An Approximate Projection Method Suitable for the Modeling of Rapidly Rotating Flows

by

Daniel Timothy Graves

B.S. (University of New Hampshire) 1989

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering-Mechanical Engineering

in the

GRADUATE DIVISION

of the

UNIVERSITY of CALIFORNIA, BERKELEY

Professor Phillip Colella, Chair
Professor Dorian Liepmann
Professor Paul Hilfinger

1996

The dissertation of Daniel Timothy Graves is approved:

Phillip Colella, Chair Date

Dorian Liepmann Date

Paul Hilfinger Date

University of California, Berkeley

1996

Abstract

Rapidly rotating flows can develop large-scale, persistent, coherent vortical structures that can develop over very long times compared with the rotational time scale. These structures are often perturbations of axisymmetric geostrophic shear flows and they exhibit distinguishing characteristics with respect to both the magnitude and the sign of the perturbational vorticity.

The structures that develop in rotating flow are localized in space compared to the very large domains involved in geophysical fluid dynamics. This implies that adaptive mesh refinement is the most efficient approach to modeling these phenomena. The two standard numerical formulations used to model geophysical fluid dynamics are not extendable to adaptive mesh refinement. Spectral methods have no way to add resolution to a particular region of space because a spectral calculation takes place in Fourier space. Finite-difference method based upon hydrostatic models are badly posed as initial-boundary value problems. Anelastic models treated using projection methods, on the other hand, are extendable to block-adaptive mesh refinement. Approximate projection methods have been shown to be the best behaved with adaptive mesh refinement. The fundamental goal of this work is to show that approximate projection methods can be used to calculate solutions in the regime of interest.

Among the achievements of this work is the development of an approximate projection algorithm suitable for the modeling of rapidly rotating flows in two space dimensions. The quasigeostrophic limit of the shallow-water equations is clarified. A projection algorithm for both the shallow-water and the quasigeostrophic equations is developed. The results of the algorithm compare favorably to the results of spectral calculations of vortical flows with background shear. The performance of the algorithm is evaluated according to several metrics appropriate for measuring how well the algorithm models the physics of ro-

tating flows. Convergence to steady-state solutions is evaluated and quantified. Finally the numerical convergence in the quasigeostrophic limit is evaluated. The difference between the numerical quasigeostrophic calculation and the numerical shallow-water calculation is evaluated to determine the extent to which the algorithm models the analytical convergence from the shallow-water equations to the quasigeostrophic equations.

Contents

List of Figures

List of Tables

Acknowledgements

I must first thank my family, who have given me continuous emotional support. Thanks also to my thesis advisor, Phil Colella, whose endless patience, compassion and humor has made graduate school a more bearable experience. Thanks to Dr. Paul Hilfinger, whose amazing editing work made this a far more readable manuscript. I especially want to thank my wife, Caryn. Words cannot express how much I cherish and depend upon her.

This work has been supported in part by the following two grants:

- "Adaptive Numerical Methods for Partial Differential Equations," US Department of Energy Mathematical, Computing, and Information Sciences Division, Grant DE-FG03-94ER25205.
- "Computational Fluid Dynamics and Combustion Dynamics," US Department of High-Performance Computing and Communications Grand Challenge Program, Grant DE-FG03-92ER25140.

Introduction

Rapidly rotating flows have unique physical effects that are of significant interest to geophysicists. These flows can develop large-scale, persistent, coherent vortical structures over very long times compared with the rotational time scale. These vortices are perturbations of geostrophic shear flows and they exhibit distinguishing characteristics with respect to both the magnitude and the sign of the perturbational vorticity. These structures have been observed both experimentally and numerically to be localized in space. The Great Red Spot of Jupiter is considered by many (including Marcus [Mar93]) to be a manifestation of this phenomenon. Gulf Stream eddies are also an example of these structures.

The physical characteristics of these phenomena dictate the design criteria for any numerical algorithm to be used to model them. For example, to be considered successful, any numerical algorithm used to model geophysical fluid dynamics must also represent accurately small perturbations of parallel shear flows, including the cumulative effect of body forces (from rotation) over long integration times.

The goal of the present investigation is to develop a finite-difference algorithm suitable for use in modeling these large-scale vortical structures. Because these phenomena are confined to small areas compared to the entire solution domain, the algorithm developed here must be one that can be extended to adaptive mesh refinement. The two standard numerical methods that are used for geophysical fluid dynamics are not extendable to adaptive mesh refinement. Spectral methods are not extendable to adaptive mesh refinement because there is no way to add resolution to a particular region of space in a calculation which takes place in Fourier space. Other numerical methods used in geophysical fluid dynamics are finite difference formulations based upon hydrostatic models. Oliger and Sundstrom [OS78] showed that this formulation of the equations of motion is badly posed for initial-boundary value problems. They showed that a modal expansion in the vertical direction shows that the number of boundary conditions required depends on the vertical wave number. For this reason, the equations are ill-posed with any pointwise, local specification of the boundary conditions. This enormously complicates the formulation of any adaptive mesh algorithm, since the coupling between meshes at coarse-fine boundaries cannot be locally specified.

The goal of the present work is to develop a finite-difference method suitable for use in computing large-scale vortical motion in geophysical flows that can also be combined with adaptive mesh refinement. The method is based on the higher-order projection method of Bell, Colella and Glaz [BCG89], an extension of Chorin's [Cho69] discrete projection method for incompressible flow. In this approach, the system of evolution equations and constraint equations are transformed using the the Hodge decomposition to a pure evolution formulation with an initial value constraint, for which it is straightforward to construct temporal discretizations. The projection formalism can also be applied to the more general constraints that can represent geophysical flows, i.e. anelastic models. Finally, a variation on the BCG algorithm has been been developed for adaptive mesh refinement by Almgren, et. al. [ABS96].

In this work, we will develop a version of the BCG algorithm suitable for computing these geophysical flows. We will develop this algorithm for a single (i.e. non-adaptive) grid, since many of the design issues can be investigated in that simpler setting. The context in which this algorithm will be developed is the rotating tank with variable depth problem. This is a good context for testing the algorithm because it is a simple physical model that contains all the relevant physics of large-scale, rotating vortical flows. Swinney [SHS93] in fact uses a rotating tank apparatus to model large-scale geophysical flows.

There are several issues to be resolved in the development of this algorithm. The first is that of long time integration with respect to rotational time scales. In the context of the rotating tank problem, the model should represent any axisymmetric, parallel shear flow as an exact discrete solution up to the accuracy of the linear system solvers.

The algorithm must also be able to represent the formation of coherent structures that are small perturbations of parallel shear flows. These coherent structures exhibit a distinguishing sign of perturbational vorticity. Persistent and stable steady-state solutions are obtained if the perturbation is of one sign and do not form if the same perturbation is of the opposite sign. These steady-state solutions can take many rotation times to evolve. Previous numerical solutions of this type using a spectral method have been studied and documented extensively by Marcus [Mar90]. Swinney, et. al. have also presented qualitative experimental data [MSS93]. Following the example of the previous numerical work, the test problems used for this issue of coherent structures will be shear layers and vortices embedded in a background shear. The solutions will be compared with the previous numerical work both in terms of qualitative behavior and convergence to steady states. Since vortices merge to create these coherent structures, some change of vorticity contour topology is involved. Physically this dissipation is accomplished by viscous effects. Numerical schemes have a variety of dissipation mechanisms. The present investigation will seek to quantify the effect of its dissipation mechanisms upon convergence to steady-state solutions. The test problem used for this portion of the investigation will be shear-layer perturbations of background shear flows.

In the previous test problems, the background shear drives the dynamics of the perturbational solutions. Another issue to be resolved is to quantify the ability of the algorithm to distinguish the effects of rotation. The test case that will be used to address this issue is the stability of shear layers in rotating flow. The stability of shear layers in rotating flow has been investigated both experimentally (by Swinney, et. al. [SHS93]) and numerically (by Lee [Lee94]). In non-rotating, inviscid incompressible flow, shear layers are unconditionally unstable. Rotation and viscosity, however, can act to stabilize the layer.

The present investigation will compare the previous numerical data with the growth rates recorded by the present algorithm.

A key issue that will be addressed in the present investigation is the relationship between the numerical solutions of the quasigeostrophic equations and the numerical solutions of the shallow-water equations. The quasigeostrophic equations describe the time evolution of vortical flows when rotation is large and vertical depth variations are small with the net effect non-vanishing. The shallow-water equations contain all the physics of the flow when the vertical length scales are small compared to the horizontal (i.e. no assumptions are made about the relative magnitude of the rotation). Analytically the quasigeostrophic equations are derived from the shallow-water equations when rotation is rapid and depth variations small. In geophysical fluid dynamics, the quasigeostrophic equations are used to describe large-scale motions in mid-latitudes. The shallow-water equations are used for global models with the intent that this limiting behavior is captured numerically. The present investigation provides a simple setting in which the extent to which this limiting behavior is actually captured can be quantified.

The first chapter discusses the relevant equations for this investigation. These equations are both the shallow-water equations and the quasigeostrophic equations. Classically, the quasigeostrophic equations are derived using asymptotics from the shallow-water equations in the limit of rapid rotation. Particular care in the present investigation is taken in deriving the quasigeostrophic equations. We do this by deriving equations for the vortical component of the vector field in the shallow-water equations by using projection formalism. At this point, asymptotics are used to simplify the resulting equations. This approach clarifies the limiting process by which solutions to the shallow-water equations converge to solutions of the quasigeostrophic equations. In this chapter the β-plane approximation is also analyzed in detail. Particular attention is given to the relationship of the depth of the fluid in the shallow-water case to the slope of the Coriolis parameter in the quasigeostrophic limit. This relationship is used in the last chapter to analyze the performance of the algorithm with respect to its convergence to the quasigeostrophic limit as rotation is increased in an appropriate fashion.

The second chapter describes the design of the numerical algorithm. Previous work in numerical methods for rotating flows is discussed along with previous work in projections methods. The design criteria for the algorithm are described in detail. These design criteria are roughly broken up into three categories. First are the issues of accuracy. The algorithm is designed to be second order in space and time. Second, the algorithm must reproduce exactly certain special exact solutions to the equations. Many of the problems here are small perturbations of axisymmetric shear flows and are integrated over long times. It is therefore necessary for the solution to reproduce as an exact discrete solution an unperturbed axisymmetric shear flow. Finally, the solution must also be able to integrate over long times without loss of stability. The chapter describes the actual discretization of the numerical method. Because of the design criteria, particular care is taken in the discretization of the forcing terms. Since an approximate projection is being used, forcing terms and pressure gradients must be balanced before being projected; otherwise, unperturbed axisymmetric shear flows will not be preserved exactly.

In the third chapter, the algorithm is validated with respect to the the design criteria in the previous chapter. Short-time convergence studies are performed to determine the order of accuracy of the method. The precise extent to which axisymmetric shear flows are preserved is measured. These tests are done both for the shallow-water and quasigeostrophic cases.

In the fourth chapter, the algorithm is run for several problems for which there exists substantial numerical and qualitative experimental data. We run the algorithm for several examples of flows that consist of small perturbations of background shear flows. The known behavior of these solutions is that they exhibit a distinguishing sign of perturbational vorticity and that under certain cases can form very robust steady-state solutions. The behavior of the algorithm in these cases is compared to the behavior of spectral method solutions and the qualitative experimental data. Stability of shear layers in rotating flows is analyzed and compared to the results of spectral calculations of viscous rotating flows.

In the final chapter, the algorithm is analyzed with respect to convergence issues. We analyze the steady-state solutions reached in terms of convergence with grid refinement. We discuss numerical conservation of circulation and kinetic energy. Finally we analyze the convergence of the algorithm with rapid rotation from the shallow-water case to the quasigeostrophic limit. We run a shallow-water problem with increasing rotation and the behavior with respect to the appropriate quasigeostrophic solution is reported.

Chapter 1

Preliminary Analysis

1.1 Model Problem

Geophysical fluid dynamics is fluid dynamics in a shallow, rotating system. The shallowness of the system leads to large-scale motions being two-dimensional by simple scaling arguments. This analysis is given by Pedlosky [Ped79]. When the rotation is sufficiently rapid, the Taylor-Proudman theorem also applies to restrict the motion of the fluid to be two-dimensional. It is therefore reasonable to use as a model problem a purely two-dimensional system.

Geophysical systems have a free surface. When considering large-scale vortical motions, however, surface waves carry too little energy and operate on far too short a time scale to affect the physics of said motions. Since it is large-scale vortical motions that are the subject of the current investigation, the model system shall therefore have a rigid lid. The rigid-lid approximation can be considered the limit in which surface waves equilibrate infinitely quickly with respect to the advective time scales of the system. See [Ped79] for a very detailed analysis.

The model problem being used here is an annulus with a variable depth. As

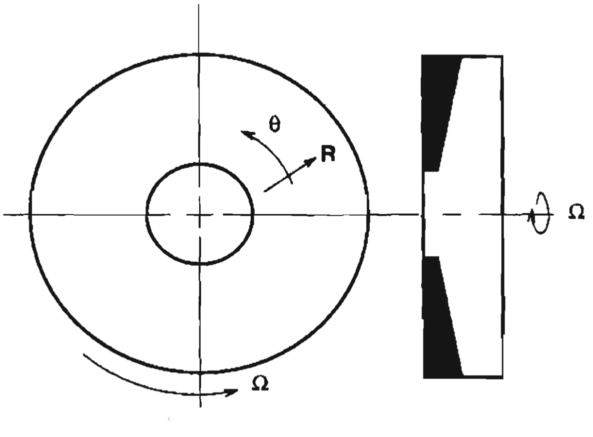


Figure 1.1: Model problem geometry.

illustrated in 1.1, the coordinate system is cylindrical:

$$\begin{aligned} \vec{u} &= (u_r, u_\theta) \\ \vec{x} &= (r, \theta) \\ r &\in [R_{in}, R_{out}] \\ \theta &\in [0, 2\pi] \end{aligned} \tag{1.1}$$

The model problem is inviscid with a rigid lid so the boundary conditions on the walls of the annulus are standard slip-wall conditions:

$$\begin{aligned} \vec{u}(r = R_{in}, \theta) \cdot \hat{r} &= 0 \\ \vec{u}(r = R_{out}, \theta) \cdot \hat{r} &= 0 \\ \frac{\partial \vec{u}(r=R_{in},\theta)}{\partial r} \times \hat{r} &= 0 \\ \frac{\partial \vec{u}(r=R_{out},\theta)}{\partial r} \times \hat{r} &= 0 \end{aligned} \tag{1.2}$$

The annulus rotates at a speed (Ω).

This model problem is chosen because it is the simplest model that contains most of the physics of geophysical fluid dynamics. Swinney, et. al. [MSS93] explain how the variable depth of a shallow water system approximates the physical effect of this changing of rotation with latitude. The mathematical extent of this approxmation will be quantified in the next section.

1.2 Governing Equations

The model system is shallow, has constant density, has a variable depth H, has a rigid lid, and is rotating with a speed Ω. The equations of motion in the model system are the following:

$$\begin{aligned} \frac{\partial \vec{u}}{\partial t} + (\vec{u}\cdot\nabla)\vec{u} &= -2\Omega \times \vec{u} - \nabla p + F_{geom}(\vec{u}, r) \\ F_{geom}(\vec{u}, r) &= (\frac{u_\theta^2}{r}, \frac{-u_\theta u_r}{r}) \\ \nabla \cdot (\vec{u}H) &= 0 \\ \vec{u}(\vec{x}, t=0) & \quad \text{given} \end{aligned} \tag{1.3}$$

Without loss of generality, the density in these equations has been set to unity. The geometry term (F_{geom}) accounts for terms in the momentum equation due to having a curvilinear coordinate system. In polar coordinates, the equations in component form are as follows:

$$\begin{aligned} \frac{\partial u_r}{\partial t} + u_r \frac{\partial u_r}{\partial r} + \frac{u_\theta}{r}\frac{\partial u_r}{\partial \theta} &= +2\Omega u_\theta - \frac{\partial p}{\partial r} + \frac{{u_\theta}^2}{r} \\ \frac{\partial u_\theta}{\partial t} + u_r \frac{\partial u_\theta}{\partial r} + \frac{u_\theta}{r}\frac{\partial u_\theta}{\partial \theta} &= -2\Omega u_r - \frac{1}{r}\frac{\partial p}{\partial \theta} - \frac{u_\theta u_r}{r} \\ \frac{1}{r}\frac{\partial (r u_r H)}{\partial r} + \frac{1}{r}\frac{\partial (u_\theta H)}{\partial \theta} &= 0 \end{aligned} \tag{1.4}$$

These are the Euler equations in two space dimensions with variable depth. The only assumption that goes into the derivation of these equations is that the vertical length-scale (L_V) is much smaller than the horizontal (L_H)

$$L_H \gg L_V \tag{1.5}$$

Without this assumption, the two-dimensionality of the system is no longer assured. The system of equations 1.3 shall be referred to here as the shallow-water equations, although that terminology is often used for free-surface problems. In the present case, the free surface effects are suppressed by the rigid lid.

1.2.1 Projection Formulation

In order to proceed further, it is necessary to introduce here some of the notation of projection formalism, a systematic mathematical approach to extracting the divergence-free component of a vector field. For clarity, let us express divergence, gradient and Laplacian as $\mathcal{D}, \mathcal{G}, \mathcal{L}$:

$$\begin{aligned} \mathcal{D}\vec{u} &\equiv \nabla \cdot \vec{u} \\ \mathcal{G}\phi &\equiv \nabla\phi \\ \mathcal{L}\phi &\equiv \nabla \cdot (\nabla\phi) \end{aligned} \tag{1.6}$$

Since the shallow-water equation 1.3 velocity is constrained by a depth-weighted divergence,

$$\mathcal{D}(H\vec{u}) = 0 \tag{1.7}$$

define depth-weighed divergence and Laplacian $\mathcal{D}_{\mathcal{H}}, \mathcal{L}_{\mathcal{H}}$ as follows:

$$\begin{aligned} \mathcal{D}_{\mathcal{H}}(\vec{u}) &\equiv \mathcal{D}(H\vec{u}) \\ \mathcal{L}_{\mathcal{H}}(\phi) &\equiv \mathcal{D}_{\mathcal{H}}\mathcal{G}(\phi) \end{aligned} \tag{1.8}$$

Given a vector field $\vec{W}$ on a bounded domain Λ where

$$\int_{\partial\Lambda} \vec{W} \cdot \hat{n} dA = 0 \tag{1.9}$$

$\vec{W}$ can be described as a sum of a divergence- free (with respect to a depth H) component field $\vec{W}_d$ and a pure gradient field $\mathcal{G}\phi$ [Cho69].

$$\begin{aligned} \vec{W} &= \vec{W}_d + \mathcal{G}\phi \\ \mathcal{D}_{\mathcal{H}}W_d &= 0 \quad \text{on} \quad \Lambda \end{aligned} \tag{1.10}$$

where

$$\begin{aligned} W_d \cdot \hat{n} &= 0 \quad \text{on} \quad \partial\Lambda \\ \mathcal{G}\phi \cdot \hat{n} &= \vec{W} \cdot \hat{n} \quad \text{on} \quad \partial\Lambda \end{aligned} \tag{1.11}$$

This is a general form of the Hodge decomposition. Given a vector field $\vec{U}$, to extract the depth-weighted divergence-free component of $\vec{U}$ (called $\vec{u}_d$), first expand $\vec{U}$ using the Hodge decomposition (equation 1.10):

$$\begin{aligned} \vec{U} &= \vec{u}_d + \mathcal{G}\phi \\ \mathcal{D}_{\mathcal{H}}\vec{u}_d &= 0 \end{aligned} \tag{1.12}$$

Taking the depth-weighted divergence of equation 1.12:

$$\mathcal{D}_{\mathcal{H}}\vec{U} = \mathcal{D}_{\mathcal{H}}\vec{W}_d + \mathcal{L}_{\mathcal{H}}\phi \tag{1.13}$$

and since

$$\mathcal{D}_{\mathcal{H}}\vec{u}_d = 0 \tag{1.14}$$

it is clear that to solve for the depth-weighted divergence-free component of a vector field $\vec{U}$ on a a domain Λ, one simply solves the following:

$$\begin{gathered} \mathcal{L}_{\mathcal{H}}\phi = \mathcal{D}_{\mathcal{H}}\vec{U} \\ \tfrac{\partial\phi}{\partial n} = \vec{U}\cdot\hat{n} \quad on \quad \partial\Lambda \\ \vec{u}_d = \vec{U} - \mathcal{G}\phi \end{gathered} \tag{1.15}$$

It is convenient to express this process in operator form. The depth-weighted projection operator is defined as that operator that extracts the depth-weighted divergence-free part from a vector field:

$$\begin{gathered} \mathcal{P}_{\mathcal{H}}\vec{U} = \vec{u}_d \\ \mathcal{P}_{\mathcal{H}} \equiv (\mathcal{I} - \mathcal{G}(\mathcal{L}_{\mathcal{H}})^{-1}\mathcal{D}_{\mathcal{H}}) \end{gathered} \tag{1.16}$$

The projection of any gradient field therefore vanishes. We can also define the operator $\mathcal{Q}_{\mathcal{H}}$, which extracts the gradient portion of a vector field:

$$\begin{gathered} \mathcal{Q}_{\mathcal{H}}\vec{U} = \nabla\phi \\ \mathcal{Q}_{\mathcal{H}} \equiv \mathcal{I} - \mathcal{P}_{\mathcal{H}} \end{gathered} \tag{1.17}$$

A special case of the projection operators $\mathcal{P}_{\mathcal{H}}$ and $\mathcal{Q}_{\mathcal{H}}$ are the more common projection operators to extract the divergence-free (in the ordinary sense) component of the vector field and the corresponding gradient. These operators are found by simply setting the depth to unity.

$$\mathcal{P}_{\mathcal{O}}, \mathcal{Q}_{\mathcal{O}} = \mathcal{P}_{\mathcal{H}}, \mathcal{Q}_{\mathcal{H}}, H \equiv 1 \tag{1.18}$$

A velocity field $\vec{u}$ is said to be $\mathcal{D}$-divergence-free if $\mathcal{D}\vec{u} = 0$; likewise a velocity field that is said to be $\mathcal{D}_{\mathcal{H}}$-divergence-free if $\mathcal{D}_{\mathcal{H}}\vec{u} = 0$.

Partial differential equations whose solutions are constrained by a divergence-free condition can be reformulated with a projection operator to eliminate the constraint. For example, the shallow-water equations (1.3) are equivalent to

$$\frac{\partial\vec{u}}{\partial t} = \mathcal{P}_{\mathcal{H}}(-(\vec{u}\cdot\nabla)\vec{u} = -2\Omega\times\vec{u} + F_{geom}(\vec{u}, r)) \tag{1.19}$$

$$\mathcal{P}_{\mathcal{H}}(\vec{u}(\cdot,t=0)) = u(\cdot,t=0) \tag{1.20}$$

The reason for this equivalence is that if $\vec{u}$ satisfies equation 1.19 then

$$\frac{\partial \mathcal{D}_{\mathcal{H}}\vec{u}}{\partial t} = 0 \tag{1.21}$$

which implies that

$$\mathcal{D}_{\mathcal{H}}\vec{u}(\cdot,t) = \mathcal{D}_{\mathcal{H}}\vec{u}(\cdot,0) = 0 \tag{1.22}$$

by equation 1.20. Since $\mathcal{D}_{\mathcal{H}}\vec{u} = 0$ for all time by equation 1.22, we have eliminated the $\mathcal{D}_{\mathcal{H}}$-divergence-free constraint from the shallow-water equations by using the projection formulation (equation 1.19). The pressure gradient is no longer necessary as the projection of any pure gradient is zero. Similarly any equation of the form

$$\begin{aligned} \tfrac{\partial \vec{u}}{\partial t} + (\vec{u}\cdot\nabla)\vec{u} &= F - \nabla p \\ \mathcal{D}u &= 0 \end{aligned} \tag{1.23}$$

can be written

$$\begin{aligned} \tfrac{\partial \vec{u}}{\partial t} &= \mathcal{P}_{\mathcal{O}}(-(\vec{u}\cdot\nabla)\vec{u} + F) \\ \mathcal{P}_{\mathcal{O}}(\vec{u}(\cdot,t=0)) &= u(\cdot,t=0) \end{aligned} \tag{1.24}$$

This is the projection formulation of the equations of motion for inviscid, incompressible ($\mathcal{D}$-divergence-free) flow with forcing. The formulation of the numerical algorithm described in Chapter 2 is based upon the projection formulation of the equations of motion.

1.2.2 Geostrophic Degeneracy

When (ΩL_H) is sufficiently large compared to the velocity, a simple scaling analysis shows that the equations of motion reduce to the statement that the pressure forces and the Coriolis forces balance:

$$2\Omega \times \vec{u} = \nabla p \tag{1.25}$$

This is the equation for geostrophic degeneracy. An asymptotic analysis is presented in later sections. Equation 1.25 is a constraint equation for geostrophic flow. To clarify the meaning of equation 1.25, we now take its curl to eliminate the pressure gradient. When this is done,

$$\nabla \times (2\Omega \times \vec{u}) = \nabla \times (\nabla p) = 0 \tag{1.26}$$

is reduced to

$$\nabla \cdot \vec{u} = 0 \tag{1.27}$$

This means that the velocity field in the limit of rapid rotation is divergence-free in the classical sense. Equation 1.25 is referred to as geostrophic "degeneracy" because this constraint on the velocity field is insufficient to determine the dynamics of the system.

1.2.3 Decomposition of the Velocity Field

In this section and the next, the system known as the quasigeostrophic equations will be derived. These are time-dependent, predictive equations for the divergence-free component of the velocity field. In the classical analysis at this point, a time-dependent, closed system of equations are derived for the vorticity $\nabla \times \vec{u}$ in a suitable limit of rapid rotation (Pedlosky [Ped79] presents this analysis). This predictive system is referred to as the quasigeostrophic equations. The present approach is to find the evolution equations for the $\mathcal{D}$-divergence-free portion of the velocity field satisfying the shallow-water equations. Once this predictive system of equations is suitably described, the appropriate asymptotics will be used to simplify the system and produce the quasigeostrophic equations in terms of primitive variables. The advantage of this approach over the classical approach is first that the equations for the $\mathcal{D}$-divergence-free velocity field (or, equivalently, the geostrophic velocity field) are obtained directly rather than through the vorticity. This approach also makes explicit the identification of a family of shallow-water equation models that correspond to each quasigeostrophic model.

In this section, the equations for time-dependent vortical motion are derived without any assumptions about the relative magnitude of the two components of the velocity field (potential flow and divergence-free components) and both components are included in the system of equations. The Hodge decomposition is used to divide the velocity field into two components, one a potential flow and the other $\mathcal{D}$-divergence-free:

$$\vec{u} = \vec{u}_o + \vec{u}_p \tag{1.28}$$

where

$$\begin{aligned} \nabla \cdot \vec{u}_o &= 0 \\ \vec{u}_p &= \nabla \phi \end{aligned} \tag{1.29}$$

By putting the equations in terms of the vorticity, classical analysis here makes an implicit assumption that the velocity field is uniquely determined by the vorticity. Because our problem domain is not simply connected, the $\mathcal{D}$-divergence-free component of the velocity field is not uniquely determined by the vorticity. In particular, one can decompose $\vec{u}_o$ further into its component derivable from the vorticity ω and its component due to circulation Γ at the inner boundary.

$$\vec{u}_o = -\nabla \times (\mathcal{L}^{-1}\omega) + \frac{\Gamma}{2\pi r}\hat{\theta} \tag{1.30}$$

The circulation at the inner radius Γ is calculated as follows:

$$\Gamma = \frac{1}{2\pi}\int_0^{2\pi} \vec{u}(r = R_{in}) \cdot \hat{t} d\theta \tag{1.31}$$

where $\hat{t}$ is the unit vector in the azimuthal direction. By Kelvin's theorem, for both the inviscid models we are considering, the circulation is constant in time [Ped79].

$$\frac{\partial \Gamma}{\partial t} = 0 \tag{1.32}$$

In the analysis that follows, we will assume that $\Gamma(t) = \Gamma(0) = 0$, which means that the $\mathcal{D}$-divergence-free component of the flow is uniquely determined by the vorticity via equation 1.30.

Now putting the decomposition (equation 1.28) into the shallow-water equations (1.3) and applying the projection $\mathcal{P}_{\mathcal{O}}$ to the acceleration terms, the result is the following:

$$\begin{gathered} \tfrac{\partial \vec{u}_o}{\partial t} = \mathcal{P}_{\mathcal{O}}(-\vec{u} \cdot \nabla \vec{u} - 2\Omega \times \vec{u} + F_{geom}(\vec{u}, r)) \\ \mathcal{D}_{\mathcal{H}}(\vec{u}_o + \vec{u}_p) = 0 \end{gathered} \tag{1.33}$$

Since u_o is divergence-free (see equation 1.29) and has no circulation at the inner boundary, it follows that

$$\begin{gathered} \mathcal{P}_{\mathcal{O}}(2\Omega \times \vec{u}_o) = 0 \\ \mathcal{P}_{\mathcal{O}}(2\Omega \times \vec{u}_p) = 2\Omega \times \vec{u}_p \end{gathered} \tag{1.34}$$

This simplifies the system of equations 1.33 to the following:

$$\frac{\partial \vec{u}_o}{\partial t} = \mathcal{P}_{\mathcal{O}}(-(\vec{u} \cdot \nabla)\vec{u} - 2\Omega \times \vec{u}_p + F_{geom}(\vec{u}, r)) \tag{1.35}$$

$$\mathcal{D}_{\mathcal{H}}(\vec{u}_o + \vec{u}_p) = 0 \tag{1.36}$$

These variables are dimensional and no assumptions have been made about the relative size of the components of the velocity field. Equation 1.28 is simply a restatement of the Hodge decomposition. Note that the Coriolis force depends only upon the potential flow velocity component $\vec{u}_p$ appears in the momentum equation is in the Coriolis force term. This implies that any assumption about the maginitude of the potential flow velocity relative to $\vec{u}_o$ corresponds to an approximation of the Coriolis force.

Now we can use equations 1.29 and 1.36 to express the potential flow component $\vec{u}_p$ in terms of $\vec{u}_o$:

$$\vec{u}_p = -\vec{u}_o \cdot \mathcal{G}(H) \tag{1.37}$$

and recalling that $\vec{u} = \mathcal{G}\phi$ we can rewrite all the relevant equations as a closed system:

$$\begin{gathered} \tfrac{\partial \vec{u}_o}{\partial t} = \mathcal{P}_{\mathcal{O}}(-(\vec{u} \cdot \nabla)\vec{u} + F_{rot} + F_{geom}(\vec{u}, r)) \\ F_{rot} = 2\Omega \times \mathcal{G}\phi \\ \phi = \mathcal{L}_{\mathcal{H}}{}^{-1}(-\vec{u}_o \cdot \mathcal{G}(H)) \end{gathered} \tag{1.38}$$

Equation 1.38 constitutes a complete system of equations, which is a generalized form of the quasigeostrophic equations in the sense that they are predictive for $\vec{u}_o$ and in the sense that the potential flow $\vec{u}_p$ need not be specified as initial data. The system 1.38 differs from the standard quasigeostrophic equations in that equation 1.25 is not satisfied. It is important to emphasize that no approximations have yet been made about the relative size of the two components of the velocity field. These equations are still completely appropriate regardless of the magnitude of rotational forcing. In the next section, these equations will be simplified to produce a more standard form of the quasigeostrophic equations using asymptotics. The momentum equation will only change in the forcing term F_{rot}. This implies that any assumption about the relative size of $\vec{u}_o$ and $\vec{u}_p$ is equivalent to an assumption about the composition of F_{rot}.

1.2.4 The Quasigeostrophic Equations

The Rossby number Ro

$$Ro = \frac{U}{2\Omega L} \tag{1.39}$$

is a ratio of the relative velocity to the velocity of the rotating frame. In large scale geophysical flows in midlatitudes, the Rossby number is small ($\approx$ 0.1, see Gill, et. al. [Gil82] for more details).

