phi} cos\phi_{\frac{1}{2}}+h^2kcos\phi_{n_\phi}} & j = n_\phi \end{array}\right\}.$$

The maximum value occurs jointly in block rows where $2 \leq j \leq n_\phi - 1$. We therefore have

$$\rho(G_{Block}) \leq \frac{2cos\phi_{\frac{1}{2}}}{2cos\phi_{\frac{1}{2}} + h^2k} \approx \frac{2}{2+h^2k}. \tag{4.51}$$

and therefore $\rho(G_{Block}) < 1$.

We know from the work of Section 4.2 that A is symmetric and positive definite. It is therefore a Stieltjes matrix. Since the diagonal entries are strictly positive whilst the off-diagonal entries are non-positive we may deduce, by Theorem 3.3, that $A^{-1} > 0$. We apply this information to Theorem 3.12. We have that $A^{-1} > 0$ and therefore in this case we may guarantee that we have

$$\rho(G_{Block}) < \rho(G_D) < 1.$$

We now attempt to find bounds on the maximum and minimum eigenvalues of $(P^{-1}A)$. The maximum is bounded by

$$\rho(P^{-1}A) \leq max \left\{ \begin{array}{ll} \frac{\frac{4}{cos\phi_1}+2cos\phi_1 cos\phi_{\frac{1}{2}}+cos\phi_{1+\frac{1}{2}}+h^2kcos\phi_1}{\frac{4}{cos\phi_1}+2cos\phi_1 cos\phi_{\frac{1}{2}}+h^2kcos\phi_1} & j=1 \\ \\ \frac{\frac{4}{cos\phi_j}+4cos\phi_j cos\phi_{\frac{1}{2}}+h^2kcos\phi_j}{\frac{4}{cos\phi_j}+2cos\phi_j cos\phi_{\frac{1}{2}}+h^2kcos\phi_j} & 2 \leq j \leq n_\phi - 1 \\ \\ \frac{\frac{4}{cos\phi_{n_\phi}}+2cos\phi_{n_\phi} cos\phi_{\frac{1}{2}}+cos\phi_{n_\phi-\frac{1}{2}}+h^2kcos\phi_{n_\phi}}{\frac{4}{cos\phi_{n_\phi}}+2cos\phi_{n_\phi} cos\phi_{\frac{1}{2}}+h^2kcos\phi_{n_\phi}} & j = n_\phi \end{array} \right\}.$$

The maximum value occurs in block rows where $j = 2$ giving

$$\rho(P^{-1}A) \leq \frac{\frac{4}{cos\phi_2}+4cos\phi_2 cos\phi_{\frac{1}{2}}+h^2kcos\phi_2}{\frac{4}{cos\phi_2}+2cos\phi_2 cos\phi_{\frac{1}{2}}+h^2kcos\phi_2} \approx \frac{8+h^2k}{6+h^2k}.$$

The minimum eigenvalue is bounded by

$$\mu_{min}(P^{-1}A) \geq min \left\{ \begin{array}{ll} \frac{cos\phi_{1-\frac{1}{2}}+h^2kcos\phi_1}{2cos\phi_1 cos\phi_{\frac{1}{2}}+h^2kcos\phi_1} & j=1 \\ \\ \frac{h^2kcos\phi_j}{2cos\phi_j cos\phi_{\frac{1}{2}}+h^2kcos\phi_j} & 2 \leq j \leq n_\phi - 1 \\ \\ \frac{cos\phi_{n_\phi+\frac{1}{2}}+h^2kcos\phi_{n_\phi}}{2cos\phi_{n_\phi} cos\phi_{\frac{1}{2}}+h^2kcos\phi_{n_\phi}} & j = n_\phi \end{array} \right\}$$

The minimum value occurs in block rows $2 \leq j \leq n_\phi - 1$ giving

$$\mu_{min}(P^{-1}A) \geq \frac{h^2k}{2cos\phi_{\frac{1}{2}}+h^2k} \approx \frac{h^2k}{2+h^2k}.$$

Hence

$$\kappa(P^{-1}A) \leq \frac{\left(\frac{8+h^2k}{6+h^2k}\right)}{\left(\frac{h^2k}{2+h^2k}\right)}.$$

(4.52)

4.5.4 ADI preconditioner

We now derive the conditioning and spectral radii estimates for the ADI preconditioned case. Recall that the splitting for the ADI preconditioner is of the form

$$A = H_\Upsilon + V_\Upsilon,$$

with

$$H_\Upsilon = \begin{pmatrix} D_1^H & & & \\ & D_2^H & & \\ & & \ddots & \\ & & & \\ & & & D_{n_\phi}^H \end{pmatrix}, \tag{4.53}$$

where

$$D_j^H = tridiag\left(\begin{array}{ccc} -\frac{1}{cos\phi_j} & \frac{2}{cos\phi_j} + \frac{h^2 k cos\phi_j}{2} & -\frac{1}{cos\phi_j} \end{array} \right) \quad 1 \leq j \leq n_\phi.$$

With periodic boundary conditions the first line of each D_j^H is of the form

$$\left(\begin{array}{cccccc} \frac{2}{cos\phi_j} + \frac{h^2 k cos\phi_j}{2} & -\frac{1}{cos\phi_j} & 0 & \cdots & 0 & -\frac{1}{cos\phi_j} \end{array} \right),$$

whilst the last row is of the form

$$\left(\begin{array}{cccccc} -\frac{1}{cos\phi_j} & 0 & \cdots & 0 & -\frac{1}{cos\phi_j} & \frac{2}{cos\phi_j} + \frac{h^2 k cos\phi_j}{2} \end{array} \right).$$

Also

$$V_\Upsilon = \begin{pmatrix} D_1^V & C_1 & & & & \\ B_2 & D_2^V & C_2 & & & \\ & B_3 & D_3^V & C_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & & & \\ & & & & D_{n_\phi-1}^V & C_{n_\phi-1} \\ & & & & B_{n_\phi} & D_{n_\phi}^V \end{pmatrix}, \tag{4.54}$$

where the $B_j's$ and $C_j's$ are as defined in (4.10) and (4.11), and

$$D_j^V = diag\left[cos\phi_{j+\frac{1}{2}} + cos\phi_{j-\frac{1}{2}} + \frac{h^2kcos\phi_j}{2}\right] \quad 1 \leq j \leq n_\phi. \tag{4.55}$$

The optimum value of Υ that should be used in the preconditioner is given by

$$\Upsilon = \sqrt{\alpha\beta}. \tag{4.56}$$

Let μ_i^H and μ_i^V, $1 \leq i \leq N$, be the eigenvalues of the matrices H_Υ and V_Υ respectively. Using the Gerschgorin Theorem 3.6 we observe that

$$min(\mu_i^H) = min(\mu_i^V) \geq \frac{h^2kcos\phi_{n_\phi}}{2}. \tag{4.57}$$

Therefore we take $\alpha = \frac{h^2kcos\phi_{n_\phi}}{2}$. For the upper bound, β, we again use the Gerschgorin theorem. We observe that

$$\begin{aligned} &\rho(V_\Upsilon) \leq \rho(H_\Upsilon) \leq \max\left\{\tfrac{4}{cos\phi_j} + \tfrac{h^2kcos\phi_j}{2}\right\} \quad 1 \leq j \leq n_\phi \\ &= \tfrac{4}{cos\phi_{n_\phi}} + \tfrac{h^2kcos\phi_{n_\phi}}{2}. \end{aligned} \tag{4.58}$$

Therefore we have

$$\begin{aligned} \Upsilon &= \sqrt{\frac{h^2kcos\phi_{n_\phi}}{2}(\frac{4}{cos\phi_{n_\phi}} + \frac{h^2kcos\phi_{n_\phi}}{2})} \\ &= \sqrt{2h^2k + \frac{h^4k^2cos^2\phi_{n_\phi}}{4}}. \end{aligned} \tag{4.59}$$

Recall that the spectrum of the eigenvalues of the ADI iteration matrix, G_{ADI}, is given by

$$\left|\frac{\Upsilon - \mu_i^H}{\Upsilon + \mu_i^H}\right| \cdot \left|\frac{\Upsilon - \mu_i^V}{\Upsilon + \mu_i^V}\right|. \tag{4.60}$$

Using the Gerschgorin bounds on the eigenvalues μ_i^H and μ_i^V in (4.57) and (4.58), we find that the spectral radius of G_{ADI} is given by

$$\rho(G_{ADI}) \leq \frac{\Upsilon - \frac{h^2kcos\phi_{High}}{2}}{\Upsilon + \frac{h^2kcos\phi_{High}}{2}} < 1, \tag{4.61}$$

which guarantees the convergence of the PCG method with ADI preconditioner. Also, given that

$$G_{ADI} = I - P^{-1}A \Longrightarrow P^{-1}A = I - G_{ADI}, \tag{4.62}$$

we can write the spectrum of the ADI preconditioned matrix $P_{ADI}^{-1}A$ in the form

$$1 - \left| \frac{\Upsilon - \mu_i^H}{\Upsilon + \mu_i^H} \right| \cdot \left| \frac{\Upsilon - \mu_i^V}{\Upsilon + \mu_i^V} \right| .$$

Again using the Gerschgorin bounds on the eigenvalues μ_i^H and μ_i^V calculated in (4.57) and (4.58), we find that the spectral radius of $P_{ADI}^{-1}A$ is given by

$$\rho(P^{-1}A) \leq 1 - \frac{\Upsilon - \frac{4}{cos\phi_{High}} - \frac{h^2kcos\phi_{High}}{2}}{\Upsilon + \frac{4}{cos\phi_{High}} + \frac{h^2kcos\phi_{High}}{2}} . \frac{\Upsilon - 4cos\phi_{High} - \frac{h^2kcos\phi_{High}}{2}}{\Upsilon + 4cos\phi_{High} + \frac{h^2kcos\phi_{High}}{2}}, \tag{4.63}$$

whilst the minimum eigenvalue of $P_{ADI}^{-1}A$ is given by

$$\mu_{min}(P^{-1}A) \geq 1 - \frac{\Upsilon - \frac{h^2kcos\phi_{High}}{2}}{\Upsilon + \frac{h^2kcos\phi_{High}}{2}} . \frac{\Upsilon - \frac{h^2kcos\phi_{High}}{2}}{\Upsilon + \frac{h^2kcos\phi_{High}}{2}} . \tag{4.64}$$

The condition number of $P_{ADI}^{-1}A$ is therefore given by

$$= \frac{1 - \frac{\Upsilon - \frac{4}{cos\phi_{High}} - \frac{h^2kcos\phi_{High}}{2}}{\Upsilon + \frac{4}{cos\phi_{High}} + \frac{h^2kcos\phi_{High}}{2}} . \frac{\Upsilon - 4cos\phi_{High} - \frac{h^2kcos\phi_{High}}{2}}{\Upsilon + 4cos\phi_{High} + \frac{h^2kcos\phi_{High}}{2}}}{1 - \frac{\Upsilon - \frac{h^2kcos\phi_{High}}{2}}{\Upsilon + \frac{h^2kcos\phi_{High}}{2}} . \frac{\Upsilon - \frac{h^2kcos\phi_{High}}{2}}{\Upsilon + \frac{h^2kcos\phi_{High}}{2}}} . \tag{4.65}$$

4.5.5 ADI preconditioner with spatially varying parameter

In this section we derive the parameters for the ADI spatially varying parameter preconditioner that we will use in the numerical experiments in Chapter 5. The optimum value of Υ_j that should be used in the preconditioner is given by

$$\Upsilon_j = \sqrt{\alpha_j \beta_j} \tag{4.66}$$

Applying this to our spherical domain problems, and using the Gerschgorin Theorem 3.6 we observe that

$$rowsum(H_\Upsilon) \geq \mu_{min}(V_\Upsilon) \geq \frac{h^2 kcos\phi_{n_\phi}}{2}. \tag{4.67}$$

Therefore we take $\alpha_j = \frac{h^2 kcos\phi_{n_\phi}}{2}$. For the upper bound, β we observe that

$$\mid rowsum(V_\Upsilon) \mid \leq \mid rowsum(H_\Upsilon) \mid \leq \frac{4}{cos\phi_j} + \frac{h^2 kcos\phi_j}{2} \quad 1 \leq j \leq n_\phi. \tag{4.68}$$

Therefore we have

$$\Upsilon_j = \sqrt{\frac{h^2 kcos\phi_{n_\phi}}{2}(\frac{4}{cos\phi_j} + \frac{h^2 kcos\phi_j}{2})}. \tag{4.69}$$

4.5.6 Comparison of preconditioners

Tables 4.1 and 4.2 summarise the Gerschgorin analysis of this section. We observe that the dominant term in the condition number estimates of A and $P^{-1}A$, where P is the diagonal preconditioner, is of the order $\frac{1}{cos^2\phi_{High}}$. The block diagonal preconditioner does not vary with $cos\phi_{High}$ which illustrates its improvement on the diagonal preconditioner. It is very difficult to see the merits of the ADI preconditioner by looking at the general forms given in Tables 4.1 and 4.2. This is much clearer for the typical values given in those tables. These values are for the case with $h = 1.0^o, \phi_{NB} = 88^o$. We can see that the condition number of the ADI preconditioned system is well over an order of magnitude smaller than that of the Block preconditioned system which in turn is well over an order of magnitude smaller than the diagonal preconditioned system. From Table 4.2 we observe that $\rho(G)$ is smallest for the ADI preconditioner followed by the Block preconditioner and then the diagonal preconditioner. Both trends lead us to conclude that we would expect the ADI preconditioner to yield the fastest rates of convergence in

Prec.	$\kappa(P^{-1}A)$	Typ.Val.
None	$\frac{4}{h^2kcos^2\phi_{High}} + \frac{3}{h^2k} + 1$	6.91×10^8
D	$\frac{4}{h^2kcos^2\phi_{High}} + \frac{4cos\phi_{High}}{h^2k}$	6.91×10^8
Block D	$\frac{\frac{8+h^2k}{6+h^2k}}{\frac{h^2k}{2+h^2k}}$	8.75×10^5
ADI	$\frac{1-\frac{\Upsilon-\frac{4}{cos\phi_{High}}-\frac{h^2kcos\phi_{High}}{2}}{\Upsilon+\frac{4}{cos\phi_{High}}+\frac{h^2kcos\phi_{High}}{2}}\cdot\frac{\Upsilon-4cos\phi_{High}-\frac{h^2kcos\phi_{High}}{2}}{\Upsilon+4cos\phi_{High}+\frac{h^2kcos\phi_{High}}{2}}}{1-\frac{\Upsilon-\frac{h^2kcos\phi_{High}}{2}}{\Upsilon+\frac{h^2kcos\phi_{High}}{2}}\cdot\frac{\Upsilon-\frac{h^2kcos\phi_{High}}{2}}{\Upsilon+\frac{h^2kcos\phi_{High}}{2}}}$	1.53×10^4

Table 4.1: 2-norm condition numbers of $\kappa(P^{-1}A)$ calculated theoretically using Gerschgorin

Prec.	$\rho(G_P)$	Typ.Val.
D	$\frac{2+2cos^2\phi_{High}}{2+2cos^2\phi_{High}+h^2kcos^2\phi_{High}}$	1.0 - 7×10^{-8}
Block	$\frac{2}{2+h^2k}$	1.0 - 2×10^{-5}
ADI	$\frac{\Upsilon-\frac{h^2kcos\phi_{n_\phi}}{2}}{\Upsilon+\frac{h^2kcos\phi_{n_\phi}}{2}}$	1.0 - 6×10^{-4}

Table 4.2: Spectral radii of the iteration matrix G calculated theoretically using Gerschgorin

our numerical experiments followed by the Block preconditioner with the diagonal preconditioner yielding the slowest rates of convergence. We will test this hypothesis in the numerical experiments of Chapter 5.

4.6 Summary

This chapter introduced the spherical model we use to approximate the anisotropic elliptic problems encountered in the barotropic solvers. We initially consider the constant depth problem in both limited area and periodic domain cases. We confirmed the validity of our discretisation scheme using truncation error analysis and derived the continuous and discrete eigenvalues and eigenvectors of the spherical Laplacian and hence those of the Helmholtz problem. The convergence of our preconditioned Conjugate Gradient method for this problem, using the proposed preconditioners, was checked. The preconditioners were assessed, with respect to their likely effect on speeds of convergence and the anisotropy, using Gerschgorin analysis. It was found that an ADI preconditioner was likely to be slightly better than using a Block diagonal preconditioner. Both were predicted to be much better than the diagonal preconditioner.

Chapter 5

Spherical domain model : numerical experiments

5.1 Introduction

We now present numerical results for our constant depth spherical model obtained using MATLAB V.6. A Limited Area problem over a segment of idealised Northern Hemisphere ocean is initially studied in Section 5.2. We then move on, in Section 5.3, to consider a periodic model with a hemisphere wide domain in the longitudinal (λ) direction. In both experiments the effect of varying the northern boundary is considered with, in some cases, the domain extending very close to the north pole. Finally in Section 5.4, the limited area problem is revised with the domain extended up to include the north pole (as a polar island) and the effects of using different Fourier modes as initial estimates are studied. The Preconditioned Conjugate Gradient method is used throughout, to iteratively solve the problems with diagonal, block diagonal and ADI preconditioners. In all three sections we first state the problem being considered and demonstrate some numerical properties of the system matrices, A. We then present some results on the numerical properties of the preconditioned methods. We will show how the mesh anisotropy of the elliptic operators considered affects the convergence (particularly in polar regions) of the diagonal preconditioned CG method, and show whether the other preconditioners improve this issue. This will be done using analysis of the eigenstructure of the preconditioned methods. We will finally, for each problem, compare the overall convergence speeds of the various preconditioned methods using practical convergence experiments with specific choices for the source functions (which contribute to the **b** terms in (4.6)).

5.2 Limited area experiments

5.2.1 Problem formulation

Recall that the formulation for the constant depth, limited area, spherical Helmholtz problem is given by

$$\begin{cases} -\frac{1}{cos\phi}\left[\frac{\partial}{\partial\lambda}\left(\frac{1}{cos\phi}\frac{\partial U}{\partial\lambda}\right)+\frac{\partial}{\partial\phi}\left(cos\phi\frac{\partial U}{\partial\phi}\right)\right]+kU=\gamma(\lambda,\phi) \\ \lambda\in(0^oE,30^oE) \quad \phi\in(10^oN,\phi_{NB}) \\ U(0^oE,\phi)=0,U(30^oE,\phi)=0 \\ U(\lambda,10^oN)=0,U(\lambda,\phi_{NB})=0 \\ \phi_{NB}\in(40^oN,89.5^oN). \end{cases} \tag{5.1}$$

The main aims of the experiments of this section are to assess the numerical effects of extending the northern boundary of the domain towards the north pole, and to investigate the efficiency of various preconditioners in resolving the resulting increased anisotropy. Discrete stepsizes of 2^0, 1^0 and $\frac{1}{2}^o$ are used in the experiments (i.e. the stepsizes are equal in both directions with

	$\phi_{NB} = 40^o N$	$\phi_{NB} = 88^o N$
$a_{1,1}$	1.313×10^4	1.313×10^4
$a_{N,N}$	1.355×10^4	1.256×10^5
$\kappa_\infty(D_1)$	3.076	3.076
$\kappa_\infty(D_{n_\phi})$	4.312	296.450

Table 5.1: Variation of matrix properties with anisotropy, $h = 1^o$, $k = 0.01$.

$\delta\lambda = \delta\phi = h = 2^o, 1^o, \frac{1}{2}^o$). The discrete mesh is of size n_λ by n_ϕ with the total number of grid points (and hence the size of A) given by $n_\lambda \times n_\phi$.

The mesh anisotropy of the operator is demonstrated in Table 5.1. We observe how the size of the diagonal elements of A changes very little with a 'low' anisotropy case such as $\phi_{NB} = 40^o$ whereas it can vary by an order of magnitude in a more mesh anisotropic case such as $\phi_{NB} = 88^o$. Also the conditioning of the diagonal blocks, D_j, of A is affected much more by considering a more anisotropic problem.

Our assertion that the anisotropy affects the spectral radii and conditioning of the unpreconditioned problem is confirmed in Tables 5.2 to 5.3. Both become larger, to just over an order of magnitude, by the movement of the northern boundary towards the pole. Both also become larger with smaller stepsizes as expected. The conditioning results using the 2-norm may be found in Appendix B.

The effect on the leading eigenvectors of A is illustrated by figures 5.1 and 5.2. For $\phi_{NB} = 88^o$N the largest components of the eigenmode are found in the polar regions, at the very edge of the northern boundary. With $\phi_{NB} = 40^o$N the components are more evenly spread about the domain and are smaller. The movement of the boundary towards the pole has the effect of increasing the largest components of the leading eigenmode and clustering

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^o N	1.072×10^5	2.657×10^4	6.552×10^3
70^o N	1.635×10^5	3.970×10^4	9.433×10^3
88^o N	1.205×10^6	2.506×10^5	4.666×10^4
89^o N	2.006×10^6	3.755×10^5	NA
89.5^o N	3.008×10^6	NA	NA

Table 5.2: Spectral Radii of system matrix A, $k = 0.01$

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^o N	2.186×10^3	544.176	134.167
70^o N	4.280×10^3	1.040×10^3	249.567
88^o N	3.123×10^4	6.509×10^3	1.217×10^3
89^o N	5.198×10^4	9.750×10^3	NA
89.5^o N	7.795×10^4	NA	NA

Table 5.3: ∞ norm condition numbers of system matrix A, $k = 0.01$

them closer to the boundary.

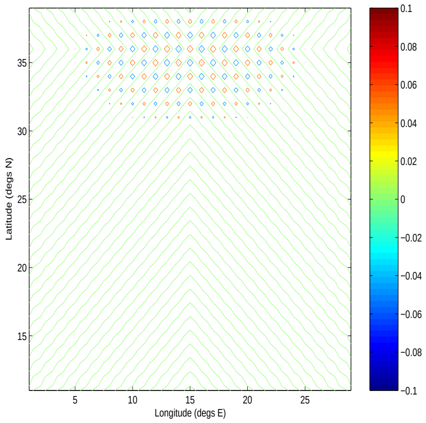

Figure 5.1: Eigenvector associated with largest eigenvalue of A for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$, $h = 1^o$, $k = 0.01$.

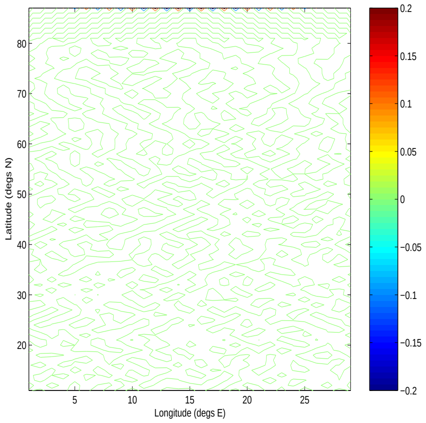

Figure 5.2: Eigenvector associated with largest eigenvalue of A for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$, $h = 1^o$, $k = 0.01$.

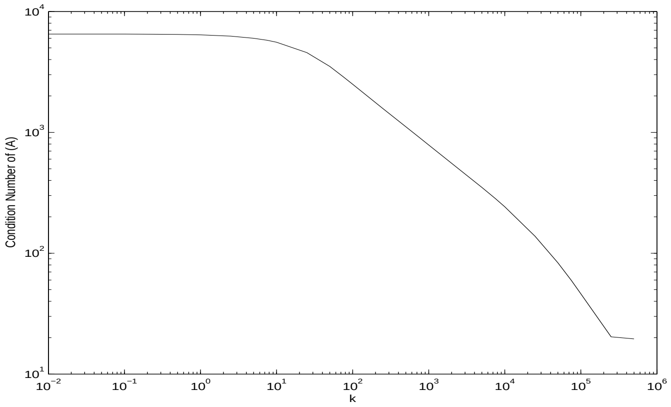

Figure 5.3: Variation of $\kappa_\infty(A)$ with k, Limited Area Helmholtz problem, $\phi_{NB} = 88^o$, $h = 1^o$.

Figure 5.3 demonstrates the effect the Helmholtz term, k, has on the conditioning of the problem. Figure 5.3 shows that increasing the magnitude of the Helmholtz term causes the conditioning to become better. As increasing the magnitude of the k term would increase the diagonal dominance of the system matrix A, this is what we would expect. Figure 5.3 shows that the change in conditioning is negligible until $k = 10.0$ when the conditioning improves dramatically until approximately $k = 2.5 \times 10^5$ when it levels off again. The full results may be found in Table B.2 in Appendix B. We use a value of $k = 0.01$ throughout our numerical experiments of this study. This retains the features of the free-surface problem ($k > 0$) but is also small enough to provide a more robust test of the preconditioners

5.2.2 Properties of preconditioned methods

Table 5.4 gives the ∞ norm condition numbers of the preconditioned system matrices, with the diagonal, block diagonal, ADI and Binormalization preconditioners, for the case where $h = 1^o$, $\phi_{NB} = 88^o$ and $k = 0.01$. We observe that, as predicted, the diagonal preconditioner significantly improves the conditioning of the system, with the block diagonal preconditioner bringing further improvement and the ADI preconditioner even more improvement. We note that the Gerschgorin bounds calculated in Section 4.5.6 and displayed in Table 4.1 are crude over-estimates of the values we have calculated here. However the qualitative behaviour of the relative convergence properties of our preconditioned methods predicted by the Gerschgorin analysis has been confirmed by the numerical calculations. The full conditioning results are shown in Appendix B in Tables B.8 - B.15, which show the ∞ and 2 norm condition numbers of the preconditioned system matrices for the diagonal, Block diagonal, ADI and Binormalization preconditioners. These results confirm that the conditioning values get larger with smaller stepsize, and as ϕ_{NB} is moved closer to the pole, as expected. Also the same pattern is observed, overall, in the size of the conditioning values between preconditioners as that shown in Table 5.4 (i.e. Largest to smallest : Binormalization, Diagonal,

Preconditioner	$\kappa_\infty(P^{-1}A)$
None	6.509×10^3
Diagonal	691.915
Block	293.920
ADI	139.311
Binormalization	694.737

Table 5.4: ∞ norm condition numbers for case where $h = 1^o$, $\phi_{NB} = 88^o$, $k = 0.01$.

Block Diagonal, ADI).

We observe from Tables 5.4 and 5.5 that the conditioning and spectral radii values for the diagonal preconditioner are slightly smaller than those of the binormalization preconditioner. Comparing the conditioning values in Table 5.4 of the two preconditioners we note that the condition $\kappa(P_D^{-1}A) \leq p\kappa(\hat{A})$ stated in Theorem 3.18 is satisfied (Here $p = 5$ due to there being at most five non-zero elements in each row of A). Also the assertion of Livne and Golub [52] that the diagonal preconditioner is the 'optimal' diagonal scaling, with regards to improving the conditioning of A when the system matrix, A, is 2-cyclic, is confirmed. Full results can be found in Appendix B in Tables B.8 - B.15 which further confirm these points.

Table 5.5 show the values for the spectral radii, $\rho(G)$, of the iteration matrices of the preconditioned methods for the case with $k = 0.01$ and $h = 1^o$. Note that they are all strictly less than one which confirms the convergence of the methods. The pattern of values with regards to the efficiency of the preconditioned methods is similar to that for the conditioning values shown in Table 5.4. The values for the ADI preconditioned iteration matrix, are the smallest, followed by the Block preconditioned matrix and then the diagonal and Binormalization preconditioned matrices. This suggest that the ADI

	ϕ_{NB}		
Preconditioner	40^o N	70^o N	88^o N
Diagonal	0.9946	0.9960	0.9960
Block diagonal	0.9879	0.9901	0.9901
ADI	0.8289	0.9273	0.9601
Binormalization	0.9973	0.9980	0.9980

Table 5.5: Spectral Radii of iteration matrix G for various preconditioners, $k = 0.01$, $h = 1^o$.

preconditioner (and to a lesser extent the Block preconditioner) ought to yield faster convergence.

An interesting feature of the results is that there is a negligible increase in the condition numbers, κ, and the spectral radii, $\rho(G)$, of the diagonal, block and Binormalization preconditioners, beyond a northern boundary of 70^o in all cases (see Appendix B). Therefore, not only do the largest eigenvalues of the preconditioned methods not vary beyond $\phi_{NB} = 70^o$, neither do the smallest since the conditioning is also unaffected. From this we deduce that the polar convergence issue arises from how the distribution of the spectrum of eigenvalues changes as ϕ_{NB} is increased. We have investigated this in our paper [12] which was presented at the 2004 ICFD Conference on Numerical Methods for Fluids. The results can be explained as followed. Figures 5.16 and 5.28, show that the eigenvectors associated with these leading eigenvalues do not have a strong signal in the polar region. The values are highest in the mid-latitudes which would be consistent with an isotropic domain. We would expect this, for example, to occur with the case where $\phi_{NB} = 40^o$N as that is closer to an isotropic case. However, with the more mesh anisotropic higher latitude boundary cases we would expect to see a strong signal in the polar region in the leading eigenmode. Figures 5.4 to 5.7, 5.8 to 5.15 and 5.16 to 5.23 provide an explanation. The eigenvalue plots of figures 5.4 to 5.7 show that, whilst the leading eigenvalue remains approximately constant, the 'next' leading eigenvalues (i.e. those associated with the second, third and fourth largest eigenvalues) cluster towards it, i.e. become larger, as the northern boundary is moved towards the pole. We would therefore expect their associated eigenvectors to become much more significant as the boundary is moved closer to the pole. Figures 5.17 to 5.19 and 5.21 to 5.23 show that the eigenvectors associated with these 'next' leading eigenvalues all possess a strong signal in the polar region close to the northern boundary. Overall we observe that as the boundary is moved closer to the pole, the 'nearly' leading eigenvalues become larger and cluster closer to the lead eigenvalue. This means that their associated eigenvectors become more significant, in terms of affecting the convergence behaviour as ϕ_{NB} increases. Also the eigenvectors themselves display much stronger signals in the polar regions as ϕ_{NB} is moved closer to the pole. More large eigenvalues with associated eigenvectors that have signals near the pole means that the convergence of the methods near the pole will be slower due to these eigenmodes.

The structure of the leading four eigenvectors of G_{Block}, with $\phi_{NB} = 88^o$, is shown in Figures 5.28 to 5.31. We note that the eigenvectors show much weaker signals in the polar area in addition to being associated with smaller eigenvalues. This leads us to conclude further that the block preconditioner will yield faster convergence and may also address the pole problem better than diagonal preconditioning. The leading four eigenvectors of G_{ADI}, for $\phi_{NB} = 88^o$, as shown in Figures 5.40 to 5.43 do have strong signals in the polar regions. However they are associated with considerably smaller eigenvalues than diagonal or block preconditioning. Whilst, from this, we might not expect the pole problem to be addressed very well, the convergence overall ought to be much faster with ADI than the other preconditioners. Also the largest magnitude eigenvalues of G_{ADI} are negative. Their associated eigenvectors have very high horizontal frequency scales, i.e. they are associated with very noisy information. Finally considering the leading eigenvectors for the binormalization scaling we note that the pattern is almost identical to that of the diagonal preconditioner. As this is essentially what binormalization scaling is this is perhaps to be expected.

For the diagonal preconditioner note that for every positive leading eigenvalue with an associated large-scale eigenvector (shown in Figures 5.8 to 5.11) there is a corresponding negative eigenvalue of identical magnitude with an associated small-scale eigenmode (shown in Figures 5.12 to 5.15). This is also the case with the block preconditioner. The eigenmodes shown in Figures 5.12 to 5.15 have strong signals in the same locations as the large positive eigenvalues, but have a very high frequency structure. If these high frequency modes exist in the approximation errors (this is likely, due to numerical round-off) then these will be slow to decay away.

The full distribution of eigenvalues of the four preconditioned matrices between -1 and 1, for 40^o and 88^o cases, is shown in Figures 5.52 to 5.59. Observe that for the Block and Diagonal preconditioners for each positive eigenvalue there is an equivalent negative eigenvalue of equal magnitude. This was predicted in Sections 4.4.2 and 4.4.3. This is not the case for the ADI preconditioner which has an unsymmetric structure or for the Binormalization iteration matrix which possesses only positive eigenvalues. The fact that the Binormalization preconditioned method does not have any negative eigenvalues is a very significant difference from the diagonal preconditioned method. This means that there are no large negative eigenvalues with associated eigenvectors with very small scale frequencies (and similar sizes of signals in the polar regions). This clustering of the eigenvalues could be of benefit in a PCG method.

