THE SHALLOW WATER EQUATIONS ON A SPHERICAL GEODESIC GRID

by Ross Heikes

David A. Randall, Principal Investigator

Colorado
State
University

DEPARTMENT OF
ATMOSPHERIC SCIENCE

PAPER NO. 524

THE SHALLOW WATER EQUATIONS ON A SPHERICAL GEODESIC GRID

by

Ross Parker Heikes

Research supported by the
U.S. Department of Energy
under Grant number DE-FG02-91ER-61218

Department of Atmospheric Science
Colorado State University
Fort Collins, CO

April 1993

Atmospheric Science Paper No. 524

Abstract

A computer model is presented that solves the stream function/velocity potential form of the shallow water equations using a new spherical geodesic grid that covers the sphere more homogeneously and isotropically than latitude-longitude grids. The geometric properties of the grid are considered. Following Masuda, the finite difference methods discretize line integrals to time-step the prognostic equations. Multigrid methods are used to solve the diagnostic equations for the stream function and velocity potential. The model is compared with the Arakawa-Lamb shallow water model and the NCAR spectral transform shallow water model using the suite of seven test cases proposed by Williamson. The model performance characteristics are presented.

The test cases show that the evolution of the fields is independent of the relative orientations of the computational grid and the flow pattern. This is particularly true when flow is directed over the pole of the grid. Also, the new model is tested using a Rossby-Haurwitz wave as initial conditions. The initial disturbance breaks down towards lower wavenumbers, but remains symmetric across the equator.

Acknowledgments

I would like to thank Dr. David Randall, Dr. Jommy Jensen, Don Dazlich and Cindy Carrick for their help in preparing this thesis. I would to thank Dr. James Hack and Dr. Ruediger Jakob at NCAR for their help with the NCAR spectral transform model.

Support for this reasearch was provided by U.S. Department of Energy Grant Number DE-FG02-91ER-61218. Computing support was provided by the National Energy Research Supercomputer Center at Lawrence Livermore National Laboratry.

Table of Contents

Chapter 1: The Shallow Water Equations

Chapter 2: The Model

Chapter 3: Numerical Results

Chapter 4: Model Performance

Chapter 5: Conclusions and Future Plans

Appendix A: Derivations

Appendix B: The Arakawa-Lamb Shallow Water Model

Appendix C: The NCAR Spectral Transform Shallow Water Model

References

Chapter 1

The Shallow Water Equations

1.1 Introduction

The shallow water equations are the simplest form of the equations of motion to describe the horizontal structure of the atmosphere. These equations describe the evolution of a fluid in response to gravitational and rotational accelerations. Their solutions represent many of the types of motion found in the real atmosphere, including slow-moving Rossby waves, and are useful for testing numerical schemes since they also produce fast-moving gravity waves. In this chapter, we will first present the more complete equations of motion, explain the terms of these equations, and show how they can be simplified by assuming the fluid is incompressible and hydrostatic to get at the shallow water equations. We will present various forms of the shallow water equations. Then we will examine the problems inherent in trying to discretize the shallow water equations in spherical geometry, and give a brief history of how others have overcome these difficulties, including a discussion of spherical geodesic grids. Finally, we show a grid constructed using the wandering electron algorithm.

1.2 Derivation of the Shallow Water Equations

1.2.1 The Equations of Motion

The equations of motion consist of the momentum equations, based on the conservation of momentum, and the continuity equation, based on the conservation of mass. We first state the equations and then examine the individual terms in detail. Ignoring the effects of viscosity, the momentum equations are given by

$$\frac{D\boldsymbol{u}}{Dt} + 2\hat{\Omega} \times \boldsymbol{u} + \hat{\Omega} \times (\hat{\Omega} \times \boldsymbol{x}_r) = -\rho \nabla p - \nabla \Phi_v, \tag{1.1}$$

and the continuity equation by

$$\rho^{-1}\frac{D\rho}{Dt} + \nabla \bullet \boldsymbol{u} = 0. \tag{1.2}$$

Here, the total derivative operator$D(\ )/Dt$ is defined by

$$\frac{D(\ )}{Dt} \equiv \frac{\partial}{\partial t}(\ ) + \boldsymbol{u} \bullet \nabla(\ ). \tag{1.3}$$

Newton's laws are inherently geared to describe the interactions between of a number of discrete bodies, such as a solar system or a bowling ball striking pins. However, the thing we wish to explain with these laws, a fluid, is a continuous, integrated thing. Therefore, we introduce the idea of a fluid parcel to mentally discretize the fluid. A parcel is small volume of the fluid and is characterized by a number of properties. It must large enough so that statistically averaged quantities, such as pressure, and well defined. For example, it does not make sense to say that a parcel containing a single molecule has a certain pressure or temperature. At the same time, a parcel must be small enough so that statistically defined quantities are uniform across it.

1.2.1.1 The Terms of the Momentum Equations

The effects of the Earth's rotation are represented by the terms $2\hat{\Omega}\times\boldsymbol{u}+\hat{\Omega}\times(\hat{\Omega}\times\boldsymbol{x})$, where $\hat{\Omega} = (0,\Omega\cos\theta,\Omega\sin\theta)$ is the vector parallel to the Earth's axis of rotation and $\Omega = 7.292\times10^{-5}\ s^{-1}$ is the angular speed of the Earth, and θ is latitude. We can derive the rotational terms in the equations by considering two coordinate systems. Let the subscript f refer to quantities measured in a fixed coordinate system, not rotating with the Earth, with its origin at the center of the Earth, and the subscript r refer to quantities measured in a rotating coordinate system such that a point attached to the surface of the Earth has zero velocity. Hence, the velocities of a point as measured in the two coordinate systems are related by

$$\frac{d\boldsymbol{x}_f}{dt} = \frac{d\boldsymbol{x}_r}{dt}+\hat{\Omega}\times\boldsymbol{x}_r. \tag{1.4}$$

Applying this relation again we have

$$\frac{d^2\boldsymbol{x}_f}{dt^2} = \frac{d^2\boldsymbol{x}_r}{dt^2}+2\hat{\Omega}\times\frac{d\boldsymbol{x}_r}{dt}+\hat{\Omega}\times(\hat{\Omega}\times\boldsymbol{x}_r). \tag{1.5}$$

So the acceleration measured in the two coordinate systems is the same with the addition of two terms to correct for rotation. The terms are necessary since Newton's second law applies only in an inertial reference frame and we wish to describe motion in the rotating coordinate system. The term $2\hat{\Omega}\times\boldsymbol{u}$ is called the Coriolis acceleration. We can give a brief, qualitative description of this term. By the definition of $\hat{\Omega}$ and the cross product, the Coriolis acceleration will have a component in the horizontal plane deflecting a parcel to the right of its direction of motion in the Northern Hemisphere, and to the left in the Southern Hemisphere. Along the equator, the Coriolis acceleration has only a vertical component. The term $\hat{\Omega} \times (\hat{\Omega} \times \boldsymbol{u})$ describes the centrifugal acceleration felt by a parcel due to the Earth's rotation. This force is directed away from the axis of rotation. We can write this vector as the gradient of a scalar known as the centrifugal potential, as follows: $\hat{\Omega} \times (\hat{\Omega} \times \boldsymbol{u}) = -\nabla (\frac{1}{2}\Omega^2 r^2)$, where $r = a\cos\theta$ and a is the radius of the Earth.

The rotational "forces" acting on a parcel, as discussed above, result from our way of viewing the parcel, from a rotating coordinate system, and are often called apparent forces. On the other hand, the two terms on the right-hand side of equation (1.1) describe actual physical forces that can produce accelerations. The first says that a parcel of air will tend to move from high pressure to low pressure because of forces acting on its boundaries. The force per unit volume is ∇p; for Newton's second law, however, we want force per unit mass, $\rho^{-1} \nabla p$.

The term $\nabla \Phi_v$ is the gradient of the Earth's gravitational potential. This term says that a parcel tends to fall to the ground because of forces acting on the mass within it. It would fall if the gravitational acceleration were not balanced by the vertical component of the pressure gradient. [See equation (1.16), later.] Since the centrifugal acceleration tends to counteract the gravitational acceleration, and both can be written as gradients of potentials, they can be combined into one term called the geopotential, Ω, which can be written as

$$\nabla \Phi = \nabla \Phi_{\nu} - \nabla \left(\frac{1}{2}\Omega^2 r^2\right). \tag{1.6}$$

The geopotential can be defined as the amount of work to move a mass from sea level through the gravitational field to a height z. That is,

$$\Phi(z) = \int_0^z g dz'. \tag{1.7}$$

The centrifugal acceleration varies with latitude but is always small compared to the Earth's gravitational acceleration, and the gravitational acceleration is nearly constant throughout the depth of the atmosphere. Hence, to a good approximation, $\Phi(z) = gz$ where $g = 9.81 \ \mathrm{m\,s^{-2}}$. From here on the symbol Φ_{ν} will be replaced by Φ.

1.2.1.2 The Mass Conservation Equation

Equation (1.2) is derived by considering a small, fixed volume imbedded in a flow. Further, suppose the volume surrounds some point in space, $\boldsymbol{x}$. Since the volume is constant, density can vary within it only if the amount of mass within the volume increasing or decreasing. If we do not allow any sources or sinks, the only way the mass can change within the volume is if mass flows through the walls. That is,

$$\frac{\partial}{\partial t}\int_V \rho dV = -\int_A \rho \boldsymbol{u} \bullet \boldsymbol{n} dA. \tag{1.8}$$

The left-hand side describes the time rate of change of mass within the volume. On the right-hand side of the equation, $\rho \boldsymbol{u} \bullet \boldsymbol{n}$ is the mass flux crossing an infinitesimal surface element of area dA, and $\boldsymbol{n}$ is outward unit normal vector. The minus sign ensures that flow directed into the volume contributes positively to the mass. Bringing the time derivative inside the integrand, and using the divergence theorem to transform the right-hand side, we get

$$\int_V \frac{\partial \rho}{\partial t} dV = -\int_V \nabla \bullet \rho \boldsymbol{u} dV, \tag{1.9}$$

$$\int_V \left(\frac{\partial \rho}{\partial t} + \nabla \bullet \rho \boldsymbol{u}\right) dV = 0. \tag{1.10}$$

Since the volume is arbitrary, and using the definition of total derivative, we get

$$\frac{\partial \rho}{\partial t} + \nabla \bullet \rho \boldsymbol{u} = 0, \tag{1.11}$$

$$\frac{\partial \rho}{\partial t} + \boldsymbol{u} \bullet \nabla \rho + \rho \nabla \bullet \boldsymbol{u} = 0, \tag{1.12}$$

$$\frac{D \rho}{D t} + \rho \nabla \bullet \boldsymbol{u} = 0. \tag{1.13}$$

1.2.2 Simplifying Assumptions

In this section, we explain several assumptions which can be used to simplify the equations of motion. These simplifying assumptions are used in the derivation of the shallow water equations.

Consider a fluid in equilibrium. Then the left-hand side of (1.1) is zero

$$\rho^{-1} \nabla p + \nabla \Phi = 0. \tag{1.14}$$

If pressure was not constant along a surface of constant geopotential, a parcel would move along such a surface and the fluid would not be in equilibrium, so p and ρ must be functions of Φ only, i.e.

$$\frac{dp}{d\Phi} = -\rho. \tag{1.15}$$

Using $\Phi(z) = gz$, we can derive a relation known as the hydrostatic equation

$$\frac{dp}{dz} = -g\rho. \tag{1.16}$$

On the large scale, the atmosphere is very nearly in a state of equilibrium. The acceleration felt by the air, due to gravity, is about 10,000 times stronger than that due to the Coriolis effect at middle latitudes. Departures of pressure and density from equilibrium are small compared with the equilibrium values of these quantities. For these reasons, the hydrostatic equation can be applied to the real atmosphere on large scales. It is convenient to partition pressure and density into equilibrium and departures from equilibrium values. That is,

$$p = p_0(z) + p'(x,y,z,t), \text{ and } \rho = \rho_0(z) + \rho'(x,y,z,t). \tag{1.17}$$

The equilibrium solution satisfies the hydrostatic equation

$$\frac{dp_0}{dz} = -g\rho_0. \tag{1.18}$$

To reiterate, the equations of motion can be written

$$\rho\left(\frac{D\boldsymbol{u}}{Dt} + 2\hat{\Omega}\times\boldsymbol{u}\right) = -\nabla p - \rho\boldsymbol{g}, \tag{1.19}$$

where $\boldsymbol{g} = (0,0,g)$. Substituting (1.17) into the equation for the vertical component, w, and using (1.18), we get

$$\rho\left[\frac{Dw}{Dt} + 2(\Omega_x v - \Omega_y u)\right] = -\frac{\partial}{\partial z}(p_0 + p') + (\rho_0 + \rho')g = \frac{\partial}{\partial z}p' + \rho' g, \tag{1.20}$$

where $\hat{\Omega} = (\Omega_x, \Omega_y, \Omega_z)$. Since p_0 depends only on z, the equations of motion become

$$\rho\left[\frac{D\boldsymbol{u}}{Dt}+2\hat{\Omega}\times\boldsymbol{u}\right] = -\nabla p'-\rho'\boldsymbol{g}. \tag{1.21}$$

Finally, we can make the assumption that the fluid is homogeneous, or incompressible, that is, $\rho = \rho_0$ is a constant and $\rho' = 0$. This assumption has two consequences. First, (1.20) becomes

$$\rho\left[\frac{D\boldsymbol{u}}{Dt}+2\hat{\Omega}\times\boldsymbol{u}\right] = -\nabla p'. \tag{1.22}$$

Second, the mass conservation equation, (1.13), reduces to just

$$\nabla\bullet\boldsymbol{u} = 0. \tag{1.23}$$

1.2.3 The Shallow Water System

The shallow water system is described in detail in Pedlosky (1986) and Gill (1982). We consider a system where $h = h(x,y,t)$ is the height of the free surface above the reference level $z = 0$, $h_s = h_s(x,y)$ is the height of the surface topography and $h^* = h-h_s$ is the depth of the fluid. See Fig. 1.1. The pressure boundary condition at the free surface is $p(x,y,h,t) = p_f$, where p_f is some arbitrary constant. Two more

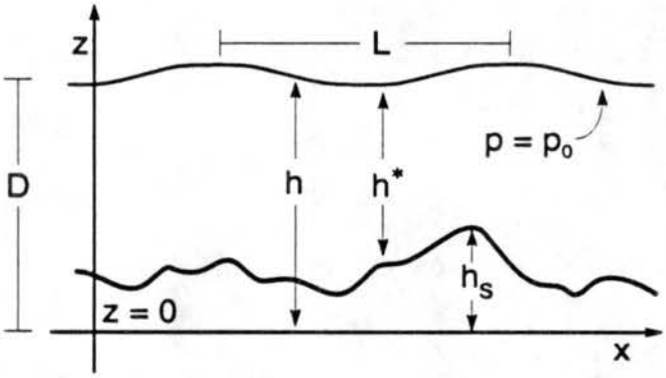

FIGURE 1.1: The shallow water system.

boundary conditions for w are intuitively clear. First, no fluid can cross the free surface, so a parcel imbedded in the free surface will satisfy

$$\frac{Dh}{Dt} = w(x, y, h, t)\,, \tag{1.24}$$

and similarly, no fluid can cross the surface topography so a parcel must follow the topography:

$$\frac{Dh_s}{Dt} = w_s, \tag{1.25}$$

or

$$\frac{\partial h_s}{\partial t} + v \bullet \nabla h_s - w_s = 0, \tag{1.26}$$

or

$$w_s(x, y, h_s, t) = v \bullet \nabla h_s. \tag{1.27}$$

It is important to assume that disturbances in the free surface have a large wavelength compared to the depth of the fluid. Suppose D is the average depth of the fluid and L is the wavelength of the typical disturbance in the free surface. Then $D/L \ll 1$. This ensures that the fluid is close to equilibrium and the hydrostatic assumption is valid.

1.2.3.1 Shallow Water Momentum Equations

We assume that the total pressure satisfies the hydrostatic relation

$$\frac{dp}{dz} = -g\rho. \tag{1.28}$$

Assuming that the fluid is incompressible, and integrating from some arbitrary depth z within the fluid to the surface gives

$$p(x, y, z, t) - p(x, y, h, t) = -g\rho(h - z). \tag{1.29}$$

Applying the boundary conditions $p(x,t,h,t) = p_f$,

$$p = g\rho(h - z) + p_f. \tag{1.30}$$

So, since $p = -g\rho z + p'(x,y,z,t)$, then $p' = g\rho h + p_f$ and

$$\nabla p' = g\rho \nabla h. \tag{1.31}$$

Typically $w \ll u, v$, so the terms in (1.22) involving w can be ignored. Define $f = 2\Omega_z = 2\Omega \sin\theta$. The horizontal momentum equation becomes

$$\frac{D\boldsymbol{v}}{Dt} - f\hat{\boldsymbol{k}} \times \boldsymbol{v} = -g\nabla h. \tag{1.32}$$

The horizontal total derivative is defined by $\frac{d}{dt}(\ ) \equiv \frac{\partial}{\partial t}(\ ) + (\boldsymbol{v} \bullet \nabla)(\ )$ where ∇ is the horizontal gradient.

1.2.3.2 Shallow Water Continuity Equation

Since the horizontal accelerations do not depend on z, we can easily integrate the continuity equation (1.23) with respect to z. Integrating from $z = h_s$ to $z = h$

$$w(x,y,h,t) - w_s(x, y, h_s, t) = -(h - h_s)\nabla \bullet \boldsymbol{v}. \tag{1.33}$$

where $\nabla \bullet \boldsymbol{v}$ is the horizontal divergence. Substitute (1.24) and (1.27), to obtain

$$\frac{\partial h}{\partial t} + \boldsymbol{v} \bullet \nabla h - \boldsymbol{v} \bullet \nabla h_s = -(h - h_s)\nabla \bullet \boldsymbol{v}. \tag{1.34}$$

Since $\partial h_s / \partial t = 0$, we can write (1.34) as

$$\frac{\partial}{\partial t}(h - h_s) + \nabla \bullet [\boldsymbol{v}(h - h_s)] = 0, \tag{1.35}$$

or, with $h^* = h - h_s$,

$$\frac{\partial}{\partial t}h^* + \nabla\bullet(\boldsymbol{v}h^*) = 0. \tag{1.36}$$

This is the continuity equation for the shallow water system.

1.3 Forms of the Shallow Water Equations

The shallow water equation can be represented in a number of different forms. These equations are presented in various texts on atmospheric dynamics, including Holton and Gill. We start with the advective form and work toward the stream function, velocity potential (SFVP) form.

1.3.1 Advective Form

This is the form just derived. To reiterate, the momentum equation (1.32) is

$$\frac{D\boldsymbol{v}}{Dt} = -f\hat{\boldsymbol{k}}\times\boldsymbol{v} - g\nabla h, \tag{1.37}$$

and the continuity equation is (1.36),

$$\frac{Dh^*}{Dt} + h^*\nabla\bullet\boldsymbol{v} = 0, \tag{1.38}$$

where we have used the definition of horizontal total derivative.

1.3.2 Vorticity and Divergence Form

Relative vorticity and divergence can be used to represent horizontal winds:

$$\zeta = \hat{\boldsymbol{k}}\bullet(\nabla\times\boldsymbol{v}), \tag{1.39}$$

$$\delta = \nabla \bullet \boldsymbol{v}. \tag{1.40}$$

Using the vector indentity $(\boldsymbol{v} \bullet \nabla)\boldsymbol{v} = \nabla\,[(\boldsymbol{v} \bullet \boldsymbol{v})/2] + (\nabla \times \boldsymbol{v}) \times \boldsymbol{v}$, we can write the momentum equations (1.37) as

$$\frac{\partial \boldsymbol{v}}{\partial t} + (f+\zeta)\,\hat{\boldsymbol{k}} \times \boldsymbol{v} = -\nabla\,\left(gh + \frac{\boldsymbol{v} \bullet \boldsymbol{v}}{2}\right). \tag{1.41}$$

The absolute vorticity is the sum of the relative and planetary vorticity, $\eta = \zeta + f$. Using (1.39), (1.40) and (1.41), we can express the equations for vorticity and divergence as follows:

$$\frac{\partial \zeta}{\partial t} = -\nabla \bullet (\eta \boldsymbol{v}), \tag{1.42}$$

$$\frac{\partial \delta}{\partial t} = \hat{\boldsymbol{k}} \bullet \nabla \times (\eta \boldsymbol{v}) - \nabla^2\,\left(gh + \frac{\boldsymbol{v} \bullet \boldsymbol{v}}{2}\right). \tag{1.43}$$

The derivation of these equations is worked out in Appendix A.

1.3.3 Stream Function and Velocity Potential Form

Since the real atmosphere is nearly geostrophic, it would be advantageous to partition the wind into rotational and divergent parts, the rotational part being more significant than the divergent. Helmholtz's Theorem states that any vector field $\boldsymbol{V}$ can be separated into rotational and divergent parts, i.e., $\boldsymbol{V} = \boldsymbol{V}_\psi + \boldsymbol{V}_\chi$, where $\nabla \bullet \boldsymbol{V}_\psi = 0$ and $\nabla \times \boldsymbol{V}_\chi = 0$. If the vector field is the horizontal wind, we can define a horizontal stream function, ψ, to express the rotational part, $\hat{\boldsymbol{k}} \times \nabla \psi$, and a velocity potential, χ, to express the divergent part, $\nabla \chi$, i.e.

$$\boldsymbol{v} = \hat{\boldsymbol{k}} \times \nabla \psi + \nabla \chi. \tag{1.44}$$

With this representation for velocity, the vorticity and divergence equations can be written in SFVP form:

$$\frac{\partial\eta}{\partial t} + \nabla\bullet(\eta\nabla\chi) - J(\eta,\psi) = 0, \tag{1.45}$$

$$\frac{\partial\delta}{\partial t} - \nabla\bullet(\eta\nabla\psi) - J(\eta,\chi) + \nabla^2(K+gh) = 0, \tag{1.46}$$

where the kinetic energy, K, is given by

$$K = \frac{1}{2}\left[\nabla\bullet(\psi\nabla\psi) - \psi\nabla^2\psi + \nabla\bullet(\chi\nabla\chi) - \chi\nabla^2\chi\right] + J(\psi,\chi). \tag{1.47}$$

The continuity equation (1.40) becomes

$$\frac{\partial}{\partial t}h^* + \nabla\bullet(h^*\nabla\chi) - J(h^*,\psi) = 0. \tag{1.48}$$

The Jacobian operator is defined by $J(A,B) \equiv \hat{\boldsymbol{k}} \bullet (\nabla A \times \nabla B)$. These equations are derived in Appendix A.

1.4 Methods to Solve the Shallow Water Equations on the Sphere

1.4.1 The pole problem

Analytic solutions for the shallow water equations do not exist except for extremely simple initial conditions, so we are forced to seek numerical solutions. There are a number of problems associated with trying to solve differential equations numerically in spherical geometry. Difficulties can arise from the use of spherical coordinate systems, and from trying to discretize the surface of the sphere. These difficulties include the pole problem which is discussed below.

Before any numerical considerations are addressed, we run into difficulties in simply specifying a coordinate system. We can distinguish between a true scalar-valued and vector-valued functions. The value of a scalar function, say temperature, is independent of coordinate system. That is, the temperature at some point in space is the same regardless of how we define the position of the point. On the other hand, the individual components of a vector valued function, such as wind, obviously differ depending on the coordinate system. In a spherical coordinate system the lines of constant longitude converge at the poles, so longitude is multivalued at the poles. This means that the components of the wind are discontinuous at the poles. As a simple example, consider a jet directed over the North Pole. This is represented by the shaded arrow in Fig. 1.2.

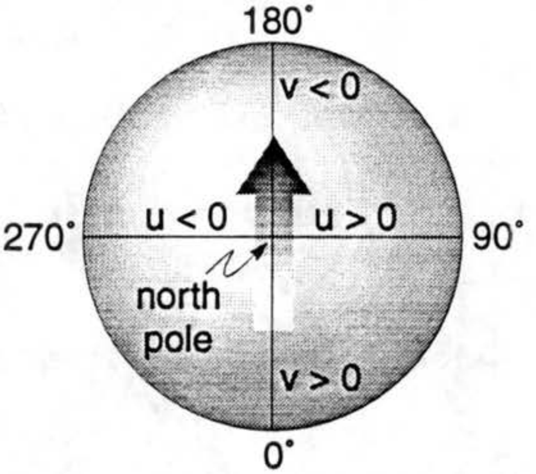

FIGURE 1.2: The wind measured at points along the prime meridian have strong northward component of wind velocity. There is a discontinuity at pole and points along international date line measure strong southward component.

Measured at points along the prime meridian, the wind will have a positive v component. However, measured along the international date line the wind will have a negative v component. A discontinuity must occur at the pole, since there is no north nor south there. Similarly, the u component of the wind measured near the pole along 90°

longitude is positive and is negative measured along 270° longitude. This problem does not occur in a Cartesian coordinate system centered on the pole. At each point along a great circle which includes the North Pole, the components measured in Cartesian coordinates are well defined and vary continuously.

Until now we have not had to express the shallow water equations in a particular coordinate system. Since we wish to solve the equations in a spherical geometry, it would make sense to express them using the spherical coordinate system. In spherical coordinates, the gradient and divergence operators have the form

$$\nabla\,(\;) \equiv \frac{\hat{i}}{a\cos\theta}\frac{\partial}{\partial\lambda}(\;) + \frac{\hat{j}}{a}\frac{\partial}{\partial\theta}(\;) \quad \text{and,} \quad \nabla\bullet\boldsymbol{v} \equiv \frac{1}{a\cos\theta}\left[\frac{\partial u}{\partial\lambda} + \frac{\partial}{\partial\theta}(v\cos\theta)\right], \tag{1.49}$$

where $\hat{i}$ is the unit vector in the longitudinal, λ, direction, and $\hat{j}$ is the unit vector in the latitudinal, θ, direction. Applying these operators gives

$$\frac{\partial u}{\partial t} + \boldsymbol{v}\bullet\nabla u - \left(f + \frac{u}{a}\tan\theta\right)v + \frac{g}{a\cos\theta}\frac{\partial h}{\partial\lambda} = 0, \tag{1.50}$$

$$\frac{\partial v}{\partial t} + \boldsymbol{v}\bullet\nabla v + \left(f + \frac{u}{a}\tan\theta\right)u + \frac{g}{a}\frac{\partial h}{\partial\theta} = 0. \tag{1.51}$$

The continuity equation becomes

$$\frac{\partial}{\partial t}h^* + \boldsymbol{v}\bullet\nabla h^* + \frac{h^*}{a\cos\theta}\left[\frac{\partial u}{\partial\lambda} + \frac{\partial}{\partial\theta}(v\cos\theta)\right] = 0. \tag{1.52}$$

As we will see later, this representation of the equations in spherical coordinates is just a special case of the equations written in a general orthogonal coordinate system.

1.4.2 Map Projections

In numerically solving shallow water equations on a sphere, an early approach was to translate the equations from spherical coordinates to a different coordinate system via map projections, thereby avoiding the problems associated with the poles in the spherical coordinate system. The general strategy is to project the equations from the sphere to a plane, and solve the equations on a regular grid in the new map coordinates. However, the surface of a sphere and that of a plane are not topologically equivalent. In other words, there does not exist a one-to-one mapping g such that for every point on the sphere $(\theta,\lambda) \in S$ there exists $(x, y) \in P \equiv \{ (x, y) | -\infty < x, y < \infty\}$ satisfying $g(\theta, \lambda) = (x, y)$. There are mappings which map almost all of S onto P; examples are given below. Unfortunately, these mappings, or projections, tend to badly distort distances and areas near the singular points of the transformation. Nevertheless, we can use a projection to map the piece of the sphere where the transformation is well behaved onto a finite area in the plane. One approach to map the entire sphere is the composite mesh method, discussed later.

We can easily derive the equations of motion in various map projections if we first express them in a general orthogonal coordinate system (x, y). Define the metric coefficients to be h_x and h_y so that the distance increment dl satisfies

$$dl^2 = h_x^2 dx^2 + h_y^2 dy^2. \qquad \textbf{(1.53)}$$

Note that the metric coefficients are distinguished from depth of the fluid by a subscript. In the (x, y) coordinate system, the horizontal velocity components are given by

$$U = h_x \frac{dx}{dt}, \tag{1.54}$$

$$V = h_y \frac{dy}{dt}. \tag{1.55}$$

Williamson (1979) gives the equations of motion for the general velocity components:

$$\frac{dU}{dt} - \left[f + \frac{1}{h_x h_y}\left(V\frac{\partial h_x}{\partial x} - U\frac{\partial h_y}{\partial y}\right)\right]V + \frac{g}{h_x}\frac{\partial h}{\partial x} = 0, \tag{1.56}$$

$$\frac{dV}{dt} + \left[f + \frac{1}{h_x h_y}\left(V\frac{\partial h_x}{\partial x} - U\frac{\partial h_y}{\partial y}\right)\right]U + \frac{g}{h_y}\frac{\partial h}{\partial y} = 0, \tag{1.57}$$

where

$$\frac{d}{dt}(\ ) = \frac{\partial}{\partial t}(\ ) + \frac{U}{h_x}\frac{\partial}{\partial x}(\ ) + \frac{V}{h_y}\frac{\partial}{\partial y}(\ ). \tag{1.58}$$

The continuity equation is given by

$$\frac{d}{dt}h^* + \frac{h^*}{h_x h_y}\left[\frac{\partial}{\partial x}(h_y U) + \frac{\partial}{\partial y}(h_x V)\right] = 0. \tag{1.59}$$

For example, if we set

$$x = \lambda \text{ and } y = \theta, \tag{1.60}$$

and set the metric coefficients

$$h_x = a\cos\theta \text{ and } h_y = a, \tag{1.61}$$

then by (1.54) and (1.55) we have

$$u = a\cos\theta\frac{d\lambda}{dt} \text{ and } v = a\frac{d\theta}{dt}. \tag{1.62}$$

Substituting (1.61) and (1.62) into (1.56), (1.57) and (1.59) gives (1.50), (1.51) and (1.52), the shallow water equations in spherical coordinates.

There are two projections commonly used for numerical meteorology -- Polar Stereographic and Mercator. Both are examples of conformal projections, that is, they preserve angles, but not distances. Also, the metric coefficients are independent of direction at a given point, i.e., $h_x = h_y$. The effects of these projections on the outlines of the continents are shown in Fig. 1.3.

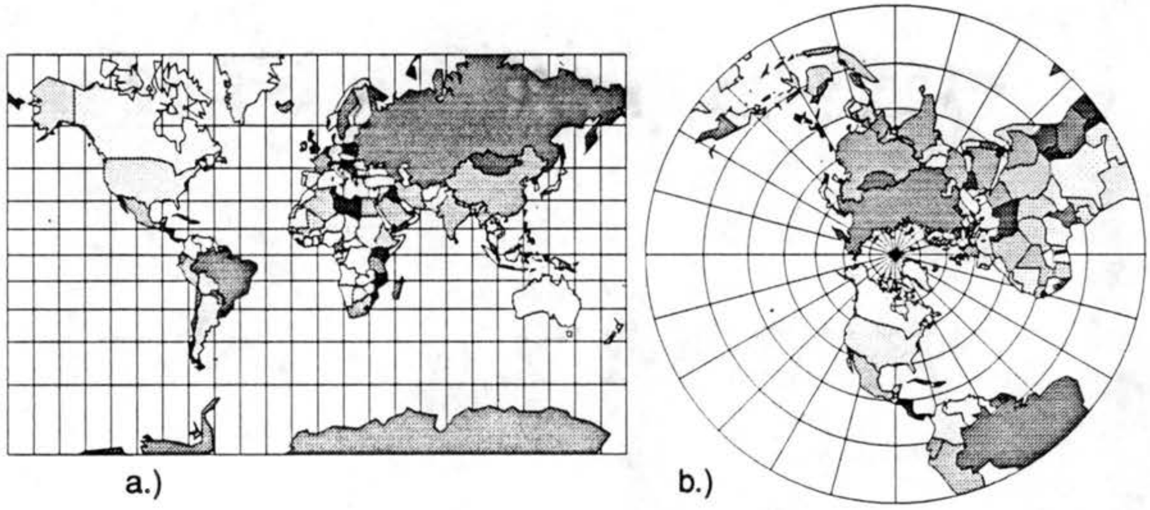

FIGURE 1.3: Map projections of the continents: a.) Mercator projection. b.) North polar stereographic projection.

Geometrically, the polar stereographic projection can be envisioned by positioning a plane tangent to the Earth at the North Pole. A line drawn from the South Pole that intersects the Earth will also intersect the plane. This line establishes a one-to-one correspondence between all points on the plane and all points on the sphere except the South Pole itself. In the plane, we can define a Cartesian coordinate system (X, Y), where the positive X axis is in the direction of the image of $\lambda = 0$, and the positive Y axis is in the direction of the image of $\lambda = \pi/2$. Obviously, a similar mapping is obtained by placing the plane tangent to the sphere at points other than the North Pole.

Haltiner (1980) gives the equations relating the projection coordinates (X, Y) and the spherical coordinates (θ, λ) :

$$X = \frac{2a\cos\theta\cos\lambda}{1+\sin\theta}, \tag{1.63}$$

$$Y = \frac{2a\cos\theta\sin\lambda}{1+\sin\theta}. \tag{1.64}$$

Note that the Jacobian of the mapping is singular at the South Pole. Taking differentials of (1.63) and (1.64) gives

$$\begin{bmatrix} dX \\ dY \end{bmatrix} = \frac{2a}{1+\sin\theta}\begin{bmatrix} -\cos\theta\sin\lambda & -\cos\lambda \\ \cos\theta\cos\lambda & -\sin\lambda \end{bmatrix}\begin{bmatrix} d\lambda \\ d\theta \end{bmatrix}. \tag{1.65}$$

Now we determine the metrics of the stereographic map projection. Substituting $x = \lambda$, $y = \theta$ and the metrics for spherical coordinates into $dl^2 = h_x^2 dx^2 + h_y^2 dy^2$ gives

$$dl^2 = (a\cos\theta)^2 d\lambda^2 + a^2 d\theta^2. \tag{1.66}$$

Solving (1.65) and substituting into (1.66) gives

$$dl^2 = \left(\frac{1+\sin\theta}{2}\right)^2 dX^2 + \left(\frac{1+\sin\theta}{2}\right)^2 dY^2. \tag{1.67}$$

The metric coefficient for the polar stereographic projection is therefore given by

$$h_x = h_y = \frac{1+\sin\theta}{2}. \tag{1.68}$$

We can define the map factor, $m(\theta)$, to be the inverse of the metric coefficients, i.e., $m(\theta) = 2/(1+\sin\theta)$. Using (1.56), (1.57) and (1.59), we can write the shallow water equations in north polar stereographic coordinates:

$$\frac{dU}{dt} - \left[f + \frac{UY - VX}{2a}\right]V + mg\frac{\partial}{\partial x}h^* = 0, \tag{1.69}$$

$$\frac{dV}{dt} + \left[f + \frac{UY - VX}{2a}\right] U + mg\frac{\partial}{\partial x}h^* = 0, \tag{1.70}$$

$$\frac{d}{dt}h^* + m^2 h^* \left[\frac{\partial}{\partial X}\left(\frac{U}{m}\right) + \frac{\partial}{\partial Y}\left(\frac{V}{m}\right)\right] = 0. \tag{1.71}$$

The total derivative is given by (1.58).

As discussed above, a finite region of the plane will only map onto a piece of the sphere, and visa-versa. One technique to map the entire sphere is to partition it, for example, into hemispheres, and project the pieces separately. Each set of projected equations then gets its boundary conditions from the solutions of the other projected equations. Phillips (1957) divided the sphere into three regions: a tropical belt, and extratropical caps to the north and south of the tropical belt. On each region, the shallow water equations are mapped to a new coordinate system. He used a Mercator coordinate system in the tropics, a polar stereographic coordinate system fixed to the sphere at the North Pole for the northern extratropical cap, and similarly, a polar stereographic coordinate system fixed to the sphere at the South Pole for the southern extratropical cap. When a computational stencil required data from outside the region covered by its coordinate system, that piece of information was obtained by interpolation within the neighboring coordinate system. The model proved unstable as a discontinuity developed at the boundary between the two coordinate systems.

More recently, Browning (1989) discusses a different composite mesh model in which the northern and Southern Hemispheres are mapped to the plane with a polar stereographic projection. The equations used for the northern projection are just (1.69), (1.70) and (1.71). The equations for the southern projection are the same as those for the northern, except for a few sign differences. This model is different from Phillips' in that the regions interior to the coordinate systems overlap a little bit as shown in Fig. 1.4.

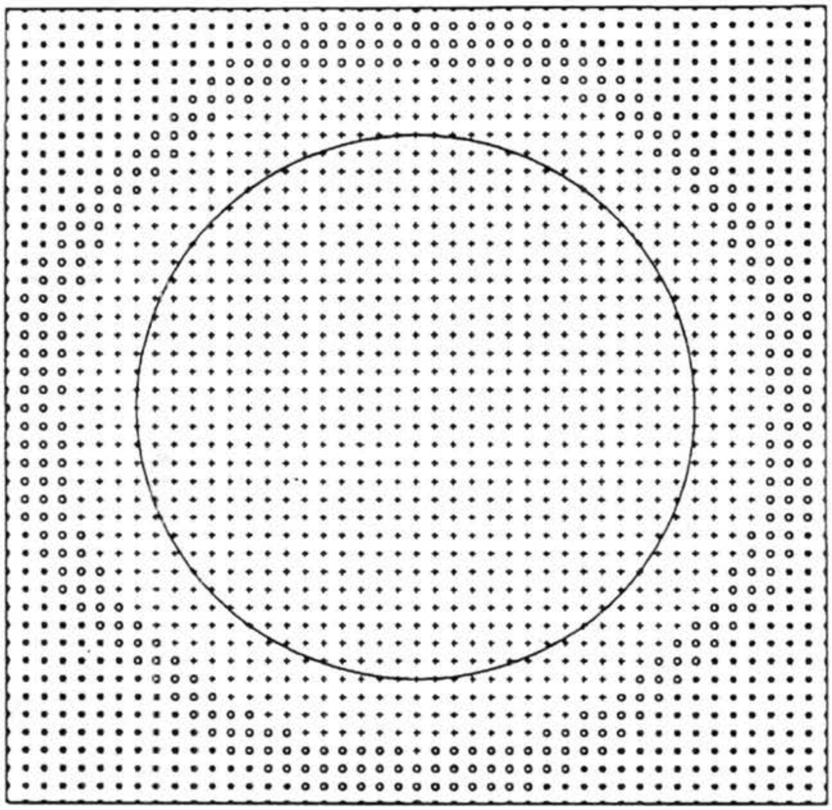

FIGURE 1.4: Composite grid method grid. Two such grids are used to cover the sphere. Points labeled with O are the boundary conditions for the points labeled with +. Values at the O points are obtained by interpolation from the other grid. The big circle is the image of the equator. Points labeled * are not used.

Values for dependent variables at grid points not covered by the current coordinate system are obtained by interpolation in the other coordinate system. The overlapping of the coordinate systems makes this scheme more stable than in Phillips' model, in which the coordinate systems were simply butted together at a certain latitude. This model is also easier to write computer code for since the equations are only expressed in one coordinate system. Browning tested the model using cases 3 and 4 described in chapter 3, and reported good results.

1.4.3 Spectral Methods

In this section we briefly discuss the spectral representation of the field variables. The discussion is not intended to be comprehensive, but merely enough to explain the spectral method's approach to solving the pole problem. The Galerkin spectral method, and the NCAR spectral shallow water model are discussed in Appendix C. Text books on spectral methods include Gottlieb (1977) and Canuto (1988), and a discussion of spectral methods applied to meteorological problems can be found in Jarraud (1983).

We can define a set of functions $Y_n^m(\theta, \lambda) \equiv P_n^m(\mu) e^{im\lambda}$, where $\mu \equiv \sin\theta$ and the $P_n^m(\mu)$ are the associated Legendre polynomials of the first kind. These functions are known as spherical harmonics. Here, m is the zonal or Fourier wavenumber, and n is the total wavenumber. The Y_n^m are the eigenfunctions of the Laplacian operator on the sphere, that is

$$\nabla^2 Y_n^m = \frac{-n(n+1)}{a^2} Y_n^m. \tag{1.72}$$

This fact in is useful in the numerical implementation, as shown in Appendix C. The functions Y_n^m form a complete and orthogonal basis.

An arbitrary scalar quantity, $\xi(\theta, \lambda)$, can be approximated with a truncated series of spherical harmonics as follows:

$$\xi(\theta, \lambda) \approx \sum_{m=-M}^{M} \sum_{n=|m|}^{N(m)} \xi_n^m P_n^m(\mu) e^{im\lambda}. \tag{1.73}$$

The associated Legendre polynomials have the property that $P_n^m(\mu) = 0$ if $|m| > n$. This determines the lower limit of the second summation. The associated Legendre polynomials also satisfy $Y_n^{-m} = \overline{Y_n^m}$, where the overbar denotes the complex conjugate. Running the zonal wavenumber from $-M$ to M in the outer summation ensures that the result is real. There have been two ways of choosing the largest total wavenumber, $N(m)$, to truncate the series. The first, called rhomboidal truncation, sets $N(m) = M + |m|$. This was the truncation used when spectral models were first created, and historic inertia alone caused it to remain popular. The second method, called triangular truncation, sets $N(m) = M$. This is the method currently used in most spectral models. It has a big advantage. Suppose that ξ_1 is an expansion of triangularly truncated at wavenumber M in spherical coordinate system (θ_1, λ_1), and ξ_2 is an expansion in a second spherical coordinate system (θ_2, λ_2). Then $\xi_1 = \xi_2$. In other words, the two fields are the same, independent of the position of the pole. A triangularly truncated expansion thus isotropically covers the sphere.