Define F_{qg} to be the rotational force in the limit of small Rossby number. In light of the discussion of geostrophic degeneracy, it can be expected that, for rapidly rotating flow (i.e., small Rossby number) the divergence-free (geostrophic) part of the velocity field ($\vec{u}_o$) will be much larger than the potential flow ($\vec{u}_p$). Given that both U and L are O(1), this assumption is quantified as follows.

$$\frac{u_p}{u_o} = O(Ro) \tag{1.40}$$

The second assumption that must be made is that the gradient of the depth of the fluid will not be so large as to invalidate the length-scale assumptions implicit in scaling derivatives (which include depth gradients).

$$\frac{L}{H}\nabla H = O(Ro) \tag{1.41}$$

When these assumptions are applied to the constraint equation 1.36, it follows that

$$\mathcal{D}(\vec{u}_p) = -\frac{1}{H}\vec{u}_o \cdot \mathcal{G}(H) \tag{1.42}$$

Again, since $\vec{u}_p = \mathcal{G}\phi$,

$$\phi = -\mathcal{L}^{-1}(\frac{1}{H}\vec{u}_o \cdot \mathcal{G}(H)) \tag{1.43}$$

Using equation 1.43, the rotational forcing term F_{qg} can be put explicitly purely in terms of the geostrophic velocity $\vec{u}_o$.

$$F_{qg} = 2\Omega \times \mathcal{G}(\mathcal{L}^{-1}(\frac{1}{H}\vec{u}_o \cdot \mathcal{G}(H))) \tag{1.44}$$

Equation 1.44, and the momentum equation

$$\frac{\partial \vec{u}_o}{\partial t} = \mathcal{P}_{\mathcal{O}}(-(\vec{u}_o \cdot \nabla)\vec{u}_o + F_{qg} + F_{geom}(\vec{u}_o, r)) \tag{1.45}$$

where we have kept only the convective acceleration terms which are of leading order in the Rossby number, are a more standard form of the quasigeostrophic equations. These are predictive equations for u_o that do not explicitly include the potential-flow term. These equations do assume a small Rossby number. Formally, the quasigeostrophic equations are appropriate in that limit of rapid rotation and vanishing potential flow where their product is finite, or symbolically

$$\begin{aligned} F_{qg} &= \lim_{\frac{\Omega L}{U_o}\to\infty, \frac{\vec{u}_p}{U_o}\to 0} (2\Omega \times \vec{u}_p) \\ &= \lim_{\frac{\Omega L}{U_o}\to\infty, \frac{\vec{u}_p}{U_o}\to 0} F_{rot} \end{aligned} \tag{1.46}$$

where F_{qg} is finite and U_o is the velocity scale of $\vec{u}_o$.

Though these (equations 1.44 and 1.45) are called the quasigeostrophic equations it is important to emphasize that that solutions to 1.35 are *exactly* geostrophic (i.e. $\mathcal{D}$-divergence-free) but are found by using a forcing term that emerges from a secondary flow that is *not* geostrophic.

1.2.5 β-Plane Approximation

Now suppose that the depth is only a function of radius, and specifically is of the form

$$\begin{array}{c} H = H(r) \\ \frac{1}{H}\mathcal{G}H = \frac{b}{L}Ro \end{array} \qquad (1.47)$$

where b is a constant. Recalling the definition of Rossby number ($Ro = \frac{U}{2\Omega L}$), equation 1.47 becomes

$$H = H_o \exp(\frac{bUr}{2L^2\Omega}) \qquad (1.48)$$

where H_o is also a constant. With this form of depth, equation 1.42 becomes

$$\mathcal{D}\vec{u}_p = \frac{-1}{H}(\vec{u}_o \cdot GH) = \frac{-\beta u_{o,r}}{\Omega} \qquad (1.49)$$

where $\beta = \frac{2bU}{L^2}$.

From equation 1.46, the form of the rotational forcing is

$$F_{qg} = -2\Omega \times \vec{u}_p \qquad (1.50)$$

Suppose we take the curl of F_{qg}.

$$\begin{aligned} \nabla \times F_{qg} &= \nabla \times (-2\Omega \times \vec{u}_p) \\ &= -2\Omega\mathcal{D}\vec{u}_p \\ &= 2\beta u_{o,r} \end{aligned} \qquad (1.51)$$

Now suppose we have a vector function $\vec{F}_a$ of the form

$$\vec{F}_a = (\beta r\hat{k}) \times \vec{u}_o \qquad (1.52)$$

The curl of this function is

$$\nabla \times \vec{F_a} = \beta u_{o,r} + (\beta r)(\frac{\partial r u_{o,r}}{\partial r} + \frac{\partial u_{o,\theta}}{\partial \theta}) \tag{1.53}$$

which is equivalent to

$$\nabla \times \vec{F_a} = \beta u_{o,r} + \beta r^2 \mathcal{D} \vec{u}_o, \tag{1.54}$$

and since $\vec{u}_o$ is $\mathcal{D}$-divergence-free,

$$\nabla \times \vec{F_a} = \beta u_{o,r} \tag{1.55}$$

So it has been shown that

$$\nabla \times \vec{F_a} = \nabla \times \vec{F}_{qg} \tag{1.56}$$

Therefore $\mathcal{P}_{\mathcal{O}}(F_a) = \mathcal{P}_{\mathcal{O}}(F_q g)$ if the circulation at the inner radius of both of these fields vanishes. The circulation of F_{qg} is given by

$$\begin{aligned} \frac{1}{2\pi} \int_0^{2\pi} F_{qg}(r = R_{in}) \cdot \hat{t} d\theta &= \frac{1}{2\pi} \int_0^{2\pi} (\beta R_{in} \hat{k}) \times \vec{u}_o(R_{in}) d\theta \\ \frac{1}{2\pi} \int_0^{2\pi} F_{qg}(r = R_{in}) \cdot \hat{t} d\theta &= \frac{\beta R_{in}}{2\pi} \int_0^{2\pi} (\vec{u}_o(R_{in}) \cdot \hat{n}) d\theta \end{aligned} \tag{1.57}$$

and the circulation of F_a is given by

$$\begin{aligned} \frac{1}{2\pi} \int_0^{2\pi} F_a(r = R_{in}) \cdot \hat{t} d\theta &= \frac{1}{2\pi} \int_0^{2\pi} (2\Omega \hat{k}) \times \vec{u}_p(R_{in}) d\theta \\ \frac{1}{2\pi} \int_0^{2\pi} F_a(r = R_{in}) \cdot \hat{t} d\theta &= \frac{2\Omega R_{in}}{2\pi} \int_0^{2\pi} (\vec{u}_p(R_{in}) \cdot \hat{n}) d\theta \end{aligned} \tag{1.58}$$

The boundary conditions for both components is a no-normal-flow condition:

$$\begin{aligned} \vec{u}_o \cdot \hat{n} &= 0 \\ \vec{u}_p \cdot \hat{n} &= 0 \end{aligned} \tag{1.59}$$

so the integrands in both equation 1.57 and equation 1.58 vanish and the circulation of both F_a and F_{qg} also vanish. Therefore it has been shown that

$$\mathcal{P}_{\mathcal{O}}(F_{qg}) = \mathcal{P}_{\mathcal{O}}((\beta r \hat{k}) \times \vec{u}_o) \tag{1.60}$$

for the case of the depth being given by equation 1.48.

Often when geophysical flows are modeled using the quasigeostrophic equations, the effect of rotation changing with latitude is modeled with a β-plane approximation. The *Beta*-plane approximation makes the rotational force linear in r:

$$F_{qg}{}^{\beta-plane} = (2\beta r\hat{k}) \times \vec{u} \tag{1.61}$$

This expansion arises from two considerations. First the effective rotation force varies smoothly with latitude ψ

$$\Omega_{eff} = \Omega_0 + \Omega_e \sin\psi \tag{1.62}$$

where ψ is the angle of latitude. Second, since the velocity is divergence-free (and two dimensional), any constant crossed with the velocity is a pure gradient and does not affect the physics of the flow. The radius r used in equation 1.61 is the radius of a polar coordinate system whose inner and outer radii are lines of latitude. When equation 1.62 is expanded in ψ about a point in midlatitudes ψ_o using a Taylor series, the result is equation 1.61, where $\beta = \Omega_e \cos(\psi_o)$.

As it will be necessary to compare solutions of the shallow-water system to solutions of the quasigeostrophic system, it must be determined what topography of depth (H) is necessary to produce this "β-effect." But equation 1.60 shows that the projection of the Coriolis forcing term with the depth given by equation 1.48 is precisely equal to the projection of the Coriolis forcing term given in equation 1.61. Therefore, we have already solved for the appropriate depth function,

$$H_{\beta-Plane} = \exp(\frac{\beta r}{\Omega}) \tag{1.63}$$

So solving the shallow-water equations with a rigid lid, a bottom given by equation 1.63, and small Rossby number is equivalent to solving the quasigeostrophic equations with a β-plane for rotational forcing. This equivalence will be explored in later sections to investigate the convergence of the shallow-water numerical scheme with small Rossby number to see if the scheme converges to the quasigeostrophic numerical solution. This an explicit example of the rotational forcing in the limit of rapid rotation and diminishing potential flow, converging to a finite quantity. Symbolically, this is written

$$F_{qg} = \lim_{\frac{\Omega L}{U_o} \to \infty, \frac{\vec{u}_p}{U_o} \to 0} (-2\Omega \times \vec{u}_p) \tag{1.64}$$

or in this specific case,

$$(2\beta r\hat{k}) \times \vec{u}_o = \lim_{\frac{\Omega L}{U_o} \to \infty, \frac{\vec{u}_p}{U_o} \to 0} (-2\Omega \times \vec{u}_p) \tag{1.65}$$

where the depth H is given by equation 1.48.

Chapter 2

Numerical Approach

2.1 Review of Previous Work

2.1.1 Numerically Modeling Rapidly Rotating Flows

The algorithms that have been used for modeling rotating flows can be roughly divided into three categories: spectral methods, particle methods, and finite-difference methods.

Spectral methods solve partial differential equations by expanding the solution into a finite number of eigenmodes and solving for the coefficients of the expansion. Hockney [Hoc65] [Hoc71] showed that spectral methods can be used effectively in solving non-rotating flows [Roa72]. Orzag [Orz70] developed transform methods that made spectral methods feasible for large calculations. Since then, spectral methods have been used extensively for both atmospheric and ocean modeling. The NCAR Community Climate Model (CCM), a very elaborate global circulation model, models atmospheric flows using a spectral transform technique. For a more comprehensive review of the extensive history of spectral methods in atmospheric and ocean modeling, see Washington and Parkinson [WP86].

Closer to the class of calculations done in the present work, Marcus [Mar93] developed a pseudo-spectral initial-value method to calculate time-dependent viscous solutions for Taylor-Couette flow (flow between rotating cylinders). The Navier-Stokes equations are solved in a rotating frame for a perturbational velocity the primary Couette-flow (very low Reynolds number) solution. The equations are spatially discretized by expanding the solution as a sum of Fourier-Chebyshev modes and temporally discretized using a fractional-step algorithm in which the advection, pressure, and viscous terms were treated separately. This method is found to exhibit large time-splitting errors near the radial boundaries. Since this is a viscous calculation, artificial numerical dissipation was not an issue. Coughlin, et. al. [CM92] use a similar method to study temporally quasi-periodic modes in Taylor-Couette flow. The algorithm is improved in efficiency by introducing shift-and-reflect symmetry to reduce the number of Fourier modes needed to fully describe the solution by half. Marcus [Mar90] extended this algorithm to calculate inviscid, quasi-geostrophic flow in a β-plane. De-aliasing was added to improve long-time performance. Since this was an inviscid calculation, the problem of numerical dissipation had to be addressed. This is done in two ways. In the shorter-time calculation, the calculation was simply stopped when enough of the numerical energy built up in the highest resolved wavenumbers that it started to flow back into smaller wavenumbers. In the longer-term runs, energy was simply removed from the highest wavenumbers to allow the calculation to proceed. Lee [Lee94] also uses this algorithm to model quasi-geostrophic flow in a β-plane. In this work the severe time-splitting error mentioned previously is reduced by making the viscous step of the time integration implicit. The advection terms are calculated using Adams-Bashforth to obtain second-order temporal accuracy.

Particle methods are a class of methods in which the vorticity is broken up into discrete blobs and the solution is calculated by computing the interaction between these chunks. For a comprehensive overview of vortex methods, see Puckett [Puc93]. Modern vortex methods were first developed by Chorin [Cho73]. These methods are typically $O(N^2)$

in operation count, where N is the number of discrete vortex elements. Fast vortex methods use the fact that the effect of vortices decreases rapidly with distance. An early version of fast vortex methods is the particle-in-cell method developed by Hockney and Eastwood [HE81], in which the N^2 interactions are replaced by a finite-difference solution to a Poisson equation. In particle-particle, particle-mesh methods, the vortices close to one another interact with a Biot-Savart potential and the far-field effects are taken into account using a finite-difference solution to a Poisson equation. An elegant version of this approach was developed by Anderson [And86] for vortex dynamics in two spatial dimensions and extended by Almgren, Buttke and Colella [ABC94] to three dimensions. Both particle-in-cell methods and particle-particle, particle-mesh methods are $O(Nlog(N))$ in operation count. Multipole methods, developed by Greengard and Rokhlin [GR87], are another class of fast particle methods. Dahleh [Dah93] developed a fast particle method for solving the quasigeostrophic vorticity equation. The method is $O(N)$ and shows very good agreement with the $O(N^2)$ formulation.

In finite-difference methods, space is divided into an Eulerian grid and divided-difference operators are used to approximate derivatives. There is an extensive amount of finite-difference technology. There are pressure-Poisson methods, in which the equation generated by taking the divergence of the momentum equation is solved for pressure. There are hydrostatic methods, in which the the vertical momentum equation is replaced by a hydrostatic condition. There are also projection methods, which will be described extensively later. The applications of finite-difference methods to rotating flows can be roughly divided into engineering applications and atmospheric and ocean applications.

There are many examples of finite-difference simulations of rotating flow not related to ocean modeling. Sørensen et. al. [SC95] developed a finite-difference algorithm to model viscous, axisymmetric rotating fluid flow in a closed cylinder with a spinning lid.

The equations are formulated in terms of vorticity, streamfunction, and circulation and are discretized with finite-difference approximations. The discretized equations solved with a modified ADI technique. Piomelli et. al. [PL95] and Cambon et. al. [CBSJ94] independently adapted large-eddy simulation algorithms for use in modeling rotating flows. The flow variables are decomposed into a large scale and a subgrid scale component. Piomelli et. al. parameterize turbulent stresses using an eddy-viscosity model. Cambon et. al. model subgrid turbulent stresses with a dynamic model based on splitting the velocity gradient into symmetric and skew-symmetric parts and putting a model for enstrophy generation directly into the pressure solve.

Many of the early incompressible, rotating flow numerical simulations were done in the context of computing ocean circulations. Bryan [Bry63], solved a set a of vertically integrated equations to examine the formation of Gulf Stream-like phenomena for various Reynolds numbers and various degrees of nonlinearity in the equations. Cox [Cox75] captured many features of ocean circulation in his multiple layer model. Semtner, et. al. [AJSM77] used a similar model to model Gulf Stream circulation. Their model is also a hydrostatic, multiple layer model which includes effects of temperature. This work studies extensively the effect of the ad-hoc dissipation (diffusive terms added to the momentum equation to keep ocean models stable) upon the ability of the model to capture boundary current effects. More recently, Semtner and Chervin [AJSC92] have used this type of hydrostatic model to compute detailed global ocean circulation. This work indicates a complex network of localized currents that provides a striking example in geophysical fluid dynamics in which adaptive mesh methods could be useful. Oliger and Sundstrom [OS78], however, showed that this formulation of the equations of motion is badly-posed for initial-boundary value problems. They showed that the number of boundary conditions required in the vertical direction depends on the vertical wave number. For this reason, the equations are ill-posed with any pointwise, local specification of the boundary conditions. This enormously complicates the formulation of any adaptive mesh algorithm, since the coupling between meshes at coarse-fine boundaries cannot be locally specified. We therefore turn to projection methods, which can be combined with adaptive mesh refinement.

2.1.2 Projection Methods

The finite-difference scheme used in the present work is a projection method, that is, one in which the divergence-free component of the velocity is projected out using the Hodge decomposition as the solution is being advanced in time. The advantage of a projection method is that it transforms a constrained system of equations (momentum with a divergence-free constraint) into a pure initial-value problem. Chorin first introduced the projection method as a finite-difference, predictor-corrector formulation to solve the Navier-Stokes equations [Cho69][Cho68]. An intermediate velocity field is found in the predictor by extrapolation using the advection-diffusion equation. In the corrector the divergence-free component is projected out using the Hodge decomposition. This scheme is order $(\Delta t + \Delta x^2)$.

Bell, Colella and Glaz [BCG89] made Chorin's algorithm second order in both space and time by coupling the diffusion-convection and the projection step more closely. The solution is extrapolated forward in time in the predictor using a second-order Godunov procedure to determine the convective acceleration at the half step. The corrector advances the solution in time treating viscous terms semi-implicitly (a Crank-Nicholson solve). The divergence-free component of the velocity is then projected out using the Hodge decomposition. This scheme is second order in both space and time but shows instabilities for any $CFL > 0.5$.

Bell, Colella and Howell [BCH91] remedied the time step constraint by introducing a MAC projection in the predictor step. In this scheme, the half-step velocities are MAC projected after being generated by Godunov extrapolation. These MAC-projected velocities are then used to calculate the convective acceleration and the scheme then proceeds as did that of [BCG89]. Approximate projections simplify numerical linear algebra by sacrificing some of the design advantages associated with discrete projections. Almgren, Bell and Szymczak [ABS96] developed an approximate projection method for the Navier-Stokes equations. Lai and Colella [LC] developed a somewhat different approximate projection method. The present investigation uses an approximate projection algorithm.

2.2 Algorithm Design Criteria

The nature of the solutions of rapidly rotating flows imposes some unique constraints upon algorithm design. A large class of rotating flows consist of small perturbations of axisymmetric shear flows, which are flows of the form

$$\vec{u} = f(r)\hat{e}_\theta \tag{2.1}$$

An initial condition of this form is an exact solution to the equations of motion. Since the solutions are often perturbations of equation 2.1, the algorithm should respect it as an exact discrete solution as well. This means that if the algorithm is given an initial condition of unperturbed equation 2.1, the solution should not change with time.

When the initial condition is a small perturbation of that given by equation 2.1, both systems of equations have stable solutions of coherent, persistent vortices that exhibit a distinguishing sign of perturbational vorticity. To be considered successful, any numerical method for rapidly rotating flow must model these physical phenomena correctly. The algorithm must model the breaking up of vortices in adverse shear as well as the persistence and merging of vortices in prograde shear. Under the correct conditions, the scheme should converge to a numerical steady state of coherent perturbational vorticity.

The physics of a two dimensional quasigeostrophic, infinite Rossby deformation-radius flow is independent of the mean component of the rotation (Ω_0). The source term in equation 1.45 is given by

$$\begin{aligned} F_{qg} &= 2\Lambda \times \vec{u} \\ \Lambda &= \Omega_0 + \beta R \\ \Omega_0 &= constant \end{aligned} \tag{2.2}$$

The component of this force (F_{qg}) that corresponds to the mean rotation ($2\Omega_0 \times U$) reduces to a pure gradient and can be considered part of the pressure gradient. Therefore it is reasonable for one to expect that the algorithm respect this physical fact. If given a nontrivial mean rotation ($\Omega_0 \neq 0$), the results should be no different from the vanishing mean rotation case.

Also it is also important that the total sum of kinetic energy in the flow remain as close to constant as possible. It would be impossible to claim that we are modeling the physics of this non-dissipative, non-forced system if the algorithm were introducing significant amounts of kinetic energy. Some dissipation of kinetic energy is reasonable but the results are also suspect if the algorithm is too dissipative.

The final design criterion will be most difficult to measure and most difficult to enforce. It was shown in the last chapter that the shallow water equations reduce to the quasigeostrophic equations in the limit of small Rossby number. The numerical scheme for shallow water should show some measure of convergence to quasigeostrophic in the same limit.

2.3 Discrete and Approximate Projections

Any vector field $\vec{W}$ in a domain Λ can be described as a sum of a divergence- free component field and a gradient field (see equation 1.10).

$$\begin{gathered}\vec{W} = \vec{W}_d + \mathcal{G}\phi \\ \mathcal{D}W_d = 0 \quad on \quad \Lambda \\ \tfrac{\partial\phi}{\partial n} = \vec{W}\cdot\hat{n} \quad on \quad \partial\Lambda\end{gathered} \tag{2.3}$$

The analytic projection operator is defined as that operator that extracts the divergence-free part from a vector field:

$$\mathcal{P}_{\mathcal{O}}W = W_d \tag{2.4}$$

where

$$\mathcal{P}_{\mathcal{O}} \equiv (I - \mathcal{G}(\mathcal{L})^{-1}\mathcal{D}) \tag{2.5}$$

This means that to define a *numerical* projection operator, it is first necessary to define numerical divergence, gradient and Laplacian operators. Discrete approximations to the analytic operators $\mathcal{D}, G$ and $\mathcal{L}$ are denoted by D, G and L.

There are two classes of numerical projections that have been investigated for algorithms of the present type: discrete and approximate. A discrete projection must fulfill two conditions. First its divergence operator is the discrete adjoint of its gradient operator with respect to some appropriate discrete scalar inner product and vector inner product.

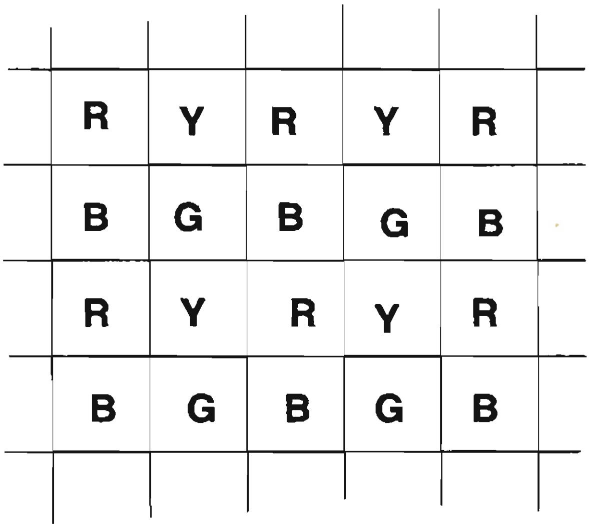
$$<\mathrm{G}\phi, \vec{u}> = <\mathrm{D}\vec{u}, \phi> \tag{2.6}$$

Second, the Laplacian operator for a discrete projection L is defined by its divergence and gradient operators:

$$\mathrm{L} \equiv \mathrm{DG} \tag{2.7}$$

which means that a discrete projection operator is of the form

$$\mathrm{P} \equiv (I - \mathrm{G}(\mathrm{DG})^{-1}\mathrm{D}) \tag{2.8}$$



R	Y	R	Y	R	
B	G	B	G	B	
R	Y	R	Y	R	
B	G	B	G	B	

Figure 2.1: Decoupled stencils for the Laplacian of a discrete projection with colocated, cell-centered velocities and standard centered-difference divergence and gradient operators. R, G, B, Y represent the four decoupled stencils for the Laplacian.

When a numerical projection satisfies equation 2.6 and equation 2.7, its output is *exactly* divergence-free with respect to its divergence operator [Cho69]. This means that applying a discrete projection multiple times is the same as applying it once:

$$\mathrm{P}(\mathrm{P}(\vec{u})) = \mathrm{P}(\vec{u}) \tag{2.9}$$

Though equation 2.9 is an attractive property, it leads to a number of disadvantages. For colocated velocities, equation 2.7 typically produces discretizations of the Laplacian that are badly behaved. For cell-centered velocities (which are necessary for other parts of the algorithm), and standard centered-difference divergence D and gradient G operators, the discrete Laplacian operator that comes out of equation 2.7 has four stencils that are completely decoupled except at boundaries. See figure 2.1 for a picture of this stencil; in the figure, the R,G,B, and Y stencils are decoupled. This makes adaptive mesh refinement extremely difficult because this decoupling has to be respected across grids (see Howell and Bell [HB] for details).

The second class of numerical projections, approximate projections, removes the constraint equation 2.7. For approximate projections, the divergence and gradient are still discrete adjoints but the Laplacian is no longer determined by the divergence and gradient operators:

$$L \not\equiv DG \tag{2.10}$$

This freedom allows the designer to choose a Laplacian operator with manageable linear algebra. The price of this freedom is the loss of the simplicity of equation 2.9; applying an approximate projection multiple times is *not* the same as applying it once.

$$\mathrm{P}(\mathrm{P}(\vec{u})) \neq \mathrm{P}(\vec{u}) \tag{2.11}$$

This complicates some aspects of the algorithm design, as will be explained in the next section.

2.4 Design of Present Algorithm: Projection Discretization

Given that the primary long-term goal of this work is to adapt high resolution, adaptive algorithms to model rapidly rotating flows, we will focus on methods based upon approximate projections. From here on, when the operator P is used, it refers to an approximate projection. Computations take place on a grid of $N_r \times N_\theta$ cells with mesh spacing $\Delta r \times \Delta\theta$ and are indexed by i in the r direction and indexed by j in the θ direction. The index is cell-centered; so any variable of the form $\eta_{i,j}$ is cell-centered and any variable of the form $\eta_{i+\frac{1}{2},j}$ is r-edge centered and $\eta_{i,j+\frac{1}{2}}$ is θ-edge centered.

2.4.1 MAC Projection

We use two projections in tis work, one for edge-centered velocities referred to as the MAC projection (as by Harlow, et. al. [HW65]) and one for cell-centered velocities (referred to as the approximate projection or simply projection). The MAC projection is a discrete projection. The MAC projection divergence is given as follows:

$$\begin{aligned} \mathrm{D}_{MAC}\vec{u}_{i,j} = & \tfrac{\Delta\theta}{\Delta V_{i,j}}(r_{i+\frac{1}{2},j}u_{r,i+\frac{1}{2},j} - r_{i-\frac{1}{2},j}u_{r,i-\frac{1}{2},j}) \\ & + \tfrac{\Delta r}{\Delta V_{i,j}}(u_{\theta,i,j+\frac{1}{2}} - u_{\theta,i,j-\frac{1}{2}}) \end{aligned} \tag{2.12}$$

where the volume of the cell is given by:

$$\Delta V_{i,j} = \Delta\theta(r^2_{i+\frac{1}{2},j} - r^2_{i-\frac{1}{2},j}) \tag{2.13}$$

The gradient for the MAC projection is given as follows:

$$\begin{aligned} \mathrm{G}_{MAC}\phi_{r,i+\frac{1}{2},j} &= \tfrac{1}{\Delta r}(\phi_{r,i+1,j} - \phi_{r,i,j}) \\ \mathrm{G}_{MAC}\phi_{\theta,i,j+\frac{1}{2}} &= \tfrac{1}{\Delta\theta r_{i,j+\frac{1}{2}}}(\phi_{\theta,i,j+1} - \phi_{\theta,i,j}) \end{aligned} \tag{2.14}$$

and the Laplacian (for interior points) is given by

$$\mathrm{L}\phi_{i,j} = \mathrm{D}_{MAC}(\mathrm{G}_{MAC}\phi) \tag{2.15}$$

For boundary edges, the flux of the gradient is assumed to be zero. For example, at the outer radius, the Laplacian is given by the following:

$$\mathrm{L}\phi_{N_r,j} = \frac{\Delta\theta}{\Delta V_{i,j}}(-r_{i-\frac{1}{2},j}\mathrm{G}_{MAC}\phi_{r,i-\frac{1}{2},j}) + \frac{\Delta r}{\Delta V_{i,j}}(\mathrm{G}_{MAC}\phi_{\theta,i,j+\frac{1}{2}} - \mathrm{G}_{MAC}\phi_{\theta,i,j-\frac{1}{2}}) \tag{2.16}$$

Functionally a MAC projection of an edge-centered velocity field $\vec{u}_e$ is calculated by first solving

$$\mathrm{DG}_{MAC}\phi = \mathrm{D}_{MAC}(\vec{u}_e). \tag{2.17}$$

with homogeneous Neumann boundary conditions on ϕ. On the boundary, the divergence is taken forcing the normal velocities at the boundary edge to zero (no-flow condition):

$$\begin{aligned} u_{r,-\frac{1}{2},j} &= 0 \\ u_{r,N_r+\frac{1}{2},j} &= 0 \end{aligned} \tag{2.18}$$

As an aside, solving the discrete Laplace's equation 2.6 with homogeneous Neumann boundary conditions requires that the right-hand side of the equation sum to zero to ensure solvability. Equation 2.12 is the conservative difference form of the divergence. Using the conservative form of the equation, along with the no flow condition (equation 2.18) ensures solvability for equation 2.17.

After equation 2.17 is solved, the gradient of ϕ is taken using one-sided differences at boundaries (actually ϕ is third-order extrapolated into a ghost cell and the gradient is taken as if there were no boundary, but this is equivalent to taking one-sided differences.) The gradient of the solution ϕ of equation 2.17 is taken using equation 2.14 and subtracted from the velocity.

$$\mathrm{P}_{MAC}\vec{u}_e = \vec{u}_e - \mathrm{G}_{MAC}\phi \tag{2.19}$$

Because the MAC projection is a discrete projection, edge velocities that have been MAC projected are exactly divergence-free with respect to D_{MAC}.

2.4.2 Cell-Centered Approximate Projection

The second projection used in the present work is the cell-centered approximate projection, which uses much of the infrastructure of the MAC projection along with a few averaging operators. To average a vector field $\vec{w}$ from centers to edges, the average of the two closest centers is taken:

$$\begin{aligned} (A_{C\to E}(\vec{w}))_{i+\frac{1}{2},j} &= \begin{pmatrix} \frac{1}{2}(w_{r,i+1,j} + w_{r,i,j}) \\ \frac{1}{2}(w_{\theta,i+1,j} + w_{\theta,i,j}) \end{pmatrix} \\ (A_{C\to E}(\vec{w}))_{i,j+\frac{1}{2}} &= \begin{pmatrix} \frac{1}{2}(w_{r,i,j+1} + w_{r,i,j}) \\ \frac{1}{2}(w_{\theta,i,j+1} + w_{\theta,i,j}) \end{pmatrix} \end{aligned} \tag{2.20}$$

We average $\vec{w}$ from cell edges by arithmetically averaging the normal components of the edges:

$$(A_{E\to C}(\vec{w}))_{i,j} = \begin{pmatrix} \frac{1}{2}(w_{r,i+\frac{1}{2},j} + w_{r,i-\frac{1}{2},j}) \\ \frac{1}{2}(w_{\theta,i,j+\frac{1}{2}} + w_{\theta,i,j-\frac{1}{2}}) \end{pmatrix} \tag{2.21}$$

Boundary edge values are extrapolated from neighboring edges. Cell-centered averages at boundaries are found by second order extrapolation into a ghost cell and averaging as if boundary were not there.

The approximate cell-centered projection has a gradient that is the average of the edge-centered gradient:

$$\mathrm{G}_{\mathrm{O}}\phi = A_{E\rightarrow C}(\mathrm{G}_{MAC}\phi), \tag{2.22}$$

a discrete divergence operator given by

$$\mathrm{D}_{\mathrm{O}}\vec{u} = \mathrm{D}_{MAC}(A_{C\rightarrow E}(\vec{u})), \tag{2.23}$$

and a discrete Laplacian exactly the same as that for the MAC:

$$\mathrm{L}\phi = \mathrm{D}_{MAC}(\mathrm{G}_{MAC}\phi) \tag{2.24}$$

To extract the divergence-free part of a cell-centered velocity field $\vec{u}_{cc}$ using this projection, one solves

$$\mathrm{L}\phi = \mathrm{D}_{\mathrm{O}}(\vec{u}_{cc}) \tag{2.25}$$

with homogeneous Neumann boundary conditions for ϕ at the inner and outer radii. Then the gradient of the solution ϕ is found using equation 2.22 and is subtracted off the velocity field:

$$\mathrm{P}^{\mathrm{O}}\vec{u}_{cc} = \vec{u}_{cc} - \mathrm{G}_{\mathrm{O}}\phi \tag{2.26}$$

Though the approximate projection derives from a discrete projection, as the product of these two averaging operators is not the identity.