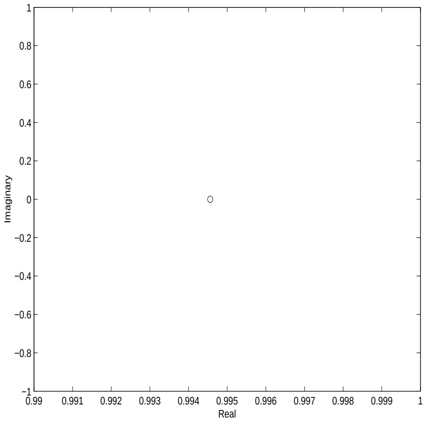

Figure 5.4: Eigenvalues of G_D between 0.99 and 1.0 for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$

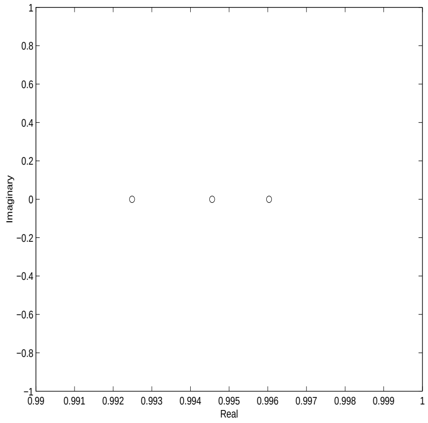

Figure 5.5: Eigenvalues of G_D between 0.99 and 1.0 for Limited Area Helmholtz problem. $\phi_{NB} = 70^o$

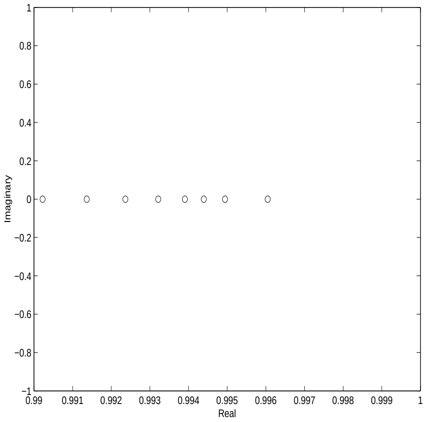

Figure 5.6: Eigenvalues of G_D between 0.99 and 1.0 for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$

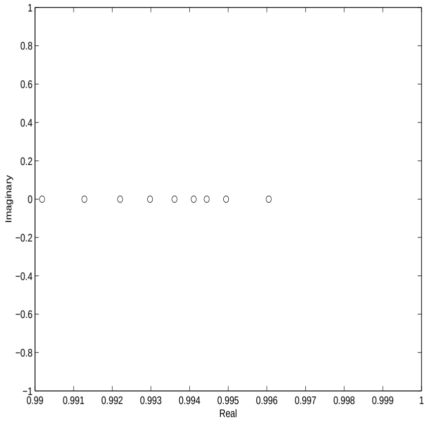

Figure 5.7: Eigenvalues of G_D between 0.99 and 1.0 for Limited Area Helmholtz problem. $\phi_{NB} = 89^o$

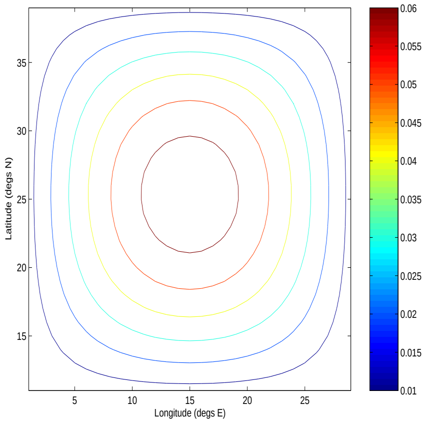

Figure 5.8: Eigenvector associated with largest eigenvalue (0.9946) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

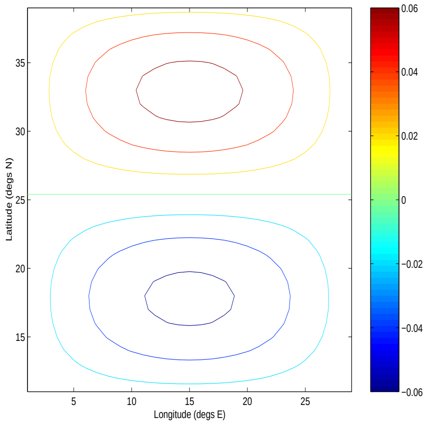

Figure 5.9: Eigenvector associated with second largest eigenvalue (0.9873) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

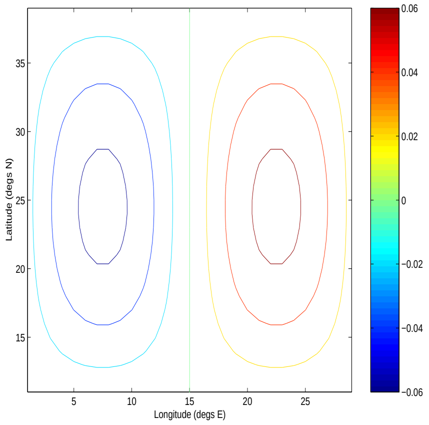

Figure 5.10: Eigenvector associated with third largest eigenvalue (0.9855) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

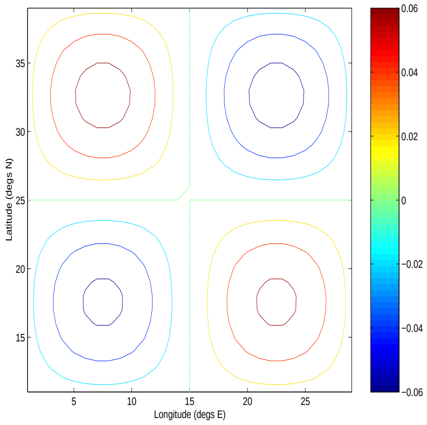

Figure 5.11: Eigenvector associated with fourth largest eigenvalue (0.9782) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

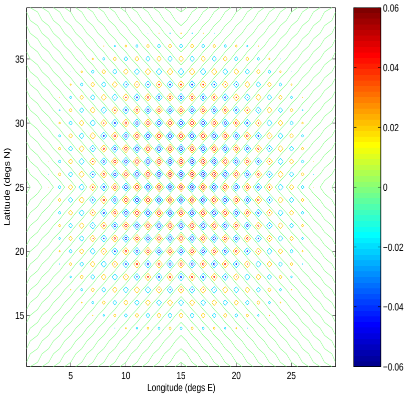

Figure 5.12: Eigenvector associated with largest negative eigenvalue (-0.9946) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

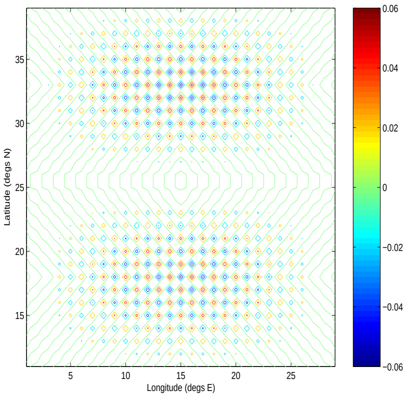

Figure 5.13: Eigenvector associated with second largest negative eigenvalue (-0.9873) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

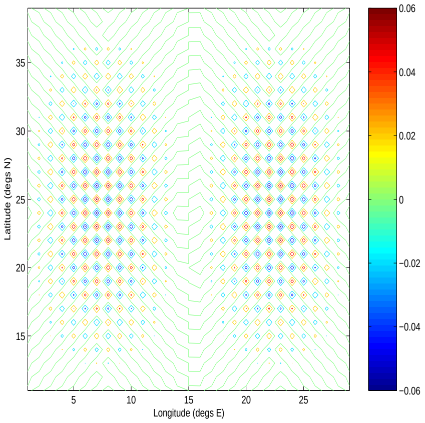

Figure 5.14: Eigenvector associated with third largest negative eigenvalue (-0.9855) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

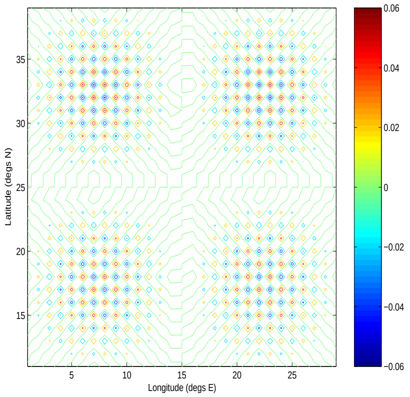

Figure 5.15: Eigenvector associated with fourth largest negative eigenvalue (-0.9782) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

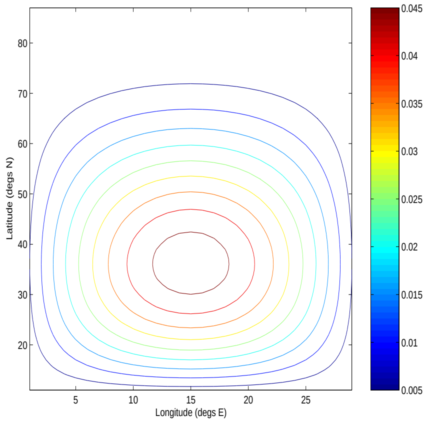

Figure 5.16: Eigenvector associated with largest eigenvalue (0.9960) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

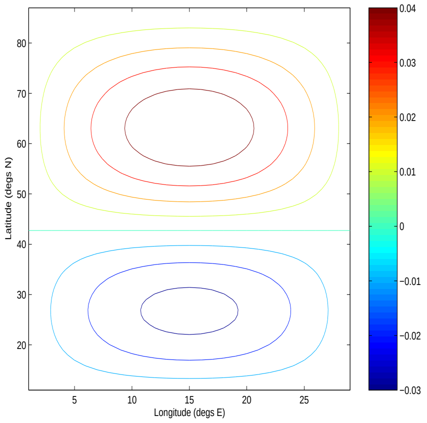

Figure 5.17: Eigenvector associated with second largest eigenvalue (0.9949) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

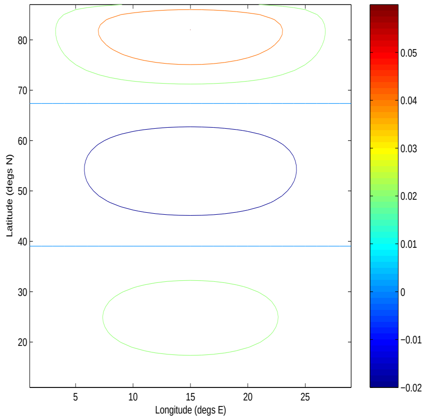

Figure 5.18: Eigenvector associated with third largest eigenvalue (0.9944) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

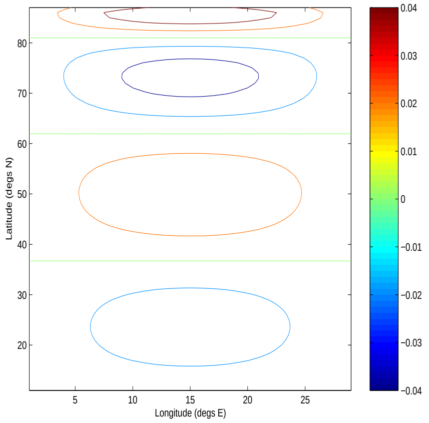

Figure 5.19: Eigenvector associated with fourth largest eigenvalue (0.9939) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

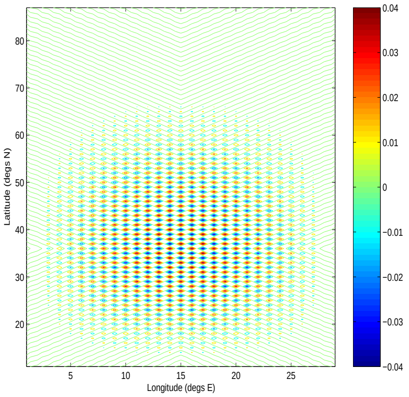

Figure 5.20: Eigenvector associated with largest negative eigenvalue (-0.9960) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

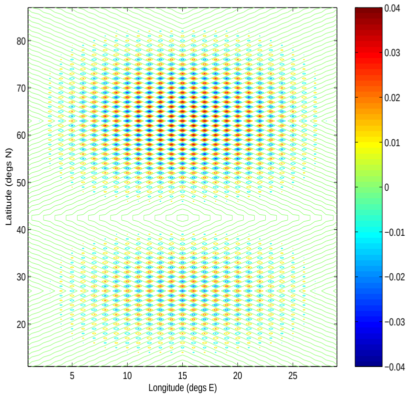

Figure 5.21: Eigenvector associated with second largest negative eigenvalue (-0.9949) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

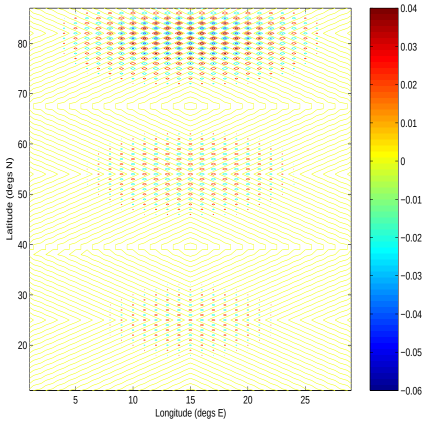

Figure 5.22: Eigenvector associated with third largest negative eigenvalue (-0.9944) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

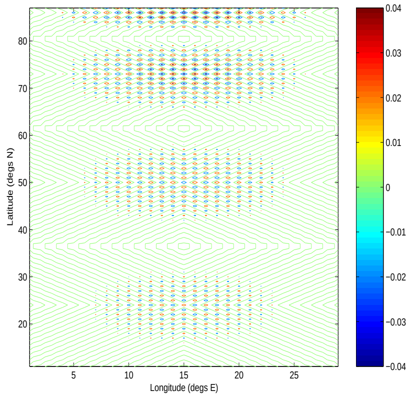

Figure 5.23: Eigenvector associated with fourth largest negative eigenvalue (-0.9939) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

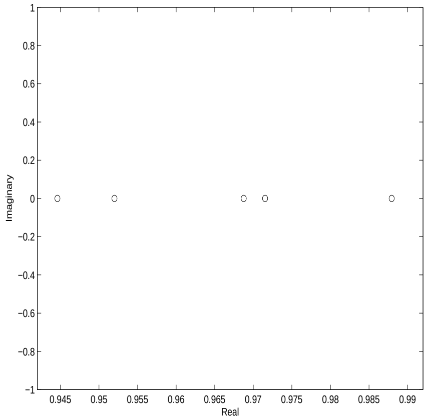

Figure 5.24: Eigenvalues of G_{Block} between 0.941 and 0.991 for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$

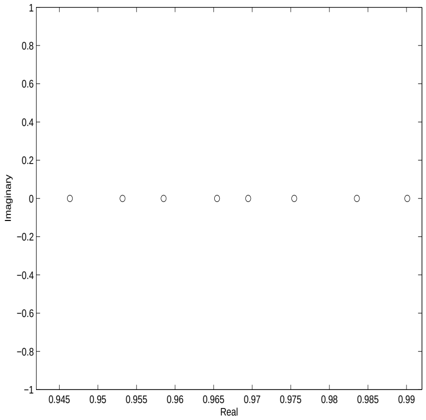

Figure 5.25: Eigenvalues of G_{Block} between 0.941 and 0.991 for Limited Area Helmholtz problem. $\phi_{NB} = 70^o$

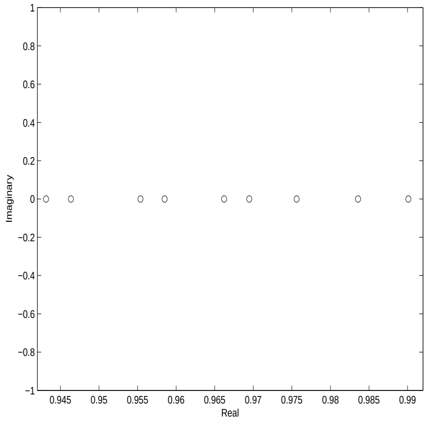

Figure 5.26: Eigenvalues of G_{Block} between 0.941 and 0.991 for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$

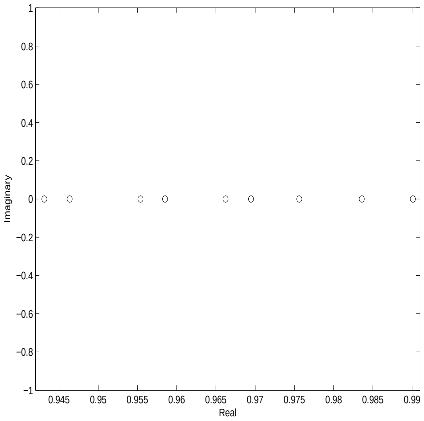

Figure 5.27: Eigenvalues of G_{Block} between 0.941 and 0.991 for Limited Area Helmholtz problem. $\phi_{NB} = 89^o$

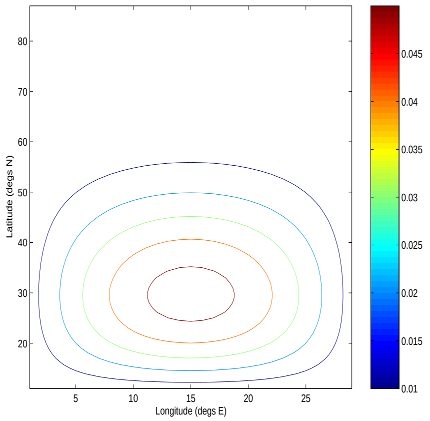

Figure 5.28: Eigenvector associated with largest eigenvalue (0.9901) of G_{Block} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

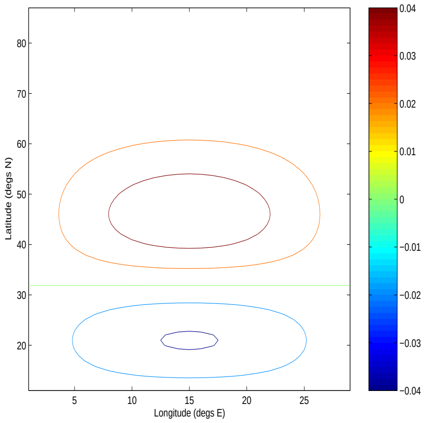

Figure 5.29: Eigenvector associated with second largest eigenvalue (0.9836) of G_{Block} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

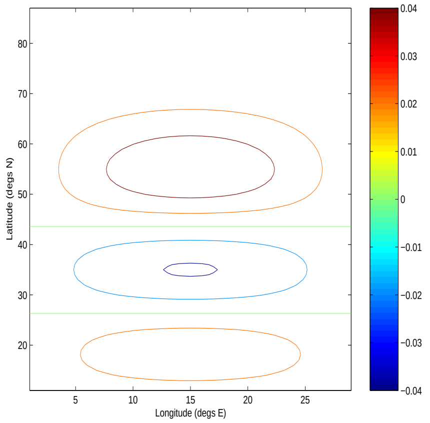

Figure 5.30: Eigenvector associated with third largest eigenvalue (0.9756) of G_{Block} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

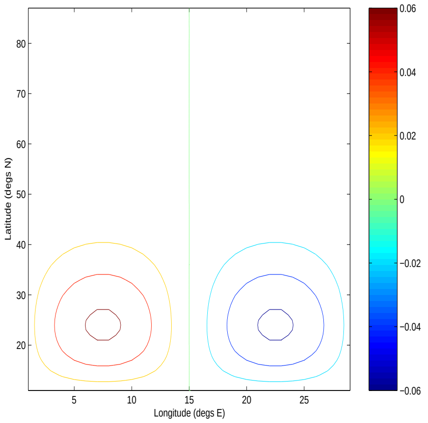

Figure 5.31: Eigenvector associated with fourth largest eigenvalue (0.9695) of G_{Block} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

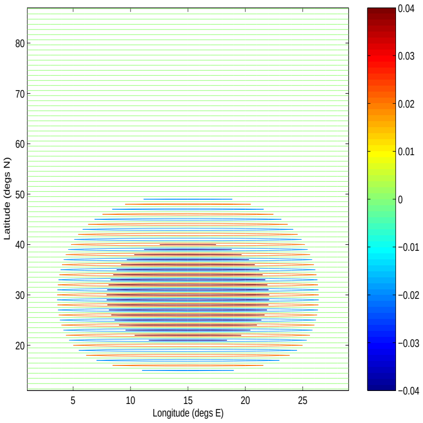

Figure 5.32: Eigenvector associated with largest negative eigenvalue (-0.9901) of G_{Block} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

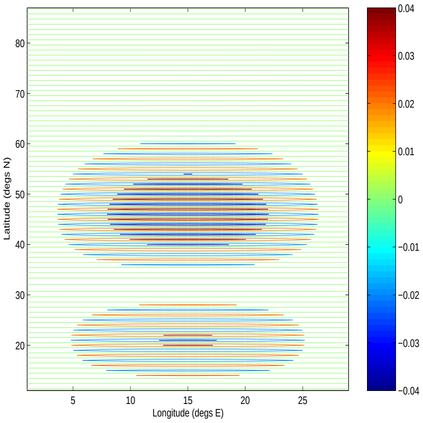

Figure 5.33: Eigenvector associated with second largest negative eigenvalue (-0.9836) of G_{Block} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

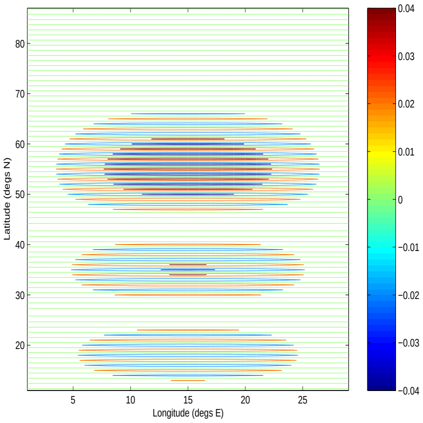

Figure 5.34: Eigenvector associated with third largest negative eigenvalue (-0.9756) of G_{Block} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

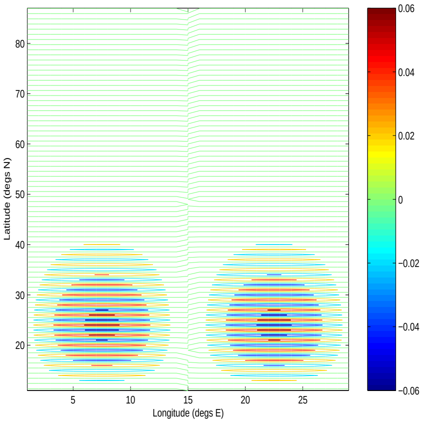

Figure 5.35: Eigenvector associated with fourth largest negative eigenvalue (-0.9695) of G_{Block} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

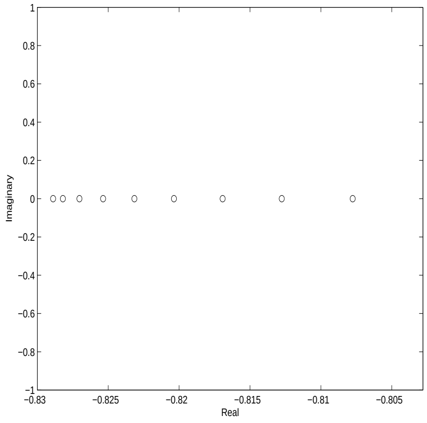

Figure 5.36: Eigenvalues of G_{ADI} between -0.8027 and -0.83 for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$

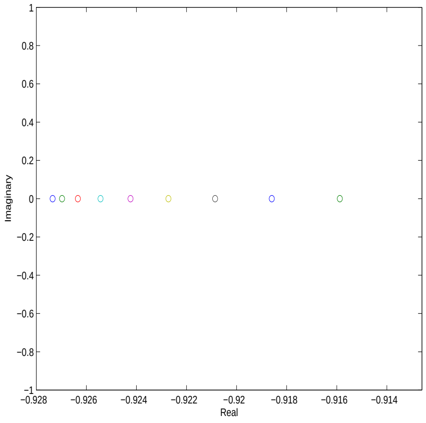

Figure 5.37: Eigenvalues of G_{ADI} between -0.9126 and -0.928 for Limited Area Helmholtz problem. $\phi_{NB} = 70^o$

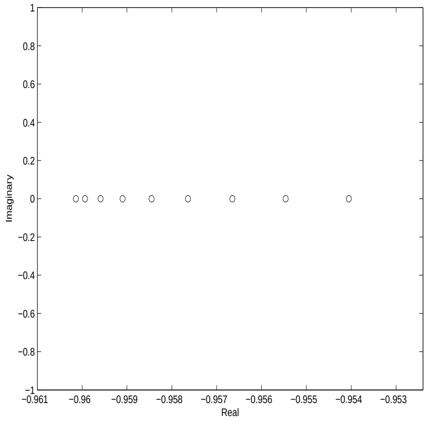

Figure 5.38: Eigenvalues of G_{ADI} between -0.9525 and -0.961 for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$

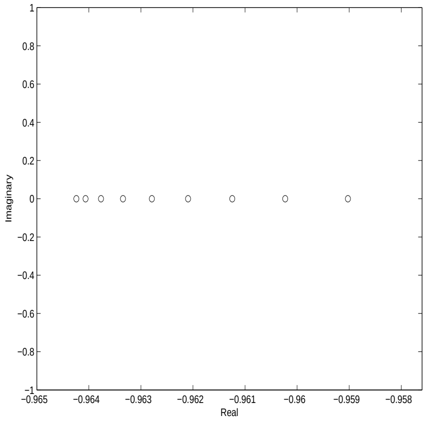

Figure 5.39: Eigenvalues of G_{ADI} between -0.9575 and -0.965 for Limited Area Helmholtz problem. $\phi_{NB} = 89^o$

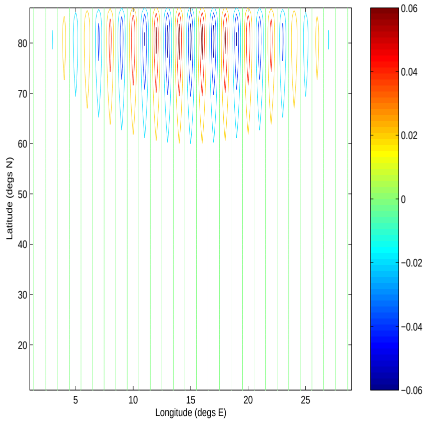

Figure 5.40: Eigenvector associated with largest eigenvalue (-0.9601) of G_{ADI} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

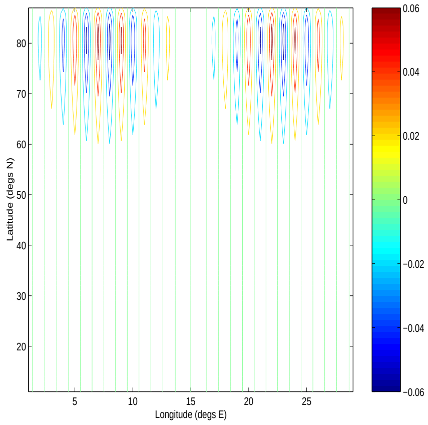

Figure 5.41: Eigenvector associated with second largest eigenvalue (-0.9599) of G_{ADI} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

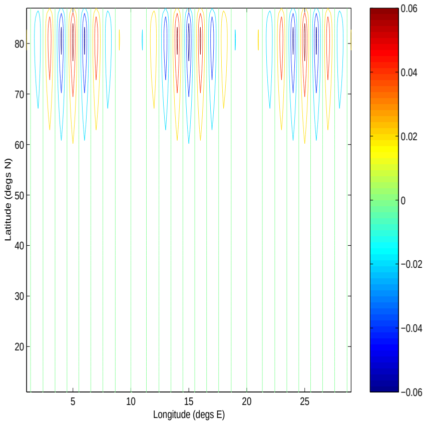

Figure 5.42: Eigenvector associated with third largest eigenvalue (-0.9596) of G_{ADI} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

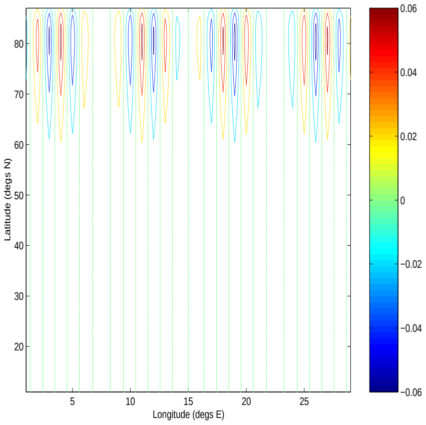

Figure 5.43: Eigenvector associated with fourth largest eigenvalue (-0.9591) of G_{ADI} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

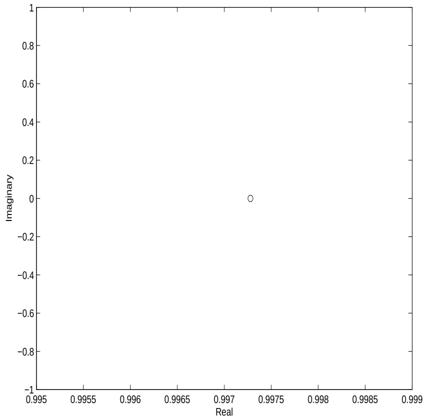

Figure 5.44: Eigenvalues of G_{BIN} between 0.995 and 0.999 for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$

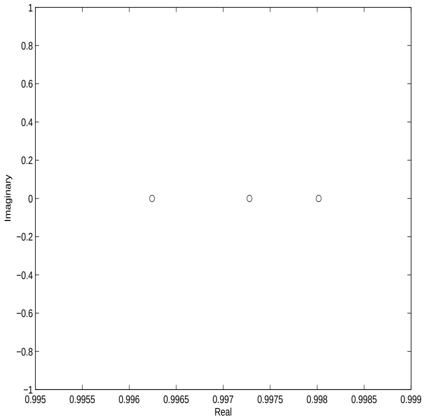

Figure 5.45: Eigenvalues of G_{BIN} between 0.995 and 0.999 for Limited Area Helmholtz problem. $\phi_{NB} = 70^o$

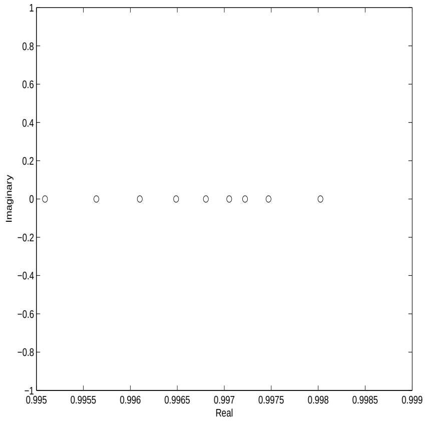

Figure 5.46: Eigenvalues of G_{BIN} between 0.995 and 0.999 for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$

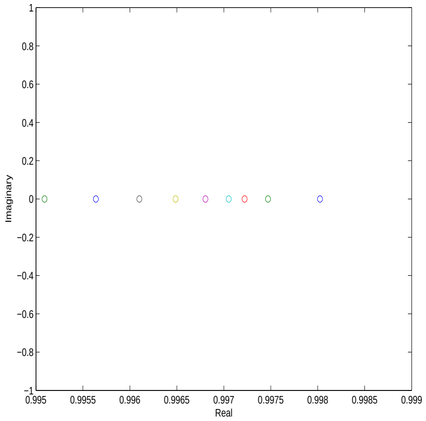

Figure 5.47: Eigenvalues of G_{BIN} between 0.995 and 0.999 for Limited Area Helmholtz problem. $\phi_{NB} = 89^o$

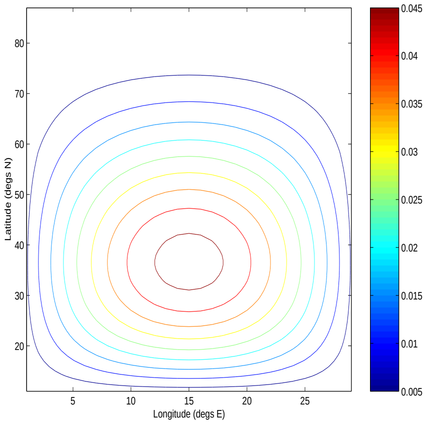

Figure 5.48: Eigenvector associated with largest eigenvalue (0.9980) of G_{BIN} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

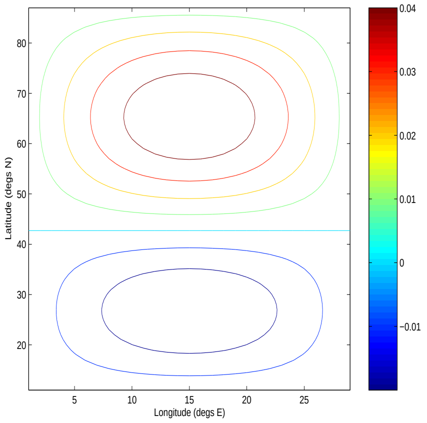

Figure 5.49: Eigenvector associated with second largest eigenvalue (0.9975) of G_{BIN} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

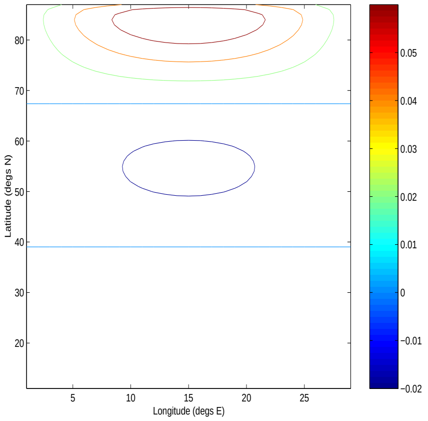

Figure 5.50: Eigenvector associated with third largest eigenvalue (0.9972) of G_{BIN} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

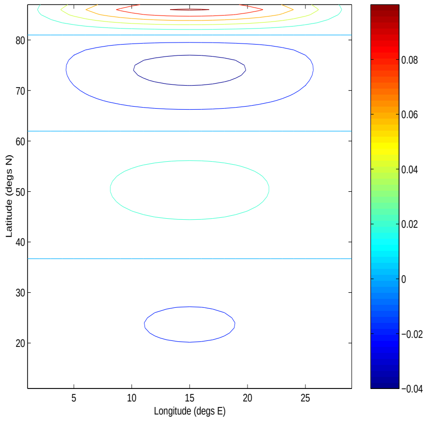

Figure 5.51: Eigenvector associated with fourth largest eigenvalue (0.9970) of G_{BIN} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

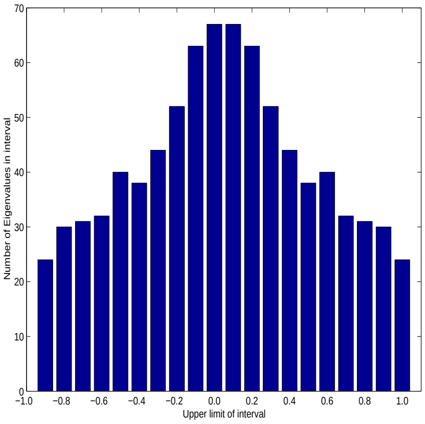

Figure 5.52: Histogram showing distribution of eigenvalues of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

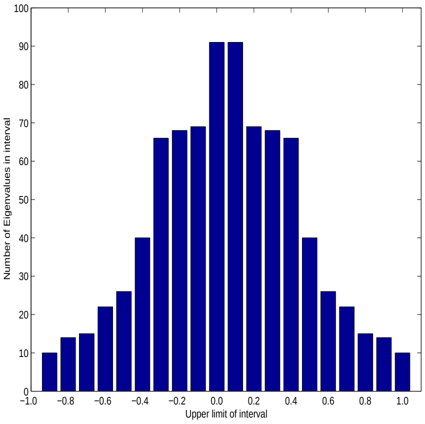

Figure 5.53: Histogram showing distribution of eigenvalues of G_{Block} for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

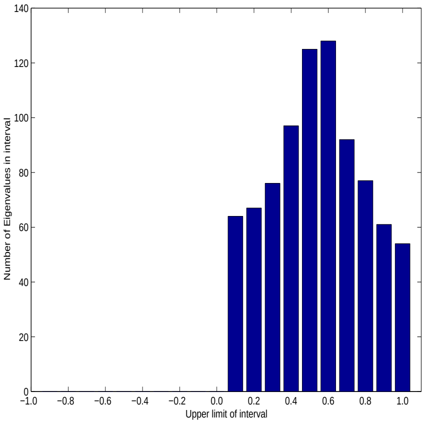

Figure 5.54: Histogram showing distribution of eigenvalues of G_{BIN} for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

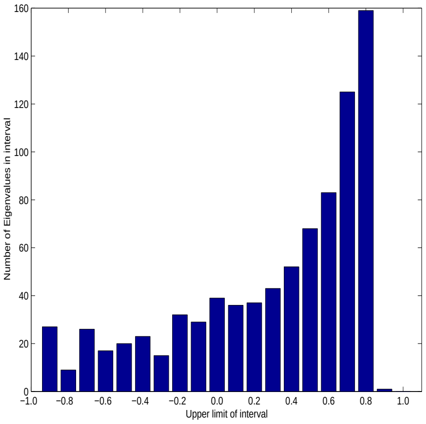

Figure 5.55: Histogram showing distribution of eigenvalues of G_{ADI} for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

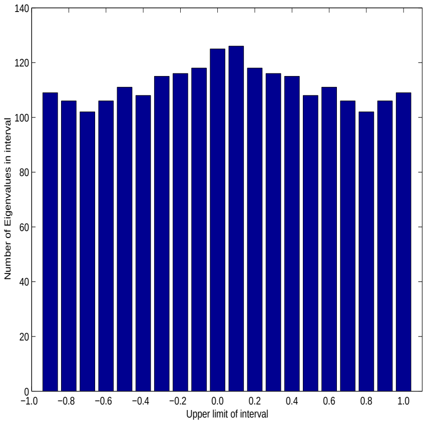

Figure 5.56: Histogram showing distribution of eigenvalues of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

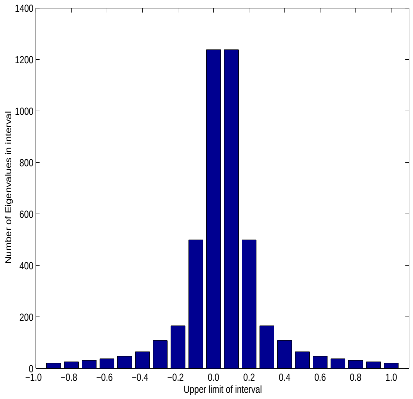

Figure 5.57: Histogram showing distribution of eigenvalues of G_{Block} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

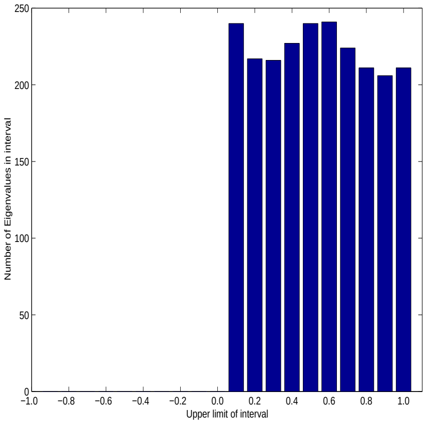

Figure 5.58: Histogram showing distribution of eigenvalues of G_{BIN} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

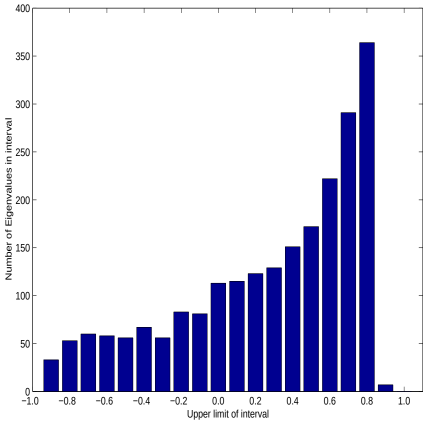

Figure 5.59: Histogram showing distribution of eigenvalues of G_{ADI} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

5.2.3 Practical convergence experiments

We ran numerical experiments to check the previous findings and assess our chosen preconditioners. The right hand source function $\gamma(\lambda, \phi)$ was fixed to yield a sine function general solution for the Helmholtz problem of

$$\begin{aligned} &U(\lambda,\phi) = sin(3\lambda)sin(d[\phi - 10]) \\ &d = \tfrac{90}{\phi_{NB}-10} \end{aligned} \tag{5.2}$$

which is consistent with the chosen boundary conditions given in (5.1). A constant 'initial guess' of $U(i,j) = 1.5$ was taken to start the iterative process. The relative residual ∞ norm error normalised by the source vector, (b), is used in the stopping criterion which is given by

$$\frac{\|\, \mathbf{r}^m \,\|_\infty}{\|\, \mathbf{b} \,\|_\infty} = \frac{\|\, \mathbf{b} - A\mathbf{U^m} \,\|_\infty}{\|\, \mathbf{b} \,\|_\infty} < 10^{-5}. \tag{5.3}$$

This criterion is used in all experiments in this study unless otherwise stated.