1.4.4 On Latitude-Longitude Grids

The obvious way to discretize the shallow water equations expressed in spherical coordinates would be to use a regular latitude-longitude grid in which the grid intervals $(\Delta\theta, \Delta\lambda)$ are constant. A discretization scheme and computer program are then straight forward except at the row of grid points next to the pole, where special considerations are necessary.

1.4.4.1 Uniformly Spaced Grids

A uniform latitude-longitude grid is the grid used in the CSU General Circulation Model, as shown in Fig. 1.5. Latitude-longitude type grids have some drawbacks. When constructing the grid for a numerical model, we should consider the horizontal structure of the phenomena being simulated. The scale of meteorological events does not vary dramatically as a function of latitude, nor does the structure vary much between the zonal and meridional directions. The grid should reflect this. The average distance to neighbors of a grid point should not depend on latitude, as it does in a latitude-longitude grid, nor should the distance between grid points in the zonal direction be substantially different than the distance in the meridional direction. The variation in average distance to neighbors is a measure of a grid's homogeneity, and variation of distance in meridional and zonal directions is a measure of its isotropy.

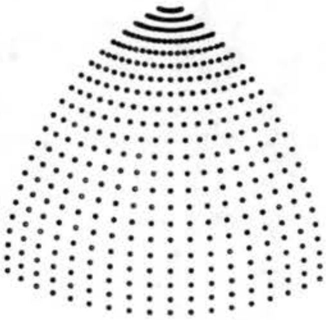

FIGURE 1.5: One octant of the CSU GCM grid. 72 grid points are used longitudinally and 44 grid points latitudinally.

Apart from the esthetic considerations, latitude-longitude grids suffer from computation complications. The top row of grid points on the CSU GCM's grid, where the *u* component of velocity is defined, (see the Arakawa-Lamb C-grid in Appendix B), is at 86 °N latitude. Arakawa and Lamb (1977) show that in order to maintain computational stability the time step Δt must satisfy

$$\frac{|C|\Delta t}{a\Delta\lambda} < \frac{(\cos\theta_j)_{\min}}{2}, \tag{1.74}$$

where C is phase speed of gravity waves, and $(\cos\theta_j)_{\min}$ is the cosine of the highest latitude. Since $a\Delta\lambda(\cos\theta_j)_{\min} = \Delta x_{\min}$, the minimum meridional grid spacing, (1.74) is analogous to the CFL condition. For the GCM grid, $\Delta x_{\min} \approx 39 \text{ km}$, and for the atmosphere $C = 300 \text{ ms}^{-1}$. Plugging into (1.74) we get $\Delta t \approx 70 \text{ seconds}$.

Typically, polar latitudes are filtered or smoothed to remove noise due to numerical instability, so that a much longer time step may be used. This is done by longitudinally smoothing selected terms in the equations. In particular, the longitudinal pressure gradient in the momentum equation and the longitudinal divergence in the continuity equation are longitudinally averaged. Consider an arbitrary field, ξ. We can express the Fourier decomposition at the grid points along a particular latitude as follows:

$$\xi_{i,j} = \operatorname{Re}\{\hat{\xi}_{i,j}\exp[\bar{i}(mi\Delta\lambda + \sigma t)]\}, \tag{1.75}$$

where $\bar{i} \equiv \sqrt{-1}$. The index i refers to the longitudinal direction, and index j refers to the latitudinal direction. The parameter m is the discrete longitudinal wavenumber. The amplitude of the longitudinal pressure gradient and divergence are multiplied by a factor $S_j(m)$, where $S_j(m)$ defined by

$$S_j(m) = \frac{\Delta\lambda}{\Delta\theta}\left[\frac{\cos\theta_j}{\sin(m\Delta\lambda/2)}\right]. \tag{1.76}$$

Actually, since we are trying to damp higher frequency components, a given component is multiplied by $S_j(m)$ only if $S_j(m) > 1$, and is multiplied by 1 otherwise. The process is only applied to the higher latitudes where the longitudinal grid spacing is small.

1.4.4.2 Kurihara Grid

To increase the grid spacing at higher latitudes, Kurihara (1965) proposed a grid which varied the number of grid points along each latitude circle. By placing fewer points at higher latitudes, Kurihara was able to more homogeneously cover the sphere. The grid is constructed by evenly placing $N+1$ grid points along the $0°$ longitude meridian from the North Pole to the equator. The point at the North Pole is given the label $j = 1$, the next latitude circle south is given the label $j = 2$, and so on until the equator is labeled $j = N+1$. Along latitude circle j there are $4(j-1)$ equally spaced grid points. One octant of the sphere is shown in Fig. 1.6. For a given N, the total number of grid points on the sphere is $4N^2 + 2$. The Southern Hemisphere is a mirror image of the Northern Hemisphere.

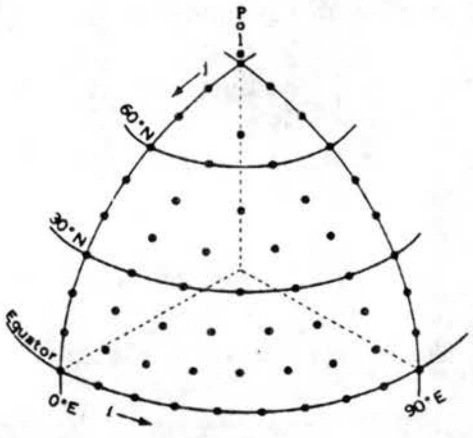

FIGURE 1.6: Kurihara grid on one octant of the sphere.

We can measure the homogeneity of the grid by examining the ratio of the zonal distance, $a\cos\theta_j\Delta\lambda_j$, and the meridional distance $a\Delta\theta$, for a grid point at latitude θ_j. Here, $\Delta\theta \equiv \frac{\pi}{2}\frac{1}{N}$ and $\Delta\lambda_j \equiv \frac{\pi}{2}\frac{1}{j-1}$. At $j = N+1$, the equator, the ratio is one, and as you move farther poleward, the ratio approaches $\pi/2$.

Kurihara built a model based on the equations for a barotropic atmosphere with a free surface. He tested the model with the Rossby-Haurwitz wave, with wave number 4 as the initial condition. This is the same set of initial conditions used by Phillips (1959), and is used in the suite of seven test cases for shallow water models proposed by Williamson (1992) and discussed in Chapter 3. The model was run with a variety of different time stepping schemes and varying amounts of viscosity. Each model run was 16 simulated days, with $N = 20$. Kurihara reported generally good results. The Rossby-Haurwitz wave consists of a wave number 4 pattern in the zonal direction. In a nondivergent barotropic model, this shape should move from west to east, without distortion. However, in several of Kurihara's runs the shape degenerated to higher wave numbers.

1.4.5 Spherical Geodesic Grids

Williamson (1968) and Sadourny (1968) simultaneously introduced a new approach to more homogeneously discretize the sphere. These grids are constructed from spherical triangles which are nearly equal in area and equilateral. Since the grid points are not regularly spaced and do not lie of orthogonal rows and columns, alternative finite difference schemes are used to discretize the equations. The grid was inspired by Buckminster Fuller's geodesic dome, and is called a spherical geodesic grid. Initial tests using the grid proved successful, and further studies were carried out. These were reported by Sadourny (1969), Williamson (1970), and Masuda (1986).

The grids are constructed from an icosahedron (20 faces and 12 vertices), one of the five Platonic solids. A conceptually simple scheme for constructing a spherical geodesic grid is to divide the edges of the icosahedral faces into equal lengths, create new smaller equilateral triangles in the plane, and then project onto the sphere. See Fig. 1.7. One can construct a more homogeneous grid by partitioning the spherical equilateral triangles instead. Williamson (1968) and Sadourny (1968) use slightly different techniques to construct their grids. However, both begin by partitioning the spherical icosahedral triangle.

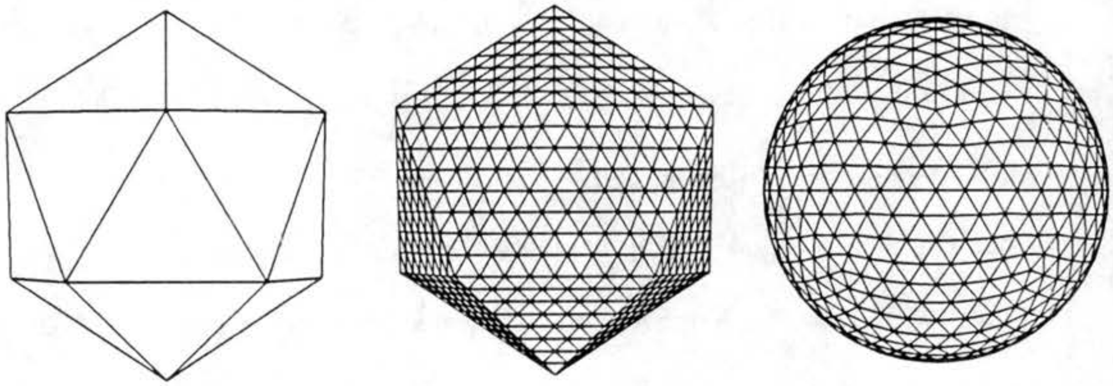

FIGURE 1.7: a.) Icosahedron. b.) Partition each face into 64 smaller triangles. c.) Project onto the sphere.

1.4.5.1 A Nondivergent Model

Williamson (1968) chose the nondivergent shallow water equations to test the new grid. By eliminating the divergence terms in (1.45) we get

$$\frac{\partial\zeta}{\partial t} = -J(\psi,\eta), \tag{1.77}$$

where ζ is relative vorticity, $\eta = \zeta + f$ is absolute vorticity and ψ is the stream function which can be determined from

$$\nabla^2\psi = \zeta. \tag{1.78}$$

The Jacobian operator, J, is defined in Appendix A. For arbitrary functions α and β, the Jacobian in differential form has the property that

$$\int_A J(\alpha,\beta)\,dA = \oint_S \alpha\frac{\partial\beta}{\partial s}ds. \tag{1.79}$$

So, integrating (1.77) over the area A we get

$$\frac{\partial}{\partial t}\int_A \zeta dA = -\oint_S u_n\eta ds, \tag{1.80}$$

where $u_n = -\frac{\partial\psi}{\partial s}$ is the velocity normal to the boundary S. See equation (1.45). Consider K triangles surrounding the grid point P_0 in Fig. 1.8. We approximate the line integral along the polygon defined by the path $P_1, P_2, \ldots, P_5, P_1$. Let ζ_0 be the relative

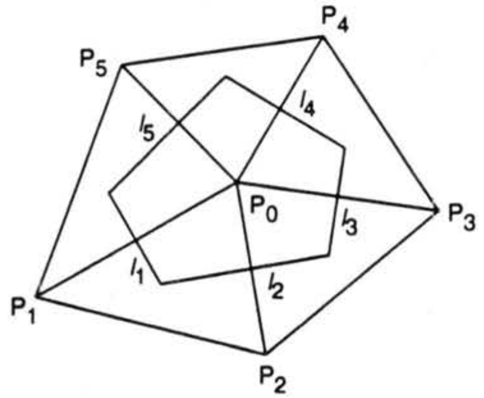

FIGURE 1.8: Configuration of grid triangles for the case $K = 5$.

vorticity defined at the point P_0. First, we make the approximation $A\zeta_0 = \int_A \zeta dA$. Let η_i and η_{i+1} be the absolute vorticity defined at points P_i and P_{i+1}, respectively. We approximate the absolute vorticity along the edge between P_i and P_{i+1} with $(\eta_i + \eta_{i+1})/2$. Similarly, $\partial\psi/\partial s \approx (\psi_{i+1} - \psi_i)/\Delta s$, where Δs is the distance from P_i to P_{i+1}. Hence, we can approximate (1.80) by

$$\frac{\partial\zeta_0}{\partial t} = \frac{1}{A}\sum_{i=1}^{K} \left(\frac{\psi_{i+1} - \psi_i}{\Delta s}\right) \left(\frac{\eta_{i+1} + \eta_i}{2}\right) \Delta s. \tag{1.81}$$

Next, we solve equation (1.78). The equation is solved via relaxation, so we must discretize the Poisson operator. Consider the smaller, inner polygon in Fig. 1.8. The walls of the polygon are formed from the perpendicular bisectors of the line segments $\overline{P_0P_i}$. In should be noted that these smaller polygons surrounding each grid point are equivalent to the Voronoi cells discussed in Chapter 2. For any arbitrary scalar function α, we can use Gauss' Theorem to write

$$\int_a \nabla^2 \alpha da = \oint_{s'} \frac{\partial\alpha}{\partial n} ds, \tag{1.82}$$

where a is the area of the small polygon, and s' is its boundary. Using (1.78) and (1.82), we get

$$\Gamma = \oint_{s'} u_s ds, \tag{1.83}$$

where Γ is the circulation around the boundary and u_s is the counterclockwise tangential velocity. By analogy with earlier derivation, we approximate $\Gamma \approx a\zeta_0$. We assume the tangential velocity $u_s = -\partial\psi/\partial n$ is constant along each wall of s', and can be approximated by $(\psi_i - \psi_0)/|P_0P_i|$, where $|P_0P_i|$ is the distance from P_0 to P_i. Define $\omega_i \equiv l_i/|P_0P_i|$, where l_i is the length of wall i. The finite-difference approximation for (1.83) is

$$\zeta_0 = \sum_{i=1}^{K} \omega_i (\psi_i - \psi_0) . \tag{1.84}$$

Let superscript m denote the iteration number of the relaxation. Then $\psi_i^{m+1} = \psi_i^m + \alpha R^m$, where α is a relaxation parameter. R^m is determined by solving (1.84) for ψ_0:

$$R^m = \frac{1}{\sum_{i=1}^{K} \omega_i} \left[\sum_{i=1}^{K} \omega_i (\psi_i^j - \psi_0^m) \zeta_0 \right] . \tag{1.85}$$

The superscript j is either m or $m+1$ depending on whether or not the neighboring grid point has been updated on the current sweep.

Williamson claims that the scheme conserves kinetic energy and the square of the vorticity as the exact equations do. When applied to an equilateral triangular grid on a plane, the scheme is second order accurate. Williamson uses an Euler forward time step initially, followed by leap-frog scheme for the following time steps.

He performed a numerical experiment with this model, using a Rossby-Haurwitz wave as his initial condition:

$$\psi(\theta, \lambda) = -279.68 \sin\theta + 136.65 \sin(6\lambda - \nu t) \sin\theta \cos^6\theta , \tag{1.86}$$

with $t = 0$. Equation (1.86) is an analytic solution of the nondivergent barotropic equation, and so is useful for evaluating the performance of the scheme. This is the same initial condition used in test case 6 discussed in Chapter 3. Williamson experimented with different relaxation parameters and grid orientations. He also did a 12-simulated-day run using (1.86) as the initial condition, and reported good results. Williamson (1971) describes a finite difference scheme that is second-order accurate on the spherical geodesic grid, but gives up certain conservation properties.

1.4.5.2 A Divergent Model

Sadourny (1968) discusses a nondivergent model very similar to Williamson's. Also, Sadourny (1969) develops a model for the divergent barotropic equations, or shallow water equations on a plane. For better or worse, Sadourny introduces a new grid indexing scheme to discretize the plane. Let $\alpha = e^{i\pi/3}$. Consider the triangular grid of points in Fig. 1.9.a Arbitrarily labeling one point in the plane 0, any grid point index is a linear combination, with integer coefficients, of the indices 1, α, α^2, α^3, α^4 and α^5. So, for

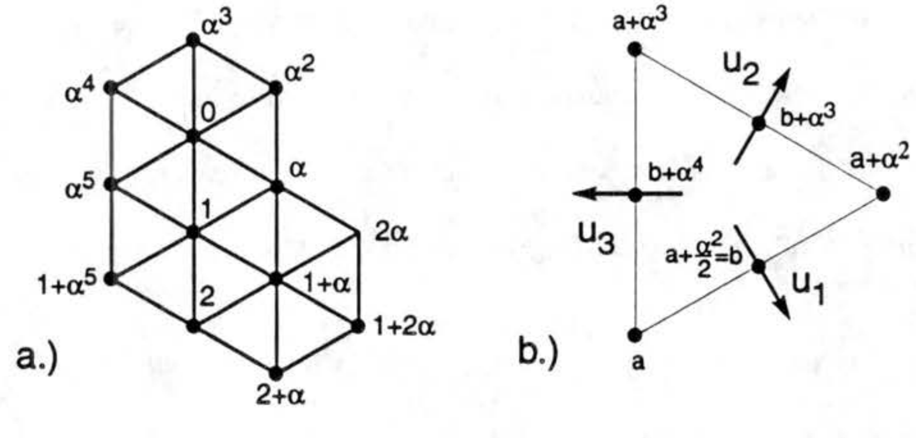

FIGURE 1.9: a.) Indexing scheme for grid points. b.) Location and direction of wind components

example, for any point a, the six neighboring points are $a+\alpha$, $a+\alpha^2$, $a+\alpha^3$, $a+\alpha^5$, $a+\alpha^5$ and $a+1$.

Let $a+\frac{\alpha^i}{2}$ denote the midpoint of the segment between a and $a+\alpha^i$. The midpoints of the segments are used to construct a second, staggered grid. An example with $i = 2$ is shown in Fig. 1.9.b. The wind will be defined on the staggered grid. The wind vector has three dependent components in the directions perpendicular to edges of the triangular grid. Thus, $\boldsymbol{v} = (u_1, u_2, u_3)$.

Consider first the nondivergent case, for which h^* is constant. The shallow water momentum equation in flux form is given by

$$\frac{\partial}{\partial t}h^* \boldsymbol{v} + \nabla\bullet(\boldsymbol{v}h^*\boldsymbol{v}) = -f\hat{\boldsymbol{k}}\times h^*\boldsymbol{v} - \frac{1}{2}\nabla\,(gh^{*2}), \tag{1.87}$$

assuming there is no topography so that $h^* = h$. The continuity equation is

$$\frac{\partial}{\partial t}h^* + \nabla\bullet(h^*\boldsymbol{v}) = 0. \tag{1.88}$$

For the nondivergent case, ignoring the Coriolis terms, (1.87) becomes

$$\frac{\partial}{\partial t}(h^* u) = -\nabla\bullet(uh^*\boldsymbol{v}), \tag{1.89}$$

where u is the velocity component in one of the three coordinate directions. We can define a stream function ψ on the triangular grid, such that $\nabla\psi = \hat{\boldsymbol{k}}\times h^*\boldsymbol{v}$. With this definition of ψ, (1.88) can be written in terms of the Jacobian operator as

$$\frac{\partial}{\partial t}(h^* u) = J(u,\psi), \tag{1.90}$$

where u is one of the three velocity components. By analogy with Williamson's model, we can discretize the Jacobian operator on the staggered grid as follows:

$$\frac{\partial}{\partial t}(h^* u) = \frac{1}{2s}\sum_j [u(b) + u(b+\alpha^i)]\,[\psi(b+\alpha^{i+1}) - \psi(b+\alpha^{i-1})], \tag{1.91}$$

where s is the area of the smaller hexagon formed by dashed lines in Fig. 1.10.a. The tangential derivatives are approximated along the dashed lines in Fig. 1.10.a. This differs from Williamson's model in which they are approximated along the solid lines. Since ψ is defined on the triangular grid, it will have to be interpolated to the staggered grid.

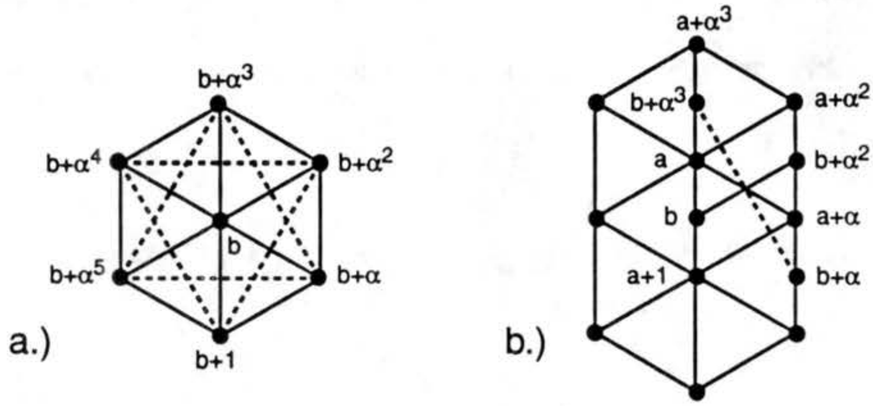

FIGURE 1.10: a.) Jacobian at grid point b. b.) Detail of grid for mass flux in momentum advection.

Moving now to the divergent case, we can write the mass flux crossing a segment of the triangular grid as

$$F(a, a+\alpha^i) = dh^*\left(a+\frac{\alpha^i}{2}\right)u\left(a+\frac{\alpha^i}{2}\right). \tag{1.92}$$

where h^* is the variable depth of the fluid and d is the distance from a to $a+\alpha^i$. u is the component of the flow perpendicular to the segment $\overline{a, a+\alpha^i}$. With this definition, we can replace the advection terms involving the stream function in the momentum equation (1.91) with terms involving the mass flux:

$$\frac{\partial}{\partial t}(h^* u)(b) = \frac{1}{2s}\sum_j [u(b) + u(b+\alpha^j)]\, F(b+\alpha^{j+1}, b+\alpha^{j-1}). \tag{1.93}$$

For example, the case when $j = 2$ and $b = a + \frac{1}{2}$ is shown in Fig. 1.10.b. The mass fluxes are only defined normally across triangular sides. The interpolation formula to give the mass flux across the dashed line in Fig. 1.10.b is

$$\begin{aligned} F(b+\alpha^{j+1}, b+\alpha^{j-1}) &= \frac{1}{4}[F(a+\alpha^{j-1}, a) + F(a, a+\alpha^{j+1}) \\ &+ F(a+\alpha^{j-1}, a+\alpha^j) + F(a+\alpha^j, a+\alpha^{j+1}) + F(a+\alpha^i+\alpha^{j-1}, a+\alpha^i) \\ &+ F(a+\alpha^i, a+\alpha^i+\alpha^{j+1}) + F(a+\alpha^i+\alpha^{j-1}, a+\alpha^i+\alpha^j) \\ &+ F(a+\alpha^i+\alpha^j, a+\alpha^i+\alpha^{j+1})\,]. \end{aligned} \tag{1.94}$$

The principle of total energy conservation is used to derive finite difference expressions for the pressure gradient terms of equation (1.87) on the staggered grid, and for the conservation of mass equation on the triangular grid. The pressure gradient terms are given by

$$\frac{3d}{2s} g h^*\left(a + \frac{\alpha^j + \alpha^{j+1}}{2}\right)[h^*(a+\alpha^j+\alpha^{j+1}) - h^*(a)]. \tag{1.95}$$

The mass conservation equation is given by

$$\frac{\partial}{\partial t} h^*(a) = \frac{1}{s}\sum_j F(a+\alpha^j, a+\alpha^{j+1}). \tag{1.96}$$

Sadourny used the model to study multiple reflections of periodic gravity waves in a channel. Since the numerical results do not involve a spherical domain or the effects of rotation, we will not discuss them here.

1.4.5.3 Masuda's Model: Numerical Results

Masuda (1986) derives an elegant model for the shallow water equations on the sphere. The derivation of the model is discussed in detail in Chapter 2. Here, we present, briefly, some of Masuda's numerical results.

Masuda also chooses the Rossby-Haurwitz wave with wavenumber 4 as his initial condition. The initial condition is discussed further in Chapter 3. Fig. 1.11 shows the evolution of the velocity potential field in a 96-simulated-day run using Masuda's model. The initial conditions are nondivergent, so initially the velocity potential is zero. As time progresses, a wavenumber 4 pattern develops. As time progresses further, the pattern at higher latitudes begins to break down, forming a wavenumber 1 pattern. Significantly, the wavenumber 1 pattern is antisymmetric across the equator, even though the initial

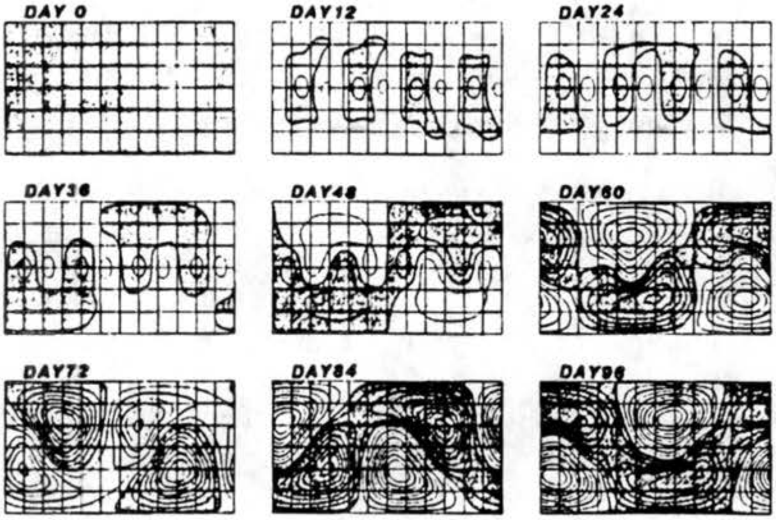

FIGURE 1.11: Masuda's velocity potential field.

condition is symmetric across the equator. Masuda believes this is due to the antisymmetry of the grid across the equator. This is discussed further in Chapter 3.

1.4.6 The Wandering Electron Grid

One approach to constructing a mesh of grid points that homogeneously covers a sphere is to model the distribution of a set of electrons on a sphere. Suppose a set of electrons is confined to the surface of a sphere. Since a given electron is repelled by every other electron, it will move to maximize the distance between it and its closest neighbor while remaining stuck to the surface. In this way, the electrons will distribute themselves evenly upon the sphere. We associate a grid point with each electron, and we associate a Voronoi cell with each grid point. Voronoi cell is defined in Section 2.1.2 of Chapter 2. The development of the antisymmetry in Masuda's models suggests that it would be advantageous to construct the grid so that it is symmetric across the equator. An equator can be defined by restricting the movement of a subset of the electrons to a great circle. The remaining electrons can be paired so that each has a mirror image in the

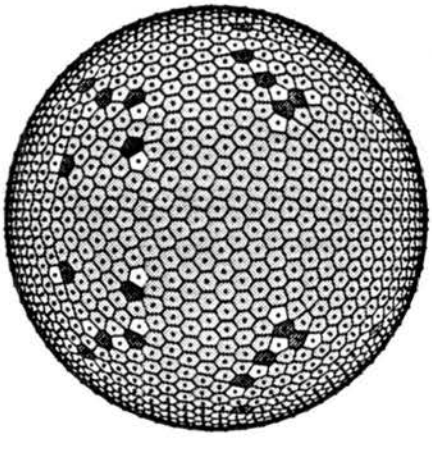

FIGURE 1.12: Wandering electron grid. White cells have five walls, light gray cells have six walls, and dark gray cells have seven walls.

opposite hemisphere. We also fix an electron at each of the poles and one electron along the equator. Without fixing the positions of some of the electrons, the overall pattern tends to wander indefinitely. Fig. 1.12 shows a grid constructed using the wandering electron algorithm. Most cells have six walls, the rest have five or seven walls. It is impossible for the Voronoi cells to all have six walls as discussed in Section 2.1.3 of Chapter 2. While this approach more or less homogeneously covers the sphere, it is impractical from a coding point of view.

CHAPTER 2

The Model

2.1 The Twisted Icosahedral Grid

2.1.1 Algorithm for Twisted Icosahedral Polyhedron

Masuda's results suggest that it is desirable to construct a spherical geodesic grid which is symmetric across the equator. In this way, initial conditions which are symmetric will remain symmetric during the integration. Such a grid can be constructed by starting with an ordinary icosahedron inscribed in a unit sphere as shown in Fig. 2.1.a. First, each face of the icosahedron is subdivided into four new faces by bisecting the edges. The result of this process is shown in Fig. 2.1.b. Next, the new vertices are projected onto the unit sphere, creating the polyhedron in shown in Fig. 2.1.c. The resulting polyhedron is not symmetric across the equator; however, by simply rotating all the faces in the southern hemisphere through $\pi/5$ radians the polyhedron can be made symmetric as shown in Fig. 2.1.d. This polyhedron is called the *twisted icosahedron*. By recursively subdividing each triangular face and projecting the new vertices onto the unit sphere, we can generate polyhedrons that increasingly approximate a sphere. Examples are shown in Fig. 2.2. These polyhedrons, whose faces are all triangles, are called generating polyhedrons. Note that six edges emanate from every vertex except for the 12

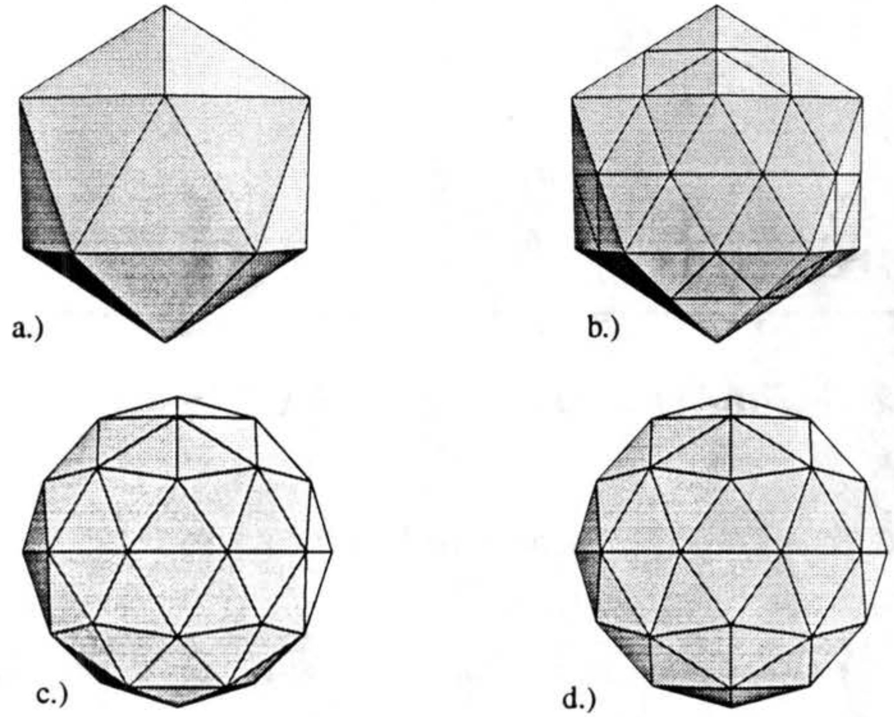

FIGURE 2.1: a.) An icosahedron inscribed in unit sphere. b.) Bisect each edge forming from each four new faces c.) Project new vertices onto unit sphere. d.) Rotate faces in southern hemisphere to form twisted icosahedron polyhedron.

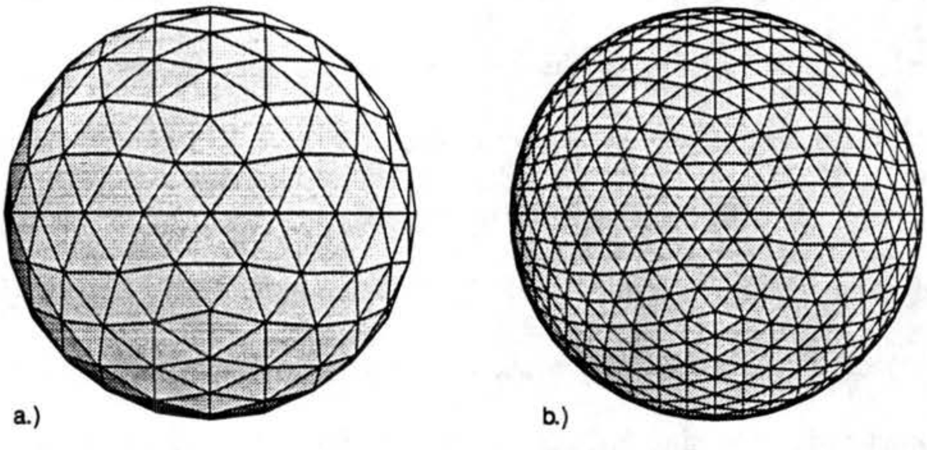

FIGURE 2.2: a.) 320 faces and 128 vertices. b.) 1280 faces and 642 vertices.

vertices on the original icosahedron. For example, Fig. 2.1.a shows that all vertices of a icosahedron are formed from five edges. The vertex that is visible in the upper-center part of Fig. 2.1.a persists in Fig.2.1.d and in figures Fig.2.2.a and Fig.2.2.b.

2.1.2 Voronoi Grid on the Sphere

We can associate with each vertex of the generating polyhedron a grid point, and with each grid point a grid cell. For numerical reasons discussed later, the grid cells are Voronoi cells, defined below. Augenbaum (1985) discusses the construction of a Voronoi grid on a sphere.

Given a set of N grid points $\{P_1, P_2, \ldots, P_N\}$ on the unit sphere S, the Voronoi cell_k associated with P_k is defined by

$$\text{cell}_k = \{ |p - P_k| \le |p - P_l| \ \ \forall \ p \in S \ \text{ and } \ l \in \{1, 2, \ldots, k-1, k+1, \ldots, N\} \}, \quad \textbf{(2.1)}$$

where $|x_1 - x_2|$ is the distance between points $x_1, x_2 \in S$ measured along the surface of the sphere. In other words, the Voronoi cell of a grid point consists of all the points on the sphere equidistant or closer to that grid point than any other grid point. Let P_0 and P_1 be neighboring grid points as shown in Fig 2.3. As a result of the above definition, a Voronoi cell has the important property that the cell wall shared by P_0 and P_1 is the perpendicular bisector of the circular arc between the two grid points. A cell wall is a segment of a great circle. The end points of a wall are the Voronoi corners, the points

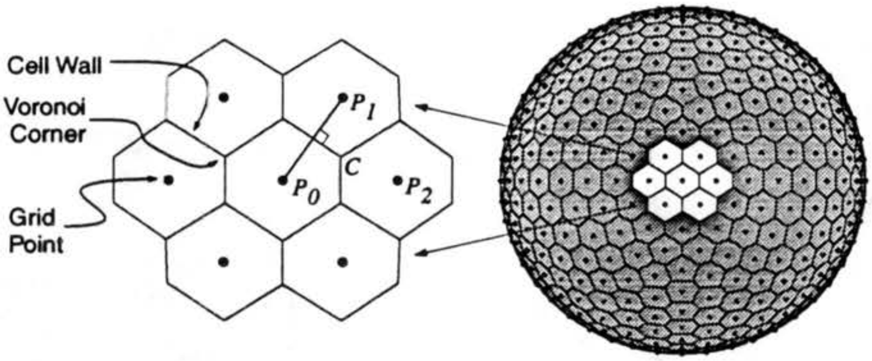

FIGURE 2.3: Voronoi cells.

where three cell walls intersect. The position of a corner is determined by three grid points. Again, consider P_0, P_1 and P_2 in Fig. 2.3. Consider these points to be in a three-dimensional space. From the definition of a Voronoi cell, the Voronoi corner C is the point on the sphere equidistant from these three grid points:

$$C = \frac{(P_2 - P_0) \times (P_1 - P_0)}{|(P_2 - P_0) \times (P_1 - P_0)|}, \tag{2.2}$$

where $(\ ) - (\ )$ denotes a vector difference, $(\ ) \times (\ )$ denotes the vector cross product, and $|(\ )|$ denotes the magnitude of a vector. Note that the points are ordered in a clockwise fashion, to insure that C lies in the same hemisphere as P_0, P_1 and P_2. Ordering in a counterclockwise fashion gives the point on the opposite side of the sphere from C.

2.1.3 Properties of the Twisted Icosahedral Grid

We can now construct a Voronoi grid using the vertices of the generating polyhedrons like those shown in Fig. 2.2 as cell centers. Examples are shown below in Fig. 2.4. The

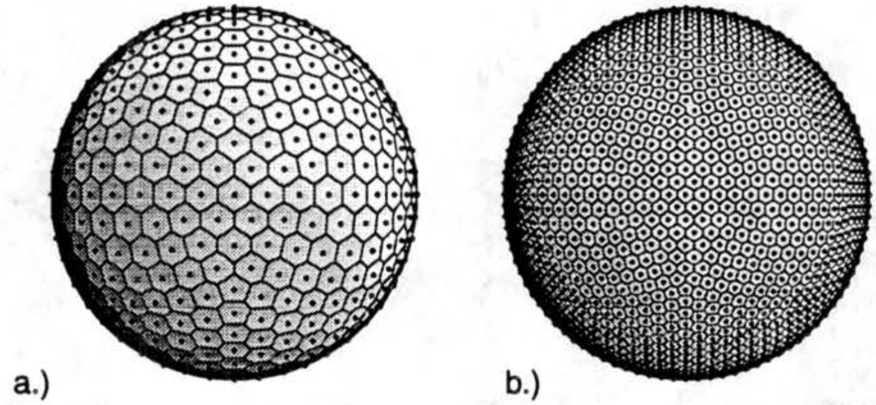

FIGURE 2.4: Voronoi grids constructed on twisted icosahedron generating polyhedron. a.) 642 cells. b.) 2562 cells.

numerical results described later were run on the grids with 2562 cells and 10242 cells. Table 2.1 shows the total number of cells, and number of cell along the equator for each level recursive subdivisions of the generating polyhedron.

Table 2.1: Properties of grid as a function of recursive subdivisions of generating polyhedron.

R	Number of cells	Number of cells along equator	Average cell area in km^2	Ratio of smallest cell to largest cell	Average distance between cell centers in km	Ratio of smallest distance to largest distance between cell centers
0	42	10	1.214×10^7	0.885	3755.5	0.881
1	162	20	3.149×10^6	0.774	1912.8	0.848
2	642	40	7.946×10^5	0.763	961.0	0.840
3	2562	80	1.991×10^5	0.742	481.1	0.838
4	10242	160	4.980×10^4	0.736	240.6	0.837
5	40962	320	1.245×10^4	0.733	120.3	0.837

The number of cells, N_c, is related to the number of recursive subdivisions, R, by

$$N_c = 5 * 2^{2R+3} + 2. \tag{2.3}$$

The proof of this relation is straight forward. The number of cells in a Voronoi grid is equal to the number of vertices in the generating polyhedron from which the grid is constructed. We can easily count the number of faces, f_R, in a generating polyhedron. In this case, the original icosahedron has 20 faces. Each of these was subdivided into four new faces to form Fig. 2.1.d, which has 60 faces. In general, we have $f_R = 20 * 4^{R+1}$. Considered separately, each triangular face has three vertices, for a total of $3 * f_R$

vertices. When the faces are fitted together to form the generating polyhedron, all but 12 vertices in the generating polyhedron are formed from six triangular faces. At these vertices, 6 triangle vertices overlap to form one vertex. The remaining 12 vertices of the generating polyhedron are formed by the overlap of 5 triangle vertices. That is, $3*f_R = 6*(N_c - 12) + 5*12$. Substituting $f_R = 20*4^{R+1}$ gives (2.3).

The twisted-icosahedral grids in many ways resemble the new family of carbon molecules known as fullerenes, and the mathematics used to study fullerenes can be used to deduce some interesting properties of the twisted icosahedral grids. Without discussing fullerenes in detail, we can at least describe the simplest, C_{60}, also known as a buckyball, by considering a soccer ball. A soccer ball, or truncated icosahedron, has twelve pentagons, each of which is surrounded by five hexagons as shown in Fig. 2.5. A C_{60} molecule is modeled by placing a carbon atom at each vertex. Fullerenes are discussed further by Chung and Sternberg (1993).

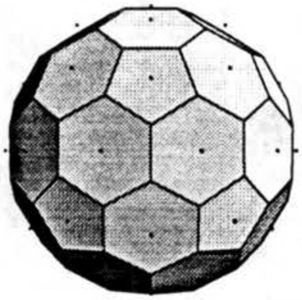

FIGURE 2.5: Truncated icosahedron. 12 pentagonal faces and 20 hexagonal faces. 60 vertices and 90 edges.

The fullerenes and the twisted-icosahedral grids have an important property in common. That is, they are both examples of polyhedra in which three edges emanate from each vertex. Such polyhedrons are called *trivalent*. Euler proved that all trivalent polyhedrons satisfy the relation

$$\sum (6-n) f_n = 12, \tag{2.4}$$

where f_n is the number of faces with n sides. The summation is over the number of different polygons comprising the polyhedron. In our case, the list of polygons includes pentagons and hexagons. Clearly, all hexagons, $n = 6$, will contribute nothing to the summation. This implies that one cannot construct a grid entirely of hexagons. In addition, Euler's formula says that a grid consisting solely of pentagons and hexagons must have exactly 12 pentagons. This formula does not indicate how many hexagons are allowed, but it can be shown that a grid can be built with almost any number of hexagons. The only number of hexagons not allowed is one.