2.4.3 Extension to Shallow Water of Projections

The shallow-water velocity has a slightly different constraint:

$$\nabla \cdot (HW_d) = 0 \tag{2.27}$$

Because of this, it is necessary to extend these projections to variable depth. To do this, the MAC divergence and the Laplacian for shallow water is weighted with H:

$$\begin{aligned} \mathrm{D}^{SW}_{MAC}(\vec{u}) &= \mathrm{D}_{MAC}(H\vec{u}) \\ \mathrm{L}^{SW}(\phi) &= \mathrm{D}_{MAC}(H\mathrm{G}_{MAC}(\phi)) \end{aligned} \tag{2.28}$$

So to apply a MAC projection of a edge-centered velocity field $\vec{u}_e$ constrained by equation 2.27, first solve

$$\mathrm{D}_{MAC}(H\mathrm{G}_{MAC}(\phi)) = \mathrm{G}_{MAC}(H\vec{u}_e) \tag{2.29}$$

and then subtract the gradient from the velocity field

$$\mathrm{P^H}_{MAC}\vec{u}_e = \vec{u}_e - G_{MAC}\phi \tag{2.30}$$

where ϕ is the solution to equation 2.29.

The extension from the MAC projection to the cell-centered projections is exactly the same as in the last section. The cell-centered, shallow-water divergence and Laplacian operators are given by the following.

$$\begin{aligned} \mathrm{D}^{SW}_{O}\vec{u} &= \mathrm{D}^{SW}_{MAC}(A_{C\to E}(\vec{u})) \\ \mathrm{L}^{SW}(\phi) &= \mathrm{D}_{MAC}(H\mathrm{G}_{MAC}(\phi)) \end{aligned} \tag{2.31}$$

So to numerically enforce the shallow-water constraint (equation 2.27) upon a vector field $\vec{u}_{cc}$ using this projection, one solves

$$\mathrm{D}_{MAC}(H\mathrm{G}_{MAC}(\phi)) = \mathrm{G}_{MAC}(H\vec{u}_{cc}) \tag{2.32}$$

with homogeneous Neumann boundary conditions for ϕ at the inner and outer radii. Then the gradient of the solution ϕ is found using equation 2.22 and is subtracted off the velocity field.

$$\mathrm{P^H}\vec{u}_{cc} = \vec{u}_{cc} - \mathrm{G_O}\phi \tag{2.33}$$

None of the averaging operators change. The gradient operators are also the same as those in the previous section.

2.5 Algorithm overview

From here on the algorithm is given in terms of the shallow-water equations. The quasigeostrophic algorithm is a straightforward variation of the method given here. In particular, to get the quasigeostrophic algorithm, one simply sets the bottom depth constant ($H = H_O$) and makes the rotation a linear function of r ($\Omega_{QG} = \Omega_0 - \beta r$).

Time is discretized in increments of Δt and time steps are denoted by the superscript n. The relevant numerical variables are related to their analytic counterparts as follows

$$\begin{aligned} \vec{u}^{n}_{i,j} &\approx \vec{u}(i\Delta r, j\Delta\theta, n\Delta t) \\ F^{n}_{i,j} &\approx F(i\Delta r, j\Delta\theta, n\Delta t) \\ \nabla p^{n+\frac{1}{2}}_{i,j} &\approx \nabla p(i\Delta r, j\Delta\theta, (n+\tfrac{1}{2})\Delta t) \end{aligned} \tag{2.34}$$

where

$$F = F_{geom} + 2\Omega \times \vec{u} \tag{2.35}$$

The current time step is denoted by n so the algorithm brings the solution from time step n to time step $n+1$.

To advance the solution in time, the projection formulation of the shallow-water equations (1.19) is discretized with particular care as to the representation of the forcing terms:

$$\begin{aligned} \vec{u}^{T} &= \vec{u}^{n} + \Delta t((\vec{u}\cdot\nabla)\vec{u}^{n+\frac{1}{2}} + F(\vec{u},\vec{x})^{n-\frac{1}{2}} - \nabla p^{n-\frac{1}{2}}) \\ \vec{u}^{n+1} &= \vec{u}^{n} + \mathrm{P}^{\mathrm{H}}(\vec{u}^{T} - \vec{u}^{n}) + \Delta t A_{E\rightarrow C}({\mathrm{P}^{\mathrm{H}}}_{MAC}(dF)) \\ \nabla p^{n+\frac{1}{2}} &= \nabla p^{n-\frac{1}{2}} + \mathrm{Q}^{\mathrm{H}}(\vec{u}^{T} - \vec{u}^{n} + \Delta t A_{E\rightarrow C}(\mathrm{Q}^{H}_{MAC}(dF)) \end{aligned} \tag{2.36}$$

where

$$dF = F^{n+\frac{1}{2}} - F^{n-\frac{1}{2}} \tag{2.37}$$

The computation of each particular term is explained in later sections. In making this choice of discretization, the design criteria of the algorithm are used. First, the algorithm is to be able to exactly preserve axisymmetric parallel shear flows, which are of the form:

$$\vec{u} = f(r)\hat{e}_{\theta} \tag{2.38}$$

This exact solution represents a balance between the F and the pressure gradient. Pressure gradients in predictor-corrector algorithms of the type used in [BCH91] are lagged in time. Since the design is constrained to use approximate projections, in order to exactly preserve this form of solution, the quantity actually sent to the projection must be zero when the velocity is of the form of equation 2.38. The source term F is therefore lagged with the pressure so they can balance exactly. If the flow is of the form of equation 2.38 and the pressure gradient is correct, the result of putting this into equation 2.36 is the following:

$$\begin{aligned} dF &= (\vec{u}\cdot\nabla)\vec{u}^{n+\frac{1}{2}} = 0 \\ \vec{u}^{T} &= \vec{u}^{n} \\ \vec{u}^{n+1} &= \vec{u}^{n} + A_{E\to C}\mathrm{P}^{H}_{MAC}(0) = \vec{u}^{n} \\ \nabla p^{n+\frac{1}{2}} &= \nabla p^{n-\frac{1}{2}} + A_{E\to C}\mathrm{Q}_{MAC}(0) = \nabla p^{n-\frac{1}{2}} \end{aligned} \tag{2.39}$$

With this fine balance, axisymmetric parallel shear is preserved to roundoff by the discretization in equation 2.36.

2.5.1 Predictor

Restating the first step from the corrector:

$$\vec{u}^{T} = \vec{u}^{n} + \Delta t(\vec{u}\cdot\nabla\vec{u}^{n+\frac{1}{2}} + F(\vec{u},\vec{x})^{n+\frac{1}{2}} - \nabla p^{n-\frac{1}{2}}) \tag{2.40}$$

Clearly much detail has been left out. The convective acceleration is calculated from half step velocities as follows:

$$\vec{u}\cdot\nabla\vec{u}_{i,j}{}^{n+\frac{1}{2}} = \frac{(u_{i+\frac{1}{2},j,r} + u_{i-\frac{1}{2},j,r})}{2\Delta r}(\vec{u}_{i+\frac{1}{2},j} - \vec{u}_{i-\frac{1}{2},j}) + \frac{(u_{i,j+\frac{1}{2},\theta} + u_{i,j-\frac{1}{2},\theta})}{2r_{i,j}\Delta\theta}(\vec{u}_{i,j+\frac{1}{2}} - \vec{u}_{i,j-\frac{1}{2}}) \tag{2.41}$$

The time-centered approximation to the source term of the equation is found using these same edge-centered velocities. This is given as follows:

$$\begin{aligned} F^{n+\frac{1}{2}}_{i+\frac{1}{2},j} &= F_{geom}(\vec{u}^{n+\frac{1}{2}}) + 2\Omega\times\vec{u}^{n+\frac{1}{2}} \\ F^{n+\frac{1}{2}}_{i+\frac{1}{2},j} &= \left(\begin{array}{l} \frac{u_{\theta,i+\frac{1}{2},j}{}^{2}}{R_{i+\frac{1}{2},j}} + 2\Omega R_{i+\frac{1}{2},j}u_{\theta,i+\frac{1}{2},j} \\ \frac{-u_{\theta,i+\frac{1}{2},j}u_{r,i+\frac{1}{2},j}}{R_{i+\frac{1}{2},j}} - 2\Omega R_{i+\frac{1}{2},j}u_{r,i+\frac{1}{2},j} \end{array}\right) \end{aligned} \tag{2.42}$$

Coriolis forces on the boundary of the domain are computed by second-order extrapolation from neighboring edges.

2.5.2 Time-Centered Edge Velocities

Clearly the gaping hole that remains in the description of the algorithm is the description of how to calculate the edge velocities (which appear in equation 2.41 and equation 2.42).

To calculate edge-centered velocities centered in time, we extrapolate it in time and space and then MAC project it to make it divergence-free. This extrapolation is of the following form ($(i+\frac{1}{2},j)$ edges are shown here. The equations for the y $(i,j+\frac{1}{2})$ are computed similarly):

$$\vec{u}^{n+\frac{1}{2}}_{i+\frac{1}{2},j} = \vec{u}^{n}_{i,j} + \Delta x \frac{\partial \vec{u}}{\partial x} + \Delta t \frac{\partial \vec{u}}{\partial t} \tag{2.43}$$

Derivatives are approximated in two ways. Transverse derivatives are calculated using upwinding to insure that information travels downstream. Van Leer limited slopes are used to help eliminate of spurious oscillations that might otherwise occur in the presence of underresolved gradients. This very simple extrapolation does in fact only eliminate spurious maxima and minima in one dimension. In two or three dimensions small overshoots or undershoots are possible and it takes a much more complicated extrapolation scheme to remedy this. The van Leer slopes used in this algorithm are discretized as follows:

$$\Delta\eta^{R,VL}_{i,j} = \text{sign}(\eta_{i+1,j} - \eta_{i-1,j}) \min(2(\eta_{i+1,j} - \eta_{i,j}), 2(\eta_{i,j} - \eta_{i-1,j}), .5(\eta_{i+1,j} - \eta_{i-1,j})) \tag{2.44}$$

while the upwind calculation is as follows:

$$\Delta\eta^{R,upwind}_{i,j} = \left\{ \begin{array}{ll} \eta_{i+1,j} - \eta_{i,j} & \text{if } (u_{r,i,j} < 0) \\ \eta_{i,j} - \eta_{i-1,j} & \text{otherwise} \end{array} \right\} \tag{2.45}$$

where (η) can be any scalar or vector associated with the flow. On the ghost cells the values of η are set by extrapolation:

$$\eta_{N_r,j} = 2\eta_{N_r-1,j} - \eta_{N_r-2,j} \tag{2.46}$$

This mimics to first order taking one-sided differences at the boundaries. By inserting equation 1.3 for the time derivative, left and right velocities on each edge are extrapolated using upwind differences for the transverse direction and Van Leer slopes for the extrapolation direction:

$$\begin{aligned} \vec{u}_{i+\frac{1}{2},j}^{n+\frac{1}{2}Left} &= \vec{u}_{i,j}^{n} + \tfrac{1}{2}(1 - \tfrac{\Delta t}{\Delta r} max(0, (u^n)_{r,i,j})\Delta \vec{u}_{i,j}{}^{R,VanLeer} \\ &\quad - \tfrac{\Delta t}{r_i j 2\Delta\theta}(u_\theta{}^n)_{i,j}\Delta\vec{u}_{i,j}{}^{\theta,upwind} + \tfrac{\Delta t}{2}(F^n)_{i,j} \\ \vec{u}_{i+\frac{1}{2},j}^{n+\frac{1}{2}Right} &= \vec{u}_{i+1,j}^{n} + \tfrac{1}{2}(-1 - \tfrac{\Delta t}{\Delta r} min(0, (u^n)_{r,i+1,j})\Delta \vec{u}_{i+1,j}{}^{R,VanLeer} \\ &\quad - \tfrac{\Delta t}{r_{i+1,j} 2\Delta\theta}(u_\theta{}^n)_{i+1,j}\Delta\vec{u}_{i+1,j}{}^{\theta,upwind} + \tfrac{\Delta t}{2}(F^n)_{i+1,j} \end{aligned} \tag{2.47}$$

For interior edges, the Riemann problem that equation 2.47 presents is solved by choosing the upwind state. The velocity used for the upwinding is the spatial average of the previous time step's velocity. The edge velocities from equation 2.47 are not divergence-free and using them for upwinding can cause large errors, as projecting out the divergence component can change the direction of the velocity field at points, causing the upwind direction to change.

$$\begin{gathered} u_{\text{eff}} = 0.5(u^n{}_{r,i,j} + u^n{}_{r,i+1,j}) \\ \vec{u}_{i+\frac{1}{2},j}^{n+\frac{1}{2},*} = \begin{pmatrix} \vec{u}_{i+\frac{1}{2},j}^{n+\frac{1}{2}Left} & \text{if } (u_{\text{eff}} > 0) \\ \vec{u}_{i+\frac{1}{2},j}^{n+\frac{1}{2}Right} & \text{otherwise} \end{pmatrix} \end{gathered} \tag{2.48}$$

After going through the analogous procedure to find θ edge-centered velocities ($\vec{u}_{i,j+\frac{1}{2}}^{n+\frac{1}{2},*}$) these velocities are then MAC projected:

$$\vec{u}_{i+\frac{1}{2},j}^{n+\frac{1}{2}} := \mathrm{P^H}_{MAC}(\vec{u}_{i+\frac{1}{2},j}^{n+\frac{1}{2},*}) \tag{2.49}$$

In this projection, the components of the edge-centered velocity which is tangential to the edge must also be corrected. This is done as follows for the $u_{r,i,j+\frac{1}{2}}$ component.

$$u_{r,i,j+\frac{1}{2}} = u_{r,i,j+\frac{1}{2}} - \tfrac{1}{4}(G_{MAC}\phi_{r,i+\frac{1}{2},j} + G_{MAC}\phi_{r,i-\frac{1}{2},j} + G_{MAC}\phi_{r,i+\frac{1}{2},j+1} + G_{MAC}\phi_{r,i-\frac{1}{2},j+1}) \tag{2.50}$$

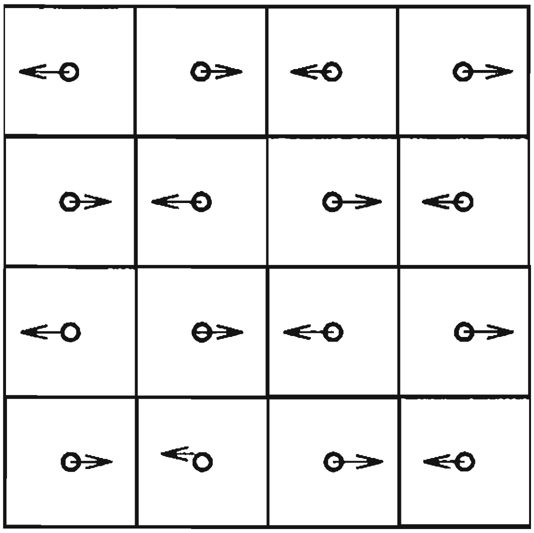


Figure 2.2: Discretely divergence-free mode of the velocity field that is obviously not divergence-free. This is the mode of the velocity that the filter seeks to eliminate.

The $u_{\theta,i+\frac{1}{2},j}$ component is corrected analogously. Since the MAC projection is a discrete projection, the fancy maneuvering of the forcing terms in the overview section is not necessary.

2.5.3 Filtering Spurious Modes

There are high wavenumber modes in the velocity field that are D_O-divergence-free but are not smooth, such as illustrated in figure 2.2. Because they are D_O-divergence-free, these modes can accumulate in the velocity field to the point that they interact with the nonlinear terms in the equations in a deleterious fashion. To control such modes, we use the approach of Lai and Colella [LC] to filter them. In particular, we find divergence and gradient pairs such that a crude approximation of a projection based on that pair damps these modes.

To keep this mode from growing, it must be filtered out of the velocity field. To do this, what is essentially a third and a fourth projection are designed. One filters divergence centered on $i+\frac{1}{2},j$ edges, the other filters divergence centered on $i,j+\frac{1}{2}$ edges. The full projection solve is not completed, however. Since the mode of the velocity that must be eliminated is very high frequency, it is sufficient to simply point-relax the solution.

Only the $i+\frac{1}{2},j$ filter is described here. The $i,j+\frac{1}{2}$ filter is a straightforward extension. This implementation of the filter will closely follow the implementation by Lai ([Lai94]) with the appropriate extensions to take into account coordinate system metrics. As with the other projections, the design starts with the definition of discrete gradient and divergence operators:

$$\begin{gathered} G_{FILT}\phi_{i,j} = \begin{pmatrix} \frac{1}{\Delta r}(\phi_{i+\frac{1}{2},j} - \phi_{i-\frac{1}{2},j}) \\ \frac{1}{4r_{i,j}\Delta\theta}(\phi_{i+\frac{1}{2},j+1} - \phi_{i+\frac{1}{2},j-1} + \phi_{i-\frac{1}{2},j+1} - \phi_{i-\frac{1}{2},j-1}) \end{pmatrix} \\ D_{FILT}\vec{u}_{i+\frac{1}{2},j} = \frac{\Delta\theta}{\Delta V_{i+\frac{1}{2},j}}(r_{i+1,j}u_{r,i+1,j} - r_{i,j}u_{r,i,j}) \\ +\frac{\Delta r}{4\Delta V_{i}phj}((u_{\theta,i+1,j+1} - u_{\theta,i+1,j}) + (u_{\theta,i+1,j} - u_{\theta,i+1,j-1}) \\ +(u_{\theta,i,j+1} - u_{\theta,i,j}) + (u_{\theta,i,j} - u_{\theta,i,j-1})) \end{gathered} \tag{2.51}$$

If this were to be done as a fully solved projection, the following would be solved:

$$D_{FILT}G_{FILT}(\phi) = D_{FILT}(\vec{u}) \tag{2.52}$$

and the gradient of the solution would be subtracted from the velocity field. Though this is a perfectly acceptable solution to the problem at hand, it is the essence of overkill. To clean up after the projection, this would introduce two more Poisson solves, twice as much computational effort as the projection itself. The velocity mode that must be eliminated is very high frequency and eight passes of point relaxation of the solution of equation 2.52 have been found to be adequate for the needs at hand.

$$(\vec{u})^{l+1} := (\vec{u})^{l} - G_{FILT}(\lambda D_{FILT}((\vec{u})^{l})) \tag{2.53}$$

where l is the iteration number. The calculation of the value of the point-relaxation parameter (λ) is done to eliminate the diagonal of the (GD) term in the equation 2.53:

$$\lambda_{i+\frac{1}{2},j} = \frac{8r_{i,j}r_{i+1,j}\Delta\theta}{\Delta r(r_{i,j}+r_{i+1,j})} + \frac{\Delta r}{\Delta\theta(r_{i,j}+r_{i+1,j})} \tag{2.54}$$

The boundary conditions on the velocity going into the filter used here are standard slip wall conditions:

$$\begin{aligned} u_{r,N_r,j} &= -u_{r,N_r-1,j} \\ u_{\theta,N_r,j} &= u_{\theta,N_r-1,j} \end{aligned} \tag{2.55}$$

2.5.4 Extension to Shallow Water

As with the projection, the filter must be somewhat modified in the case of shallow water where the velocity field is constrained slightly differently:

$$\nabla \cdot (HW_d) = 0 \tag{2.56}$$

It is necessary to extend the filter to variable depth. To do this, the filter divergence and Laplacian for shallow water is simply weighted with H:

$$D_{FILT}{}^{S}W(\vec{u}) = D_{FILT}(H\vec{u}) \tag{2.57}$$

The gradient G_{FILT} does not change. So to filter out the spurious modes of a velocity velocity field $\vec{u}_e$ constrained by equation 2.56, we simply iterate with the point relaxation scheme

$$(\vec{u})^{l+1} := (\vec{u})^{l} - G_{FILT}(\lambda D_{FILT}((\vec{u})^{l})) \tag{2.58}$$

with a relaxation parameter λ modified to take into account variable depth

$$\lambda_{i+\frac{1}{2},j} = \frac{8r_{i,j}r_{i+1,j}\Delta\theta}{H_{i+\frac{1}{2},j}\Delta r(r_{i,j}+r_{i+1,j})} + \frac{\Delta r}{H_{i+\frac{1}{2},j}\Delta\theta(r_{i,j}+r_{i+1,j})} \tag{2.59}$$

We have found that the number of iterations required to filter out spurious modes is insensitive to depth variation.

2.6 Numerical Analysis Issues

2.6.1 Multigrid and Point Relaxation

Linear systems arising from the discritizations of elliptic equations are solved using multigrid iteration are solved by multigrid. Multigrid is an iterative procedure rather than a direct solution. Standard multigrid uses point relaxation to make the residual resolvable on a coarser grid, averages the residual onto this coarser grid, relaxes again and so on until the grid is a single cell. At this point one interpolates the correction up to finer and finer grids, relaxing at each step, until reaching the original grid. This cycle one repeats until norm of the residual goes below a given threshold. There are many variations on this "V cycle"; for a more complete discussion see Briggs [Bri87].

The first issue that must be discussed here is relaxation. If the grid is square and uniform, a simple explicit point-relaxation scheme is sufficient to damp all high wavenumber modes of the residual so that the residual can be represented on a coarser grid. If the grid cells are not square and uniform, then multigrid cannot hope to work with point relaxation. Define an L to be a linear operator and ρ to be be the source term in the equation.

$$L\phi = \rho \tag{2.60}$$

Putting this into residual-correction form, where δ is the correction and R is the residual, this becomes

$$L\delta = R \tag{2.61}$$

Using m for the iteration number and λ for the relaxation parameter, any relaxation scheme is of the form

$$\begin{aligned} \delta^{m+1} &= \delta^{m} + \lambda(L\delta - R) \\ \delta^{m+1} &= M(\delta^{m}) - \lambda R \end{aligned} \tag{2.62}$$

In point relaxation, a fully explicit method, the term $L\delta$ is evaluated at iteration m (the old iteration). More robust implicit relaxation schemes will be discussed in this section. In general, the relaxation parameter (λ) is chosen to eliminate the diagonal term of the correction (solve for the (i,i) coefficient of δ in equation 2.62 and set it to zero). Stability considerations come into play, however, with explicit schemes. The residual can be expanded in terms of it discrete Fourier components.

$$\begin{aligned} R &= \sum_{k_x}\sum_{k_y} C_{k_x,k_y} W^{k_x,k_y} \\ W^k &= exp(\frac{2\pi(k_x j_x + k_y j_y)}{N})) \end{aligned} \tag{2.63}$$

A relaxation scheme is sufficient for multigrid convergence if the residual's highest wavenumber Fourier component is eliminated after relaxation is completed. This means that the residual can be completely represented on a coarser grid. the core principle of multigrid.

As an illustrative example, suppose that the operator is a standard five-point Laplacian in cartesian coordinates with a nonunity aspect ratio ($\Delta x \neq \Delta y$). The equation (the equivalent of equation 2.61) then becomes

$$\frac{1}{\Delta x^2}(\phi_{j_x+1,j_y} + \phi_{j_x-1,j_y} - 2\phi_{j_x,j_y}) + \frac{1}{\Delta y^2}(\phi_{j_x,j_x+1} + \phi_{j_x,j_y-1} - 2\phi_{j_x,j_y}) = \rho_{j_x,j_y} \tag{2.64}$$

Point relaxation (the first iteration, $m = 1$, with an initial solution ϕ^0 set to zero) becomes the following:

$$\begin{aligned} \phi^{m+1} &= \phi^m + \lambda(L\phi^m - \rho) \\ \phi^1 &= -\lambda\rho \\ R^1 &= L\phi^1 - \lambda\rho \\ R^1 &= -L(\lambda\rho) - \lambda\rho \\ R^1 &= Q(\rho) \end{aligned} \tag{2.65}$$

Expanding (ρ) into discrete Fourier components and defining (γ) as the symbol of the point relaxation operator (Q), we can determine the size of each Fourier component of the scheme.

$$\begin{aligned} \beta_x &= \frac{2\pi j_x k_x}{N} \\ \beta_y &= \frac{2\pi j_y k_y}{N} \\ W^{k_x,k_y} &= exp(i(\beta_x + \beta_y) \\ \rho &= \sum_{kx=-\frac{N}{2}+1}^{\frac{N}{2}} \sum_{ky=-\frac{N}{2}+1}^{\frac{N}{2}} C_{k_x,k_y} W^{k_x,k_y} \\ R &= \sum_{kx=-\frac{N}{2}+1}^{\frac{N}{2}} \sum_{ky=-\frac{N}{2}+1}^{\frac{N}{2}} D_{k_x,k_y} W^{k_x,k_y} \\ D_{k_x,k_y} &= \gamma_{k_x,k_y} C_{k_x,k_y} \end{aligned} \tag{2.66}$$

Clearly, for a relaxation scheme to eliminate the highest wavenumber for any arbitrary right-hand side, the symbol of the scheme must go to zero at that wavenumber. The symbol (γ)

for the point relaxation scheme and the relaxation parameter are the following:

$$\begin{aligned} \gamma(k_x, k_y) &= \tfrac{\lambda}{\Delta x^2}(-2cos(\beta_x) + 2) + \tfrac{\lambda}{\Delta y^2}(-2cos(\beta_y) + 2) - 1 \\ \lambda &= \tfrac{\Delta x^2 \Delta y^2}{4(\Delta x^2 + \Delta y^2)} \end{aligned} \tag{2.67}$$

The relaxation parameter (λ) is half that which exactly cancels the diagonal of the operator (M). This is due to a stability consideration for point relaxation. More sophisticated relaxation methods such as Gauss-Seidel can use larger relaxation parameters. Next, the wavenumber is set to the highest resolved discrete wavenumber ($kx = ky = \frac{N}{2}$) and the largest value of the symbol at this wavenumber is computed.

$$\begin{aligned} \beta_x &= \pi j_x \\ \beta_y &= \pi j_y \\ \gamma(k_x = \tfrac{N}{2}, k_y = \tfrac{N}{2})^{MAX} &= 1 - (\tfrac{\Delta x^2 \Delta y^2}{(\Delta x^2 + \Delta y^2)} + \tfrac{\Delta y^2}{4\Delta x^2} + \tfrac{\Delta x^2}{4\Delta y^2}) \end{aligned} \tag{2.68}$$

Clearly this function goes to zero only if the aspect ratio is unity ($\Delta x = \Delta y$). Empirically, it has been found that point relaxation does not reduce high wavenumbers sufficiently for multigrid to converge if the aspect ratio gets much higher than 1.5:1.

One way to make a more robust relaxation scheme is to make equation 2.62 either partially or fully implicit to make the range of stable relaxation parameters larger. There are many variations on this theme of implicit relaxtion but the one used here is a variation on the alternating direction implicit (ADI) scheme for solving both the heat equation and some nonlinear systems (see Anderson [JDA95] for a description of ADI schemes). This scheme is only appropriate for diffusive systems. Moving back to cylindrical coordinates, assume we are solving a linear system that is separable into radial and azimuthal operators, i.e.:

$$L\phi = (L_R + L_\theta)\phi = \rho \tag{2.69}$$

Putting this into residual correction form (see equation 2.61), the scheme is to relax implicitly in one direction, then implicitly in the other. In this case first we solve in the radial direction, then we use the solution from that calculation to solve in the azimuthal direction.

$$\begin{aligned} \delta^{m+\frac{1}{2}} &= \delta^m + \lambda(L_R\delta^{m+\frac{1}{2}} - R) \\ (I - \lambda L_R)\delta^{m+\frac{1}{2}} &= \delta^m - \lambda R \\ (I - \lambda L_\theta)\delta^{m+1} &= \delta^{m+\frac{1}{2}} \end{aligned} \tag{2.70}$$

It is often desirable to perform more than one relaxation at a level of multigrid. To do this, the residual is computed and the previous correction (δ^m) is set to zero before each relaxation. This reduces equation 2.70 to the simpler and more symmetric form

$$\begin{aligned} R &= L^m_\phi - \rho \\ (I - \lambda L_R)\delta^{\frac{1}{2}} &= -\lambda R \\ (I - \lambda L_\theta)\delta^1 &= \delta^{\frac{1}{2}} \\ \phi^{m+1} &= \phi^m + \delta^1 \end{aligned} \tag{2.71}$$

where m is the iteration number. Since this scheme is implicit, any relaxation parameter is stable. The relaxation parameter picked here is the maximum of the relaxation parameters that would cancel the diagonals of each directional operator (i.e. the optimal relaxation parameter for each direction):

$$\lambda = \max(\lambda_{L_R}, \lambda_{L_\theta}) \tag{2.72}$$

In cylindrical coordinates, each directional relaxation parameter is a function of radius and therefore so is the overall relaxation parameter. This relaxation scheme has been tried for a inner to outer radius ratio of 4:1 for solving Poisson's equation and it works very well where point relaxation is marginal at best.

2.6.2 Time-Step Constraints in a Rotating Coordinate System

Ordinary (i.e. nonrotating) incompressible flow projection method algorithms have a time step that is constrained by the CFL condition. Specifically, throughout the numerical domain:

$$\begin{aligned} \Delta t &< \tfrac{\Delta r}{u_r} \\ \Delta t &< \tfrac{\Delta \theta}{u_\theta} \end{aligned} \tag{2.73}$$

This is simply a statement that a fluid particle is not allowed to traverse more than one cell over one time step.

Rotating flow requires an additional constraint upon the time step. To understand this, consider the ordinary differential equation

$$\frac{d\vec{u}}{dt} = -2\Omega \times \vec{u} \tag{2.74}$$

Equation 2.74 will be used to determine any constraints to the stability of the scheme brought on by the rotational term in the equations of motion. The present algorithm is a predictor-corrector scheme. In the context of the system of equations 2.74 this means that a velocity at the half time step is found

$$\vec{u}^{\,n+\frac{1}{2}} = \vec{u}^{\,n} - \frac{1}{2}\Delta t(2\Omega \times \vec{u}^{\,n}) \tag{2.75}$$

and this half-step velocity is used to advance the solution in time:

$$\vec{u}^{\,n+1} = \vec{u}^{\,n} - \Delta t(2\Omega \times \vec{u}^{\,n+\frac{1}{2}}) \tag{2.76}$$

The linear operator that this scheme represents is derived in two stages as well. First the half step velocity is given by

$$\vec{u}^{n+\frac{1}{2}} = \begin{bmatrix} 1 & \Omega\Delta t \\ -\Omega\Delta t & 1 \end{bmatrix} \vec{u}^{\,n} \tag{2.77}$$

and the solution at the new time is given by

$$\vec{u}^{n+1} = \begin{bmatrix} 1 & 2\Omega\Delta t \\ -2\Omega\Delta t & 1 \end{bmatrix} \vec{u}^{\,n+\frac{1}{2}} \tag{2.78}$$

Multiplying the two operators, the solution at the new time is given by:

$$\vec{u}^{n+1} = \begin{bmatrix} 1-2(\Omega\Delta t)^2 & -3\Omega\Delta t \\ 3\Omega\Delta t & 1-2(\Omega\Delta t)^2 \end{bmatrix} \vec{u}^{\,n} \tag{2.79}$$

Defining equation 2.79 to be of the form

$$\vec{u}^{n+1} = A\vec{u}^{n} \tag{2.80}$$

the eigenvalues of the operator A determine the numerical stability of the scheme. In order for the scheme to be considered stable, all eigenvalues λ of the matrix A must increase no more quickly than linearly in time:

$$|\lambda| < Cn\Delta t \tag{2.81}$$

(the constant C is arbitrary). The eigenvalues for the matrix A are given by

$$\lambda = 1 - 2(\Omega\Delta t)^2 \pm 3\imath\Omega\Delta t \tag{2.82}$$

and the magnitude of this complex number is given by

$$|\lambda| = 1 + 5(\Omega\Delta t)^2 + 4(\Omega\Delta t)^4 \tag{2.83}$$

From equation 2.83, it is clear that the only way that the eigenvalue λ can satisfy equation 2.81 is if the following is true:

$$\Omega\Delta t < 1 \tag{2.84}$$

Equation 2.84 is the additional constraint upon the time step for predictor-corrector methods for rotating flows. Put together with the ordinary Courant-Friedrichs-Levy (CFL) constraint, the time step for the present algorithm is calculated as follows:

$$\Delta t = \min(\frac{K}{\Omega}, \frac{Kr\Delta\theta}{u_\theta}, \frac{K\Delta r}{u_r}) \tag{2.85}$$

where K is the CFL number and is constrained to be a positive number less than unity.