The results of the full numerical experiments can be seen in Tables 5.6 and 5.7 and in Figure 5.60. As expected the number of iterations required to achieve the convergence tolerance varied with the conditioning of the system. More iterations were required as the northern boundary was moved towards the pole and as smaller stepsizes were taken. The ADI preconditioner took the fewest iterations in all cases followed by the block diagonal, Binormalization and diagonal preconditioners respectively (The parameter values used for the ADI preconditioner in our experiments may be found in Appendix B, in Table B.1). This pattern is continued to a lesser degree in the CPU time results shown in Table 5.7. The CPU times only include the computational time used to perform the iteration sweeps. Any 'off-line' calculations are not included in the timings. Figure 5.60 starkly illustrates the benefits of ADI; comparing the convergence, latitudinally, of the diagonal and block diagonal preconditioned methods at the time ADI has converged shows all three preconditioners to be approximately equal in equatorial and mid-latitude regions. However, ADI is better than block in polar regions which in turn has converged more than the Binormalization scaled and diagonal preconditioned CG methods.

The ADI preconditioner with spatially varying parameter did not perform very well in the experiments. An example of the performance of the preconditioner is shown in Table 5.8 for the 1^o stepsize, $\phi_{NB} = 88^o$ case. The parameters for the preconditioner were calculated using the Gerschgorin estimates from (4.69) in Section 4.5.5. It can be observed that the results for the spatially varying parameter ADI preconditioner compare unfavourably with the corresponding results for the other preconditioners. It is likely that the poor performance was caused by the sensitivity of the preconditioner to the parameter values combined with the fact that the values were calculated using the crude Gerschgorin estimates. In addition it was confirmed numerically that the matrices H_Υ and V_Υ for the stationary ADI preconditioned method do not commute, in any of the cases considered here.

Whilst in CPU terms the Binormalization scaling performed slightly worse than the diagonal preconditioner, in some cases it took fewer iterations to converge. This is curious as the condition numbers and spectral radii appear to favour the diagonal preconditioner. However the distribution of the eigenvalues is more favourable with the Binormalization scaling as its preconditioned iteration matrix has strictly positive eigenvalues (recall that the iteration matrix of the diagonal preconditioned matrix has a $\pm$ eigenstructure). This form of 'clustering' is the likely explanation for its good performance relative to the diagonal preconditioner.

As a sub-experiment we investigated the propagation of errors about the domain with respect to the various preconditioners employed. Using the same general solution (5.2) and stopping criteria as the previous experiments we set the initial guess vector to be equal to that solution in all rows of the mesh except one. The results of taking the first (near equatorial) and last (near polar) rows of the grid to be the 'error' row for the 88^o case are shown in Figures 5.61 to 5.68. These figures show the variation of the relative residual

	40^o			70^0			88^o			89^o		89.5^o
Prec.	$\frac{1}{2}^o$	1^o	2^o	$\frac{1}{2}^o$	1^o	2^o	$\frac{1}{2}^o$	1^o	2^o	$\frac{1}{2}^o$	1^o	$\frac{1}{2}^o$
Diag	169	80	37	262	129	62	353	167	77	358	171	371
Blo	115	55	27	146	73	36	152	75	37	154	76	154
ADI	36	23	15	45	29	19	47	30	20	49	33	55
Bin	171	81	38	262	130	63	330	158	77	332	160	332

Table 5.6: Number of iterations to convergence tolerance $\frac{||\mathbf{r}^m||_\infty}{||\mathbf{b}||_\infty} < 10^{-5}$, sine function general solution, $k = 0.01$

	40^o			70^0			88^o			89^o		89.5^o
Prec.	$\frac{1}{2}^o$	1^o	2^o	$\frac{1}{2}^o$	1^o	2^o	$\frac{1}{2}^o$	1^o	2^o	$\frac{1}{2}^o$	1^o	$\frac{1}{2}^o$
Diag	5.2	0.8	0.3	24.6	2.3	0.4	48.7	4.0	1.2	51.1	4.8	51.8
Blo	7.0	1.0	0.4	27.4	2.5	0.5	41.9	3.8	1.1	44.1	4.3	44.6
ADI	4.9	1.0	0.4	17.7	2.2	0.5	27.3	3.6	0.9	32.5	3.8	34.7
Bin	5.9	0.8	0.3	24.7	2.3	0.4	45.7	3.9	1.2	48.3	4.5	48.6

Table 5.7: CPU times for methods to reach convergence tolerance $\frac{||\mathbf{r}^m||_\infty}{||\mathbf{b}||_\infty} < 10^{-5}$, sine function general solution, $k = 0.01$

error latitudinally at convergence (i.e. when the convergence criteria has just been met), for the diagonal, block, ADI and Binormalization preconditioned methods. With the Block and ADI preconditioners the errors are marginally more evenly spread but with all four preconditioners there is a tendency for the errors to propagate to, or remain in, the polar region.

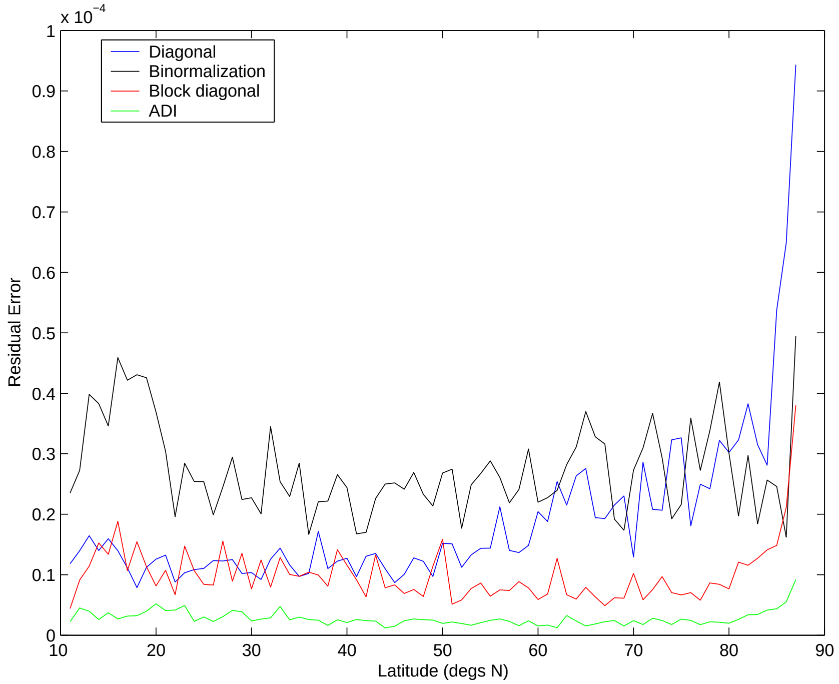

Figure 5.60: Latitudinal variance in relative residual errors, after fixed CPU time (ADI converged), for Limited Area Helmholtz problem with Diagonal, Block and ADI preconditioners. $\phi_{NB} = 88^o$, $h = 1^o$, $k = 0.01$.

	Stepsize		
	$\frac{1}{2}^o$	1^o	2^o
Its	351	127	54
CPU	228.2	16.4	2.9

Table 5.8: Iterations to convergence tolerance and associated CPU times, ADI preconditioner with varying parameter, $\phi_{NB} = 88^o$, $h = 1^o$, $k = 0.01$.

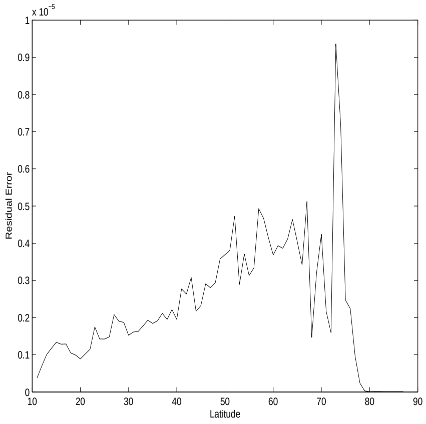

Figure 5.61: Latitudinal variation in residual errors at convergence. Error mode in first row of grid. Diagonal Preconditioner

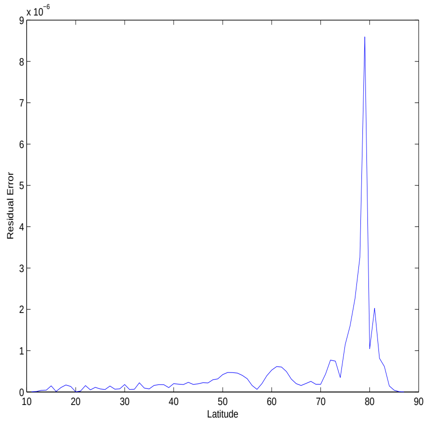

Figure 5.62: Latitudinal variation in residual errors at convergence. Error mode in last row of grid. Diagonal Preconditioner

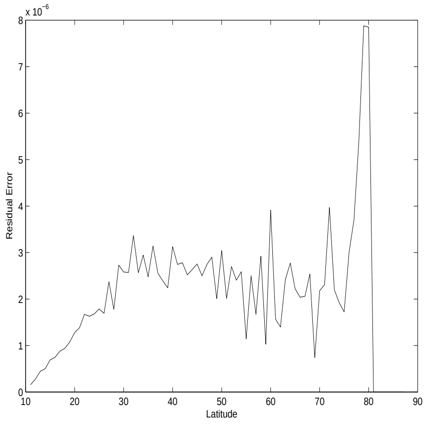

Figure 5.63: Latitudinal variation in residual errors at convergence. Error mode in first row of grid. Block Preconditioner

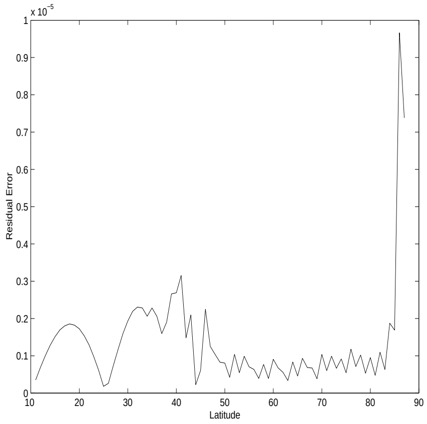

Figure 5.64: Latitudinal variation in residual errors at convergence. Error mode in last row of grid. Block Preconditioner

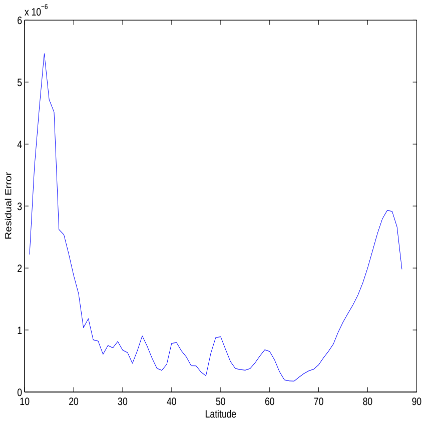

Figure 5.65: Latitudinal variation in residual errors at convergence. Error mode in first row of grid. ADI Preconditioner

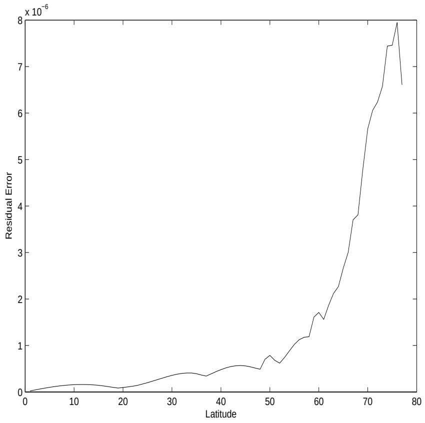

Figure 5.66: Latitudinal variation in residual errors at convergence. Error mode in last row of grid. ADI Preconditioner

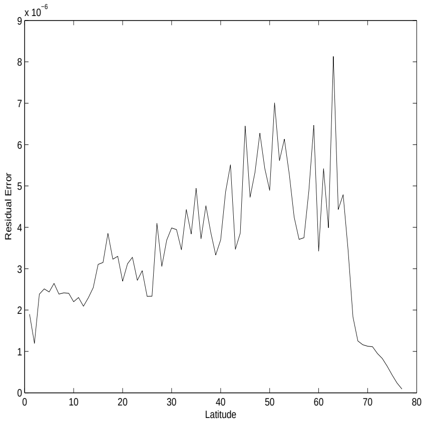

Figure 5.67: Latitudinal variation in residual errors at convergence. Error mode in first row of grid. Binormalization Preconditioner

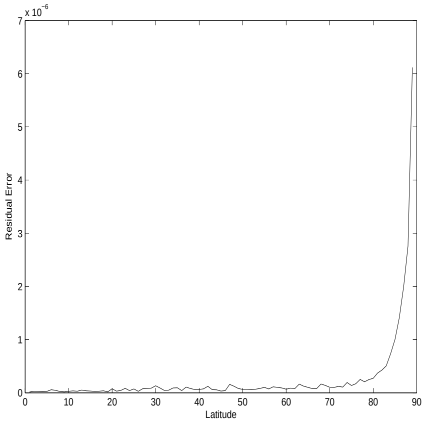

Figure 5.68: Latitudinal variation in residual errors at convergence. Error mode in last row of grid. Binormalization Preconditioner

5.3 Periodic domain experiments

5.3.1 Problem formulation

We now move on to consider the periodic domain case for the constant depth spherical Helmholtz problem. Recall that this is given by

$$\begin{cases} -\frac{1}{cos\phi}\left[\frac{\partial}{\partial\lambda}\left(\frac{1}{cos\phi}\frac{\partial U}{\partial\lambda}\right)+\frac{\partial}{\partial\phi}\left(cos\phi\frac{\partial U}{\partial\phi}\right)\right]+kU=\gamma(\lambda,\phi) \\ \lambda\in(0^oW,0^oE) \quad \phi\in(10^oN,\phi_{NB}) \\ U(0^oW,\phi)=U(0^oE,\phi) \\ \frac{\partial U(0^oW,\phi)}{\partial\lambda}=\frac{\partial U(0^oE,\phi)}{\partial\lambda} \\ U(\lambda,10^oN)=0, U(\lambda,\phi_{NB})=0 \\ \phi_{NB}\in(40^oN,89.5^oN). \end{cases} \tag{5.4}$$

As before the main aims of the experiments of this section are to assess the effects on the properties of the iterative methods of extending the northern boundary, and to investigate the efficiency of the proposed preconditioners. In addition we consider the numerical effects of using periodic boundary conditions in the longitudinal (λ) direction. We retain Dirichlet boundary conditions in the ϕ direction. Discrete stepsizes of 2^0 and 1^0 are used.

5.3.2 Properties of preconditioned methods

Table 5.9 shows the ∞ norm condition numbers for the preconditioned system matrices for the $h = 1^o$, $\phi = 88^o$, $k = 0.01$ case. The ADI preconditioner again improves the conditioning of the system the most, followed by the Block diagonal preconditioner, and then the diagonal preconditioner. This trend is confirmed by the full results shown in Appendix B in Tables B.17 to B.20. The condition numbers again increase with smaller stepsizes and as ϕ_{NB} is moved closer to the pole.

Preconditioner	$\kappa_\infty(P^{-1}A)$
None	1.21×10^5
Diagonal	2.23×10^4
Block	3.25×10^3
ADI	972.656

Table 5.9: ∞ norm condition numbers for case where $h = 1^o$, $\phi_{NB} = 88^o$, $k = 0.01$

	Stepsize	
ϕ_{NB}	1^o	2^o
40^o N	2.66×10^4	6.59×10^3
70^o N	3.98×10^4	9.53×10^3
88^o N	2.51×10^5	4.72×10^4
89^o N	3.76×10^5	NA

Table 5.10: Spectral Radii of system matrix A, $k = 0.01$

	ϕ_{NB}		
Preconditioner	40^oN	70^oN	88^oN
Diagonal	0.9976	0.9996	0.9999
Block diagonal	0.9946	0.9987	0.9994
ADI	0.9052	0.9595	0.9739

Table 5.11: Spectral Radius of iteration matrix G for various preconditioners, $k = 0.01$, $h = 1^o$

The spectral radii of the iteration matrices are shown in Table 5.11. The spectral radii of the G matrices are all less than 1 guaranteeing the convergence of the numerical method, with the values for G_D the largest followed by G_{Block} and then G_{ADI}, further suggesting that the ADI preconditioner should yield the fastest convergence rates. The full results may be found in Appendix B in Tables B.21 to B.23. Again we note the increasing values of these with decreasing stepsizes, and as ϕ_{NB} is moved closer to the pole.

A similar pattern to the Limited Area case is also noted in the form of the leading eigenvectors of the iteration matrices, G. Strong polar signals are observed in the leading eigenvectors of the diagonal preconditioner in Figures 5.69 to 5.72. Significantly smaller polar signals are observed in the leading eigenvectors of the Block preconditioner as shown in Figures 5.73 to 5.76, and in the ADI preconditioner as shown in Figures 5.77 to 5.80. In conjunction with the associated eigenvalues, this leads us to again expect the block preconditioner, and particularly the ADI preconditioner to provide faster convergence and to address the pole problem more effectively than the diagonal preconditioner.

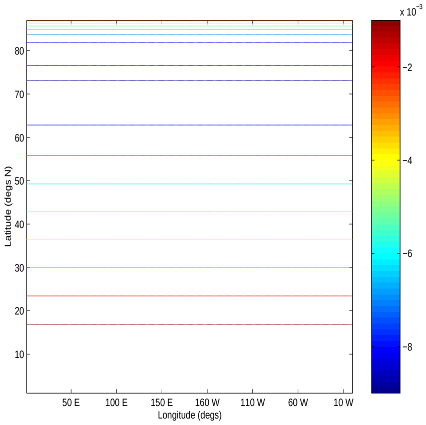

Figure 5.69: Eigenvector associated with largest eigenvalue (0.9999) of G_D for Periodic domain Helmholtz problem. $\phi_{NB} = 88^o$.

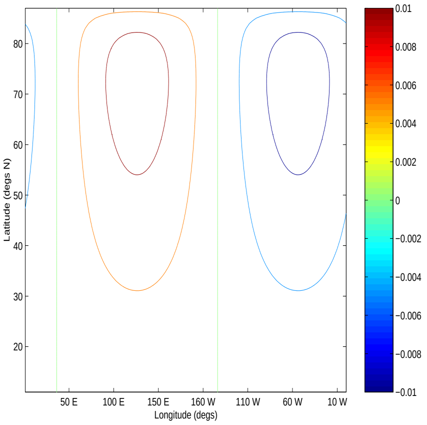

Figure 5.70: One Eigenvector associated with joint second largest eigenvalue (0.9998) of G_D for Periodic domain Helmholtz problem. $\phi_{NB} = 88^o$.

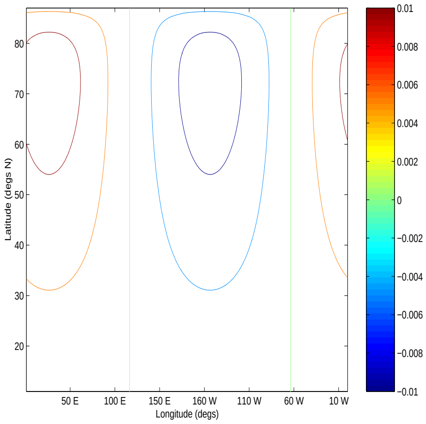

Figure 5.71: Other Eigenvector associated with joint second largest eigenvalue (0.9998) of G_D for Periodic domain Helmholtz problem. $\phi_{NB} = 88^o$.

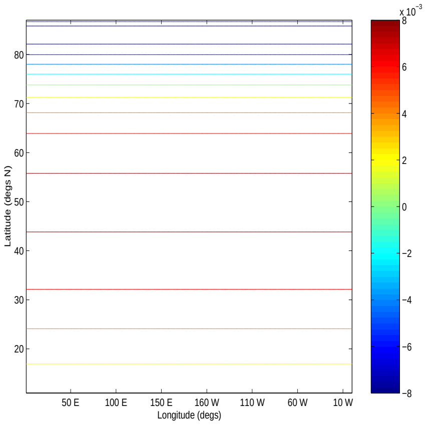

Figure 5.72: Eigenvector associated with third largest eigenvalue (0.9997) of G_D for Periodic domain Helmholtz problem. $\phi_{NB} = 88^o$.

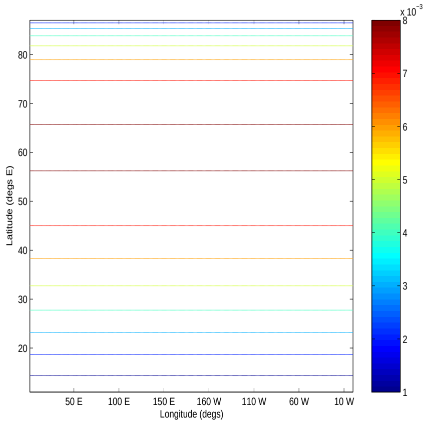

Figure 5.73: Eigenvector associated with largest eigenvalue (0.9994) of G_{Block} for Periodic domain Helmholtz problem. $\phi_{NB} = 88^o$.

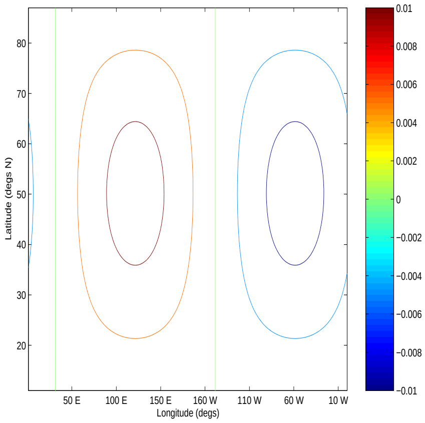

Figure 5.74: One Eigenvector associated with joint second largest eigenvalue (0.9988) of G_{Block} for Periodic domain Helmholtz problem. $\phi_{NB} = 88^o$.

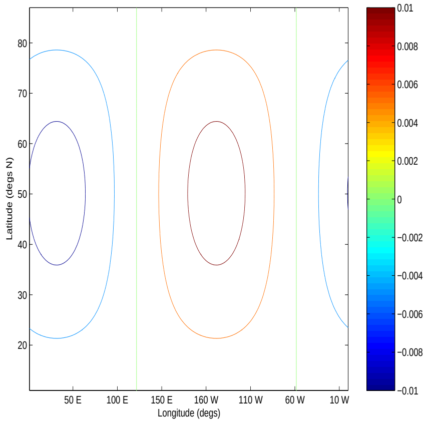

Figure 5.75: Other Eigenvector associated with joint second largest eigenvalue (0.9988) of G_{Block} for Periodic domain Helmholtz problem. $\phi_{NB} = 88^o$.

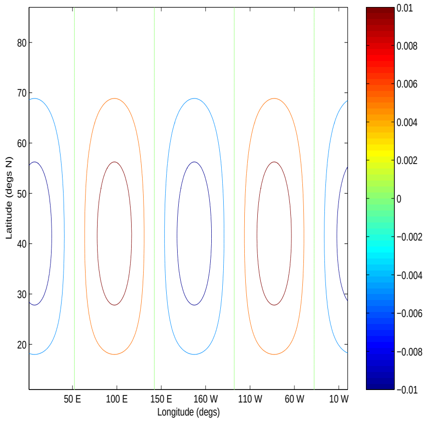

Figure 5.76: Eigenvector associated with third largest eigenvalue (0.9978) of G_{Block} for Periodic domain Helmholtz problem. $\phi_{NB} = 88^o$.

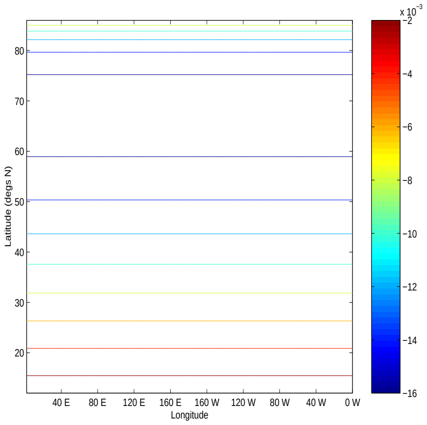

Figure 5.77: Eigenvector associated with largest eigenvalue (0.9739) of G_{ADI} for Periodic domain Helmholtz problem. $\phi_{NB} = 88^o$.

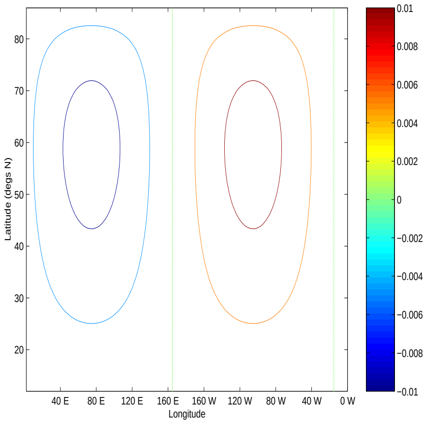

Figure 5.78: One Eigenvector associated with joint second largest eigenvalue (0.9738) of G_{ADI} for Periodic domain Helmholtz problem. $\phi_{NB} = 88^o$.

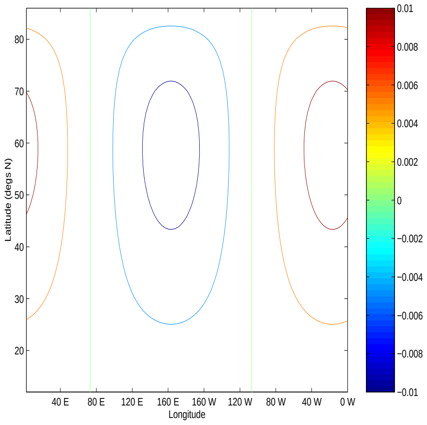

Figure 5.79: Other Eigenvector associated with joint second largest eigenvalue (0.9738) of G_{ADI} for Periodic domain Helmholtz problem. $\phi_{NB} = 88^o$.

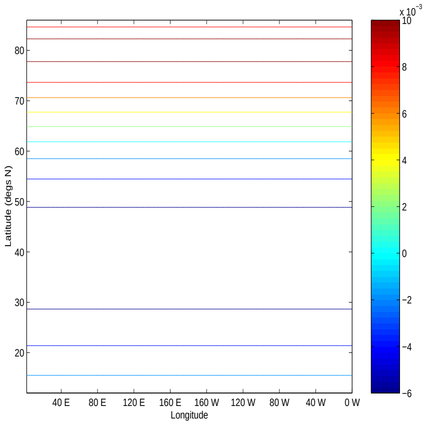

Figure 5.80: Eigenvector associated with third largest eigenvalue (0.9737) of G_{ADI} for Periodic domain Helmholtz problem. $\phi_{NB} = 88^o$.

	40^o		70^0		88^o		89^o
Prec.	1^o	2^o	1^o	2^o	1^o	2^o	1^o
Diag	156	75	323	159	564	275	632
Blo	94	47	177	88	220	107	222
ADI	39	26	70	45	87	55	88

Table 5.12: Number of iterations to convergence tolerance $\frac{||\mathbf{r}^m||_\infty}{||\mathbf{b}||_\infty} < 10^{-5}$, sine function general solution, $k = 0.01$

	40^o		70^0		88^o		89^o
Prec.	1^o	2^o	1^o	2^o	1^o	2^o	1^o
Diag	18.1	1.3	109.7	7.7	510.4	20.8	524.3
Blo	21.9	1.5	118.4	8.5	416.5	17.0	431.5
ADI	18.5	1.6	99.3	9.1	346.4	16.4	359.2

Table 5.13: CPU times, sine function general solution, $k = 0.01$

5.3.3 Practical convergence experiments

The assertion that the ADI preconditioned method (and to a lesser extent the block preconditioned method) should provide faster convergence is confirmed by the convergence results shown in Tables 5.12 to 5.13. This time we used a general solution of the form

$$\begin{aligned} &U(\lambda, \phi) = sin(\lambda)sin(d[\phi - 10]) \\ &d = \tfrac{90}{\phi_{NB}-10} \end{aligned} \tag{5.5}$$

which is consistent with the boundary conditions given in (5.4). The relative residual error normalised by $\mathbf{b}$ is used as the stopping criterion as in the Limited Area experiments of Section 5.2.3.

5.4 Unforced problem : Fourier modes as initial errors

5.4.1 Problem formulation

This final section of numerical experiments returns to a Limited Area problem for the constant depth spherical model defined in Section 4.8.1. Here we allow the domain to extend fully in the latitudinal direction to include the north pole. We also extend the domain to 90^o E in the longitudinal direction. Dirichlet boundary conditions of $U = 0$ are taken everywhere on the boundary of the domain. We also use a Poisson type equation with the 'Helmholtz' term $k = 0$ and a zero forcing function on the right hand side. The problem we consider is as follows :

$$\begin{cases} \frac{1}{cos\phi}\left[\frac{\partial}{\partial\lambda}\left(\frac{1}{cos\phi}\frac{\partial U}{\partial\lambda}\right) + \frac{\partial}{\partial\phi}\left(cos\phi\frac{\partial U}{\partial\phi}\right)\right] = 0 \\ \lambda \in (0^oE, 90^oE) \quad \phi \in (0^oN, 90^oN) \\ U(0^oE,\phi) = 0, U(90^oE,\phi) = 0 \\ U(\lambda, 0^oN) = 0, U(\lambda, 90^oN) = 0. \end{cases} \tag{5.6}$$

The general solution for this problem is $U(\lambda, \phi) = 0$.

5.4.2 Properties of preconditioned methods

Table 5.14 gives the conditioning and spectral radii information used to assess the convergence of the various preconditioned methods. As before we note the increasing values (apart from the ADI parameter) with the decreased stepsize. Again the spectral radii of the iteration matrices are all less than 1 guaranteeing convergence of the numerical method,with the values for G_D the largest, followed by G_{Block} and then G_{ADI} suggesting that the ADI preconditioner should again yield the fastest convergence rates. The same pattern and conclusion can be shown by the conditioning results. The effect of considering the worst anisotropy case (with no 'helpful' Helmholtz term) is further shown by Figure 5.81. The eigenvalues of G_D in Figure 5.81 are clustered near 1, the convergence limit, suggesting that a large number of eigenmodes with large eigenvalue could contribute to the error. The form of the leading eigenmodes of G_D show a similar pattern to that observed in the Limited area problem, as do the leading eigenvectors for the block preconditioner, and the leading eigenvectors for the ADI preconditioner. These are displayed in Appendix B. Again the leading eigenvectors of the block preconditioner have a much weaker polar signal than the eigenvectors of the diagonal preconditioner. Also, as before, the leading eigenvectors of the ADI preconditioner do have a significant polar signal. However they are associated with considerably smaller eigenvalues than diagonal or block preconditioning. We would therefore expect the overall convergence to be faster with ADI even if the pole problem is not directly addressed.

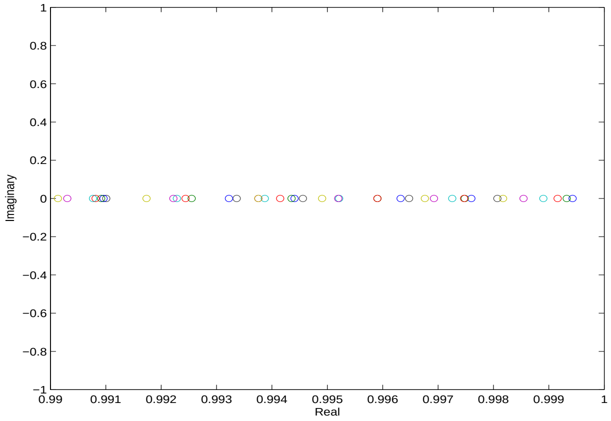

Figure 5.81: Leading eigenvalues of G_D for $k = 0$, Limited Area problem

	Stepsize	
	1^o	2^o
$\rho(A)$	7.52×10^5	9.40×10^4
$\kappa_\infty(A)$	1.21×10^5	1.52×10^4
$\kappa(P_D^{-1}A)$	5.37×10^3	1.33×10^3
$\kappa(P_{Blo}^{-1}A)$	1.11×10^3	427.327
$\kappa(P_{ADI}^{-1}A)$	270.243	104.043
$\rho(G_D)$	0.9994	0.9977
$\rho(G_{Blo})$	0.9975	0.9927
$\rho(G_{ADI})$	0.9853	0.9814
ADI value	956.760	347.958

Table 5.14: ∞-norm condition numbers, spectral radii of A, and spectral radii of G for all preconditioners for unforced limited area problem

5.4.3 Practical convergence experiments

The aim of the numerical experiments is to investigate how quickly the initial conditions we use are damped down to the zero solution (depending on their form and the preconditioned method used). We chose the initial errors (our initial guesses in this case) to be Fourier Modes of the form

$$U^0 = sin(m\lambda)sin(n\phi),$$

and investigate the number of iterations required for convergence when varying the mode numbers m, n.

A different convergence criterion is required for these experiments as $\| \mathbf{b} \|_\infty$ is effectively zero. We use the relative residual error normalised by the initial residual :

$$\frac{\| \mathbf{r}^m \|_\infty}{\| \mathbf{r}^0 \|_\infty} = \frac{\| \mathbf{b} - A\mathbf{U}^m \|_\infty}{\| \mathbf{b} - A\mathbf{U}^0 \|_\infty} < 10^{-5}. \tag{5.7}$$

Table 5.15 shows the number of iterations required for convergence using this criterion. The low mode number cases follow the general pattern of previous experiments (and the pattern suggested by the convergence data in Table 5.14) with ADI performing faster than Block and then diagonal. However for cases where m and n are large we note that diagonal outperforms block (and even ADI for very high mode number cases). Overall, however ADI and to an extent block methods, 'damp' the Fourier mode errors more evenly which, from an overall convergence point of view, is more effective.

Figures 5.82 to 5.90 show examples of the convergence history of the residual errors, at equatorial, mid, and polar latitudes, for the three preconditioners considered in this section. Although the ADI and Block preconditioners yield faster convergence overall than the diagonal preconditioner (i.e. they damp the error modes more evenly), their convergence histories show that the residual errors in the polar regions are still slower to converge than those at lower latitudes.