2.2 Finite Difference Approximation to the Shallow Water Equations

Most of this section is based on Masuda (1985). Using the twisted-icosahedral grid, we construct a model based on the SFVP form of the shallow water equations. As mentioned earlier, this form has the big advantage that all the fields are expressed as true scalars. This avoids the difficulties of vector valued quantities near the poles. To reiterate, the time rate of change of absolute vorticity, divergence and geopotential are given by

$$\frac{\partial \eta}{\partial t} = -\nabla \bullet (\eta \nabla \chi) + J(\eta, \psi), \tag{2.5}$$

$$\frac{\partial\delta}{\partial t} = \nabla\bullet(\eta\nabla\psi) + J(\eta,\chi) - \nabla^2(K+\phi), \tag{2.6}$$

$$\frac{\partial}{\partial t}\phi^* = -\nabla\bullet(\phi^*\nabla\chi) + J(\phi^*,\psi). \tag{2.7}$$

Here $\phi = gh$, and $\phi^* = gh^*$. Stream function and velocity potential are defined by $\nabla^2\psi = \zeta = \eta - f$ and $\nabla^2\chi = \delta$. K is the kinetic energy:

$$K = \frac{1}{2}\left[\nabla\bullet(\psi\nabla\psi) - \psi\nabla^2\psi + \nabla\bullet(\chi\nabla\chi) - \chi\nabla^2\chi\right] + J(\psi,\chi). \tag{2.8}$$

We next show quantities conserved by the continuous equations, and formulate the finite difference equations so that they approximately conserve these quantities. Define potential vorticity q, total energy E, and potential enstrophy Z as follows

$$q = \frac{\eta}{\phi},\quad E = \frac{1}{2}\phi^2 + \phi K \text{ and, } Z = \frac{1}{2}\phi q^2. \tag{2.9}$$

With these quantities, we can write prognostic equations for total energy and potential enstrophy as follows

$$\frac{\partial E}{\partial t} = -\nabla\bullet(E\nabla\chi) + J(E,\psi), \text{ and } \frac{\partial Z}{\partial t} = -\nabla\bullet(Z\nabla\chi) + J(Z,\psi). \tag{2.10}$$

Consider arbitrary scalars α and β. Using Gauss' theorem we can write

$$\int_A J(\alpha,\beta)\,dA = \oint_C \alpha\frac{\partial\beta}{\partial s}ds, \tag{2.11}$$

$$\int_A \nabla\bullet(\alpha\nabla\beta)\,dA = \oint_C \alpha\frac{\partial\beta}{\partial n}ds, \tag{2.12}$$

$$\int_A \nabla^2\alpha dA = \oint_C \frac{\partial\alpha}{\partial n}ds. \tag{2.13}$$

Integration of the Jacobian, flux divergence and Laplacian operators over the surface of a sphere, Σ, satisfies

$$\int_{\Sigma} J(\alpha,\beta)\, dA = 0, \quad \int_{\Sigma} \nabla\bullet(\alpha\nabla\beta)\, dA = 0, \text{ and } \int_{\Sigma} \nabla^2 \alpha dA = 0. \tag{2.14}$$

Hence, integrating equations (2.7) and (2.10) over Σ shows that mass, total energy and potential enstrophy are conserved by the continuous equations. Also, since the global integrals of vorticity and divergence are initially zero, (2.5) and (2.6) imply the global integral of these quantities will remain zero for all time. We can construct the finite difference formule to approximately conserve these quantities as well.

The finite-difference approach we use approximates the line integrals on the right-hand sides of equations (2.11), (2.12) and (2.13). Consider a typical cell as shown in Fig. 2.6. Let α_i and β_i be the values of functions α and β evaluated at the cell centers P_i. Let

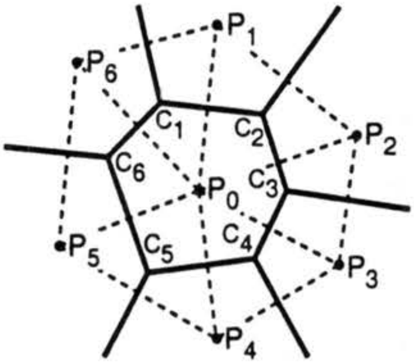

FIGURE 2.6: Typical hexagonal cell.

$$b_i = (\beta_0 + \beta_{i-1} + \beta_i)/3 \tag{2.15}$$

be an approximation of β defined at the cell corners C_i. This is a good approximation since the Voronoi grid ensures that the corner is equidistant from the three cell centers. The approximation is improved as the hexagons become more perfect. Let A_c be the area of the cell. If the cell is small, $J(\alpha, \beta)$ can be considered constant across the cell, and we can approximate $\int_A J(\alpha, \beta)\, dA \approx J(\alpha, \beta)|_{P_0} A_c$ where $J(\alpha, \beta)|_{P_0}$ is the Jacobian evaluated at P_0 Using (2.11) we can we approximate the Jacobian by

$$J(\alpha, \beta)|_{P_0} \approx \frac{1}{A_c} \sum_{i=1}^{N} (\frac{\alpha_0 + \alpha_i}{2}) \left(\frac{b_{i+1} - b_i}{l_i}\right) l_i, \tag{2.16}$$

where $N = 5$ for pentagons and $N = 6$ for hexagons. In (2.16), the expression

$$(\alpha_0 + \alpha_i) / 2 \tag{2.17}$$

approximates α evaluated along the cell wall. This is reasonable since each wall bisects the arc between cell centers. Let l_i be the length of the arc between corners C_i and C_{i+1}. The expression

$$(b_{i+1} - b_i) / l_i \tag{2.18}$$

approximates the tangential derivative along the cell wall. If the grid is comprised of perfect hexagons, then the arc between cell centers is also the perpendicular bisector of the cell wall. In this case, the approximation of the tangential derivative is second order accurate at the same point where α is approximated along the cell wall. The final l_i approximates the infinitesimal element of curve ds. Equation (2.16) can be rewritten as

$$J(\alpha, \beta)|_{P_0} \approx \frac{1}{A_c} \sum_{i=1}^{N} (\alpha_0 + \alpha_i) (\beta_{i+1} - \beta_{i-1}) . \tag{2.19}$$

Similarly, we can approximate the flux divergence and the Laplacian operators. Let L_i be the length of the arc between cell centers P_0 and P_i. The flux divergence can be approximated by

$$\nabla \bullet (\alpha \nabla \beta)|_{P_0} \approx \frac{1}{A_c} \sum_{i=1}^{N} \frac{l_i}{L_i} \left(\frac{\alpha_0 + \alpha_i}{2}\right) (\beta_i - \beta_0) . \tag{2.20}$$

The expression

$$(\beta_i - \beta_0) / L_i \tag{2.21}$$

approximates the derivative normal to the curve. Finally, the Laplacian can be approximated with

$$\nabla^2 \alpha|_{P_0} \approx \frac{1}{A_c} \sum_{i=1}^{N} \frac{l_i}{L_i} (\alpha_i - \alpha_0) . \tag{2.22}$$

These difference operators will give the finite difference equations about the same conservation properties as the continuous equations. The difference operators approximate the line integrals with a summation. One summand of the summation of an operator at an arbitrary cell, will also occur in the cell's neighbor's summation, but with the opposite sign. Since the surface of the sphere is a closed domain, the summation over all the cells will exactly cancel. Hence, if time differentiation is left continuous, absolute vorticity, divergence and mass are conserved by spatially decretized operators.

An Euler forward time step used for the first time step, followed by leap-frog time steps for each successive time step. An Euler forward time step is performed every 100 time steps to eliminate the computational mode.

2.3 Multigrid Solver

One computational cycle of the model can be divided into two steps. First, the model time steps the prognostic equations for new values of absolute vorticity, divergence and geopotential. These fields are then used to solve Poisson's equation for stream function and velocity potential, that is, $\nabla^2\psi = \eta - f$ and $\nabla^2\chi = \delta$, and the cycle repeats. Efficient numeric implementation of this cycle hinges on being able to quickly solve the elliptic equations. The recursive nature of the grid lends itself well to multigrid methods.

2.3.1 Motivation and review of multigrid methods in 2D

This section is based on Fulton (1986) and Brandt (1977). Consider Poisson's equation with Dirichlet boundary conditions:

$$\begin{aligned} \nabla^2 U(x,y) &= f(x,y) \quad \text{in} \quad \Omega, \\ U(x,y) &= g(x,y) \quad \text{on} \quad \partial\Omega, \end{aligned} \tag{2.23}$$

where Ω is the interior and $\partial\Omega$ the boundary of the unit square $[0,1]\times[0,1]$. Let $N \in \{2^M | M \in \text{positive integer}\}$, and let $h = 1/N$. We can discretize the domain as follows:

$$\Omega_h = \{ (x_j, y_k) = (jh, kh) | 0 \le j, k \le N \}. \tag{2.24}$$

The continuous problem can be approximated with second-order finite difference operators:

$$L^h U^h_{j,k} \equiv \frac{U^h_{j+1,k} - 2U^h_{j,k} + U^h_{j-1,k}}{h^2} + \frac{U^h_{j,k+1} - 2U^h_{j,k} + U^h_{j,k-1}}{h^2} = f^h_{j,k} \ \forall \ 0 < i, j < N$$

$$U^h_{j,k} = g^h_{j,k} \ \forall \ i = 0, i = N, j = 0, \text{or} \ j = N, \tag{2.25}$$

where $f^h_{j,k} = f(x_j, y_k)$ and $g^h_{j,k} = g(x_j, y_k)$. $U^h = \{U^h_{j,k} | 0 < j, k < N\}$ is the solution of the linear system of equations defined by (2.25).

It is this linear system that we wish to solve. Considerable effort has gone into the study of this system and many schemes have been devised to solve it including a family of schemes called the point iterative methods. These methods generate a sequence of approximate solutions which converges to the true solution U^h. So, if $u = \{u_{j,k} | 0 < j, k < N\}$ is an element of this sequence, then we can define the error $v^h = \{v^h_{j,k} = U^h_{j,k} - u^h_{j,k} | 0 < j, k < N\}$. Clearly, if we knew v^h the problem would be solved. One of the point iterative schemes is the Gauss-Seidel algorithm. The scheme can best be described by the computer code to perform it. This is shown in Fig. 2.7. Each execution of this block of code generates an element in the sequence of approximation solutions to (2.25). Updating `u(J,K)` requires four pieces of information, two from the current solution, namely `u(J-1,K)` and `u(J,K-1)`, and two from the previously

```
    DO K=1,N-1
       DO J=1,N-1
          u(J,K) = 0.25*(u(J+1,K)+u(J-1,K)+u(J,K+1)+u(J,K-1)
 $                   -h*h*f(J,K))
       ENDDO
    ENDDO
```

FIGURE 2.7: FORTRAN code to perform one iteration of Gauss-Seidel.

computed solution, `u(J+1,K)` and `u(J,K+1)`. If we denote $\bar{u}^h_{j,k}$ as a value from the current solution and $u^h_{j,k}$ as a value from the previous solution, then as we sweep through $0 < j, k < N$ each $\bar{u}_{j,k}$ satisfies

$$\frac{u^h_{j+1,k} - 2\bar{u}^h_{j,k} + \bar{u}^h_{j-1,k}}{h^2} + \frac{u^h_{j,k+1} - 2\bar{u}^h_{j,k} + \bar{u}^h_{j,k-1}}{h^2} = f^h_{j,k}. \qquad \textbf{(2.26)}$$

Subtracting (2.26) from (2.25) gives

$$\bar{v}_{j,k} = \frac{1}{4}\left(v_{j+1,k} + \bar{v}_{j-1,k} + v_{j,k+1} + \bar{v}_{j,k-1}\right), \qquad \textbf{(2.27)}$$

where $\bar{v}^h_{j,k} = U^h_{j,k} - \bar{u}^h_{j,k}$.

Following Fulton and Brandt, we can expand v^h into a discrete Fourier series expanded about some arbitrary point grid point (x_j, y_k). The discrete Fourier modes are given by

$$E_\theta(x_j, y_k) = e^{i(j\theta_1 + k\theta_2)}. \qquad \textbf{(2.28)}$$

The discrete vector wavenumber

$$\theta = (\theta_1, \theta_2) \in \{(2\pi h\alpha, 2\pi h\beta) \mid -N/2 \le \alpha, \beta \le N/2 \text{ and } \alpha, \beta \in \text{integer}\} \qquad \textbf{(2.29)}$$

ensures modes which can be resolved by the grid. Suppose the Fourier expansion of the error v^h has a mode $A_\theta E_\theta$ where A_θ is the amplitude. The error associated with the next approximate solution will have a mode $\bar{A}_\theta E_\theta$. Substituting (2.28) into (2.27) gives

$$\bar{A}_\theta = \frac{1}{4}\left(e^{-i\theta_1}\bar{A}_\theta + e^{i\theta_1}A_\theta + e^{-i\theta_2}\bar{A}_\theta + e^{i\theta_2}A_\theta\right). \qquad \textbf{(2.30)}$$

Equation (2.30) relates the amplitudes A_θ and $\bar{A}_\theta$ of mode E_θ in Fourier expansion about grid point (x_j, y_k). We can define the convergence factor as follows

$$\mu(\theta) = \left|\frac{\bar{A}_\theta}{A_\theta}\right| = \left|\frac{e^{i\theta_1} + e^{i\theta_2}}{4 - e^{-i\theta_1} - e^{-i\theta_2}}\right|. \tag{2.31}$$

Fig. 2.8 shows a plot of $\mu(\theta)$ as a function of θ_1 and θ_2. This figure makes an important point. Modes with wavelengths on the order of the grid spacing will be quickly damped with just a few iterations of Gauss-Seidel. On the other hand, modes with long wavelengths are damped slowly. Hence, the effect of Gauss-Seidel iteration is primarily to smooth the error, and secondly to reduce it. Initially, the sequence of approximate solutions will converge quickly, but once the error is smooth, continuing to perform to perform Gauss-Seidel iterations is inefficient. Again, if we knew v^h we would be done. Since v^h is smooth, its structure can be accurately represented on a coarser grid. So, we might as well solve for v^h on the coarser grid, since it requires less numerical work. The error on the coarse grid, call it v^H, can then be interpolated to the fine grid and used to update u^h. This is main idea behind multigrid methods.

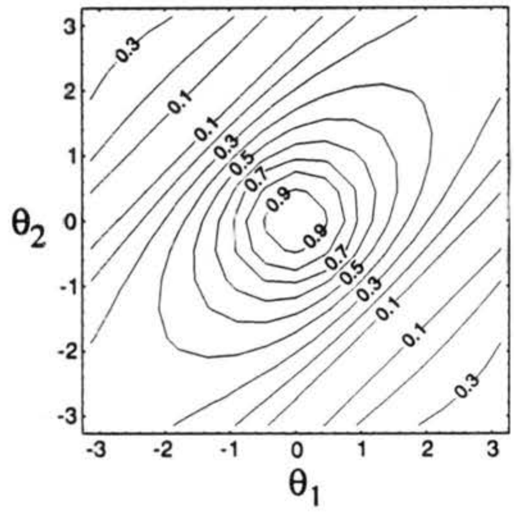

FIGURE 2.8: Convergence factor $\mu(\theta)$. Amplitude of small wavenumber modes damped slowly, while modes with grid scale wavelength damped quickly.

Fulton makes the above ideas concise with a two-level multigrid scheme. Define a coarser grid

$$\Omega_H = \{ (x_j, y_k) = (2jh, 2kh) \,|\, 0 \le j, k \le N/2 \} . \tag{2.32}$$

We can define the difference operator L^H analogous to L^h defined in (2.25) except with grid spacing $H = 2h$. So, if the residual is defined $r^h = f^h - L^h u^h$, then the linear system of equations for error on the fine grid is given by

$$L^h U^h = f^h, \tag{2.33}$$

$$L^h (v^h + u^h) = f^h, \tag{2.34}$$

$$L^h v^h = f^h - L^h u^h = r^h. \tag{2.35}$$

Let I_h^H denote the *injection* operator. This operator simply copies values from the fine grid to the coarse grid at grid points common to both grids. Similarly, let I_H^h denote the interpolation operator. This operator interpolates values from the course grid to the fine grid. The algorithm for a two-level multigrid cycle first applies a few iterations of Gauss-Seidel to the linear system

$$L^h u^h = f^h, \tag{2.36}$$

then approximates the error v^h with a few iterations of Gauss-Seidel on the linear system

$$L^H v^H = I_h^H r^h. \tag{2.37}$$

Finally, a new approximate solution u^h is created by adding the error correction to the old approximate solution:

$$u^h \leftarrow u^h + I_H^h v^H. \tag{2.38}$$

Of course there is no reason to limit the algorithm to just two grids. Trying to solve the linear problem $L^H v^H = r^H$ is the same as trying to solve $L^h u^h = f^h$, so we might as well apply the multigrid technique to $L^H v^H = r^H$. In general, we can use M grids or levels. Toward this end, let $1 \le l \le M$ and $h_l = 2^{-l}$. We define a set of grids

$$\Omega_{h_l} = \{ (h_l j, h_l k) \mid 0 < j, k < 2^l \}. \tag{2.39}$$

On each grid, define the linear system on a coarser grid l,

$$L^l u^l = f^l, \tag{2.40}$$

(the superscript now denotes level, not grid spacing) where the right-hand side uses information from the finer grid, $l + 1$:

$$f^l = I_{l+1}^l r^{l+1} = I_{l+1}^l (f^{l+1} - L^{l+1} u^{l+1}). \tag{2.41}$$

By analogy with equation (2.38), we transfer information from the coarse grid to the fine grid with

$$u^{l+1} \leftarrow u^{l+1} + I_l^{l+1} u^l. \tag{2.42}$$

2.3.2 Multigrid solver on twisted icosahedral grid

Now we import the ideas of the 2D multigrid method to the twisted icosahedral grid. First, the grid is not comprised of rectangular elements as is the 2D grid. So, interpolating from the coarse to the fine grid is a little more complicated. Fig. 2.9.a shows the relation between hexagonal cells for two different grid resolutions. The residual correction at cells common to both grids can be transferred from the coarse grid

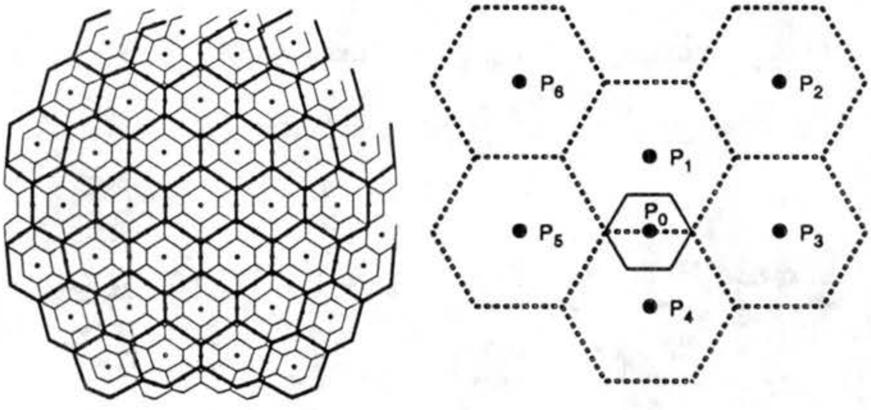

FIGURE 2.9: a.) Two grid resolutions. b.) Interpolation stencil. Values at grid points $P_1, P_2, \ldots, P_6$ are interpolated to get P_0.

to the fine grid. The remaining grid points of the fine grid all lie on a cell wall of the coarse grid. Fig. 2.9.b shows six cells on the coarse grid, their walls shown with dashed lines, and one cell on the fine grid, its walls shown with thin, solid lines. This stencil is used to associate every grid point on the fine grid with six grid points on the coarse grid. Each point has spherical coordinates $P_i = (\theta_i, \lambda_i)$ for $0 \leq i \leq 6$. Suppose we assign arbitrary values $f_1, f_2, \ldots, f_6$ at grid points $P_1, P_2, \ldots, P_6$. Then there exists a quadratic surface determined by constants $c_1, c_2, \ldots, c_6$ such that

$$\begin{gathered} c_1\theta_1^2 + c_2\theta_1\lambda_1 + c_3\lambda_1^2 + c_4\theta_1 + c_5\lambda_1 + c_6 = f_1 \\ c_1\theta_2^2 + c_2\theta_2\lambda_2 + c_3\lambda_2^2 + c_4\theta_2 + c_5\lambda_2 + c_6 = f_2 \\ \ldots \\ c_1\theta_6^2 + c_2\theta_6\lambda_6 + c_3\lambda_6^2 + c_4\theta_6 + c_5\lambda_6 + c_6 = f_6. \end{gathered} \qquad (2.43)$$

Solving the linear system, the constants have the form $c_i = c_i(\theta_1, \ldots, \theta_6, \lambda_1, \ldots, \lambda_6, f_1, \ldots, f_6)$ for $1 \leq i \leq 6$. At the fine grid point P_0 we can write

$$c_1\theta_0^2 + c_2\theta_0\lambda_0 + c_3\lambda_0^2 + c_4\theta_0 + c_5\lambda_0 + c_6 = f_0. \tag{2.44}$$

Substituting $c_1, c_2, \ldots, c_6$ and rearranging things gives

$$w_1 f_1 + w_2 f_2 + w_3 f_3 + w_4 f_4 + w_5 f_5 + w_6 f_6 = f_6, \tag{2.45}$$

where $w_i = w_i(\theta_1, \ldots, \theta_6, \lambda_1, \ldots, \lambda_6)$. A *Mathematica* program computes the weights $w_1, w_2, \ldots, w_6$ in advance and stores them to a file which is read by the model at the beginning of execution. *Mathematica* is software to symbolically perform mathematics. The software is described by Wolfram (1991).

Next, we need to be able to perform relaxation on the grid. Using (2.22) and referring to Fig. 2.6, we can translate the continuous problem $\nabla^2 U = f$ to difference form:

$$\frac{1}{A_c}\sum_{i=1}^{N}\frac{l_i}{L_i}(u_i - u_0) = f_0 \quad \forall \text{ cells}, \tag{2.46}$$

where f_0 is f evaluated at P_0. Solving (2.46) for u_0 we get

$$u_0 = \left[\sum_{i=1}^{N}\frac{l_i}{L_i}\right]^{-1}\left[\sum_{i=1}^{N}\frac{l_i u_i}{L_i} - A_c f_0\right]. \tag{2.47}$$

Performing (2.47) at every grid point is analogous to the block of FORTRAN code in Fig. 2.7.

The model uses a variation of the full multigrid cycle. A flow chart for the algorithm is shown in Fig. 2.10. Again, let M be the number of grid resolutions, so l must satisfy $1 \le l \le M$. $l = 1$ corresponds to the coarsest grid and $l = M$ corresponds to the finest grid.

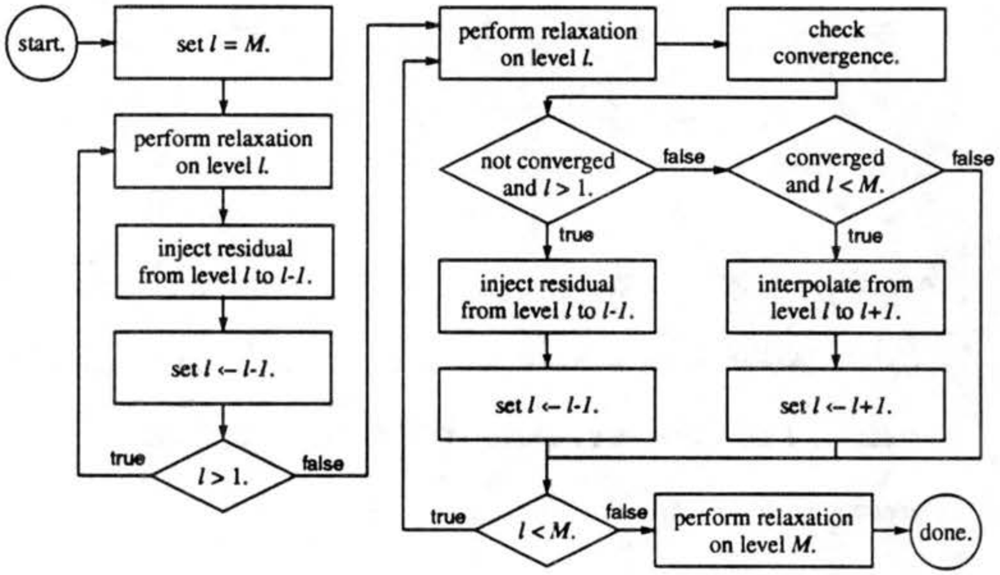

FIGURE 2.10: Flow chart for full multigrid cycle used in model.

The convergence criterion determines whether or not an error correction should be interpolated and added to the next finer grid. Denote an element of a sequence of error corrections by v^{save}. Suppose several relaxations are performed on v^{save} to produce a new element of the sequence which we denote by v. To compare these two approximations, the model computes the following number:

$$\varepsilon = \frac{\max\ \{ \left| v_i^{\text{save}} - v_i \right| \ \ \forall \ \text{cells} \}}{I\,(v^{\text{save}} - v)}, \tag{2.48}$$

where

$$I\,(v^{\text{save}} - v) = \left[\sum_i^{\text{all cells}} \text{area}_i \right]^{-1} \left[\sum_i^{\text{all cells}} \text{area}_i \left| v_i^{\text{save}} + v_i \right| \right], \tag{2.49}$$

and area_i is area of cell i. If ε is small, then the sequence of error corrections is converging, and the error correction at this resolution is smooth and should be interpolated and added to the next finer grid. On the other hand, if ε is large after several relaxation iterations, we should solve this problem on the next coarser grid.

There several weaknesses in the way the model performs the multigrid scheme that need improvement. The first involves the interpolation. Since the PDEs being solved are second-order, rule of thumb says linear interpolation should suffice. However, numerical experimentation indicates that quadratic interpolation works quite a bit better, and it is unclear why this is true. Also, the quadratic interpolation itself needs to be improved with a better interpolation scheme. Second, the model doesn't really use a Gauss-Seidel type relaxation scheme; instead, it uses Jacobi type iterations. The cells can be stepped through in, more-or-less, lexicographical order as explained section (2.4), below. There are no lateral boundary conditions on the surface of a sphere, so a Gauss-Seidel sweep begins in the same place it ends. Since Gauss-Seidel uses values just computed, a slight discontinuity occurs between the row of cells where the sweep begins and finishes. This discontinuity taints the final solution. More Jacobi iterations are required to smooth the error than if a Gauss-Seidel iteration were used and twice the memory. A scheme analogous to red-black on a square grid, red-black-green on a grid comprised of hexagons, might solve this problem since there would be less emphasis on the position where the sweep begins and ends.

2.4 Computer Implementation

This section covers some of the ugly computer details that might not be immediately obvious. In order to translate the finite difference equations to computer code, we need to be able to store the scalar fields in computer memory. The grid can be conveniently partitioned into five subdivisions, each roughly shaped like a segment of an orange, or a lune. The cells comprising one of these subdivisions are shown in white in Fig. 2.11.a.

Fig. 2.11.b shows how a scalar field defined at these cells is stored in a two-dimensional

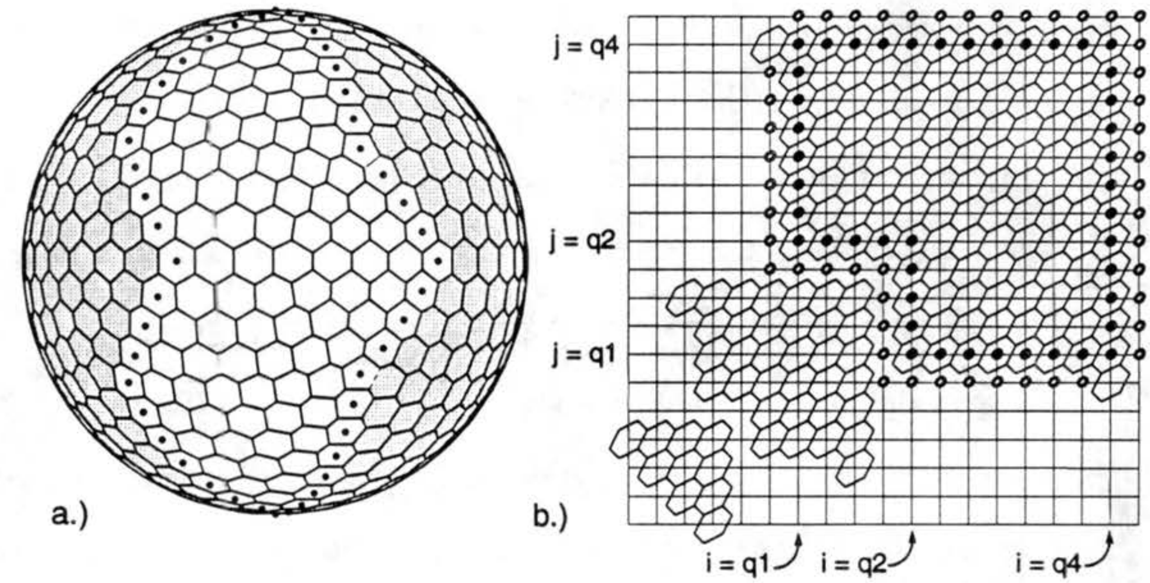

FIGURE 2.11: a.) One of five lune shaped subdivision of the grid with 642 cells. b.) Layout of lune shaped subdivisions in computer memory for the first three grid resolutions.

block of computer memory. The intersections of the horizontal and vertical lines represent locations in memory, or array elements. The upper-right group of array elements with hexagons drawn around them store a scalar field for the white cells in Fig. 2.11.a. For example, the eight white cells along the equator in Fig. 2.11.a correspond to the eight memory locations that are contiguously positioned along the diagonal of the square block of memory. Cells from different grid resolutions that cover the same lune shaped region of the sphere reside in the same block of memory. The middle group of hexagons in Fig. 2.11.b represents the cells of the grid with 128 cells covering the same region of the sphere. The group of 10 hexagons in the lower-left part of Fig. 2.11.b represents the cells of the lowest resolution grid. To store a scalar field variable at a resolution with 642 cells as shown in Fig. 2.11.a requires five square arrays of memory like the one in Fig. 2.11.b. We will call one of these square blocks of memory a *panel*.

This scheme can be extended to higher resolutions in an obvious way. Storing the fields in this way helps to simplify the multigrid code, but quite a bit of the allocated memory is unused. This is not really a problem since the whole model uses only a small fraction of the memory available on most computers.

To compute a finite difference approximation at an arbitrary cell requires information from neighboring cells. Every cell on the grid is surrounded by other cells, but not every memory location is surrounded by memory locations containing field values. For example, consider the right most white cell along the equator in Fig. 2.11.a. Its corresponding position in memory is the upper-right most array element with a hexagon drawn around it. In forming a finite difference approximation for this cell, we will need values from six memory locations. Three of the values are contained in the same panel as the center point, and the remaining three values are stored in a different panel. In general, numerical operations at memory locations depicted with solid black dots in Fig. 2.11.b require information from other panels. The subroutine called `wrap` fills the array elements depicted with open circles with the appropriate numbers.

Finally, we show one of the model's procedures. This will demonstrate how the cells are indexed in a more-or-less lexicographical order. Fig. 2.12 shows the computer code to calculate the finite difference Laplacian. [See equation (2.22).] In the code, the variable `M` determines the resolution of the finest grid. Fig. 2.11.b shows one panel for the 642 cell grid. This corresponds to `M` = 3. So, the ninth line of code will set `q1` = 6, `q2` = 10, `q3` = 17. Note that in the C computer language, all arrays are indexed starting from 0. In this procedure, the array `A` contains a scalar field. This array has three indices,

```
void Laplacian(scalar_panel area_inv,scalar_panel_six ldL,
               scalar_fine temp, scalar A)
{
   int qq1[6] = {1, 3,  6, 11, 20,  37};
   int qq2[6] = {2, 5, 10, 19, 36,  69};
   int qq4[6] = {3, 8, 17, 34, 67, 132};
   int i, j, k, n, q1, q2, q4;

   q1 = qq1[M-1]; q2 = qq2[M-1]; q4 = qq4[M-1];

/* NORTH POLE */
   temp[0][r4][0] = area_inv[q4][q1-1] *
      (ldL[1][q4][q1-1]*(A[4][q4][q1]-A[0][q4][q1-1]) +
       ldL[2][q4][q1-1]*(A[3][q4][q1]-A[0][q4][q1-1]) +
       ldL[3][q4][q1-1]*(A[2][q4][q1]-A[0][q4][q1-1]) +
       ldL[4][q4][q1-1]*(A[1][q4][q1]-A[0][q4][q1-1]) +
       ldL[5][q4][q1-1]*(A[0][q4][q1]-A[0][q4][q1-1]));

/* SOUTH POLE */
   temp[0][0][r4] = area_inv[q1-1][q4] *
      (ldL[0][q1-1][q4]*(A[0][q1][q4]-A[0][q1-1][q4])+
       ldL[1][q1-1][q4]*(A[1][q1][q4]-A[0][q1-1][q4])+
       ldL[3][q1-1][q4]*(A[2][q1][q4]-A[0][q1-1][q4])+
       ldL[4][q1-1][q4]*(A[3][q1][q4]-A[0][q1-1][q4])+
       ldL[5][q1-1][q4]*(A[4][q1][q4]-A[0][q1-1][q4]));

/* INTERIOR HEXAGONS  */
   for (k = 0; k <= 4; k++) {
     for (j = q1; j <= q2-1; j++)
#pragma _CRI ivdep
         for (i = q2; i <= q4; i++)
            temp[k][j-shift][i-shift] = area_inv[j][i] *
            (ldL[0][j][i]*(A[k][j+1][i]   - A[k][j][i]) +
             ldL[1][j][i]*(A[k][j+1][i+1] - A[k][j][i]) +
             ldL[2][j][i]*(A[k][j][i+1]   - A[k][j][i]) +
             ldL[3][j][i]*(A[k][j-1][i]   - A[k][j][i]) +
             ldL[4][j][i]*(A[k][j-1][i-1] - A[k][j][i]) +
             ldL[5][j][i]*(A[k][j][i-1]   - A[k][j][i]));
      for (j = q2; j <= q4; j++)
#pragma _CRI ivdep
         for (i = q1; i <= q4; i++)
            temp[k][j-shift][i-shift] = area_inv[j][i] *
            (ldL[0][j][i]*(A[k][j+1][i]   - A[k][j][i]) +
             ldL[1][j][i]*(A[k][j+1][i+1] - A[k][j][i]) +
             ldL[2][j][i]*(A[k][j][i+1]   - A[k][j][i]) +
             ldL[3][j][i]*(A[k][j-1][i]   - A[k][j][i]) +
             ldL[4][j][i]*(A[k][j-1][i-1] - A[k][j][i]) +
             ldL[5][j][i]*(A[k][j][i-1]   - A[k][j][i]));}
}
```

FIGURE 2.12: Block of code to calculate the finite difference Laplacian. See equation (2.22).

that is, `A[k][j][i]`. The left most index determines which panel the array element is on. Recall that a scalar field is stored in five panels, so, $0 \le \mathtt{k} \le 4$. Looking at Fig. 2.11.b, it is clear that the inner `j` and `i` loops will step through all the grid points on a given panel at resolution $\mathtt{M} = 3$.

Chapter 3

Numerical Results

3.1 Introduction

The model described in Chapter 2 was run using the suite of seven test cases proposed by Williamson (1992). The same test cases were also run using the NCAR spectral transform model, and the Arakawa and Lamb (1981) shallow water model for purposes of comparing height field errors, globally integrated invariants and performance. For all the test cases, the Arakawa-Lamb (hereafter AL) model, described in Appendix B, was run with 4° latitude by 5° longitude resolution, and the Twisted Icosahedral Grid (hereafter TIG) model was run with 2562 cells for all the test cases and 10242 cells for tests 1, 5, 6 and 7. The NCAR spectral model, described in Appendix C, was run at triangular truncation T42. This suite is appropriate for testing the TIG model since several of the cases directly test the effect of flow over the poles. Table 3.1 summarizes the seven test cases. In all test cases except case 5 there is no surface topography, so $h^* = h$.

Table 3.1: Overview of the seven test cases used to test the model.

Test Case	Comments	Full Equations	Steady State Solution	Analytic Solution Exists	Topography	Flow directed over the poles
1.) Advection of Cosine Bell over the Pole	Wind field is prescribed and nondivergent. Height equation reduces to advection equation with constant coefficients. Advecting current can be directed over the pole	no (wind field not predicted)	no	yes	no	yes
2.) Global Steady State Nonlinear Zonal Geostrophic Flow	Solid body rotation or zonal flow with the necessary geostrophic height field to maintain steady state. Coriolis parameter can be made a function of latitude.	yes	yes	yes	no	yes
3.) Steady State Nonlinear Zonal Geostrophic Flow with Compact Support	Simulates mid-latitude jet. Jet can be directed over the poles of the computational grid.	yes	yes	yes	no	yes
4.) Forced Nonlinear System with a Translating Low	Forcing terms are used to construct unsteady test case with analytic solution. Simulates short-wave trough embedded in a westerly jet.	yes	no	yes	no	no
5.) Zonal Flow over an Isolated Mountain	Steady zonal flow of test case 2 is directed over conical mountain centered at mid-latitudes.	yes	no	no	yes	no
6.) Rossby-Haurwitz Wave	Rossby-Haurwitz wave with zonal wavenumber 4.	yes	no	no (yes, for nondivergent equations)	no	no
7.) Analyzed 500 mb Height and Wind Field Initial Conditions	Observed 500 mb height field for 0000 GMT December 21, 1978. Mean height field set to 10 km. Strong flow over north pole.	yes	no	no	no	yes

3.2 Analysis of Results

We define error measurements to quantitatively measure the difference between an approximate numerical solution and the actual solution, or at least between two

approximate solutions. For some arbitrary scalar function ξ, define the area weighted average $I(\xi)$ to be

$$I(\xi) = \frac{1}{A_e} \sum_{i}^{\text{all grid points}} \text{area}_i \xi_i, \tag{3.1}$$

where area_i is the area of cell i, ξ_i is the discrete value of ξ at cell i, and

$$A_e = \sum_{i}^{\text{all grid points}} \text{area}_i \tag{3.2}$$

is the area of the earth. Let $h(\theta, \lambda, t)$ be a numerically approximated height field at some time t and $h_T(\theta, \lambda, t)$ be the exact height at that same time. We can define the following measures of error based on the l_1, l_2 and l_∞ vector norms:

$$l_1(h) \equiv \frac{I[|h(\theta, \lambda) - h_T(\theta, \lambda)|]}{I[|h_T(\theta, \lambda)|]}, \tag{3.3}$$

$$l_2(h) \equiv \frac{\{I[(h(\theta, \lambda) - h_T(\theta, \lambda))^2]\}^{1/2}}{\{I[h_T(\theta, \lambda)^2]\}^{1/2}}, \tag{3.4}$$

$$l_\infty(h) \equiv \frac{\max_{\text{all } \theta, \lambda}|h(\theta, \lambda) - h_T(\theta, \lambda)|}{\max_{\text{all } \theta, \lambda}|h_T(\theta, \lambda)|}. \tag{3.5}$$

The quantity derived in (3.4) is the normalized RMS of the difference between the numeric solution and the exact solution. While (3.3) and (3.4) measure the quality of the numeric solution as a whole, (3.5) measures the worst grid point on the grid. The wind fields are not compared since the TIG model does not directly predict the wind.

In addition, we can measure the models' performance by recording how well they conserve certain globally integrated invariants. The quantities we will check include mass $\xi_1 = h^*$, total energy $\xi_2 = [h^* v \bullet v + g(h^2 - h_s^2)]/2$, and potential enstrophy, $\xi_3 = \eta^2/(2h^*)$. For each time step, we compute and record the normalized integral

$$I_i(\xi(t)) = \frac{I[\xi(\theta, \lambda, t)] - I[\xi(\theta, \lambda, 0)]}{I[\xi(\theta, \lambda, 0)]} \tag{3.6}$$

for $i = 1, 2, 3$.

Also, the area-weighted global integrals of vorticity, $I(\zeta)$, and divergence, $I(\delta)$, are computed and recorded each time step. These quantities are initially zero and should remain zero during the integration.

3.3 Test Case 1: Advection of a Cosine Bell over the Pole

3.3.1 Description

This test case differs from later test cases in that the winds are not predicted, but prescribed. Instead of time stepping the wind equations, an analytical formula is used to update the winds each time step. Since the prescribed wind field is nondivergent, the height equation reduces to just the advection equation for h^* with constant coefficients, and the initial height field is simply advected along by the wind. Expressed mathematically, $\nabla \bullet v = 0$, and v is constant in time, so the height equation (1.38) becomes just the linear advection equation

$$\frac{\partial h}{\partial t} + v \bullet \nabla h = 0. \tag{3.7}$$

Without numerical dispersion, the height field should maintain its initial shape. The advecting wind is a jet along a great circle all the way around the planet, and the initial height field is a cosine shaped bump of fluid. We will define the parameter α to be the angle between the axis of solid body rotation and the polar axis of the numerical grid. In testing the models, we use $\alpha = 0.0,\ 0.05,\ \pi/2 - 0.05$ and $\pi/2$. These values allow comparison of advecting the bell along the equator, and advecting it over the pole. Presumably, the errors should be smaller as the bell moves along the equator than when the bell passes over the pole, since moving along the equator avoids the problems associated with the pole. Fig. 3.1 shows the wind field when $\alpha = 0.05$ and $\alpha = \pi/2 - 0.05$.

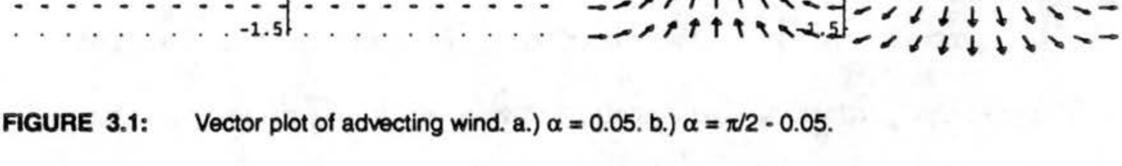

FIGURE 3.1: Vector plot of advecting wind. a.) $\alpha = 0.05$. b.) $\alpha = \pi/2 - 0.05$.

In the Arakawa-Lamb shallow water model, the u and v components of the wind are given by

$$u(\theta, \lambda) = u_0(\cos\theta\cos\alpha + \sin\theta\cos\lambda\sin\alpha) \tag{3.8}$$

$$v(\theta, \lambda) = -u_0\sin\lambda\sin\alpha, \tag{3.9}$$

where u_0 is the magnitude of the advecting wind. The bell is blown around the earth every 12 days, so, $u_0 = 2\pi a/(12 \text{ days}) = 38.6 \text{ m s}^{-1}$.