Chapter 3

Algorithm Validation

A series of design criteria for the algorithm were enumerated in the first chapter. This chapter is devoted to quantifying the extent to which the present algorithm succeeds in satisfying these design criteria.

3.1 Vortical Flows Embedded in Background Shear

A very important subset of geophysical flows are those in which a flow is a perturbation of a large background shear. See figure 3.1 for an illustration. Real geophysical phenomena are often manifestations of shear-driven, vortical flows. A notable example of this is the Great Red Spot of Jupiter [Mar93]. These flows are therefore an important regime for the present algorithm to be able to model. These flows are also well understood in the case of constant-sign background shears, which makes them an excellent test problem for the present algorithm.

Throughout this section, a shallow-water calculation (denoted by SW) is one where the bottom depth varies and the rotation is constant.

$$\begin{aligned} H_{sw} &= H(r) \\ F_{rot,sw} &= 2\Omega_{sw} \times \vec{u} \\ \Omega_{sw} &= \Omega_0 = constant \end{aligned} \tag{3.1}$$

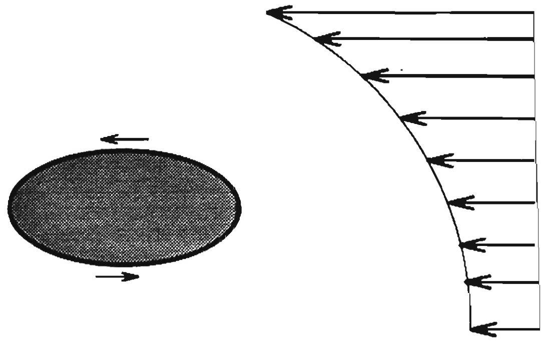


Figure 3.1: Illustration of a vortex embedded in a background shear. The flow consists of two components: the background shear, which drives the flow, and the small perturbational vortex. The dynamics of this situation are well understood and have been studied extensively.

Similarly, a quasigeostrophic calculation (denoted by QG) is one where the bottom depth is constant and the rotation is set to a β-plane.

$$\begin{aligned} H_{qg} &= H_0 = constant \\ F_{rot,qg} &= 2\Omega_{qg} \times \vec{u} \\ \Omega_{sw} &= \Omega_0 - \beta r \end{aligned} \tag{3.2}$$

In the previous chapter, it was shown that the shallow-water equations converge to the quasigeostrophic equation in the limit of rapid rotation with vanishing variation bottom topography. Convergence to this limit will be investigated in the fifth chapter. These tests are meant test the algorithm for the limits of fast rotation (small Rossby number) and slow rotation (large Rossby number). Since the quasigeostrophic limit represents fast rotation, the shallow-water problems are run with relatively slow rotation.

3.1.1 Axisymmetric Geostrophic Shear Test

Before an algorithm can be said to handle vortical flows embedded in a background shear correctly, it should be able to represent an axisymmetric shear flow without any perturbations. Here the algorithm is tested to see whether it fulfilled the requirement that

Variable	Value
CFL	0.5
Ω (for SW run)	0.5/s
β (for QG run)	0.5/(m s)
R_{in}	0.33m
R_{out}	1.33m
H_0	0.1m
S (SW bottom slope)	0.1m/m

Table 3.1: Parameters used in axisymmetric shear test.

it respect that fact that any velocity of the form

$$U = f(r)e_\theta \tag{3.3}$$

is an exact solution and should remain constant in time. An initial condition was used

$$\begin{gathered} u_\theta = \frac{1}{6(R_{out}-R_{in})^2}(v_0 r(r-(R_{out}-R_{in}))) \\ u_r = 0.0 \end{gathered} \tag{3.4}$$

where

$$v_0 = \Omega(R_{out} - R_{in}) \tag{3.5}$$

A graph of this velocity profile is given in figure 3.2. For the shallow-water case, the code was run with the bottom depth being set to

$$H = \frac{1}{2}H_0 S(2r - R_{in} - R_{out}) \tag{3.6}$$

The grid size is 32 × 160 and all parameters are contained in table 3.6.

The algorithm was also tested for the quasigeostrophic case by setting the depth constant and setting the rotational forcing as a β-plane:

$$\Omega_{qg} = -\beta r \tag{3.7}$$

Since one of the tests performed here is to see whether a constant rotation term affects the results, the constant term is set to zero here to keep the results of this test separate from

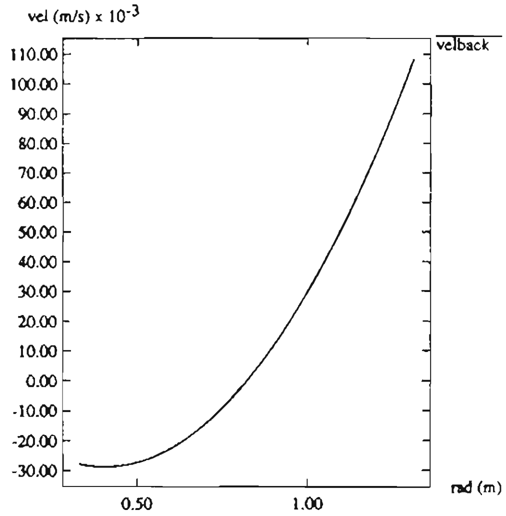


Figure 3.2: Axisymmetric shear profile.

time	SW u_r Max/Min	QG u_r Max/Min
$0.00\ t_{rot}$	0.0/0.0	0.0/0.0
$4.0 \cdot 10^{-3}\ t_{rot}$	$3.1008 \cdot 10^{-18}/-4.1450 \cdot 10^{-15}$	$3.1384 \cdot 10^{-15}/-1.7889 \cdot 10^{-18}$
$8.0 \cdot 10^{-3}\ t_{rot}$	$2.1781 \cdot 10^{-7}\ /-9.7260 \cdot 10^{-8}$	$1.4762 \cdot 10^{-8}\ /-6.1603 \cdot 10^{-8}$
$1.19 \cdot 10^{-2} t_{rot}$	$9.2894 \cdot 10^{-7}\ /-7.468 \cdot 10^{-7}$	$2.0967 \cdot 10^{-7}\ /-2.0921 \cdot 10^{-7}$
$1.59 \cdot 10^{-2} t_{rot}$	$1.9351 \cdot 10^{-6}\ /-1.5998 \cdot 10^{-6}$	$3.4250 \cdot 10^{-7}\ /-3.4211 \cdot 10^{-7}$

Table 3.2: Maxima and minima in (m/s) of numerical radial velocity with initial condition an axisymmetric geostrophic shear given in text. Time is presented in rotation times (t_{rot}= 12.5s.) See text for definitions.

the constant-rotation test results. The results of this test are given in figure 3.2 for various fractions of a rotation time t_{rot}, which is defined as

$$t_{rot} = \frac{2\pi}{\Omega} \tag{3.8}$$

in the shallow-water case, and

$$t_{rot} = \frac{2\pi}{\beta(R_{out} - R_{in})} \tag{3.9}$$

for the quasigeostrophic case. The smallness of the radial velocity is used as the metric of compliance. With exact arithmetic, the radial velocity would be exactly zero. In the present work, several factors limit the smallest value of u_r that can be expected. Defining this smallest expectable value to be ϵ_r, we can expect this value to increase with solver tolerance (ϵ_{sol}, the value of the residual at which the multigrid solver is set to stop iterating). We can also expect ϵ_r to increase linearly with the number of time steps (N_t). Computing the discrete gradient of the multigrid solution is expected to also increase ϵ_r. We estimate the ϵ_r as follows:

$$\epsilon_r = \frac{N_t \epsilon_{sol} L}{\Delta r} = N_t N_r \epsilon_{sol} \tag{3.10}$$

In the present case, three steps have been executed ($N_t = 3$) and the solver tolerance ϵ_{sol} is $1 \cdot 10^{-8}$. Therefore we estimate ϵ_r to be approximately $9.6 \cdot 10^{-7}$. Both the quasigeostrophic case and the shallow-water case preserve the solution quite well by this metric. The radial velocity is kept five orders of magnitude smaller than the azimuthal and in the worst case is only about twice the size of ϵ_r. The algorithm performs well in this test because of the careful balancing of the Coriolis force terms that occurs in the corrector step (equation 2.36). Without this careful balancing, numerical error would provide sufficient perturbation to drive the solution well away from equation 2.1.

3.2 Two Oppositely Signed Vortices in a Background Shear

Next several more algorithm tests will be performed with a particular vortical flow superimposed on a constant sign background shear [Mar90]. The precise formulation is as follows:

$$\begin{aligned}
{v^{back}}_{\theta} &= \frac{v_0 r(2r-R_{in}-R_{out})}{6.0(R_{out}-R_{in})^2} \\
v_r^{back} &= 0.0 \\
v_r^{pert} &= \frac{0.16 v_0(R_{out}-R_{in})}{R} sin^2(\pi\frac{r-R_{in}}{(R_{out}-R_{in})})(sin^{31}(\frac{4\theta+\pi}{4})cos(\frac{2\theta+\pi}{4}) - sin^{31}(\frac{2\theta-\pi}{4})cos(\frac{2\theta-\pi}{4}) \\
v_\theta^{pert} &= \frac{-0.02 v_0(R_{out}-R_{in})}{R} sin(pi\frac{r-R_{in}}{(R_{out}-R_{in})})cos(\pi\frac{r-R_{in}}{(R_{out}-R_{in})})(sin^{32}(\frac{2\theta+\pi}{4}) - sin^{32}(\frac{2\theta-\pi}{4})) \\
\vec{u} &= \vec{v}^{back} + \vec{v}^{pert}
\end{aligned} \tag{3.11}$$

This field consists of a background shear (shown in figure 3.2) with two smooth vortices of opposite sign superimposed. An illustration of this field is given in figure 3.3, contrasting the perturbational and total fields. Subtracting off the background shear makes the situation much clearer, as the perturbation in the total field is completely swamped by the background shear flow.

3.2.1 Convergence Test

Bell, Colella, and Glaz [BCG89] perform a truncation error analysis that shows that this class of projection methods is second order in space and time. This is not sufficient, however to produce numerical results which converge to second order with grid refinement. Numerical stability is also necessary. Since analytical stability analysis of nonlinear problems is not possible in general, we must perform a grid refinement study to calculate the numerical rate of convergence. If the algorithm is found to converge to second order, we may conclude that it is sufficiently numerically stable. With this in mind, we perform a convergence test to measure the algorithm's spatial and temporal accuracy. The initial condition is given by the pair of smooth vortices imposed on a background shear (the algebraic formula for the velocity field is given in 3.11.

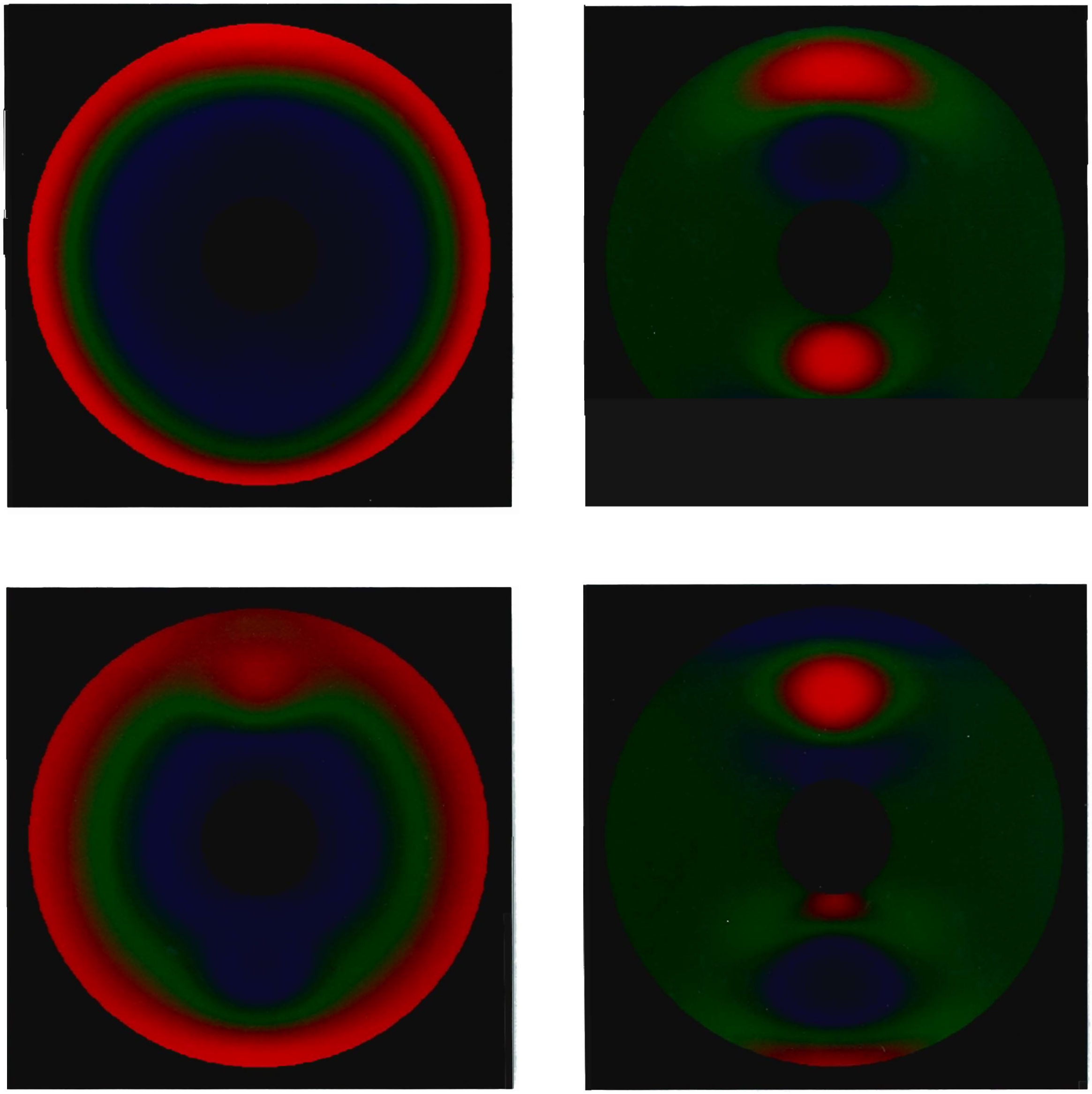


Figure 3.3: Oppositely signed vortices superimposed on a background shear. Maximums are in red, minimums in blue. The upper left is the total azimuthal velocity (max. 0.108 m/s, min. -0.043m/s). The upper right is the perturbational azimuthal velocity (max. 0.017m/s, and min. of -0.017m/s). The lower left picture is the total vorticity (max. 0.475/s, min. -0.211/s). The lower right is the perturbational vorticity (max. 0.156/s, min. -0.156/s).

Variable	Value
CFL	0.5
Ω (for SW test)	0.5/s
β (for QG test)	0.5/(m s)
R_{in}	0.33m
R_{out}	1.33m
v_0	0.5m/s

Table 3.3: Parameters used in quasigeostrophic and shallow-water convergence tests.

Var	$p(L_1)$	$p(L_2)$	$p(L_\infty)$
ω	2.27	2.04	1.49
u_r	2.43	2.41	2.06
u_θ	2.76	2.56	2.43

Table 3.4: Quasigeostrophic convergence test using several numerical norms for time= 5 s = 0.4 rotation times.

The flow is run out to a specified time for three different grids and the results are compared to compute the order of accuracy of the scheme. To understand this, consider in one dimension numerical solutions with an order of accuracy of p and with grid spacings of h, 2h, and 4h (u^h, u^{2h}, u^{4h}) and compare these with the exact solution (u^E). Assuming that the solution is a smooth function of space,

$$u^h{}_{i,j} = u^E(ih) + C(ih)h^p + 0(h^{p+1}) \tag{3.12}$$

where C is a smooth function. From equation 3.12 we may estimate p as follows:

$$p \approx \frac{\ln(\frac{||u_{2h}-u_{4h}||}{||u_h-u_{2h}||})}{\ln(2)} \tag{3.13}$$

The convergence test is calculated with both L_2 and L_{inf} norms. The parameters used for the convergence test are given in table 3.3. Rotation times are calculated using equation 3.8 for the shallow-water case and equation 3.9 for the quasigeostrophic case.

The results of the calculation using the L_2 norm are shown in tables 3.5 and 3.6.

Var	$\|\|32x160-16x80\|\|_2$	$\|\|64x320-32x160\|\|_2$	$p(L_2)$
ω	$1.889\cdot10^{-3}$	$4.582\cdot10^{-4}$	2.04
u_r	$5.244\cdot10^{-5}$	$9.725\cdot10^{-6}$	2.43
u_θ	$2.082\cdot10^{-4}$	$3.068\cdot10^{-5}$	2.56

Table 3.5: Quasigeostrophic convergence test using several numerical variables and the L_2 norm for time= 5 s = 0.4 rotation times.

Var	$\|\|32x160-16x80\|\|_2$	$\|\|64x320-32x160\|\|_2$	$p(L_2)$
ω	$1.996\cdot10^{-3}$	$4.247\cdot10^{-4}$	2.23
u_r	$7.535\cdot10^{-5}$	$1.616\cdot10^{-5}$	2.30
u_θ	$2.443\cdot10^{-4}$	$4.550\cdot10^{-5}$	2.42

Table 3.6: Shallow-water convergence test using several numerical variables and the L_2 norm for time= 5 s = 0.4 rotation times.

Var	$p(L_1)$	$p(L_2)$	$p(L_\infty)$
ω	2.39	2.23	1.94
u_r	2.22	2.30	2.87
u_θ	2.26	2.42	2.44

Table 3.7: Shallow-water convergence test using several numerical norms for time= 5 s= 0.4 rotation times.

Var	$\|\|32x160-16x80\|\|_2$	$\|\|64x320-32x160\|\|_2$	$p(L_2)$
ω	$9.684\cdot10^{-3}$	$3.324\cdot10^{-3}$	1.54
u_r	$3.571\cdot10^{-4}$	$8.991\cdot10^{-5}$	1.99
u_θ	$1.009\cdot10^{-4}$	$2.787\cdot10^{-4}$	1.86

Table 3.8: Shallow-water convergence test using several numerical variables and the L_2 norm for time= 25 s = 2.0 rotation times.

Var	$p(L_1)$	$p(L_2)$	$p(L_\infty)$
ω	1.81	1.54	0.78
u_r	2.17	1.99	1.43
u_θ	1.95	1.86	1.28

Table 3.9: Shallow-water convergence test using several numerical norms for time= 25 s = 2.0 rotation times.

Var	$\|\|32x160 - 16x80\|\|_2$	$\|\|64x320 - 32x160\|\|_2$	$p(L_2)$
ω	$7.601 \cdot 10^{-3}$	$2.465 \cdot 10^{-3}$	1.62
u_r	$1.979 \cdot 10^{-4}$	$3.683 \cdot 10^{-5}$	2.43
u_θ	$8.611 \cdot 10^{-4}$	$1.510 \cdot 10^{-4}$	2.51

Table 3.10: Quasigeostrophic convergence test using several numerical variables and the L_2 norm for time= 25 s = 2.0 rotation times.

Var	$p(L_1)$	$p(L_2)$	$p(L_\infty)$
ω	1.88	1.62	1.05
u_r	2.53	2.43	2.10
u_θ	2.50	2.51	1.93

Table 3.11: Quasigeostrophic convergence test using several numerical norms for time= 25 s = 2.0 rotation times.

A variety of quantities are used for the designated variable. The grid refinement study was also performed using the L_1 and the L_∞ norm. These results are shown in tables 3.4 and 3.7. Predictably, the more stringent max norm shows a lower order of accuracy. Still the scheme (in both the quasigeostrophic and shallow-water regimes) is clearly second order in the velocity by any reasonable measure.

Bell, Colella, and Glaz ([BCG89]) got for similar refinements orders of accuracy between 1.85 and 2.09 (they use L_2 norms). Clearly the results for this algorithm (shown in table 3.5) compare well to these. We can therefore conclude that we have shown second order convergence and that the algorithm exhibits sufficient numerical stability to make the observed accuracy match the accuracy predicted by the truncation error analysis. These are convergence tests for *short* times. Convergence tests at far longer times require much more careful consideration about the role of numerical dissipation in convergence to steady-state solutions. Long-time convergence tests will also use norms that take into account that the azimuthal coordinate is arbitrary. These tests are done in chapter five.

3.2.2 Quasigeostrophic Mean Rotation Test

The next test of the algorithm is to show that the algorithm respects the fact that the physics of the quasigeostrophic flow are independent any mean rotation component. Analytically, physics of a two-dimensional, quasigeostrophic, infinite Rossby deformation-radius flow is independent of the mean component of the rotation (Ω_0) because

$$F_{s,0} = 2\Omega_0 \times \vec{u} \tag{3.14}$$

is a pure gradient when ($\nabla \cdot \vec{u} = 0$).

To test the algorithm in this respect, the same two oppositely signed vortices imposed on a background shear problem is used (the algebraic formula for the velocity field is given in 3.11. The grid size is 32 × 160 and flow parameters are contained in table 3.12.

Variable	Value
CFL	0.5
Ω_0	0.5/s
β	0.5/(m s)
R_{in}	0.33m
R_{out}	1.33m
v_0	0.5m/s

Table 3.12: Parameters used in quasigeostrophic mean rotation test.

	Max/Min u_θ (m/s)	Max/Min u_θ (m/s)
$\Omega_0 =$	0	0.5rad/s
t=5s	$1.1245 \cdot 10^{-1}/-4.0893 \cdot 10^{-2}$	$1.1245 \cdot 10^{-1}/-4.0850 \cdot 10^{-2}$
t=10s	$1.1485 \cdot 10^{-2}/-3.9546 \cdot 10^{-2}$	$1.1497 \cdot 10^{-2}/-3.9560 \cdot 10^{-2}$
t=15s	$1.1554 \cdot 10^{-1}/-3.8214 \cdot 10^{-2}$	$1.1583 \cdot 10^{-1}/-3.8384 \cdot 10^{-2}$
t=25s	$1\ 1.1786 \cdot 10^{-1}/-3.6401 \cdot 10^{-2}$	$1.1790 \cdot 10^{-1}/-3.6839 \cdot 10^{-2}$

Table 3.13: Azimuthal velocity using zero mean rotation and using a significant mean rotation in the quasigeostrophic case.

The problem is run twice, once with

$$F_s = 2(\Omega_0 + \beta r) \times \vec{u} + F_{geom} \tag{3.15}$$

and once with

$$F_s = 2(\beta r) \times \vec{u} + F_{geom} \tag{3.16}$$

Ideally the results of the two runs should be the same.

Results of this test are shown in table 3.13. As the metric of compliance, we shall use as ϵ_r the smallest difference between the velocity with a mean rotation ($\vec{u}_w$) and the velocity without a mean rotation ($\vec{u}_{wo}$) that we can numerically expect:

$$\epsilon_r \equiv \max(vu_w) - \max(\vec{u}_{wo}) \tag{3.17}$$

With exact arithmetic, this difference would be exactly zero. We use equation 3.10 to estimate ϵ_r. The number of time steps N_t is approximately 25 (for the t= 5s case) and the

	Max/Min u_r (m/s)	Max/Min u_r (m/s)
$\Omega_0 =$	0	0.5rad/s
t=5s	$1.1579 \cdot 10^{-2}/-1.0969 \cdot 10^{-2}$	$1.1583 \cdot 10^{-2}/-1.0927 \cdot 10^{-2}$
t=10s	$1.1127 \cdot 10^{-2}/-1.0124 \cdot 10^{-2}$	$1.1152 \cdot 10^{-2}/-1.0042 \cdot 10^{-2}$
t=15s	$9.7427 \cdot 10^{-3}/-8.7773 \cdot 10^{-3}$	$9.8572 \cdot 10^{-3}/-8.6785 \cdot 10^{-3}$
t=25s	$6.3530 \cdot 10^{-3}/-6.5809 \cdot 10^{-3}$	$6.7503 \cdot 10^{-3}/-6.5148 \cdot 10^{-3}$

Table 3.14: Radial velocity using zero mean rotation and using a significant mean rotation in the quasigeostrophic case.

solver tolerance ϵ_{sol} is $1 \cdot 10^{-8}$. Therefore we estimate ϵ_r to be approximately $1.4 \cdot 10^{-5}$ m/s for this case. Both the quasigeostrophic case and the shallow-water stay within $4.0 \cdot 10^{-5}$m/s. Pictures of these results are identical and will be shown in chapter 4. That these solutions stay so close to one another even though the grid is relatively coarse indicates that the algorithm does indeed respect the physics of quasigeostrophic flow.

Chapter 4

Results

4.1 Adverse and Prograde Vortices in Background Shear

The shallow-water equations (equation 1.3) and the quasigeostrophic equations both have as an exact solution any velocity of the form

$$\vec{u} = f(r)\hat{e}_\theta \tag{4.1}$$

When the initial condition is a small perturbation of equation 4.1, both systems of equations have also been shown to have stable solutions of coherent, persistent vortices that exhibit a distinguishing sign of perturbational vorticity. Swinney, et. al. [MSS93] showed this experimentally and Marcus [Mar84] supports this with his spectral calculations. Specifically, define the initial condition to be of the form

$$\vec{u} = f(r)\hat{e}_\theta + \epsilon\vec{u}^{\,pert} \tag{4.2}$$

The background shear is defined to be the following

$$\sigma_{back} = r\,\frac{\partial\frac{f(r)}{r}}{\partial r} \tag{4.3}$$

and the perturbational vorticity is defined in the usual way

$$\omega_{pert} = \frac{1}{r}\left(\frac{\partial(ru_\theta{}^{pert})}{\partial r} - \frac{\partial u_r{}^{pert}}{\partial\theta}\right) \tag{4.4}$$

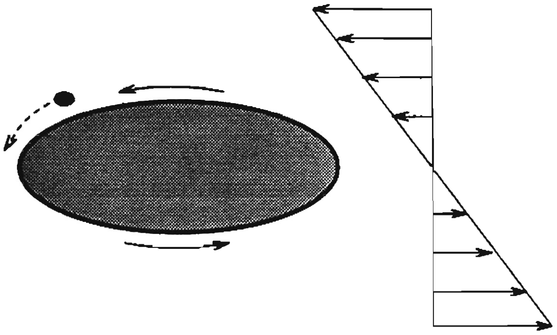


Figure 4.1: Vortex in prograde shear. A small blob perturbed away from a large vortex is pushed (due to the Biot-Savart effect of the large vortex) back toward the larger vortex where it can remerge.

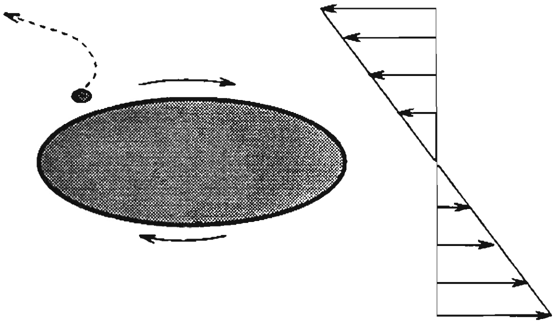


Figure 4.2: Vortex in adverse shear. A small blob perturbed away from the large vortex is stripped away.

Moore and Saffman [MS71] investigated the stability of two dimensional elliptical blobs of constant vorticity ($\omega_{pert} = \omega_0$) in constant background shear ($\sigma_{back} = \sigma_0$). In their linear stability analysis, they found that the ratio ($R = \frac{\sigma_0}{\omega_0}$) determines whether there is a solution. If this ratio is positive, (prograde shear) there is one stable steady-state solution. If this ratio is negative (adverse shear), and if ($|R| > 0.15$) there is no steady solution. If the ratio is negative and ($|R| < 0.15$) there are two solutions but one is linearly unstable and the other is unstable to finite perturbations (see Marcus [Mar93] for more details of this analysis).

Marcus [Mar93] provides a more intuitive picture, which is reproduced in figures

4.1 and 4.2. The crux of the idea is that a blob of vorticity perturbed away from the larger vortex will be stripped away in adverse shear and will be kept close to the larger vortex in prograde shear. Same-sign vortices in rotating flow tend to merge when they are close enough together so the perturbed blob remerges with the larger vortex. The difference in behavior is due to the difference in the direction of the Biot-Savart velocity induced upon the blob by the larger vortex.

4.1.1 Oppositely Signed Vortices in Background Shear

To test whether the present algorithm models this behavior correctly, two smooth vortices of opposite sign are put into a background shear of constant sign. The algebraic formula for the velocity field is given in equation 3.11. The grid size is 64 × 320 and all parameters are contained in table 4.1. Marcus [Mar90] uses a spectral method to investigate the behavior this very same pair of vortices in background shear for quasigeostrophic flow. These calculations are well-documented and for such a simple geometry, spectral methods are very accurate. The results of Marcus will therefore be used as a baseline to examine the performance of the present algorithm. Both the shallow-water case and the quasigeostrophic case will be compared to the Marcus results.

It should be noted that the pictures shown are ten percent perturbation of the background flow. Unless the background flow is subtracted off (to give the pictures shown), the dynamics of the flow are virtually invisible. These flows are also run out to the point where these subtle perturbational flows converge to some steady state. This often involves long integration times with respect to rotational times. Therefore these problems are a reasonable test as to whether projection methods are suitable for representing vortical flows which involve long integration times.

The time evolution of the solution is shown in figure 4.3 for the quasigeostrophic case and shown in figure 4.5 for the shallow-water case. We present results in terms of

Variable	Value
v_0	0.5m/s
$\beta(forQG)$	0.5/(m s)
$\Omega(forSW)$	0.5rad/s
R_{in}	0.33m
R_{out}	1.33m
S (bottom slope for SW)	0.05m/m
H_0 (mean depth for SW)	0.1m

Table 4.1: Parameters used in the opposite sign vortices in background shear problem.

rotation times. A rotation time for the shallow-water case is defined in the obvious way:

$$T_{rot}^{SW} = 2\frac{\pi}{\Omega} \tag{4.5}$$

For the quasigeostrophic case, any mean rotation does not affect the physics of the flow. In this case, the rotation is defined with the Coriolis parameter slope β.

$$T_{rot}^{QG} = 2\frac{\pi}{\beta((R_{out} - R_{in}))} \tag{4.6}$$

The adverse vortex (blue) gets torn apart almost immediately. The prograde vorticity stays coherent and merges into one large and one small vortex. Both the time evolution of the solution and the steady state solution agree well with Marcus' results. The shallow-water example represents the solution for slow rotation times compared with velocity time scales (the Rossby number of the shallow-water solution is of order unity). The quasigeostrophic solution represents fast rotation times with respect to velocity time scales (the limit of vanishing Rossby number). Still the two solutions are qualitatively the same. This means the background shear, rather than the rotation, is really driving the flow.