		Diagonal		Block		ADI	
m	n	1^o	2^o	1^o	2^o	1^o	2^o
2	2	191	78	88	44	52	28
4	4	107	50	82	40	36	18
6	6	50	24	72	34	21	15
8	8	24	12	59	27	17	12
10	10	15	5	48	21	16	11
20	20	3	NA	16	NA	9	NA
2	4	203	83	88	43	51	26
2	6	205	83	88	43	45	26
2	8	208	84	82	43	42	18
2	10	208	84	82	40	37	16
2	20	209	NA	46	NA	33	NA

Table 5.15: Number of iterations to convergence tolerance of $\frac{||\mathbf{r}^m||_\infty}{||\mathbf{r}^o||_\infty} < 10^{-5}$, for various Fourier modes defined by m,n

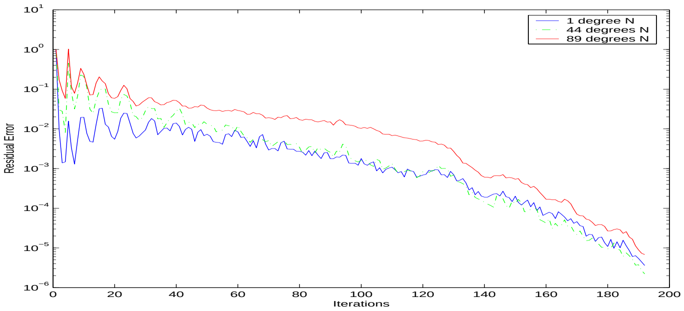

Figure 5.82: Logarithmic convergence history of residual errors at three latitudes, 1^o stepsize, Diagonal Preconditioner, $m = 2$, $n = 2$

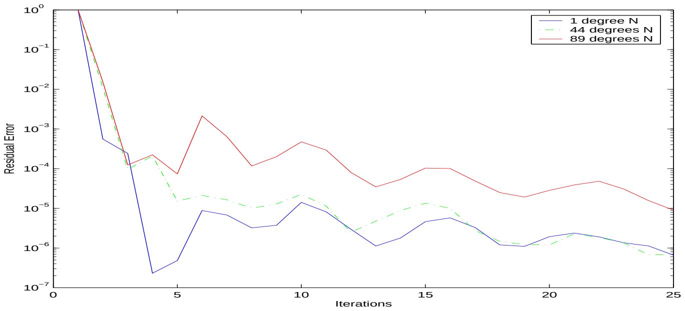

Figure 5.83: Logarithmic convergence history of residual errors at three latitudes, 1^o stepsize, Diagonal Preconditioner, $m = 8$, $n = 8$

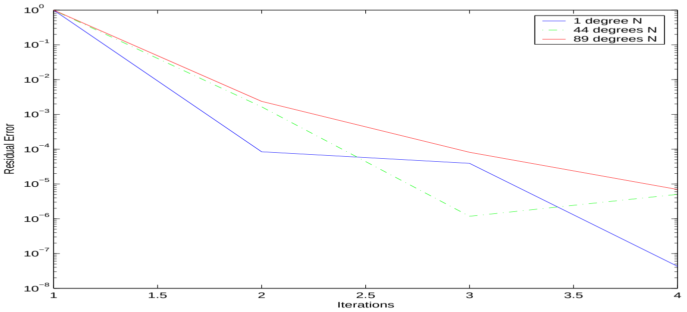

Figure 5.84: Logarithmic convergence history of residual errors at three latitudes, 1^o stepsize, Diagonal Preconditioner, $m = 20$, $n = 20$

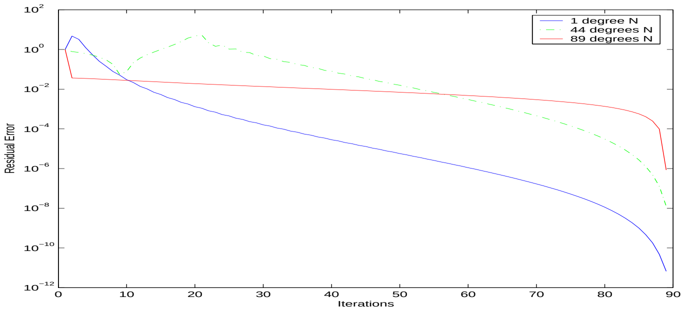

Figure 5.85: Logarithmic convergence history of residual errors at three latitudes, 1^o stepsize, Block Preconditioner, $m = 2$, $n = 2$

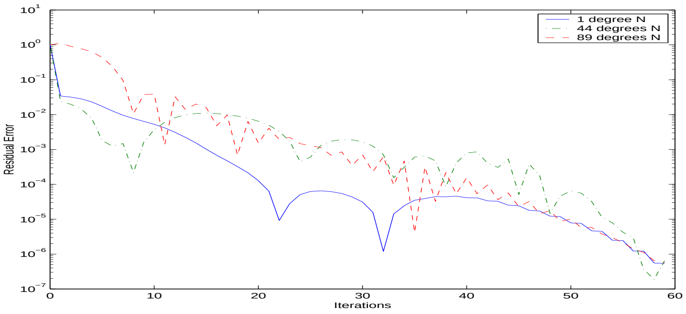

Figure 5.86: Logarithmic convergence history of residual errors at three latitudes, 1^o stepsize, Block Preconditioner, $m = 8$, $n = 8$

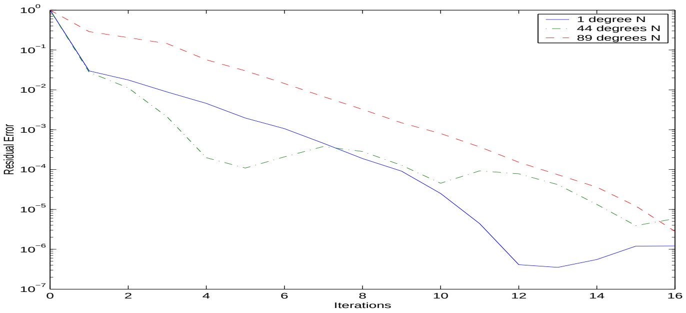

Figure 5.87: Logarithmic convergence history of residual errors at three latitudes, 1^o stepsize, Block Preconditioner, $m = 20$, $n = 20$

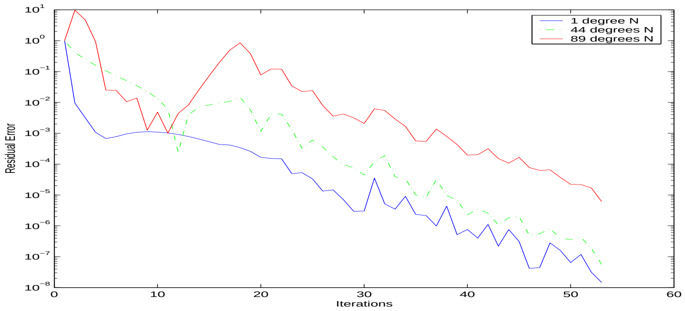

Figure 5.88: Logarithmic convergence history of residual errors at three latitudes, 1^o stepsize, ADI Preconditioner, $m = 2$, $n = 2$

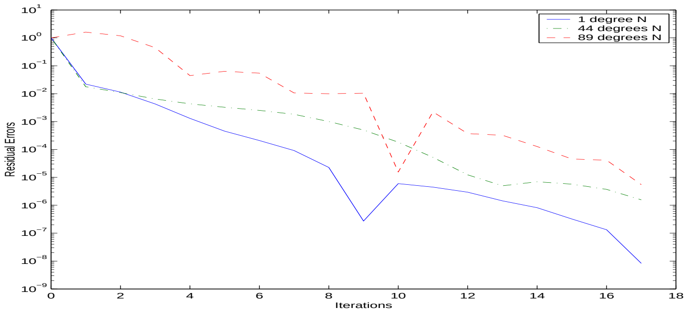

Figure 5.89: Logarithmic convergence history of residual errors at three latitudes, 1^o stepsize, ADI Preconditioner, $m = 8$, $n = 8$

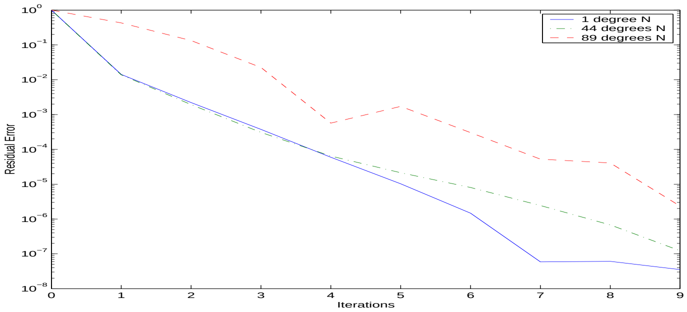

Figure 5.90: Logarithmic convergence history of residual errors at three latitudes, 1^o stepsize, ADI Preconditioner, $m = 20$, $n = 20$

5.5 Summary

This chapter presented numerical results from using the spherical model introduced in Chapter 4. Overall the theoretical findings of Chapter 4 were largely confirmed with the ADI preconditioned yielding the fastest convergence followed by block, diagonal and Binormalization preconditioners respectively. The ADI preconditioner with spatially varying parameter did not perform very well in the experiments. This is likely to have been caused by the sensitivity of the preconditioner to the parameter values that are used, combined with the fact that the values were calculated using the crude Gerschgorin estimates. For very large matrices, accurate calculation of the eigenvalue bounds and hence the parameter values is likely to be very expensive for the form of the preconditioner chosen (with one parameter per row of the grid) with possibly little gain in convergence speed. A possible compromise would be to try with a small number of values (2 to 10) calculated more accurately.

It was noted that the condition numbers of the preconditioned systems, and the associated spectral radii of the iteration matrices G, changed very little as the anisotropy was increased by moving the northern boundary closer to the pole, when using the diagonal and block diagonal preconditioners in the Limited Area problem (and to a lesser extent in the Periodic domain problem). Also it was observed that the leading eigenvectors of G did not have strong polar signals. However it was also noted that the 'nearly' leading eigenvalues of the iteration matrices G became larger as ϕ_{NB} was increased, clustering around the lead eigenvalue, and that the associated 'nearly' leading eigenvectors did have strong polar signals. Therefore it was deduced that the polar convergence issue is caused by the increased importance, in more mesh anisotropic problems, of secondary eigenvectors with strong polar signals. These findings for the Limited Area problem are summarised in our paper [12]. It was also noted that the non-zero eigenvalues of the iteration matrices for the block and diagonal preconditioners occurred in ± pairs as predicted. Finally we showed that the eigenvalues of the iteration matrix, G_{Bin} for the Binormalization preconditioned method were all strictly positive.

The leading four eigenvectors of the block preconditioned (and ADI for the periodic case) iteration matrices were seen to display smaller polar signals than the diagonal preconditioned iteration matrices. Also block and ADI preconditioning were shown to damp the spectrum of Fourier error modes more evenly than diagonal preconditioning. Despite this the convergence histories of all three preconditioners showed that the residual errors in the polar regions were the last to converge.

Chapter 6

Spherical domain : varying depth, H, problems

6.1 Introduction

In this chapter we progress to investigating problems which include a varying depth function, H, within the elliptic operators, such as that used in both the rigid-lid and free surface formulations. We firstly examine, in Section 6.2, the free surface case where the elliptic operator is of the form $-\nabla \cdot (H\nabla)U + kU$. We then consider, in Section 6.3, the corresponding rigid-lid formulation where the operator is of Poisson type $-\nabla \cdot (\frac{1}{H}\nabla)$.

6.2 Varying ocean depth H

In this section we shall describe the discretisation formulation used for a H varying Modified Helmholtz problem. We also derive Gerschgorin estimates of the conditioning of the preconditioned system matrices, in an analogous way to the previous chapter, and again perform numerical experiments to test our theoretical findings. In particular we examine a model of an idealised Continental Shelf and investigate the effect a sharp change in topography has on the numerics of the problem.

6.2.1 Problem formulation and discretisation

We now consider a Modified Helmholtz type problem, with varying topography, of the form $-\nabla \cdot (H\nabla) + kU = \gamma$ with $k \geq 0$ and we assume that the topography is a function of λ and ϕ with $H = H(\lambda, \phi) > 0$ across the whole domain. We retain our fixed mesh of $n_\lambda \times n_\phi$ grid points. We return to our theoretical domain of a segment of Northern Hemisphere ocean with Dirichlet boundary conditions at all boundaries. We solve the following Limited Area problem

$$\begin{cases} -\frac{1}{cos\phi}\left[\frac{\partial}{\partial\lambda}\left(\frac{H}{cos\phi}\frac{\partial U}{\partial\lambda}\right) + \frac{\partial}{\partial\phi}\left(Hcos\phi\frac{\partial U}{\partial\phi}\right)\right] + kU = \gamma(\lambda, \phi) \\ \lambda \in (30^oW, 0^oW) \quad \phi \in (10^oN, \phi_{NB}) \\ U(30^oW, \phi) = 0, U(0^oW, \phi) = 0 \\ U(\lambda, 10^oN) = 0, U(\lambda, \phi_{NB}) = 0 \\ \phi_{NB} \in (40^oN, 89.5^oN) \\ H = H(\lambda, \phi) > 0. \end{cases} \tag{6.1}$$

The following five-point discretisation scheme is used for this problem

$$\begin{aligned} &-\left[\frac{1}{cos^2\phi_j}\left(\frac{(U_{i+1j}-U_{ij})H_{i+\frac{1}{2},j}}{\delta\lambda^2} - \frac{(U_{ij}-U_{i-1j})H_{i-\frac{1}{2},j}}{\delta\lambda^2}\right)\right. \\ &\left.+\frac{1}{cos\phi_j}\left(\frac{cos\phi_{j+\frac{1}{2}}H_{i,j+\frac{1}{2}}(U_{ij+1}-U_{ij})}{\delta\phi^2} - \frac{cos\phi_{j-\frac{1}{2}}H_{i,j-\frac{1}{2}}(U_{ij}-U_{ij-1})}{\delta\phi^2}\right)\right] + kU_{ij} = \gamma(\mu_i, \phi_j). \end{aligned} \tag{6.2}$$

The variables have been positioned in a 'B' grid format as shown in Figure 2.2. The H values in the scheme (6.2) are calculated by taking an average of the two H values at the half step points either side. e.g $H_{i,j+\frac{1}{2}} = \frac{1}{2}\left[H(\lambda_{i-\frac{1}{2}}, \phi_{j+\frac{1}{2}}) + H(\lambda_{i+\frac{1}{2}}, \phi_{j+\frac{1}{2}})\right]$. We again use the natural ordering for our grid-points. Therefore in this case our system matrix A has the structure

$$A = \begin{pmatrix} D_1 & C_1 & & & & \\ B_2 & D_2 & C_2 & & & \\ & B_3 & D_3 & C_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & & D_{n_\phi - 1} & C_{n_\phi - 1} \\ & & & & B_{n_\phi} & D_{n_\phi} \end{pmatrix},$$

where

$$D_j = tridiag \left[\begin{array}{c} -\frac{H_{i-\frac{1}{2},j}}{cos\phi_j \delta\lambda^2}, \\ \frac{H_{i-\frac{1}{2},j}}{cos\phi_j \delta\lambda^2} + \frac{H_{i+\frac{1}{2},j}}{cos\phi_j \delta\lambda^2} + \frac{cos\phi_{j+\frac{1}{2}} H_{i,j+\frac{1}{2}}}{\delta\phi^2} + \frac{cos\phi_{j-\frac{1}{2}} H_{i,j-\frac{1}{2}}}{\delta\phi^2} + kcos\phi_j, \\ -\frac{H_{i+\frac{1}{2},j}}{cos\phi_j \delta\lambda^2}, \end{array} \right] \tag{6.3}$$

$$B_j = diag \left[-\frac{cos\phi_{j-\frac{1}{2}} H_{i,j-\frac{1}{2}}}{\delta\phi^2} \right] \qquad 2 \leq j \leq n_\phi, \tag{6.4}$$

$$C_j = diag \left[-\frac{cos\phi_{j+\frac{1}{2}} H_{i,j+\frac{1}{2}}}{\delta\phi^2} \right] \qquad 1 \leq j \leq n_\phi - 1. \tag{6.5}$$

From the definitions (6.3) to (6.5), it is straightforward to observe that each block D_j is symmetric and this, combined with the fact that

$$B_j = diag \left[-\frac{cos\phi_{j-\frac{1}{2}} H_{i,j-\frac{1}{2}}}{\delta\phi^2} \right] = C_{j-1}, \tag{6.6}$$

is enough for us to conclude that the matrix A is symmetric.

6.2.2 Properties of A and the preconditioned methods

In this section we confirm that the matrix properties that we want in order to guarantee the convergence of our preconditioned methods still hold in these extended cases. Note firstly that the form of the system matrix A is analogous with the system matrix used in Chapter 4. The difference is with the addition of H terms representing the ocean depth. By definition these are taken to be strictly greater than zero. Since we are still assuming that we have $\delta\lambda, \delta\phi > 0$ and $cos\phi \in (0, 1)$ then we may deduce in an analogous manner to Section 4.4.1 that the matrix entries we assume to be non-zero cannot become zero anywhere in the domain. From this we may deduce that the connected graph of the system matrix A is strongly connected and therefore, via Theorem 3.1, that our matrix is irreducible.

We now consider the diagonal dominance of A. For the cases with $k > 0$ (with Dirichlet or periodic boundary conditions in the λ direction) we observe that we have

$$a_{ii} > \sum_{j=1, j\neq i}^{n} \mid a_{ij} \mid,$$

for a general row. We therefore have a matrix which is strictly diagonally dominant. For the cases with $k = 0$ we do not have strict diagonal dominance except in certain rows (with either set of boundary conditions in the λ direction. We have diagonal dominance in all other rows hence our matrix is still irreducibly diagonally dominant. In addition since $a_{ii} > 0$, and $a_{ij} \leq 0$ for $i \neq j$ we again have via Theorem 3.2 that A is nonsingular with strictly positive eigenvalues and is positive definite. We may also deduce by definition that A is a Stieltjes matrix and therefore, via Theorems 3.3 and 3.4, that A is an M-matrix with $A^{-1} > 0$. Further since A is a block-tridiagonal matrix it follows using Theorem 3.15 that A is consistently ordered and hence via Theorem 3.13 that A has property A.

Since all of the diagonal entries of A are strictly positive and A is strictly or irreducibly diagonally dominant and positive-definite, we may deduce, using Theorem 3.11 that the diagonal preconditioned method is convergent. For the block-preconditioner we use the same setup as given in (4.23). As

P in this case can be shown to be symmetric positive-definite, and since A is an M-matrix, it may be deduced using Theorem 3.12 that the block preconditioned method is convergent. For the ADI preconditioners we again use matrices H_Υ and V_Υ of the forms given in (4.53) and (4.54) where

$$D_j^H = tridiag \begin{bmatrix} -\frac{H_{i-\frac{1}{2},j}}{cos\phi_j \delta\lambda^2}, \\ \frac{H_{i-\frac{1}{2},j}}{cos\phi_j \delta\lambda^2} + \frac{H_{i+\frac{1}{2},j}}{cos\phi_j \delta\lambda^2} + \frac{kcos\phi_j}{2}, \\ -\frac{H_{i+\frac{1}{2},j}}{cos\phi_j \delta\lambda^2}, \end{bmatrix}, \quad 1 \leq j \leq n_\phi, \tag{6.7}$$

and

$$D_j^V = diag \left[\frac{cos\phi_{j+\frac{1}{2}} H_{i,j+\frac{1}{2}}}{\delta\phi^2} + \frac{cos\phi_{j-\frac{1}{2}} H_{i,j-\frac{1}{2}}}{\delta\phi^2} + \frac{kcos\phi_j}{2} \right], \quad 1 \leq j \leq n_\phi. \tag{6.8}$$

From this it can be shown that if $k > 0$, using either Dirichlet or periodic boundary conditions in the λ direction, or if $k = 0$ for Dirichlet cases, then the matrices H_Υ and V_Υ are strictly or irreducibly diagonally dominant, are positive-definite with strictly positive eigenvalues, and are therefore Stieltjes matrices. Therefore it may be concluded in those cases that the ADI preconditioned methods converge for $\Upsilon > 0$. We cannot guarantee the convergence for cases where periodic boundary conditions are used in the λ direction with $k = 0$.

6.2.3 Gerschgorin convergence estimates

In this section, in an analogous way to Section 4.5, we use the Gerschgorin Theorem 3.6 to put bounds on the spectral radii and condition numbers of the preconditioned matrices. We now have a domain with a varying topography function. We will only consider a general case here where we assume that we know only the largest depth, $\hat{H}$ across the whole domain and we will derive crude bounds using that to obtain qualitative information on the preconditioned methods. We again assume that we have a constant stepsize in both directions i.e. $h = \delta\lambda = \delta\phi$. We find that we have

$$\rho(A) \leq \frac{4\hat{H}}{cos\phi_{High}} + 2\hat{H}cos\phi_{High} + \hat{H}cos\phi_{High} + h^2kcos\phi_{High}, \tag{6.9}$$

$$\mu_{min}(A) \geq h^2kcos\phi_{High}. \tag{6.10}$$

Therefore we have

$$\kappa_2(A) \leq \frac{4\hat{H}}{h^2kcos^2\phi_{High}} + \frac{3\hat{H}}{h^2k} + 1. \tag{6.11}$$

When using a diagonal preconditioner we have

$$\rho(G_D) \leq \frac{2 + 2cos^2\phi_{High}}{2 + 2cos^2\phi_{High} + \frac{h^2k}{\hat{H}}cos^2\phi_{High}} \tag{6.12}$$

$$\implies \rho(G_D) < 1.$$

Also

$$\rho(P^{-1}A) \leq \frac{4 + 4cos^2\phi_{High}}{2 + 2cos^2\phi_{High} + \frac{h^2k}{\hat{H}}cos^2\phi_{High}}, \tag{6.13}$$

$$\mu_{min}(P^{-1}A) \geq \frac{\frac{h^2k}{\hat{H}}cos^2\phi_{High}}{2 + 2cos^2\phi_{High} + \frac{h^2k}{\hat{H}}cos^2\phi_{High}}, \tag{6.14}$$

hence

$$\kappa(P^{-1}A) \leq \frac{4}{\frac{h^2k}{\hat{H}}cos^2\phi_{High}} + \frac{4cos\phi_{High}}{\frac{h^2k}{\hat{H}}}. \tag{6.15}$$

We again observe that for the diagonal preconditioner

$$\kappa_2(P_D^{-1}A) < \kappa_2(A).$$

For the block preconditioner we have

$$\rho(G_{Block}) \leq \frac{2}{2 + \frac{h^2k}{\hat{H}}} \tag{6.16}$$

$$\implies \rho(G_{Block}) < 1.$$

Again we note that

$$\rho(G_{Block}) < \rho(G_D) < 1.$$

Also we have

$$\rho(P^{-1}A) \leq \frac{8 + \frac{h^2k}{\hat{H}}}{6 + \frac{h^2k}{\hat{H}}}, \tag{6.17}$$

$$\mu_{min}(P^{-1}A) \geq \frac{\frac{h^2k}{\hat{H}}}{2 + \frac{h^2k}{\hat{H}}}, \tag{6.18}$$

hence

$$\kappa(P^{-1}A) \leq \frac{\frac{8+\frac{h^2k}{\hat{H}}}{6+\frac{h^2k}{\hat{H}}}}{\frac{\frac{h^2k}{\hat{H}}}{2+\frac{h^2k}{\hat{H}}}}. \tag{6.19}$$

For the ADI preconditioner we have

$$\rho(G_{ADI}) \leq \frac{\Upsilon - \frac{h^2kcos\phi_{n_\phi}}{2\hat{H}}}{\Upsilon + \frac{h^2kcos\phi_{n_\phi}}{2\hat{H}}} < 1, \tag{6.20}$$

$$\rho(P^{-1}A) \leq 1 - \frac{\Upsilon - \frac{4}{cos\phi_{n_\phi}} - \frac{h^2kcos\phi_{n_\phi}}{2\hat{H}}}{\Upsilon + \frac{4}{cos\phi_{n_\phi}} + \frac{h^2kcos\phi_{n_\phi}}{2\hat{H}}} . \frac{\Upsilon - 4cos\phi_{n_\phi}cos\phi_{\frac{1}{2}} - \frac{h^2kcos\phi_{n_\phi}}{2\hat{H}}}{\Upsilon + 4cos\phi_{n_\phi}cos\phi_{\frac{1}{2}} + \frac{h^2kcos\phi_{n_\phi}}{2\hat{H}}}, \tag{6.21}$$

$$\mu_{min}(P^{-1}A) \geq 1 - \frac{\Upsilon - \frac{h^2kcos\phi_{n_\phi}}{2\hat{H}}}{\Upsilon + \frac{h^2kcos\phi_{n_\phi}}{2\hat{H}}} . \frac{\Upsilon - \frac{h^2kcos\phi_{n_\phi}}{2\hat{H}}}{\Upsilon + \frac{h^2kcos\phi_{n_\phi}}{2\hat{H}}}, \tag{6.22}$$

hence

$$\kappa(P^{-1}A) \leq \frac{1 - \frac{\Upsilon - \frac{4}{cos\phi_{High}} - \frac{h^2kcos\phi_{High}}{2\hat{H}}}{\Upsilon + \frac{4}{cos\phi_{High}} + \frac{h^2kcos\phi_{High}}{2\hat{H}}} . \frac{\Upsilon - 4cos\phi_{High} - \frac{h^2kcos\phi_{High}}{2\hat{H}}}{\Upsilon + 4cos\phi_{High} + \frac{h^2kcos\phi_{High}}{2\hat{H}}}}{1 - \frac{\Upsilon - \frac{h^2kcos\phi_{High}}{2\hat{H}}}{\Upsilon + \frac{h^2kcos\phi_{High}}{2\hat{H}}} . \frac{\Upsilon - \frac{h^2kcos\phi_{High}}{2\hat{H}}}{\Upsilon + \frac{h^2kcos\phi_{High}}{2\hat{H}}}}, \tag{6.23}$$

We note that the conditioning and spectral radii estimates for the block preconditioned method do not depend on $cos\phi$. We would therefore expect the block preconditioner to outperform the diagonal preconditioner in this extended case. Again the ADI estimates do not tell us very much generally. The pattern of

Prec.	$P^{-1}A$	$\rho(G_P)$
None	$1.17{\times}10^{13}$	-
D	$1.17{\times}10^{13}$	$1.0-2{\times}10^{-13}$
Block	$7.01{\times}10^{9}$	$1.0-2{\times}10^{-10}$
ADI	$1.24{\times}10^{9}$	$1.0-8{\times}10^{-7}$

Table 6.1: Spectral radii of the iteration matrix G and 2-norm condition numbers of $P^{-1}A$ calculated theoretically using Gerschgorin

likely convergence of the preconditioned methods is made clearer by Table 6.1 which shows the typical values of the spectral radii and the conditioning for the preconditioned methods for the case where $\hat{H} = 8000$, $k = 0.01$, $h = 1^o$ and $cos\phi_{High} = 0.03$ (equivalent to taking $\phi_{NB} = 88^o$). We note that the pattern of spectral radii and condition numbers again suggests that ADI should yield the fastest convergence followed by block preconditioning and then diagonal preconditioning.

6.2.4 Numerical experiments

In our experiments we examine a Continental Shelf case where the ocean topography is a constant depth of $2000m$ across the domain apart from a 4^o wide trench of depth $8000m$ in the longitudinal direction which runs the entire length of the domain. These conditions yield the following Limited

Area problem

$$\begin{cases} \frac{1}{cos\phi}\left[\frac{\partial}{\partial\lambda}\left(\frac{H}{cos\phi}\frac{\partial U}{\partial\lambda}\right)+\frac{\partial}{\partial\phi}\left(Hcos\phi\frac{\partial U}{\partial\phi}\right)\right]+kU=\gamma(\lambda,\phi) \\ \lambda\in(30^oW,0^oW) \quad \phi\in(10^oN,\phi_{NB}) \\ U(30^oW,\phi)=0, U(0^oW,\phi)=0 \\ U(\lambda,10^oN)=0, U(\lambda,\phi_{NB})=0 \\ \phi_{NB}\in(40^oN,89.5^oN) \\ H(30^oW-18^oW,\phi)=2000m \\ H(17^oW-14^oW,\phi)=8000m \\ H(13^oW-0^oW,\phi)=2000m. \end{cases} \tag{6.24}$$

Table 6.2 gives the conditioning and spectral radii information computed in our numerical experiments. Again we note the same pattern to the values as in previous experiments : The diagonal preconditioner contributes the highest condition number of the preconditioned system and the highest value for the spectral radii of the iteration matrix, G. The Block preconditioner has the next highest values with ADI yielding the lowest. We would therefore again expect the ADI preconditioned method to converge in the fewest iterations followed by Block and then Diagonal.

The leading eigenvectors of G for the diagonal and Block preconditioned methods showed little sign of being affected by the topography profile of H. For example the leading eigenvectors for the $\phi_{NB}=88^o, h=1^o, k=0.01$ case appeared almost exactly identical to Figures 5.16 to 5.19 and 5.28 to 5.31 respectively. The leading eigenvectors of G_{ADI} did appear to be sensitive to the height profile as Figures 6.1 to 6.4 show. The signal is all along the trench. The dominant eigenmodes still have small horizontal scales and still have larger components at the pole.

Tables 6.3 and 6.4 show the results of our practical convergence experiments in this extended, H-varying case. We again used a right hand source

	Stepsize	
	1^o	2^o
$\kappa_\infty(A)$	2.282×10^4	4.138×10^3
$\kappa(P_D^{-1}A)$	2.828×10^3	680.434
$\kappa(P_{Blo}^{-1}A)$	1.366×10^3	311.145
$\kappa(P_{ADI}^{-1}A)$	228.022	96.967
$\rho(G_D)$	0.9978	0.9917
$\rho(G_{Blo})$	0.9941	0.9776
$\rho(G_{ADI})$	0.9655	0.9323
ADI parameter	1.384×10^3	577.5

Table 6.2: ∞-norm condition numbers, spectral radii of A, and spectral radii of G for all preconditioners for Limited Area problem, $\phi_{NB} = 88^o$

function $\gamma(\lambda, \phi)$ fixed to yield a sine function general solution of

$$\begin{aligned} &U(\lambda,\phi) = sin(3\lambda)sin(d[\phi-10]) \\ &d = \tfrac{90}{\phi_{NB}-10} \end{aligned} \tag{6.25}$$

which is consistent with the chosen boundary conditions. A constant 'initial guess' of $U(i,j) = 1.5$ was again taken to start the iterative process. The relative residual ∞ norm error normalised by the source vector, (b), was used in the stopping criterion. We again found that the ADI preconditioned method yielded the fastest convergence followed by the block and then diagonally preconditioned methods. This was as we predicted qualitatively using Gerschgorin techniques in Section 6.2.3.

	Stepsize		
Prec.	$\frac{1}{2}^o$	1^o	2^o
D	387	179	81
Block	156	77	38
ADI	49	31	20

Table 6.3: Iterations to convergence, H-varying problem, $\phi_{NB} = 88^o$

	Stepsize		
Prec.	$\frac{1}{2}^o$	1^o	2^o
D	55.0	4.7	1.3
Block	44.3	4.3	1.2
ADI	30.4	3.6	1.0

Table 6.4: CPU times, H-varying problem, $\phi_{NB} = 88^o$

6.3 Varying ocean depth ($\frac{1}{H}$) operator

We now move on to describe the discretisation formulation used for a $\frac{1}{H}$ varying Poisson problem. We continue using the model of an idealised Continental Shelf and investigate the effect a sharp change in topography has on the numerics of the problem with this operator. We also revisit the constant topography case to consider some properties of the Chebyshev Semi-Iterative method.

6.3.1 Problem formulation and discretisation

We consider a Poisson type problem of the form $-\nabla \cdot (\frac{1}{H}\nabla)U = \gamma$. We retain our fixed mesh of $n_\lambda \times n_\phi$ grid points on the domain of a theoretical segment of Northern Hemisphere ocean. The domain is the same as that used in the experiments of Section 5.4. We retain Dirichlet boundary conditions at all boundaries and use ($\frac{1}{H}$) varying topography. We thus consider the following

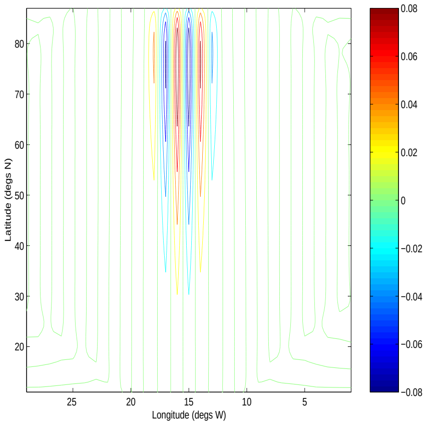

Figure 6.1: Eigenvector associated with largest eigenvalue (-0.9655) of G_{ADI} for H-varying problem. $\phi_{NB} = 88^o$.

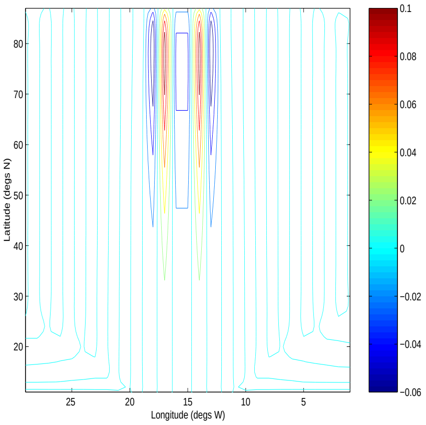

Figure 6.2: Eigenvector associated with second largest eigenvalue (-0.9622) of G_{ADI} for H-varying problem. $\phi_{NB} = 88^o$.

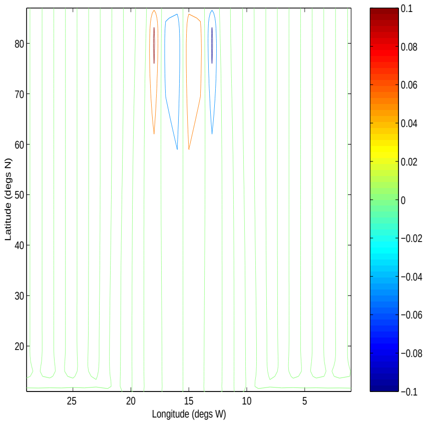

Figure 6.3: Eigenvector associated with third largest eigenvalue (-0.9545) of G_{ADI} for H-varying problem. $\phi_{NB} = 88^o$.

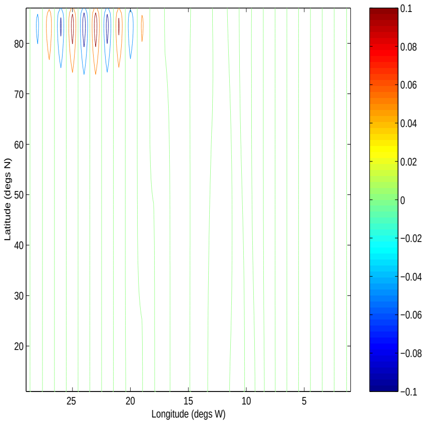

Figure 6.4: Eigenvector associated with third largest eigenvalue (-0.9515) of G_{ADI} for H-varying problem. $\phi_{NB} = 88^o$.

problem :

$$\begin{cases} \frac{1}{cos\phi}\left[\frac{\partial}{\partial\lambda}\left(\frac{1}{Hcos\phi}\frac{\partial U}{\partial\lambda}\right)+\frac{\partial}{\partial\phi}\left(\frac{cos\phi}{H}\frac{\partial U}{\partial\phi}\right)\right]=\gamma(\lambda,\phi) \\ \lambda\in(90^oW,0^oW) \quad \phi\in(0^oN,90^oN) \\ U(90^oW,\phi)=0,U(0^oW,\phi)=0 \\ U(\lambda,0^oN)=0,U(\lambda,90^oN)=0 \\ H=H(\lambda,\phi). \end{cases} \tag{6.26}$$

The following five-point discretisation scheme is used, with the $\frac{1}{H}$ values calculated by averaging, in an analogous manner to the H varying case :

$$\begin{aligned} &-\left[\frac{1}{cos^2\phi_j}\left(\frac{(U_{i+1j}-U_{ij})}{H_{i+\frac{1}{2},j}\delta\lambda^2}-\frac{(U_{ij}-U_{i-1j})}{H_{i-\frac{1}{2},j}\delta\lambda^2}\right)\right. \\ &\left.+\frac{1}{cos\phi_j}\left(\frac{cos\phi_{j+\frac{1}{2}}(U_{ij+1}-U_{ij})}{H_{i,j+\frac{1}{2}}\delta\phi^2}-\frac{cos\phi_{j-\frac{1}{2}}(U_{ij}-U_{ij-1})}{H_{i,j-\frac{1}{2}}\delta\phi^2}\right)\right]+kU_{ij}=\gamma(\mu_i,\phi_j). \end{aligned} \tag{6.27}$$

6.3.2 Properties of A and the preconditioned method

In these experiments the Red-Black ordering is used to order the grid-points. The resulting matrix A is then normalised by multiplying through by the inverse of the diagonal elements i.e. a diagonal preconditioning. Our preconditioned problem therefore has the structure

$$P^{-1}A=\begin{pmatrix} I_1 & F \\ F^* & I_2 \end{pmatrix}, \tag{6.28}$$

where I_1 and I_2 are Identity matrices of size $n_r{\times}n_r$ and $n_b{\times}n_b$ respectively. The matrices F and F^* are of size $n_r{\times}n_b$ and $n_b{\times}n_r$ respectively and $P=D=diag(A)$.

The H values used are assumed to be strictly greater than zero, by definition. Since we are still assuming that we have $\delta\lambda,\delta\phi>0$ and $cos\phi\in(0,1)$ then we may deduce in an analogous manner to Section 4.4.1 that the matrix entries we assume to be non-zero cannot become zero anywhere in the domain. From this we may deduce that the connected graph of the system matrix A is strongly connected and therefore, via Theorem 3.1, that our matrix is irreducible. We also have

$$\sum_{j=1}^{n_r} \mid f_{ij} \mid \leq 1 \quad 1 \leq i \leq n_b \tag{6.29}$$

and

$$\sum_{j=1}^{n_b} \mid f^*_{ij} \mid \leq 1 \quad 1 \leq i \leq n_r \tag{6.30}$$

where $f_{ij} \in F$ and $f^*_{ij} \in F^*$ with strict inequality in at least one row of each relation. The matrix A is therefore diagonally dominant with strict diagonal dominance in at least one row. Since A is irreducible it follows that it is irreducibly diagonally dominant. We may therefore deduce, via Theorem 3.11, that the diagonal preconditioned method is convergent. In addition since $a_{ii} > 0$, and $a_{ij} \leq 0$ for $i \neq j$, we again have, via Theorem 3.2 that A is nonsingular with strictly positive eigenvalues and is positive definite. We may also deduce by definition that A is a Stieltjes matrix and therefore, via Theorems 3.3 and 3.4, that A is an M-matrix with $A^{-1} > 0$. Further since A is a block-tridiagonal matrix it follows using Theorem 3.15 that A is consistently ordered and hence via Theorem 3.13 that A has property A.