For the TIG model, the wind is given in terms of stream function and velocity potential. Namely,

$$\psi(\theta, \lambda) = -au_0(\sin\theta\cos\alpha - \cos\lambda\cos\theta\sin\alpha), \tag{3.10}$$

$$\chi(\theta, \lambda) = 0. \tag{3.11}$$

The initial height field for both models is given by

$$h(\theta, \lambda) = \begin{cases} (h_0/2)(1 + \cos(\pi r/R)) & \text{if } r < R \\ 0 & \text{if } r \geq R, \end{cases} \tag{3.12}$$

where $h_0 = 1000 \text{ m}$ is the maximum height of the bell. Initially, the bell is centered at $(\theta_c, \lambda_c) = (0, 3\pi/2)$. The variable r is the distance between some arbitrary point on the sphere (θ, λ) and (θ_c, λ_c). The distance r is given by $r = \cos^{-1}[\sin\theta_c\sin\theta + \cos\theta_c\cos\theta\cos(\lambda - \lambda_c)]$. The radius of the bell R is chosen to be $R = a/3$.

3.3.2 Numerical Results

The NCAR, Arakawa-Lamb, TIG-2562 and TIG-10242 models were run with a time step of 600 seconds for a total simulated time of 12 days, one trip around the earth. The height field errors $l_1(h)$, $l_2(h)$ and $l_\infty(h)$ are shown below in Fig. 3.2, Fig. 3.3, Fig. 3.4 and Fig. 3.5. Note that different values are used for the range of the error for each

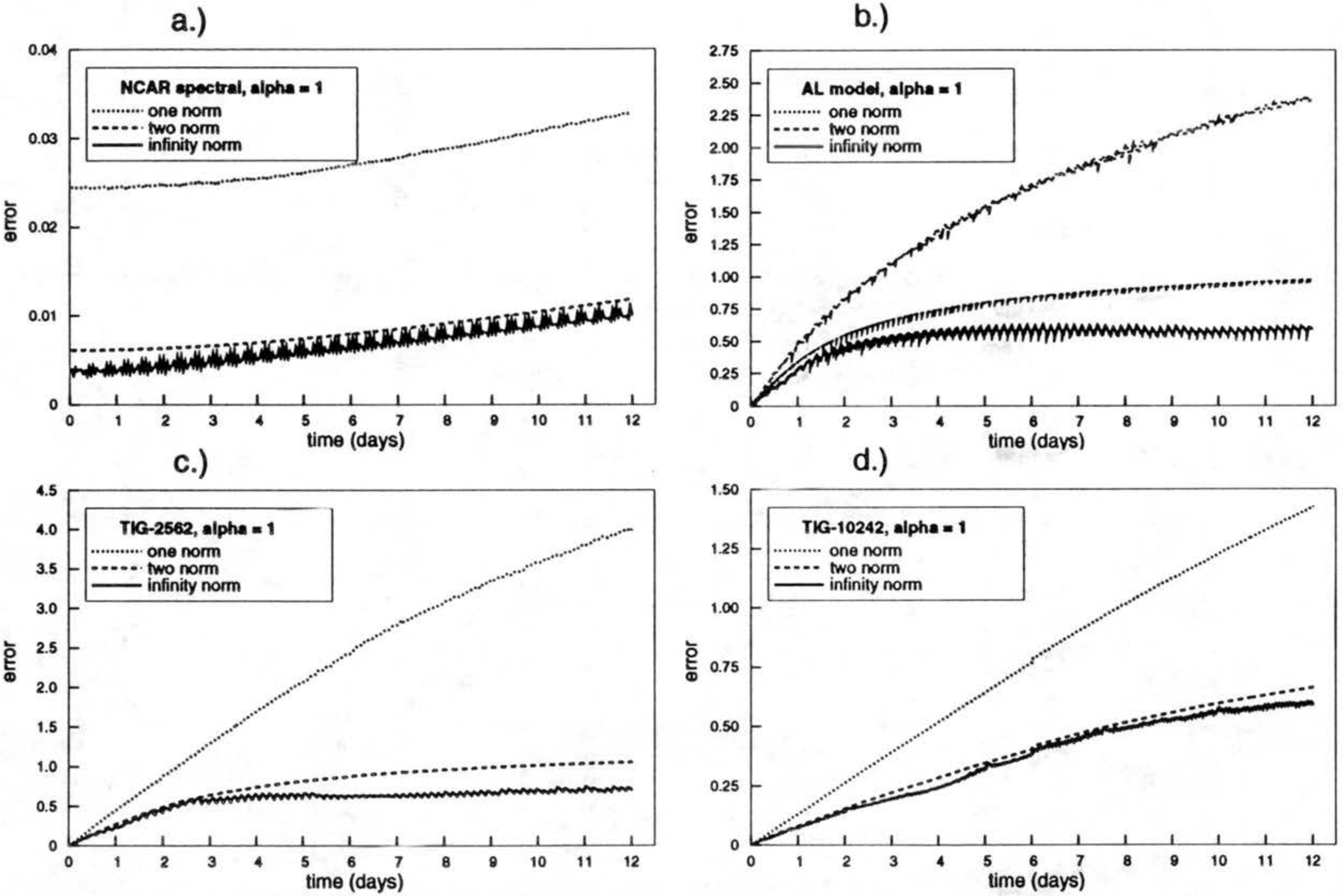

FIGURE 3.2: Test case 1. Height field errors. α = 0.0. a.) NCAR spectral b.) Arakawa-Lamb shallow water model. c.) TIG model with 2562 cells. d.) TIG model with 10242 cells.

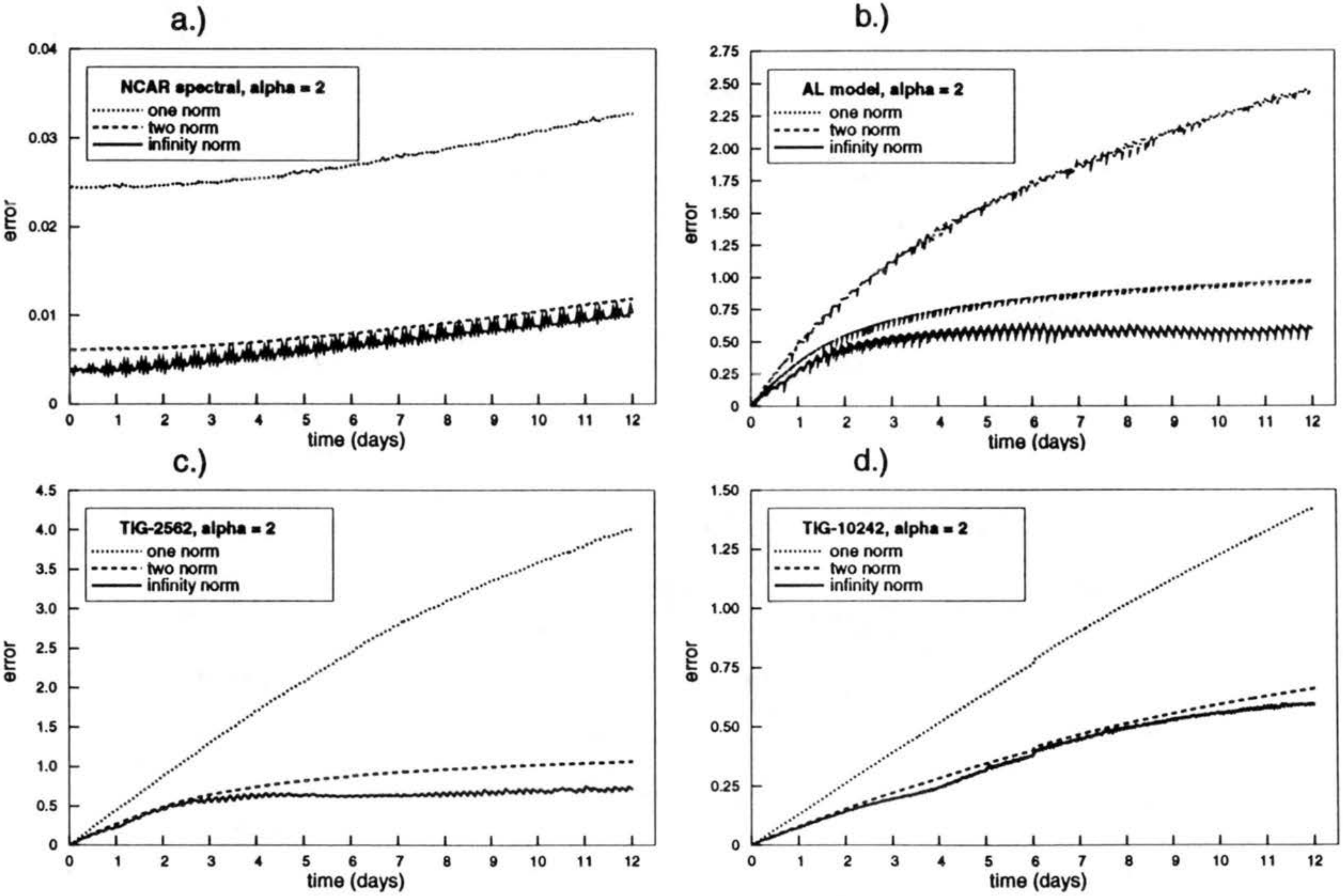

FIGURE 3.3: Test case 1. Height field errors. α = 0.05. a.) NCAR spectral b.) Arakawa-Lamb shallow water model. c.) TIG model with 2562 cells. d.) TIG model with 10242 cells.

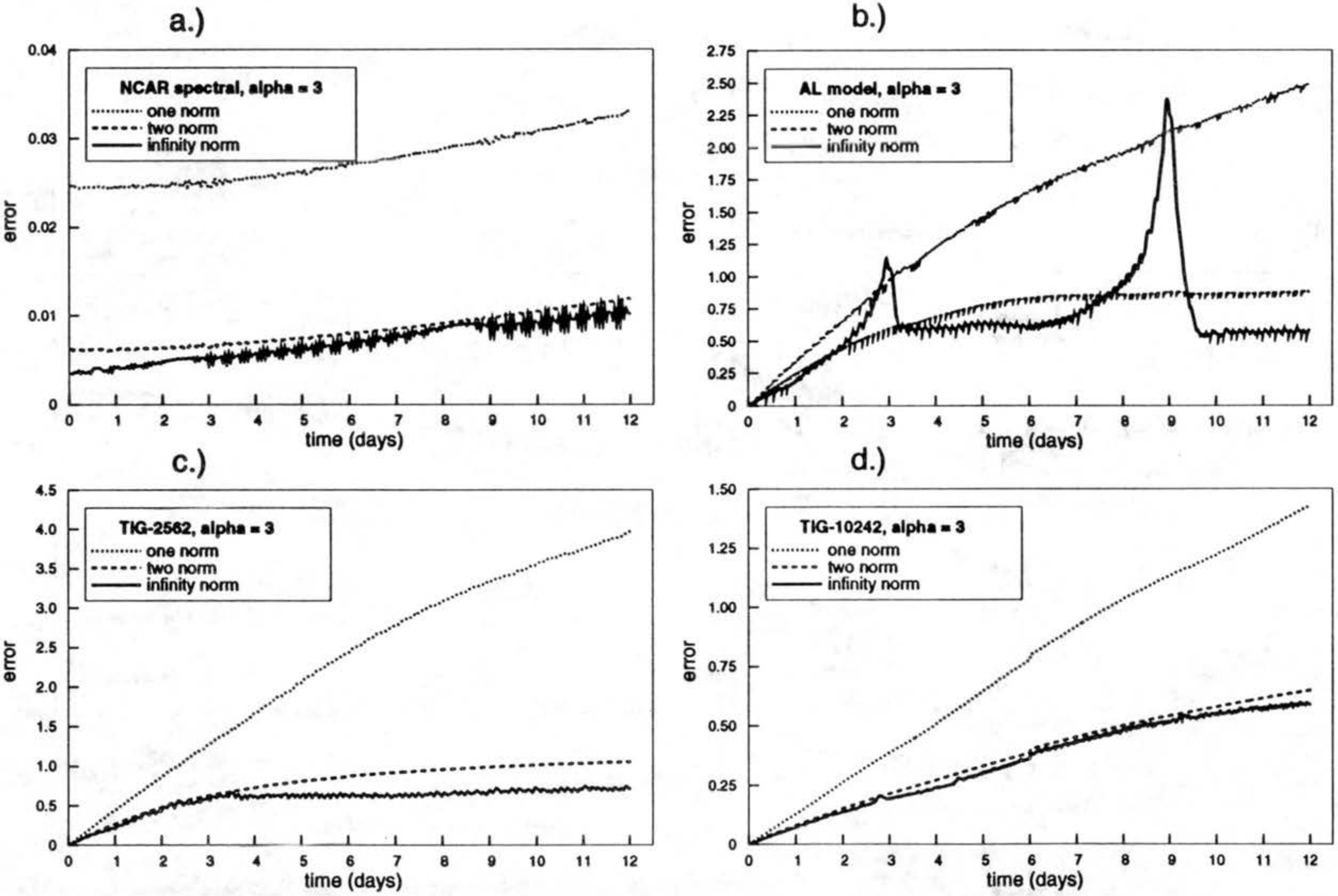

FIGURE 3.4: Test case 1. Height field errors. $\alpha = \pi/2 - 0.05$. a.) NCAR spectral b.) Arakawa-Lamb shallow water model. c.) TIG model with 2562 cells. d.) TIG model with 10242 cells.

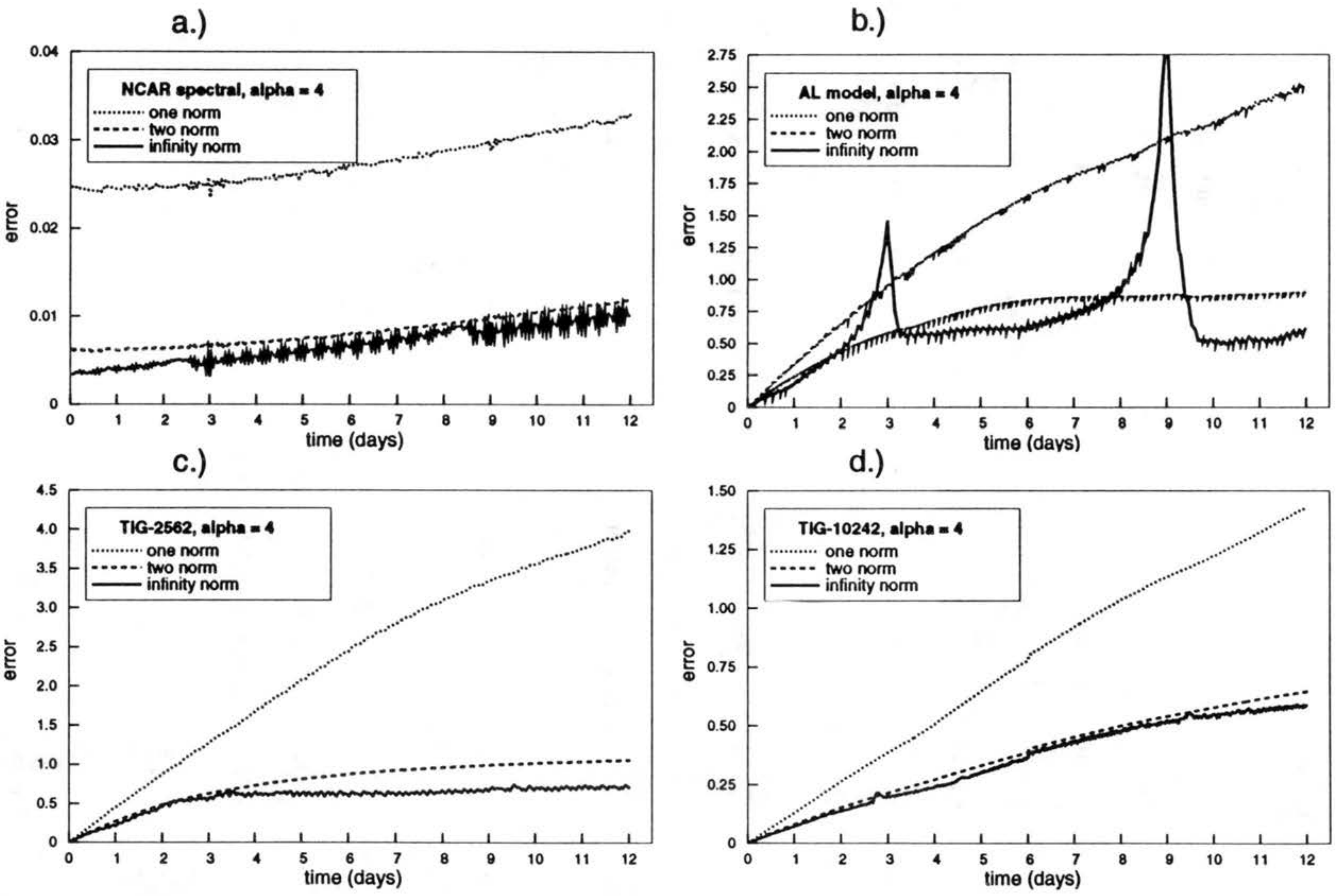

FIGURE 3.5: Test case 1. Height field errors. $\alpha = \pi/2$. a.) NCAR spectral b.) Arakawa-Lamb shallow water model. c.) TIG model with 2562 cells. d.) TIG model with 10242 cells.

model. In the plot's legend box, alpha=1 denotes $\alpha = 0.0$, alpha=2 denotes $\alpha = 0.05$, alpha=3 denotes $\alpha = \pi/2 - 0.05$ and alpha=4 denotes $\alpha = \pi/2$.

Looking at the $\alpha = 0.0$, the spectral model does quite a bit better than the finite difference models. This is true for all values of α. The reason for this has to do with computational dispersion, which is discussed below. The plots for $\alpha = 0.0$, and $\alpha = 0.05$ look pretty similar. The TIG models both do an Euler forward time step at the beginning of the sixth day to eliminate the computational mode. Looking at TIG-10242, you can see a slight kink in the l_1 norm at the start of day six, and a little noise in the l_∞ plot after day six.

The cases $\alpha = \pi/2 - 0.05$ and $\alpha = \pi/2$, are more interesting. The spectral model develops a bit of noise in the l_∞-norm after passing over the poles, and there is a kink in the l_1-norm right at the poles. For the AL shallow water model there are two striking spikes in the l_∞-norm at the poles. Since the infinity norm measures the biggest error at any individual grid point, we can conclude at least one grid point must be way off. However, the l_1 and l_2-norms measure the overall error, and are pretty similar to the $\alpha = 0.$ and $\alpha = 0.05$ cases. The TIG-2562 and TIG-10242 are similar to the $\alpha = 0.$ and $\alpha = 0.05$ cases for all three norms, with TIG-10242 better than TIG-2562. The TIG-10242 model shows a slight kink at as bell passes over the pole.

Fig. 3.6 to Fig. 3.17 show orthographic projections of the height field on days 3, 6, 9 and 12 of the model runs. The continental outlines are only for purposes of orientation. For the cases $\alpha = 0.0$, and $\alpha = 0.05$, the view point is along the equator, and for the cases $\alpha = \pi/2 - 0.05$ and $\alpha = \pi/2$ the view point is along the great circle formed by

90° and 270° degrees of longitude. In all the plots, the contour interval is 100 m. It should be noted that the 0 m contour is not plotted, since it tends to clutter the plot. Fig. 3.6 to Fig. 3.9 show the results from the AL shallow water model. Fig. 3.6 shows the case $\alpha = 0.0$. The most notable feature of this plot is the computational dispersion. [This is discussed in any computational fluids book, see, for example, Haltiner (1980)]. As the bell moves along the equator, the short wavelength modes travel with a slower phase speed than the longer wavelength modes. Hence, the overall shape tends to flatten toward the downwind side, while the gradient steepens on the upwind side, and the individual modes combine to form regions of negative height behind the bell. In general, the bell travels with a phase speed less than it should. The view point in each of these plots is looking straight down on the point where the center of the bell should be at that time. We can see the highest point slowly falls behind as time progresses. This defect is also present in the TIG models, since they are finite difference models, but is not present in the NCAR model since spectral models suffer no dispersion error for linear advection. Fig. 3.8 shows the case $\alpha = \pi/2 - 0.05$, advection nearly over the pole. Here, the bell becomes antisymmetric in the direction it is traveling. Finally, Fig. 3.9 shows the case $\alpha = \pi/2$. Here too, the bell becomes asymmetric in the direction of travel. Also, the center of the bell should pass directly over the poles.

Fig. 3.10 to Fig. 3.13 show plots of the TIG-2562 model. Fig. 3.10 shows the case $\alpha = 0.0$. Comparing with Fig. 3.6, we can see the bell stays rounder in the TIG model, but spreads out more. The AL model does not seem to spread away from the direction of travel as much. The bell remains more symmetric in the direction of travel. It is interesting that it develops two lobes of negative height behind the bell.

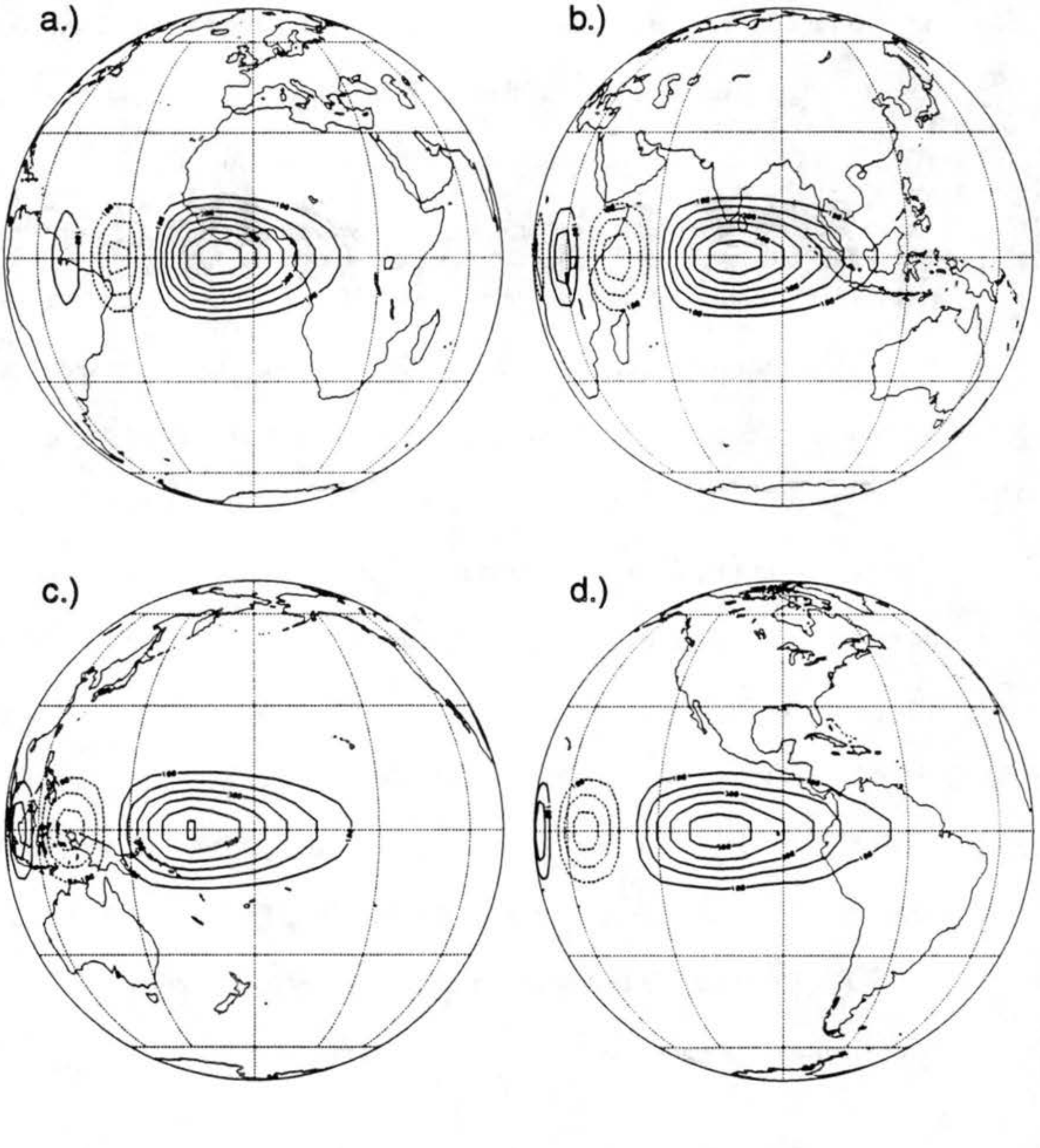

FIGURE 3.6: Test case 1. Height field (m). Arakawa-Lamb shallow water model. $\alpha = 0.0$. 100 m contour interval. a.) Day 3. b.) Day 6. c.) Day 9. d.) Day 12.

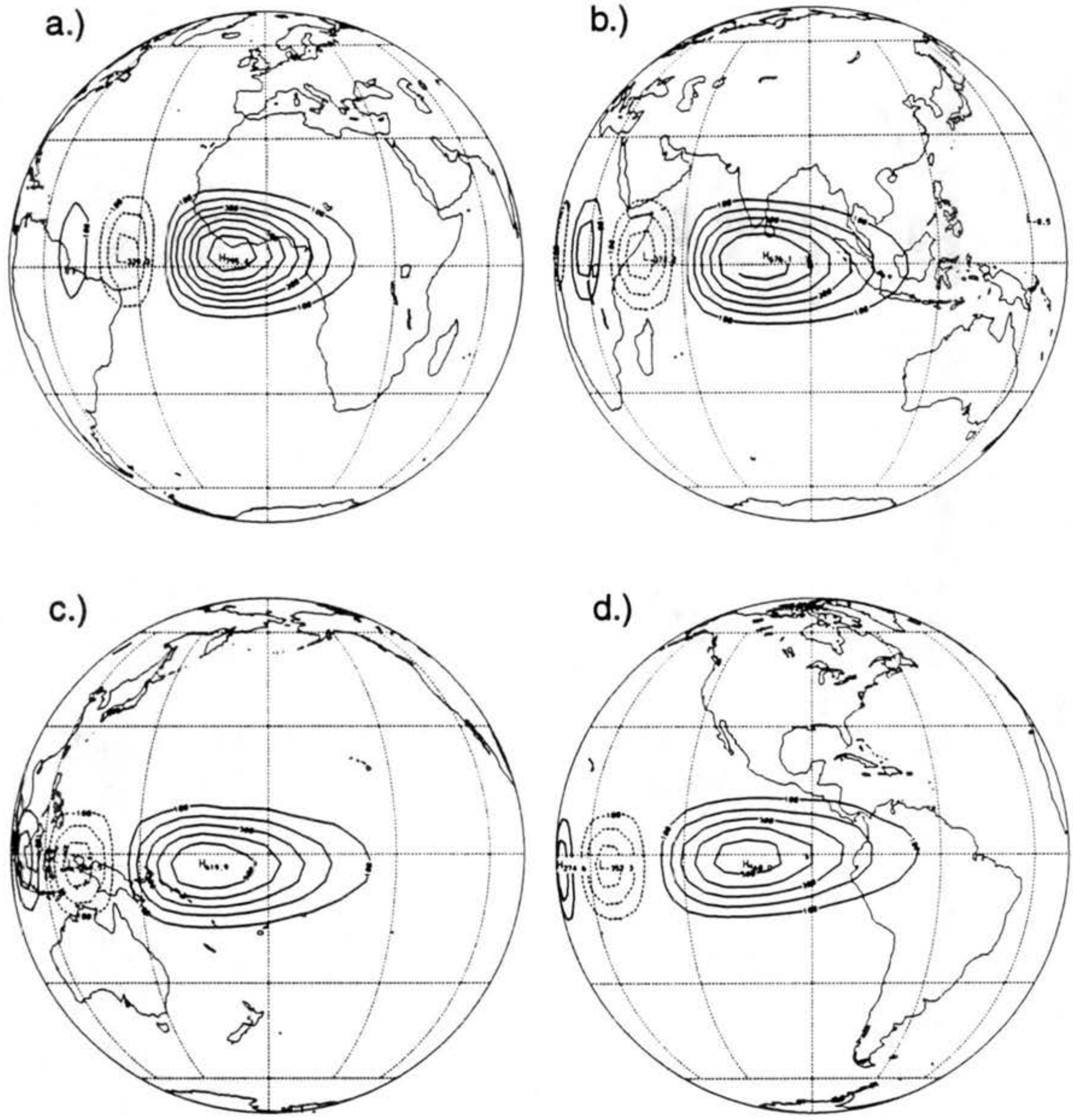

FIGURE 3.7: Test case 1. Height field (m). Arakawa-Lamb shallow water model. $\alpha = 0.05$. 100 m contour interval. a.) Day 3. b.) Day 6. c.) Day 9. d.) Day 12.

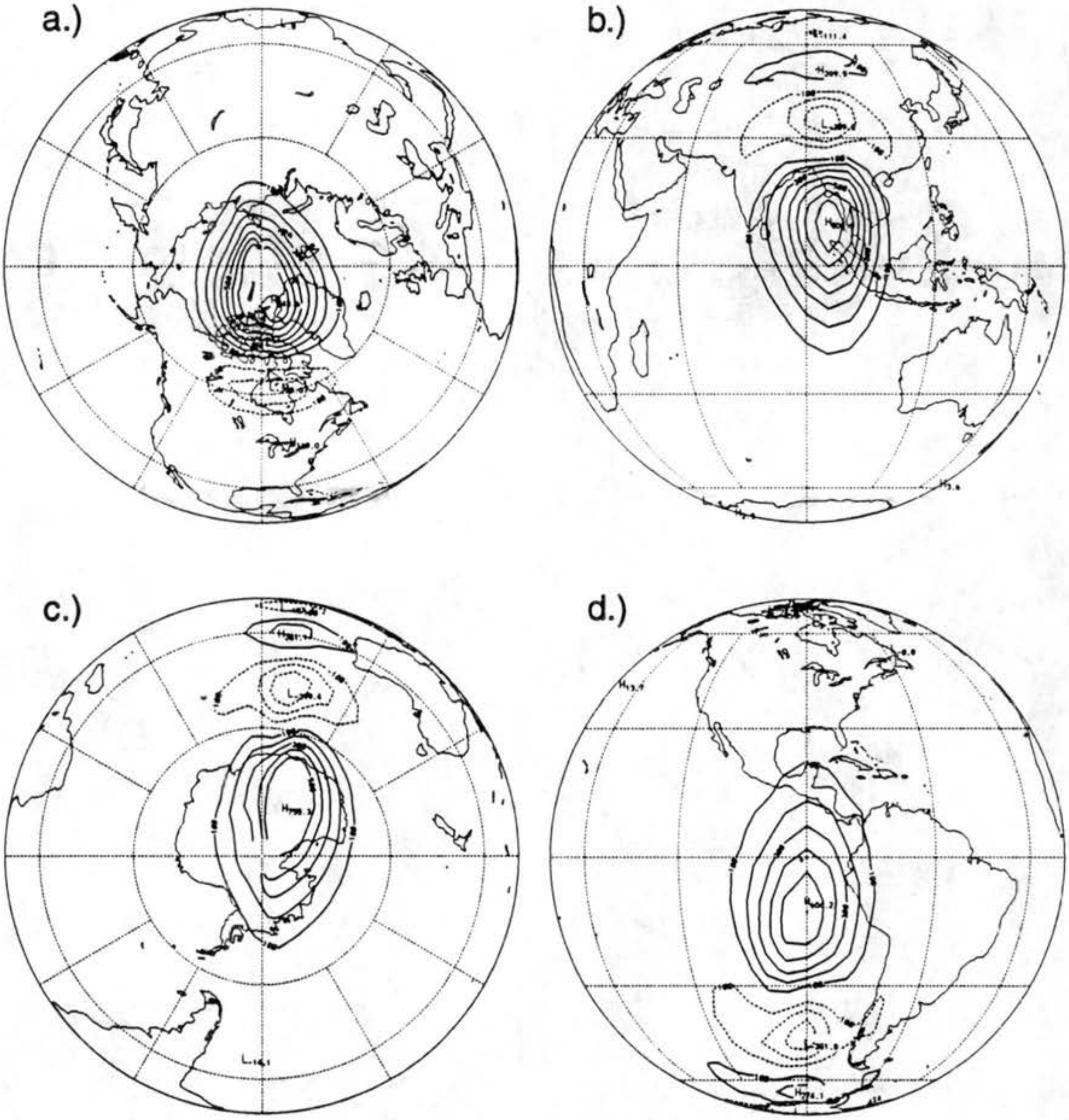

FIGURE 3.8: Test case 1. Height field (m). Arakawa-Lamb shallow water model. $\alpha = \pi/2 - 0.05$. 100 m contour interval. a.) Day 3. b.) Day 6. c.) Day 9. d.) Day 12.

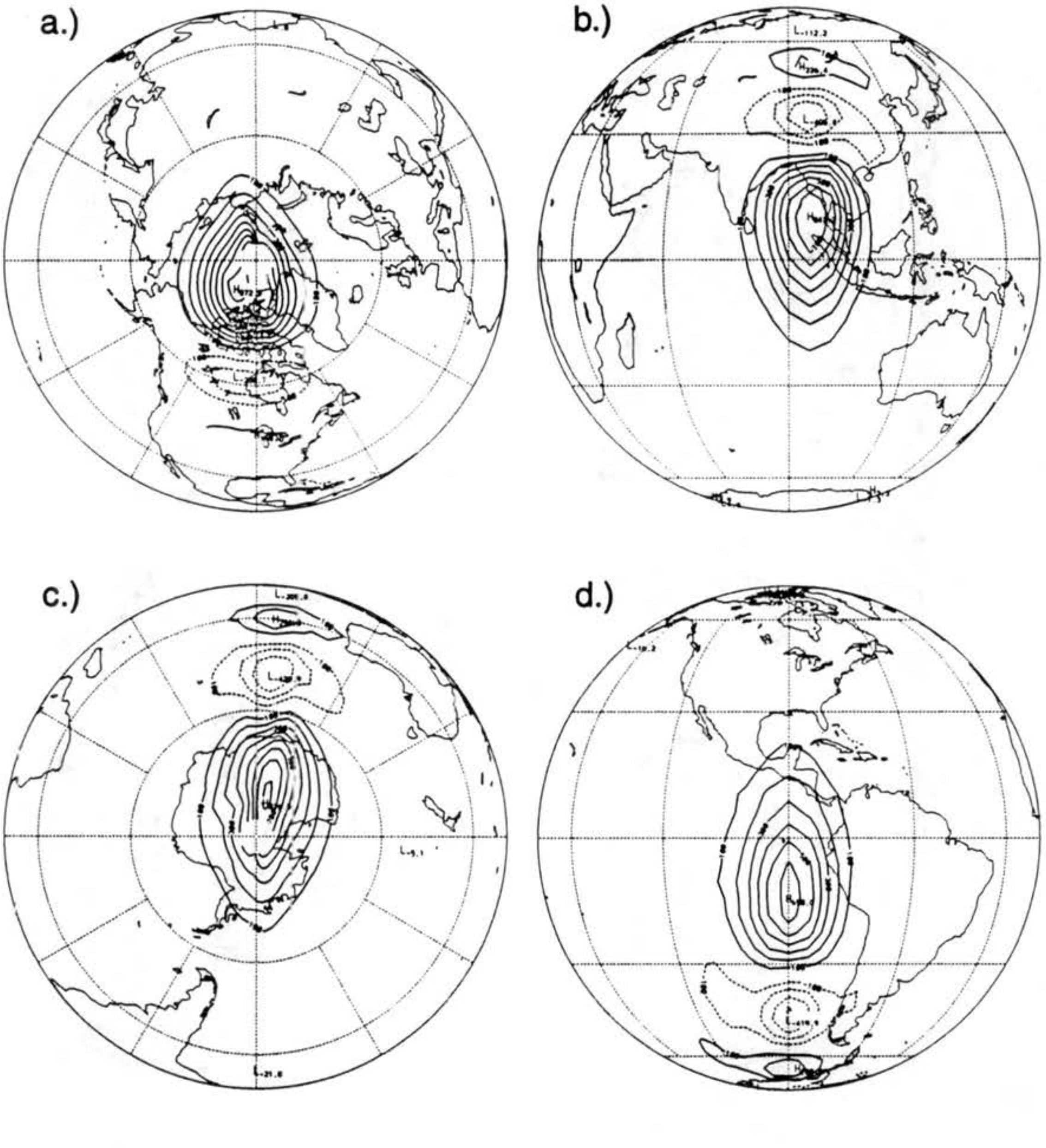

FIGURE 3.9: Test case 1. Height field (m). Arakawa-Lamb shallow water model. $\alpha = \pi/2$. 100 m contour interval. a.) Day 3. b.) Day 6. c.) Day 9. d.) Day 12.

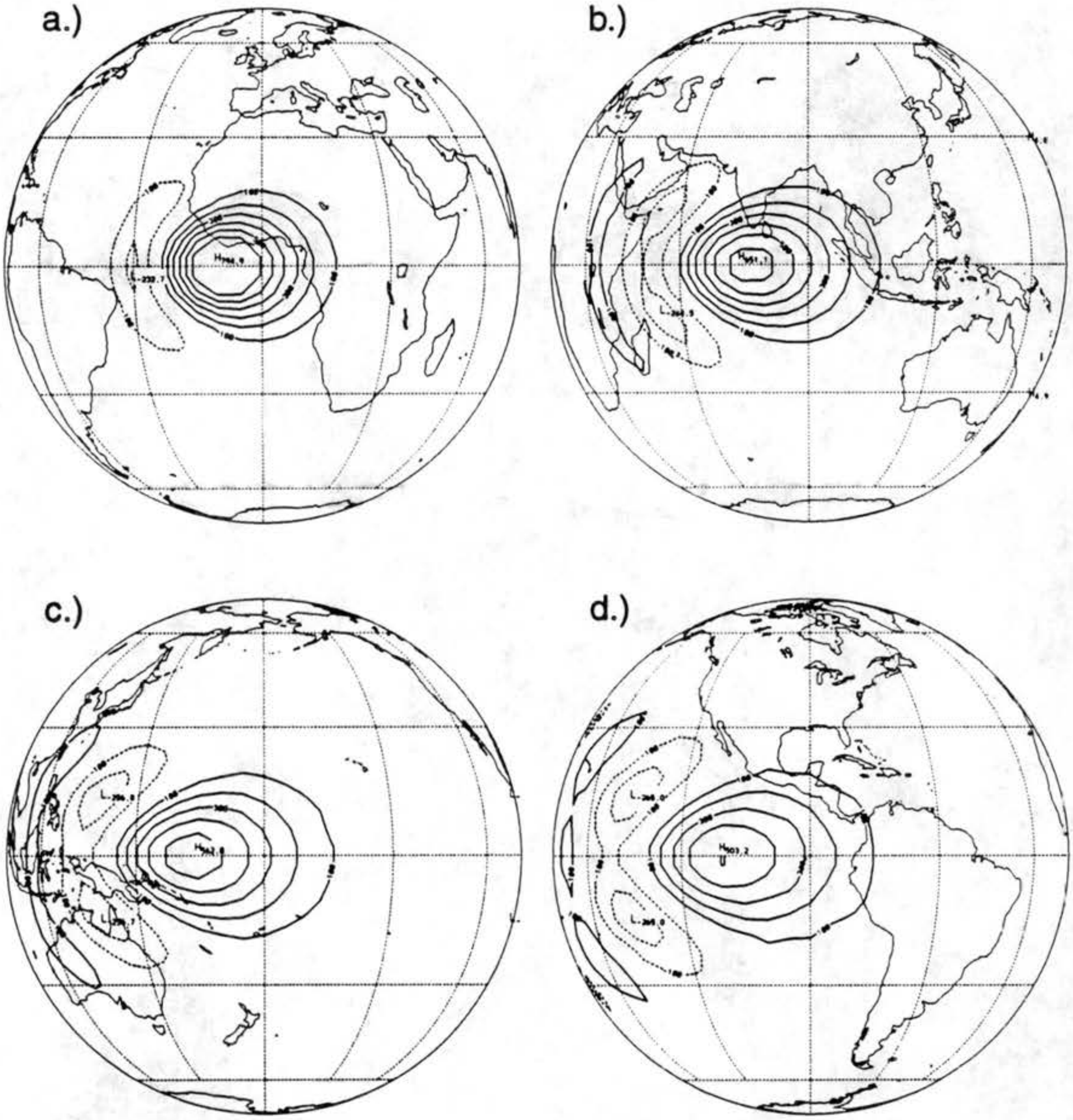

FIGURE 3.10: Test case 1. Height field (m). TIG-2562. $\alpha = 0.0$. 100 m contour interval a.) Day 3. b.) Day 6. c.) Day 9. d.) Day 12.