4.1.2 Shear Layers in Constant-Sign Background Shear

The previous example shows that the algorithm models well the phenomenon of a distinguishing sign of potential vorticity in rotating flows with background shear. A

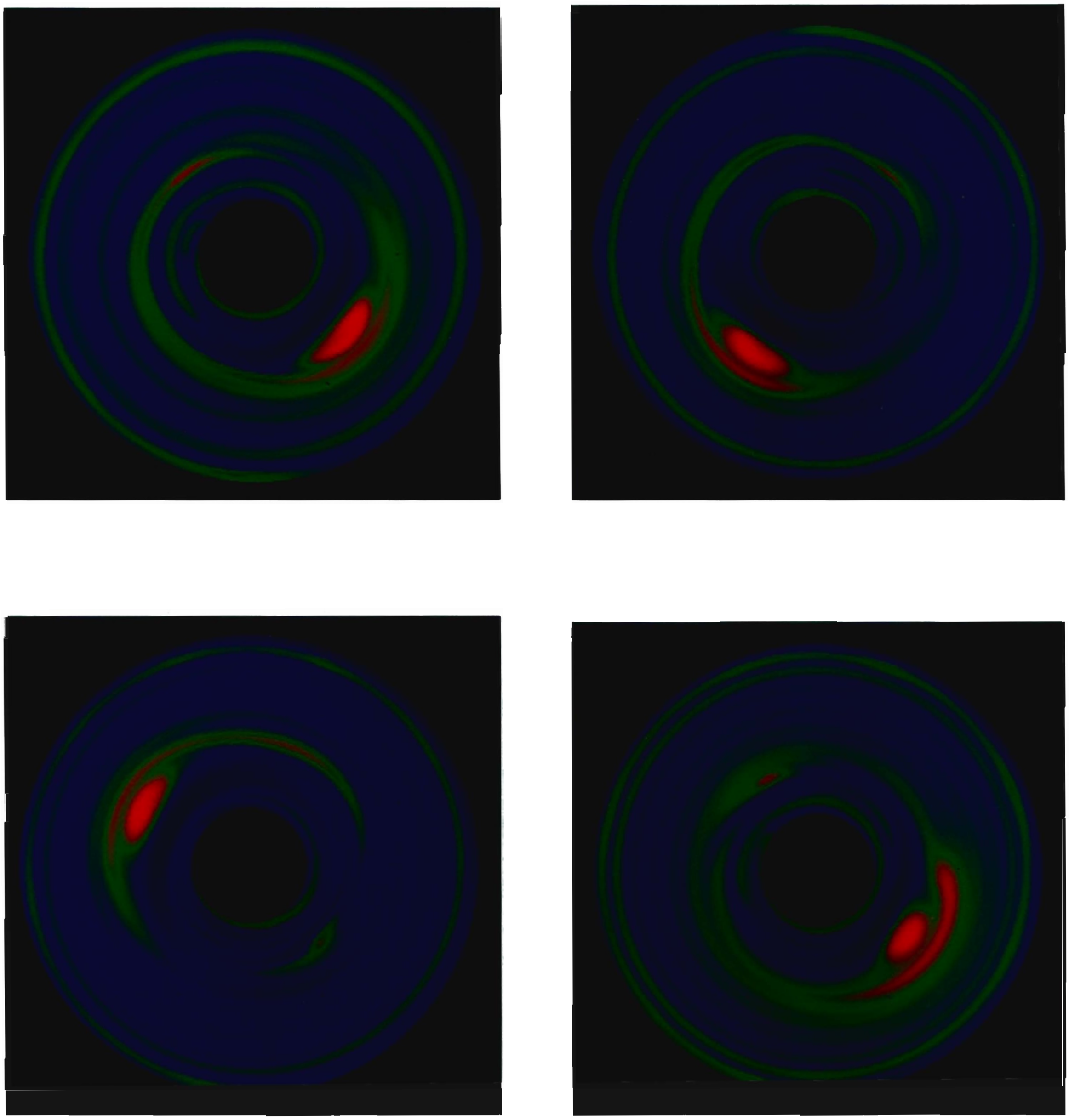


Figure 4.3: Perturbational vorticity of the oppositely signed vortices in background shear (QG) problem for a 64 × 320 grid at times (left to right, top to bottom) 0s, 25s, 50s, 100s. This is equivalent to 0, 1.98, 3.97, and 7.95 rotational times (see text for definitions). Max. (in red) is 0.156/s, min. (in blue) is -0.156/s.

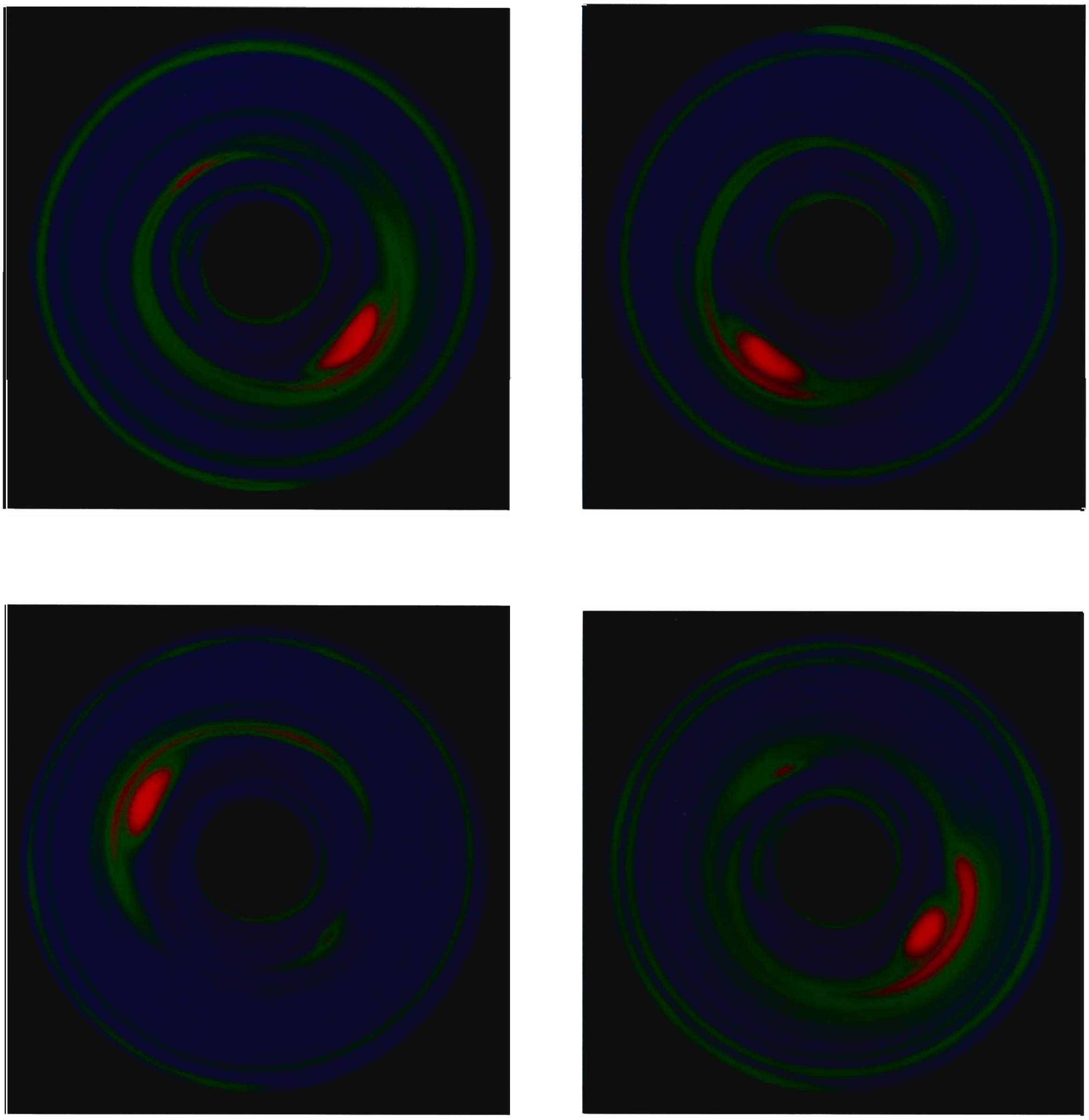


Figure 4.4: Perturbational vorticity of the oppositely signed vortices in background shear (QG) problem for a 64 × 320 grid at times (left to right, top to bottom) 240s, 280s, 320s, 400s. This is equivalent to 19.1, 22.2, 25.4, and 31.8 rotational times (see text for definitions). Max. (in red) is 0.156/s, min. (in blue) is -0.156/s.

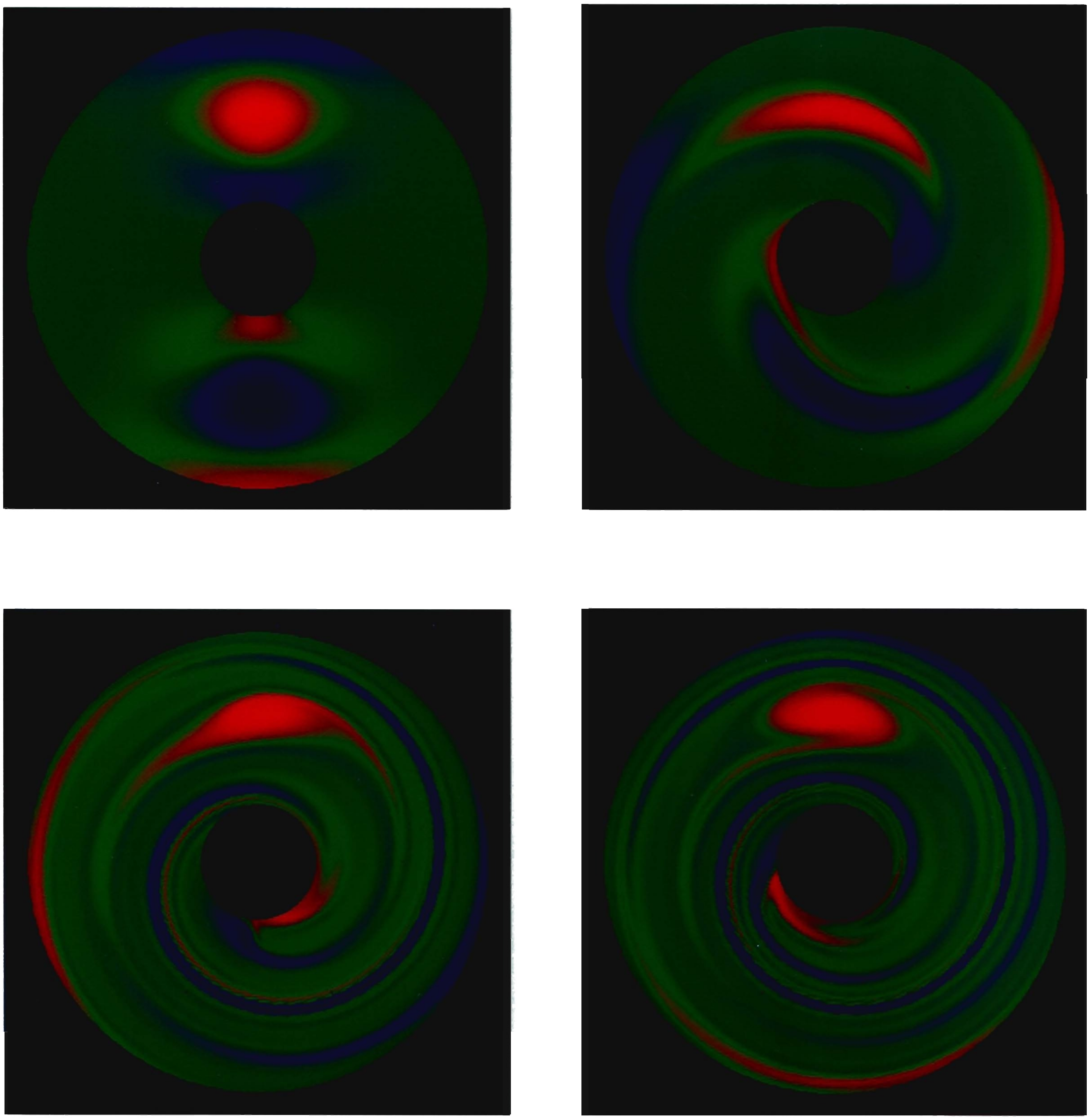


Figure 4.5: Perturbational vorticity of the oppositely signed vortices in background shear (shallow water) problem for a 64 × 320 grid at times (left to right, top to bottom) 0s, 20s, 60s, 100s. This is equivalent to 0, 1.59,4.77, and 7.95 rotational times (see text for definitions). Max perturbational vorticity (in red) is 0.156/s, min (in blue) is -0.156/s.

more striking example of this phenomenon is shown with shear layers superimposed over a background shear. A perturbation in the form of a shear layer is used instead of vortices. The behavior of the perturbational shear layer is determined by whether it has the same or opposite sign vorticity as that of the background shear flow.

A shear layer in a background shear flow can behave in one of two ways. If the perturbational shear layer has the opposite sense of rotation as the background shear flow (adverse background shear), then the perturbational layer is torn apart by the background shear in the same way that vortices in adverse shear are torn apart. If the perturbational shear layer is of the same sign as the background shear (favorable or prograde background shear), the perturbational shear layer rolls up due to Kelvin-Helmholtz instability. After rolling up, the resultant vortices merge into one. The favorable background shear case is studied extensively by Marcus [Mar90].

The present investigation includes both the prograde and adverse shear layers. Given the following background shear profile:

$$\begin{aligned} v^{back}{}_{\theta} &= \frac{v_0 r(2r - R_{in} - R_{out})}{6.0(R_{out} - R_{in})^2} \\ v_r^{back} &= 0.0 \end{aligned} \tag{4.7}$$

the prograde shear layer perturbational velocity is given by:

$$\begin{aligned} v_r^{pert} &= \frac{0.0833 v_0 (R_{out} - R_{in})}{r}(\tanh(\frac{r - 0.833((R_{out} - R_{in}))}{(R_{out} - R_{in})}) - \tanh(\frac{R_{in} - 0.833((R_{out} - R_{in}))}{(R_{out} - R_{in})}) \\ v_\theta^{pert} &= 0.008 v_0 (\sin(3\theta) + \sin(4\theta)) \end{aligned} \tag{4.8}$$

and the adverse shear layer perturbational velocity is given by:

$$\begin{aligned} v_r^{pert} &= \frac{-0.0833 v_0 (R_{out} - R_{in})}{r}(\tanh(\frac{r - 0.833((R_{out} - R_{in}))}{(R_{out} - R_{in})}) - \tanh(\frac{R_{in} - 0.833((R_{out} - R_{in}))}{(R_{out} - R_{in})}) \\ v_\theta^{pert} &= 0.008 v_0 (\sin(3\theta) + \sin(4\theta)) \end{aligned} \tag{4.9}$$

In both cases total velocity is given by:

$$\vec{u} = \vec{v}^{\,back} + \vec{v}^{\,pert} \tag{4.10}$$

Graphs of the initial velocity and vorticity for the adverse shear layers are given in figure 4.9 and figure 4.10. Graphs of the initial velocity and vorticity for the prograde shear layers

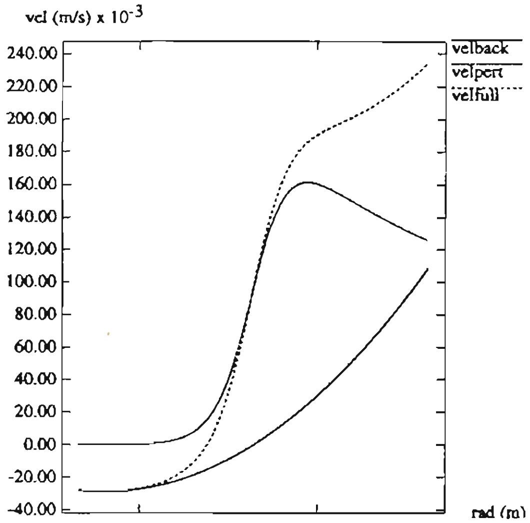


Figure 4.7: Initial velocity of prograde shear layer.

are given in figure 4.7 and figure 4.8. Both initial conditions were run on a 64 × 320 grid and the parameters used in the calculations are given in table 4.2.

The time evolution of perturbational vorticity for the prograde shear layer is given in figure 4.11 (quasigeostrophic case) and in figure 4.13 (shallow-water case). The results are given in terms of rotation times as defined in equations 4.5 and 4.6. Both the time evolution of the solution and the steady-state match very closely the results of Marcus in [Mar84] in the quasigeostrophic case (the case that Marcus solves). The layer rolls up and the separate vortices merge into one over many rotational times. To make certain axial symmetry is broken, the radial perturbation here has two wavelengths instead of Marcus' one; so the very early stages of the time evolution look somewhat different here.

The time evolution of perturbational vorticity for the adverse shear layer is given in 4.16 (quasigeostrophic run) and 4.15 (shallow-water run). The layer breaks up very quickly and the adverse vorticity disperses. This is consistent with Moore and Saffman's [MS71] analysis.

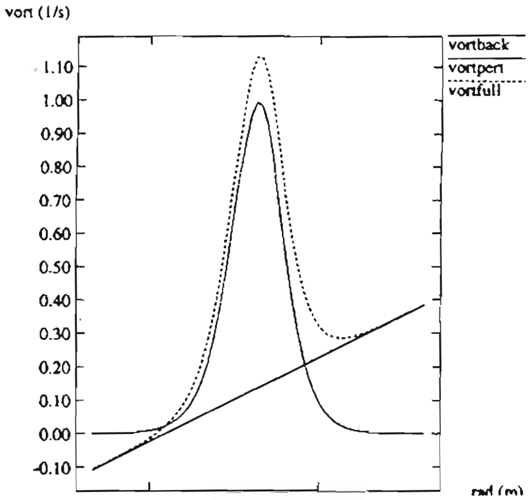


Figure 4.8: Initial vorticity of prograde shear layer.

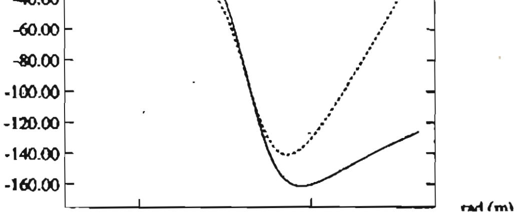


Figure 4.9: Initial velocity of adverse shear layer.

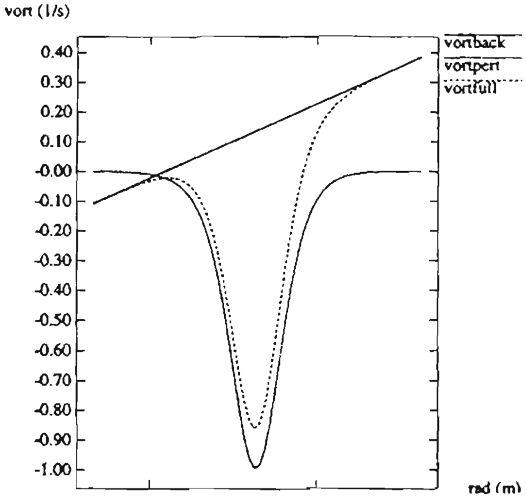


Figure 4.10: Initial vorticity of adverse shear layer.

Variable	Value
β (for QG test)	0.5/(m s)
Ω (for SW test)	0.5rad/s
R_{in}	0.33m
R_{out}	1.33m
v_0	0.5m/s
H_0 (mean depth for SW)	0.1m
S (bottom slope for SW)	0.05m/m

Table 4.2: Parameters used in shear layer runs (both adverse and prograde).

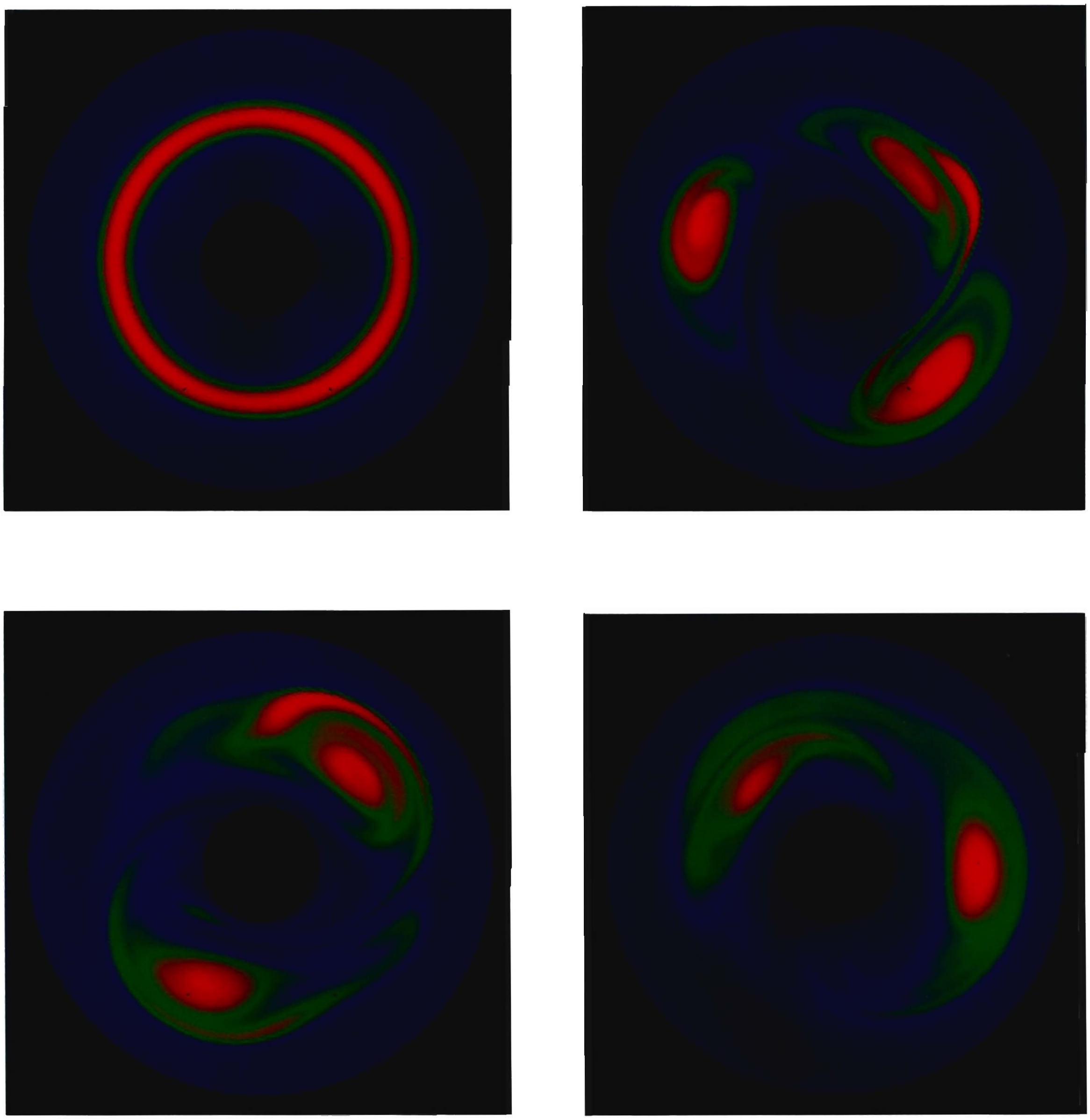


Figure 4.11: Perturbational vorticity of the prograde shear layer (quasigeostrophic run) for a 64 × 320 grid at times (left to right, top to bottom) 0s, 60s, 120s, 210s. This is equivalent to 0, 4.77, 9.55, and 16.7 rotational times (see text for definitions). Max perturbational vorticity (in red) is 0.53/s; Min (in blue) is -0.079/s

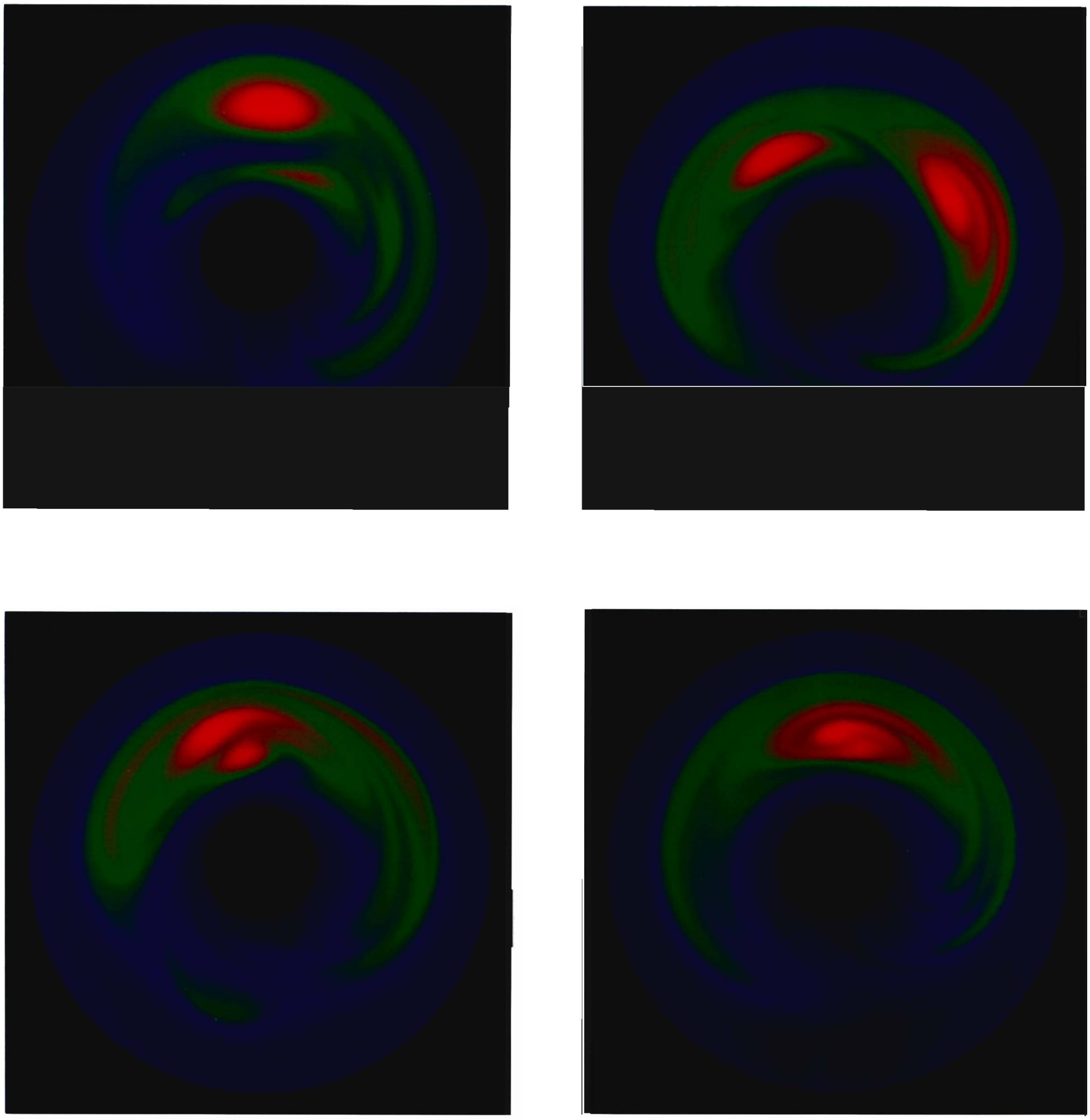


Figure 4.12: Perturbational vorticity of the prograde shear layer (quasigeostrophic run) for a 64 × 320 grid at times (left to right, top to bottom) 240s, 300s, 390s, 480s. This is equivalent to 19.1, 23.9, 31.0, and 38.2 rotational times (see text for definitions). Max perturbational vorticity (in red) is 0.53/s; Min (in blue) is -0.079/s

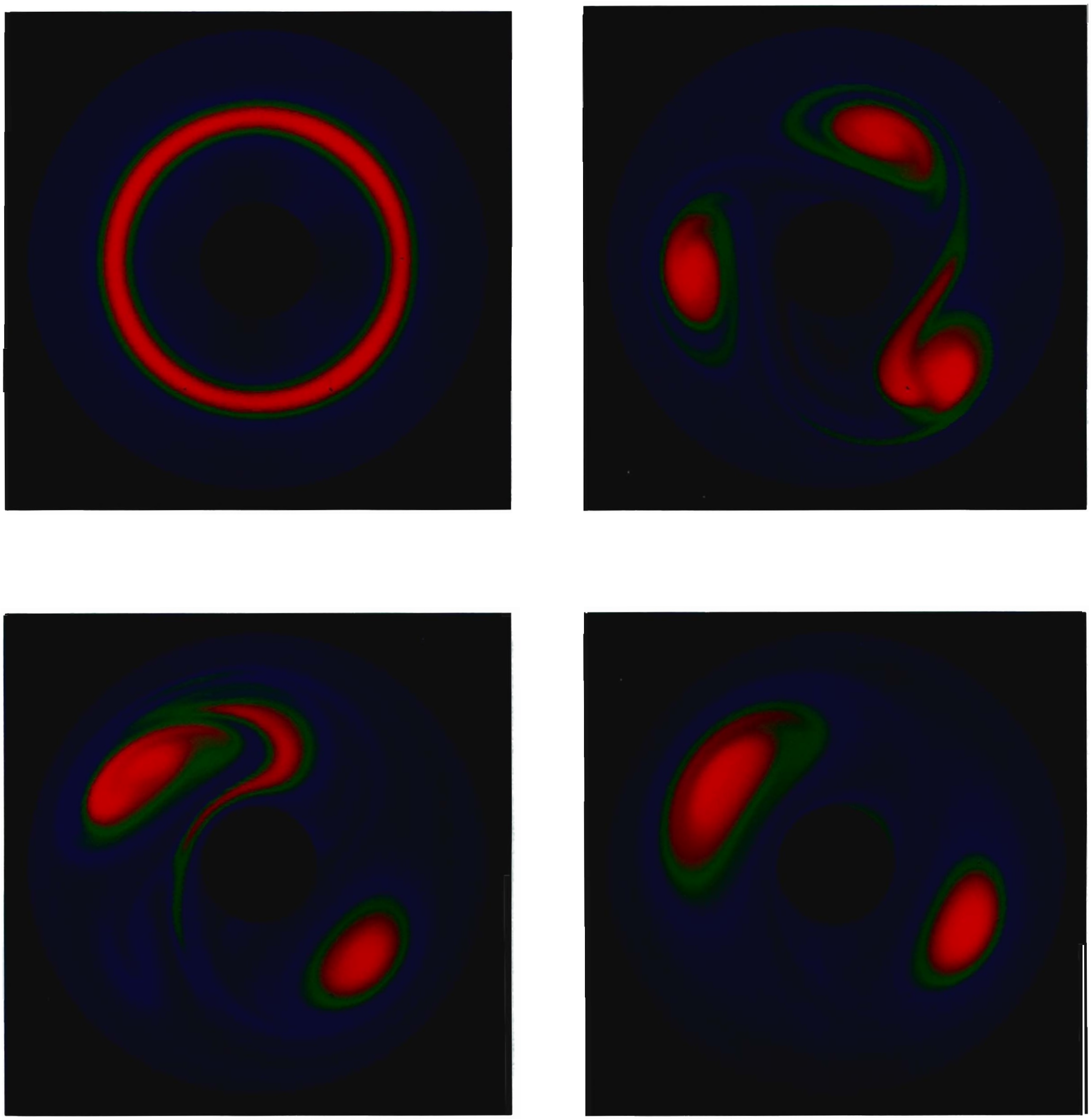


Figure 4.13: Perturbational vorticity of the prograde shear layer (shallow water run) for a 64 × 320 grid at times (left to right, top to bottom) 0s, 50s, 100s, 250s. This is equivalent to 0, 3.98, 7.95, and 19.9 rotational times (see text for definitions). Max perturbational vorticity (in red) is 0.53/s, min (in blue) is -0.079/s.