6.3.3 Numerical experiments

The aims of the experiments of this section are to demonstrate some of the convergence properties of the Cyclic Chebyshev Semi-Iterative method and to demonstrate the sensitivity of the elliptic operator to sharp variations in H. We consider the problem (6.26) and use a Continental shelf topography

Setup	$\rho(G_D)$	$\kappa_\infty(P_D^{-1})$
Constant H	0.9994	5.37×10^3
Continental Shelf	0.9998	5.94×10^4

Table 6.5: Spectral radii of G and condition numbers of $P_D^{-1}A$

profile of the form

$$\begin{aligned}H(90^oW - 61^oW, \phi) &= 2000m\\ H(60^oW - 31^oW, \phi) &= 8000m\\ H(30^oW - 0^oW, \phi) &= 2000m,\end{aligned}$$

to illustrate the sensitivity of the operator to sharply changing H. A constant topography (i.e. $H(\lambda, \phi) = 2000$) case is used as a 'control' in the comparison. From Table 6.5 we note that the spectral radii of G_D and, to a lesser extent, the conditioning of the system matrix are indeed sensitive to the topography profile.

We revisit the experiments of section 5.4.3 investigating the 'damping' of Fourier initial errors, this time using the Chebyshev Semi-Iterative Method. We chose the initial errors (our initial guesses in this case) to be Fourier Modes of the form

$$U^0 = sin(m\lambda)sin(n\phi),$$

and investigate the number of iterations required for convergence when varying the mode numbers m, n. We use the relative residual error normalised by the initial residual as the stopping criterion. Tables 6.6 and 6.7 show the results of doing this with constant height and Continental shelf cases. For the constant height case we note that the Chebyshev method using the spectral radius shown in Table 6.5 causes the various Fourier modes to be damped approximately evenly. This is not quite the case with the Continental Shelf

	m=2		m=20	
$\rho(G)$	n=2	n=20	n=2	n=20
0.9938	616	508	58	54
0.9967	463	381	71	75
0.9986	303	249	110	107
0.9989	269	221	122	120
0.9991	232	189	138	134
0.9993	190	160	157	156
0.9994	160	162	171	165
0.9995	142	174	185	177
0.9997	167	206	218	214
0.9999	705	1006	1043	1032

Table 6.6: Number of iterations to convergence of various Fourier Modes : Constant Depth case

model, although the spectral radius is fairly accurate : A $\rho(G)$ value of between 0.9997 and 0.9998 would appear to be ideal. It is possible that this discrepancy is due to the fact that in this $\frac{1}{H}$ varying case, Fourier modes are less representative physically of the error modes involved.

6.4 Summary

This chapter extended the basic spherical model introduced in Chapter 4 to include problems which include a varying depth function, H, within the elliptic operators. We firstly examined a case which was analogous with the free surface formulation where the elliptic operator is of the form $-\nabla \cdot (H\nabla)U + kU$. We again showed that ADI ought to yield the fastest convergence, followed by Block and Diagonal, by examination of the spectral radii of G and

	m=2		m=20	
$\rho(G)$	n=2	n=20	n=2	n=20
0.9938	1153	1042	319	255
0.9986	603	549	213	177
0.9993	422	386	171	166
0.9995	356	326	191	190
0.9997	282	262	225	226
0.9998	195	252	274	286
0.9999	764	1010	1082	1102

Table 6.7: Number of iterations to convergence of various Fourier Modes : Continental Shelf case

the conditioning of the preconditioned systems. However it was noted that whilst the leading eigenvectors of G_D and G_{Block} were not very sensitive to the height profile the ADI preconditioned iteration matrix, G_{ADI}, was. We then moved on to consider problems where the operator is of Poisson type $-\nabla \cdot (\frac{1}{H}\nabla)$ similar to the rigid-lid formulation. We showed that by considering the conditioning of the system matrix as well as the size of the spectral radii of G_D that the problem is sensitive to the variations in H. We also highlighted the importance of using an accurate value for the spectral radii of G_D. Using an accurate value causes the convergence of all error modes to be approximately equal, when using the Chebyshev semi-iterative method, in the constant depth case. The convergence of the modes varied a lot more with the use of less accurate choices of the spectral radii, $\rho(G_D)$. The slight discrepancy in the Continental shelf case was attributed to the fact that Fourier modes were less physically representative of the errors involved in this case.

Chapter 7

The nine point operator

7.1 Introduction

We move on, in this chapter, to investigate the use of a special nine-point discretisation operator of the general form used for solving the free surface problem. We discuss the exact form of the discretisation of the full operator in the next section, putting the operator in the context of the B grid discussed in Section 2.2.3 and detailing its discretisation stencil. We perform truncation error analysis on the discretisation operator in order to confirm its consistency. We also show that the gradient and divergence operators for the finite-difference form of the BCS model formulation have analogues of the positive-definite property. In Section 7.3 we consider a constant depth version of the nine-point operator. We investigate the convergence properties of the system matrix and consider the modification of our preconditioners for use with the nine-point operator. We also revisit the Limited Area problem of Chapters 4 and 5 with some numerical experiments.

7.2 Free-surface nine-point elliptic operator

In this section we introduce the full nine-point free-surface operator solved in the free-surface formulation of the BCS ocean model. Recall that the problem solved in the free-surface formulation is of the form

$$-\frac{1}{acos\phi}\left[\frac{\partial}{\partial\lambda}\left(\frac{H}{acos\phi}\frac{\partial\eta'}{\partial\lambda}\right)+\frac{\partial}{\partial\phi}\left(\frac{Hcos\phi}{a}\frac{\partial\eta'}{\partial\phi}\right)\right]+\beta a^2\eta'=\mathbf{S}(\lambda,\phi). \quad (7.1)$$

with appropriate boundary conditions at the boundaries of the domain (including island boundaries : see Dukowicz et al [24] and the discussion in Section 2.3.2). Also it is assumed that $H > 0$ in the interior of the ocean. In the rest of this section we will describe the full nine-point stencil used to discretise the problem (7.1). We will demonstrate the positive-definiteness of the general discrete operator using a finite difference analogue. We will also show the consistency of the discretisation scheme with the continuous problem using truncation error analysis

7.2.1 Nine-point discrete operator stencil

We consider the full free-surface operator with varying topography and give the details for the stencil in this case. Using a compass notation where a general point $P(i,j)$ is surrounded by grid points labelled as in Figure 7.1 The 'contribution' to the discretisation scheme from each direction of the

NW	N	NE
X	X	X
W	P	E
X	X	X
SW	S	SE
X	X	X

Figure 7.1: Nine-point operator stencil written in 'directional' notation

stencil is given as follows :

$$
\begin{aligned}
&P = \tfrac{1}{4}\left(\frac{\delta\phi}{\delta\lambda}\left\{\frac{[H(i+\frac{1}{2},j+\frac{1}{2})+H(i-\frac{1}{2},j+\frac{1}{2})]}{cos\phi_{j+\frac{1}{2}}}+\frac{[H(i+\frac{1}{2},j-\frac{1}{2})+H(i-\frac{1}{2},j-\frac{1}{2})]}{cos\phi_{j-\frac{1}{2}}}\right\}\right.\\
&\left.+\frac{\delta\lambda(cos\phi_{j+\frac{1}{2}}[H(i+\frac{1}{2},j+\frac{1}{2})+H(i-\frac{1}{2},j+\frac{1}{2})]+cos\phi_{j-\frac{1}{2}}[H(i+\frac{1}{2},j-\frac{1}{2})+H(i-\frac{1}{2},j-\frac{1}{2})])}{\delta\phi}\right)+\beta a^2 cos\phi_j\delta\lambda\delta\phi\\
&W = \tfrac{1}{4}\left(-\frac{\delta\phi}{\delta\lambda}\left\{\frac{H(i-\frac{1}{2},j+\frac{1}{2})}{cos\phi_{j+\frac{1}{2}}}+\frac{H(i-\frac{1}{2},j-\frac{1}{2})}{cos\phi_{j-\frac{1}{2}}}\right\}+\frac{\delta\lambda(cos\phi_{j+\frac{1}{2}}H(i-\frac{1}{2},j+\frac{1}{2})+cos\phi_{j-\frac{1}{2}}H(i-\frac{1}{2},j-\frac{1}{2}))}{\delta\phi}\right)\\
&E = \tfrac{1}{4}\left(-\frac{\delta\phi}{\delta\lambda}\left\{\frac{H(i+\frac{1}{2},j+\frac{1}{2})}{cos\phi_{j+\frac{1}{2}}}+\frac{H(i+\frac{1}{2},j-\frac{1}{2})}{cos\phi_{j-\frac{1}{2}}}\right\}+\frac{\delta\lambda(cos\phi_{j+\frac{1}{2}}H(i+\frac{1}{2},j+\frac{1}{2})+cos\phi_{j-\frac{1}{2}}H(i+\frac{1}{2},j-\frac{1}{2}))}{\delta\phi}\right)\\
&N = \tfrac{1}{4}\left(\frac{[H(i+\frac{1}{2},j+\frac{1}{2})+H(i-\frac{1}{2},j+\frac{1}{2})]\delta\phi}{cos\phi_{j+\frac{1}{2}}\delta\lambda}-\frac{cos\phi_{j+\frac{1}{2}[H(i+\frac{1}{2},j+\frac{1}{2})+H(i-\frac{1}{2},j+\frac{1}{2})]}\delta\lambda}{\delta\phi}\right)\\
&S = \tfrac{1}{4}\left(\frac{[H(i+\frac{1}{2},j-\frac{1}{2})+H(i-\frac{1}{2},j-\frac{1}{2})]\delta\phi}{cos\phi_{j-\frac{1}{2}}\delta\lambda}-\frac{cos\phi_{j-\frac{1}{2}}[H(i+\frac{1}{2},j-\frac{1}{2})+H(i-\frac{1}{2},j-\frac{1}{2})]\delta\lambda}{\delta\phi}\right)\\
&SW = \tfrac{1}{4}\left(-\frac{H(i-\frac{1}{2},j-\frac{1}{2})\delta\phi}{cos\phi_{j-\frac{1}{2}}\delta\lambda}-\frac{cos\phi_{j-\frac{1}{2}}H(i-\frac{1}{2},j-\frac{1}{2})\delta\lambda}{\delta\phi}\right)\\
&SE = \tfrac{1}{4}\left(-\frac{H(i+\frac{1}{2},j-\frac{1}{2})\delta\phi}{cos\phi_{j-\frac{1}{2}}\delta\lambda}-\frac{cos\phi_{j-\frac{1}{2}}H(i+\frac{1}{2},j-\frac{1}{2})\delta\lambda}{\delta\phi}\right)\\
&NW = \tfrac{1}{4}\left(-\frac{H(i+\frac{1}{2},j-\frac{1}{2})\delta\phi}{cos\phi_{j+\frac{1}{2}}\delta\lambda}-\frac{cos\phi_{j+\frac{1}{2}}H(i+\frac{1}{2},j-\frac{1}{2})\delta\lambda}{\delta\phi}\right)\\
&NE = \tfrac{1}{4}\left(-\frac{H(i+\frac{1}{2},j+\frac{1}{2})\delta\phi}{cos\phi_{j+\frac{1}{2}}\delta\lambda}-\frac{cos\phi_{j+\frac{1}{2}}H(i+\frac{1}{2},j+\frac{1}{2})\delta\lambda}{\delta\phi}\right),
\end{aligned}
\tag{7.2}
$$

7.2.2 Positive-definite property of operator

We now show that the free surface operator, formed from $\nabla\cdot(H\nabla)$ is negative-definite (Equivalent to finding positive-definiteness for $-\nabla\cdot(H\nabla)$). Note that we are not proving here that any particular matrix approximating the elliptic problem (7.1), with appropriate boundary conditions, is positive-definite. We are talking about a property of the general discrete operator using a finite difference analogue. We will show that the gradient and divergence operators for the finite-difference form of the BCS model formulation have analogues of the positive-definite property. It will still necessary to prove the positive-definiteness of any particular system matrix A arising from a given problem formulation. We will do this for a specific problem in Section 7.3.

We will assume that $\beta = 0$ in this analysis for the purposes of simplicity of writing. The term β is a constant which adds to the positivity of a given operator if it is greater than zero (as found in the free-surface formulation). We consider the 'worst case' where $\beta = 0$ here and prove the positive-definiteness for the finite-difference analogue of the general operator. We may then conclude positive-definiteness for the analogue of the operator with $\beta > 0$.

Recall that the free surface height η is stored at the centre of the main grid cells at points with integer indices denoted by subscripts i and j. The depth of the ocean, H, and depth integrated velocities u and v are stored at the corners of these cells at points with half integer indices. We use (u, v) to denote the analogue of $H\nabla\eta$.

The lengths of the main grid cell at the $(i, j)^{th}$ point will be denoted $\delta x_{i,j}$ and $\delta y_{i,j}$. Similarly the lengths of the cells centred at the half integer point $(i + \frac{1}{2}, j + \frac{1}{2})$ will be denoted by $\delta x_{i+\frac{1}{2},j+\frac{1}{2}}$ and $\delta y_{i+\frac{1}{2},j+\frac{1}{2}}$. The use of x and y to denote the length directions illustrates the fact that the results of this section could be applied to a variety of co-ordinate classes. Here we have a latitude-longitude grid with $\delta x_{i,j} = acos\phi_j \delta\lambda$ and $\delta y_{i,j} = a\delta\phi$. The analogue of the scalar product of two functions defined at integer points $a_{i,j}$ and $b_{i,j}$ will be defined to be

$$\sum_{i,j} a_{i,j} b_{i,j} \delta x_{i,j} \delta y_{i,j}. \tag{7.3}$$

The simplest form of the gradient operator will also be used :

$$(\nabla\eta)_{i+\frac{1}{2},j+\frac{1}{2}} = \left(\begin{array}{c} \frac{1}{2\delta x_{i+\frac{1}{2},j+\frac{1}{2}}} (\eta_{i+1,j+1} + \eta_{i+1,j} - \eta_{i,j+1} - \eta_{i,j}) \\ \frac{1}{2\delta y_{i+\frac{1}{2},j+\frac{1}{2}}} (\eta_{i+1,j+1} + \eta_{i,j+1} - \eta_{i+1,j} - \eta_{i,j}) \end{array} \right). \tag{7.4}$$

The divergence operator at a point with integer indices will be based on the integral of the flux through the faces of the cell :

$$\begin{array}{l} 2\nabla \cdot (u,v)_{i,j} \delta x_{i,j} \delta y_{i,j} = \delta x_{i+\frac{1}{2},j+\frac{1}{2}} v_{i+\frac{1}{2},j+\frac{1}{2}} + \delta x_{i-\frac{1}{2},j+\frac{1}{2}} v_{i-\frac{1}{2},j+\frac{1}{2}} \\ -\delta x_{i+\frac{1}{2},j-\frac{1}{2}} v_{i+\frac{1}{2},j-\frac{1}{2}} - \delta x_{i-\frac{1}{2},j-\frac{1}{2}} v_{i-\frac{1}{2},j-\frac{1}{2}} \\ +\delta y_{i+\frac{1}{2},j+\frac{1}{2}} u_{i+\frac{1}{2},j+\frac{1}{2}} + \delta y_{i-\frac{1}{2},j+\frac{1}{2}} u_{i-\frac{1}{2},j+\frac{1}{2}} \\ -\delta y_{i+\frac{1}{2},j-\frac{1}{2}} u_{i+\frac{1}{2},j-\frac{1}{2}} - \delta y_{i-\frac{1}{2},j-\frac{1}{2}} u_{i-\frac{1}{2},j-\frac{1}{2}}. \end{array} \tag{7.5}$$

We need to calculate (7.3) with $a_{i,j} = \eta_{i,j}$ and $b_{i,j} = \nabla \cdot (u,v)_{i,j}$ as given in (7.5) with $(u,v)_{i+\frac{1}{2},j+\frac{1}{2}} = H_{i+\frac{1}{2},j+\frac{1}{2}} (\nabla\eta)_{i+\frac{1}{2},j+\frac{1}{2}}$ and the gradient operator given by (7.4). We re-organise the summation in (7.3) gathering together the terms from $b_{i,j} = \nabla \cdot (u,v)_{i,j}$ evaluated at $\left(i+\frac{1}{2}, j+\frac{1}{2}\right)$. This corresponds to an integration by parts.

$$\begin{array}{l} \sum_{i,j} \eta_{i,j} \nabla \cdot (u,v)_{i,j} \delta x_{i,j} \delta y_{i,j} \\ = \sum_{i,j} \frac{\delta x_{i+\frac{1}{2},j+\frac{1}{2}} v_{i+\frac{1}{2},j+\frac{1}{2}}}{2} \left(\eta_{i,j} + \eta_{i+1,j} - \eta_{i,j+1} - \eta_{i+1,j+1}\right) \\ + \sum_{i,j} \frac{y_{i+\frac{1}{2},j+\frac{1}{2}} u_{i+\frac{1}{2},j+\frac{1}{2}}}{2} \left(\eta_{i,j} + \eta_{i,j+1} - \eta_{i+1,j} - \eta_{i+1,j+1}\right). \end{array} \tag{7.6}$$

Since we are taking $(u,v)_{i+\frac{1}{2},j+\frac{1}{2}} = H_{i+\frac{1}{2},j+\frac{1}{2}} (\nabla\eta)_{i+\frac{1}{2},j+\frac{1}{2}}$ in (7.6) and the depth of the ocean is zero at the boundaries of the domain, the summations on the right hand side of (7.6) will extend precisely over points where the depth is non-zero. Then using (7.4) one obtains

$$\sum_{i,j} \eta_{i,j} \left[\nabla \cdot (H\nabla\eta)\right]_{i,j} \delta x_{i,j}\delta y_{i,j} = -\sum_{i,j} H_{i+\frac{1}{2},j+\frac{1}{2}} M_{i+\frac{1}{2},j+\frac{1}{2}} \delta x_{i+\frac{1}{2},j+\frac{1}{2}} y_{i+\frac{1}{2},j+\frac{1}{2}}, \tag{7.7}$$

where

$$M_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{4}\left[\frac{(\eta_{i,j}+\eta_{i+1,j}-\eta_{i,j+1}-\eta_{i+1,j+1})^2}{\delta y^2_{i+\frac{1}{2},j+\frac{1}{2}}} + \frac{(\eta_{i,j}+\eta_{i,j+1}-\eta_{i+1,j}-\eta_{i+1,j+1})^2}{\delta x^2_{i+\frac{1}{2},j+\frac{1}{2}}}\right]. \tag{7.8}$$

The relation in (7.8) implies that the analogue of $\nabla \cdot (H\nabla\eta)$ is negative-definite for functions other than those with the analogue of $\nabla\eta = 0$.

7.2.3 Truncation error analysis

We now derive the Truncation error for the nine-point discretisation scheme and confirm its consistency with the differential equation. This property is needed to ensure the convergence of the discrete solution to that of the continuous problem as the step sizes ($\delta\lambda$ and $\delta\phi$) go to zero.

For this analysis we use the following version of the operator :

$$-\left[\frac{\partial}{\partial\lambda}\left(H\frac{\partial\eta}{\partial\lambda}\right) + \frac{\partial}{\partial\phi}\left(Hcos^2\phi\frac{\partial\eta}{\partial\phi}\right)\right] + \beta cos^2\phi a^2\eta = \gamma(\lambda,\phi). \tag{7.9}$$

Let

$$\begin{aligned} HU_{\pm\pm} &= \tfrac{H(i\pm\frac{1}{2},j\pm\frac{1}{2})}{\delta\lambda^2}, \\ HV_{\pm\pm} &= \tfrac{H(i\pm\frac{1}{2},j\pm\frac{1}{2})cos^2\phi(j\pm\frac{1}{2})}{\delta\phi^2}. \end{aligned}$$

The truncation error of the scheme is hence given by

$$
\begin{aligned}
&T.E. = \tfrac{1}{4}(HU_{++} + HU_{+-} + HU_{-+} + HU_{--} + HV_{++} + HV_{+-} + HV_{-+} + HV_{--})\eta \\
&+\tfrac{1}{4}(HU_{++} - HV_{++} + HU_{-+} - HV_{-+})(\eta + \delta\phi\eta_\phi + \tfrac{\delta\phi^2}{2!}\eta_{\phi\phi} + HOT) \\
&+\tfrac{1}{4}(HU_{+-} - HV_{+-} + HU_{--} - HV_{--})(\eta - \delta\phi\eta_\phi + \tfrac{\delta\phi^2}{2!}\eta_{\phi\phi} + HOT) \\
&+\tfrac{1}{4}(-HU_{++} + HV_{++} - HU_{+-} + HV_{--})(\eta + \delta\lambda\eta_\lambda + \tfrac{\delta\lambda^2}{2!}\eta_{\lambda\lambda} + HOT) \\
&+\tfrac{1}{4}(-HU_{-+} - HV_{-+} + HU_{--} - HV_{--})(\eta - \delta\lambda\eta_\lambda + \tfrac{\delta\lambda^2}{2!}\eta_{\lambda\lambda} + HOT) \\
&+\tfrac{1}{4}(-HU_{--} - HV_{--})(\eta - \delta\lambda\eta_\lambda - \delta\phi\eta_\phi + \tfrac{\delta\lambda^2}{2!}\eta_{\lambda\lambda} + \delta\lambda\delta\phi\eta_{\lambda\phi} + \tfrac{\delta\phi^2}{2!}\eta_{\phi\phi} + HOT) \\
&+\tfrac{1}{4}(-HU_{+-} - HV_{+-})(\eta + \delta\lambda\eta_\lambda - \delta\phi\eta_\phi + \tfrac{\delta\lambda^2}{2!}\eta_{\lambda\lambda} - \delta\lambda\delta\phi\eta_{\lambda\phi} + \tfrac{\delta\phi^2}{2!}\eta_{\phi\phi} + HOT) \\
&+\tfrac{1}{4}(-HU_{-+} - HV_{-+})(\eta - \delta\lambda\eta_\lambda + \delta\phi\eta_\phi + \tfrac{\delta\lambda^2}{2!}\eta_{\lambda\lambda} - \delta\lambda\delta\phi\eta_{\lambda\phi} + \tfrac{\delta\phi^2}{2!}\eta_{\phi\phi} + HOT) \\
&+\tfrac{1}{4}(-HU_{++} - HV_{++})(\eta + \delta\lambda\eta_\lambda + \delta\phi\eta_\phi + \tfrac{\delta\lambda^2}{2!}\eta_{\lambda\lambda} + \delta\lambda\delta\phi\eta_{\lambda\phi} + \tfrac{\delta\phi^2}{2!}\eta_{\phi\phi} + HOT) \\
&+\beta cos^2\phi a^2\eta
\end{aligned}
$$

$$
\begin{aligned}
&= \tfrac{\delta\lambda\eta_\lambda}{4}(2HU_{-+} + 2HU_{--} - 2HU_{++} - 2HU_{+-}) \\
&+\tfrac{\delta\phi\eta_\phi}{4}(-2HV_{++} - 2HV_{-+} + 2HV_{+-} + 2HV_{--}) \\
&+\tfrac{\delta\lambda^2\eta_{\lambda\lambda}}{2!.4}(-2HU_{-+} - 2HU_{++} - 2HU_{--} - 2HU_{+-}) \\
&+\tfrac{\delta\phi^2\eta_{\phi\phi}}{2!.4}(-2HV_{++} - 2HV_{-+} - 2HV_{+-} - 2HV_{--}) \\
&+\tfrac{\delta\lambda\delta\phi\eta_{\lambda\phi}}{4}(-HU_{--} - HV_{--} + HU_{+-} + HV_{+-} + HU_{-+} + HV_{-+} - HU_{++} - HV_{++}) \\
&+\beta cos^2\phi a^2\eta + HOT
\end{aligned}
$$

$$
\begin{aligned}
&= \tfrac{\delta\lambda\eta_\lambda}{2}\left[\tfrac{H(i-\frac{1}{2},j+\frac{1}{2})+H(i-\frac{1}{2},j-\frac{1}{2})-H(i+\frac{1}{2},j+\frac{1}{2})-H(i+\frac{1}{2},j-\frac{1}{2})}{\delta\lambda^2}\right] \\
&+\tfrac{\delta\phi\eta_\phi}{2}\left[\tfrac{-\left\{H(i+\frac{1}{2},j+\frac{1}{2})+H(i-\frac{1}{2},j-\frac{1}{2})\right\}cos^2\phi_{j+\frac{1}{2}}+\left\{H(i+\frac{1}{2},j-\frac{1}{2})+H(i-\frac{1}{2},j-\frac{1}{2})\right\}cos^2\phi_{j-\frac{1}{2}}}{\delta\phi^2}\right] \\
&+\tfrac{\delta\lambda^2\eta_{\lambda\lambda}}{4}\left[-\tfrac{H(i-\frac{1}{2},j+\frac{1}{2})-H(i-\frac{1}{2},j-\frac{1}{2})-H(i+\frac{1}{2},j+\frac{1}{2})-H(i+\frac{1}{2},j-\frac{1}{2})}{\delta\lambda^2}\right] \\
&+\tfrac{\delta\phi^2\eta_{\phi\phi}}{4}\left[\tfrac{-\left\{H(i+\frac{1}{2},j+\frac{1}{2})+H(i-\frac{1}{2},j+\frac{1}{2})\right\}cos^2\phi_{j+\frac{1}{2}}-\left\{H(i+\frac{1}{2},j-\frac{1}{2})+H(i-\frac{1}{2},j-\frac{1}{2})\right\}cos^2\phi_{j-\frac{1}{2}}}{\delta\phi^2}\right] \\
&+\tfrac{\delta\lambda\delta\phi\eta_{\lambda\phi}}{4}\left[\tfrac{H(i-\frac{1}{2},j+\frac{1}{2})-H(i-\frac{1}{2},j-\frac{1}{2})-H(i+\frac{1}{2},j+\frac{1}{2})+H(i+\frac{1}{2},j-\frac{1}{2})}{\delta\lambda^2}\right. \\
&\left.+\tfrac{\left\{-H(i+\frac{1}{2},j+\frac{1}{2})+H(i-\frac{1}{2},j-\frac{1}{2})\right\}cos^2\phi_{j+\frac{1}{2}}+\left\{H(i+\frac{1}{2},j-\frac{1}{2})-H(i-\frac{1}{2},j-\frac{1}{2})\right\}cos^2\phi_{j-\frac{1}{2}}}{\delta\phi^2}\right] \\
&+\beta cos^2\phi a^2\eta + HOT.
\end{aligned}
$$

where HOT are higher order terms. Expanding about $H(\lambda_i, \phi_j)$ and cancelling gives

$$\begin{aligned}
&T.E. = -H_\lambda \eta_\lambda - H\eta_{\lambda\lambda} \\
&+\tfrac{\eta_\phi}{\delta\phi}\Big[(-H - \tfrac{\delta\phi}{2}H_\phi - \tfrac{\delta\lambda^2}{4.2!}H_{\lambda\lambda} - \tfrac{\delta\phi^2}{4.2!}H_{\phi\phi})(cos^2\phi - \delta\phi cos\phi sin\phi - \tfrac{\delta\phi^2}{4}cos^2\phi + \tfrac{\delta\phi^2}{4}sin^2\phi) \\
&+(H - \tfrac{\delta\phi}{2}H_\phi + \tfrac{\delta\lambda^2}{4.2!}H_{\lambda\lambda} + \tfrac{\delta\phi^2}{4.2!}H_{\phi\phi})(cos^2\phi + \delta\phi cos\phi sin\phi - \tfrac{\delta\phi^2}{4}cos^2\phi + \tfrac{\delta\phi^2}{4}sin^2\phi)\Big] \\
&+\tfrac{\eta_{\phi\phi}}{2}\big[(-H - \tfrac{\delta\phi}{2}H_\phi)(cos^2\phi - \delta\phi cos\phi sin_\phi) + (-H + \tfrac{\delta\phi}{2}H_\phi)(cos^2\phi + \delta\phi cos\phi sin_\phi)\big] \\
&+\beta cos^2\phi a^2\eta + O(\delta\lambda^2) + O(\delta\phi^2) + O(\delta\lambda\delta\phi) \\
&= -H_\lambda\eta_\lambda + Hcos\phi sin_\phi - H_\phi cos^2\phi\eta_\phi - H\eta_{\lambda\lambda} - Hcos^2\phi\eta_{\phi\phi} + \beta cos^2\phi a^2\eta \\
&+O(\delta\lambda^2) + O(\delta\phi^2) + O(\delta\lambda\delta\phi).
\end{aligned} \tag{7.10}$$

The terms in (7.10), other than the $O(\delta\lambda^2)$, $O(\delta\phi^2)$ and $O(\delta\lambda\delta\phi)$ terms, are just the differential operator. Therefore we have

$$T.E. = O(\delta\lambda^2) + O(\delta\phi^2) + O(\delta\lambda\delta\phi),$$

which means that the scheme is consistent with the differential equation.

7.3 Nine-point operator : constant depth problem

In this section we shall consider the numerical convergence properties of the nine-point operator in a reduced case : a constant depth problem. We assume that we have $H(\lambda, \phi) = 1$ across the domain (i.e. analogous to our problems of Chapter 4 and 5). We revisit the limited area model of chapters 4 and 5, this time using the nine-point discretisation operator. We again take as our domain a theoretical segment of Northern Hemisphere ocean with Dirichlet boundary conditions at all boundaries. The problem we consider is of the form :

$$\begin{cases} -\frac{1}{cos\phi}\left[\frac{\partial}{\partial\lambda}\left(\frac{1}{cos\phi}\frac{\partial U}{\partial\lambda}\right)+\frac{\partial}{\partial\phi}\left(cos\phi\frac{\partial U}{\partial\phi}\right)\right]+kU=\gamma(\lambda,\phi) \\ \lambda\in(0^oE,30^oE) \quad \phi\in(10^oN,\phi_{NB}) \\ U(0^oE,\phi)=0, U(30^oE,\phi)=0 \\ U(\lambda,10^oN)=0, U(\lambda,\phi_{NB})=0 \\ \phi_{NB}\in(40^oN,89.5^oN). \end{cases} \tag{7.11}$$

We will investigate the properties of the system matrix A arising from the discretisation of this problem. In particular we will prove the positive-definiteness of A using some of the techniques from Section 7.2.2. We then move on to prove the convergence properties of the preconditioned methods and discussion on how the preconditioners we have introduced may be adapted for use in this nine-point case. We then consider some numerical experiments demonstrating the performances of the preconditioned methods in this nine-point case.

7.3.1 Properties of A

The discretisation stencil (7.2) in the constant depth case is given by

$$
\begin{aligned}
&P = \tfrac{1}{2}\left(\frac{\delta\phi}{\delta\lambda}\left\{\frac{1}{cos\phi_{j+\frac{1}{2}}} + \frac{1}{cos\phi_{j-\frac{1}{2}}}\right\}\right.\\
&\left.+\frac{\delta\lambda(cos\phi_{j+\frac{1}{2}} + cos\phi_{j-\frac{1}{2}})}{\delta\phi}\right) + \beta a^2 cos\phi_j \delta\lambda\delta\phi\\
&W = \tfrac{1}{4}\left(-\frac{\delta\phi}{\delta\lambda}\left\{\frac{1}{cos\phi_{j+\frac{1}{2}}} + \frac{1}{cos\phi_{j-\frac{1}{2}}}\right\} + \frac{\delta\lambda(cos\phi_{j+\frac{1}{2}} + cos\phi_{j-\frac{1}{2}})}{\delta\phi}\right)\\
&E = \tfrac{1}{4}\left(-\frac{\delta\phi}{\delta\lambda}\left\{\frac{1}{cos\phi_{j+\frac{1}{2}}} + \frac{1}{cos\phi_{j-\frac{1}{2}}}\right\} + \frac{\delta\lambda(cos\phi_{j+\frac{1}{2}} + cos\phi_{j-\frac{1}{2}})}{\delta\phi}\right)\\
&N = \tfrac{1}{2}\left(\frac{\delta\phi}{cos\phi_{j+\frac{1}{2}}\delta\lambda} - \frac{cos\phi_{j+\frac{1}{2}}\delta\lambda}{\delta\phi}\right)\\
&S = \tfrac{1}{2}\left(\frac{\delta\phi}{cos\phi_{j-\frac{1}{2}}\delta\lambda} - \frac{cos\phi_{j-\frac{1}{2}}\delta\lambda}{\delta\phi}\right)\\
&SW = \tfrac{1}{4}\left(-\frac{\delta\phi}{cos\phi_{j-\frac{1}{2}}\delta\lambda} - \frac{cos\phi_{j-\frac{1}{2}}\delta\lambda}{\delta\phi}\right)\\
&SE = \tfrac{1}{4}\left(-\frac{\delta\phi}{cos\phi_{j-\frac{1}{2}}\delta\lambda} - \frac{cos\phi_{j-\frac{1}{2}}\delta\lambda}{\delta\phi}\right)\\
&NW = \tfrac{1}{4}\left(-\frac{\delta\phi}{cos\phi_{j+\frac{1}{2}}\delta\lambda} - \frac{cos\phi_{j+\frac{1}{2}}\delta\lambda}{\delta\phi}\right)\\
&NE = \tfrac{1}{4}\left(-\frac{\delta\phi}{cos\phi_{j+\frac{1}{2}}\delta\lambda} - \frac{cos\phi_{j+\frac{1}{2}}\delta\lambda}{\delta\phi}\right).
\end{aligned}
\tag{7.12}
$$

This discretisation stencil still generates a matrix equation of the form

$$A\mathbf{U} = \mathbf{b}.$$

In this nine-point case we have

$$
A = \left(\begin{array}{cccccc}
D_1 & C_1 & & & & \\
B_2 & D_2 & C_2 & & & \\
 & B_3 & D_3 & C_3 & & \\
 & & \ddots & \ddots & \ddots & \\
 & & & & D_{n_\phi - 1} & C_{n_\phi - 1} \\
 & & & & B_{n_\phi} & D_{n_\phi}
\end{array}\right),
$$

where

$$D_j = tridiag\left[\begin{array}{c} \frac{1}{4}\left(-\frac{\delta\phi}{\delta\lambda}\left\{\frac{1}{cos\phi_{j+\frac{1}{2}}}+\frac{1}{cos\phi_{j-\frac{1}{2}}}\right\}+\frac{\delta\lambda(cos\phi_{j+\frac{1}{2}}+cos\phi_{j-\frac{1}{2}})}{\delta\phi}\right), \\ \frac{1}{2}\left(\frac{\delta\phi}{\delta\lambda}\left\{\frac{1}{cos\phi_{j+\frac{1}{2}}}+\frac{1}{cos\phi_{j-\frac{1}{2}}}\right\}+\frac{\delta\lambda(cos\phi_{j+\frac{1}{2}}+cos\phi_{j-\frac{1}{2}})}{\delta\phi}\right)+kcos\phi_j\delta\lambda\delta\phi, \\ \frac{1}{4}\left(-\frac{\delta\phi}{\delta\lambda}\left\{\frac{1}{cos\phi_{j+\frac{1}{2}}}+\frac{1}{cos\phi_{j-\frac{1}{2}}}\right\}+\frac{\delta\lambda(cos\phi_{j+\frac{1}{2}}+cos\phi_{j-\frac{1}{2}})}{\delta\phi}\right), \end{array}\right] \tag{7.13}$$

$$B_j = tridiag\left[\begin{array}{c} \frac{1}{4}\left(-\frac{\delta\phi}{cos\phi_{j-\frac{1}{2}}\delta\lambda}-\frac{cos\phi_{j-\frac{1}{2}}\delta\lambda}{\delta\phi}\right), \\ \frac{1}{2}\left(\frac{\delta\phi}{cos\phi_{j-\frac{1}{2}}\delta\lambda}-\frac{cos\phi_{j-\frac{1}{2}}\delta\lambda}{\delta\phi}\right), \\ \frac{1}{4}\left(-\frac{\delta\phi}{cos\phi_{j-\frac{1}{2}}\delta\lambda}-\frac{cos\phi_{j-\frac{1}{2}}\delta\lambda}{\delta\phi}\right) \end{array}\right], \tag{7.14}$$

$$C_j = tridiag\left[\begin{array}{c} \frac{1}{4}\left(-\frac{\delta\phi}{cos\phi_{j+\frac{1}{2}}\delta\lambda}-\frac{cos\phi_{j+\frac{1}{2}}\delta\lambda}{\delta\phi}\right), \\ \frac{1}{2}\left(\frac{\delta\phi}{cos\phi_{j+\frac{1}{2}}\delta\lambda}-\frac{cos\phi_{j+\frac{1}{2}}\delta\lambda}{\delta\phi}\right), \\ \frac{1}{4}\left(-\frac{\delta\phi}{cos\phi_{j+\frac{1}{2}}\delta\lambda}-\frac{cos\phi_{j+\frac{1}{2}}\delta\lambda}{\delta\phi}\right) \end{array}\right]. \tag{7.15}$$

From the definitions (7.13) to (7.15), it is straightforward to observe that each block D_j, B_j and C_j is symmetric and this, combined with the fact that

$$B_j = C_{j-1},$$

is enough for us to conclude that the matrix A is symmetric. We will consider other properties of the system matrix A in the next section.