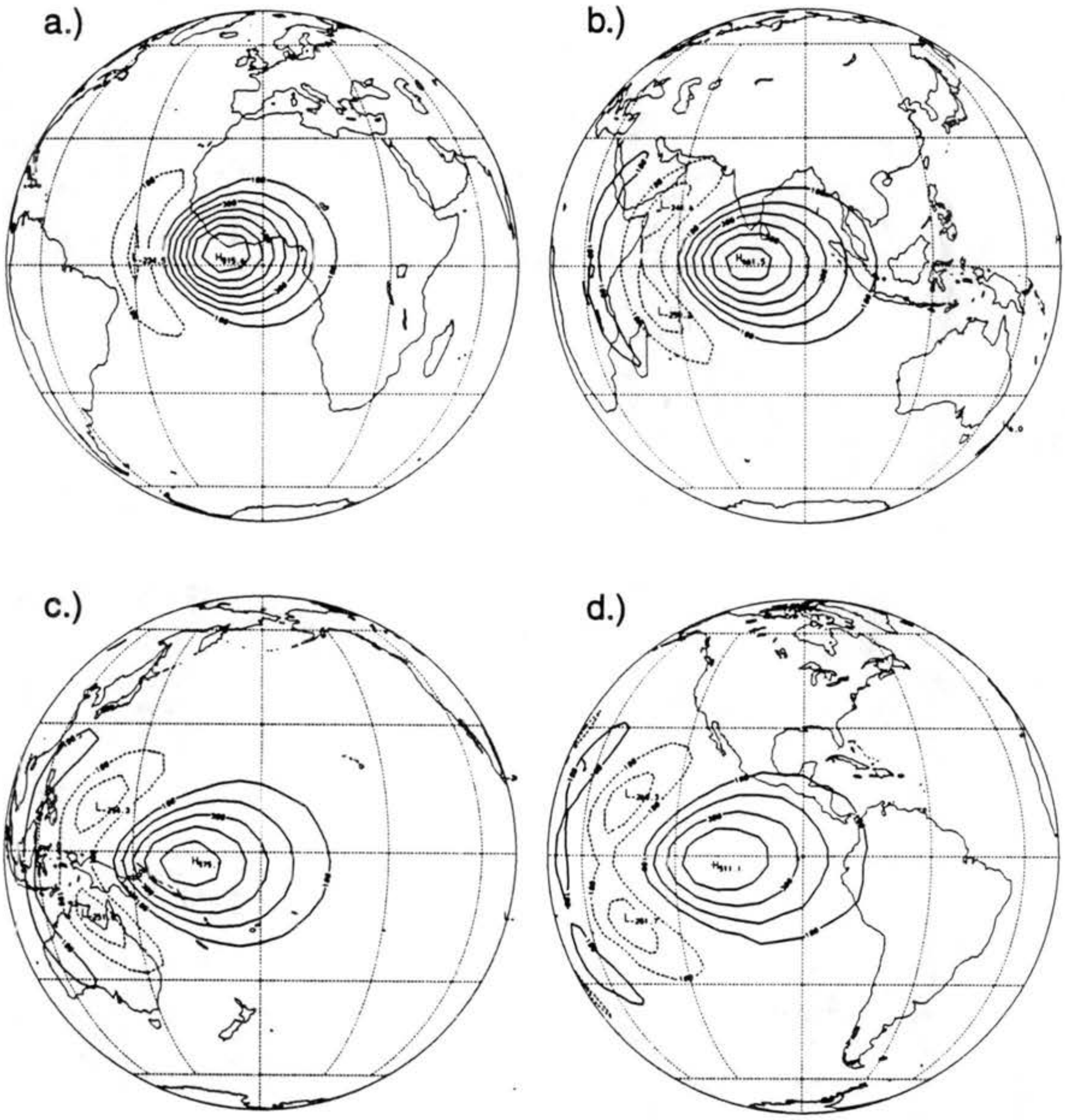

FIGURE 3.11: Test case 1. Height field (m). TIG-2562. $\alpha = 0.05$. 100 m contour interval a.) Day 3. b.) Day 6. c.) Day 9. d.) Day 12.

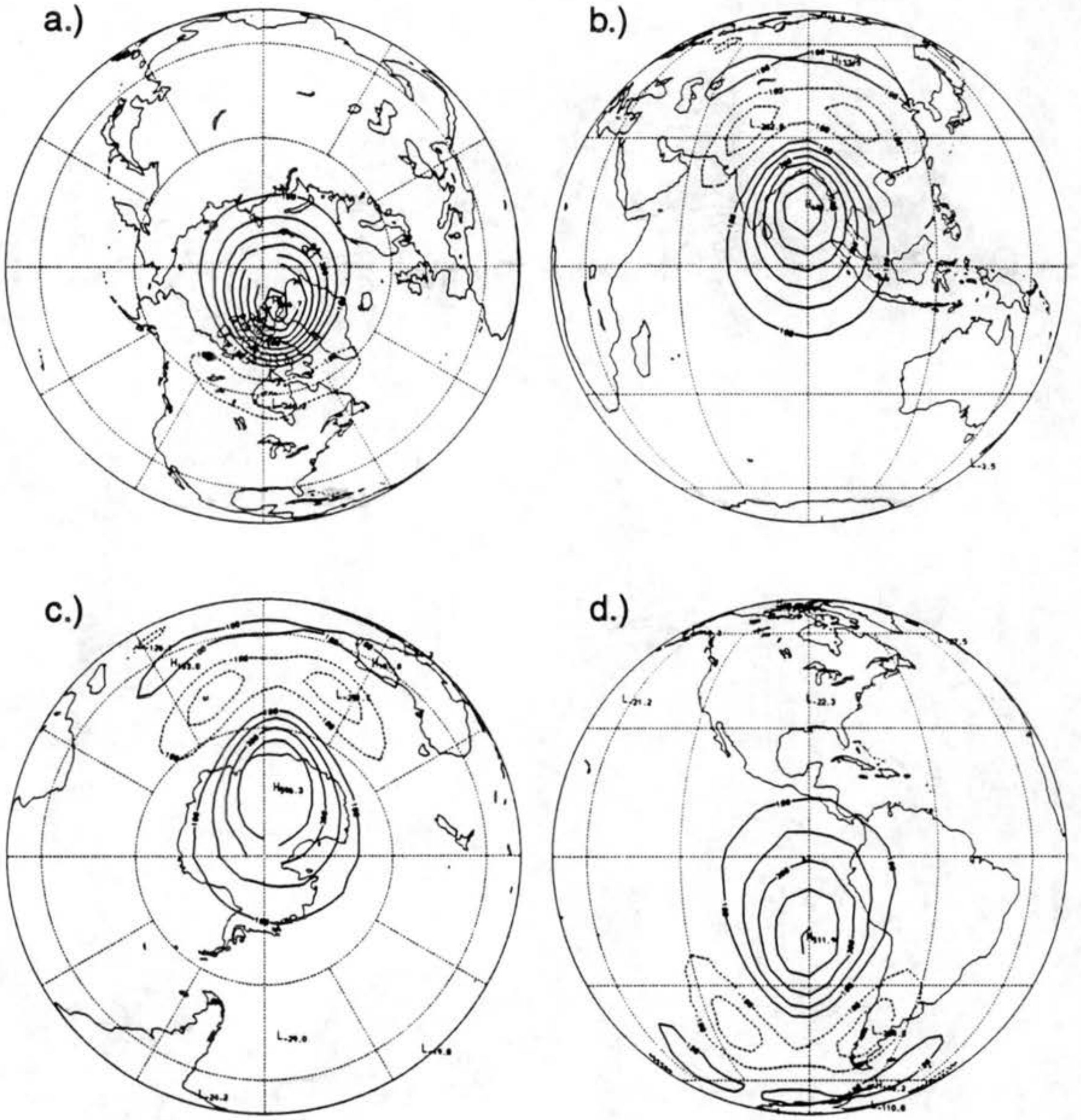

FIGURE 3.12: Test case 1. Height field (m). TIG-2562. $\alpha = p/2 - 0.05$. 100 m contour interval a.) Day 3. b.) Day 6. c.) Day 9. d.) Day 12.

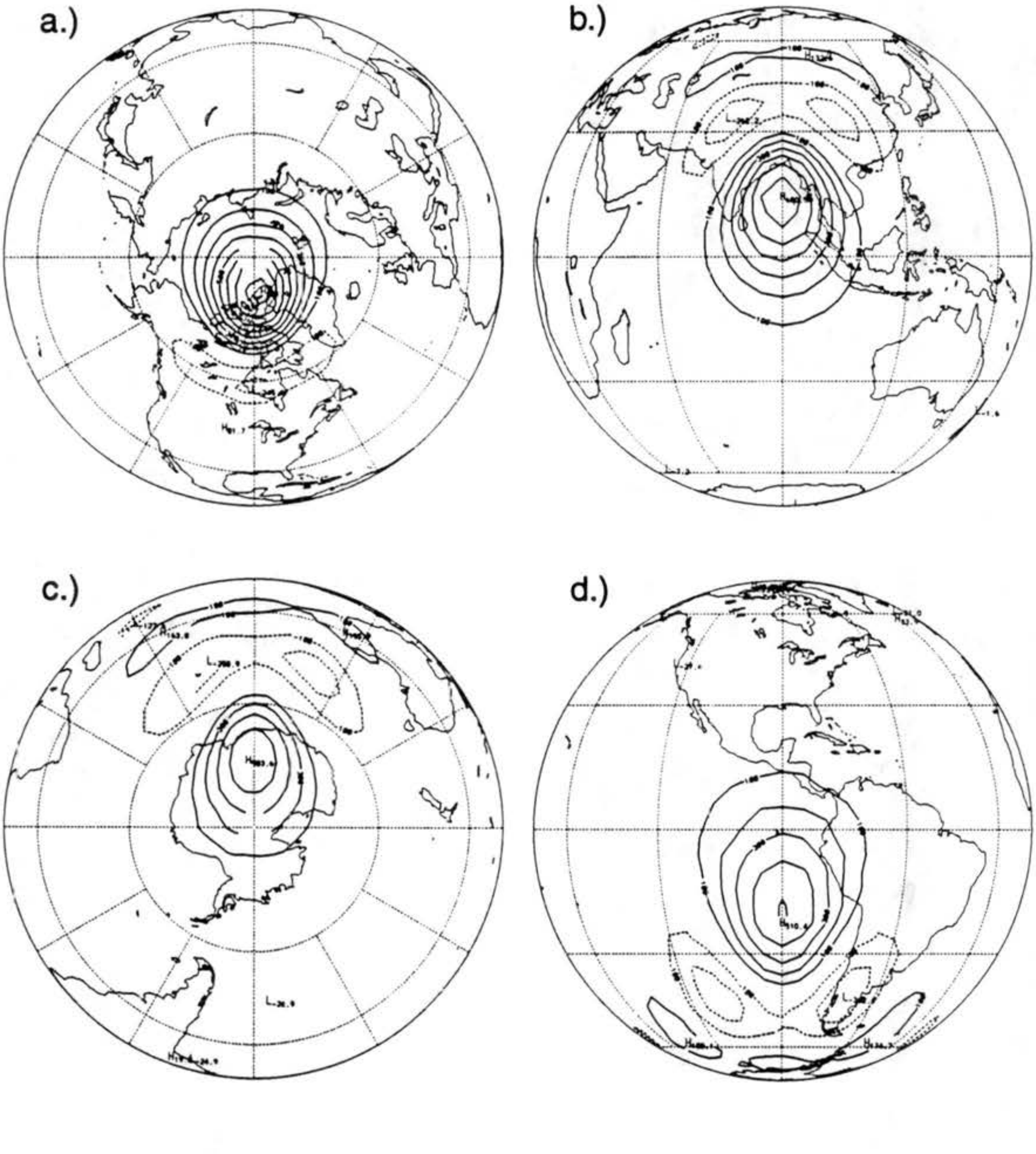

FIGURE 3.13: Test case 1. Height field (m). TIG-2562. $\alpha = p/2$. 100 m contour interval a.) Day 3. b.) Day 6. c.) Day 9. d.) Day 12.

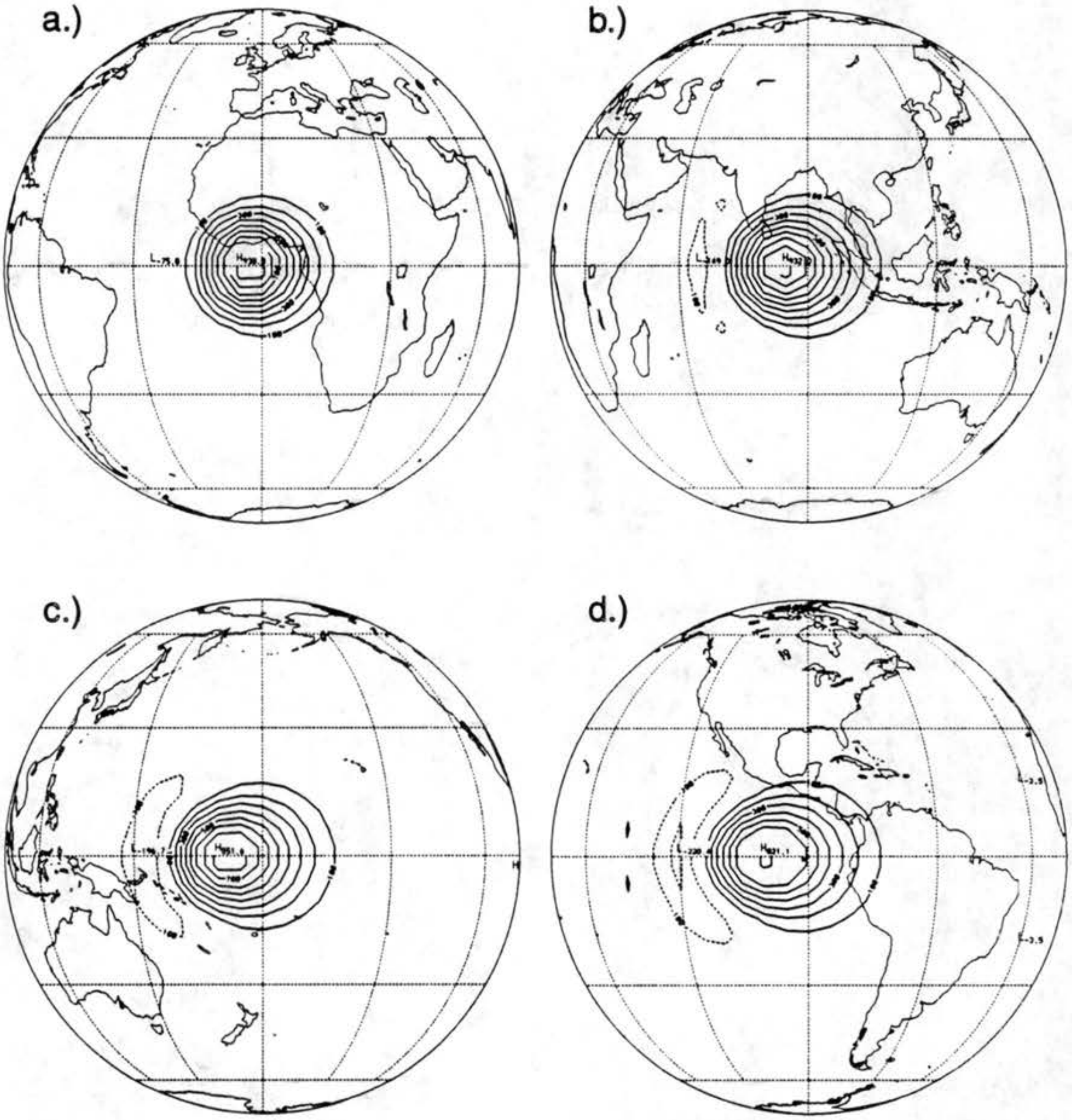

FIGURE 3.14: Test case 1. Height field (m). TIG-10242. α = 0.0. 100 m contour interval a.) Day 3. b.) Day 6. c.) Day 9. d.) Day 12.

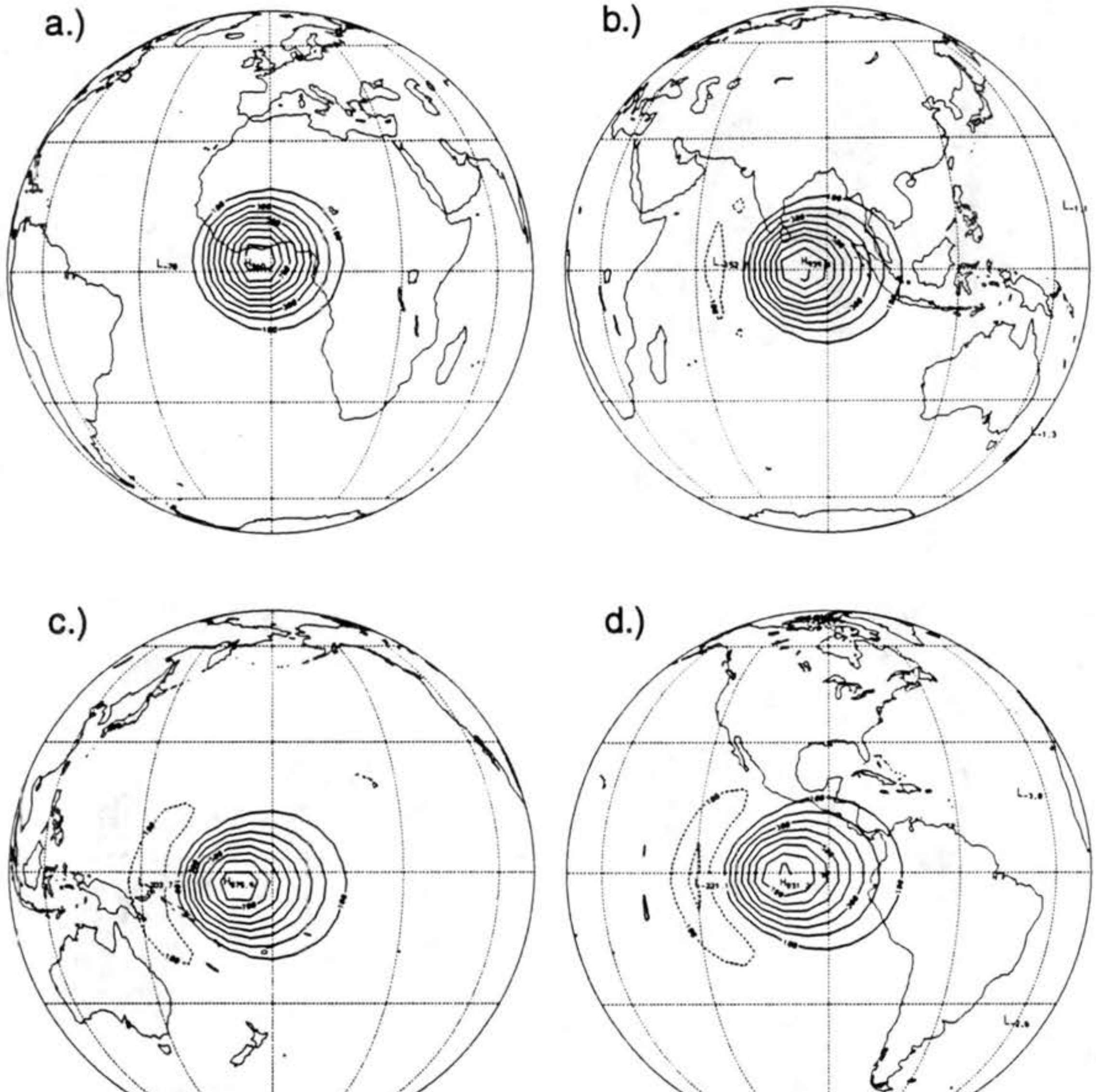

FIGURE 3.15: Test case 1. Height field (m). TIG-10242. $\alpha = 0.05$. 100 m contour interval a.) Day 3. b.) Day 6. c.) Day 9. d.) Day 12.

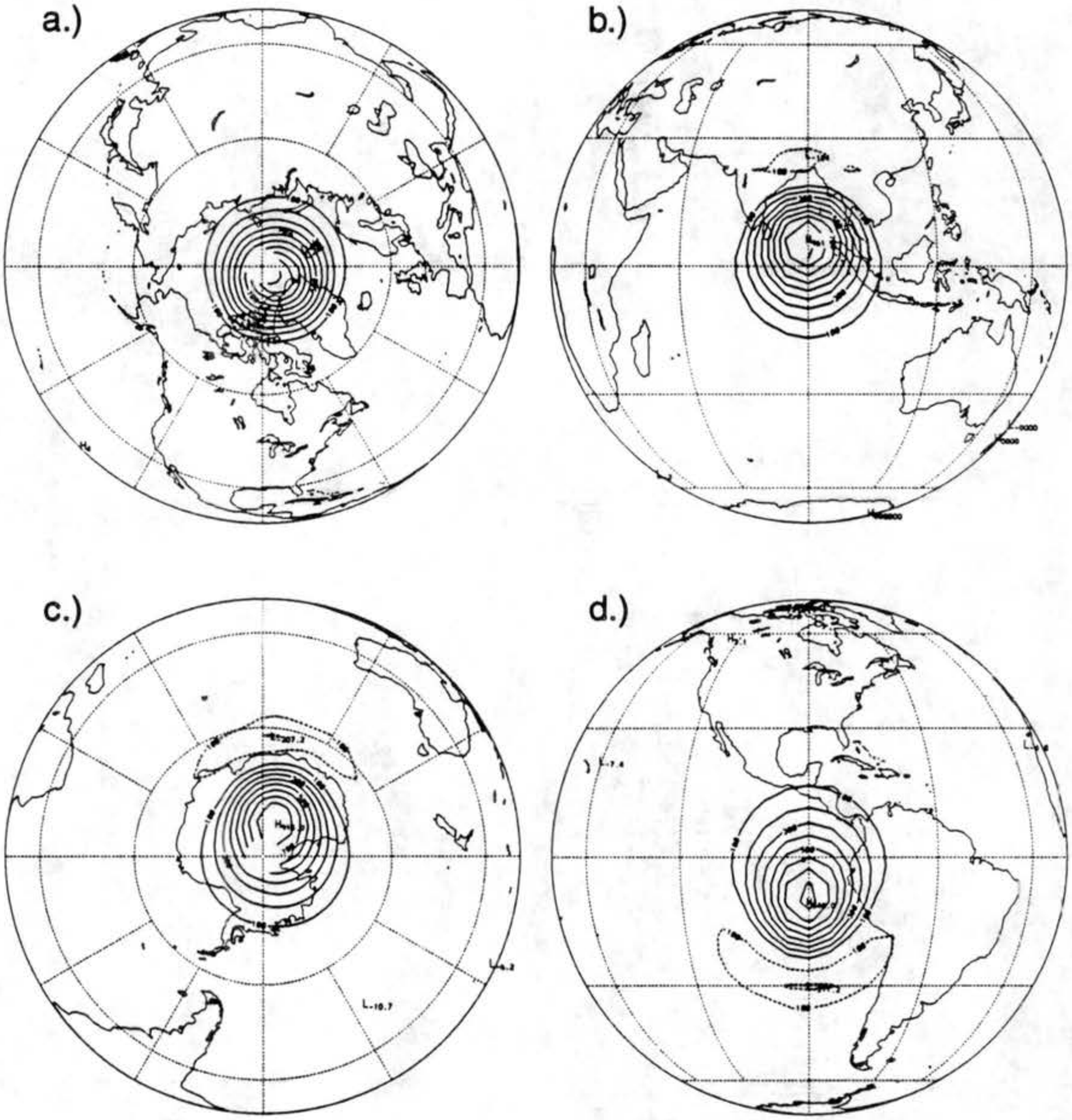

FIGURE 3.16: Test case 1. Height field (m). TIG-10242. $\alpha = p/2 - 0.05$. 100 m contour interval a.) Day 3. b.) Day 6. c.) Day 9. d.) Day 12.

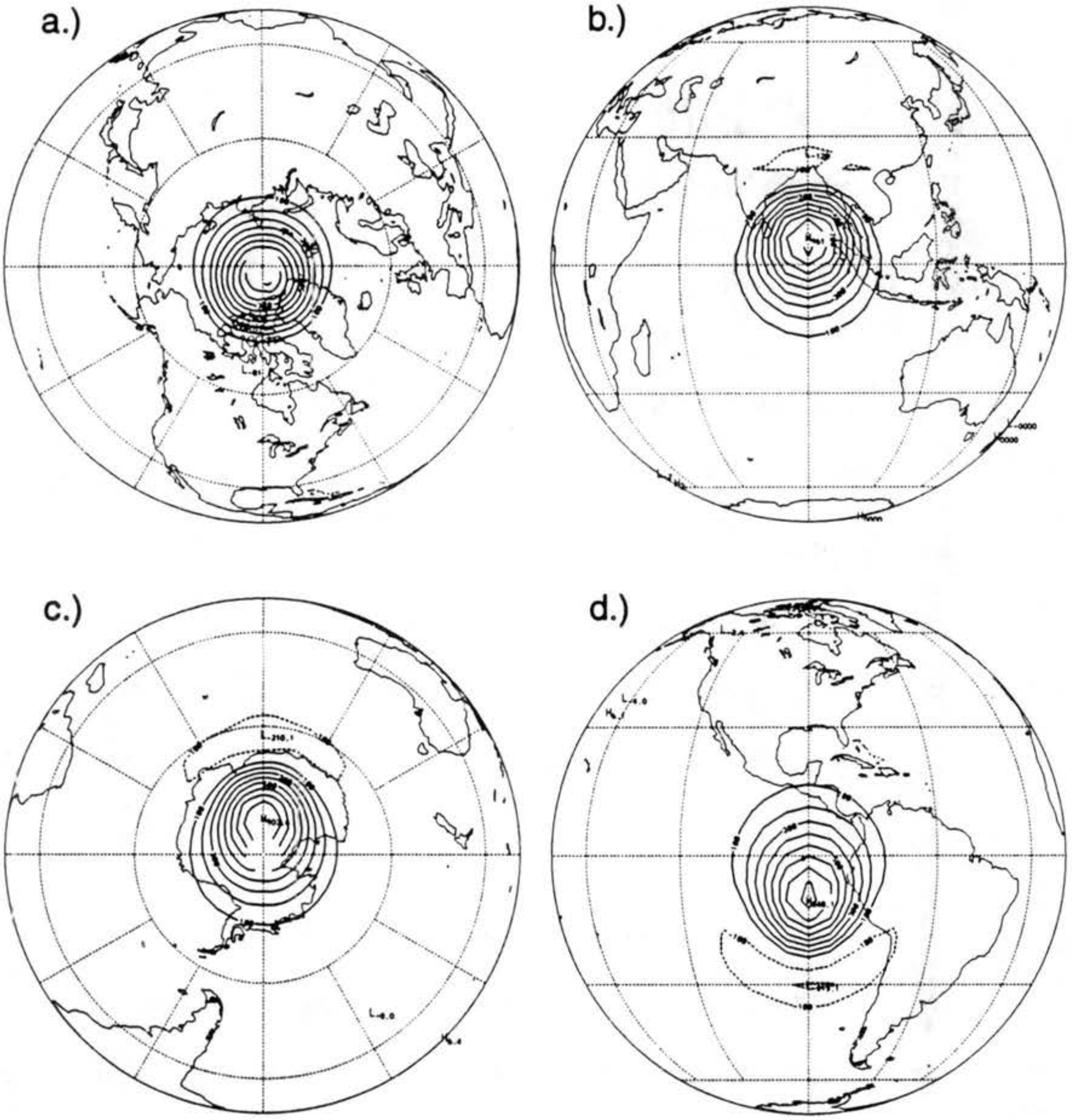

FIGURE 3.17: Test case 1. Height field (m). TIG-10242. $\alpha = p/2$. 100 m contour interval a.) Day 3. b.) Day 6. c.) Day 9. d.) Day 12.

Fig. 3.14 to Fig. 3.17 show plots of the TIG-10242 model. Here, the bell remains more intact, and does not travel as slowly as the other two models. The increase in resolution makes a big difference, but of course, it comes at a computation cost.

3.4 Test Case 2: Global Steady State Nonlinear Geostrophic Flow

3.4.1 Description

This test case models a steady state solution of the shallow water equations. The zonal wind is kept perfectly steady by the appropriate geostrophic balance. To test the effect of the pole, the Coriolis force is rotated, making it a function of latitude. If the computational grid is considered fixed in space, by changing the Coriolis parameter, we are, in effect, changing the direction of the axis of the earth's rotation so that it is no longer coincident with the axis through the poles of the grid. In this way, flow can be directed over the poles of the grid. The Coriolis parameter is given by

$$f = 2\Omega(-\cos\lambda\cos\theta\sin\alpha + \sin\theta\cos\alpha)\,. \tag{3.13}$$

We will direct the flow to be parallel and nearly parallel to the equator of the grid, and nearly directly over and directly over the pole. That is, as in test case 1, the parameter α is set $\alpha = 0.0$, $\alpha = 0.05$, $\alpha = \pi/2 - 0.05$ and $\alpha = \pi/2$. Clearly, (3.13) reduces to the *regular* Coriolis parameter when $\alpha = 0$. As the models run, the fields should remain constant.

For the Arakawa-Lamb shallow water model, the initial u and v components of the wind are given by

$$u = u_0(\cos\theta\cos\alpha + \cos\lambda\sin\theta\sin\alpha), \tag{3.14}$$

$$v = -u_0\sin\lambda\sin\alpha. \tag{3.15}$$

For the TIG, model the wind field is given by

$$\psi = -au_0(\sin\theta\cos\alpha - \cos\lambda\cos\theta\sin\alpha), \tag{3.16}$$

$$\eta = (\frac{2u_0}{a} + 2\Omega)(-\cos\lambda\cos\theta\sin\alpha + \sin\theta\cos\alpha). \tag{3.17}$$

Since the initial condition is nondivergent, $\chi = 0$ and $\delta = 0$. The geopotential field is given by

$$gh = gh_0 - \left(a\Omega u_0 + \frac{u_o^2}{2}\right)(-\cos\lambda\cos\theta\sin\alpha + \sin\theta\cos\alpha)^2. \tag{3.18}$$

3.4.2 Numerical Results

Running the AL model with α different than zero would have required considerable model modification. So, it was run only with $\alpha = 0$. The height error is shown in Fig. 3.18. These plots are produced with equations (3.3), (3.4) and (3.5). The height field after 5 days is shown in Fig. 3.19.

Fig. 3.20 shows the height field error for TIG-2562 model. Again, alpha=1 denotes $\alpha = 0.0$, alpha=2 denotes $\alpha = 0.05$, alpha=3 denotes $\alpha = \pi/2 - 0.05$ and alpha=4 denotes $\alpha = \pi/2$. The l_∞-norm error is a little higher for the case of $\alpha = \pi/2 - 0.05$ and $\alpha = \pi/2$, but looks smoother. However, the l_1 and l_2-norms are about the same for all four values of α. Fig. 3.21 shows the height field error for the NCAR spectral model.

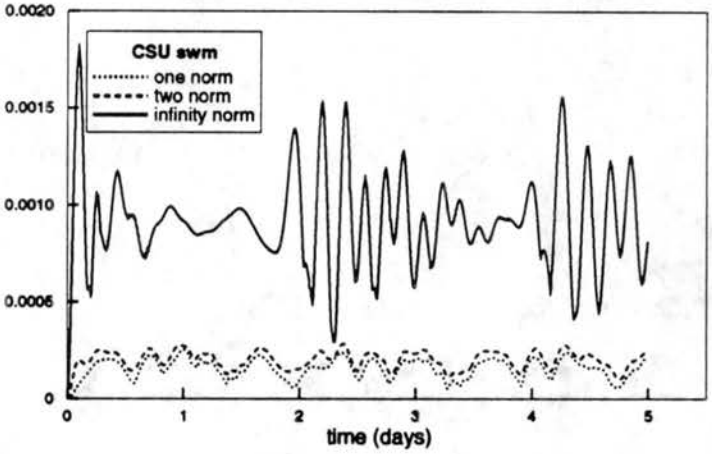

FIGURE 3.18: Test case 2. Height error AL shallow water model. α = 0.0.

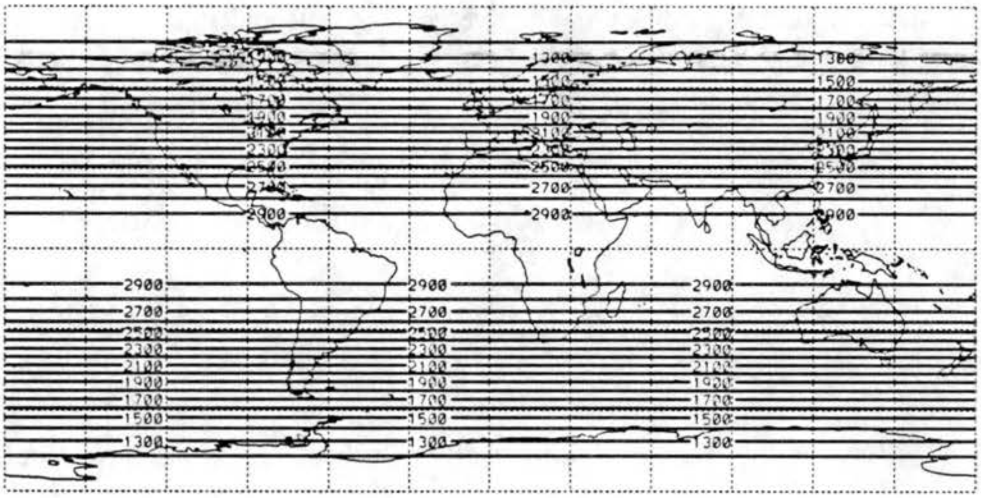

FIGURE 3.19: Test case 2. Height field (m) after 5 days of AL shallow water model. Contour interval 100 m.

In this case, the initial conditions have at exact spectral representation, so the numerical results are somewhat misleading.

Fig. 3.22 shows the simulated height field after 5 days for the four values of α; they all look pretty similar. These plots use cylindrical equidistant projections centered over the axis of flow. The little ripples in the contour lines result from interpolating the TIG grid to the plotting grid. Fig. 3.23 shows the analytic height minus the numeric height.

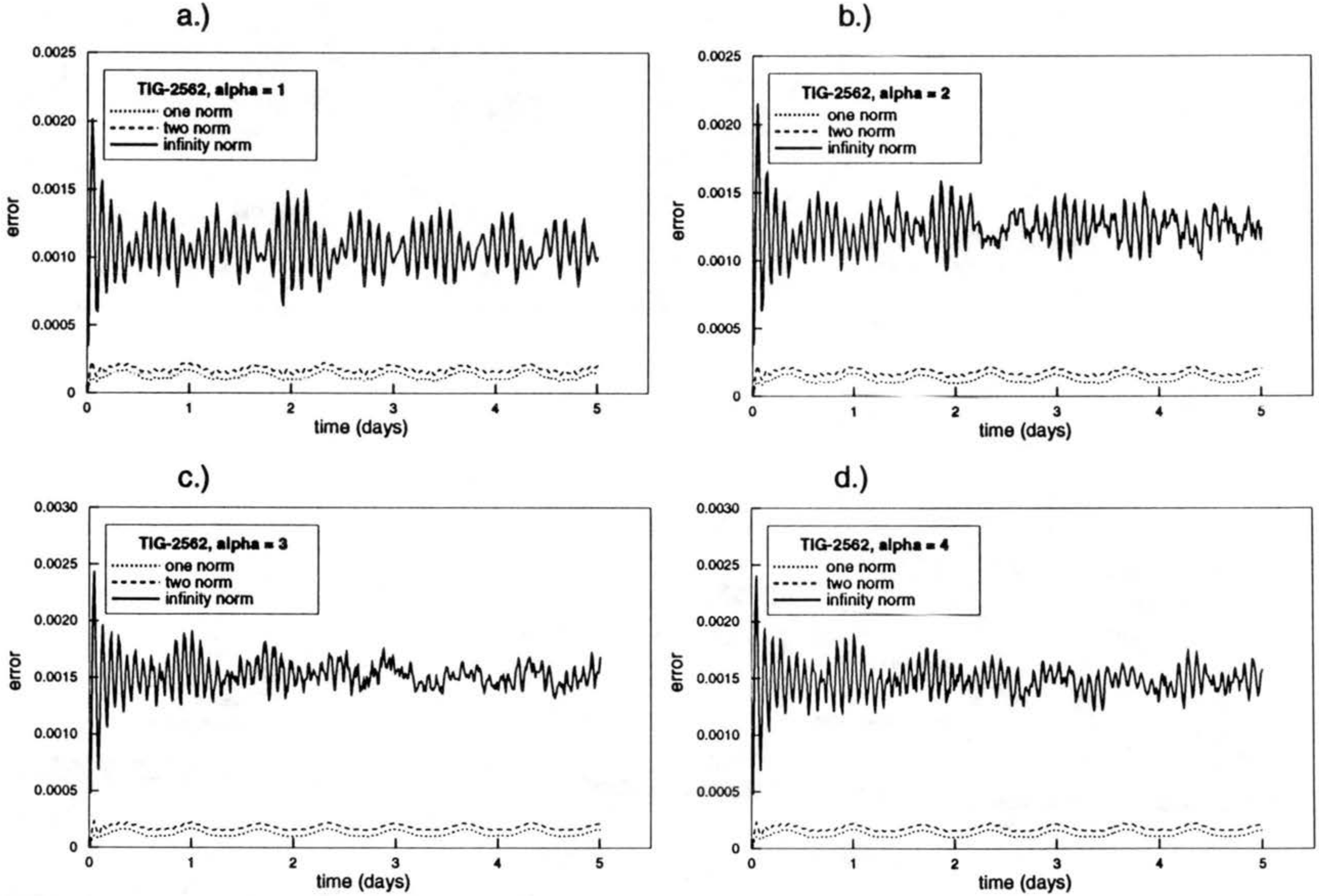

FIGURE 3.20: Test case 2. Height error for TIG-2562. a.) a = 0.0. b.) a = 0.05. c.) a = $\pi/2$ - 0.05. d.) a = $\pi/2$.

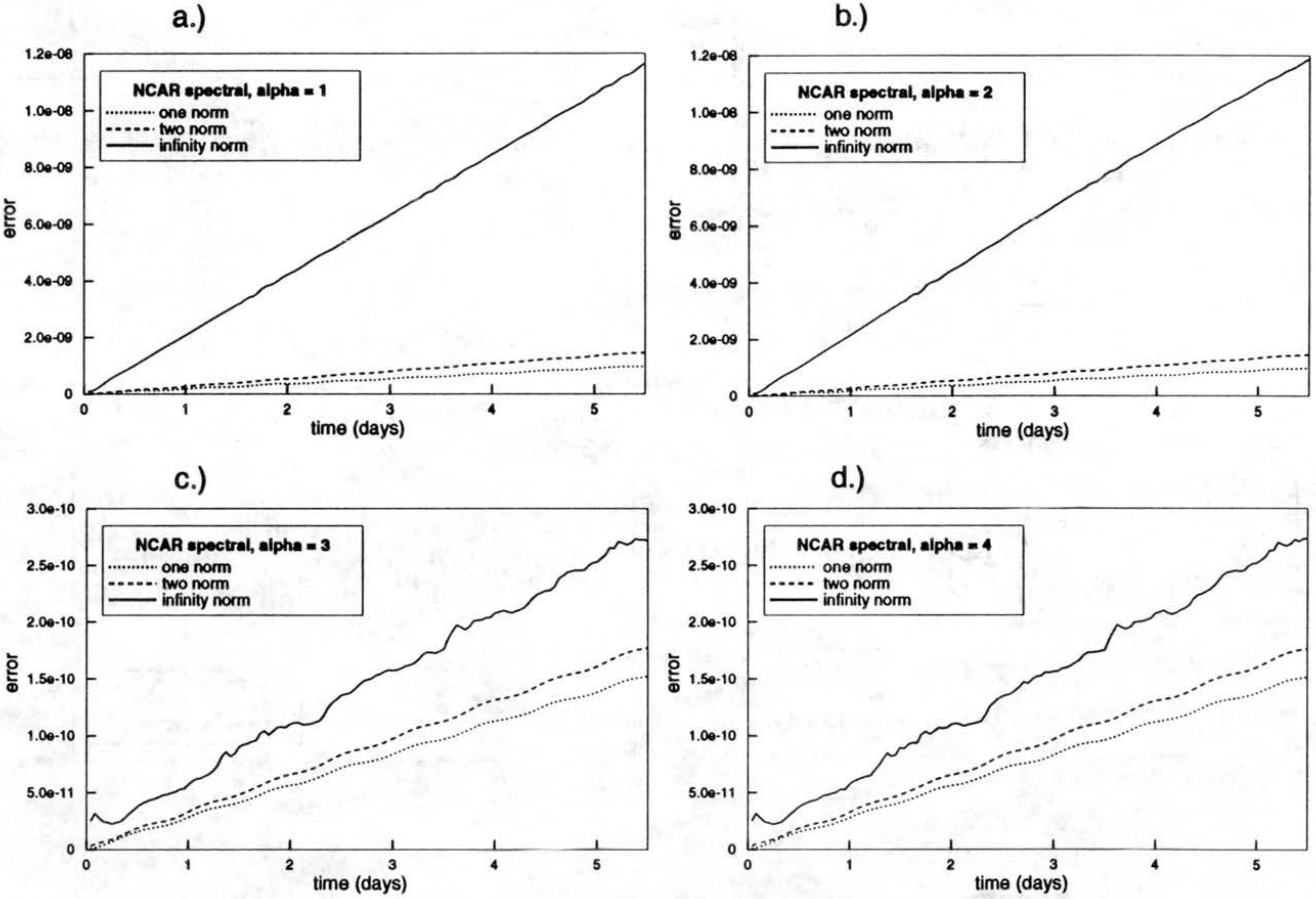

FIGURE 3.21: Test case 2. Height error for NCAR spectral. a.) $\alpha = 0.0$. b.) $\alpha = 0.05$. c.) $\alpha = \pi/2 - 0.05$. d.) $\alpha = \pi/2$.