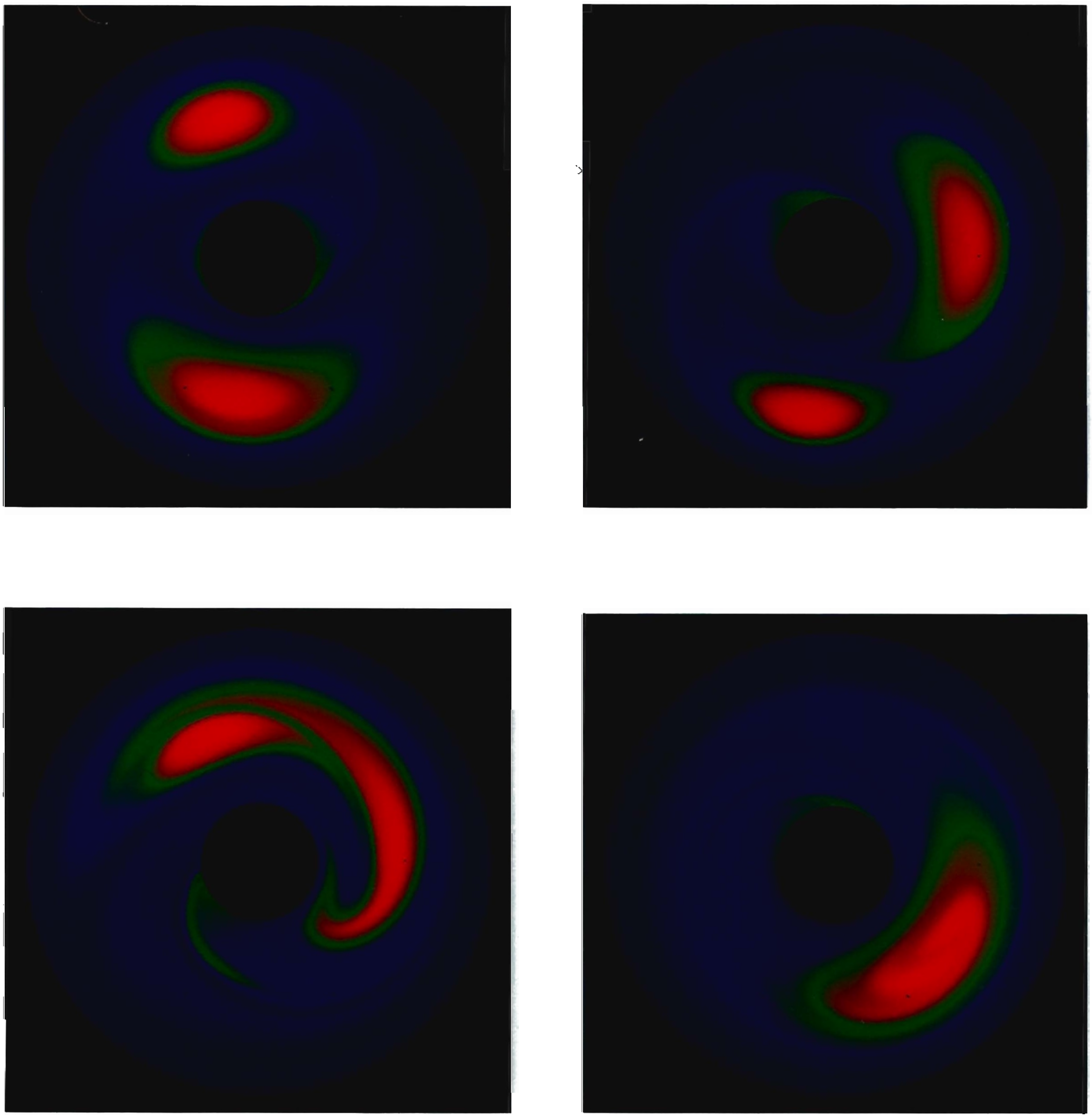


Figure 4.14: Perturbational vorticity of the prograde shear layer (shallow water run) for a 64 × 320 grid at times (left to right, top to bottom) 300s, 350s, 400s, 550s. This is equivalent to 23.8, 27.8, 31.8, and 43.7 rotational times (see text for definitions). Max perturbational vorticity (in red) is 0.53/s, min (in blue) is -0.079/s.

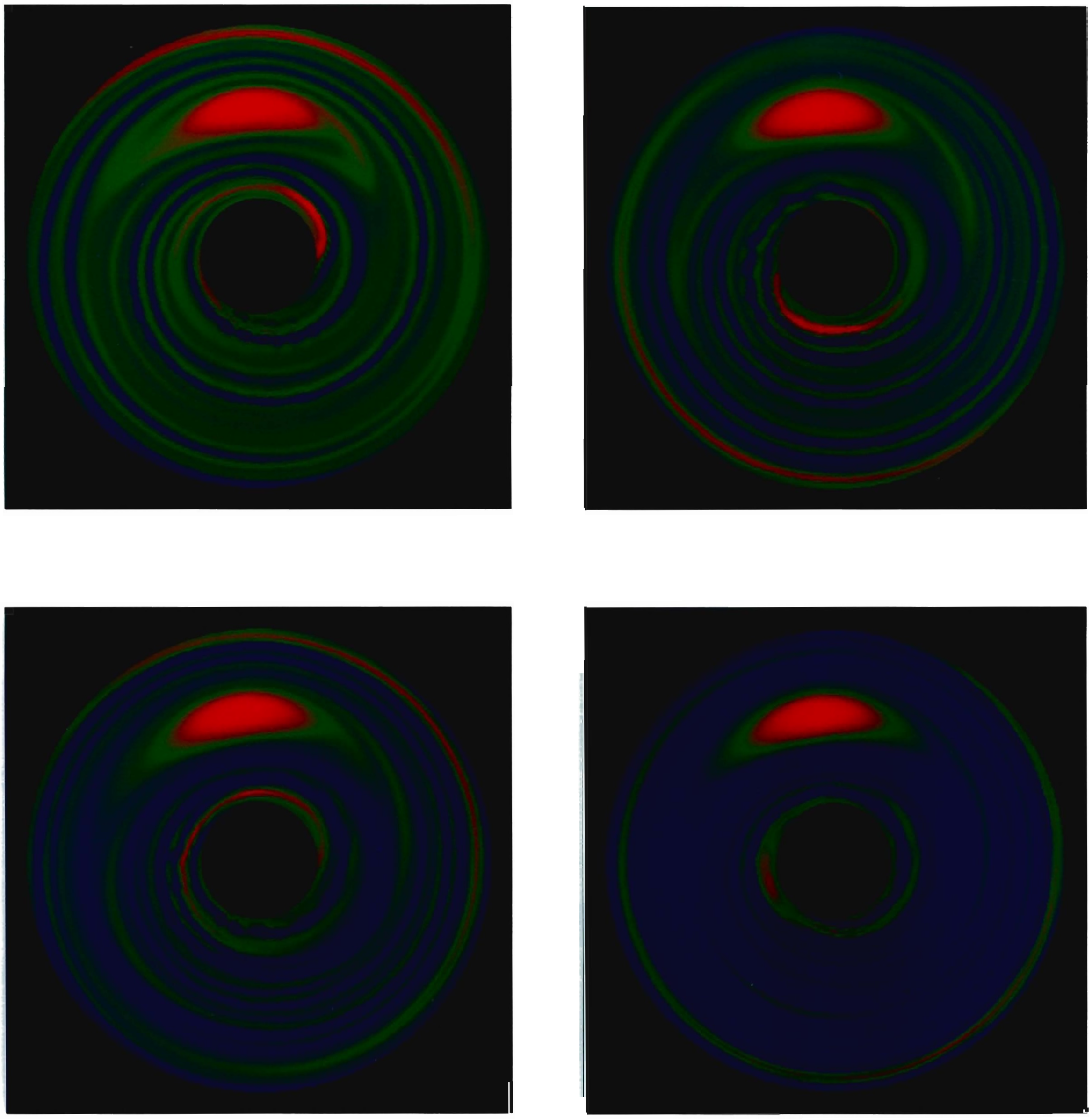


Figure 4.6: Perturbational vorticity of the oppositely signed vortices in background shear (shallow water) problem for a 64 × 320 grid at times (left to right, top to bottom) 140s, 180s, 220s, 280s. This is equivalent to 11.1, 14.3, 17.5, and 22.3 rotational times (see text for definitions). Max perturbational vorticity (in red) is 0.156/s, min (in blue) is -0.156./s.

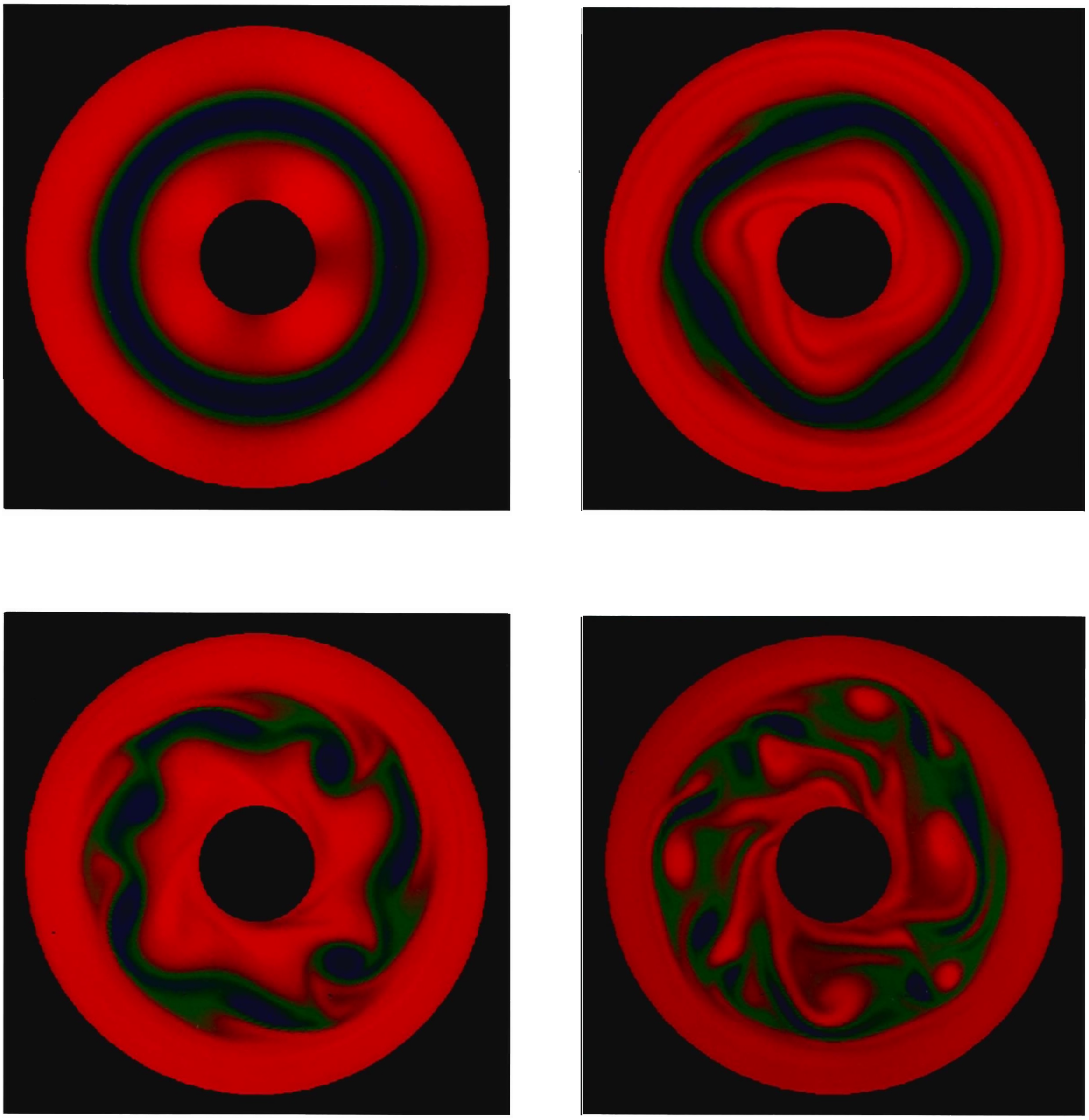


Figure 4.15: Perturbational vorticity of the adverse shear layer (shallow water run) for a 64 × 320 grid at times (left to right, top to bottom) 0s, 60s, 120s, 300s. This is equivalent to 0, 4.77, 9.54, and 23.9 rotational times (see text for definitions). Max perturbational vorticity (in red) (in red) is 0.72/s, min (in blue) is -0.53/s.

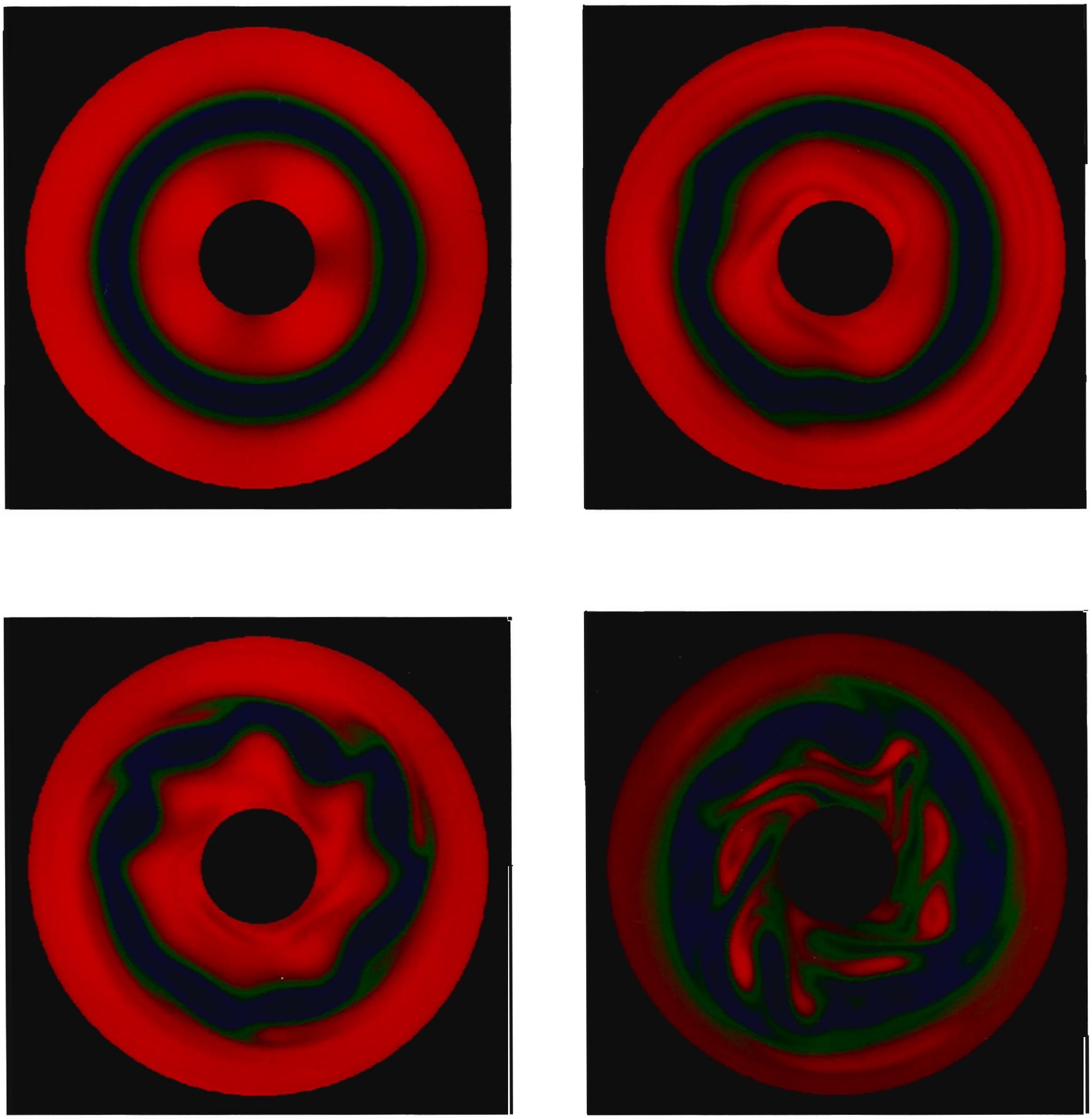


Figure 4.16: Perturbational vorticity of the adverse shear layer (quasigeostrophic run) for a 64 × 320 grid at times (left to right, top to bottom) 0s, 60s, 120s, 240s. This is equivalent to 0, 9.5, 19.1, and 38.2 rotational times (see text for definitions). Max perturbational vorticity (in red) is 0.72/s, min (in blue) is -0.53/s.

Variable	Value
v_0	0.5m/s
R_{in}	0.33m
R_{out}	1.33m
s	$0.1m/m$
H_O	$0.187m$

Table 4.3: Parameters used in the oppositely signed vortex problem in background shear with no rotational forcing.

4.1.3 Vortices in Background Shear Without Rotational Forcing

Though satisfying, the previous two sections' results are problematic. Since situations with very different rotational forcing were tried in both cases and the results were very similar, it could clearly be hypothesized that rotation is not really important in this situation.

Certainly it has been established that the algorithm performs vortex dynamics correctly. But if with this particular problem rotation is not important to the dynamics, it has not been established that the algorithm performs correctly in a situation where rotation drives the dynamics. Because the Moore and Saffman [MS71] analysis is not specifically for rotating flows, and also because slowly rotating shallow-water results and the quasigeostrophic results are so similar, one might suspect that it is the background shear that is driving the flow. Since the goal of the present work is to validate an algorithm suitable for modeling rotating flow, it is worthwhile to run the same problem without rotational forcing.

With this in mind, we run the oppositely signed vortices in background shear as a shallow-water problem with the rotational forcing set to zero. The algebraic formula for the velocity field is given in equation 3.11. The grid is 64 × 320 and all parameters are contained in table 4.3.

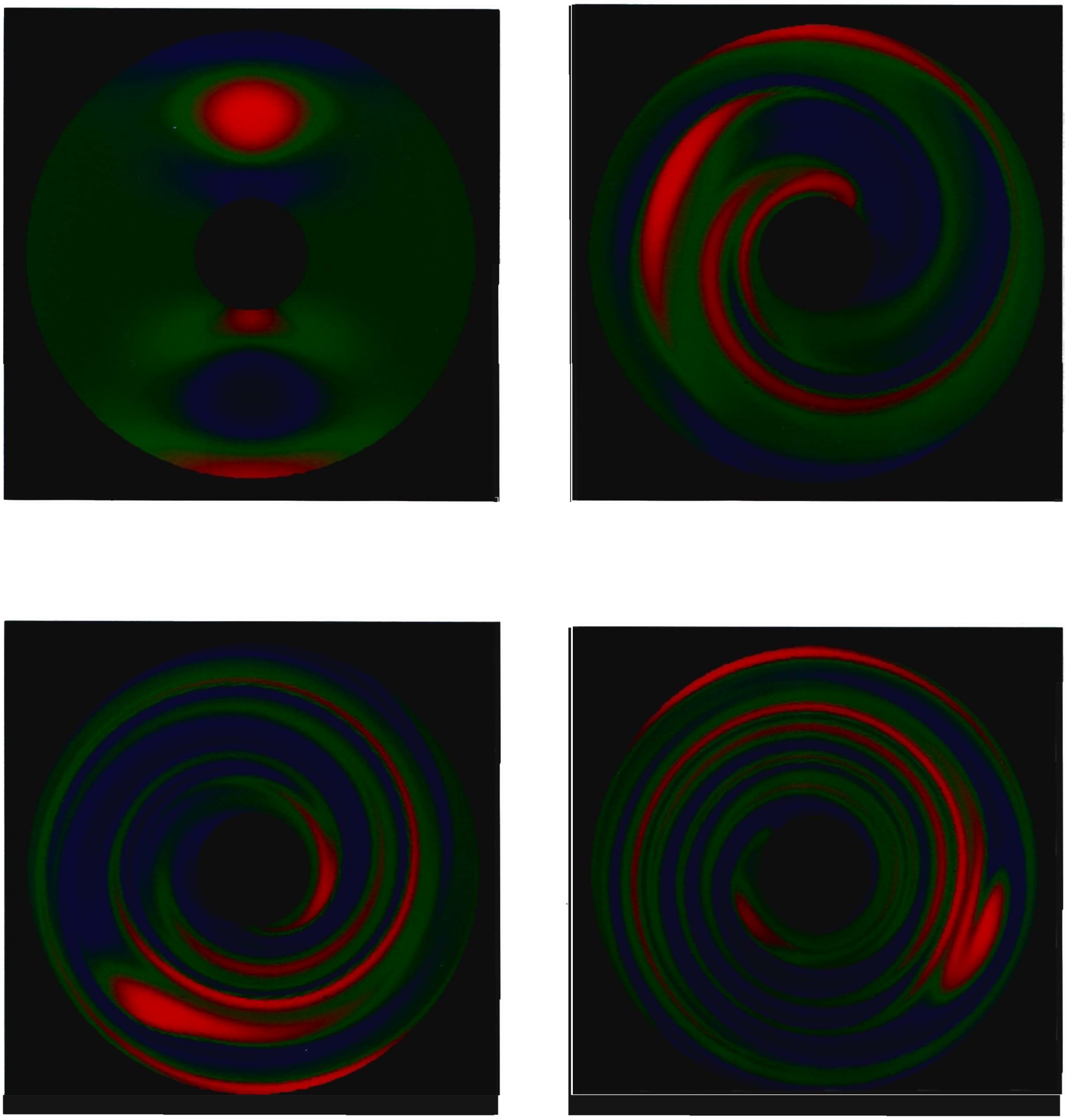


Figure 4.17: Perturbational vorticity of the oppositely signed vortices in background shear without rotational forcing for a 64 × 320 grid at times (left to right, top to bottom) 0s, 40s, 80s, 120s. Max perturbational vorticity (in red) is 0.156/s, min (in blue) is -0.156/s.

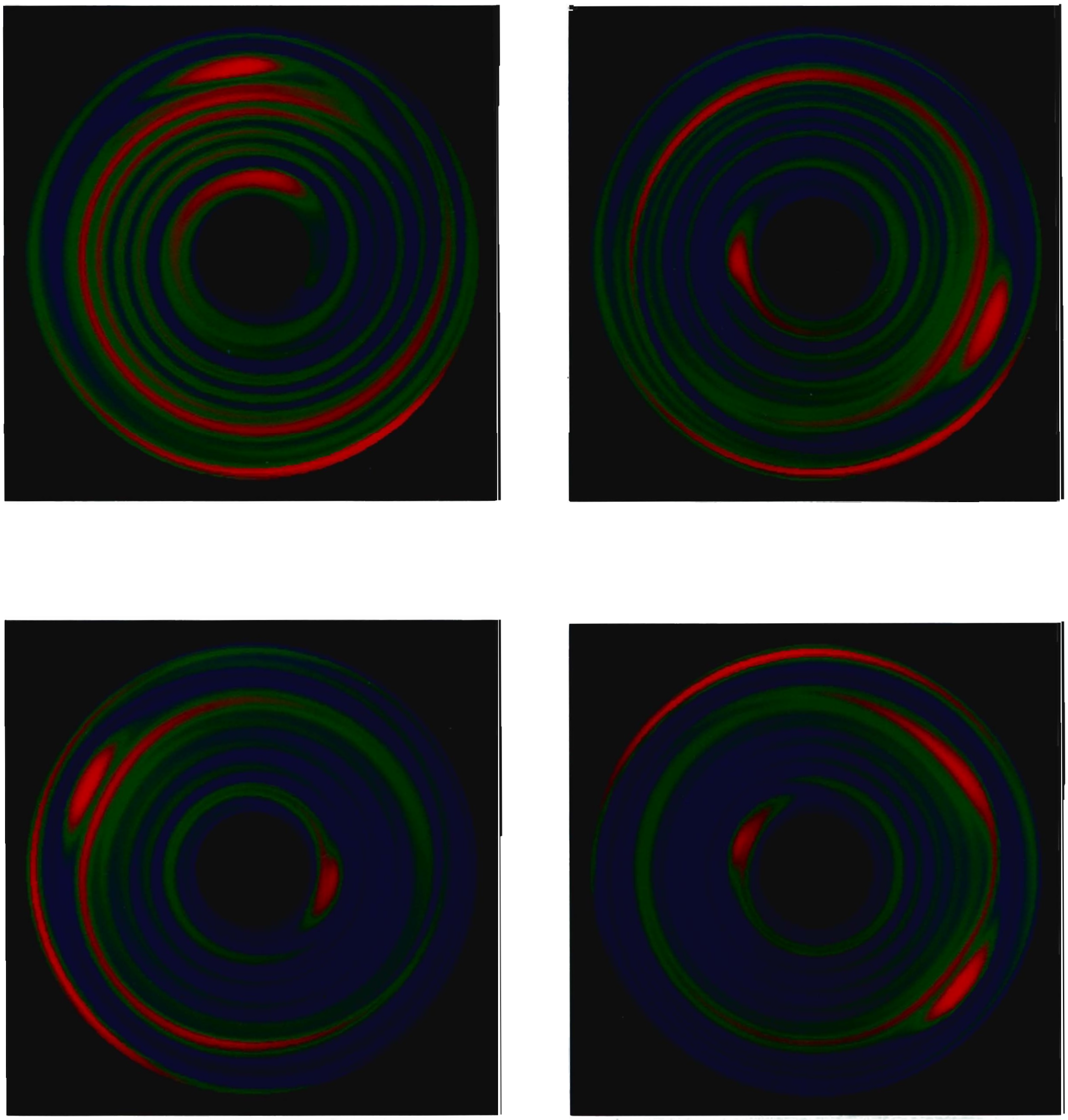


Figure 4.18: Perturbational vorticity of the oppositely signed vortices in background shear without rotational forcing for a 64 × 320 grid at times (left to right, top to bottom) 160s, 240s, 300s, 360s. Max perturbational vorticity (in red) is 0.156/s, min (in blue) is -0.156/s.

An illustration of the time history of the perturbational vorticity of the oppositely signed vortex in background shear initial condition is given in 4.17. The qualitative behavior of the solution is identical to both the shallow-water (with rotational forcing) case and the quasigeostrophic case. Clearly it is indeed the background shear that is forcing the solution. However, the different solutions (shallow-water with rotational forcing, quasigeostrophic, and shallow-water without rotational forcing) arrive at a steady state at very different time scales. Clearly the rotation facilitates the merger of vortices because without rotation, the vortices stay distinct much longer.

4.1.4 Stability of Shear Layers in Rotating Shallow Water

To give more confidence that the present algorithm performs correctly in a situation where rotation is very important, the stability of shear layers in rotating flow is now considered. Swinney, et. al. [SHS93] study extensively the stability of shear layers in rotating, viscous, shallow flows. Shear layers in nonrotating flow are unconditionally unstable but in rotating flow, the shear layer is stabilized somewhat by the rotation of the system coupled with viscous shear and Ekman pumping. Swinney, et al. find very good agreement with predictions of the neutral stability conditions of the problem done with a spectral calculation by Lee [Lee94].

Inviscid shear layers are always unstable even with rotation [Lee94]. Lee shows that rotation is important in determining the rate of this instability. With the current algorithm, since it is inviscid, it is not possible to reproduce the neutral stability curves of the viscous problem here. Lee, however, does produce growth rates of the perturbational solution in cases where the shear layers are unstable. Since the primary mode of the instability is a Kelvin-Helmholtz instability (which is well modeled by inviscid theory), the present algorithm is used to predict these growth rates and compare with the spectral calculations performed by Lee for the viscous case.

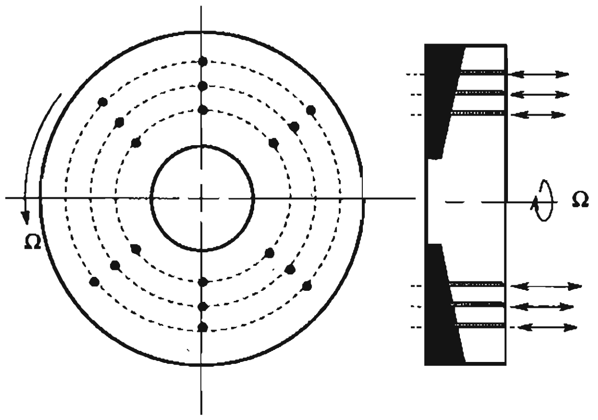


Figure 4.19: Schematic of the apparatus of the Swinney et. al. experiment. The real experiment has many more inlet and outlet ports. The experimentalist has control of which ports are operating and whether a particular port is acting as an inlet or outlet port.

Their experimental apparatus resembles figure 4.19. Pumps inject fluid into one azimuthal slit and remove the fluid from the system from a slit at a different radius. An azimuthal jet is created through the balance of this forcing with Ekman pumping and viscous dissipation. The full set of equations that Lee solves are as follows:

$$\begin{aligned} \frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} &= \nabla p + (\beta r \hat{e}_z) \times \vec{u} + \vec{F}_i + \nu \nabla^2 \vec{u} - 2E\vec{u} \\ F_i &= \frac{2\Omega}{Hr} \int_{R_{in}}^{r} W_{inj}(r') r' dr' \hat{e}_\theta \\ \nabla \cdot \vec{u} &= 0 \end{aligned} \tag{4.11}$$

These are the quasigeostrophic equations with forcing, viscosity, and Ekman dissipation. W_{inj} is the injection profile over the slit, which is assumed to be parabolic. β is the Coriolis parameter's slope, ν is the kinematic viscosity, and E is the inverse of the Ekman time scale and is given by [Lee94]:

$$E = (\frac{\Omega\nu}{H^2})^{\frac{1}{2}} \tag{4.12}$$

To generate the equilibrium profile of the velocity, azimuthal and time derivatives taves are set to zero and the resulting balance equation for the steady state azimuthal velocity v_{eq} is solved. That balance equation is given as follows:

$$\nu(\frac{1}{r}\frac{d}{dr}r\frac{d}{dr}-\frac{1}{r^2})v_{eq}=2Ev_{eq}-\frac{2\Omega}{Hr}\int\limits_{R_{in}}^{r}W_{inj}(r')r'dr'\hat{e}_\theta \tag{4.13}$$

The injection velocity profile W_{inj} is assumed to be parabolic and of course the mass out of one slit is balanced by the mass into the next. Therefore by specifying one pumping rate P, W_{inj} is determined to be following

$$W_{inj}(r)=\left(\begin{array}{ll} C_1(1-\frac{1}{L_p{}^2}(r-R_1)^2) & \text{if } (R_1-L_p)<r<(R_1+L_p) \\ C_2(1-\frac{1}{L_p{}^2}(r-R_2)^2) & \text{if } (R_2-L_p)<r<(R_2+L_p) \\ 0 \quad \text{otherwise} & \end{array}\right) \tag{4.14}$$

where R_1, R_2 are the radial locations of the slits, L_p is half the width of the slit, and the constants C_1, C_2 are given by the conservation constraint

$$\begin{aligned} 2\pi\int\limits_{R_1-L_p}^{R_1+L_p}W_{inj}(r')r'dr' &= P \\ 2\pi\int\limits_{R_2-L_p}^{R_2+L_p}W_{inj}(r')r'dr' &= -P \end{aligned} \tag{4.15}$$

To generate the initial azimuthal velocity profile, 4.13 is solved by using centered difference approximations for the derivatives and directly solving the resultant matrix. After v_{eq} is generated as described above, the unperturbed velocity field $\vec{u}_U$ is defined to be

$$\vec{u}_U=(0,v_{eq}) \tag{4.16}$$

and the initial velocity field is defined to be this unperturbed field with a perturbation of wavenumber m and magnitude ϵ:

$$\begin{aligned} \vec{u}_O(r,\theta) &= \vec{u}_U+\vec{u}_{pert} \\ \vec{u}_{pert} &= (\epsilon\sin(2\pi m\theta),0) \end{aligned} \tag{4.17}$$

We then run the code a small number of time steps (on the order of ten) to allow the perturbation to grow. It is this somewhat evolved solution that is used as the initial condition. The reason for this step is that, numerically at least, the growth rate of a mode might depend on the shape of that mode; this procedure allows the algorithm to pick out the most

Variable	Value
CFL	0.5
P	$6.0 \cdot 10^{-5} \frac{m^3}{s}$
Ω	$2\pi \frac{rad}{s}$
ν	$1.0 \cdot 10^{-5} \frac{m^2}{s}$
R_{in}	$0.108m$
R_{out}	$0.432m$
R_1	$0.189m$
R_2	$0.351m$
m	7
ϵ	0.1

Table 4.4: Parameters used to generate initial azimuthal velocity profile.

Variable	Value
CFL	0.5
t_f	$3.0s$
t_i	$1.0s$
β	$3.36/(ms)$

Table 4.5: Parameters used in quasigeostrophic growth rate investigation.