7.3.2 Theoretical convergence analysis

In this section we will investigate the properties of the preconditioned matrices required to demonstrate the convergence of the preconditioned methods. Firstly we note that we are still assuming that we have $\delta\lambda, \delta\phi > 0$ and $cos\phi \in (0, 1)$. Therefore we may deduce in an analogous manner to Section 4.4.1 that the matrix entries we assume to be non-zero cannot become zero anywhere in the domain. From this we may deduce that the connected graph of the system matrix A is strongly connected and therefore, via Theorem 3.1, that our matrix is irreducible.

We now wish to show that A is nonsingular and has only strictly positive eigenvalues. We may then deduce that is is positive definite and that the unpreconditioned conjugate gradient method will converge. Observe that our system matrix A may be written in partitioned form (3.7). In order to conclude that A is nonsingular and possesses positive, real eigenvalues we require, via Theorems 3.8, 3.9 and 3.10, to show that A is block strictly(or irreducibly) diagonally dominant i.e. we require

$$\left(\| A_{j,j}^{-1} \|\right)^{-1} \geq \sum_{l=1, l\neq j}^{n_\phi} \| A_{j,l} \| \quad \forall 1 \leq j \leq n_\phi,$$

with strict inequality for at least one i. Therefore in order to demonstrate the strict block diagonal dominance of A we need to show that

$$\begin{aligned} &\left(\| D_j^{-1} \|\right)^{-1} \geq \| B_j \| + \| C_j \| \\ &\Longrightarrow 1 \geq \| D_j^{-1} \| \left(\| B_j \| + \| C_j \|\right), \end{aligned} \tag{7.16}$$

in some norm with strict inequality for at least one j. We use the L_2-norm with

$$\| A \|_2 = max \mid \mu_i \left(A^T A\right) \mid^{\frac{1}{2}}, \tag{7.17}$$

where the μ_i are eigenvalues. We assume that we have $\delta\lambda = \delta\phi = h$ and $k > 0$. From this it follows that the D_j's are symmetric and strictly diagonally dominant with $d_{ii}^j > 0$, $d_{ik}^j \leq 0$ where $D_j = \left\{d_{ik}^j\right\}$. Hence the D_j's are positive definite.

In order to get bounds on the norms we use the Gerschgorin Circle Theorems. The eigenvalues, μ of D_j satisfy

$$\mid \mu - R_j \mid \leq \frac{1}{2}\left(\frac{1}{cos\phi_{j+\frac{1}{2}}} + \frac{1}{cos\phi_{j-\frac{1}{2}}} - cos\phi_{j+\frac{1}{2}} - cos\phi_{j-\frac{1}{2}}\right), \tag{7.18}$$

where

$$R_j = \frac{1}{2}\left(\frac{1}{cos\phi_{j+\frac{1}{2}}} + \frac{1}{cos\phi_{j-\frac{1}{2}}} + cos\phi_{j+\frac{1}{2}} + cos\phi_{j-\frac{1}{2}} + kcos\phi_j h^2\right). \qquad (7.19)$$

Therefore

$$\begin{aligned} &-\tfrac{1}{2}\left(\tfrac{1}{cos\phi_{j+\frac{1}{2}}} + \tfrac{1}{cos\phi_{j-\frac{1}{2}}} - cos\phi_{j+\frac{1}{2}} - cos\phi_{j-\frac{1}{2}}\right) + R_j \leq \mu \\ &\leq R_j + \tfrac{1}{2}\left(\tfrac{1}{cos\phi_{j+\frac{1}{2}}} + \tfrac{1}{cos\phi_{j-\frac{1}{2}}} - cos\phi_{j+\frac{1}{2}} - cos\phi_{j-\frac{1}{2}}\right). \end{aligned} \qquad (7.20)$$

Hence the smallest eigenvalue of D_j satisfies

$$\mu_{min} \geq cos\phi_{j+\frac{1}{2}} + cos\phi_{j-\frac{1}{2}} + kcos\phi_j h^2 = \delta_j. \qquad (7.21)$$

Since D_j is symmetric the eigenvalues of D_j are real allowing us to obtain these bounds. Also

$$\| D_j \|_2 = \mu_{max}, \qquad (7.22)$$

and

$$\begin{aligned} &\| D_j^{-1} \|_2 = \tfrac{1}{\mu_{min}} \\ &\Longrightarrow \| D_j^{-1} \|_2^{-1} = \mu_{min} \\ &\Longrightarrow \| D_j^{-1} \|_2^{-1} \geq \delta_j. \end{aligned} \qquad (7.23)$$

We also have

$$\| B_j \|_2 + \| C_j \|_2 = \frac{\delta\phi}{cos\phi_{j+\frac{1}{2}}\delta\lambda} + \frac{\delta\phi}{cos\phi_{j-\frac{1}{2}}\delta\lambda}. \qquad (7.24)$$

This is generally larger than $\| D_j^{-1} \|_2$. Hence it is not possible to show block diagonal dominance for this problem using the Block Gerschgorin technique.

In order to prove the positive-definiteness of our nine-point operator we need to show that $\mathbf{U}^T A\mathbf{U} > 0$ for any $\mathbf{U} \neq 0$. We firstly take our nine-point finite-difference operator and simplify by multiplying through by $\frac{4}{\delta\lambda\delta\phi}$. We then consider the product $A\mathbf{U}$. A general line, associated with the point ij, of this product is given by

$$\begin{aligned}
&\tfrac{1}{cos\phi_{j+\frac{1}{2}}\delta\lambda^2}\left[2U_{ij}-U_{i+1j}-U_{i-1j}+2U_{ij+1}-U_{i-1j+1}-U_{i+1j+1}\right]\\
&+\tfrac{1}{cos\phi_{j-\frac{1}{2}}\delta\lambda^2}\left[2U_{ij}-U_{i+1j}-U_{i-1j}+2U_{ij-1}-U_{i-1j-1}-U_{i+1j-1}\right]\\
&+\tfrac{cos\phi_{j+\frac{1}{2}}}{\delta\phi^2}\left[2U_{ij}+U_{i+1j}+U_{i-1j}-2U_{ij+1}-U_{i+1j+1}-U_{i-1j+1}\right]\\
&+\tfrac{cos\phi_{j-\frac{1}{2}}}{\delta\phi^2}\left[2U_{ij}+U_{i+1j}+U_{i-1j}-2U_{ij-1}-U_{i+1j-1}-U_{i-1j-1}\right]\\
&+4kcos\phi_jU_{ij}.
\end{aligned}\tag{7.25}$$

We then form the product $\mathbf{U}^TA\mathbf{U}$ and sum over the interior points of the domain to give

$$\begin{aligned}
&\textstyle\sum_{j=1}^{n_\phi-1}\sum_{i=1}^{n_\lambda-1}\left(\tfrac{1}{cos\phi_{j+\frac{1}{2}}\delta\lambda^2}\left[U_{ij}(U_{ij}-U_{i+1j}+U_{ij+1}-U_{i+1j+1})\right]\right)\\
&\textstyle+\sum_{j=1}^{n_\phi-1}\sum_{i=2}^{n_\lambda}\left(\tfrac{1}{cos\phi_{j+\frac{1}{2}}\delta\lambda^2}\left[U_{ij}(U_{ij}-U_{i-1j}+U_{ij+1}-U_{i-1j+1})\right]\right)\\
&\textstyle+\sum_{j=2}^{n_\phi}\sum_{i=1}^{n_\lambda-1}\left(\tfrac{1}{cos\phi_{j-\frac{1}{2}}\delta\lambda^2}\left[U_{ij}(U_{ij}-U_{i+1j}+U_{ij-1}-U_{i+1j-1})\right]\right)\\
&\textstyle+\sum_{j=2}^{n_\phi}\sum_{i=2}^{n_\lambda}\left(\tfrac{1}{cos\phi_{j-\frac{1}{2}}\delta\lambda^2}\left[U_{ij}(U_{ij}-U_{i-1j}+U_{ij-1}-U_{i-1j-1})\right]\right)\\
&\textstyle+\sum_{j=1}^{n_\phi-1}\sum_{i=1}^{n_\lambda-1}\left(\tfrac{cos\phi_{j+\frac{1}{2}}}{\delta\phi^2}\left[U_{ij}(U_{ij}+U_{i+1j}-U_{ij+1}-U_{i+1j+1})\right]\right)\\
&\textstyle+\sum_{j=1}^{n_\phi-1}\sum_{i=2}^{n_\lambda}\left(\tfrac{cos\phi_{j+\frac{1}{2}}}{\delta\phi^2}\left[U_{ij}(U_{ij}+U_{i-1j}-U_{ij+1}-U_{i-1j+1})\right]\right)\\
&\textstyle+\sum_{j=2}^{n_\phi}\sum_{i=1}^{n_\lambda-1}\left(\tfrac{cos\phi_{j-\frac{1}{2}}}{\delta\phi^2}\left[U_{ij}(U_{ij}+U_{i+1j}-U_{ij-1}-U_{i+1j-1})\right]\right)\\
&\textstyle+\sum_{j=2}^{n_\phi}\sum_{i=2}^{n_\lambda}\left(\tfrac{cos\phi_{j-\frac{1}{2}}}{\delta\phi^2}\left[U_{ij}(U_{ij}+U_{i-1j}-U_{ij-1}-U_{i-1j-1})\right]\right)\\
&\textstyle+\sum_{j=1}^{n_\phi}\sum_{i=1}^{n_\lambda}\left(4kcos\phi_jU_{ij}^2\right),
\end{aligned}$$

$$
\begin{aligned}
&= \textstyle\sum_{j=1}^{n_\phi-1}\sum_{i=1}^{n_\lambda-1}\Big(\frac{1}{cos\phi_{j+\frac{1}{2}}\delta\lambda^2}\left[U_{ij}(U_{ij}-U_{i+1j}+U_{ij+1}-U_{i+1j+1})\right.\\
&\left.+U_{i+1j}(U_{i+1j}-U_{ij}+U_{i+1j+1}-U_{ij+1})\right]\\
&+\tfrac{1}{cos\phi_{j+\frac{1}{2}}\delta\lambda^2}\left[U_{ij+1}(U_{ij+1}-U_{i+1j+1}+U_{ij}-U_{i+1j})\right.\\
&\left.+U_{i+1j+1}(U_{i+1j+1}-U_{ij+1}+U_{i+1j}-U_{ij})\right]\\
&+\tfrac{cos\phi_{j+\frac{1}{2}}}{\delta\phi^2}\left[U_{ij}(U_{ij}+U_{i+1j}-U_{ij+1}-U_{i+1j+1})\right.\\
&\left.+U_{i+1j}(U_{i+1j}+U_{ij}-U_{i+1j+1}-U_{ij+1})\right]\\
&+\tfrac{cos\phi_{j+\frac{1}{2}}}{\delta\phi^2}\left[U_{ij+1}(U_{ij+1}+U_{i+1j+1}-U_{ij}-U_{i+1j})\right.\\
&\left.+U_{i+1j+1}(U_{i+1j+1}+U_{ij+1}-U_{i+1j}-U_{ij})\right]\\
&+4kcos\phi_jU_{ij}^2\Big)\,,
\end{aligned}
$$

$$
\begin{aligned}
&= \textstyle\sum_{j=1}^{n_\phi-1}\sum_{i=1}^{n_\lambda-1}\frac{1}{cos\phi_{j+\frac{1}{2}}\delta\lambda^2}\left[U_{ij}-U_{i+1j}+U_{ij+1}-U_{i+1j+1}\right]^2\\
&+\textstyle\sum_{j=1}^{n_\phi-1}\sum_{i=1}^{n_\lambda-1}\frac{cos\phi_{j+\frac{1}{2}}}{\delta\phi^2}\left[U_{ij}+U_{i+1j}-U_{ij+1}-U_{i+1j+1}\right]^2\\
&+\textstyle\sum_{j=1}^{n_\phi}\sum_{i=1}^{n_\lambda}4kcos\phi_jU_{ij}^2.
\end{aligned}
\tag{7.26}
$$

The expression clearly consists of strictly positive terms and hence the expression (7.26) is strictly positive for all U, i and j. Therefore the matrix A is positive-definite. As it is also symmetric it is a Stieltjes matrix and hence by Theorem 3.4 is an M-matrix.

The matrix A is a block-tridiagonal matrix. Therefore by Theorems 3.15 and 3.13 it is consistently ordered and hence has Property A. Therefore the diagonal and block diagonal preconditioned methods converge and the eigenvalues of the preconditioned matrices occur in ± pairs. Finally we note that the diagonal preconditioner will provide the 'optimum' diagonal scaling, with regards to conditioning, when applying the Binormalization scaling.

Another important point to consider is the form of the ADI preconditioner with this nine-point discretisation. ADI methods have been used in conjunction with nine-point operators before [20], but that was for the standard nine-point discretisation operator described in Forsythe and Wasow [33]. The matrix properties (Stieltjes matrices) required for the matrices in the ADI splitting would not be satisfied in this case if we were to attempt the same 'directional' scheme. However, as we are considering ADI as a preconditioner, it is possible for us to use only those parts of the nine-point operator which have the required properties. In the nine-point case we are considering we use

$$H_\Upsilon = \begin{pmatrix} D_1^H & & & \\ & D_2^H & & \\ & & \ddots & \\ & & & \\ & & & D_{n_\phi}^H \end{pmatrix},$$

where

$$D_j^H = tridiag \begin{bmatrix} \frac{1}{4}\left(-\frac{\delta\phi}{\delta\lambda}\left\{\frac{1}{cos\phi_{j+\frac{1}{2}}} + \frac{1}{cos\phi_{j-\frac{1}{2}}}\right\}\right), \\ \frac{1}{2}\left(\frac{\delta\phi}{\delta\lambda}\left\{\frac{1}{cos\phi_{j+\frac{1}{2}}} + \frac{1}{cos\phi_{j-\frac{1}{2}}}\right\}\right) + \frac{k}{2}cos\phi_j\delta\lambda\delta\phi, \\ \frac{1}{4}\left(-\frac{\delta\phi}{\delta\lambda}\left\{\frac{1}{cos\phi_{j+\frac{1}{2}}} + \frac{1}{cos\phi_{j-\frac{1}{2}}}\right\}\right) \end{bmatrix}, \qquad (7.27)$$

and

$$V_\Upsilon = \begin{pmatrix} D_1^V & DC_1 & & & & \\ DB_2 & D_2^V & DC_2 & & & \\ & DB_3 & D_3^V & DC_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & & & \\ & & & & D_{n_\phi-1}^V & DC_{n_\phi-1} \\ & & & & DB_{n_\phi} & D_{n_\phi}^V \end{pmatrix},$$

where

$$D_j^V = \frac{1}{2}\left(\frac{\delta\lambda(cos\phi_{j+\frac{1}{2}} + cos\phi_{j-\frac{1}{2}})}{\delta\phi}\right) + \frac{k}{2}cos\phi_j\delta\lambda\delta\phi, \tag{7.28}$$

$$DB_j = -\frac{1}{4}\left(\frac{\delta\lambda cos\phi_{j-\frac{1}{2}}}{\delta\phi}\right), \tag{7.29}$$

$$DC_j = -\frac{1}{4}\left(\frac{\delta\lambda cos\phi_{j+\frac{1}{2}}}{\delta\phi}\right), \tag{7.30}$$

With these choices H_Υ and V_Υ can be shown to be Stieltjes matrices for $k > 0$ with Dirichlet or periodic boundary conditions in the λ direction and for $k > 0$ with Dirichlet conditions. For those cases we may deduce that the ADI preconditioned method is convergent for $\Upsilon > 0$.

7.3.3 Numerical experiments

In this section we present results from some numerical experiments of the problem 7.11. Again we investigate the effects of extending the northern boundary of the domain towards the pole and investigate how well the proposed preconditioners address the polar convergence issue in this nine-point case. Discrete stepsizes of 2^o, 1^o and $\frac{1}{2}^o$ are again used in both horizontal directions.

Tables 7.1, and 7.2 give the spectral radii of A and the spectral radii iteration matrices, G, with the various preconditioners, for the case where $k = 0.01$, $h = 1^o$, $\phi_{NB} = 88^o$. The full results for the spectral radii of the preconditioned iteration matrices are shown in Appendix B in Tables B.24 to B.27. Again we note the increasing of $\rho(G)$ with decreasing stepsizes and as ϕ_{NB} is moved closer to the pole. The spectral radii of the diagonal, Binormalization and block diagonal preconditioned methods appear to have a limit on their size as the boundary is moved closer to the pole, as in the Limited Area case from Chapter 5. The spectral radii of the G matrices are

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^o N	4.946	4.791	4.548
70^o N	10.381	9.698	8.708
88^o N	68.592	53.573	38.335
89^o N	107.313	77.234	NA
89.5^o N	154.752	NA	NA

Table 7.1: Spectral Radii of system matrix A, $k = 0.01$, nine-point

	ϕ_{NB}		
Preconditioner	40^oN	70^oN	88^oN
Diagonal	0.9936	0.9971	0.9979
Block diagonal	0.9900	0.9955	0.9966
ADI	0.8788	0.9065	0.9515
Binormalization	0.9951	0.9977	0.983

Table 7.2: Spectral Radii of G_D, $k = 0.01$, nine-point

all less than 1 guaranteeing convergence of the numerical method. The value for G_{ADI} (the ADI preconditioner) is the smallest followed by the Block, diagonal and Binormalization preconditioners respectively, as in Chapter 5.

Table 7.3 gives values for the conditioning of the problem with the preconditioners considered in this section, for the case where $k = 0.01$, $h = 1^o$, $\phi_{NB} = 88^o$. We observe that the smallest values are given by the ADI preconditioned system followed by Block, Diagonal and Binormalization preconditioners respectively. This pattern is confirmed in the full results that are shown in Appendix B in Tables B.28 to B.37. We further note that Theorem 3.18 is still satisfied for all cases considered ($p = 9$ for this problem).

Preconditioner	$K_\infty(P^{-1}A)$
None	1.093×10^4
Diagonal	1.220×10^3
Block	894.938
ADI	89.909
BIN	1.483×10^3

Table 7.3: ∞ norm condition numbers for $h = 1^o$, 88^o, $k = 0.01$ case, nine-point

Overall therefore we would expect ADI preconditioning to yield the fastest convergence followed by Block, Diagonal and Binormalization preconditioning respectively.

The minimum eigenvalues of the system matrices, A were also checked. For all of the stepsizes and ϕ_{NB} values considered the eigenvalues of A were all strictly positive. As an additional check we tested each of the system matrices, A, by calculating their Cholesky factorizations ([38]) using the inbuilt MATLAB function, CHOL. This function only works if the matrix being factorized is symmetric positive-definite. In all cases considered the Cholesky Factorization existed proving, experimentally, that our system matrices are indeed symmetric positive-definite.

A similar pattern to previous cases is also noted in the form of the leading eigenvectors of the iteration matrices, G. Strong polar signals are observed in the leading eigenvectors of the Diagonal and Binormalization preconditioners (Figures B.47 to B.50). Significantly lesser polar signals are observed in the leading eigenvectors of the Block preconditioner as shown in Figures B.35 to B.38. In conjunction with the lower associated eigenvalues, this leads us to again expect the Block preconditioner to provide faster convergence, and address the pole problem more effectively than the Binormalization and Diagonal preconditioners. The leading four eigenvectors of G_{ADI}, for $\phi_{NB} = 88^o$, as shown in figures B.43 to B.46 do have strong signals in the polar regions. However they are associated with considerably smaller eigenvalues than diagonal or block preconditioning. Whilst, from this, we might not expect the pole problem to be addressed very well, the convergence overall ought to be much faster with ADI than the other preconditioners.

Tables 7.4 and 7.5 show the number of iterations to convergence, and the associated CPU times, for our preconditioned methods using the relative residual error normalised by **b** as a stopping criterion. We again used a right hand source function $\gamma(\lambda, \phi)$ fixed to yield a sine function general solution of

$$\begin{aligned} &U(\lambda, \phi) = sin(3\lambda)sin(d[\phi - 10]) \\ &d = \tfrac{90}{\phi_{NB}-10} \end{aligned} \tag{7.31}$$

which is consistent with the chosen boundary conditions. A constant 'initial guess' of $U(i, j) = 1.5$ was again taken to start the iterative process. The results are as expected apart from ADI, which appears to perform very badly, despite the theoretical calculations that showed it ought to be clearly the most efficient preconditioner. The results in Tables appear to show that it is the choice of stopping criterion that is vital here. The ADI preconditioner performs considerably better when the relative residual error normalised by the initial residual is used as a stopping criterion as in Table. ADI performs better still when the absolute error is used as the stopping criterion. Of course the absolute error typically cannot be used in practice as it requires explicit knowledge of the solution we are attempting to find. However it, and the results for the relative errors, indicates that the choice of stopping criterion can have a serious impact on the efficiency of certain preconditioners, and hence on the amount of computing time required.

	40^o			70^0			88^o			89.0^o		89.5^o
Prec.	$\frac{1}{2}^o$	1^o	2^o	$\frac{1}{2}^o$	1^o	2^o	$\frac{1}{2}^o$	1^o	2^o	$\frac{1}{2}^o$	1^o	$\frac{1}{2}^o$
Diag	124	60	27	239	113	50	330	160	72	342	170	365
Blo	114	54	27	145	71	35	154	76	43	154	76	154
ADI	176	109	34	183	119	51	230	134	62	237	139	241
Bin	126	61	30	212	103	51	288	140	71	295	148	303

Table 7.4: Number of iterations to convergence, sine function general solution, nine point

	40^o			70^0			88^o			89.0^o		89.5^o
Prec.	$\frac{1}{2}^o$	1^o	2^o	$\frac{1}{2}^o$	1^o	2^o	$\frac{1}{2}^o$	1^o	2^o	$\frac{1}{2}^o$	1^o	$\frac{1}{2}^o$
Diag	6.0	0.7	0.3	33.4	2.6	0.4	75.6	5.5	0.6	81.5	6.1	89.2
Blo	10.9	1.3	0.4	41.9	3.4	0.6	66.4	4.8	0.7	72.3	5.6	76.4
ADI	34.9	5.2	1.2	101.0	6.0	1.4	204.8	9.8	1.7	214.5	9.5	218.0
Bin	6.1	0.7	0.4	29.6	2.4	0.4	68.3	4.9	0.7	74.9	5.6	80.3

Table 7.5: CPU times, sine function general solution,nine point

	40^o			70^0			88^o			89^o		89.5^o
Prec.	$\frac{1}{2}^o$	1^o	2^o	$\frac{1}{2}^o$	1^o	2^o	$\frac{1}{2}^o$	1^o	2^o	$\frac{1}{2}^o$	1^o	$\frac{1}{2}^o$
Diag	96	50	25	162	81	42	176	94	50	177	94	178
Block	89	46	23	106	55	31	112	62	35	113	62	113
ADI	45	35	25	52	38	27	56	40	29	57	40	57

Table 7.6: Number of iterations to convergence tolerance $\frac{||\mathbf{r}^k||_\infty}{||\mathbf{r}^0||_\infty} < 10^{-5}$, sine function general solution

	40^o			70^0			88^o			89^o		89.5^o
Prec.	$\frac{1}{2}^o$	1^o	2^o	$\frac{1}{2}^o$	1^o	2^o	$\frac{1}{2}^o$	1^o	2^o	$\frac{1}{2}^o$	1^o	$\frac{1}{2}^o$
Diag	62	30	22	114	55	30	133	65	33	134	65	134
Block	54	27	14	70	35	18	70	35	18	70	35	70
ADI	14	10	6	17	12	8	23	15	10	25	16	27

Table 7.7: Number of iterations to convergence tolerance $|| \mathbf{U} - \mathbf{U}^m ||_\infty < 10^{-2}$, sine function general solution

7.4 Summary

In this chapter we further extended our spherical domain model by investigating the use of a special nine-point discretisation operator, analogous to that used for solving the free surface problem. We discussed the exact form of the discretisation operator and proved its consistency to the model problem using truncation error analysis. We also expanded on the use of 'Implicit' boundary conditions for islands referred to by Dukowicz [24]. We proved the positive-definiteness of the problem using a finite difference analogue. We then considered a reduced problem with a constant depth profile across the domain and again demonstrated the symmetric positive-definiteness of the problem. This was also shown experimentally using numerical experiments revisiting the Limited Area problem of previous chapters. It was found that, in all cases considered, the eigenvalues of A were strictly positive and that a Cholesky factorization of A existed. Hence the matrices, A, used in the experiments were positive-definite.

The form of the eigenvectors of the iteration matrices of the various preconditioned methods were shown to be similar to that obtained by using the standard five point discretisation scheme. The same was generally true for the spectral radii of the iteration matrices, G, and the condition numbers of the preconditioned system matrices. One small difference was that the conditioning values for the Binormalization preconditioning were slightly better than those of the diagonal preconditioner. The eigenvalues of G_D and G_{Block}, for the diagonal and Block preconditioners respectively, again came in ± pairs as predicted.

Numerical experiments were run to test practically the relative merits of the preconditioners. Similar results to previous chapters were found with Block preconditioning generally outperforming Diagonal and Binormalization preconditioning. The main difference was with the ADI preconditioner, which required careful consideration in its implementation with the nine-point scheme. An ADI scheme was proposed which was found to perform comparatively badly in the numerical experiments. This was despite it being theoretically much the best preconditioner, as demonstrated by the relatively small values for the spectral radii of the preconditioned iteration matrix, G_{ADI}, and the condition numbers of the preconditioned system. The problem was shown to be caused by the choice of stopping criterion (residual error normalised by $\mathbf{b}$). When the residual error normalised by the initial residual, and particularly the absolute error, were used in the stopping criterion, the benefits of ADI became much clearer : it was by far the more efficient preconditioner. Therefore the ADI preconditioned method is providing the most accurate solution in a given amount of time; the convergence criteria being used (relative residual error normalised by $\mathbf{b}$) was not recognising this.

Chapter 8

Conclusions and further work

8.1 Conclusions

The aims of this study were to highlight the problem of slow convergence in polar regions of the elliptic problems solved in free-surface ocean models using preconditioned iterative methods, to explain how the mesh anisotropy of the latitude-longitude co-ordinate causes the polar convergence problem, and to address the convergence issue by suggesting preconditioners which could be used to reduce the problem and speed up the overall convergence of the preconditioned iterative methods. This would improve the efficiency of the overall ocean models.

The anisotropic elliptic operators which are the focus of the work in this thesis were reviewed in Chapter 2 in the context of ocean modelling. We described the Bryan-Cox-Semtner (BCS) model used widely today by ocean groups worldwide. We focussed on the two formulations of the BCS model most commonly used : rigid-lid and implicit free surface. The forms of the elliptic equations which arise with these formulations were summarised as were the computational advantages and disadvantages of their implementation.

The numerical methods used to iteratively solve the corresponding discrete equations were reviewed in Chapter 3. For the rigid-lid formulation the Chebyshev semi-iterative method is used. With implicit free surface a conjugate gradient method with diagonal (jacobi) preconditioning is used. We gave introductory theory for the use of some alternative preconditioners : block Jacobi, Alternating-Direction-Implicit, and Binormalization scaling.

Chapter 4 introduced the spherical model used to approximate the anisotropic elliptic problems encountered in the barotropic ocean solvers. We initially consider the constant depth problem in both limited area and periodic domain cases. We confirmed the validity of our discretisation scheme using truncation error analysis and derived the continuous and discrete eigenvalues and eigenvectors of the spherical Laplacian and hence those of the Helmholtz problem. The convergence of our preconditioned Conjugate Gradient method for this problem, using the proposed preconditioners, was checked. The preconditioners were assessed, with respect to their likely effect on speeds of convergence and the anisotropy, using Gerschgorin analysis. It was found that an ADI preconditioner was likely to be slightly better than using a Block diagonal preconditioner. Both were predicted to be much better than the diagonal preconditioner. This was largely confirmed by the numerical experiments of Chapter 5. The ADI preconditioner with spatially varying parameter did not perform very well in the experiments. This is likely to have been caused by the sensitivity of the preconditioner to the parameter values that are used, combined with the fact that the values were calculated using the crude Gerschgorin estimates.

It was noted that the condition numbers of the preconditioned systems, and the associated spectral radii of the iteration matrices G, changed very little as the anisotropy was increased by moving the northern boundary of the domain closer to the pole, when using the diagonal and Block diagonal preconditioners in the Limited Area problem (And to a lesser extent in the Periodic domain problem). Also it was observed that the leading eigenvectors of the iteration matrices, G, did not have strong polar signals. However it was also noted that the 'nearly' leading eigenvalues of the iteration matrices, G, became larger as the northern boundary of the domain was moved closer to the pole, clustering around the lead eigenvalue and that the associated 'nearly' leading eigenvectors did have strong polar signals. It was deduced that this in part leads to the slower convergence of the iterations in polar regions. It was also noted that the non-zero eigenvalues of the iteration matrices for the Block and Diagonal preconditioners occurred in $\pm$.

The leading four eigenvectors of the Block preconditioned (and ADI for the periodic case) iteration matrices were shown to display smaller polar signals than the diagonal preconditioned iteration matrices. Also Block and ADI preconditioning was shown to damp the spectrum of Fourier error modes more evenly than diagonal preconditioning. Despite this the convergence histories of all three preconditioners showed that the residual errors in the polar regions were the last to converge.

Chapter 6 extended the basic spherical model introduced in the previous to include problems which include a varying depth function, H, within the elliptic operators. We firstly examined a case which was analogous with the free surface formulation where the elliptic operator is of the form $-\nabla \cdot (H\nabla)U + kU$. We again showed that ADI ought to yield the fastest convergence, followed by Block and Diagonal, by examination of the spectral radii of the preconditioned iteration matrices, G, and the conditioning of the preconditioned systems. However it was noted that whilst the leading eigenvectors of the diagonal and block diagonal preconditioned methods were not very sensitive to the height profile, the ADI preconditioned iteration matrix, G_{ADI}, was. We then moved on to consider problems where the operator is of Poisson type $-\nabla \cdot (\frac{1}{H}\nabla)$ similar to the rigid-lid formulation. We showed that by considering the conditioning of the system matrix as well as the size of the spectral radii of the diagonal preconditioned iteration matrix, G_D ,that the problem is sensitive to the variations in H. We also highlighted the importance of using an accurate value for the spectral radii of G_D. Using an accurate value causes the convergence of all error modes to be approximately equal, when using the Chebyshev semi-iterative method, in the constant depth case. The convergence of the modes varied a lot more with the use of less accurate choices of the spectral radii, $\rho(G_D)$. The slight discrepancy in the Continental shelf case was attributed to the fact that Fourier modes were less physically representative of the errors involved in this case.

In Chapter 7 we further extended our spherical domain model by investigating the use of a special nine-point discretisation operator, analogous to that used for solving the free surface problem. We discussed the exact form of the discretisation operator and proved its consistency to the model problem using truncation error analysis. We also expanded on the use of 'Implicit' boundary conditions for islands referred to by Dukowicz [24]. We proved the positive-definiteness of the problem using a finite difference analogue. We then considered a reduced problem with a constant depth profile across the domain and again demonstrated the symmetric positive-definiteness of the problem. This was also shown experimentally using numerical experiments revisiting the Limited Area problem of previous chapters. It was found that, in all cases considered, the eigenvalues of A were strictly positive and that a Cholesky factorization of A existed.

The form of the eigenvectors of the iteration matrices of the various preconditioned methods were shown to be similar to that obtained by using the standard five point discretisation scheme. The same was generally true for the spectral radii of the iteration matrices, G, and the condition numbers of the preconditioned system matrices. The eigenvalues of G_D and G_{Block}, for the diagonal and Block preconditioners respectively, again came in ± pairs.

Numerical experiments were run to test practically the relative merits of the preconditioners. Similar results to previous chapters were found with Block preconditioning generally outperforming Diagonal and Binormalization preconditioning. The main difference was with the ADI preconditioner, which required careful consideration in its implementation with the nine-point scheme. An ADI scheme was proposed which was found to perform comparatively badly in the numerical experiments. This was despite it being theoretically much the best preconditioner, as demonstrated by the relatively small values for the spectral radii of the preconditioned iteration matrix, G_{ADI}, and the condition numbers of the preconditioned system. The problem was shown to be caused by the choice of stopping criterion (residual error normalised by $\mathbf{b}$). When the residual error normalised by the initial residual, and particularly the absolute error, were used in the stopping criterion, the benefits of ADI became much clearer : it was by far the more efficient preconditioner. Therefore the ADI preconditioned method is providing the most accurate solution in a given amount of time; the convergence criteria being used were not recognising this. We conclude therefore that the choice of stopping criterion is crucial and that great care must be taken to choose a criteria that reflects the accuracy of the solution. The absolute error ensures the accuracy of the solution and is the most direct measure of accuracy that could be used here. However it relies on knowledge of the exact solution and hence is not practical to use. The relative residual error normalised by the initial residual is related to the average rate of convergence and gives an impression of the relative improvement in the solution. It does not really reflect the real accuracy of the solution. The relative residual error normalised by the right hand side function does give some measure of the accuracy of the solution but can be a poor bound (as the ADI results of Section 7.3.3. show) when the conditioning of a problem is poor.

To summarise, we have identified

- How the mesh anisotropy of the spherical elliptic operators, which are solved in ocean models, affects the convergence of the iterative methods used to solve them.

- That 'secondary' eigenmodes with strong polar signals cause the polar convergence problem.

- Various alternative preconditioners (Block, Binormalization, ADI) have been suggested for use with the PCG method with diagonal preconditioning, used in the free-surface formulation, most of which have provided some improvement of the problem.

- How the preconditioners may be adapted for the nine-point discretisation scheme used in the free-surface formulation.

- Importance of stopping criteria has been highlighted. Criteria based on the normalised residual errors are typically used in practice; we have shown that the performance of our iterative methods with certain stopping criteria may be very different to the actual accuracy (absolute errors).

A number of further alternative preconditioners that may be used are suggested in the next section along with some other possible extensions to the research.

8.2 Further work

One of the many open questions of this study is what other preconditioners could be investigated for use with our mesh anisotropic spherical domain problems? We have restricted ourselves to studying preconditioners which may all be classed as approximate sparse inverse preconditioners. The diagonal preconditioner used in the free-surface model is the simplest of this class. The sparse inverse preconditioners have the advantage of making use of the clear structure of the matrix systems, being reasonably straightforward to parallelise for use with vector processors, and being able to accommodate multiply connected domains : i.e. domains with islands in them. One possible extension of this study would be to test the preconditioners we have considered in the full MO free-surface model (with the inclusion of islands).

As we have shown that the system matrix for the nine-point operator possesses a Cholesky decomposition, it would be possible to use the Incomplete Cholesky Factorization([6], [21], [38]) as a preconditioner. The incomplete Cholesky factorization keeps the factors used in the preconditioning artificially sparse to improve storage and CPU time used. This option is discussed further in Axelsson [6], Duff and Van Der Vorst [21] and Golub and Van Loan [38].

Another option would be to use the Schur complement as a preconditioner. The use of the Schur Complement as a preconditioner for a Navier-Stokes type problem is detailed in Elman et al [28], [29].