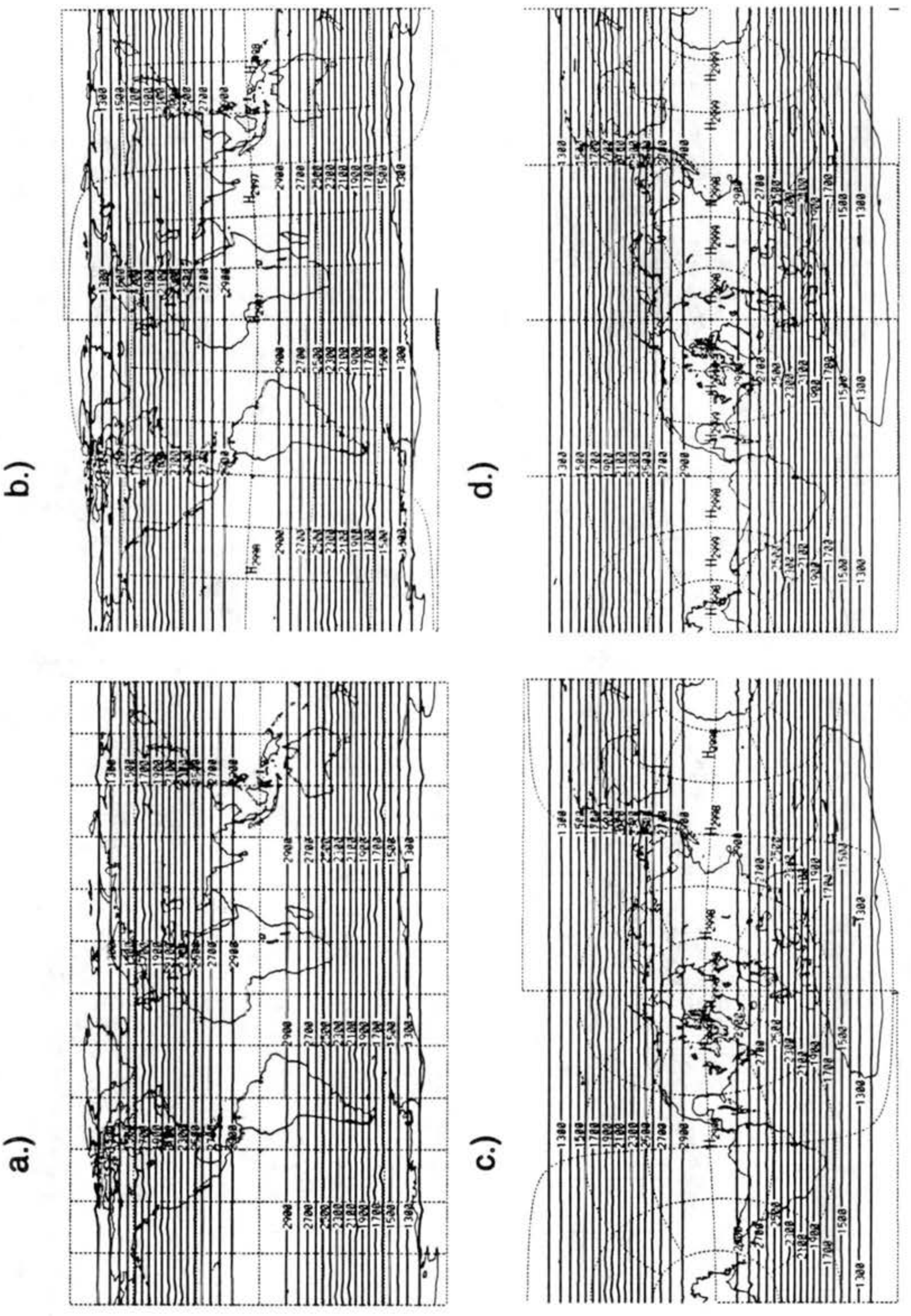

FIGURE 3.22: Test case 2. TIG-2562 model. Height field (m) after 5 days. 100 m contour interval. a.) $\alpha = 0.0$. b.) $\alpha = 0.05$. c.) $\alpha = \pi/2 - 0.05$. d.) $\alpha = \pi/2$.

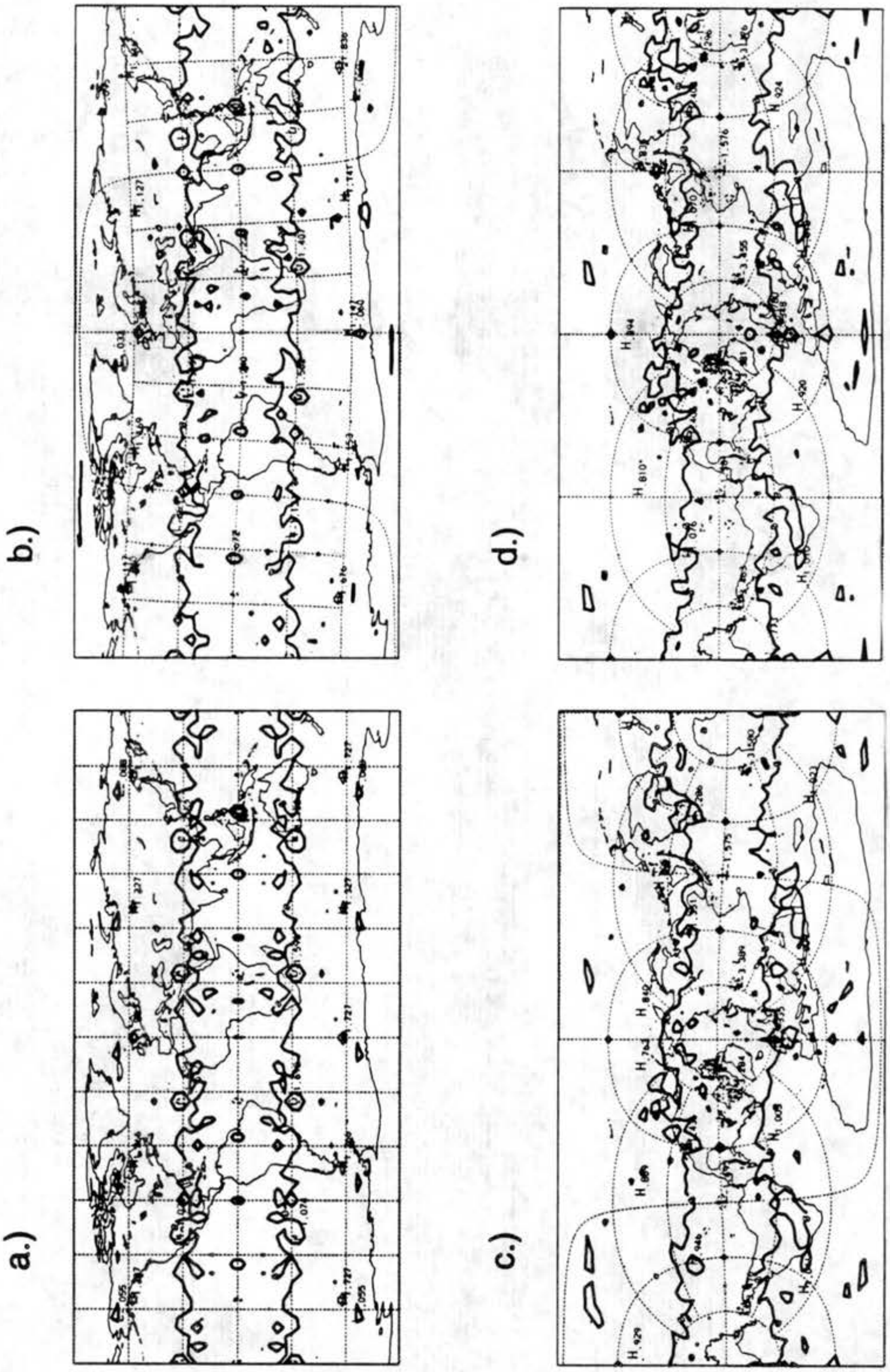

FIGURE 3.23: Test case 2. TIG-2562 model. Analytic height field (m) minus simulated height field (m) after 5 days. 1 m contour interval. a.) $\alpha = 0.0$. b.) $\alpha = 0.05$. c.) $\alpha = \pi/2 - 0.05$. d.) $\alpha = \pi/2$.

There is some wavenumber 5 tendencies in the difference at the latitude of the grid pentagons, and some wavenumber 10 tendencies along the equator. These are probably due to the structure of the grid. This point is discussed further in test case 3.

3.5 Test Case 3: Steady State Nonlinear Geostrophic Flow with Compact Support.

3.5.1 Description

This test case was introduced by Browning (1989) in an attempt to formulate a test case which more closely resembles the real atmosphere. Here, the wind is purely zonal, like test case 2, but nonzero only in a range of latitudes. It is intended to simulate the observed u component of the wind at about 10 km. Again, as the integration progresses, nothing should change.

Consider the infinitely differentiable function $b(x)$ defined by

$$b(x) = \{ \begin{array}{ll} 0 & \text{if } x \le 0 \\ e^{-1/x} & \text{if } x > 0. \end{array} \tag{3.19}$$

For this test case, the initial conditions for wind in terms of u and v are given by

$$u(x) = u_0 b(x) b(x_e - x) e^{4/x_e} \tag{3.20}$$

$$v = 0, \tag{3.21}$$

where $x(\theta) = x_e (\theta - \theta_b)(\theta_e - \theta_b)^{-1}$, $u_0 = (2\pi a)/(12 \text{ days})$, $\theta_b = -\pi/6$, $\theta_e = \pi/2$ and $x_e = 0.3$. The function $x(\theta)$ maps the interval $\theta_b \le \theta \le \theta_e$ onto the interval $0 \le x \le x_e$, so u is nonzero only on $\theta_b \le \theta \le \theta_e$. See Fig. 3.24.

Integrating (3.20) and (3.21) gives wind in terms of stream function and velocity potential:

$$\psi(x) = -a\frac{\theta_e - \theta_b}{x_e}u_0 e^{4/x_e}\int_{x_e}^{x} e^{-x_e[\xi(x_e-\xi)]^{-1}}d\xi, \tag{3.22}$$

$$\chi = 0. \tag{3.23}$$

The profile of initial u and ψ are shown in Fig. 3.24

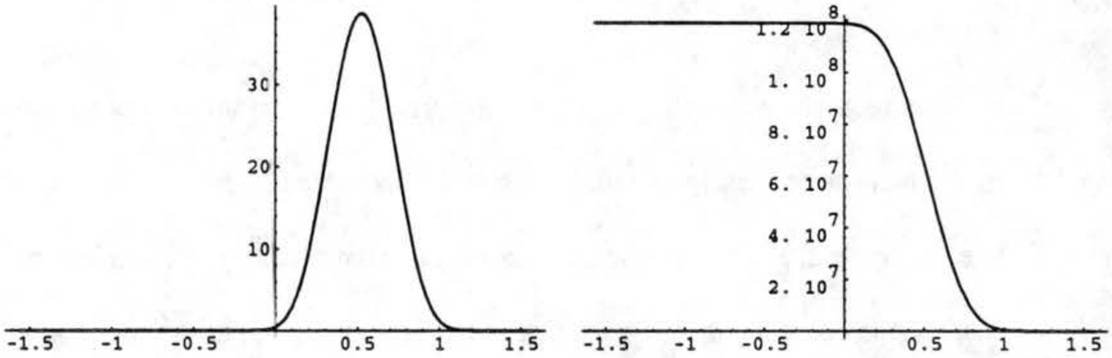

FIGURE 3.24: Initial profiles of zonal wind and stream function.

To maintain steady state, the height must satisfy

$$\frac{u^2\tan\theta}{a} + \frac{g}{a}\frac{\partial h}{\partial\theta} + fu = 0, \tag{3.24}$$

which is obtained by substituting (3.20) and (3.21) into the equation for the v wind component in spherical coordinates. Integrating (3.24) from the South Pole to some arbitrary latitude gives

$$h(\theta) = h_0 - \frac{a}{g}\int_{-\pi/2}^{\theta}\left(2\Omega\sin\tau + \frac{u(\tau)\tan\tau}{a}\right)u(\tau)\,d\tau. \tag{3.25}$$

Mathematica was used to numerically integrate (3.22) and (3.25) to obtain the initial conditions for this test case.

To this point, the axis of the Earth's rotation and the axis through the poles of the numeric grid have been considered coincident. If the axis of the grid is tilted, we can study the effect of directing the jet over a pole of the grid. We define two spherical coordinate systems, one in which the axis through the poles is coincident to the axis of the earth's rotation, and one in which the axis is coincident with the polar axis of the numerical grid. Denote a position measured in the second, tilted coordinate system as (θ', λ'). Let the angle between the two coordinate system's axes be α. For this test case, we will set α to be $\alpha = 0.0$ and $\alpha = \pi/3$. For a given grid point, (θ', λ'), in the tilted coordinate system, we need to compute its corresponding latitude in the untilted coordinate system. The two systems are related by

$$\sin\theta = \sin\theta'\cos\alpha - \cos\theta'\cos\lambda'\sin\alpha, \tag{3.26}$$

$$\sin\lambda\cos\theta = \sin\lambda'\cos\theta'. \tag{3.27}$$

So, for example, the Coriolis parameter $f = 2\Omega\sin\theta$ at each grid point is given by

$$f = 2\Omega(\sin\theta'\cos\alpha - \cos\theta'\cos\lambda'\sin\alpha). \tag{3.28}$$

Similarly, we can use (3.26) to compute the initial wind from (3.20) and initial height from (3.25) at each grid point.

3.5.2 Numerical Results

The height errors for the NCAR spectral, the Arakawa-Lamb model and TIG-2562 model with $\alpha = 0.0$ are shown in Fig. 3.25, and the height errors for $\alpha = \pi/3$ are

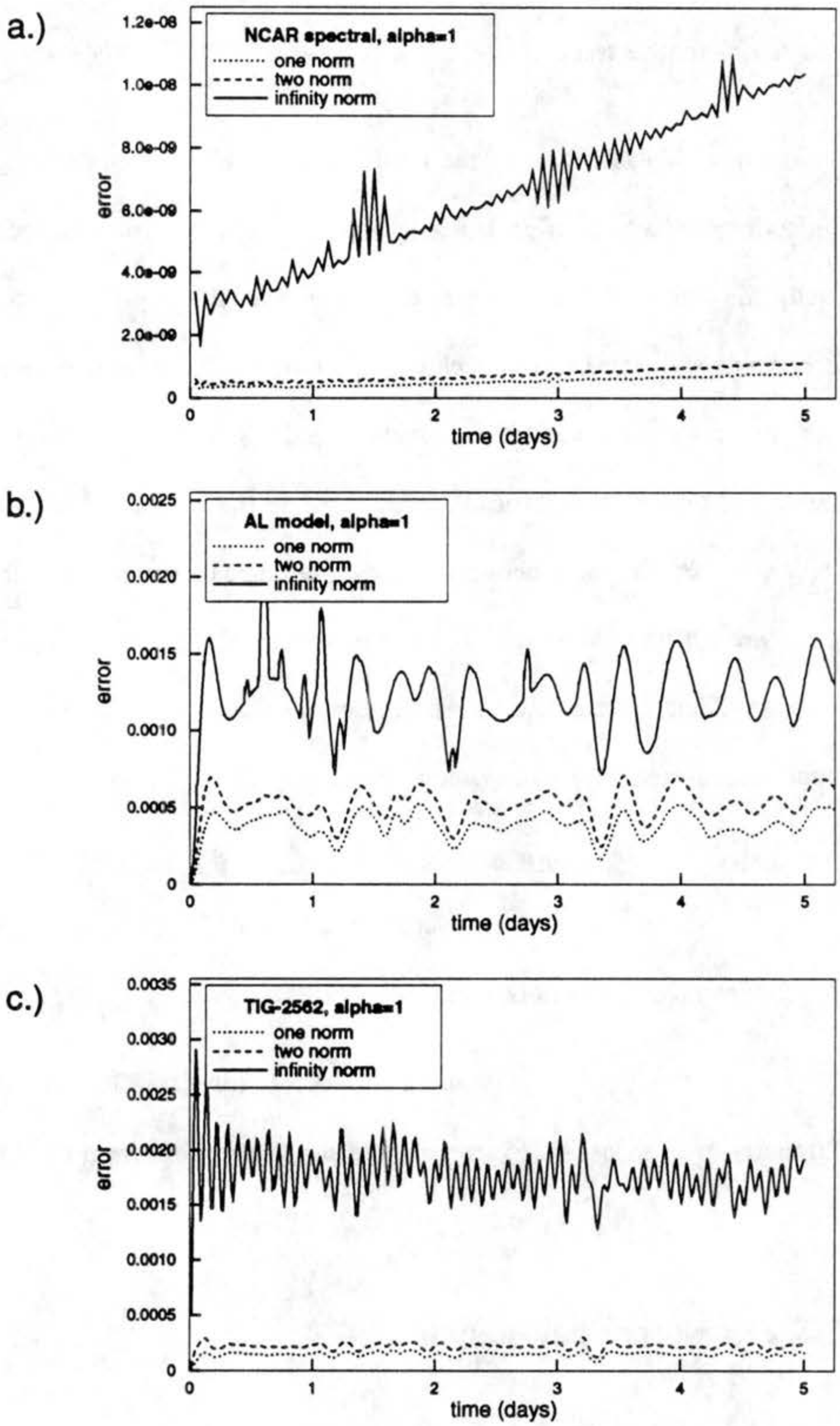

FIGURE 3.25: Test case 3. Height errors. $\alpha = 0.0$. a.) NCAR spectral. b.) Arakawa-Lamb shallow water model. c.) TIG-2562 model

shown in Fig. 3.26 Again, the AL model is run only with $\alpha = 0$. Now, in the plots,

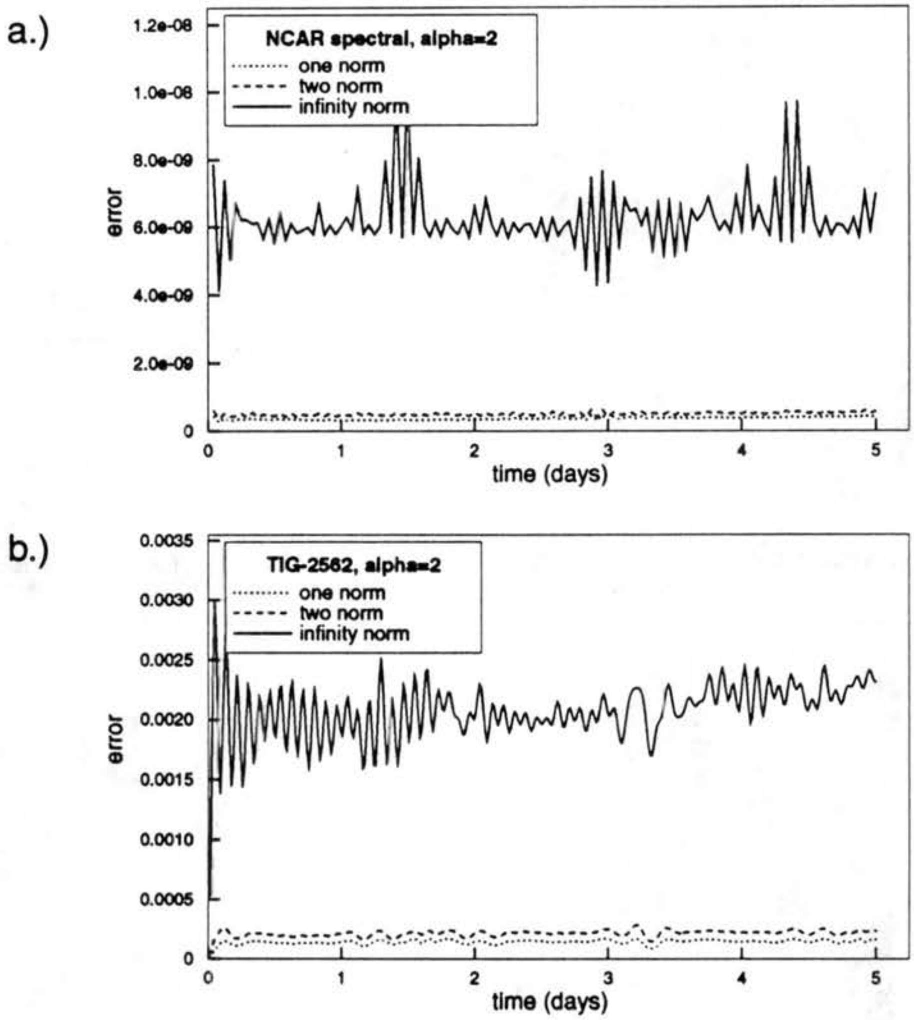

FIGURE 3.26: Test case 3 height errors. $\alpha = \pi/3$. a.) NCAR spectral. b.) TIG-2562 model

alpha=1 denotes $\alpha = 0.0$ and alpha=2 denotes $\alpha = \pi/3$.

The numeric height field and analytic minus numeric height fields for the Arakawa-Lamb model are shown in Fig. 3.27. The numeric height field and analytic height field minus numeric height field for the TIG-2562 model with $\alpha = 0.0$ are shown in Fig. 3.28. We can see the biggest differences occur an the latitude where the jet is strongest. There is a definite wavenumber 5 pattern to the difference. Fig. 3.29 shows the cell area

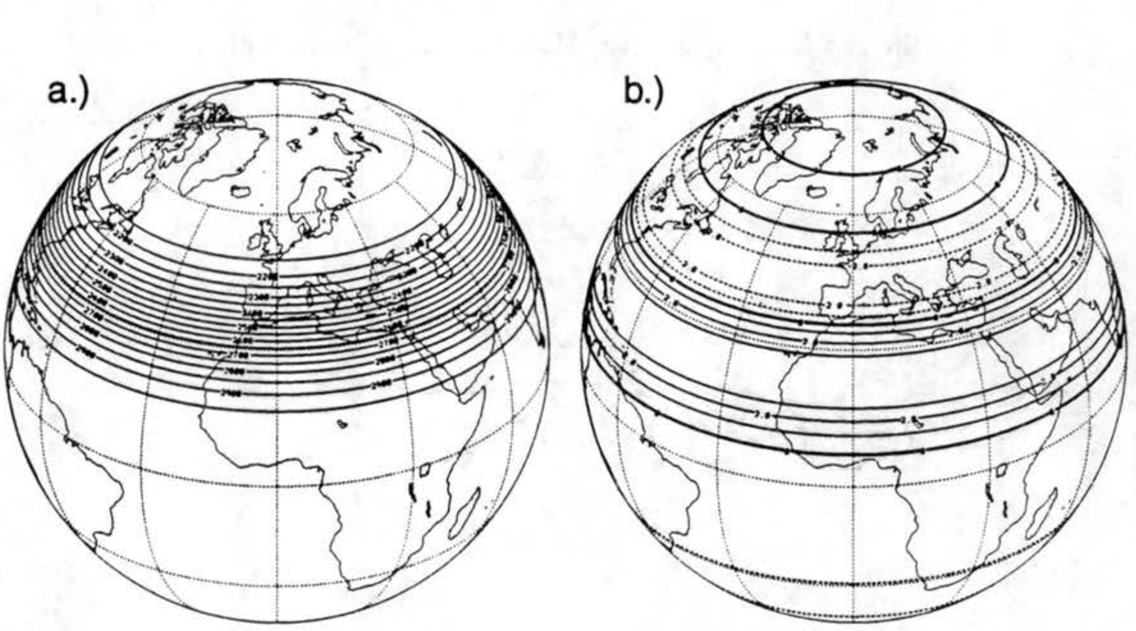

FIGURE 3.27: Test case 3. Arakawa-Lamb shallow water model after five days. $\alpha = 0.0$. a.) Numeric height field. 100 m contour interval. b.) Analytic height field minus numeric height field. 1 m contour interval.

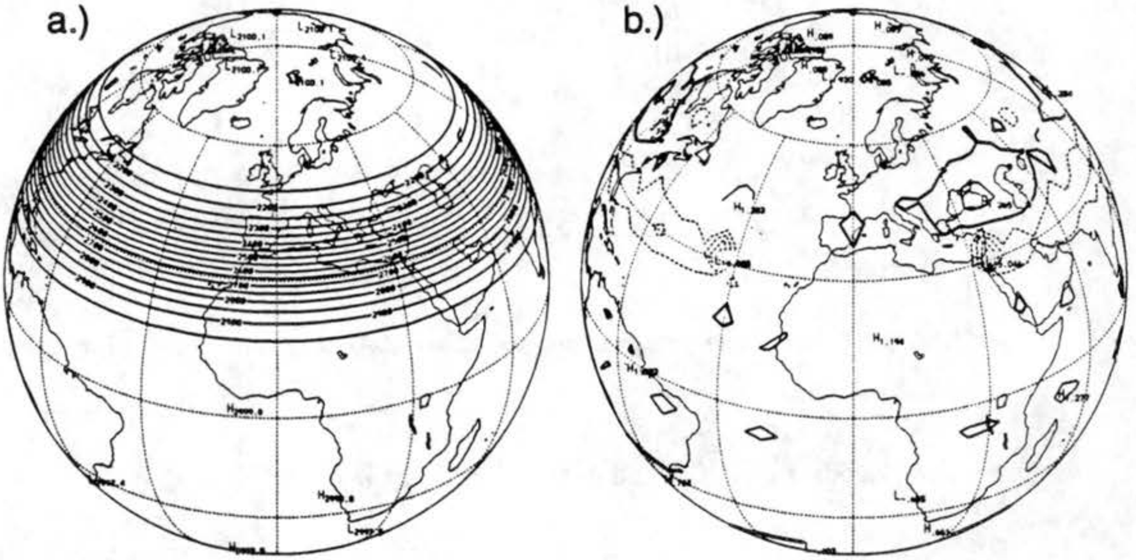

FIGURE 3.28: Test case 3. TIG-2562 model after five days. $\alpha = 0.0$. a.) Numeric height field. 100 m contour interval. b.) Analytic height field minus numeric height field. 1 m contour interval.

divided by the average cell area for the grid with 2562 cells. The grid pentagons are at ±26.6° latitude, and 0°, 72°, 144°, 216° and 288° longitude. These are the smallest cells on the grid. The biggest error is at the saddle points of the area contour, where the

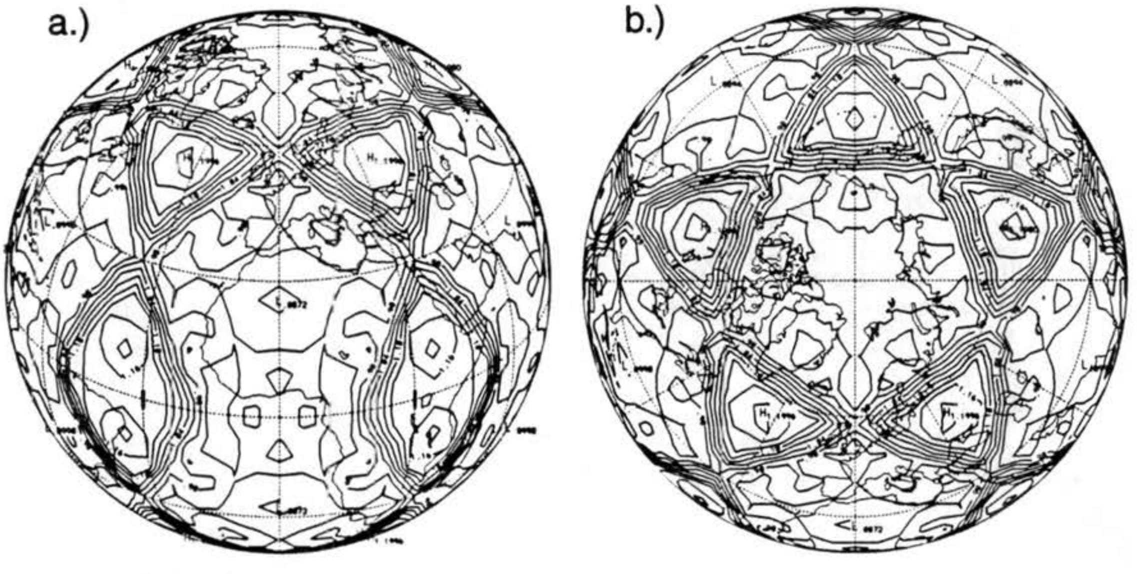

FIGURE 3.29: Test case 3. Contour plot of cell area divided by average cell area for twisted icosahedral grid with 2562 cells.

gradient of area is large. This result suggests the need to construct the grid so that cell area varies more continuously. The numeric height field and analytic height field minus numeric height field for the TIG-2562 model with $\alpha = \pi/3$ are shown in Fig. 3.30. There appears to be less of a correspondence between the position of errors and the area

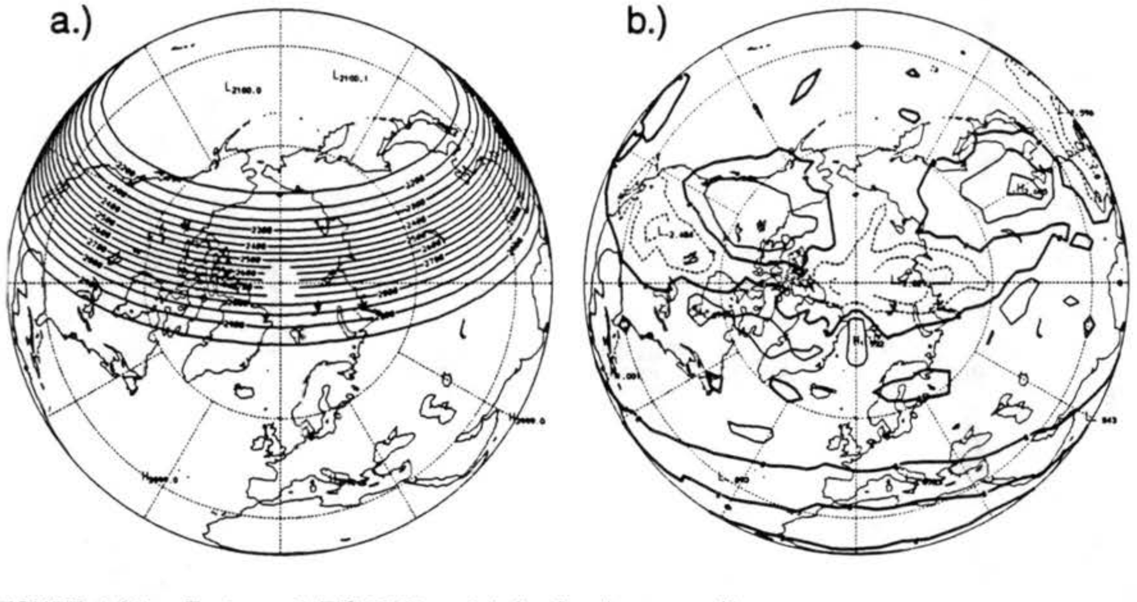

FIGURE 3.30: Test case 3. TIG-2562 model after five days. $\alpha = \pi/3$.
a.) Numeric height field. 100 m contour interval. b.) Analytic height field minus numeric height field. 1 m contour interval.

distribution of the grid when $\alpha = \pi/3$. In this case, the symmetries of the grid and fluid flow are not as well aligned.

Fig. 3.31 shows the conservation of total energy and potential enstrophy based on equation (3.6) for the Arakawa-Lamb model. All three models conserve mass exactly, so these plots are not shown. Fig. 3.32 and Fig. 3.33 show the conservation of total energy, potential enstrophy, globally integrated vorticity and globally integrated divergence for the NCAR spectral and TIG-2562 models with $\alpha = 0.0$. Note that the integrated vorticity and divergence have been multiplied by 10^{12}.

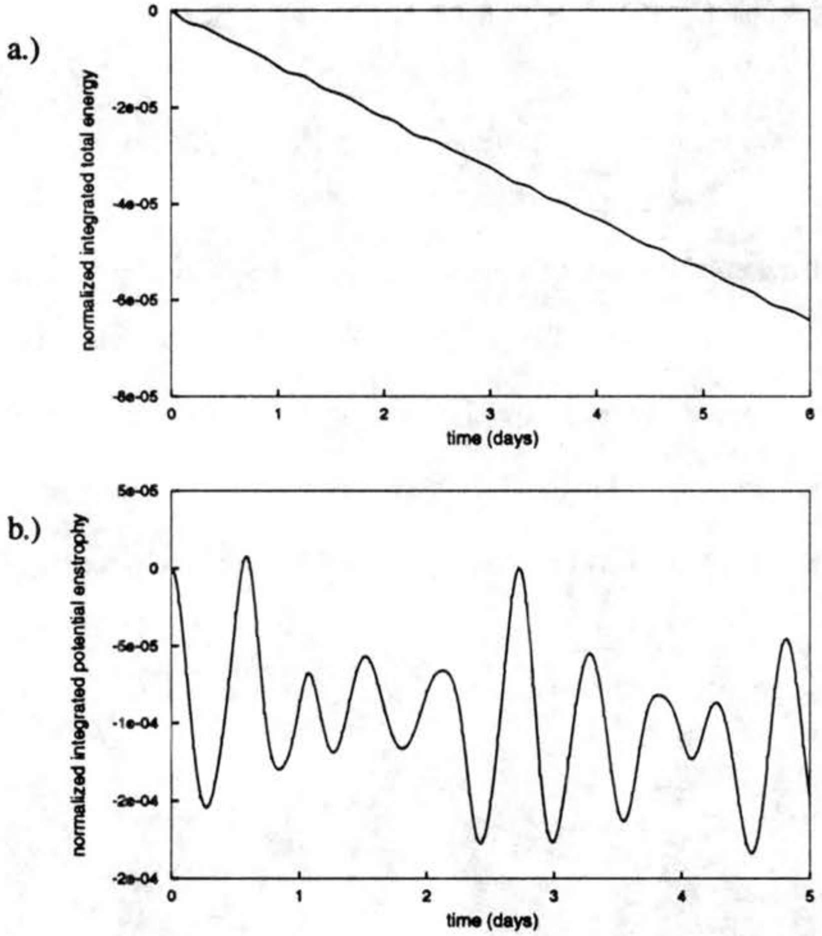

FIGURE 3.31: Test case 3. Arakawa-Lamb model. a.) Normalized integrated total energy. b.) Normalized integrated potential enstrophy.

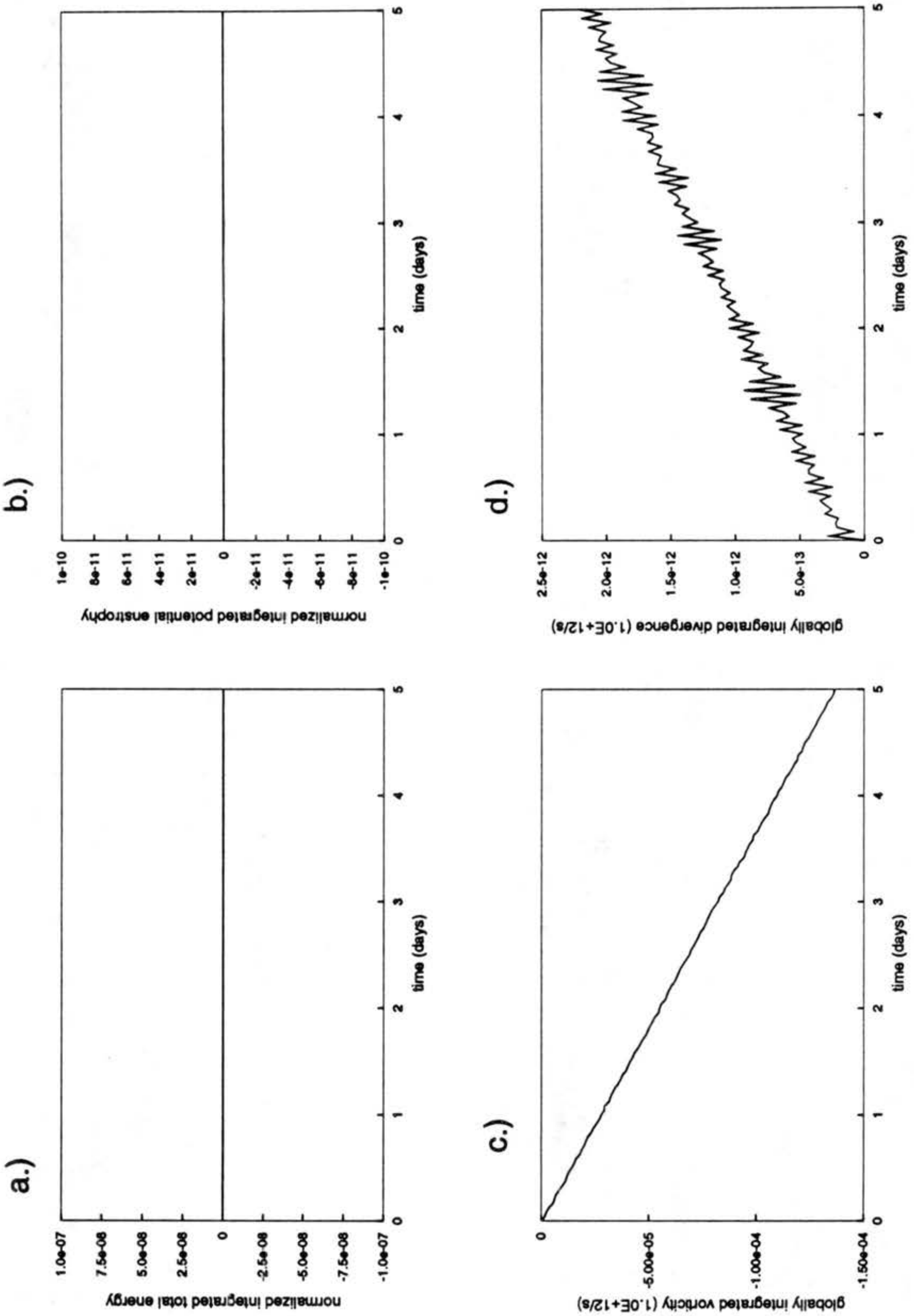

FIGURE 3.32: Test case 3. NCAR spectral model. a.) Normalized integrated total energy. b.) Normalized integrated potential enstrophy. c.) Integrated vorticity. d.) Integrated divergence.

Test Case 3: Steady State Nonlinear Geostrophic Flow with Compact Support.

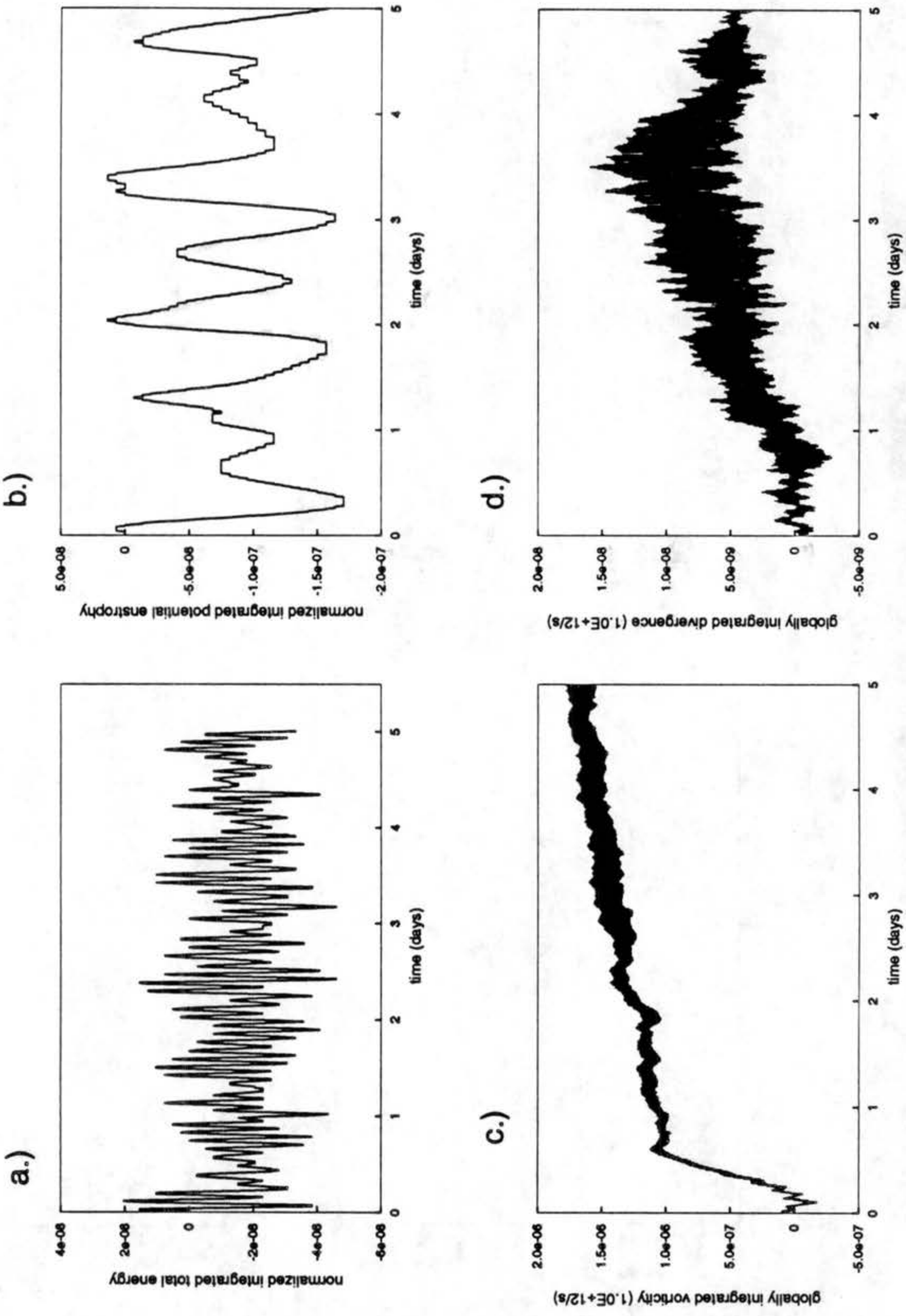

FIGURE 3.33: Test case 3. TIG-2562 model. α=0.0. a.) Normalized integrated total energy. b.) Normalized integrated potential enstrophy. c.) Integrated vorticity. d.) Integrated divergence.