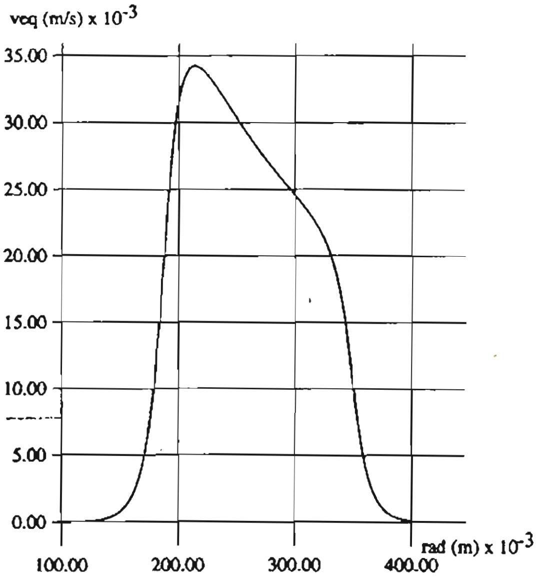
unstable mode and therefore solutions at different times can be compared more consistently. The wavenumber m is set to be the wavelength recorded in both Swinney's experiments and Lee's calculations as the most unstable wavelength for the viscous calculation. The parameters used in the present calculation to generate the unperturbed azimuthal velocity are given in 4.4. These parameters are chosen to match Swinney's experiment and Lee's calculations. A graph of the initial azimuthal velocity field is given in 4.20.

Once the initial condition is established, we run the code for a short time and compare the solution to the initial condition to calculate the growth rate of the solution. Following the example of Lee [Lee94], it is assumed that, for short times, the solution will be of the form

$$\vec{u} = \vec{u}_U + K \exp(im(\phi - ct)) \qquad (4.18)$$

Variable	Value
Ω	$6.28 rad/s$
cfl	0.5
t_f	$3.0s$
t_i	$1.0s$
s	$0.1m/m$
H_O	$0.187m$

Table 4.6: Parameters used in shallow-water growth rate investigation.



where K, ϕ are constants. It follows directly from equation 4.18 that the effective rate of growth c of a numerical solution is calculated as follows:

$$c = \frac{1}{m(t - t_s)} \ln\left(\frac{||\vec{u}(r, \theta, t) - \vec{u}_U||}{||\vec{u}(r, \theta, t_s) - \vec{u}_U||}\right) \tag{4.19}$$

where t_s is the starting time of the growth rate investigation. In the present investigation, both max and L_2 norms are used to calculate the growth rate c.

The investigation is divided into two parts, a quasigeostrophic run (whose parameters are given in table 4.5), and a shallow-water run (whose parameters are given in table 4.6). To assure that both runs are using the same relative magnitude of rotational forcing, the slope of the Coriolis slope parameter β in the quasigeostrophic run is related to the shallow-water bottom slope s as follows:

$$\beta = \frac{\Omega s}{H_O} \tag{4.20}$$

where H_O is the mean bottom depth.

Define the perturbational vorticity to be the curl of the perturbational velocity at a given time.

$$\omega_{pert} = \nabla \times (\vec{u} - \vec{u}_U) \tag{4.21}$$

The time evolution of ω_{pert} is given in figure 4.22 (in the quasigeostrophic case) and in figure 4.24 (in the shallow-water case) and the total vorticity's (that is, the curl of the full velocity field's) time evolution is given in figure 4.21 (in the quasigeostrophic case) and in figure 4.23 (for the shallow-water case). The perturbational vorticity graphs make clear that the initial perturbation is not the mode that grows. It is clearly justified to wait until the growing mode is established before beginning the growth rate calculation. It is also clear that, once the growing mode is established, it remains dominant throughout the calculation.

The numerical growth rates are given in tables 4.7 (for the quasigeostrophic run) and 4.8 (for the shallow-water run). The value of c found by Lee [Lee94] is $5.35 \cdot 10^{-2}$/s.

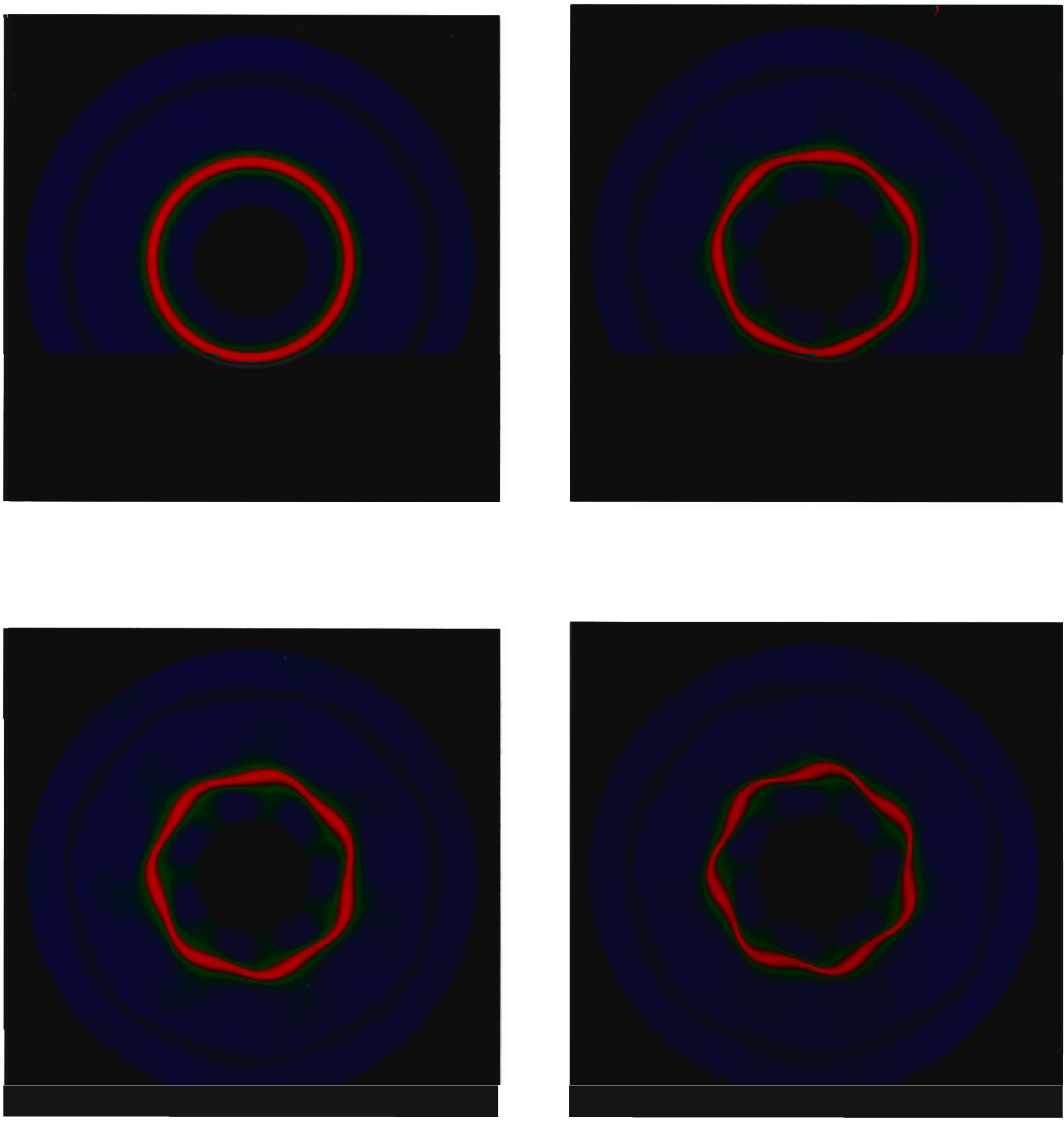


Figure 4.21: Total vorticity (the cross product of $\vec{u}$) of the azimuthal jet (quasigeostrophic case) as it evolves in time for a 128 × 640 grid at times (left to right, top to bottom) 0s, 1s, 2s, 3s. This corresponds to 0, 1, 2 and 3 rotational times. Max. vorticity (in red) is 1.31/s, min (in blue) is -0.71/s.

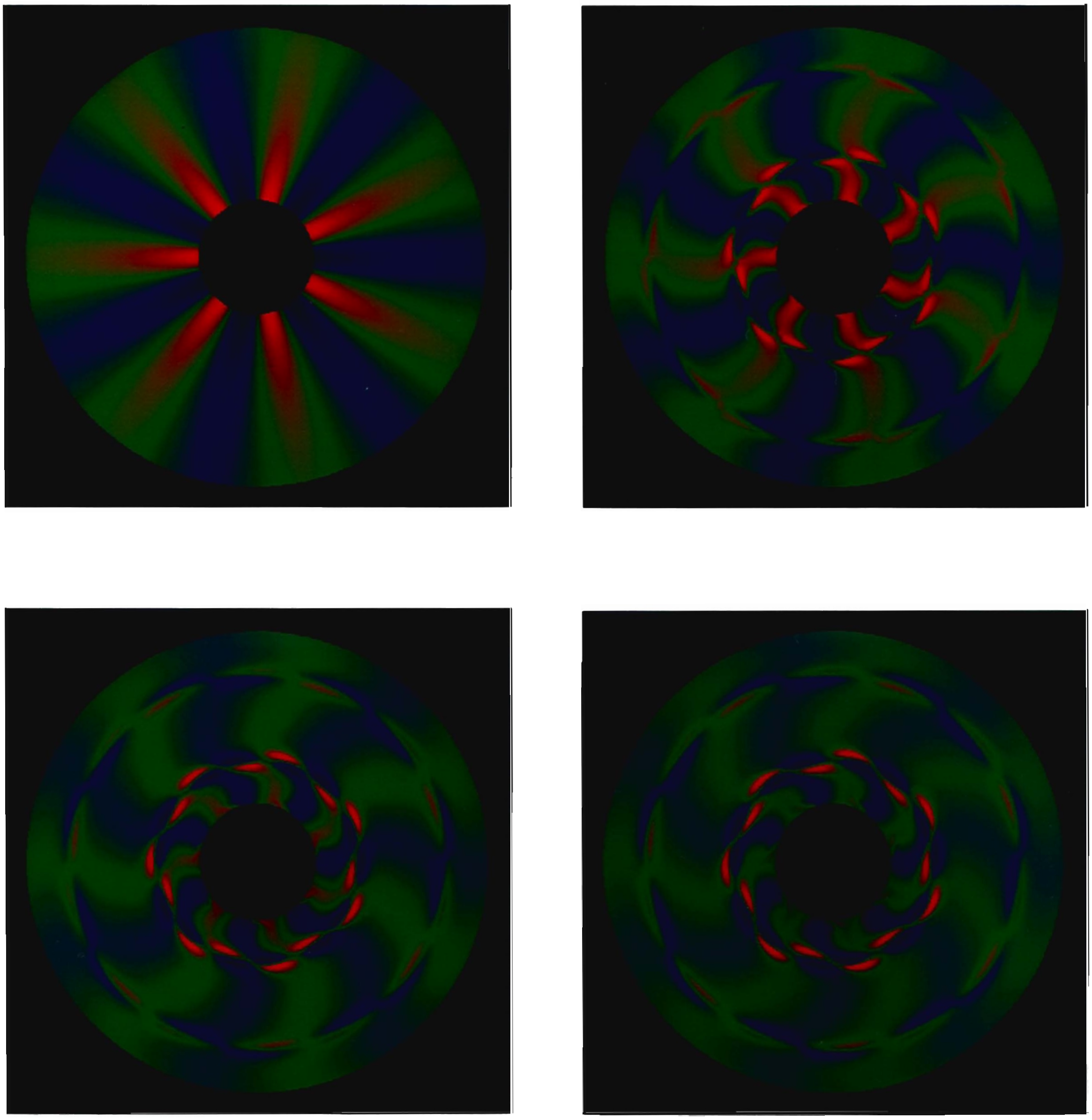


Figure 4.22: Perturbational vorticity (the cross product of $\vec{u}-\vec{u}_U$) of the azimuthal jet (quasigeostrophic case) as it evolves in time for a 128 × 640 grid at times (left to right, top to bottom) 0s (max. 0.21/s, min. -0.21/s), 1s (max. 0.25/s, min. -0.25/s), 2s (max. 0.45/s, min. -0.42/s), 3s (max. 0.63/s, min -0.58/s). This corresponds to 0, 1, 2 and 3 rotational times. Maximums in red, minimums in blue, midpoint in green.

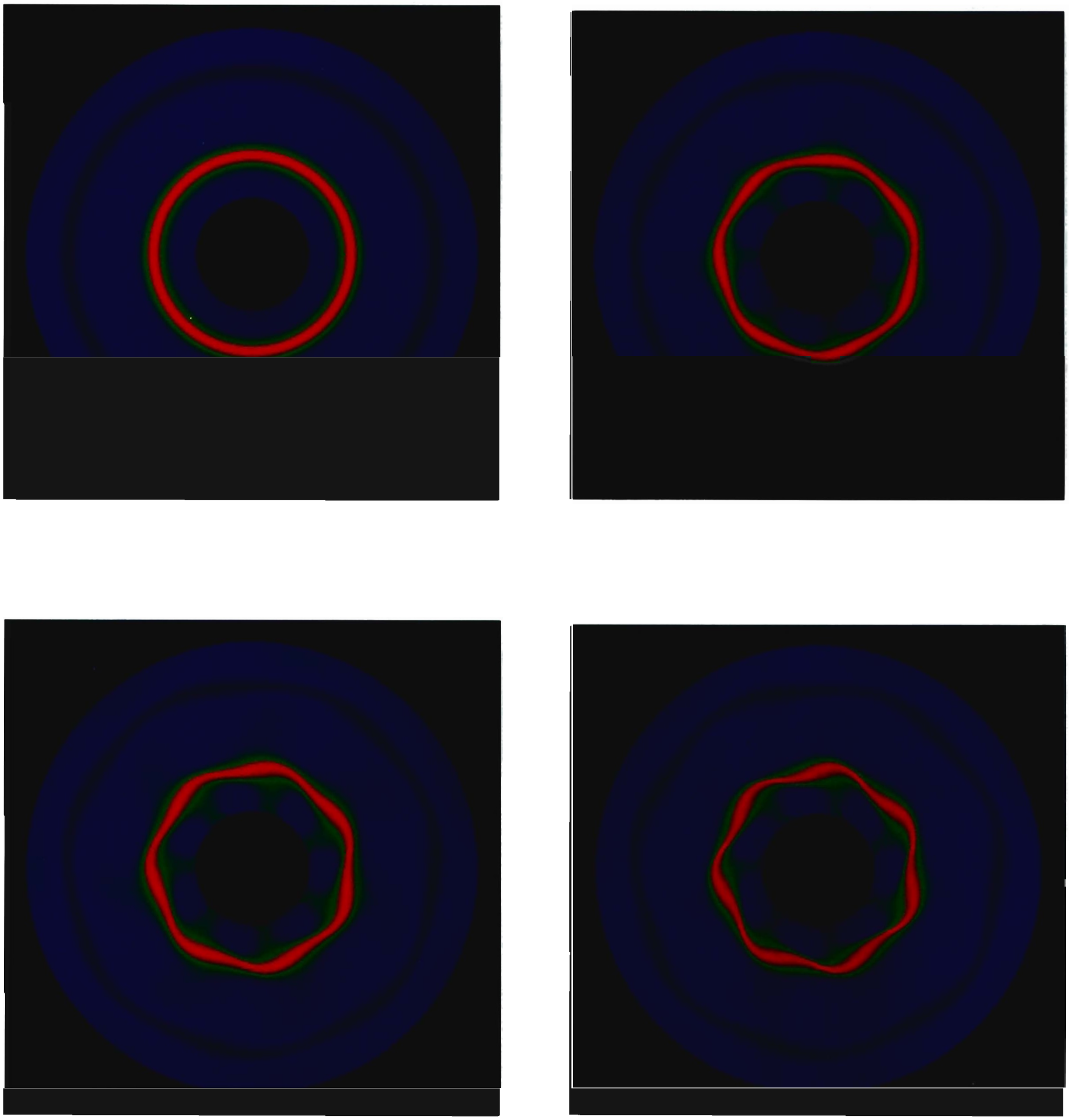


Figure 4.23: Total vorticity (the cross product of $\vec{u}$) of the azimuthal jet (shallow water case) as it evolves in time for a 128 × 640 grid at times (left to right, top to bottom) 0s, 1s, 2s, 3s. This corresponds to 0, 1, 2 and 3 rotational times. Max. vorticity (in red) is 1.31/s, min (in blue) is -0.71/s.

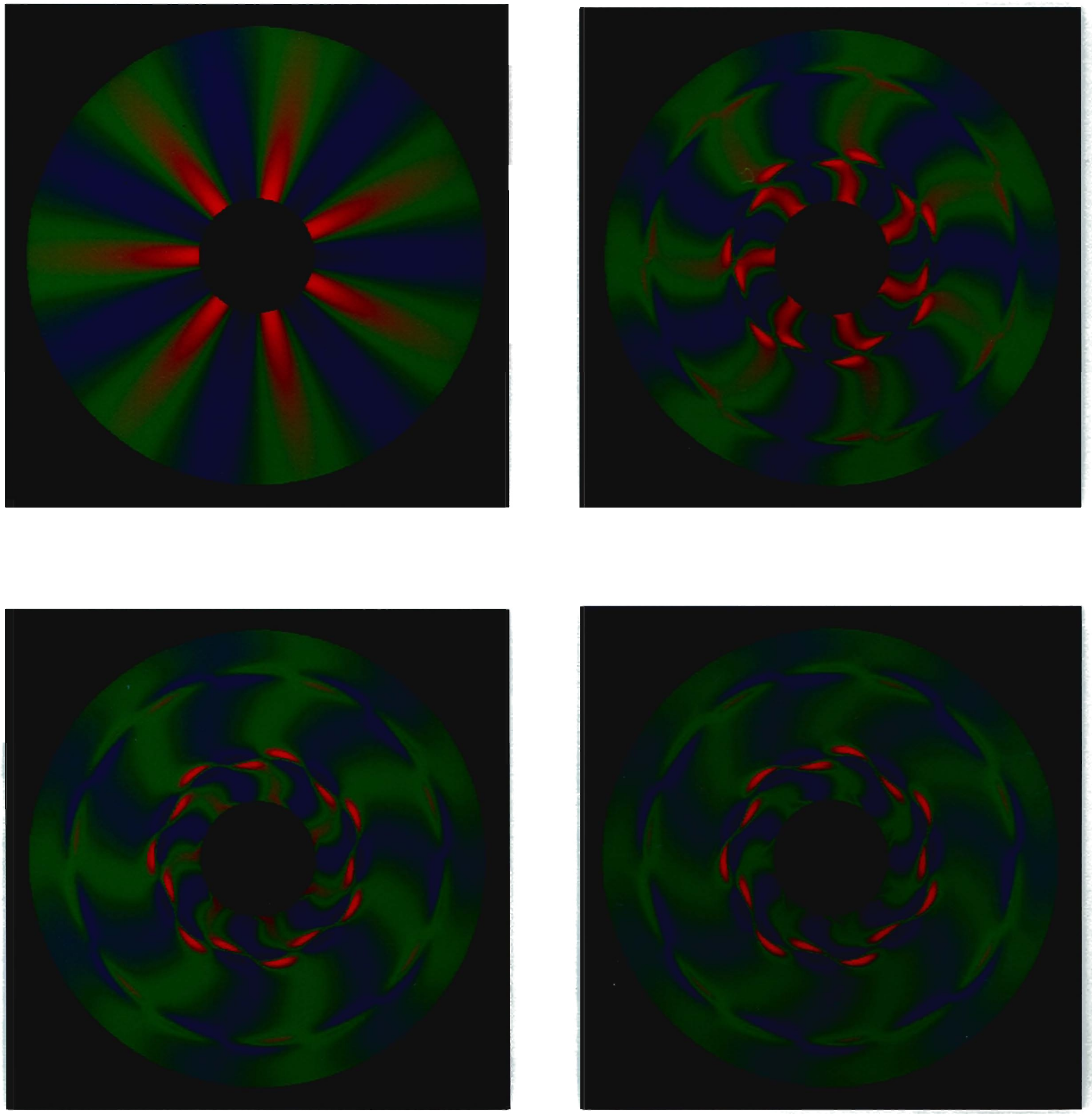


Figure 4.24: Perturbational vorticity (the cross product of $\vec{u} - \vec{u}_U$) of the azimuthal jet (shallow water case) as it evolves in time for a 128 × 640 grid at times (left to right, top to bottom) 0s (max. 0.21/s, min. -0.21/s), 1s (max. 0.27/s, min. -0.27/s), 2s (max. 0.49/s, min. -0.45/s), 3s (max. 0.68/s, min -0.61/s). This corresponds to 0, 1, 2 and 3 rotational times. Maximums in red, minimums in blue, midpoint in green.

Grid	$c(L_2)(1/s)$	$c(L_{max})(1/s)$
32x160	$4.482 \cdot 10^{-2}$	$4.029 \cdot 10^{-2}$
64x320	$3.802 \cdot 10^{-2}$	$2.496 \cdot 10^{-2}$
128x640	$3.526 \cdot 10^{-2}$	$2.246 \cdot 10^{-2}$

Table 4.7: Quasigeostrophic growth rates. The growth rate that Lee finds with his spectral calculation of the viscous problem with Ekman pumping is $8.88 \cdot 10^{-2}/s$ and the analytical growth rate predicted by Lee is $5.35 \cdot 10^{-2}/s$.

Grid	$c(L_2)(1/s)$	$c(L_{max})(1/s)$
32x160	$4.583 \cdot 10^{-2}$	$4.327 \cdot 10^{-2}$
64x320	$3.763 \cdot 10^{-2}$	$2.535 \cdot 10^{-2}$
128x640	$3.457 \cdot 10^{-2}$	$2.260 \cdot 10^{-2}$

Table 4.8: Shallow-water growth rates. The growth rate that Lee finds with his spectral calculation of the viscous problem with Ekman pumping is $8.88 \cdot 10^{-2}/s$ and the analytical growth rate predicted by Lee is $5.35 \cdot 10^{-2}/s$.

The present results converge with grid refinement to a value much smaller than the value shown by Lee. The difference in the results is due to the fact that Lee includes in his calculation the viscous effects of molecular dissipation and Ekman pumping which are not accounted for in the present investigation. In his work, Lee showed that these effects can stabilize shear layer in rotating flow if the pumping rate is below a critical value. What this investigation seems to indicate is that viscous effects can also destabilize (in the sense of giving a larger growth rate) a shear layer in a rotating flow if the pumping rate is above the critical value.

Finally, in table 4.9, we present results obtained for the case of zero rotation and zero bottom slope. Again these growth rates are clearly converged with respect to grid refinement; furthermore, these rates are smaller (by about 40%) than the results in tables 4.7 and 4.8. This shows that our method is resolving significant rotational effects.

Grid	$c(L_2)(1/s)$	$c(L_{max})(1/s)$
32x160	$2.133 \cdot 10^{-2}$	$1.266 \cdot 10^{-2}$
64x320	$2.344 \cdot 10^{-2}$	$1.431 \cdot 10^{-2}$
128x640	$2.366 \cdot 10^{-2}$	$1.437 \cdot 10^{-2}$

Table 4.9: Growth rates without rotation or bottom slope. The growth rate that Lee finds with his spectral calculation of the viscous problem with Ekman pumping is $1.031 \cdot 10^{-1}/s$.

Chapter 5

Convergence Issues

5.1 Convergence to Steady States

Chapter 4 demonstrated that this algorithm performs vortex interactions correctly. Chapter 3 demonstrated that the algorithm is second order for short times. We have not demonstrated that the algorithm converges to the correct steady state-solution even though "unphysical" processes are, in some sense, bringing that steady-state solution about.

Brown and Minion [BM95] performed a study of several numerical methods to examine the effect of an under-resolved grid on the numerical solution. Their results show that an under-resolved Godunov-projection method (much like the present algorithm), though it converges the correct solution upon refinement, produces spurious solutions. They ran a viscous shear layer and found that spurious vortices form between the roll-up vortices in the under-resolved case.

These findings are relevant to this study because we are perfoming long term, inviscid calculations. In the present case, as vortices merge by bending and folding about one another or as shear roll up, the length scales of the interactions will get smaller and smaller until *any* grid refinement will become underresolved. It is at these length scales in the physical world that viscous dissipation (or, in the case of geophysical flows, turbulent mixing) allows the vortices to merge by allowing vorticity lines to change topology so a steady state can be achieved. In an inviscid numerical calculation, "unphysical" dissipation accomplishes the same purpose.

As in the previous chapter, the tests done here are performed in two stages. First a series of quasigeostrophic calculations are run until achieving steady-state. This is done for a series of grids, to determine the rate of convergence of the steady-state solution with grid refinement. Next the same is done with a shallow-water calculation. As with the last chapter, the Rossby number used in the shallow-water calculation is fairly large since the quasigeostrophic regime is the limit of infinitesimal Rossby number and it was considered desirable to cover as large a range of parameter space as possible with the two sets of calculations.

5.1.1 Metrics of Convergence

First the measures of convergence must be carefully defined. Since the azimuthal coordinate is arbitrary, a solution must be considered converged if with time the solution only changes by a rigid body rotation. The measure of the difference between two solutions must therefore be independent of the azimuthal coordinate and must go to zero if the only difference is a rigid body rotation. The measure of the difference between two solutions used here is as follows:

$$||u - v|| = \min_{0<\theta_0<2\pi}(||u_{\theta+\theta_0} - v_\theta||) \tag{5.1}$$

Normalized with the time difference between the two states, define the measure of convergence (Γ) to be the following

$$\Gamma = \tfrac{||u_1 - u_2||}{|t_1 - t_2|} \tag{5.2}$$

The solution is considered converged to steady state when (Γ) approaches zero. Of course, since Γ will never exactly reach zero, a threshold value will be chosen by picking the smallest value of constant Γ which matches dump times at the various grid refinements.

Since the dissipation processes are of course not independent of grid refinement, a coarse grid solution will converge to steady-state more quickly than a fine grid solution. Because of this, when steady-state convergence tests are performed, solutions at different times are compared at the same level of convergence (the same Γ) rather than at the same solution time. This is equivalent to looking at the time evolution of the solution as an iteration toward steady state.

5.1.2 Convergence to Steady State of Prograde Shear Layer

The initial condition used to test for steady state is the prograde shear layer whose time evolution is shown in figure 4.13 for the shallow-water case and in figure 4.11 for the quasigeostrophic case. The algebraic formulation for the prograde shear layer is given in equation 4.8. This initial condition is chosen because it takes the longest to converge to steady state. In this way, this initial condition is the most demanding measure of steady-state convergence among the initial conditions that have been used in the present investigation. Defining Γ_0 to be the fixed value of Γ used in the in the convergence test, the parameters used in the test are given in table 5.1.

The plot of Γ_T versus solution time for three different grids is given in figure 5.1 (for the shallow-water case) and figure 5.2 (for the quasigeostrophic case). Clearly the solutions are indeed converging to steady state but at vastly different rates. The quasigeostrophic convergence test is given in tables 5.5 and 5.4. The shallow-water convergence test is given in tables 5.3 and 5.2. The convergence tests in both cases show that the solution is indeed converging to a self-consistent steady state. Though the shallow-water results are somewhat better than the quasigeostrophic, in neither case is the convergence second order or even close to second order. This is due to several factors. First, it is impossible to actually fix Γ_T for the test since data is only available at discrete times (the times at which data was

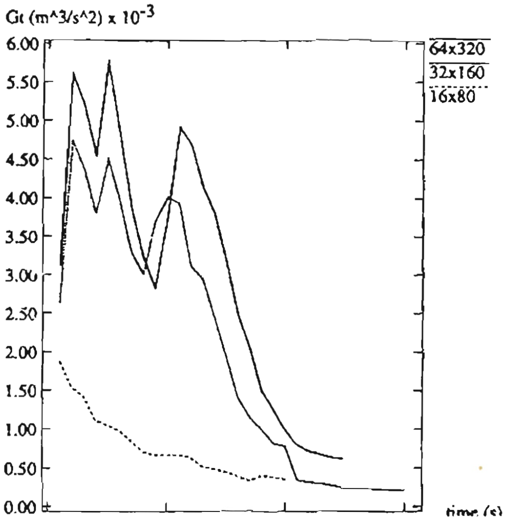


Figure 5.1: Graph of Γ vs. time for the shallow-water prograde shear layer for three separate grids. The L_2 norm is used to calculate Γ.

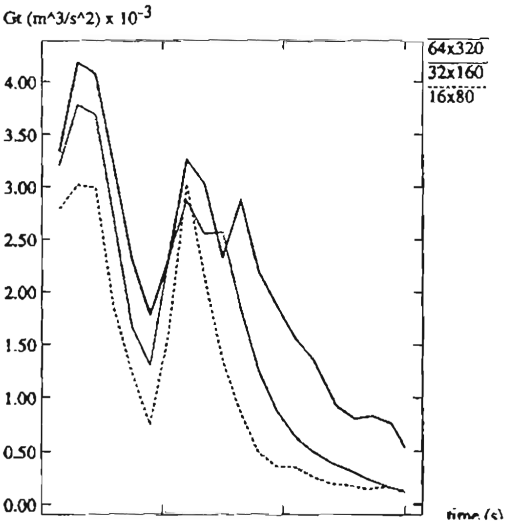


Figure 5.2: Graph of Γ vs. time for the quasigeostrophic prograde shear layer for three separate grids. The L_2 norm is used to calculate Γ.

Variable	Value
β (for QG test)	0.5/(m s)
Ω (for SW test)	0.5rad/s
R_{in}	0.33m
R_{out}	1.33m
v_0	0.5m/s
H_0 (mean depth for SW)	0.1m
S (bottom slope for SW)	0.05m/m
Γ_0(SW)	$8.1 \cdot 10^{-4} m^3/s^2$
Γ_0(QG)	$1.8 \cdot 10^{-3} m^3/s^2$

Table 5.1: Parameters used in prograde shear layer steady-state convergence test. Γ_O is the value of Γ used to indicate convergence.

Var	$p(L_1)$	$p(L_2)$	$p(L_\infty)$
u_r	1.21	0.99	0.54
u_θ	1.76	1.60	1.26

Table 5.2: Shallow-water convergence test using several numerical norms for $\Gamma_0 = 8.1x10^{-4} m^3/s^2$

Var	$\|\|32x160 - 16x80\|\|_2$	$\|\|64x320 - 32x160\|\|_2$	$p(L_2)$
u_r	$6.320 \cdot 10^{-3}$	$3.180 \cdot 10^{-3}$	0.99
u_θ	$5.487 \cdot 10^{-3}$	$1.667 \cdot 10^{-3}$	1.60

Table 5.3: Shallow-water convergence test using several numerical variables and the L_2 norm for $\Gamma_0 = 8.1x10^{-4} m^3/s^2$

Var	$p(L_1)$	$p(L_2)$	$p(L_\infty)$
u_r	0.21	0.33	0.43
u_θ	0.38	0.41	0.78

Table 5.4: Quasigeostrophic steady-state convergence test using several numerical norms for $\Gamma_T = 1.8x10^{-3} m^3/s^2$

Var	$\|\|32x160 - 16x80\|\|_2$	$\|\|64x320 - 32x160\|\|_2$	$p(L_2)$
u_r	$4.893 \cdot 10^{-3}$	$3.885 \cdot 10^{-3}$	0.33
u_θ	$6.997 \cdot 10^{-3}$	$5.249 \cdot 10^{-3}$	0.78

Table 5.5: Quasigeostrophic steady-state convergence test using several numerical variables and the L_2 norm for $\Gamma_T = 1.8x10^{-3}m^3/s^2$

dumped during the run). The actual data is taken at is just that dump time at which Γ_T is closest to the chosen value Γ_{T0}. The second factor that might be reducing the order of convergence is the fact that the dissipation mechanisms that bring the solution toward steady state are not grid independent. This fact is clear from figure 5.1, which clearly shows that the different grids are moving toward steady state at different rates. These "unphysical" dissipation mechanisms, since they are not grid independent, clearly should affect the convergence (with grid refinement) rate of the steady state solution.