Various other preconditioners have been used to study anisotropic problems (although mostly constant parameter cases). One example is circulant block-factorization (CBF) preconditioners. The use of CBF preconditioners with anisotropic problems was discussed in Lirkov et al [51], having been introduced for use as preconditioners with periodic elliptic problems in an earlier study [50], and for general elliptic problems in Chan and Chan [14]. CBF preconditioners are shown to combine some of the advantages of (Block) Incomplete LU/Cholesky factorization and Block-Circulant methods. More information on Circulant preconditioners may be found in Chan and Chan [14], and Lirkov et al [50], [51].

Mawson [58] demonstrated the clear applicability of multigrid methods to elliptic problems in spherical geometry. A more recent idea has been to use multigrid as a preconditioner for a Gradient type method such as CG. Useful introductions to multigrid methods may be found in Brandt [10] and Briggs [11]. The use of Multigrid as a preconditioner was considered initially by Kettler [45], and more recently by Tatebe [74]. It was confirmed by Tatebe [74] that the multigrid method satisfies the conditions required for a preconditioner with the Conjugate gradient method; that the preconditioned system matrix of the multigrid preconditioned CG method should be similar to a symmetric positive-definite matrix. It is concluded that the multigrid preconditioned CG method has the properties that the number of iterations to convergence do not increase with finer meshes, and is effective with ill-conditioned problems. The main problem with such a method would be in the resolution of islands on each grid scale. Should this be resolved then the fast convergence of Multigrid preconditioned CG, as demonstrated by Tatebe [74], would make this a strong candidate for consideration as a preconditioner for an anisotropic elliptic operator in a spherical domain ocean model.

Bibliography

[1] Abramowitz, M., Stegun, I.A., Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables, *New York : Dover press*, 1972.

[2] Adcroft, A., Campin, J.M., Heimbach, P., Hill, C., Marshall, J., The MITgcm. Online documentation, *Massachusetts Institute of Technology, USA*, 2002. Available online at mitgcm.org/sealion/online-documents/

[3] Ames, W.F., Numerical Methods for Partial Differential Equations, *Academic press*, 1977.

[4] Arakawa, A., Lamb, V.R., Computational Design of the Basic Dynamical Processes of the UCLA General Circulation Model, *Methods of Computational Physics*, 17, 1977.

[5] Arge, E., Kunoth, A., An efficient ADI-solver for scattered data problems with global smoothing, *J.Comp.Phys.*, 139, 343-358, 1998.

[6] Axelsson, O., Iterative Solution Methods, *Cambridge University Press*, 1994.

[7] Edited by Bell, M.J., Notes on how to choose parameter values for the Cox numerical ocean circulation model, *Forecasting Research Division Technical Report No.135*, 1994.

[8] Bell, M.J., Semi-implicit schemes and various grids for ocean dynamics, *Met.Office Internal report*, 2000.

[9] Birkhoff, G. Varga, R., Implicit Alternating Direction Methods. *Trans. Amer. Math. Soc.*, 92, 13-24, 1959.

[10] Brandt, A., Multi-level Adaptive Solutions to Boundary-Value problems, *Math.Comp.*, 31, 333-390, 1977.

[11] Briggs, W.L., A Multigrid Tutorial, *SIAM Books, Philadelphia*, 1987.

[12] Brown, D.E., Nichols, N.K., Bell, M.J., Preconditioners for Anisotropic Problems in Spherical Geometry, submitted for publication to *Int.J.Num.Meth.Fluids.*, 2004.

[13] Bryan, K., A numerical method for the study of the circulation of the World Ocean, *J.Comput.Phys*, 4, 347-376, 1969.

[14] Chan, R.H., Chan, T.F., Circulant Preconditioners for Elliptic Problems, *J.Num.Lin.Alg.Appl.*, 1, 77-101, 1992.

[15] Chan, T., Van Der Vorst, H.A., Approximate and Incomplete Factorizations, *UCLA*, Technical Report 94-27, 1994.

[16] Chan, T., Elman, H.C., Fourier Analysis of iterative methods for elliptic problems, *SIAM.Rev.*, 31(1), 20-49, 1989.

[17] Concus, P., Golub, G.H., O'Leary, D.P., A Generalized Conjugate Gradient Method for the Numerical Solution of Elliptic Partial Differential Equations, in *Sparse Matrix Computations*, ed.J.R.Bunch and D.J.Rose, Academic Press, New York, 1976.

[18] Cox, M.D., A primitive equation, three-dimensional model of the ocean, *GFDL Ocean Group Technical Report*, 1, Geophys.Fluid.Dyn.Lab., Princeton University, Princeton, N.J., 143pp, 1984.

[19] Doss, S., Miller, K., Dynamic ADI-methods for elliptic equations, *SIAM.J.Num.Anal.*, 16(5), 837-856, 1979.

[20] Douglas, J.JR., Garder A.O., Pearcy, C., Multistage Alternating Direction Methods, *SIAM.J.Num.Anal.*, 3(4), 1966.

[21] Duff, I.S., Van Der Vorst, H.A., Preconditioning and Parallel Preconditioning, *CERFACS*, Technical Report TR/PA/98/25, 1998.

[22] Duff, I.S., Meurant, G.A., The effect of ordering on Preconditioned Conjugate Gradients, *BIT*, 29, 635-657, 1989.

[23] Dukowicz, J.K., Smith, R.D., A Reformulation and Implementation of the Bryan-Cox-Semtner Ocean Model on the Connection Machine, *J.Atmos.Oceanic.Technol.*, 10, 195-208, 1993.

[24] Dukowicz, J.K., Smith, R.D., Implicit free-surface method for the Bryan-Cox-Semtner Ocean Model, *J.Geophys.Res.*, 99, 7991-8014, 1994.

[25] Dukowicz, J.K., Dvinsky, A., Approximate factorization as a high order splitting for the implicit flow equations, *J.Comp.Phys.*, 102, 336-347, 1992.

[26] Dukowicz, J.K., Mesh effects for Rossby waves, *J.Comp.Phys.*, 119, 188-194, 1995.

[27] Eisenstat, S., Elman, H., Schultz, M., Variational iterative methods for nonsymmetric systems of linear equations, *SIAM.J.Num.Anal.*, 20(2), 345-357, 1983.

[28] Elman, H., Silvester, D., Wathen, A., Block preconditioners for the discrete incompressible Navier-Stokes equations, *Int.J.Num.Meth.Fluids.*, 20.

[29] Elman, H., Silvester, D., Wathen, A., Performance and Analysis of Saddle Point Preconditioners for the Discrete Steady-State Navier-Stokes Equations, *Numerische Mathematik*, 90, 641-664, 2002.

[30] Ezer, T. and Mellor, G.L., Sensitivity studies with the North Atlantic Princeton Ocean Model. Ocean Circulation Model Evaluation Experiments for the North Atlantic Basin, E. P. Chassignet and P. Malanotte-Rizzoli (Eds.), *Dyn.Atmos.Ocean.*, 32, 185-208, 2000.

[31] Fairweather, G., Mitchell, A., A new computational procedure for ADI methods, *SIAM.J.Num.Anal.*, 4(2), 163-170, 1967.

[32] Feingold, D.G., Varga, R., Block Diagonally Domainant matrices and Generalizations of the Gerschgorin theorem, *Pacific.Jour.Math.*, 12, 1241-1250, 1962.

[33] Forsythe, G.E., Wasow, W.R., Finite-Difference Methods for Partial Differential Equations, *John Wiley and Sons*, 1960.

[34] Fulton, S., Ciesielski, P., Schubert, W., Multigrid methods for elliptic problems, *Mon.Wea.Rev*, 114, 943-959, 1986.

[35] Gent, P.R., Bryan, F.O., Danabosoglu, G., Doney, S.C., Holland, W.R., Large, W.G., McWilliams, J.C., The NCAR Climate System Model Global Ocean Component, *J.Climate*, 11, 1287-1306, 1998.

[36] Gill, A.E., Atmosphere-Ocean Dynamics, *Academic Press*, 1982.

[37] Gill, P., Murray, W., Wright, M., Practical Optimization, *Academic Press*, 1981.

[38] Golub, G.H., Van Loan, C.F., Matrix Computations, *John Hopkins University Press*, 1983.

[39] Golub, G.H., Kent, M.D., Estimates of eigenvalues for iterative methods, *Math. of Comp.*, 53, 619-626, 1989.

[40] Hackbusch, W., Iterative Solution of Large Sparse Systems of Equations, *Springer-Verlag*, 1994.

[41] Hageman, L.A., Young, D.M., Applied Iterative Methods, *Academic Press*, 1981.

[42] Halliwell, G.R., Evaluation of vertical co-ordinate and vertical mixing algorithms in the HYbrid Co-ordinate Ocean Model (HYCOM), *Oce.Mod.*, 7, 285-322, 2004.

[43] Jancic, Z.J., A stable centered difference scheme free of two-grid interval noise, *Mon.Wea.Rev.*, 102, 319-323, 1974.

[44] Kantha, L.H., Clayson, C.A., Numerical models of Ocean and Oceanic processes, *Academic Press*, 2000.

[45] Kettler, R., Analysis and Comparison of Relaxation Schemes in Robust Multigrid and Preconditioned Conjugate Gradient Methods, in *Multigrid Methods*(Hackbusch, W., Trottenburg, U., eds), vol 960 of *Lecture Notes in Mathematics (Springer Verlag)*, 502-534, 1982.

[46] Killworth, P.D., Stainforth, D., Webb, D.J., Paterson, S.M., The development of a free-surface Bryan-Cox-Semtner ocean model, *J.Phys.Ocean.*, 21, 1333-1348, 1991.

[47] Killworth, P.D., Chelton, D., De Szoeke, R., The speed of Observed and Theoretical Long Extratropical Planetary waves, *J.Phys.Ocean.*, 27, 1946-1966, 1997.

[48] Kincaid, D., Young, D., The modified Successive Overrelaxation Method with fixed parameters, *Math.Comp.*, 26, 119, 705-717, 1972.

[49] Le Roux, D.Y., Staniforth, A., Lin, C.A., Finite Elements for Shallow-Water Equation Ocean models, *Mon.Wea.Rev.*, 126, 1931-1951, 1998.

[50] Lirkov, I.D., Margenov, S.D., Vassilevski, P.S., Circulant Block Factorization Preconditioners for Elliptic Problems, *Computing*, 53, 59-74, 1994.

[51] Lirkov, I.D., Margenov, S.D., Zikatanov, L.T., Circulant Block Factorization Preconditioning of Anisotropic Elliptic Problems, *Computing*, 58, 245-258, 1997.

[52] Livne, O.E., Golub, G.H., Scaling by binormalization, *SCCM Technical Report SCCM-03-11, Stanford University, USA*, 2003.

[53] Madec, G., Delecluse, P., Imbard, M., Levy, C., OPA 8.1 Ocean General Circulation Model reference manual, *Note du Pole de modelisation, Institut Pierre-Simon Laplace*, 11, 1998. Available online at www.lodyc.jussieu.fr/opa/

[54] Madec, G., Imbard, M., A global ocean mesh to overcome the North Polar singularity, *Clim. Dyn.*, 12, 381-388, 1996.

[55] Malhotra, S., Douglas, C.C., Schultz, M., Parameter Choices for ADI-like methods on Parallel Computers. *Comput.Appl.Math.*, 17(3), 221-236, 1998.

[56] Marshall, J., Adcroft, A., Hill, C., Perelmen, L., Heisey, C., A finite volume, incompressible Navier-Stokes model for studies of the ocean on parallel computers, *J.Geophys.Res.*, 102(C3), 5753-5766, 1997.

[57] Marsland, S.J., Haak. H., Jungclaus, J.H., Latif, M., and Roske, F., The Max-Planck-Institute global ocean/sea ice model with orthogonal curvilinear coordinates, *Ocean Modelling*, 5(2), 91-127, 2003.

[58] Mawson, M., Numerical solution of Elliptic equations using Multigrid methods, *Met.Office Internal Report*, 1994.

[59] Muller, C., Sirovich, L., John, F., Analysis of Spherical Symmetries in Euclidean Spaces, *Springer-Verlag*, 1997.

[60] Navon, I. Legler, D., Conjugate gradient methods for large-scale minimization in Meteorology, *Mon.Wea.Rev.*, 115, 1479-1502, 1987.

[61] Ng, M.K., Preconditioning of elliptic problems by approximation in the transform domain, *Aust.Nat.Uni.Tech.Rep.*, 1997.

[62] Nichols, N.K., Numerical solution of elliptic differential equations, *PHd Thesis, Oxford University* 1969.

[63] Pacanowski, R.C., Griffies, S.M., The MOM 3.0 Manual, *GFDL/NOAA Princeton*, 1999. Available online at www.gfdl.noaa.gov/

[64] Randall, D.A., Geostrophic Adjustment and the Finite-Difference Shallow Water Equations, *Mon.Wea.Rev.*, 122, 1371-1377, 1994.

[65] Rickard, G. Cresswell, D., Possible schemes for replacement of the North Polar island with a North Polar point in the ocean model, *Met.Office Internal report*, 2000.

[66] Semtner, A.J., A general circulation model for the World Ocean, *UCLA Dept.of Met.Tech Report*, 8, 1974.

[67] Semtner, A.J., Finite-difference formulation of a world ocean model, *Advanced Physical Oceanographic Numerical Modelling*, edited by J.J.O'Brien, D. Reidel Publishing Company, 187-202, 1986.

[68] Skamarock, W.C., Smolarkiewicz, P.K., Klemp, J.B., Preconditioned Conjugate-Residual Solvers for Helmholtz Equations in Nonhydrostatic Models, *Mon.Wea.Rev.*, 125, 587-599, 1997.

[69] Smolarkiewicz, P. Margolin, L., Variational methods for elliptic problems in fluid models, *Nat.Cen.Atmos.Res., Los Alamos*, 137-159.

[70] Smolarkiewicz, P. Margolin, L., Variational solver for elliptic problems in atmospheric flows, *Appl. Math and Comp. Sci.*, 4, 527-551, 1994.

[71] Song, Y.T, Wright, D.G., A semi-implicit ocean circulation model using a generalized topography following co-ordinate system, *J.Comp.Phys.*, 115, 228-244, 1994.

[72] Spiteri, P., Miellou, J.C., Bahi, J.M., Evolution of parameters for the optimization of SSOR and ADI preconditioning, *Num.Algo.*, 29, 249-265, 2002.

[73] Starke, G., Optimal Alternating Direction Implicit parameters for non-symmetric systems of linear equations, *SIAM.J.Num.Anal.*, 28(5), 1431-1445, 1991.

[74] Tatebe, O., The Multigrid Preconditioned Conjugate Gradient Method, *Sixth Copper Mountain Conference on Multigrid Methods*, 1993.

[75] Varga, R.S., Matrix Iterative Analysis, *Prentice Hall*, 1962.

[76] Wachspress, E., Habetler, G., an alternating-direction implicit iteration technique, *J.Soc.Indust.Appl.Math.*, 8(2), 403-421, 1960.

[77] Wachspress, E., Optimum alternating direction implicit iteration parameters for a model problem, *J.Soc.Indust.Appl.Math.*, 10(2), 339-350, 1962.

[78] Webb, D.J., de Cuevas, B.A., Coward, A.C.C., The first main run of the OCCAM global ocean model, *Southampton Oceanography Centre, Internal Report of James Rennell Division*, 34, 1998.

[79] Widlund, O., On the rate of convergence of an alternating direction implicit method in a noncommutative case, *Math.Comp.*, 20, 500-515, 1966.

[80] Young, D.M., Convergence Properties of the Symmetric and Unsymmetric Successive Overrelaxation Methods and Related Methods, *Math.Comp.*, 24, 793-807, 1970.

[81] Young, D.M., Iterative solution of large linear systems, *Academic press*, 1971.

[82] Young, D.M., A New Class of Parallel Alternating-Type Iterative Methods, *J.Comp.Appl.Math.*, 74, 331-344, 1996.

Appendix A

Free surface formulation

$$\begin{aligned}
&\tfrac{u^{n+1}-u^{n-1}}{2\delta t}-fv^{\alpha'}=-g\tfrac{1}{acos\phi}\tfrac{\partial\eta^{\alpha}}{\partial\lambda}+G^{\lambda,n},\\
&\tfrac{v^{n+1}-v^{n-1}}{2\delta t}+fu^{\alpha'}=-g\tfrac{1}{a}\tfrac{\partial\eta^{\alpha}}{\partial\phi}+G^{\phi,n},\\
&\tfrac{\eta^{n+1}-\eta^{n}}{\delta t}+\tfrac{1}{acos\phi}\left[\tfrac{\partial Hu^{\theta}}{\partial\lambda}+\tfrac{\partial Hv^{\theta}cos\phi}{\partial\phi}\right]=0,
\end{aligned}\tag{A.1}$$

where

$$\begin{aligned}
u^{\alpha'}&=\alpha'u^{n+1}+(1-\alpha'-\gamma')u^{n}+\gamma'u^{n-1},\\
v^{\alpha'}&=\alpha'v^{n+1}+(1-\alpha'-\gamma')v^{n}+\gamma'v^{n-1},\\
\eta^{\alpha}&=\alpha\eta^{n+1}+(1-\alpha-\gamma)\eta^{n}+\gamma\eta^{n-1},\\
u^{\theta}&=\theta u^{n+1}+(1-\theta)u^{n}\\
v^{\theta}&=\theta v^{n+1}+(1-\theta)v^{n}.
\end{aligned}\tag{A.2}$$

We require to calculate u^{n+1} and v^{n+1} and substitute them into the u^{θ} and v^{θ} terms. We have

$$u^{n+1}-\tau f\alpha'v^{n+1}=u^{n-1}+\tau f\left[(1-\alpha'-\gamma')v^{n}+\gamma'v^{n-1}\right]-\frac{\tau g}{acos\phi}\frac{\partial\eta^{\alpha}}{\partial\lambda}+\tau G^{\lambda,n}\tag{A.3}$$

and

$$\tau f\alpha'u^{n+1}+v^{n+1}=v^{n-1}-\tau f\left[(1-\alpha'-\gamma')u^{n}+\gamma'u^{n-1}\right]-\frac{\tau g}{a}\frac{\partial\eta^{\alpha}}{\partial\phi}+\tau G^{\phi,n}\tag{A.4}$$

Taking (A.3) + $\tau f \alpha'$ (A.4) and dividing by $1 + \tau^2 f^2 \alpha'^2$ gives

$$u^{n+1} = \tfrac{1}{1+\tau^2 f^2 \alpha'^2} \Big\{ u^{n-1} + \tau f \left[(1-\alpha'-\gamma') v^n + \gamma' v^{n-1} \right] - \tfrac{\tau g}{acos\phi} \tfrac{\partial \eta^\alpha}{\partial \lambda}$$
$$+\tau G^{\lambda,n} + \tau f \alpha' v^{n-1} - \tau^2 f^2 \alpha' \left[(1-\alpha'-\gamma') u^n + \gamma' u^{n-1} \right] - \tfrac{\tau^2 f \alpha' g}{a} \tfrac{\partial \eta^\alpha}{\partial \phi} + \tau^2 f \alpha' G^{\phi,n} \Big\} \tag{A.5}$$

Similarly taking $-\tau f \alpha'$ (A.3) + (A.4) and dividing by $1 + \tau^2 f^2 \alpha'^2$ gives

$$v^{n+1} = \tfrac{1}{1+\tau^2 f^2 \alpha'^2} \Big\{ -\tau f \alpha' u^{n-1} - \tau^2 f^2 \alpha' \left[(1-\alpha'-\gamma') v^n + \gamma' v^{n-1} \right] + \tfrac{\tau^2 f \alpha' g}{acos\phi} \tfrac{\partial \eta^\alpha}{\partial \lambda}$$
$$-\tau^2 f \alpha' G^{\lambda,n} + v^{n-1} - \tau f \left[(1-\alpha'-\gamma') u^n + \gamma' u^{n-1} \right] - \tfrac{\tau g}{a} \tfrac{\partial \eta^\alpha}{\partial \phi} + \tau G^{\phi,n} \Big\} \tag{A.6}$$

Now substitute u^{n+1} and v^{n+1} in u^θ and v^θ terms in equation (A.1). Collect terms involving η^{n+1} on the left hand side of the equation and place all others terms on the right hand side. This gives

$$\frac{2(1+\tau^2 f^2 \alpha'^2)\eta^{n+1}}{\alpha \theta g \tau^2} - \frac{1}{acos\phi} \frac{\partial}{\partial \lambda} \left(\frac{H}{acos\phi} \frac{\partial \eta^{n+1}}{\partial \lambda} \right) - \frac{1}{a} \frac{\partial}{\partial \phi} \left(\frac{Hcos\phi}{a} \frac{\partial \eta^{n+1}}{\partial \phi} \right) = S(\lambda, \phi) \tag{A.7}$$

Explicit time differencing is used for the Coriolis terms ($\alpha', \gamma' = 0$). Dukowicz [24] considers options for choosing α, θ, τ subject to stability and mode damping considerations.

Appendix B

Additional numerical results

B.1 Limited area

Table B.1 gives the parameter values used in the ADI preconditioner in the Limited Area case. We note that the parameter increases with smaller stepsizes and as ϕ_{NB} gets very small or very large. Also displayed in this Appendix section are the full results for the ∞-norm condition numbers of A for varying k (for $\phi_{NB} = 88^o$, $h = 1^o$), the full spectral radii results for the iteration matrices, G, as well as the leading eigenvectors of G_{Block}, G_{ADI} and G_{Bin} for the $\phi_{NB} = 40^o$ case, and full results for the ∞ and 2 norm condition numbers of the preconditioned system matrices.

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^o N	1472.8	732.8	361.9
70^o N	945.6	466.9	227.3
88^o N	1565.7	714.5	308.9
89^o N	1895.6	821.6	NA
89.5^o N	2215.7	NA	NA

Table B.1: ADI Parameter values, sine function general solution

k	$\kappa_\infty(A)$	k	$\kappa_\infty(A)$	k	$\kappa_\infty(A)$
0.0	6.511×10^3	7.5	5.779×10^3	2500.0	496.850
0.001	6.510×10^3	10.0	5.569×10^3	5000.0	350.435
0.01	6.509×10^3	25.0	4.564×10^3	7500.0	283.834
0.1	6.500×10^3	50.0	3.511×10^3	1.0×10^4	242.184
0.25	6.483×10^3	75.0	2.886×10^3	2.5×10^4	138.008
0.5	6.456×10^3	100.0	2.498×10^3	5.0×10^4	83.358
0.75	6.429×10^3	250.0	1.575×10^3	7.5×10^5	59.509
1.0	6.403×10^3	500.0	1.113×10^3	1.0×10^5	46.214
2.5	6.247×10^3	750.0	908.333	2.5×10^5	20.357
5.0	6.004×10^3	1000.0	786.107	5.0×10^5	19.547

Table B.2: ∞ norm condition numbers of system matrix A, varying k, $\phi_{NB} = 88^o$, $h = 1^o$.

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^o N	1.489×10^3	369.417	91.348
70^o N	3.017×10^3	733.169	174.635
88^o N	2.225×10^4	4.631×10^3	864.540
89^o N	3.705×10^4	6.939×10^3	NA
89.5^o N	5.556×10^4	NA	NA

Table B.3: 2 norm condition numbers of system matrix A, $k = 0.01$

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^o N	0.9986	0.9946	0.9783
70^o N	0.9990	0.9960	0.9842
88^o N	0.9990	0.9960	0.9842
89^o N	0.9990	0.9960	NA
89.5^o N	0.9990	NA	NA

Table B.4: Spectral Radii of iteration matrix G for diagonal preconditioner, $k = 0.01$

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^o N	0.9970	0.9879	0.9528
70^o N	0.9975	0.9901	0.9614
88^o N	0.9975	0.9901	0.9614
89^o N	0.9975	0.9901	NA
89.5^o N	0.9975	NA	NA

Table B.5: Spectral Radii of iteration matrix G for block diagonal preconditioner, $k = 0.01$

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^o N	0.9105	0.8289	0.6859
70^o N	0.9629	0.9273	0.8602
88^o N	0.9793	0.9601	0.9241
89^o N	0.9842	0.9642	NA
89.5^o N	0.9871	NA	NA

Table B.6: Spectral Radii of iteration matrix G for ADI preconditioner, $k = 0.01$

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^oN	0.9993	0.9973	0.9891
70^oN	0.9995	0.9980	0.9921
88^oN	0.9995	0.9980	0.9921
89^oN	0.9995	0.9980	NA
89.5^oN	0.9995	NA	NA

Table B.7: Spectral radii of iteration matrix G for Binormalization scaling, $k = 0.01$

B.2 Periodic domain problem

Table B.16 gives the parameter values used in the ADI preconditioner for the periodic domain case. We note here that again the values increase with smaller stepsizes and in this case decrease as ϕ_{NB} is moved closer to the pole. Figure B.13 again shows that the largest values in magnitude in the leading eigenvector of A are found clustered near the northern boundary. Also displayed in this Appendix section are the full results for the ∞ norm condition numbers of the preconditioned system matrices, and the spectral radii of the preconditioned iteration matrices.

B.3 Unforced problem : Fourier Modes as initial errors

The eigenvectors associated with the leading eigenvalues of the preconditioned iteration matrices, G, are displayed in this appendix section.

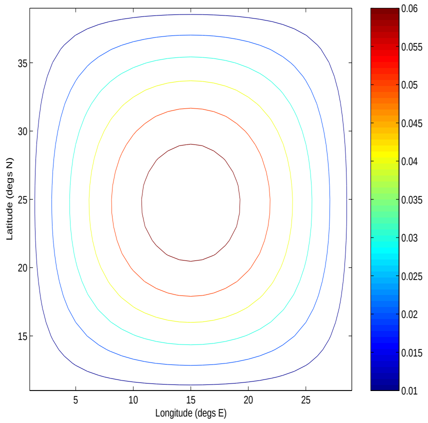

Figure B.1: Eigenvector associated with largest eigenvalue (0.9879) of G_{Block} for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

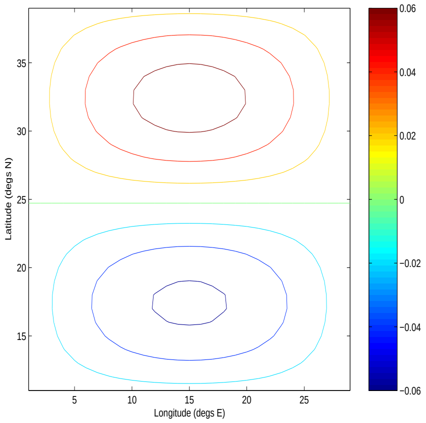

Figure B.2: Eigenvector associated with second largest eigenvalue (0.9715) of G_{Block} for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

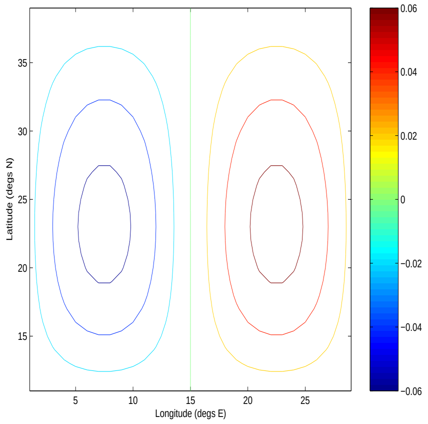

Figure B.3: Eigenvector associated with third largest eigenvalue (0.9668) of G_{Block} for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

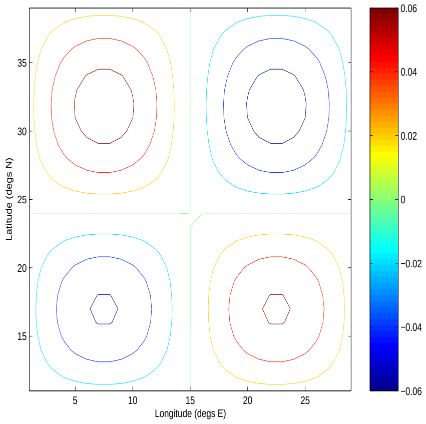

Figure B.4: Eigenvector associated with fourth largest eigenvalue (0.9520) of G_{Block} for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

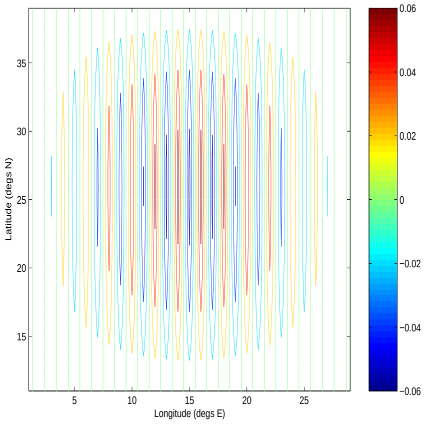

Figure B.5: Eigenvector associated with largest eigenvalue (-0.8289) of G_{ADI} for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

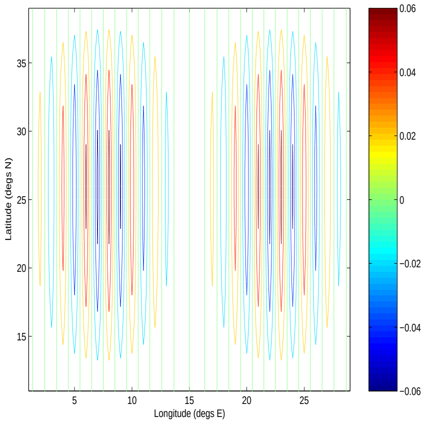

Figure B.6: Eigenvector associated with second largest eigenvalue (-0.8282) of G_{ADI} for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

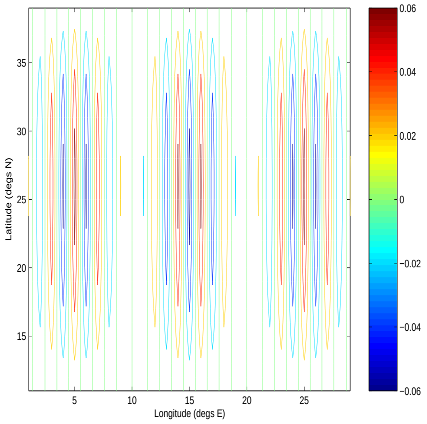

Figure B.7: Eigenvector associated with third largest eigenvalue (-0.8270) of G_{ADI} for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

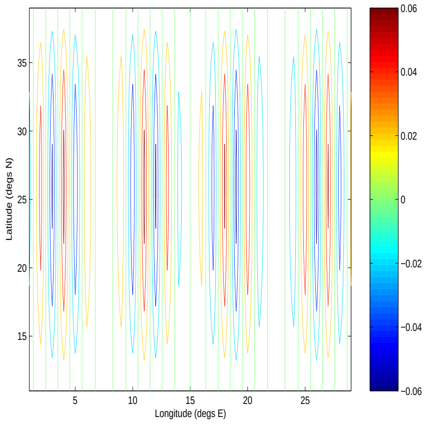

Figure B.8: Eigenvector associated with fourth largest eigenvalue (-0.8254) of G_{ADI} for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

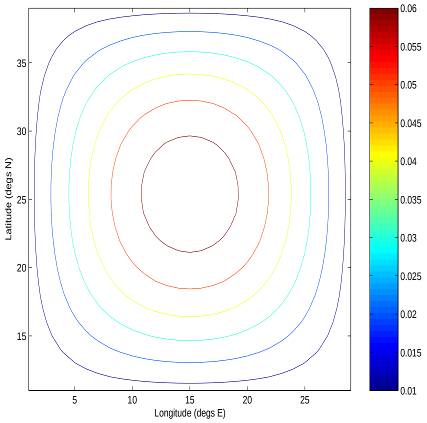

Figure B.9: Eigenvector associated with largest eigenvalue (0.9973) of G_{BIN} for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

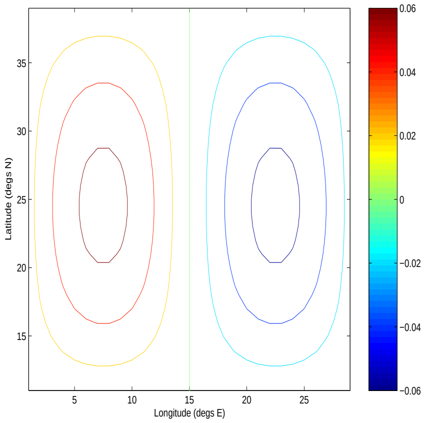

Figure B.10: Eigenvector associated with second largest eigenvalue (0.9936) of G_{BIN} for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

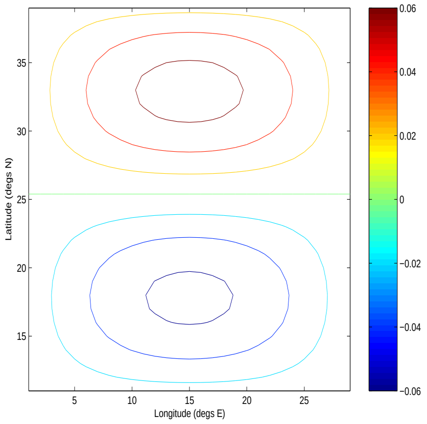

Figure B.11: Eigenvector associated with third largest eigenvalue (0.9928) of G_{BIN} for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

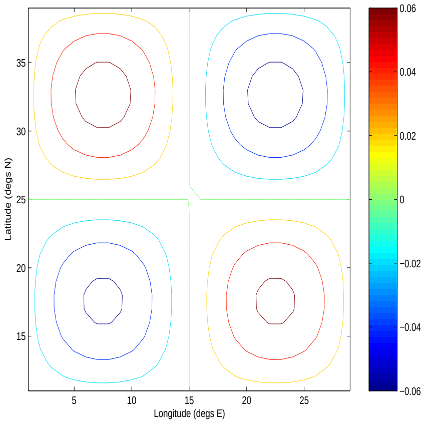

Figure B.12: Eigenvector associated with fourth largest eigenvalue (0.9891) of G_{BIN} for Limited Area Helmholtz problem. $\phi_{NB} = 40^o$.

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^o N	2.129×10^3	531.649	131.954
70^o N	2.766×10^3	691.359	171.910
88^o N	2.768×10^3	691.915	172.076
89^o N	2.768×10^3	691.915	NA
89.5^o N	2.768×10^3	NA	NA

Table B.8: ∞ norm condition numbers of preconditioned system matrix $P^{-1}A$, diagonal preconditioner, $k = 0.01$

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^o N	1.468×10^3	366.588	91.141
70^o N	2.014×10^3	503.280	125.379
88^o N	2.022×10^3	505.214	125.869
89^o N	2.022×10^3	505.214	NA
89.5^o N	2.022×10^3	NA	NA

Table B.9: 2 norm condition numbers of preconditioned system matrix $P^{-1}A$, diagonal preconditioner, $k = 0.01$

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^o N	962.963	241.440	60.815
70^o N	1.171×10^3	293.816	74.204
88^o N	1.172×10^3	293.920	74.229
89^o N	1.172×10^3	293.920	NA
89.5^o N	1.172×10^3	NA	NA

Table B.10: ∞ norm condition numbers of preconditioned system matrix $P^{-1}A$, block diagonal preconditioner, $k = 0.01$

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^o N	658.734	164.871	41.405
70^o N	801.994	201.275	50.859
88^o N	802.679	201.289	50.863
89^o N	802.679	201.289	NA
89.5^o N	802.679	NA	NA

Table B.11: 2 norm condition numbers of preconditioned system matrix $P^{-1}A$, block diagonal preconditioner, $k = 0.01$

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^o N	152.354	62.681	22.504
70^o N	191.989	82.067	30.534
88^o N	324.273	139.311	37.593
89^o N	333.676	142.386	NA
89.5^o N	337.933	NA	NA

Table B.12: ∞ norm condition numbers of preconditioned system matrix $P^{-1}A$, ADI preconditioner, $k = 0.01$

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^o N	20.516	10.254	5.137
70^o N	27.788	14.448	7.574
88^o N	29.192	14.693	7.856
89^o N	32.464	15.902	NA
89.5^o N	33.886	NA	NA

Table B.13: 2 norm condition numbers of preconditioned system matrix $P^{-1}A$, ADI preconditioner, $k = 0.01$

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^oN	2.131×10^3	531.945	132.069
70^oN	2.778×10^3	694.101	172.232
88^oN	2.781×10^3	694.737	172.380
89^oN	2.781×10^3	694.737	NA
89.5^oN	2.781×10^3	NA	NA

Table B.14: ∞ norm condition numbers of DAD using binormalization, Limited Area

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^oN	1.469×10^3	367.479	91.932
70^oN	2.015×10^3	504.059	126.166
88^oN	2.023×10^3	505.947	126.646
89^oN	2.023×10^3	505.947	NA
89.5^oN	2.023×10^3	NA	NA

Table B.15: 2 norm condition numbers of DAD using binormalization, Limited Area

	Stepsize	
ϕ_{NB}	1^o	2^o
40^o N	629.746	310.058
70^o N	272.457	134.292
88^o N	159.343	78.755
89^o N	149.298	NA

Table B.16: ADI parameters values, $k = 0.01$

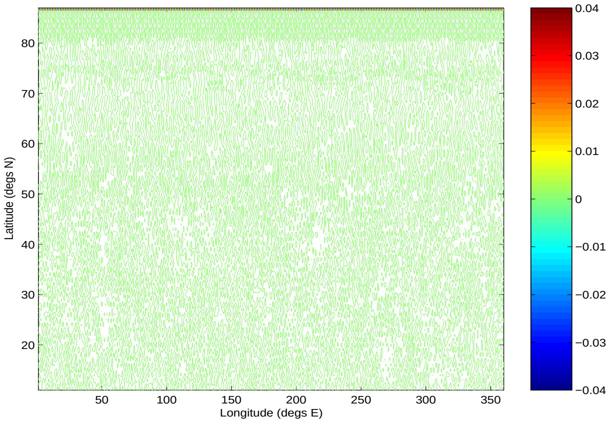

Figure B.13: Eigenvector associated with largest eigenvalue of A for Periodic domain Helmholtz problem. $\phi_{NB} = 88^o$.