3.6 Test Case 4: Forced Nonlinear System with a Translating Low

3.6.1 Description

Unlike the previous steady state nonlinear test cases, there are few examples of analytic solutions to the unsteady nonlinear equations. However, by including forcing terms in the shallow water equation, we can test the models in an unsteady nonlinear setting in which we know an analytic solution. In this case the forcing terms are designed so that the equations simulate a short-wave trough imbedded in a middle level tropospheric jet. It was used by Browning (1989) to test the composite mesh model described in chapter 1.

The shallow water equations written in advective form in spherical coordinates with forcing terms, F_u, F_v and F_h are given by

$$\frac{du}{dt} - \frac{uv\tan\theta}{a} + \frac{g}{a\cos\theta}\frac{\partial h}{\partial\lambda} - fv = F_u, \tag{3.29}$$

$$\frac{dv}{dt} + \frac{uu\tan\theta}{a} + \frac{g}{a}\frac{\partial h}{\partial\theta} + fu = F_v, \tag{3.30}$$

$$\frac{dh}{dt} + \frac{h}{a\cos\theta}\left[\frac{\partial u}{\partial\lambda} + \frac{\partial}{\partial\theta}(v\cos\theta)\right] = F_h, \tag{3.31}$$

where we have assumed $h = h^*$. By substituting the equations for the desired flow into (3.29), (3.30) and (3.31), we can construct the forcing term as follows:

$$F_u = \frac{d\hat{u}}{dt} - \frac{\hat{u}\hat{v}\tan\theta}{a} + \frac{g}{a\cos\theta}\frac{\partial\hat{h}}{\partial\lambda} - f\hat{v}, \tag{3.32}$$

$$F_v = \frac{d\hat{v}}{dt} + \frac{\hat{u}\hat{u}\tan\theta}{a} + \frac{g}{a}\frac{\partial\hat{h}}{\partial\theta}f\hat{u}, \tag{3.33}$$

$$F_h = \frac{d\hat{h}}{dt} + \frac{\hat{h}}{a\cos\theta}\left[\frac{\partial\hat{u}}{\partial\lambda} + \frac{\partial}{\partial\theta}(\hat{v}\cos\theta)\right]. \tag{3.34}$$

We take the curl and divergence of (3.32) and (3.33), to obtain the forcing terms for the vorticity and divergence equations, F_η and F_δ. The flow is given by

$$\hat{u} = \bar{u} - \frac{\overline{\psi}_\theta}{a}, \tag{3.35}$$

$$\hat{v} = \frac{\overline{\psi}_\lambda}{a\cos\theta}, \tag{3.36}$$

$$g\hat{h} = g\bar{h} + f\overline{\psi}, \tag{3.37}$$

where

$$\bar{u} = u_0\sin^{14}(2\theta), \tag{3.38}$$

$$g\hat{h} = gh_0 - \int_{-\pi/2}^{\theta}[af(\tau) + \bar{u}(\tau)\tan\tau]\,\bar{u}(\tau)\,d\tau, \tag{3.39}$$

$$\overline{\psi} = \psi_0 e^{-\sigma[(1-C)/(1+C)]}, \tag{3.40}$$

where

$$C = \sin\theta_0\sin\theta + \cos\theta_0\cos\theta\cos\left(\lambda - \frac{u_0}{a}t - \lambda_0\right). \tag{3.41}$$

In the above equations, we set $\psi_0 = -0.03\,(gh_0/f_0)$, $\sigma = (12.74244)^2$, $gh_0 = 10^5\ \mathrm{m^2/s^2}$, $f_0 = 2\Omega\sin(\pi/4)$ and $u_0 = 20\ \mathrm{m/s}$. The center of the low is initially at $(\theta_0, \lambda_0) = (\pi/4, 0)$.

We wrote a *Mathematica* program to symbolically perform the calculus in equations (3.32) to (3.41) and produce FORTRAN code for F_u and F_v, and C code for F_η and F_δ. These ends up being rather complicated process and a good source for coding errors. The initial height field in shown in Fig. 3.34.

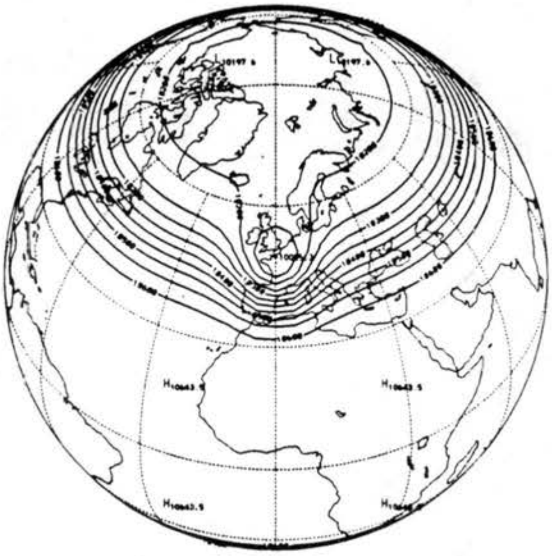

FIGURE 3.34: Test case 4. Initial height field. 50 m contour interval.

3.6.2 Numerical Results

The model results were never very good. The pattern shown in Fig. 3.34 should be advected along without changing shape at about 22° per day; however, it travels much slower than this in the models. It moves about 60° in four days in the AL model and about 45° in four days in the TIG model. Also, the low weakens considerably, more so in the AL model than in the TIG model. Fig. 3.35 shows the evolution of the Arakawa-Lamb model for four days, and Fig. 3.36 shows the same thing for the TIG-2562 model.

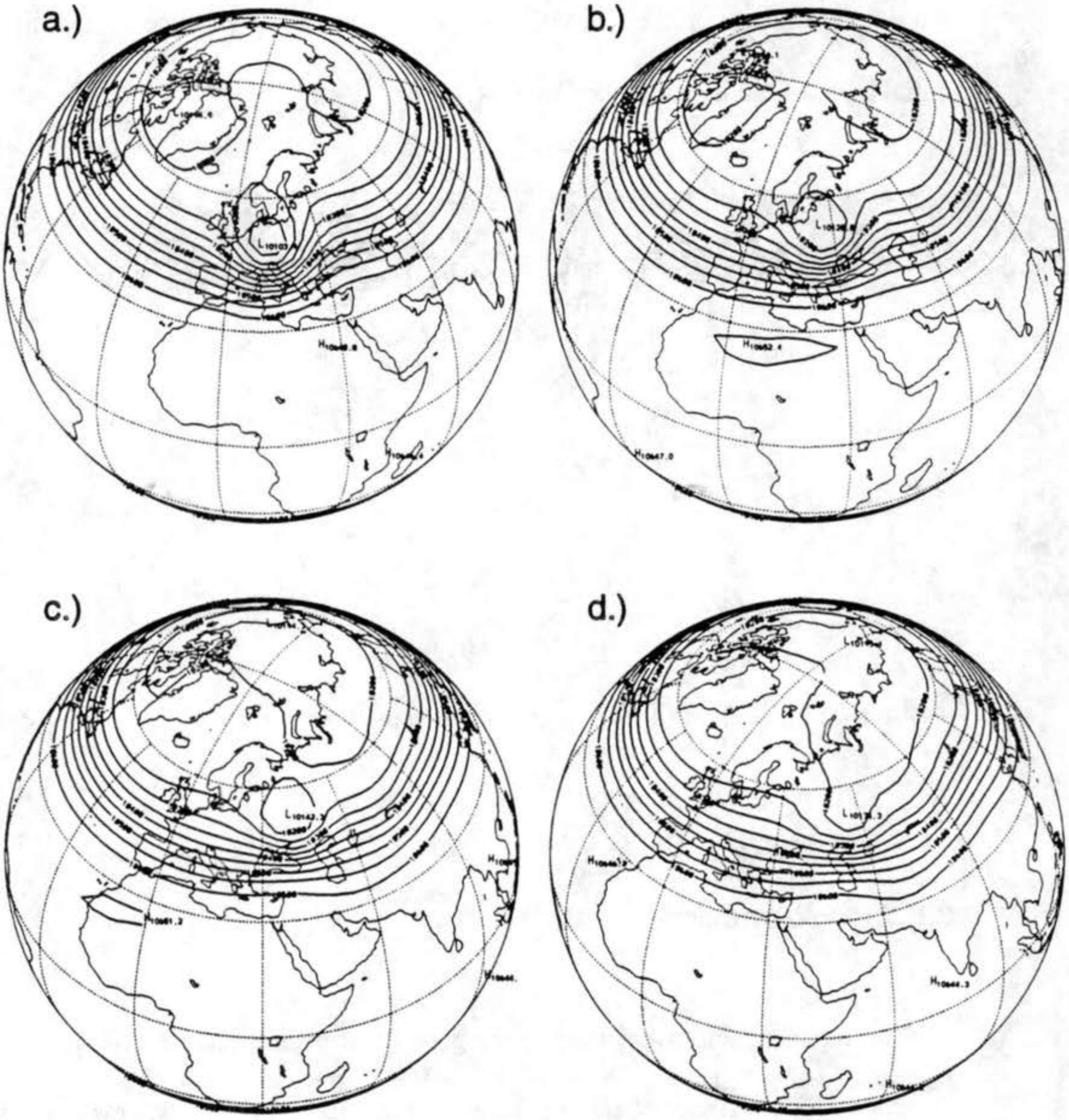

FIGURE 3.35: Test case 4. Height field for Arakawa-Lamb model. 50 m contour interval. a.) Day 1. b.) Day 2. c.) Day 3. d.) Day 4.

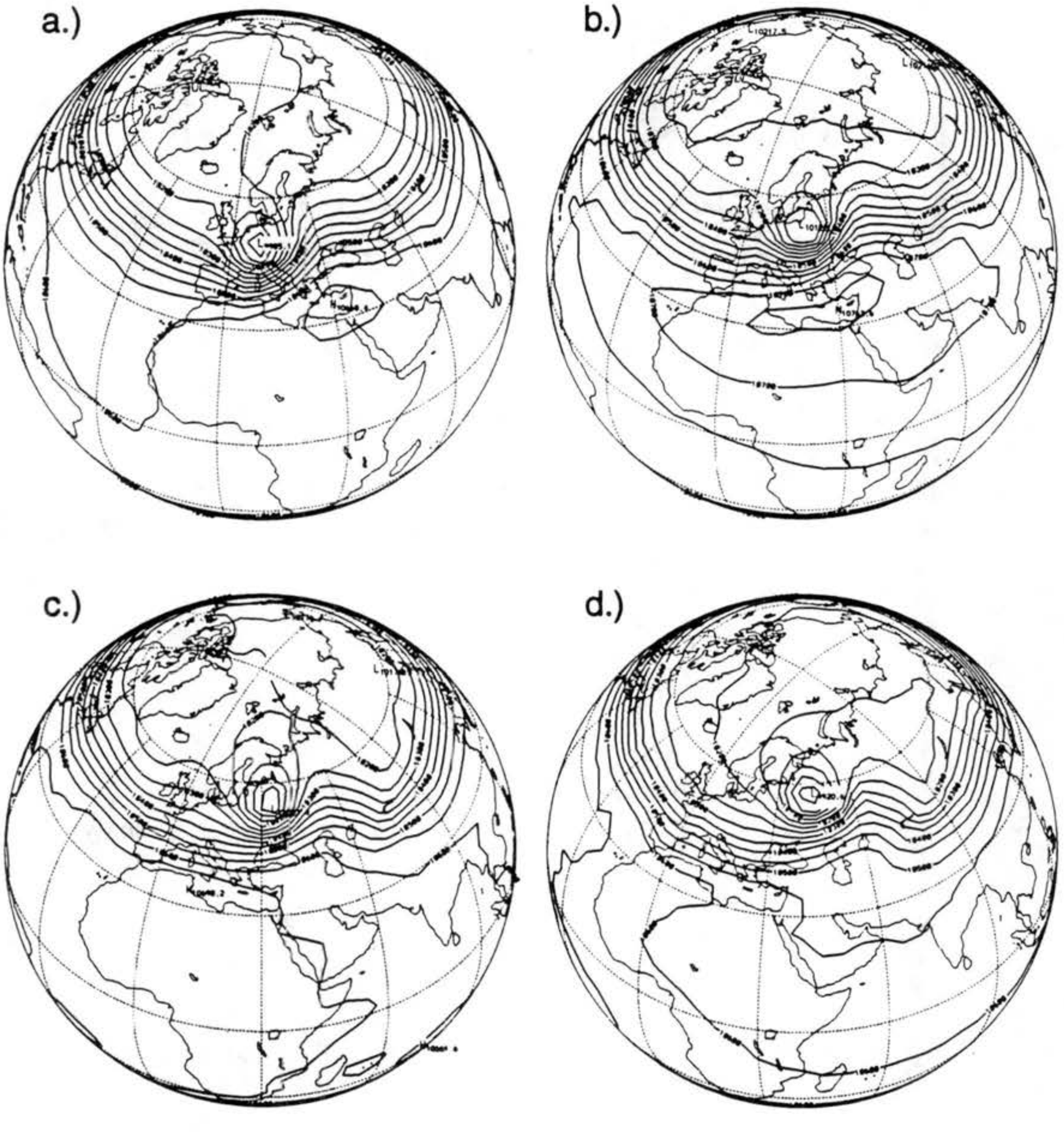

FIGURE 3.36: Test case 4. Height field for TIG-2562 model. 50 m contour interval. a.) Day 1. b.) Day 2. c.) Day 3. d.) Day 4.

3.7 Test Case 5: Zonal Flow over an Isolated Mountain.

3.7.1 Description

This is only test case in the suite with topography. In this test zonal flow is directed over a mountain, a hemispherical bump, centered at 30 °N latitude and 40 °W longitude. While it does not directly address the pole problem, it does test each model's ability to handle topography. This set of initial conditions was first used by Takacs (1988) to study the effect of a posteriori method for conservation of integral invariants.

The initial wind is nondivergent and zonal as in test case 2. The components of the wind are given by

$$u = u_0 \cos\theta, \tag{3.42}$$

$$v = 0, \tag{3.43}$$

where $u_0 = 20\ \text{m}\ \text{s}^{-1}$. The initial height of the free surface is given by

$$h = h_0 - \frac{1}{g}\left(a\Omega u_0 + \frac{u_0^2}{2}\right)\sin^2\theta, \tag{3.44}$$

where $h_0 = 5960\ \text{m}$. The SFVP representation for the wind is the same as in test case 2.

The surface topography is given by

$$h_s = h_{s0}\left(1 - \frac{r}{R}\right), \tag{3.45}$$

where $h_{s0} = 2000\ \text{m}$, $R = \pi/9$, and

$$r = \min\left[R, \sqrt{(\lambda-\lambda_c)^2-(\theta-\theta_c)^2}\right]. \tag{3.46}$$

The center in the surface bump is at $\theta_c = \pi/6$ and $\lambda_c = 3\pi/2$. A plot of the surface topography is shown in Fig. 3.37

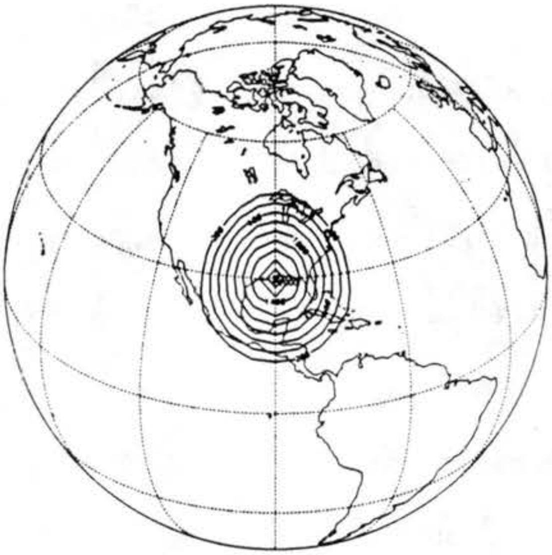

FIGURE 3.37: Test case 5. Surface topography. 200 m contour interval.

3.7.2 Numerical Results

This test case, unlike the previous test cases, does not have an analytic solution. Instead the numerical results are compared to high resolution spectral model results which are considered to be the true solution. The NCAR spectral model was run with T106 resolution, and the spectral coefficients written to a file called REF0114.cdf every 24 hours. The daily spectral coefficient for the u and v components of the wind, geopotential, absolute vorticity and divergence are included in the file in netCDF formate. REF0114.cdf is available from Rudy Jacobs at NCAR. Pieces of the source code for the NCAR spectral model were used to write a program to inverse transform the coefficients at a arbitrary point on the sphere.

Fig. 3.38, Fig. 3.39 and Fig. 3.40 show the height fields of the three models after 5 days, 10 days and 15 days, respectively. The TIG model is run with 2562 cells and 10242 cells. The TIG-10242 model used a 300 second time step, while the other three models were run with a 600 second time step. In each case we can see a trough develop down stream from the mountain. As the fluid flows over the mountain, its speed increases in order to transport the same amount of material per unit time through the shallower depth. The increase in easterly speed of a parcel increases its angular momentum. To conserve angular momentum, the parcel veers turn toward the south where the radius of the Earth is larger. After a parcel has moved over the mountain, its velocity decreases in the increasing depth of fluid. This reduces its eastward component, and decreasing its angular momentum, so the parcel must turn northward to conserve angular momentum.

The u component of the wind is westerly in both the northern and southern hemispheres, so there are no critical latitudes, that is, latitude bands of no zonal flow. Critical latitudes block the propagation of waves across them. For example, on the Earth, the trade winds block the propagation of disturbances across the equator. However, in the model, there are no easterlies and waves from the northern hemisphere can propagate into the southern.

Looking at the plots after 15 days, we see the TIG-2562 model does not build quite as high a ridge down stream from the mountain as the other three models.

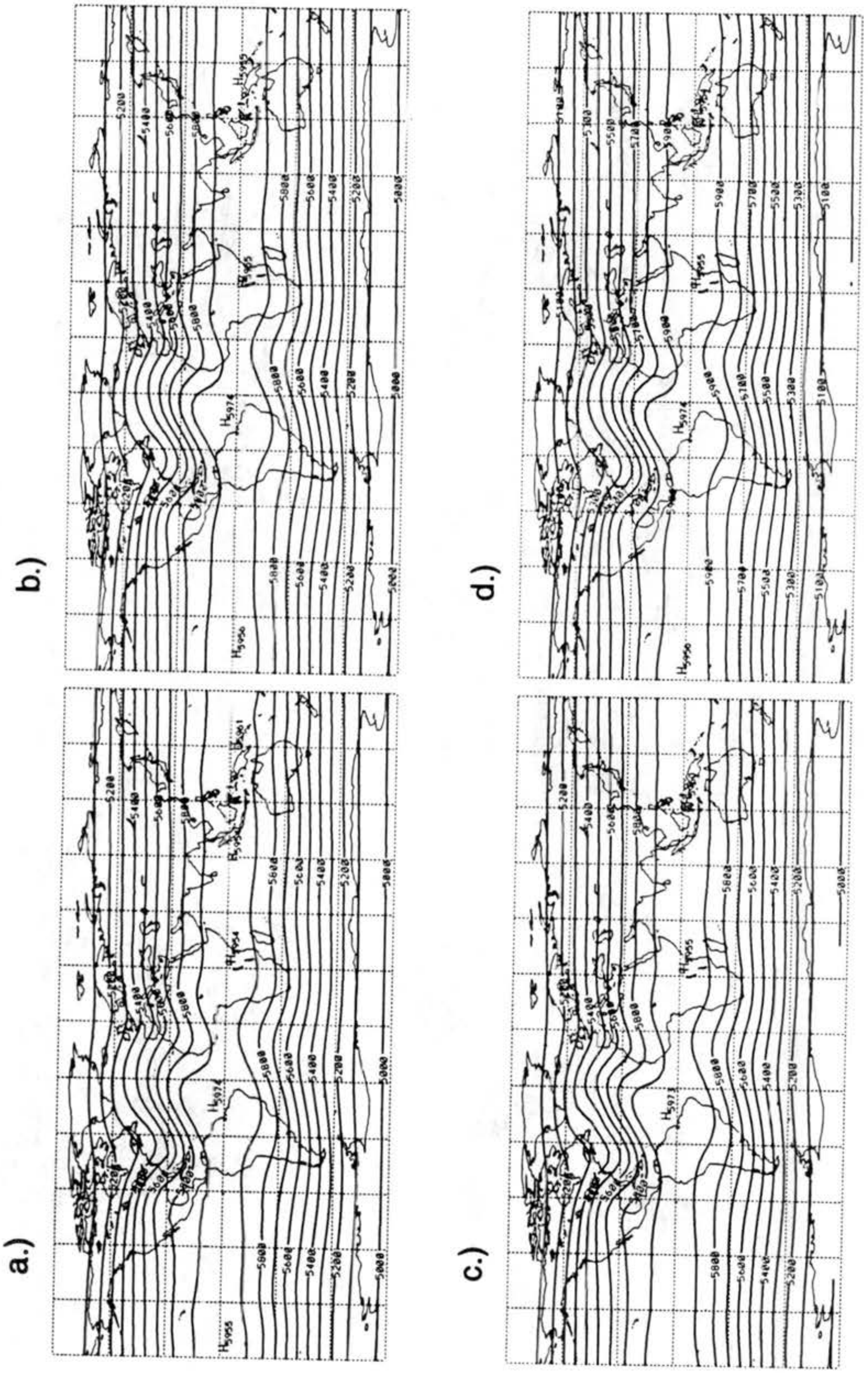

FIGURE 3.38: Test case 5. Height field after 5 days. 100 m contour interval. a.) NCAR T106 model. b.) Arakawa-Lamb model. c.) TIG-2562 model. d.) TIG-10242 model.

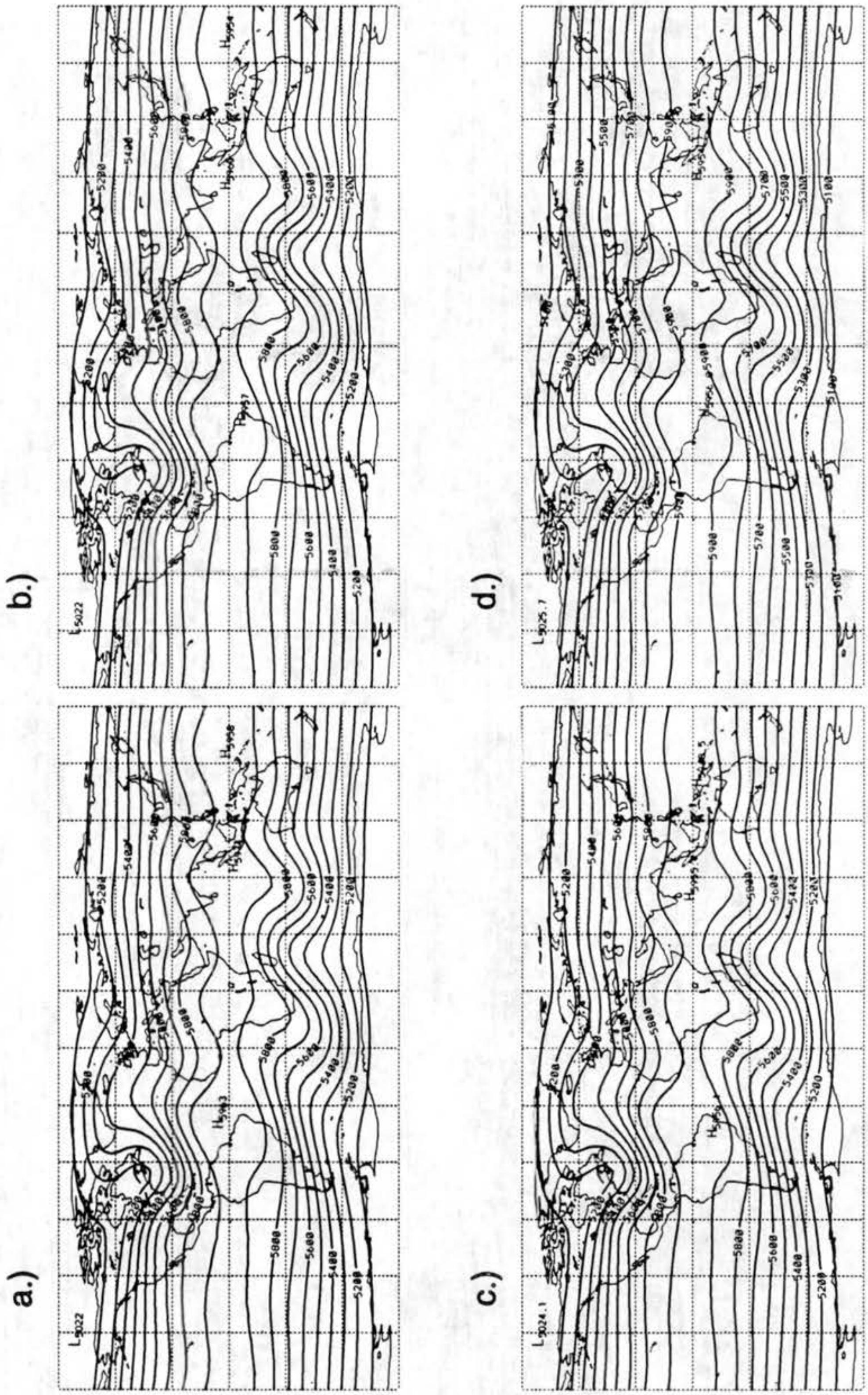

FIGURE 3.39: Test case 5. Height field after 10 days. 100 m contour interval. a.) NCAR T106 model. b.) Arakawa-Lamb model. c.) TIG-2562 model. d.) TIG-10242 model.

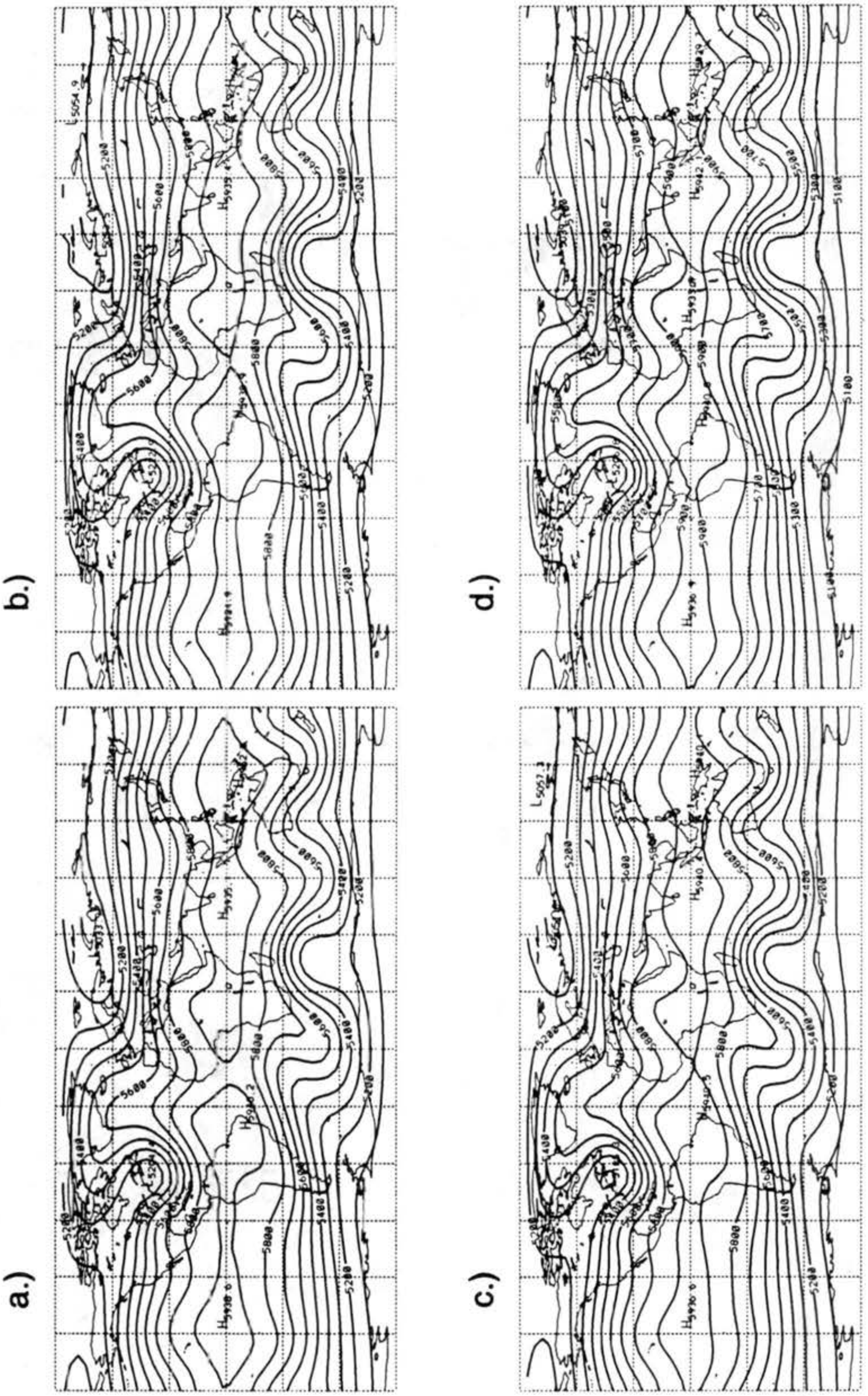

FIGURE 3.40: Test case 5. Height field after 15 days. 100 m contour interval. a.) NCAR T106 model. b.) Arakawa-Lamb model. c.) TIG-2562 model. d.) TIG-10242 model.

Fig. 3.41, Fig. 3.42 and Fig. 3.43 show the NCAR T106 height field minus AL height field and the NCAR T106 height field minus the two resolutions of TIG model height fields after 5 days, 10 days and 15 days. There are significantly smaller errors in the TIG-10242 model than in the TIG-2562 model. The big difference in height at the center of the mountain results from an anomaly in the NCAR T106 solution.

Fig. 3.44 shows the conservation of total energy and potential enstrophy for the Arakawa-Lamb model. Fig. 3.45 and Fig. 3.46 show the conservation properties of the NCAR T42 spectral model and TIG-2562 model.

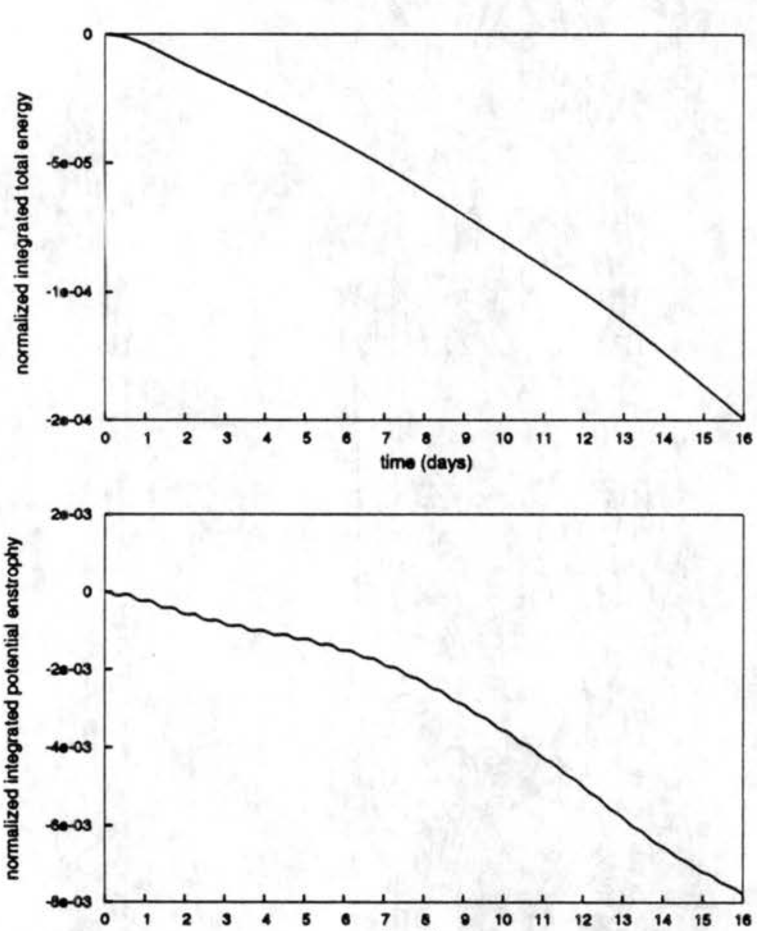

FIGURE 3.44: Test case 5. Arakawa-Lamb model. a.) Normalized integrated total energy. b.) Normalized integrated potential enstrophy.

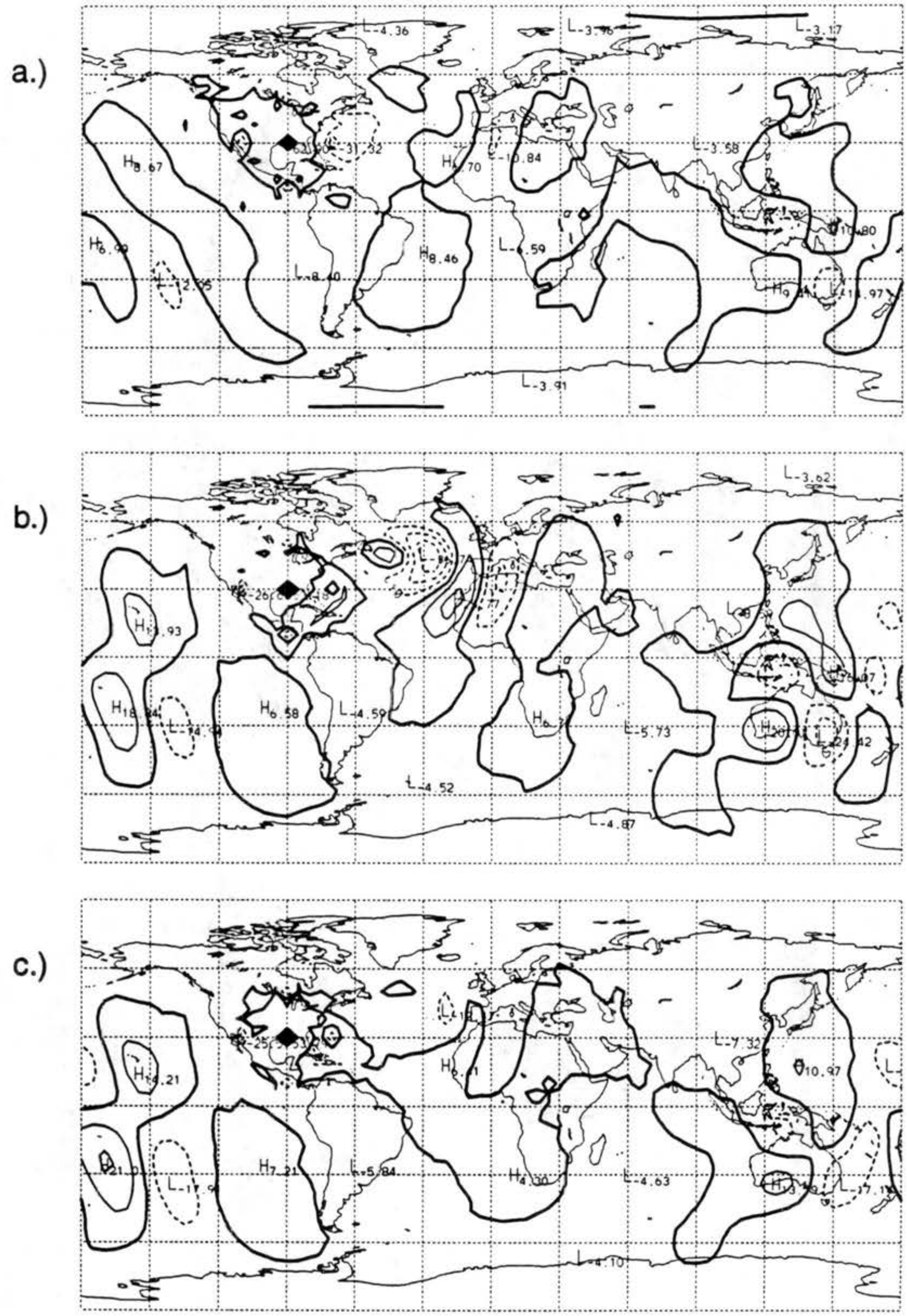

FIGURE 3.41: Test case 5. Difference in height field after 5 days. 10 m contour interval. a.) NCAR T106 model minus Arakawa-Lamb model. b.) NCAR T106 model minus TIG-2562 model. c.) NCAR T106 model minus TIG-10242 model.

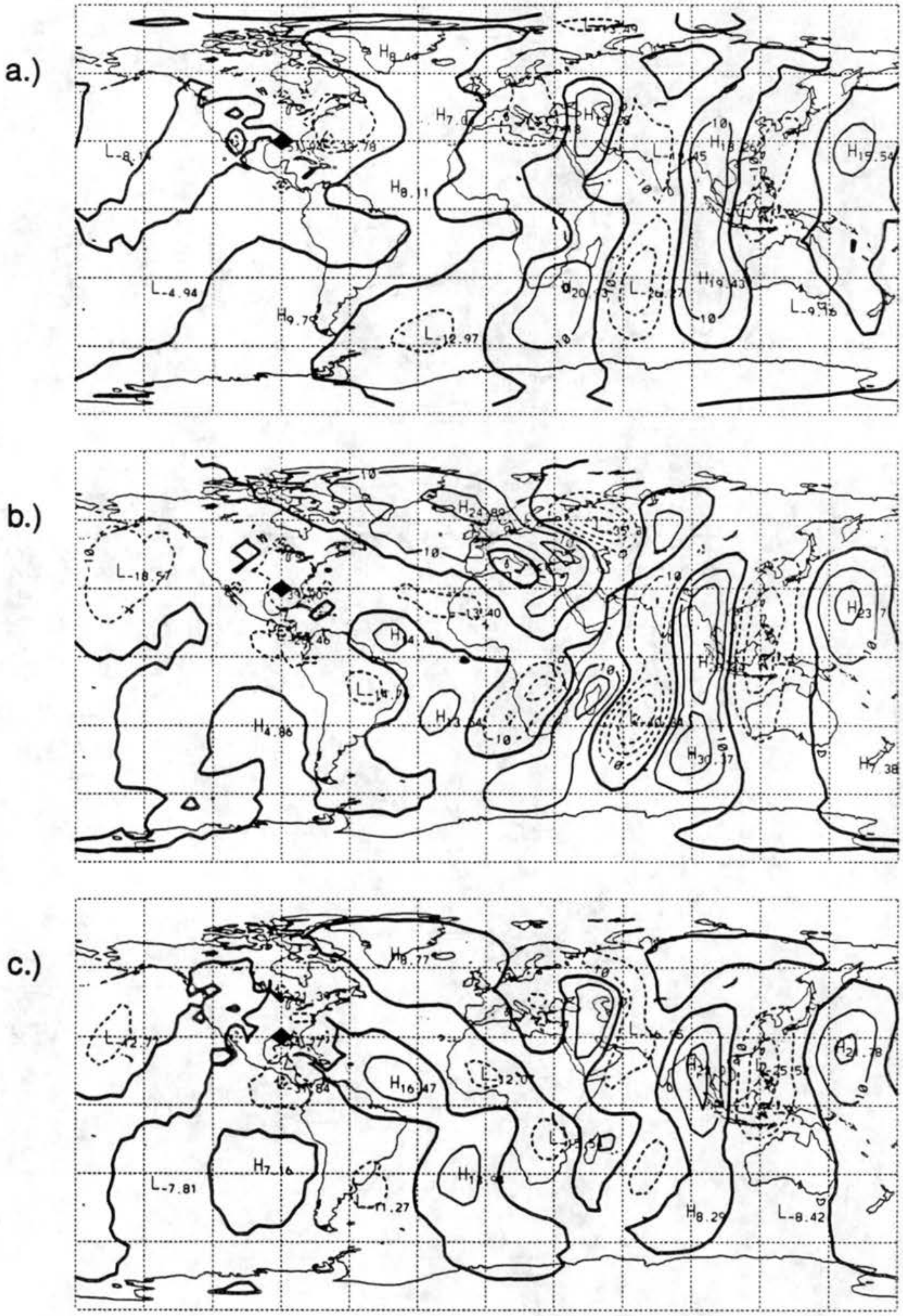

FIGURE 3.42: Test case 5. Difference in height field after 10 days. 10 m contour interval. a.) NCAR T106 model minus Arakawa-Lamb model. b.) NCAR T106 model minus TIG-2562 model. c.) NCAR T106 model minus TIG-10242 model.