5.2 Conservation of Circulation and Kinetic Energy

We also must show that the algorithm converges to a solution which conserves both kinetic energy and circulation. Taking the dot product of equation 1.3 and the mass flux vector ($\vec{u}H$), and integrating over the entire domain gives the time evolution equation for the total kinetic energy (E):

$$\frac{\partial E}{\partial t} = \frac{\partial}{\partial t}(\int_\gamma \frac{H}{2}\vec{u} \cdot \vec{u}) = 0 \tag{5.3}$$

Equation 5.3 is a statement that the total kinetic energy of the system must remain constant. This is another very familiar result from ordinary inviscid, incompressible flow. The implication of this result is that neither a rotating coordinate frame nor a variable depth with a rigid lid can add kinetic energy to a flow. [Ped79].

Integrating equation 1.3 over the entire domain yields (γ) another significant result:

$$\frac{\partial \Gamma}{\partial t} = \frac{\partial}{\partial t}(\int_{\partial\gamma} \vec{u} \cdot \vec{dl}) = 0 \tag{5.4}$$

which by Green's Theorem is equivalent to

$$\frac{\partial \Gamma}{\partial t} = \frac{\partial}{\partial t}(\int_{\gamma} \omega dA) = 0 \tag{5.5}$$

where Γ is the circulation of the system. Equation 5.5 is the equivalent of Kelvin's theorem, which states that circulation for inviscid incompressible flow is constant in time. In a sense, this is a surprising result since the physics of shallow-water flow are very different from standard Euler flow.

Though the present algorithm is not explicitly designed to conserve either kinetic energy or circulation, these quantities are quite important to the physics of the system so it is important to know to what extent numerical kinetic energy and circulation conserved in the integral sense.

We use as an initial condition the same prograde shear layer run in the previous test. The time evolution of the solution is shown in figure 4.13 for the shallow-water case and in figure 4.11 for the quasigeostrophic case. The algebraic formulation for the prograde shear layer is given in equation 4.8. The shear layer takes many rotation times to roll up and merge into one vortex. We therefore consider the prograde shear layer problem to be a reasonable test of the performance of the algorithm over long integration times with respect to rotation time. We shall use this problem as a measure of how well the algorithm conserves kinetic energy for long integration times with respect to the rotational time scale. The parameters used in this test are given in table 5.6. Again both shallow-water and quasigeostrophic cases are investigated. To conform with the first chapter's definitions, kinetic energy is defined as follows:

$$E_K = \int \frac{H}{2}(\vec{u} \cdot \vec{u}) dA \tag{5.6}$$

Variable	Value
Ω (for SW test)	0.5/s
β (for QG test)	0.5/(m s)
R_{in}	0.33m
R_{out}	1.33m
v_0	0.5m/s
S (bottom slope for SW)	0.05m/m
H_0 (mean depth for SW)	0.1m

Table 5.6: Parameters used in quasigeostrophic and shallow-water tests for kinetic energy and circulation conservation.

Though this is a peculiar way to weight the kinetic energy for the quasigeostrophic case, by defining the total kinetic energy by equation 5.6, the total kinetic energy is conserved for both the quasigeostrophic case and the shallow-water case.

Graphs of numerical kinetic energy versus time are given in figures 5.3 (quasigeostrophic) and 5.5 (shallow-water). Graphs of numerical circulation versus time are given in tables 5.4 (quasigeostrophic) and 5.6 (shallow-water). Define the order of convergence of the algorithm for a particular conserved quantity (η) to be (p_η). Given numerical solutions with grid spacings of h, $2h$, and $4h$ and manipulating equation 3.12 and solving for p_η, gets the following:

$$p_\eta = \frac{1}{\ln(2)} \ln(\frac{|\eta_{2h}(t_f) - \eta_{2h}(0)|}{|\eta_h(t_f) - \eta_h(0)|}) \tag{5.7}$$

where t_f is the final time. Since the quantities η represents are single numbers, the issues of norms in equation 3.12 are not an issue here. Using equation 5.7 gives an estimate of the rate of convergence of the algorithm toward conserving (η).

A table of convergence rates of various conserved quantities is given in tables 5.7 (quasigeostrophic) and 5.8 (shallow-water). The solution clearly is approaching a state in which kinetic energy is conserved at a rate which is second order with grid refinement. Though the circulation results are clearly not in the aymptotic regime, the algorithm is

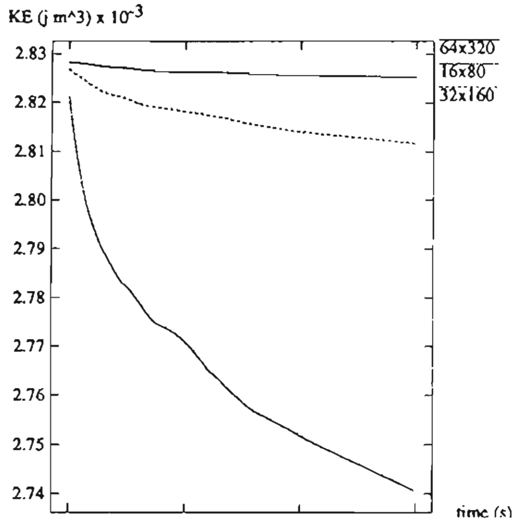


Figure 5.3: Sum of the flow kinetic energy for the quasigeostrophic prograde shear layer problem for three separate grids.

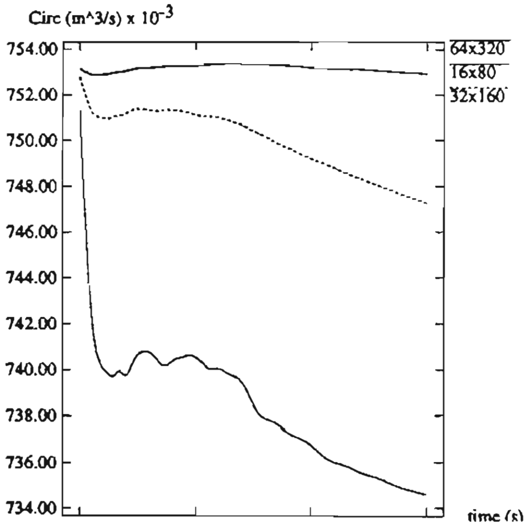


Figure 5.4: Total circulation for the quasigeostrophic prograde shear layer problem for three separate grids.

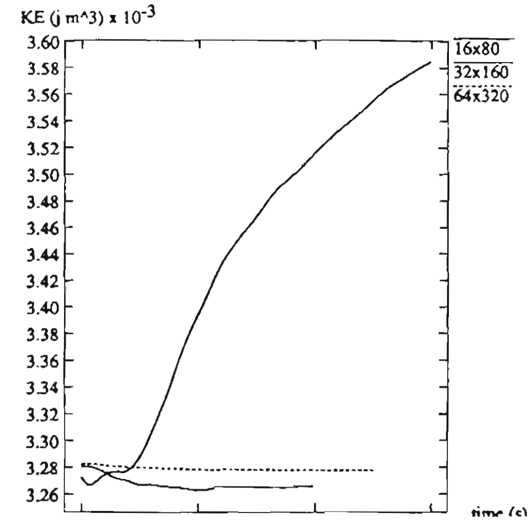


Figure 5.5: Sum of the flow kinetic energy for the shallow-water prograde shear layer problem for three separate grids.

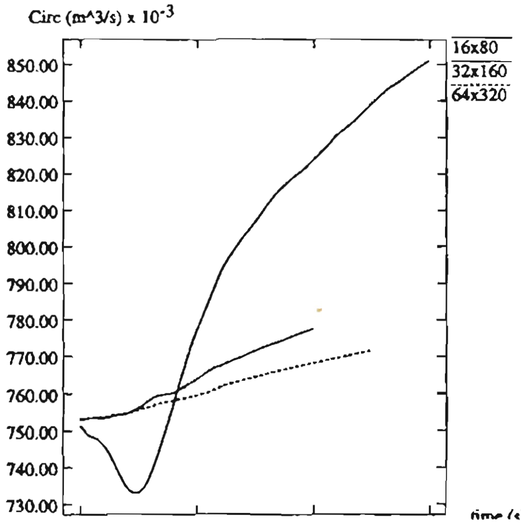


Figure 5.6: Total circulation for the shallow-water prograde shear layer problem for three separate grids.

Var	$\Delta\eta_{16x80}$	$\Delta\eta_{32x160}$	$\Delta\eta_{64x320}$	$p(16-32)$	$p(32-64)$
E_K	$7.255 \cdot 10^{-5}$	$1.347 \cdot 10^{-5}$	$2.754 \cdot 10^{-6}$	2.43	2.29
Γ	$1.535 \cdot 10^{-2}$	$4.107 \cdot 10^{-3}$	$1.828 \cdot 10^{-5}$	1.90	7.81

Table 5.7: Quasigeostrophic convergence test for kinetic energy and circulation for the prograde shear problem. The final time is 450s. Circulation is in m^3/s and kinetic energy is in jm^4.

Var	$\Delta\eta_{16x80}$	$\Delta\eta_{32x160}$	$\Delta\eta_{64x320}$	$p(16-32)$	$p(32-64)$
E_K	$2.440 \cdot 10^{-4}$	$1.457 \cdot 10^{-5}$	$4.653 \cdot 10^{-6}$	4.06	1.65
Γ	$7.273 \cdot 10^{-2}$	$2.478 \cdot 10^{-2}$	$6.467 \cdot 10^{-4}$	1.55	5.26

Table 5.8: Shallow-water convergence test for kinetic energy and circulation for the prograde shear problem. The final time is 400s. Circulation is in m^3/s and kinetic energy is in jm^4.

converging to a state in which circulation is conserved. We have therefore shown that the algorithm respects conservation of kinetic energy and circulation in rotating flows.

5.3 Convergence to the Quasigeostrophic Limit

In the first chapter, it was shown analytically that the shallow-water equations converge to the quasigeostrophic equations in the limit of rapid rotation. Specifically, it was shown that in the limit in which $Ro \rightarrow 0$, the system of equations 1.3 converges to equation 1.45. In the course of this investigation, several solutions have been calculated both with a relatively large Rossby number (the shallow-water examples thus far have had a Rossby number of about unity) and in the quasigeostrophic limit, which represents the limit of rapid rotation. What has not been shown is that, in the limit of decreasing Rossby number, the numerical shallow-water solution converge the the numerical quasigeostrophic solution.

There are two sources of difference between the shallow-water solution and the quasigeostrophic solution. First, the quasigeostrophic solution contains only a portion of the shallow-water velocity field. Using the Hodge decomposition, the shallow-water velocity field can be divided into two components, a $\mathcal{D}$-divergence-free component $\vec{u}_o$ and a potential flow component $\vec{u}_p$.

$$\begin{aligned} \vec{u} &= \vec{u}_o + \vec{u}_p \\ \mathcal{D}\vec{u}_o &= 0 \\ \nabla \times \vec{u}_p &= 0 \end{aligned} \tag{5.8}$$

The quasigeostrophic equations only solve for $\vec{u}_o$ while the shallow-water velocity field contains both $\vec{u}_o$ and $\vec{u}_p$.

Second, the forcing terms in the two sets of the equations are different. The quasigeostrophic forcing term F_{qg} is the shallow-water forcing term F_{rot} in the limit of rapid rotation and diminishing potential flow

$$F_{qg} = \lim_{\Omega \to \infty, \vec{u}_p \to 0} F_{rot} \tag{5.9}$$

As shown in the first chapter, in the case of the β-plane approximation, when the depth H is given by

$$H_{\beta-Plane} = \exp(\frac{\beta r}{\Omega}) \tag{5.10}$$

the quasigeostrophic rotational forcing is given as follows:

$$(2\beta r\hat{k}) \times \vec{u}_o = \lim_{\Omega \to \infty, \vec{u}_p \to 0} (2\Omega \times \vec{u}_p) \tag{5.11}$$

The forcing terms of the two systems of equations being different provides the second source of difference between quasigeostrophic solutions and shallow-water solutions. As the solution proceeds in time, this component of the difference should increase for a given Rossby number.

We approximate the time derivative of the velocity $\vec{a} = \frac{\partial u}{\partial t}$ after one time step as follows:

$$\vec{a}_{i,j} = \frac{\vec{u}^1_{i,j} - \vec{u}^0_{i,j}}{\Delta t} \tag{5.12}$$

Variable	Value
CFL	0.5
β	$3.0 \cdot 10^{-2}/(\mathrm{m\ s})$
R_{in}	0.33m
R_{out}	1.33m
v_o	0.1m/s
H_0	1.0m
Δt	$7.8 \cdot 10^{-3}$s

Table 5.9: Parameters used in the test for convergence in the geostrophic limit.

where $\vec{u}^1$ is the velocity after one time step and $\vec{u}^0$ is the initial velocity. We wish to isolate only the effect of the different forcing terms upon the numerics. To do this, we project the gradient component out of the shallow-water $\vec{a}$ field before comparing the two fields. We look at the quantity

$$D_o = \mathcal{P}_{\mathcal{O}}(\vec{a}_{sw}) - \vec{a}_{qg} \tag{5.13}$$

which measures only dynamic effects. We run with an initial condition is run for ten different Rossby numbers with the depth function given by equation 5.11, and compare these solutions to the quasigeostrophic solution with the appropriate β by looking at D_o. The initial condition used is given as follows:

$$\begin{aligned} u_{\theta,i,j} &= v_o(-(r_{i,j} - R_{in}) - 0.25(r_{i,j} - R_{in})^2) \\ u_{r,i,j} &= v_o \sin(4\theta) \end{aligned} \tag{5.14}$$

After it has been projected, the initial condition specified in equation 5.14 has net circulation associated with it at the inner boundary. This must be subtracted from the velocity initial condition because the analysis of the quasigeostrophic limit depends upon the velocity being uniquely determined by the vorticity. If there is any net circulation, this is no longer the case. Defining $\vec{u}^{\,dv}$ as the velocity field described in equation 3.11, the inner boundary circulation Γ is calculated

$$\Gamma = \sum_{j=1}^{N_\theta}(u_e R_{in} \Delta\theta) \tag{5.15}$$

where u_e is the azimuthal velocity component extrapolated to the inner edge as follows:

$$u_e = \frac{3}{2}u_{\theta,1,j} - \frac{1}{2}u_{\theta,2,j} \tag{5.16}$$

The velocity component associated with the circulation is then subtracted off.

$$\vec{u}_{i,j} = \vec{u}^{adv}_{i,j} - \frac{\Gamma}{2\pi r_{i,j}} \tag{5.17}$$

The grid size is 64 × 320 and all parameters are contained in table 4.1.

Graphs of D_o vs. Rossby number are given in figures 5.7 and 5.8. Ten points are taken per graph, with $Ro = 0.1$, $Ro = 0.2$, etc. Ideally all of these graphs would approach zero as the Rossby number becomes small, showing that the shallow-water solution is converging to the quasigeostrophic solution with decreasing Rossby number. Both components of D_o decrease sharply with decreasing Rossby number. Clearly the numerics are indeed capturing to a large degree the quasigeostrophic limit. These results are preliminary, however. This test involves only a single time step so clearly the long time numerical behavior of the algorithm with respect to the quasigeostrophic limit has not been measured.

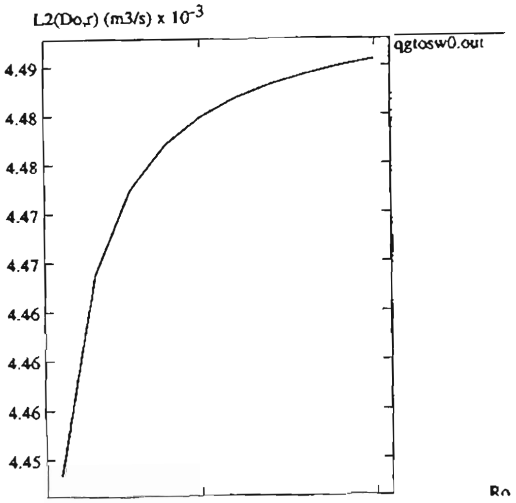


Figure 5.7: The L_2 norm of $D_{o,r}$ vs. *Ro* for time $= 7.8 \cdot 10^{-3}$ s.

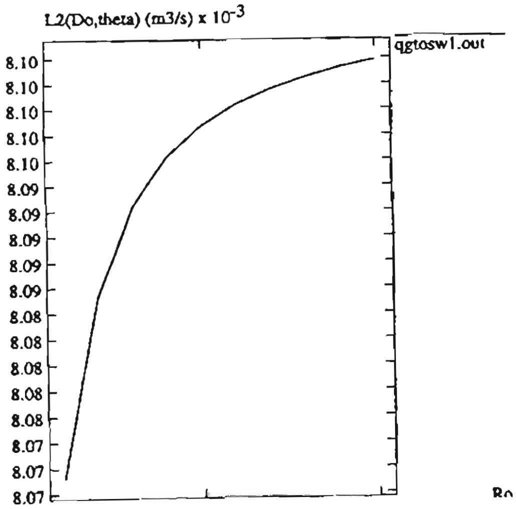


Figure 5.8: The L_2 norm of $D_{o,\theta}$ vs. *Ro* for time $= 7.8 \cdot 10^{-3}$ s.

Chapter 6

Discussion and Conclusions

In the present investigation an approximate projection method appropriate for modeling vortical, rotating flows has been developed. This algorithm has been systematically tested and its performance quantified in terms of the design criteria based upon the physics of rotating flows.

First the relevant equations for this investigation were derived. The shallow-water equations 1.3 are the Euler equations in a rotating, two-dimensional coordinate system. The shallow-water equations emerge from the assumption that vertical length scales are much shorter than horizontal length scales. Classically the quasigeostrophic equations are usually derived using asymptotics from the shallow-water equations in the limit of rapid rotation. In the present investigation, a generalized form of the quasigeostrophic equations 1.38 were derived without the benefit of asymptotics by using the tools of projection formalism. With projection formalism the velocity field was divided into its $\mathcal{D}$-divergence-free component $\vec{u}_o$ and its potential flow component $\vec{u}_p$. Classical analysis shows that the effect of rapid rotation is to force $\vec{u}_o$ to be dominant. The present investigation showed that any assumption about the relative size of $\vec{u}_o$ and $\vec{u}_p$ is precisely equivalent to an assumption of the composition of the rotational forcing term in the momentum equation $F_{rot} = 2\Omega \times \vec{u}_p$.

At this point asymptotics were employed by assuming the Rossby number Ro is small and the ratio $\frac{\vec{u}_p}{\vec{u}_o} = O(Ro)$. The result of this is to simplify equation 1.38, resulting in equation 1.45, a more standard form of the quasigeostrophic equations. This provided some insight into the nature of the quasigeostrophic limit, specifically the rotational forcing term in the quasigeostrophic limit F_{qg} is given by

$$\begin{aligned} F_{qg} &= \lim_{\frac{\Omega L}{U_o}\to\infty,\frac{\vec{u}_p}{U_o}\to 0} (2\Omega \times \vec{u}_p) \\ F_{qg} &= \lim_{\frac{\Omega L}{U_o}\to\infty,\frac{\vec{u}_p}{U_o}\to 0} F_{rot} \end{aligned} \tag{6.1}$$

where F_{qg} is finite. In this context, the β-plane approximation (where $F_{qg} = (\beta r \hat{k}) \times \vec{u}_o$) is explained as an example of equation 6.1 where the depth function is given by equation 1.48.

After the derivation of the relevant equations, the design of the numerical algorithm was described in detail and then analyzed with respect to very specific design criteria. First are the issues of accuracy. The algorithm was designed to be second-order in space in time. Short time convergence studies are given in tables 3.4 and 3.5 for the quasigeostrophic case and tables 3.7 and 3.6 for the shallow-water case. These tests show that, by certain reasonable metrics, the algorithm is indeed shows second-order convergence with grid refinement.

The physical design criteria have to do with the fact that many of the problems in geophysical fluid dynamics are vortical structures which are small perturbations of parallel shear flows. It is therefore necessary for the solution to reproduce as an exact discrete solution an unperturbed parallel shear flow. Numerically this design point was tested by simply using as an initial condition an unperturbed parallel shear flow. The results of this test, given in table 3.2, clearly show that the the algorithm preserves unperturbed parallel shear flow solutions. The reason for this success was that particular care was taken in the discretization of the forcing terms equation 2.36. Since an approximate projection is being used, forcing terms and pressure gradients must be balanced before being projected;

otherwise, unperturbed parallel shear flows will not be preserved exactly.

The aforementioned vortical structures can develop over long times and exhibit a distinguishing sign of perturbational vorticity. A vortex whose perturbational vorticity is the same sign as the background shear (a "prograde" vortex) persists and these vortices persist over very long times (compared to the rotational time scale). A vortex whose perturbational vorticity is the opposite sign as the background shear (an "adverse" vortex) is broken up. To test the algorithm's ability to model this behavior, the algorithm was run for several examples of flows which consist of small perturbations of background shear flows. The first example run was a pair of opposite sign vortices in background shear whose algebraic formulation is given in equation 3.11. The time evolution of the solution is shown in figure 4.3 for the quasigeostrophic case and shown in figure 4.5 for the shallow-water case. The opposite sign vortex is broken up very quickly and the same sign vortex persists. Another example of one of these flows is a shear layer in a background shear. The time evolution of perturbational vorticity for the prograde shear layer is given in figure 4.11 (quasigeostrophic case) and in figure 4.13 (shallow-water case). The layer rolls up due to a Kelvin-Helmholtz instability and the resulting vortices merge to form a very persistent solution. The time evolution of perturbational vorticity for the adverse shear layer is given in figure 4.16 (quasigeostrophic run) and figure 4.15 (shallow-water run). The adverse layer was torn apart quite quickly by the background shear. Clearly the algorithm models the physical behavior of perturbational solutions of background shear flows quite well. Since the prograde cases were run for many rotational times, it is also clear the algorithm is stable for long integration times with respect to rotational times.

Since the previous class of flows had very similar results in both a shallow-water case with large Rossby number and the quasigeostrophic case (which represents the limit of small Rossby number), those solutions are being driven less by the rotational forcing than by the background shear. To try to quantify the ability of the algorithm to model a flow in which rotation is very important, stability of shear layers in rotating flows was analyzed and compared to the results of spectral calculations of viscous rotating flows. Rotation very much alters the stability characteristics of rotating flows. In viscous rotating flows, it is even possible for shear layers to be stabilized. Since the present calculation is inviscid, only the unstable rotating shear layers were compared with experiment and spectral calculation of viscous rotating flows. The results of the investigation are given in table 4.7 for the quasigeostrophic case and table 4.8 for the shallow-water case. The algorithm converges to a self-consistent growth rate with grid refinement that is substantially less (approximately twenty percent) less than the viscous calculation.

Finally the algorithm was analyzed with respect to convergence issues. In inviscid rotating flows, both circulation and kinetic energy are analytically conserved. To test the performance of the algorithm in this respect, the change in numerical kinetic energy and circulation are measured for the prograde shear layer problem. Convergence studies with respect to kinetic energy and circulation convergence with grid refinement are given in table 5.7 in the quasigeostrophic case and table 5.8 for the shallow-water case. Even though the algorithm was not specifically designed to conserve kinetic energy or circulation, convergence in both cases is excellent. The kinetic is conserved to second-order with grid refinement and the circulation shows even better convergence rates.

Next the algorithm was analyzed with respect to convergence to steady state solutions. Long-term inviscid calculations were made here with vortices merging and shear layers breaking up. Vortices merge (and shear layers break up) by bending and folding over one another until the length scales get too small to be resolved. In the physical world, at these length scales, viscous dissipation becomes dominant, which allows the vortex lines to change topology so that a steady state can be reached. In an inviscid numerical calculation, "unphysical" dissipation accomplishes the same thing. We must therefore determine whether this "unphysical" numerical dissipation, which necessarily depends on grid refinement, leads to a self-consistent steady state if the same calculation is run for several different grids. The prograde shear layer problem was run for three separate grids and the convergence of the resulting solution with grid refinement was analyzed. The convergence studies are given in table 5.2 for the shallow-water case and table 5.4 for the quasigeostrophic case. Depending on which numerical norm was used, the algorithm showed steady state convergence rates of somewhat less than unity. Clearly the numerical dissipation in the algorithm indeed brings about a self-consistent steady state solution but can reduce the accuracy of the method.

Finally the algorithm was analyzed to see if the numerical shallow-water solution converges to the numerical quasigeostrophic solution with decreasing Rossby number. Graphs of the relevant measure of the difference between the shallow-water and quasigeostrophic numerical fields is given in figures 5.7 and 5.8 for a run time of a single rotational time for a Rossby number of unity. Both components of this difference decrease with with decreasing Rossby number, precisely as the analysis indicates it should. Clearly the numerics are indeed capturing to some degree the quasigeostrophic limit.

We have presented an approximate projection method algorithm suitable for modeling rotating flows. We have used projection formalism to clarify the nature of the quasigeostrophic limit. We have shown that the algorithm captures well the vortex dynamics of flows typical of those found in large-scale geophysical fluid dynamics over long integration times. From this base, adaptive mesh algorithms for geophysical fluid dynamics can be developed.

Bibliography

[ABC94] Ann S. Almgren, Thomas Buttke, and Phillip Colella. A fast vortex method in three dimensions. *Journal of Computational Physics*, 113(2):177–200, 1994.

[ABS96] A. S. Almgren, J. B. Bell, and W. G. Szymczak. A numerical method for the incompressible Navier-Stokes equations based on an approximate projection. *SIAM J. on Sci. Comp.*, 17:358–369, 1996.

[AJSC92] Jr.. Albert J. Semtner and R.M. Chervin. Ocean general circulation from a global eddy-resolviing model. *Journal Geophysical Research-Oceans*, 97:5493–5550, 1992.

[AJSM77] Jr.. Albert J. Semtner and Yale Mintz. Numerical investigation the gulf stream and mid-ocean eddies. *Journal of the Physical Oceanography*, 7:208–229, 1977.

[And86] C. Anderson. A method of local corrections for computing the velocity field due to a distribution of vortex blobs. *J. Comput. Phys.*, 62:111–123, 1986.

[BCG89] J. B. Bell, P. Colella, and H. M. Glaz. A second-order projection method for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 85:257–283, 1989.

[BCH91] J. B. Bell, P. Colella, and L. H. Howell. An efficient second order projection method for viscous incompressible flow. In *AIAA 10th Comp. Fluid Dynamics Conf.*, pages 360–367, 1991.

[BM95] David L. Brown and Michael L. Minion. Performance of under-resolved two-dimensional incompressible flow simulations. *J. Comput. Phys.*, 122:165–183, 1995.

[Bri87] W. L. Briggs. *A Multigrid Tutorial.* SIAM, Philadelphia, PA, 1987.

[Bry63] Kirk Bryan. A numerical investigation of a nonlinear model of a wind-driven ocean. *Journal of the Atmospheric Sciences*, 20:594–606, 1963.

[CBSJ94] Claude Cambon, Jean-Pierre Benoit, Liang Shao, and Laurent Jacquin. Stability analysis and large-eddy simulation of rotating turbulence with organized eddies. *J. Fluid Mech.*, 278:175–200, 1994.

[Cho68] A. J. Chorin. Numerical solutions of the Navier-Stokes equations. *Math. Comp.*, 22:745–762, 1968.

[Cho69] A. J. Chorin. On the convergence of discrete approximations to the Navier-Stokes equations. *Math. Comp.*, 23:341–353, 1969.

[Cho73] A. J. Chorin. Numerical study for slightly viscous flow. *J. Fluid Mech.*, 57:785–796, 1973.

[CM92] K. T. Coughlin and P. S. Marcus. Modulated waves in taylor-couette flow. *J. Fluid Mech.*, 234:19–46, 1992.

[Cox75] M. D. Cox. A baroclinic model of the world's ocean. In *Numerical Models of Ocean Circulation*, pages 107–120. National Academy of Sciences, Washington, DC, 1975.

[Dah93] Marie Dillon Dahleh. Fast numerical method for the solution of the quasi-geostrophic vorticity equation. *Numerical Methods for Partial Differential Equations*, 9:135–154, 1993.

[Gil82] Adrian E. Gill. *Atmosphere-Ocean Dynamics.* Academic Press, New York, NY, 1982.

[GR87] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73(2):325–348, 1987.

[HB] Louis H. Howell and John B. Bell. An adaptive-mesh projection method for viscous incompressible flow. To appear, SIAM Journal of Scientific Computing.

[HE81] Roger W. Hockney and James W. Eastwood. *Computer Simulation Using Particles.* McGraw-Hill, New York, NY, 1981.

[Hoc65] R. W. Hockney. A fast direct solution of poisson's equation using fourier analysis. *J. Association of Computing Machinery*, 12:95, 1965.

[Hoc71] R. W. Hockney. The potential calculation and some applications. *Methods in Computational Physics*, 9, 1971.

[HW65] F.J. Harlow and J.E. Welsh. Numerical calculation of time dependent viscous incompressible flow with free surface. *Physics of Fluids*, 8, 1965.

[JDA95] Jr. John D. Anderson. *Computational Fluid Dynamics: The Basics with Applications.* McGraw-Hill, New York, NY, 1995.

[Lai94] M. F. Lai. *A Projection Method for Reacting Flow in the Zero Mach Number Limit.* PhD thesis, University of California, Berkeley, 1994.

[LC] M. F. Lai and P. Colella. An approximate projection method for the incompressible Navier-Stokes equations. unpublished.

[Lee94] Changhoon Lee. *Basic Instability and Transition to Chaos in a Rapidly Rotating Annulus on a β-plane.* PhD thesis, University of California, Berkeley, 1994.

[Mar84] P. S. Marcus. Simulation of taylor-couette flow. part 1. numerical methods and comparison with experiment. *J. Fluid Mech.*, 146:45–64, 1984.

[Mar90] P. S. Marcus. Vortex dynamics in a shearing zonal flow. *J. Fluid Mech.*, 215:393–430, 1990.

[Mar93] P. S. Marcus. Jupiters great red spot and other vortices. *Annu. Rev. Astron. Astrophys.*, 31:523–573, 1993.

[MS71] D. W. Moore and P. G. Saffman. Structure of a line vortex in an imposed strain. In John H. Olsen, Arnold Goldburg, and Milton Rogers, editors, *Aircraft Wake Turbulence and its detection.* Plenum Press, New York, NY, 1971.

[MSS93] S. D. Meyers, J. Sommeria, and Harry L. Swinney. Laboratory study of the dynamics of jovian-type vortices. *Physica D*, 37:515–530, 1993.

[Orz70] S. A. Orzag. Transform method for calculation of vector-coupled sums: Application to the spectral form of the vorticity equation. *Journal of Atmospheric Sciences*, 27:890–895, 1970.

[OS78] J. Oliger and A. Sundstrom. Theoretical and practical aspects of some initial boundary value problems in fluid dynamics. *SIAM Journal on Applied Mathematics*, 35(3):419–446, 1978.

[Ped79] Joseph Pedlosky. *Geophysical Fluid Dynamics.* Springer-Verlag, New York, NY, 1979.

[PL95] Ugo Piomelli and Junhui Liu. Large-eddy simulation of rotating channel flows using a localized dynamic model. *Physics of Fluids*, 7(4):839–848, April 1995.

[Puc93] Elbridge Gerry Puckett. Vortex methods: An introduction and survey of selected research topics. In Max D. Gunzburger and Roy A. Nicolaides, editors, *Incompressible Computational Fluid Dynamcis Trends and Advances.* Cambridge University Press, Cambridge, 1993.

[Roa72] Patrick J. Roache. *Computational Fluid Dynamics.* Hermosa Publishers, Albuquerque, NM, 1972.

[SC95] Jens Nørkær Sørensen and Eric Adler Christensen. Direct numerical simulation of rotating fluid flow in a closed cylinder. *Physics of Fluids*, 7(4):764–778, 1995.

[SHS93] T. H. Solomon, W. J. Holloway, and Harry L. Swinney. Shear flow instabilities and Rossby waves in barotropic flow in a rotating annulus. *Phys. Fluids A*, 5(8):1971–1982, 1993.

[WP86] Warren M. Washington and Claire L. Parkinson. *An Introduction to Three-Dimensional Climate Modeling.* University Science Books, Mill Valley, CA, 1986.