	Stepsize	
ϕ_{NB}	1^o	2^o
40^o N	1.04×10^3	259.106
70^o N	8.27×10^3	1.99×10^3
88^o N	1.21×10^5	2.82×10^4
89^o N	2.09×10^5	NA

Table B.17: ∞ norm condition numbers of system matrix A, $k = 0.01$

	Stepsize	
ϕ_{NB}	1^o	2^o
40^o N	1.02×10^3	254.832
70^o N	5.79×10^3	1.45×10^3
88^o N	2.23×10^4	1.13×10^4
89^o N	2.97×10^4	NA

Table B.18: ∞ norm condition numbers of preconditioned system matrix $P^{-1}A$, diagonal preconditioner, $k = 0.01$

	Stepsize	
ϕ_{NB}	1^o	2^o
40^o N	458.936	114.528
70^o N	1.54×10^3	506.151
88^o N	3.25×10^3	1.14×10^3
89^o N	3.48×10^3	NA

Table B.19: ∞ norm condition numbers of preconditioned system matrix $P^{-1}A$, block diagonal preconditioner, $k = 0.01$

	Stepsize	
ϕ_{NB}	1^o	2^o
40^o N	142.698	35.577
70^o N	398.572	152.981
88^o N	972.656	336.559
89^o N	1.02×10^3	NA

Table B.20: ∞ norm condition numbers of preconditioned system matrix $P^{-1}A$, ADI preconditioner, $k = 0.01$

	Stepsize	
ϕ_{NB}	1^o	2^o
40^o N	0.9976	0.9904
70^o N	0.9996	0.9983
88^o N	0.9999	0.9996
89^o N	0.9999	NA

Table B.21: Spectral Radii of iteration matrix G for diagonal preconditioner, $k = 0.01$

	Stepsize	
ϕ_{NB}	1^o	2^o
40^o N	0.9946	0.9788
70^o N	0.9987	0.9950
88^o N	0.9994	0.9976
89^o N	0.9994	NA

Table B.22: Spectral Radii of iteration matrix G for block diagonal preconditioner, $k = 0.01$

	Stepsize	
ϕ_{NB}	1^o	2^o
40^o N	0.9052	0.8140
70^o N	0.9595	0.9149
88^o N	0.9739	0.9492
89^o N	0.9787	NA

Table B.23: Spectral Radii of iteration matrix G for ADI preconditioner, $k = 0.01$

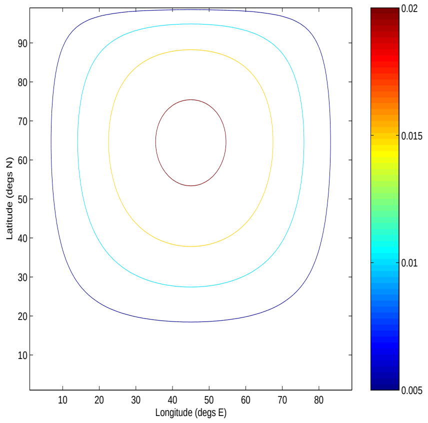

Figure B.14: Eigenvector associated with largest eigenvalue (0.9994) of G_D for unforced limited area problem

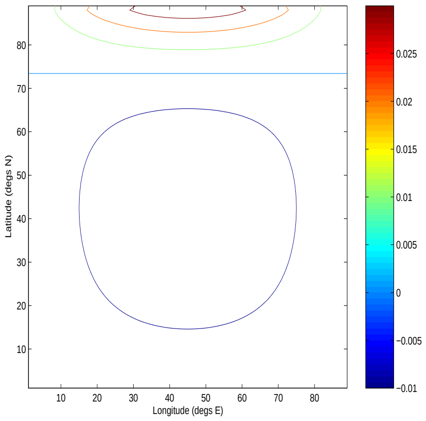

Figure B.15: Eigenvector associated with second largest eigenvalue (0.9993) of G_D for unforced limited area problem

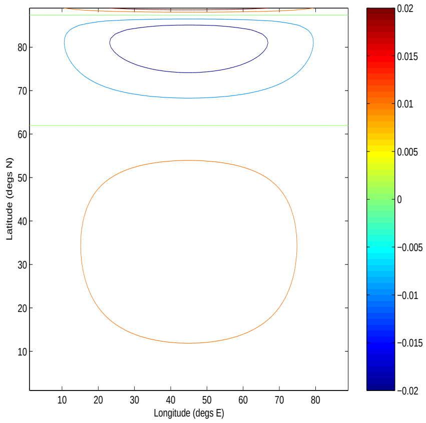

Figure B.16: Eigenvector associated with third largest eigenvalue (0.9992) of G_D for unforced limited area problem

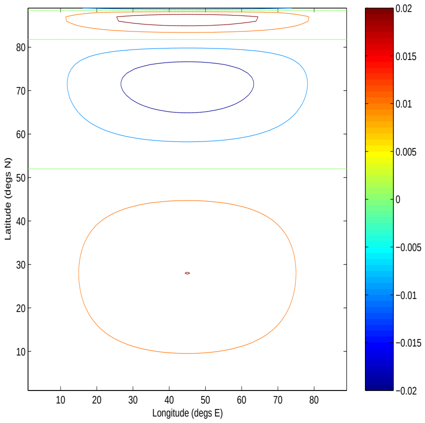

Figure B.17: Eigenvector associated with fourth largest eigenvalue (0.9989) of G_D for unforced limited area problem

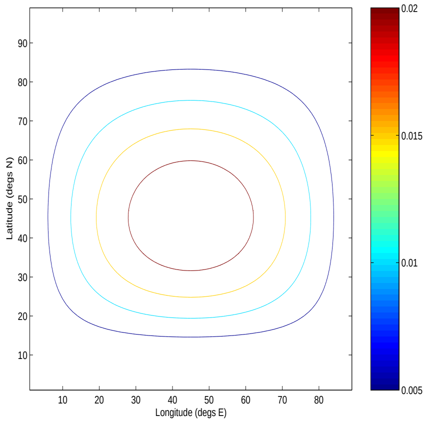

Figure B.18: Eigenvector associated with largest eigenvalue (0.9975) of G_{Block} for unforced limited area problem

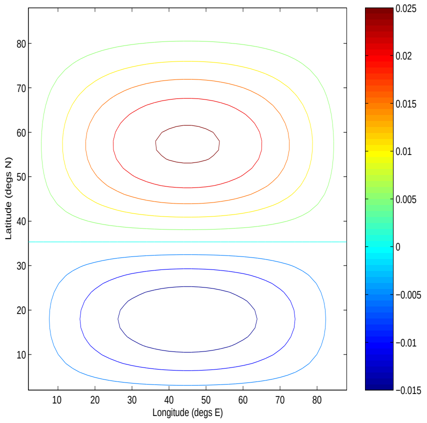

Figure B.19: Eigenvector associated with joint second largest eigenvalue (0.9950) of G_{Block} for unforced limited area problem

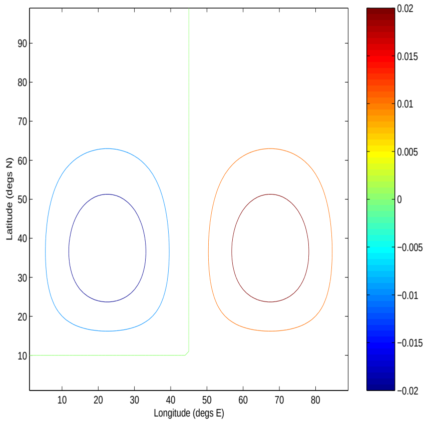

Figure B.20: Eigenvector associated with other joint second largest eigenvalue (0.9950) of G_{Block} for unforced limited area problem

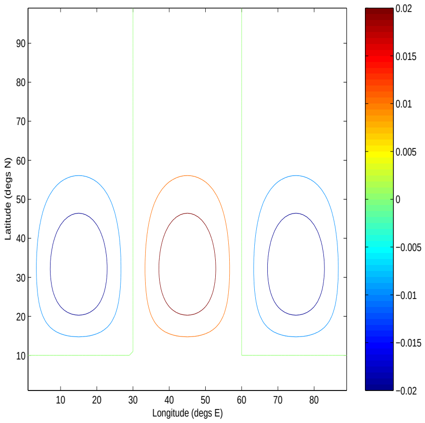

Figure B.21: Eigenvector associated with fourth largest eigenvalue (0.9915) of G_{Block} for unforced limited area problem

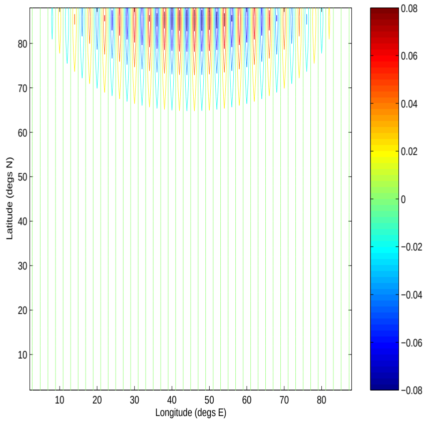

Figure B.22: Eigenvector associated with largest eigenvalue (-0.9853) of G_{ADI} for unforced limited area problem

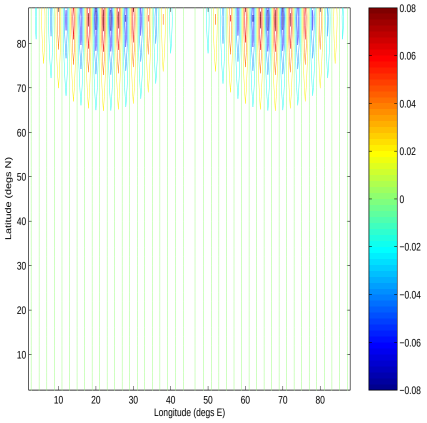

Figure B.23: Eigenvector associated with second largest eigenvalue (-0.9851) of G_{ADI} for unforced limited area problem

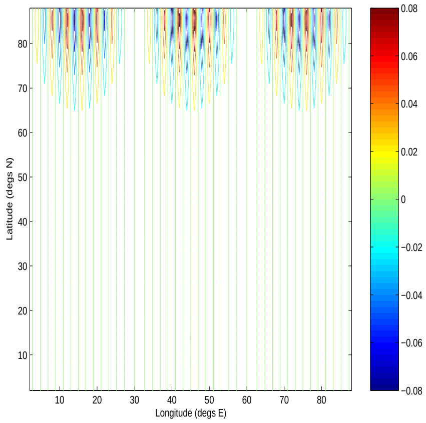

Figure B.24: Eigenvector associated with third largest eigenvalue (-0.9847) of G_{ADI} for unforced limited area problem

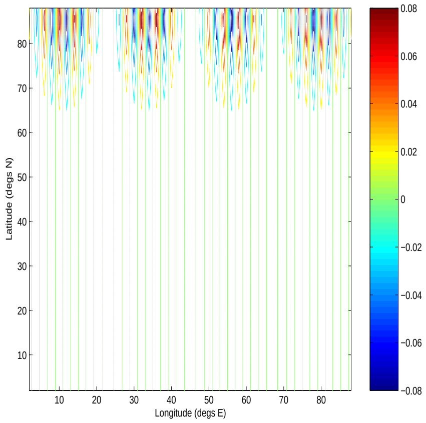

Figure B.25: Eigenvector associated with fourth largest eigenvalue (-0.9842) of G_{ADI} for unforced limited area problem

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^o N	0.9984	0.9936	0.9746
70^o N	0.9992	0.9971	0.9887
88^o N	0.9994	0.9979	0.9939
89^o N	0.9994	0.9979	NA
89.5^o N	0.9994	NA	NA

Table B.24: Spectral Radii of G_D, $k = 0.01$, nine-point

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^o N	0.9975	0.9900	0.9603
70^o N	0.9989	0.9955	0.9819
88^o N	0.9992	0.9966	0.9866
89^o N	0.9992	0.9967	NA
89.5^o N	0.9992	NA	NA

Table B.25: Spectral Radii of G_{Block}, $k = 0.01$, nine-point

B.4 Nine-point operator

Figure B.26 again shows that, with the nine-point operator, the largest values in magnitude in the leading eigenvector of A are found clustered near the northern boundary. Also displayed in this Appendix section are the full results for the spectral radii of the preconditioned iteration matrices, G, and the ∞ and 2 norm condition numbers of the preconditioned system matrices.

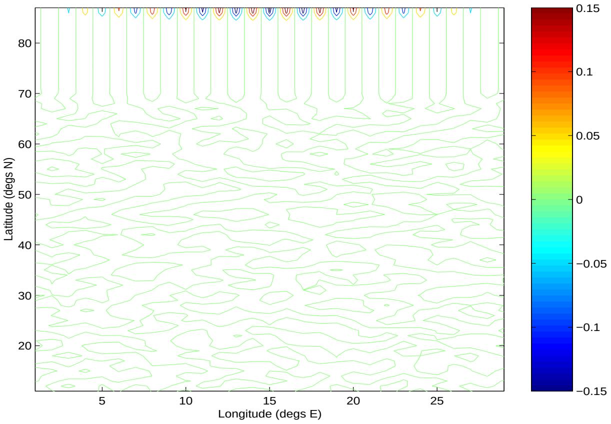

Figure B.26: Eigenvector associated with largest eigenvalue of A for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$. Nine Point Operator

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
$40^o N$	0.9523	0.8788	0.8094
$70^o N$	0.9698	0.9065	0.8200
$88^o N$	0.9813	0.9515	0.9092
$89^o N$	0.9847	0.9583	NA
$89.5^o N$	0.9872	NA	NA

Table B.26: Spectral radii of G_{ADI}, nine point

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
$40^o N$	0.9988	0.9951	0.9806
$70^o N$	0.9994	0.9977	0.9910
$88^o N$	0.9996	0.9983	0.9933
$89^o N$	0.9996	0.9983	NA
$89.5^o N$	0.9996	NA	NA

Table B.27: Spectral radii of G_{BIN}, nine point

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^o N	1.390×10^3	343.330	83.836
70^o N	5.148×10^3	1.232×10^3	282.637
88^o N	3.996×10^4	7.568×10^3	1.365×10^3
89^o N	6.053×10^4	1.093×10^4	NA
89.5^o N	8.735×10^4	NA	NA

Table B.28: ∞ norm condition numbers of system matrix A, $k = 0.01$, nine point

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^o N	903.003	219.307	52.612
70^o N	3.301×10^3	772.211	174.430
88^o N	2.442×10^4	4.775×10^3	859.166
89^o N	3.825×10^4	6.891×10^3	NA
89.5^o N	5.517×10^4	NA	NA

Table B.29: 2 norm condition numbers of system matrix A, $k = 0.01$, nine point

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^o N	$1.356{\times}10^3$	336.026	82.565
70^o N	$3.387{\times}10^3$	861.252	210.611
88^o N	$4.887{\times}10^3$	$1.220{\times}10^3$	303.578
89^o N	$4.973{\times}10^3$	$1.229{\times}10^3$	NA
89.5^o N	$8.832{\times}10^3$	NA	NA

Table B.30: ∞ norm condition numbers of system matrix $P^{-1}A$, Diagonal preconditioner, $k = 0.01$, nine point

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^oN	813.721	203.937	51.436
70^oN	$1.758{\times}10^3$	440.031	110.616
88^oN	$2.381{\times}10^3$	595.809	149.801
89^oN	$2.392{\times}10^3$	598.436	NA
89.5^oN	$2.394{\times}10^3$	NA	NA

Table B.31: 2 norm condition numbers of system matrix $P^{-1}A$, Diagonal preconditioner, $k = 0.01$, nine point

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^oN	$1.201{\times}10^3$	300.516	75.037
70^oN	$2.572{\times}10^3$	640.579	160.721
88^oN	$3.531{\times}10^3$	894.938	220.306
89^oN	$3.538{\times}10^3$	897.453	NA
89.5^oN	$3.578{\times}10^3$	NA	NA

Table B.32: ∞ norm condition numbers of system matrix $P^{-1}A$, Block Preconditioner, $k = 0.01$, nine-point

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^o N	795.569	198.717	49.414
70^o N	$1.756{\times}10^3$	438.617	109.229
88^o N	$2.378{\times}10^3$	594.101	148.092
89^o N	$2.389{\times}10^3$	596.732	NA
89.5^o N	$2.391{\times}10^3$	NA	NA

Table B.33: 2 norm condition numbers of system matrix $P^{-1}A$, Block Preconditioner, $k = 0.01$, nine-point

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^oN	167.074	63.021	20.503
70^oN	189.096	71.836	25.144
88^oN	247.249	89.909	32.649
89^oN	254.950	94.602	NA
89.5^oN	256.331	NA	NA

Table B.34: ∞ norm condition numbers of $P^{-1}A$ ADI preconditioner, nine point

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^oN	39.529	19.064	8.530
70^oN	45.278	19.322	8.952
88^oN	61.323	26.738	11.886
89^oN	67.081	30.714	NA
89.5^oN	71.020	NA	NA

Table B.35: 2 norm condition numbers of $P^{-1}A$ ADI preconditioner, nine point

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^o N	1.297×10^3	322.559	79.780
70^o N	3.740×10^3	923.928	224.071
88^o N	6.316×10^3	1.483×10^3	332.762
89^o N	6.342×10^3	1.489×10^3	NA
89.5^o N	6.349×10^3	NA	NA

Table B.36: ∞ norm condition numbers of DAD using binormalization, nine point

	Stepsize		
ϕ_{NB}	$\frac{1}{2}^o$	1^o	2^o
40^o N	885.637	216.044	52.131
70^o N	2.299×10^3	559.440	133.479
88^o N	3.161×10^3	783.395	189.448
89^o N	3.177×10^3	788.261	NA
89.5^o N	5.517×10^3	NA	NA

Table B.37: 2 norm condition numbers of DAD using binormalization, nine point

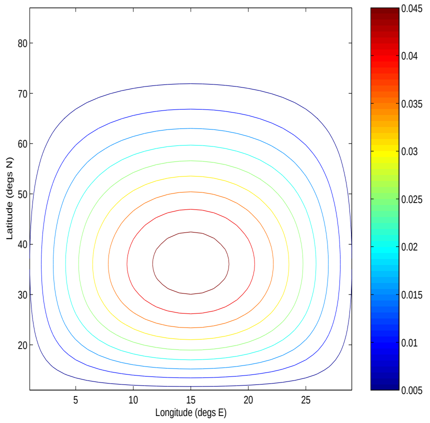

Figure B.27: Eigenvector associated with largest eigenvalue (0.9979) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$. Nine Point Operator

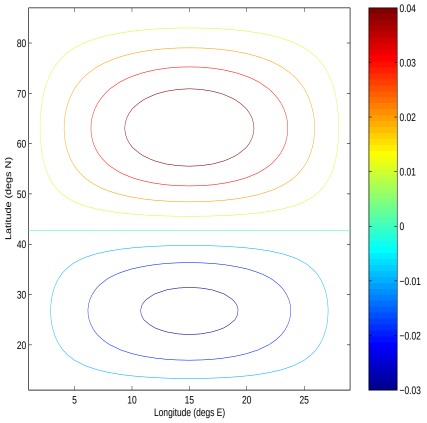

Figure B.28: Eigenvector associated with second largest eigenvalue (0.9968) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$. Nine Point Operator

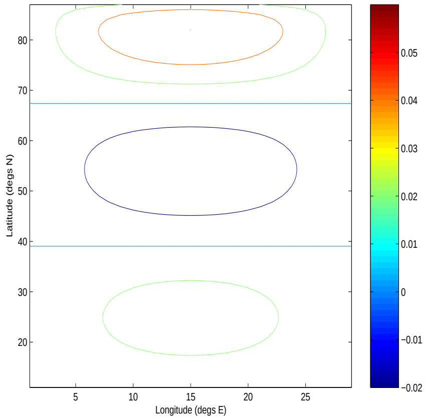

Figure B.29: Eigenvector associated with third largest eigenvalue (0.9961) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$. Nine Point Operator

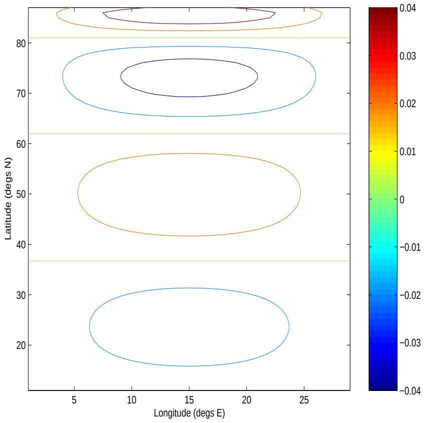

Figure B.30: Eigenvector associated with fourth largest eigenvalue (0.9956) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$. Nine Point Operator

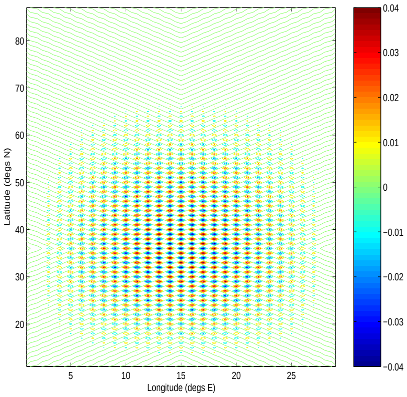

Figure B.31: Eigenvector associated with largest negative eigenvalue (-0.9979) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$. Nine Point Operator

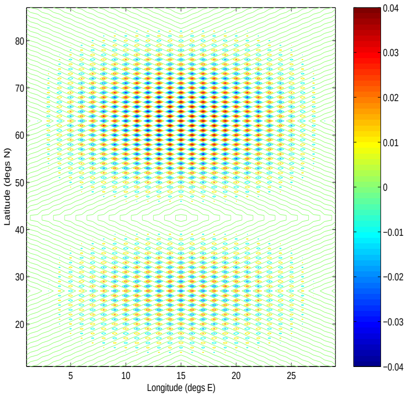

Figure B.32: Eigenvector associated with second largest negative eigenvalue (-0.9968) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$. Nine Point Operator

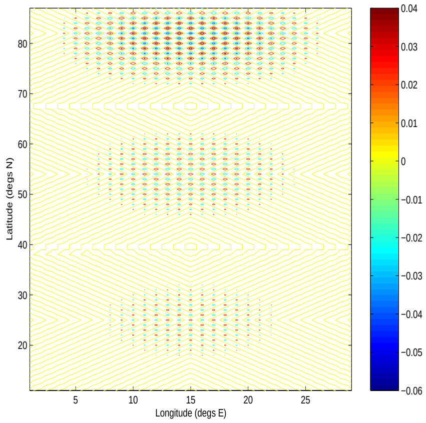

Figure B.33: Eigenvector associated with third largest negative eigenvalue (-0.9961) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$. Nine Point Operator

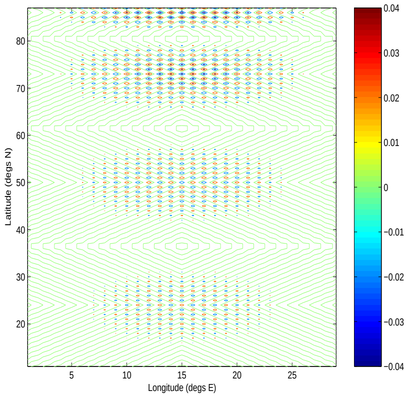

Figure B.34: Eigenvector associated with fourth largest negative eigenvalue (-0.9956) of G_D for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$. Nine Point Operator

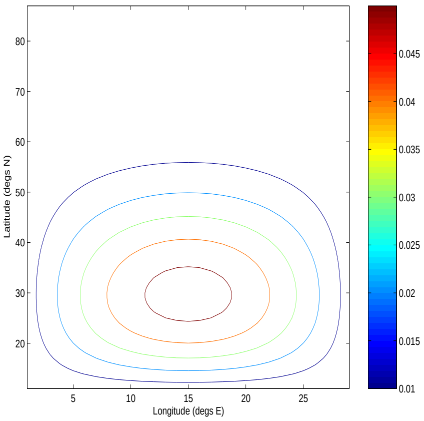

Figure B.35: Eigenvector associated with largest eigenvalue (0.9966) of G_{Block} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$. Nine Point Operator

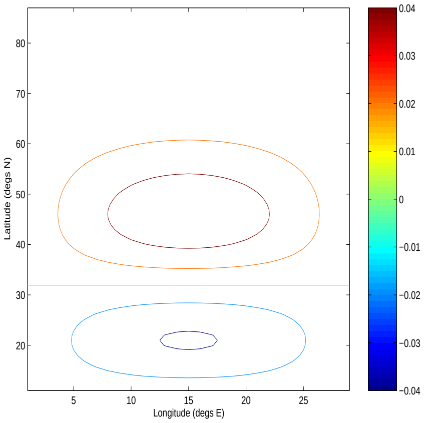

Figure B.36: Eigenvector associated with second largest eigenvalue (0.9935) of G_{Block} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$. Nine Point Operator

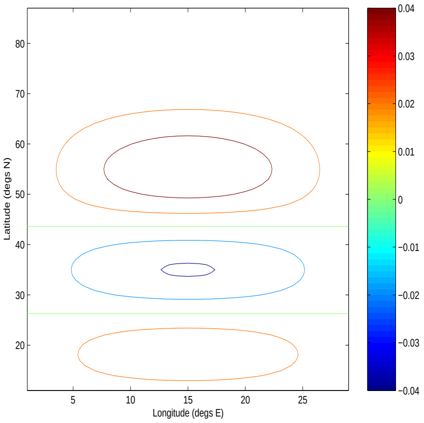

Figure B.37: Eigenvector associated with third largest eigenvalue (0.9929) of G_{Block} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$. Nine Point Operator

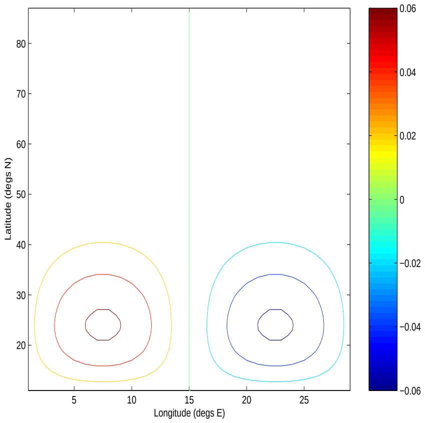

Figure B.38: Eigenvector associated with fourth largest eigenvalue (0.9900) of G_{Block} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$. Nine Point Operator

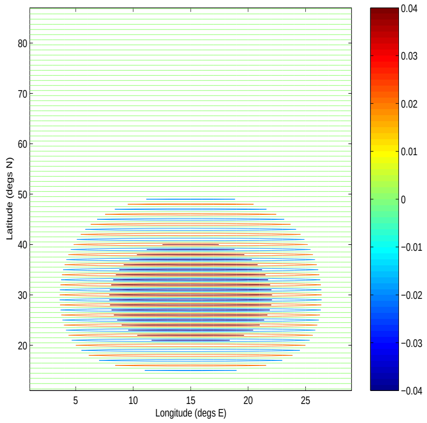

Figure B.39: Eigenvector associated with largest negative eigenvalue (-0.9966) of G_{Block} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$. Nine Point Operator

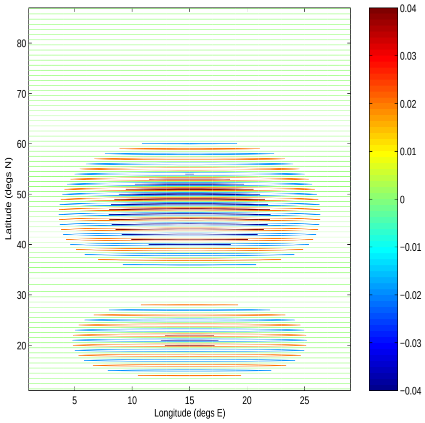

Figure B.40: Eigenvector associated with second largest negative eigenvalue (-0.9935) of G_{Block} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$. Nine Point Operator

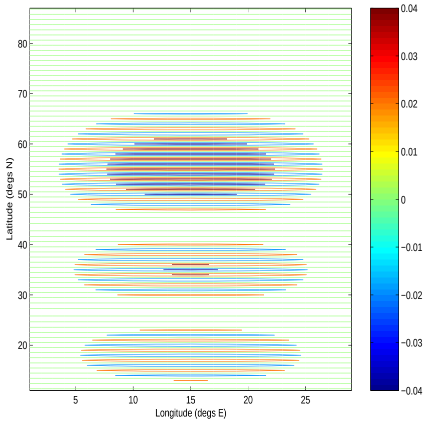

Figure B.41: Eigenvector associated with third largest negative eigenvalue (-0.9929) of G_{Block} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$. Nine Point Operator

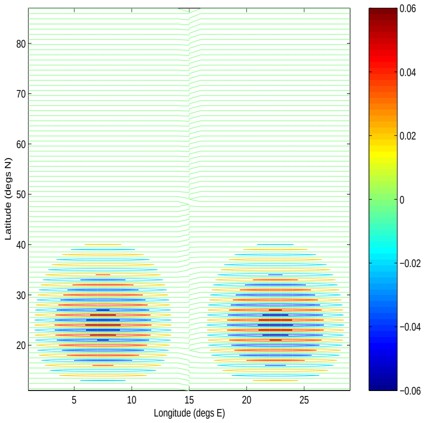

Figure B.42: Eigenvector associated with fourth largest negative eigenvalue (-0.9900) of G_{Block} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$. Nine Point Operator

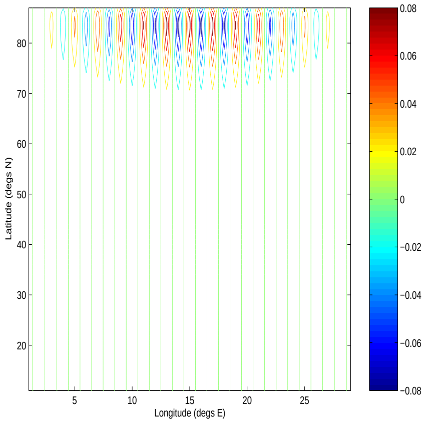

Figure B.43: Eigenvector associated with largest eigenvalue (-0.9515) of G_{ADI} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$. Nine Point Operator

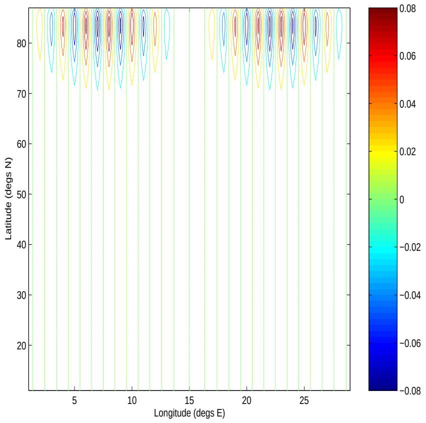

Figure B.44: Eigenvector associated with second largest eigenvalue (-0.9512) of G_{ADI} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$. Nine Point Operator

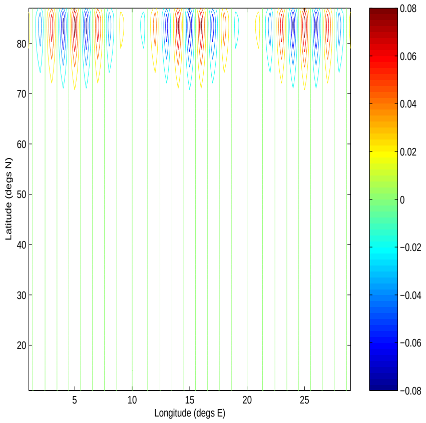

Figure B.45: Eigenvector associated with third largest eigenvalue (-0.9508) of G_{ADI} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$. Nine Point Operator

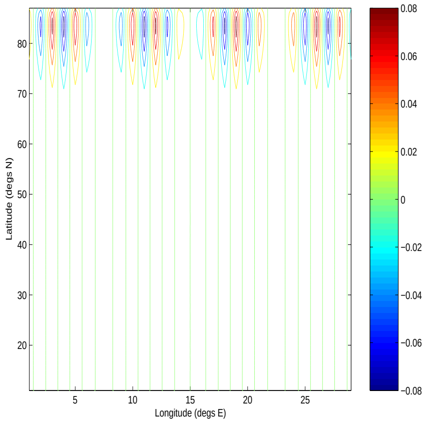

Figure B.46: Eigenvector associated with fourth largest eigenvalue (-0.9502) of G_{ADI} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$. Nine Point Operator

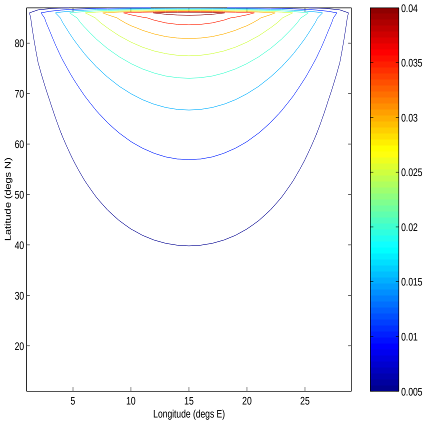

Figure B.47: Eigenvector associated with largest eigenvalue (0.9983) of G_{BIN} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

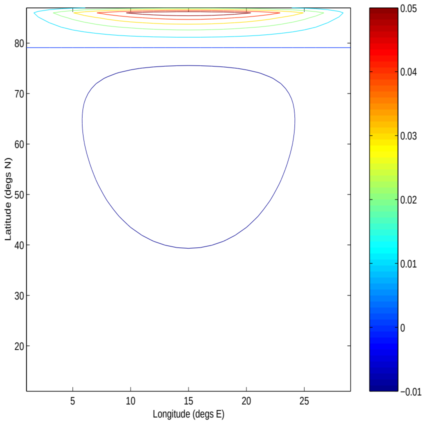

Figure B.48: Eigenvector associated with second largest eigenvalue (0.9972) of G_{BIN} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

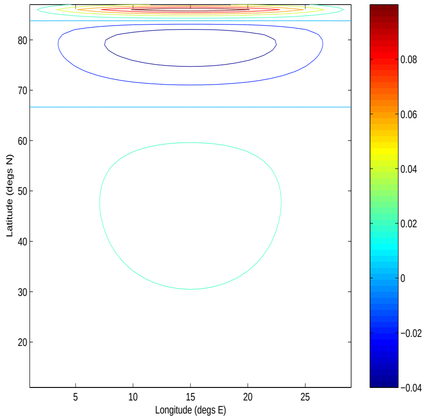

Figure B.49: Eigenvector associated with third largest eigenvalue (0.9971) of G_{BIN} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.

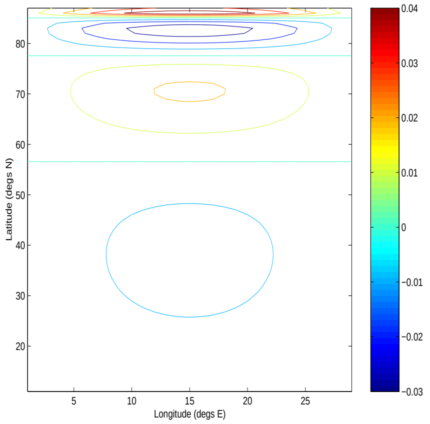

Figure B.50: Eigenvector associated with fourth largest eigenvalue (0.9969) of G_{BIN} for Limited Area Helmholtz problem. $\phi_{NB} = 88^o$.