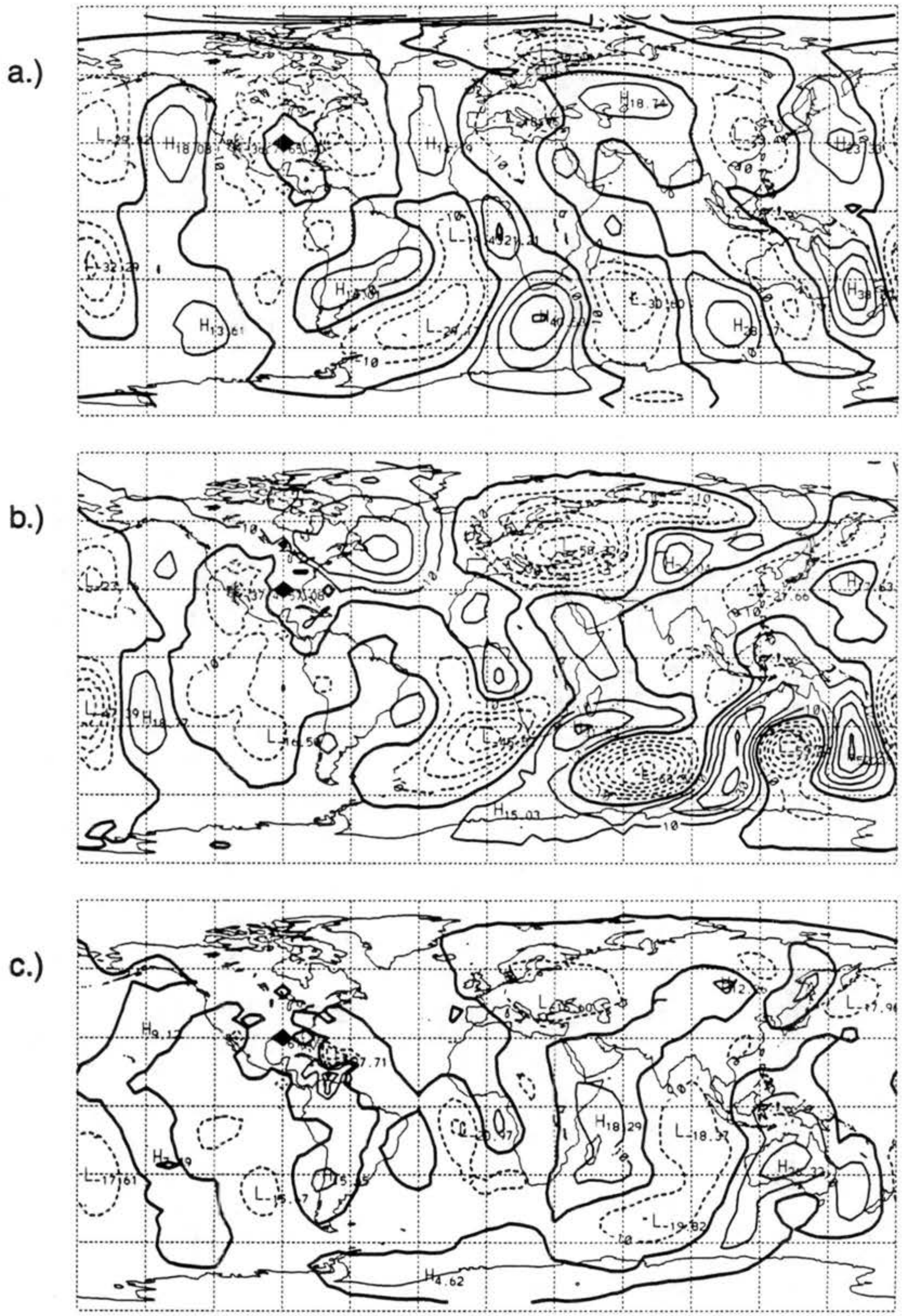

FIGURE 3.43: Test case 5. Difference in height field after 15 days. 10 m contour interval. a.) NCAR T106 model minus Arakawa-Lamb model. b.) NCAR T106 model minus TIG-2562 model. c.) NCAR T106 model minus TIG-10242 model.

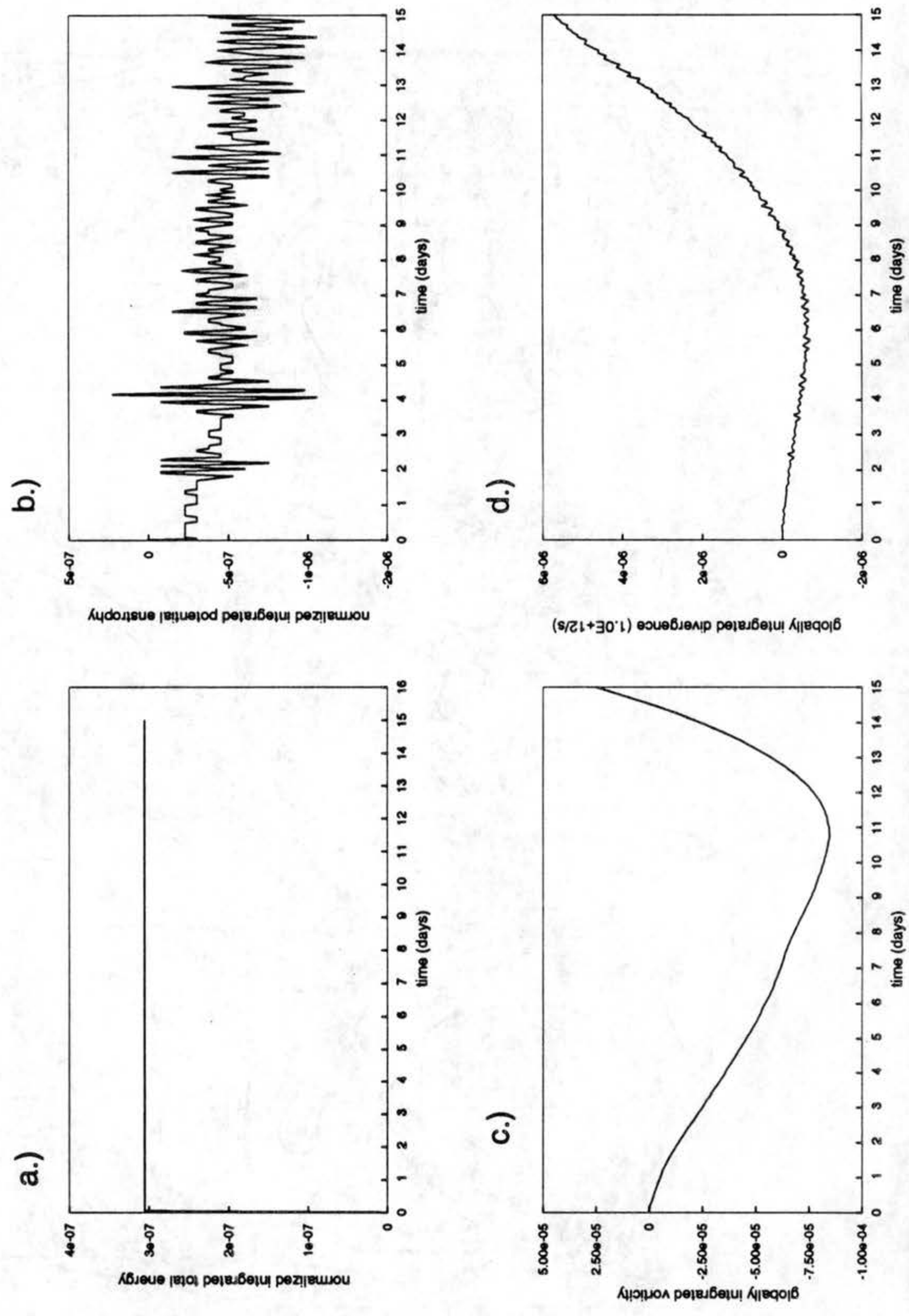

FIGURE 3.45: Test case 5. NCAR spectral model. a.) Normalized integrated total energy. b.) Normalized integrated potential enstrophy. c.) Integrated vorticity. d.) Integrated divergence.

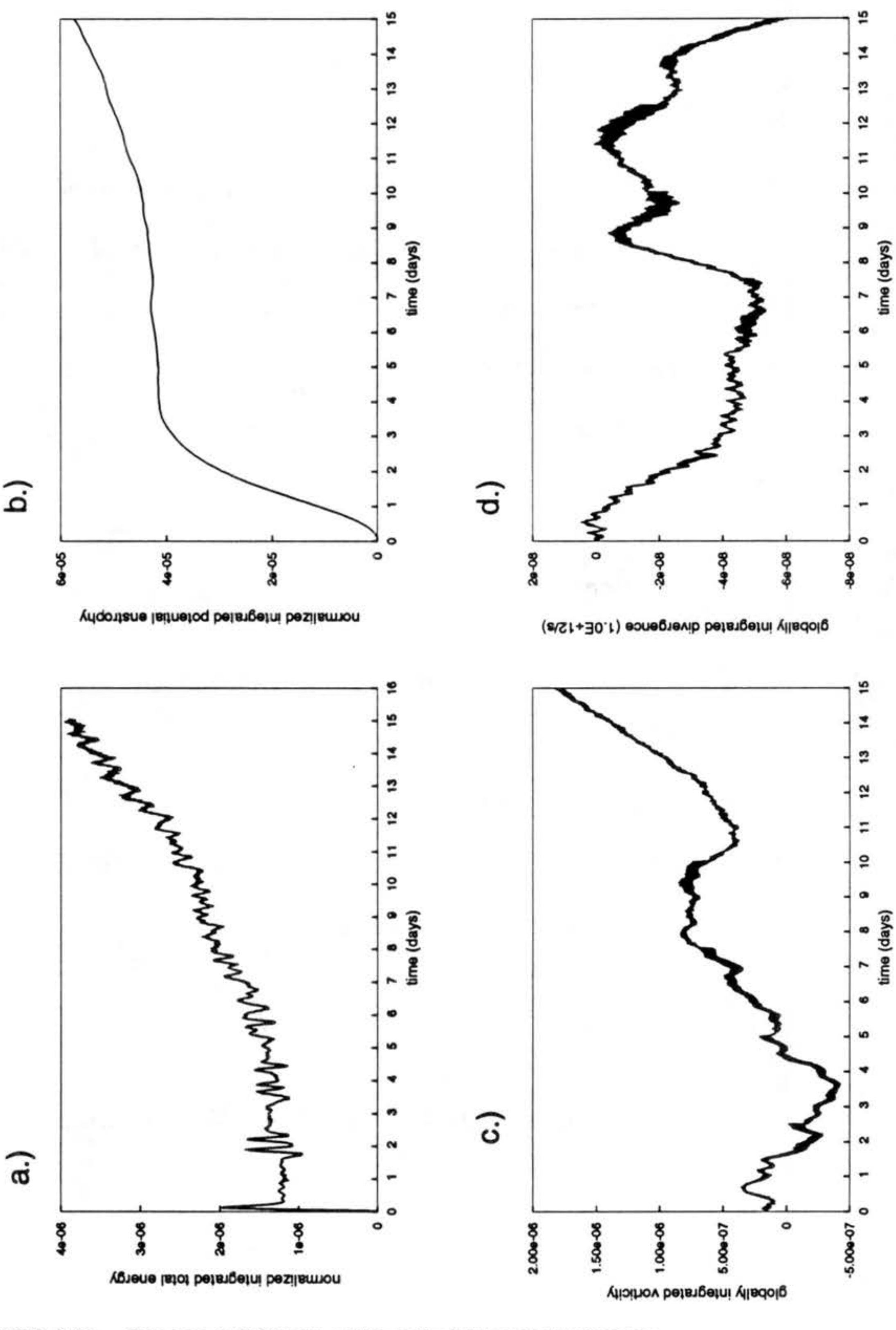

FIGURE 3.46: Test case 5. TIG-2562. a.) Normalized integrated total energy. b.) Normalized integrated potential enstrophy. c.) Integrated vorticity. d.) Integrated divergence.

3.8 Test case 6: Rossby-Haurwitz Wave.

3.8.1 Description

This set of initial conditions was first used by Phillips (1959) and has been used by many investigators since. It is the same initial condition used by Masuda (1986), although with different parameters, so, at least qualitatively we can compare the results from the TIG model with Masuda's results. Since the TIG model is the same as Masuda's model except for the grid and elliptic equation solver, we would expect similar results.

There is no divergence in the initial conditions. The initial velocity is given by

$$u = a\omega\cos\theta + aK\cos^{R-1}\theta(R\sin^2\theta - \cos^2\theta)\cos R\lambda, \tag{3.47}$$

$$v = -aKR\cos^{R-1}\theta\sin\theta\sin R\lambda, \tag{3.48}$$

where the parameters ω and K are set to $\omega = K = 7.848\times10^{-6}\ \mathrm{s}^{-1}$. The parameter R is the zonal wave number of the disturbance. Here we use $R = 4$ to compare the models. Masuda describes results with other wavenumbers.

The initial geopotential field is given by

$$gh = gh_0 + a^2A(\theta) + a^2B(\theta)\cos R\lambda + a^2C(\theta)\cos 2R\lambda, \tag{3.49}$$

where

$$A(\theta) = gh_0 + \frac{\omega}{2}(2\Omega + \omega)\cos^2\theta \tag{3.50}$$

$$+\frac{1}{4}K^2\cos^{2R}\theta\,[\,(R+1)\cos^2\theta+(2R^2-R-2)-2R^2\cos^{-2}\theta\,]\,,$$

$$B(\theta)\ =\ \frac{2\,(\Omega+\omega)\,K}{(R+1)\,(R+2)}\cos^R\theta\,[\,(R^2+2R+2)\ -\ (R+1)^2\cos^2\theta\,]\,, \tag{3.51}$$

$$C(\theta)\ =\ \frac{1}{4}K^2\cos^{2R}\theta\,[\,(R+1)\cos^2\theta-(R+2)\,]\,. \tag{3.52}$$

The mean height of the free surface h_0 is set $h_0 = 8\times10^3$ m. The initial height is shown in Fig. 3.47.

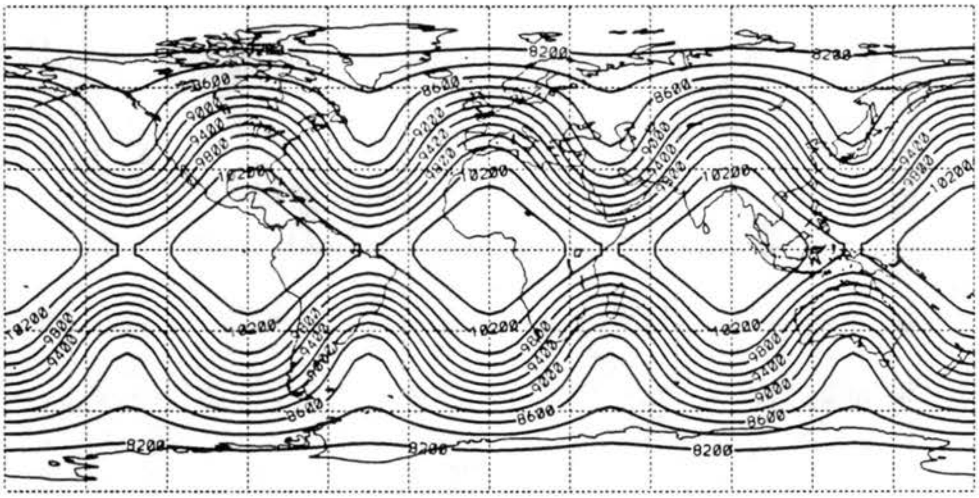

FIGURE 3.47: Test case 6. Initial height field when R=4. 200 m contour interval.

The Rossby-Haurwitz wave is an analytic solution to the nondivergent shallow equations. The shape of the free surface moves from west to east with an angular velocity ν given by

$$\nu\ =\ \frac{R\omega\,(R+3)\ -2\Omega}{(R+1)\,(R+2)}\,. \tag{3.53}$$

With the above values of R and ω, we get $\nu = 2.47\times10^{-6}$ rad s^{-1}, or about 12 degrees per day.

In terms of stream function and velocity potential, the winds are given by

$$\psi = -a^2\omega\sin\theta + a^2K\cos^R\theta\sin\theta\cos R\lambda, \tag{3.54}$$

$$\chi = 0. \tag{3.55}$$

3.8.2 Numerical Results

Again, since there is no analytic solution to the divergent shallow water equations with these initial conditions, we will compare results with those of a high resolution spectral model. In this case the daily spectral coefficients are available from NCAR in a netCDF file called REF0092.cdf

Fig. 3.48, Fig. 3.49 and Fig. 3.50 show the height field after 1 day, 7 days and 14 days, respectively. Looking at the TIG-2562 model after 14 days, we can see the initial pattern that starts with a zonal wavenumber 4 structure, is changing to a wavenumber 1 pattern at higher latitudes. The same thing occurs in Masuda's results. However, while Masuda's fields develops wavenumber 1 patterns that are antisymmetric across the equator, the TIG model height field remains symmetric across the equator. Lorenz (1972) and Hoskins (1973) discuss the conditions necessary for a Rossby-Haurwitz wave as a solution to the nondivergent shallow water equations with a β-plane approximation to become unstable. Suppose we partition the stream function $\psi = \psi_0 + \psi'$, where ψ_0 is the stream function for the Rossby-Haurwitz wave, and ψ' is a slight perturbation. Lorenz and Hoskins show that under certain conditions ψ' will amplify with time, and that ψ' will have a zonal wavenumber smaller than the given Rossby-Haurwitz wave. They also show that a Rossby-Haurwitz wave with a high zonal wave number is less

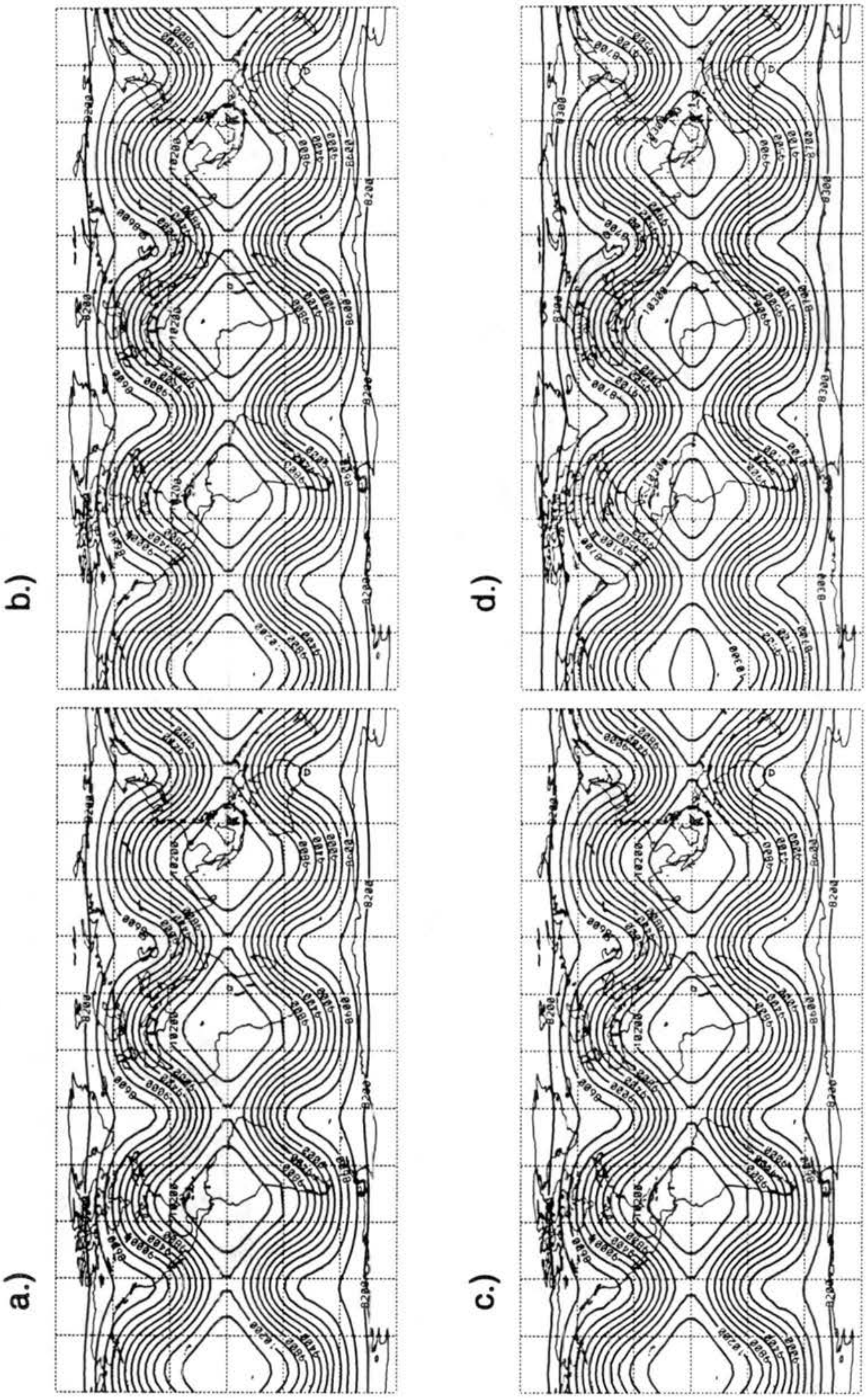

FIGURE 3.48: Test case 6. Height field after 1 day. R=4. 200 m contour interval. a.) NCAR T106 model. b.) Arakawa-Lamb model. c.) TIG-2562 model. d.) TIG-10242 model.

Test case 6: Rossby-Haurwitz Wave.

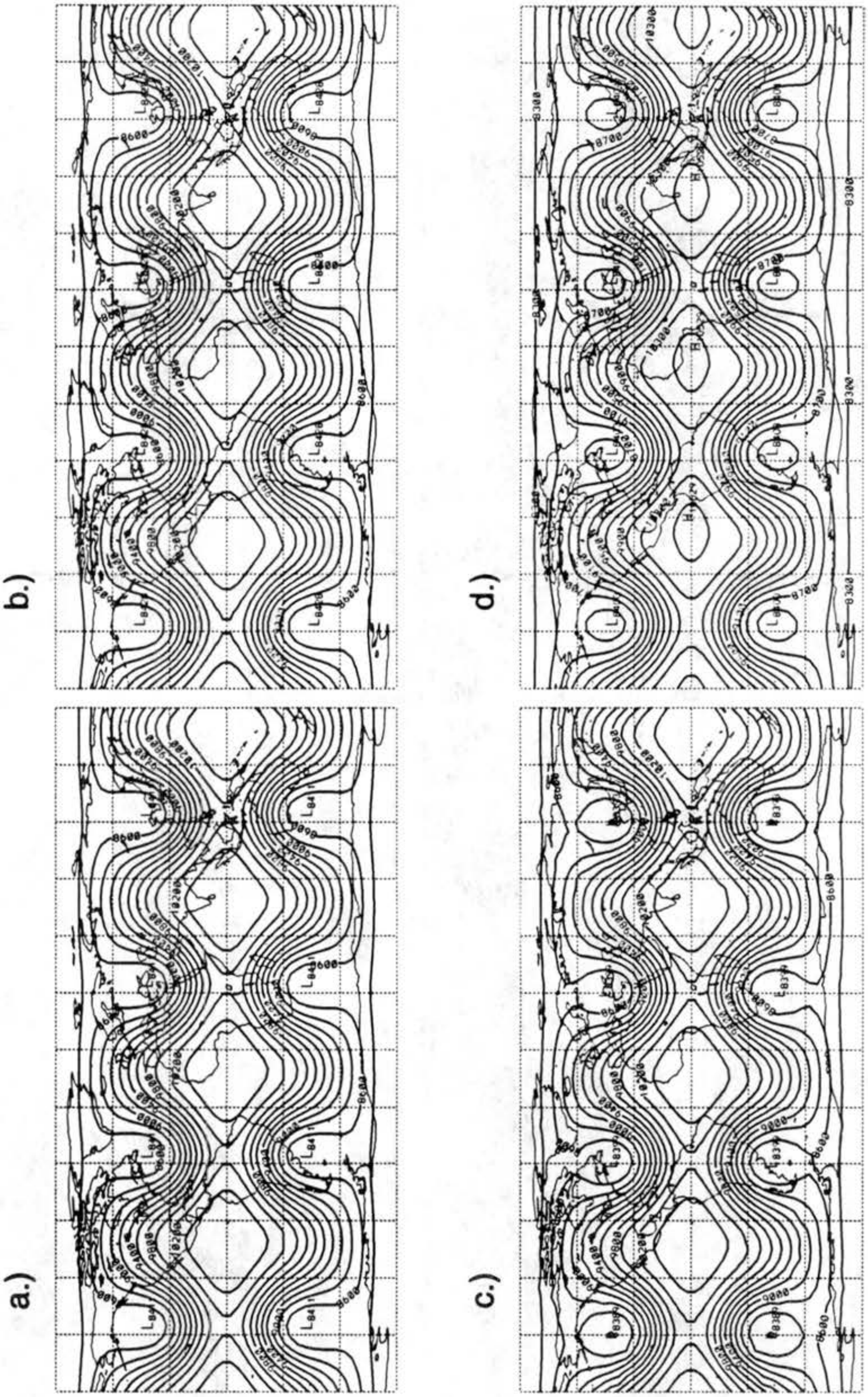

FIGURE 3.49: Test case 6. Height field after 7 day. R=4. 200 m contour interval. a.) NCAR T106 model. b.) Arakawa-Lamb model. c.) TIG-2562 model. d.) TIG-10242 model.

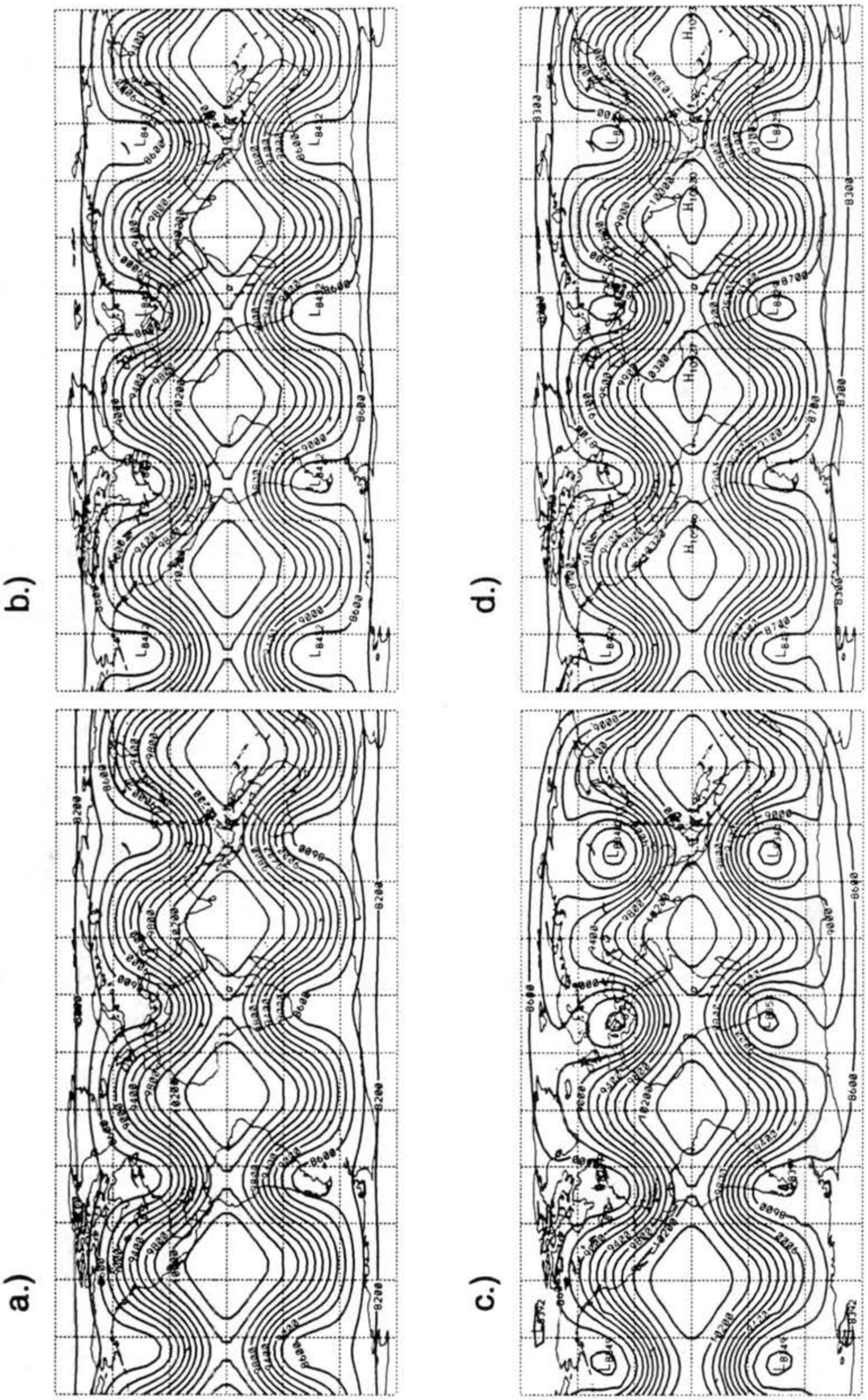

FIGURE 3.50: Test case 6. Height field after 14 day. R=4. 200 m contour interval. a.) NCAR T106 model. b.) Arakawa-Lamb model. c.) TIG-2562 model. d.) TIG-10242 model.

stable, that is ψ' will amplify faster, than one with a low zonal wavenumber. In the TIG models, the flow must interact with the grid to create ψ'. Looking at Fig. 3.50, the zonal wavenumber 1 pattern is not as strong in the TIG-10242 model as in the TIG-2562 model. It may be that the initial perturbation for the higher resolution grid is smaller.

Fig. 3.51 shows the evolution of the height field for the Rossby-Haurwitz wave with zonal wave number $R = 5$ for the TIG-10242 model. While the overall shape of the wave changes, no wavenumber 1 pattern develops at higher latitudes. One would think the wavenumber 5 structure of the grid would cause a strong perturbation in the wavenumber 5 flow pattern. Fig. 3.52 shows the evolution of the Rossby-Haurwitz wave with zonal wavenumber $R = 6$ for the TIG-10242 model. Clearly, the pattern is changing toward smaller wavenumbers faster than the $R = 4$ case.

Fig. 3.53, Fig. 3.54, and Fig. 3.55 show the NCAR T106 height field minus AL height field and the NCAR T106 height field minus the two resolutions of TIG model height fields after 1 days, 7 days and 14 days, respectively. Again, the higher resolutions TIG-10242 does a substantially better job than does TIG-2562. We can see large error develop in the regions of sharp gradient as the finite difference fields are advected too slowly.

Fig. 3.56 shows the conservation properties of the Arakawa-Lamb model. Fig. 3.57 shows the conservation properties of the NCAR T42 spectral model. It should be noted the NCAR model was run with no diffusion. Fig. 3.58 shows the conservation properties of the TIG-2562 model.

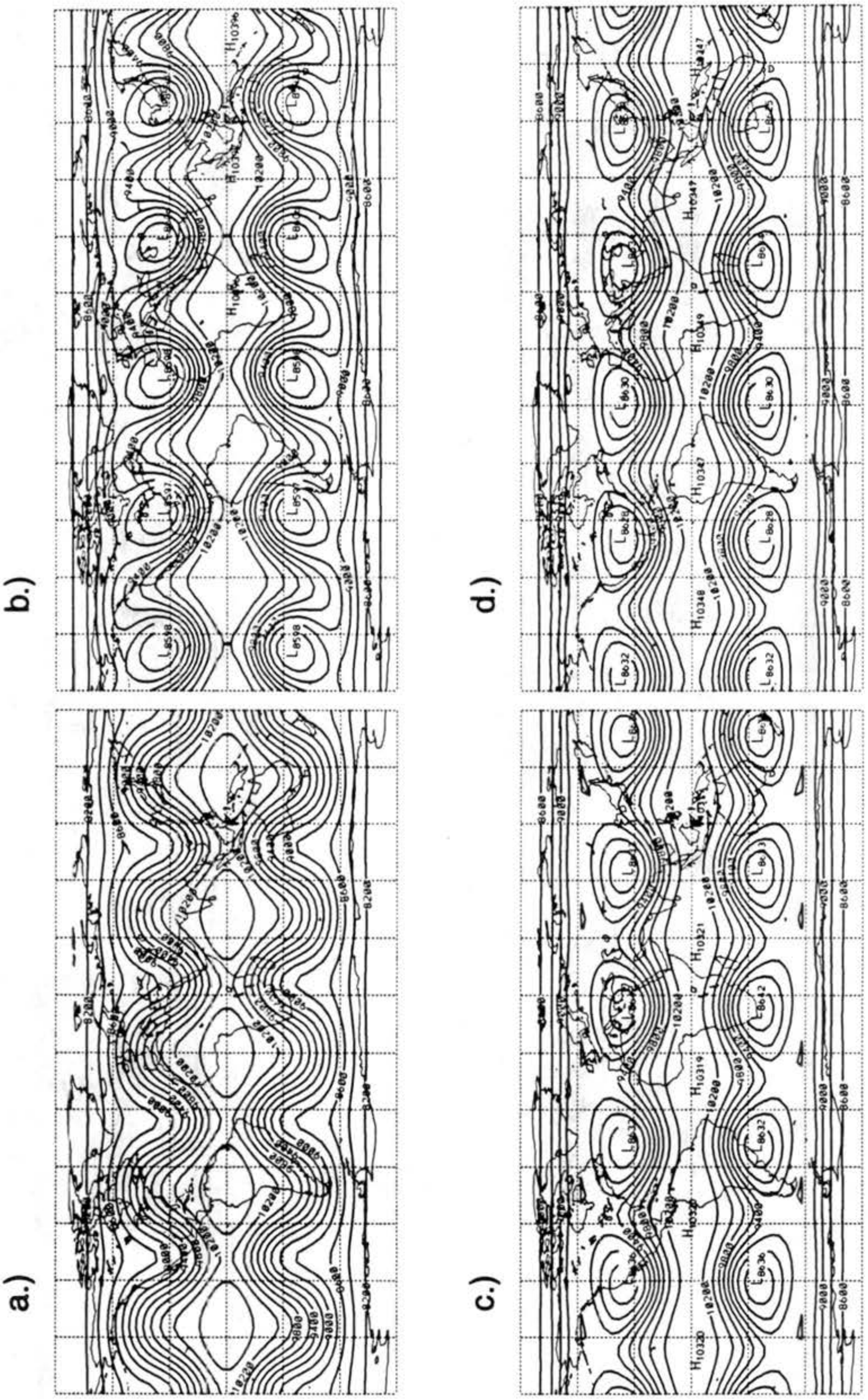

FIGURE 3.51: Test case 6. Evolution of height field for zonal wave number R=5 for the TIG-10242 model. 200 m contour interval. a.) Day 0. b.) Day 5. c.) Day 10. d.) Day 14.

Test case 6: Rossby-Haurwitz Wave.

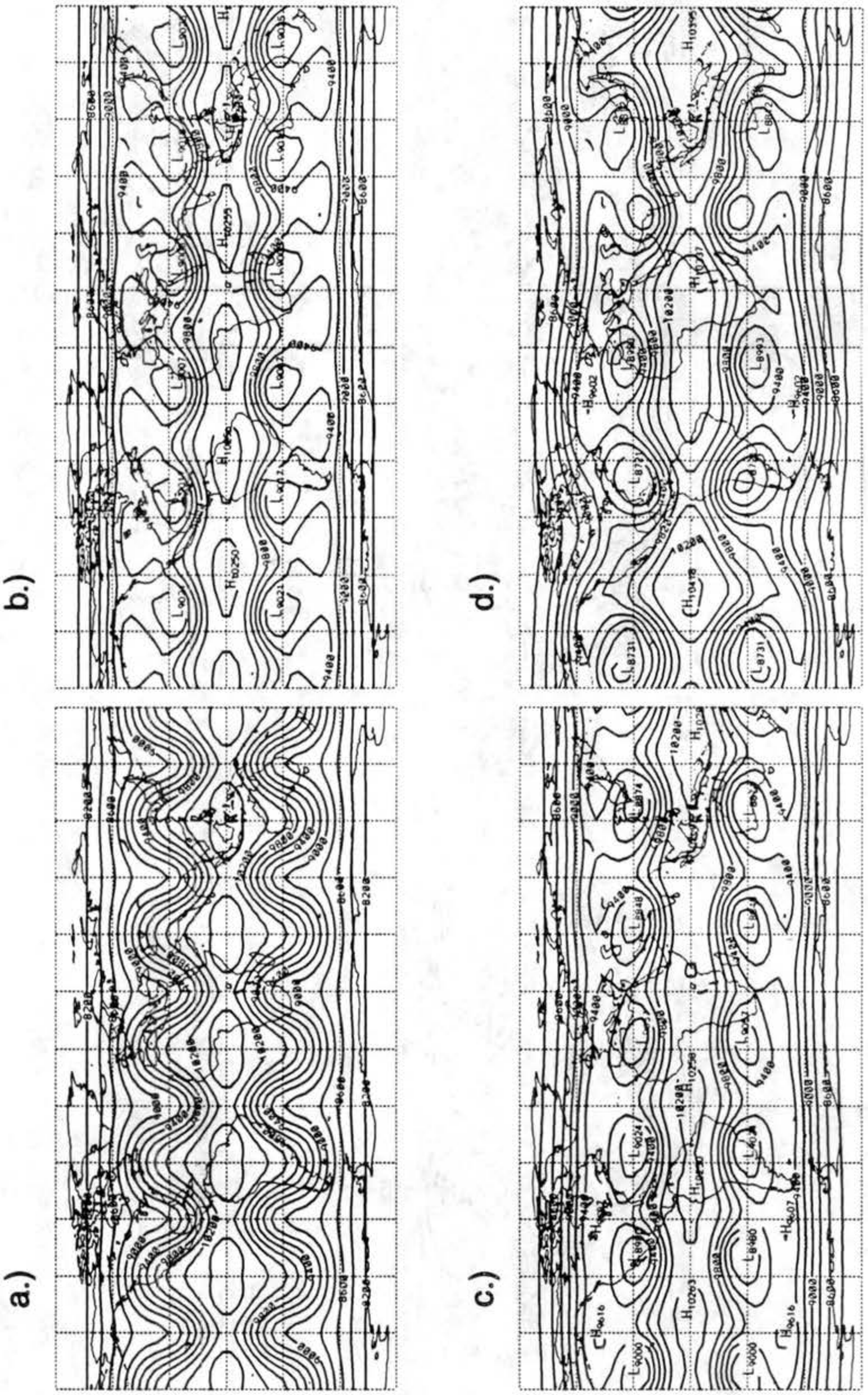

FIGURE 3.52: Test case 6. Evolution of height field for zonal wave number R=6 for the TIG-10242 mode. 200 m contour interval. a.) Day 0. b.) Day 5. c.) Day 10. d.) Day 14.

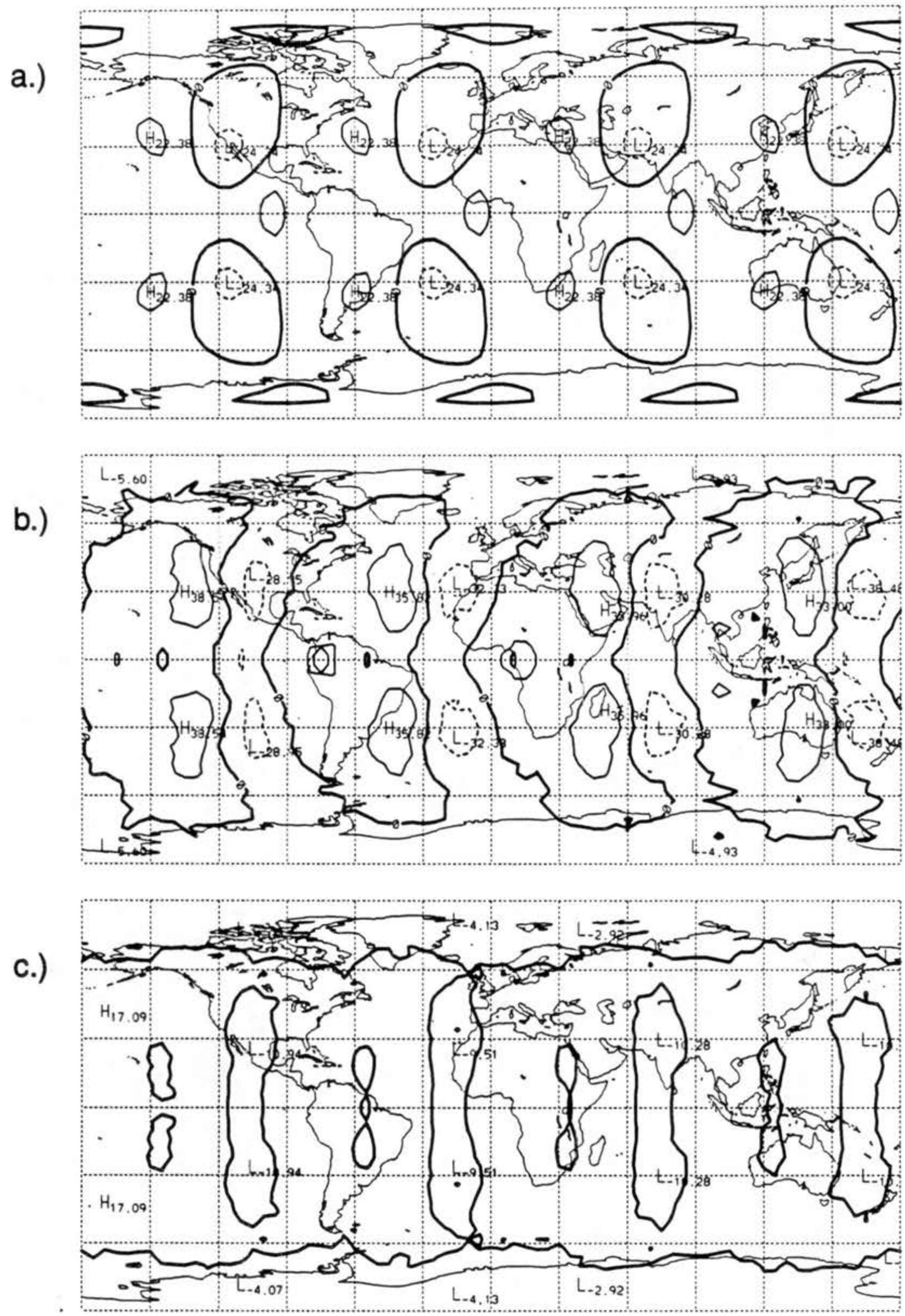

FIGURE 3.53: Test case 6. Difference in height field after 1 day. 20 m contour interval. a.) NCAR T106 model minus Arakawa-Lamb model. b.) NCAR T106 model minus TIG-2562 model. c.) NCAR T106 model minus TIG-10242 model.

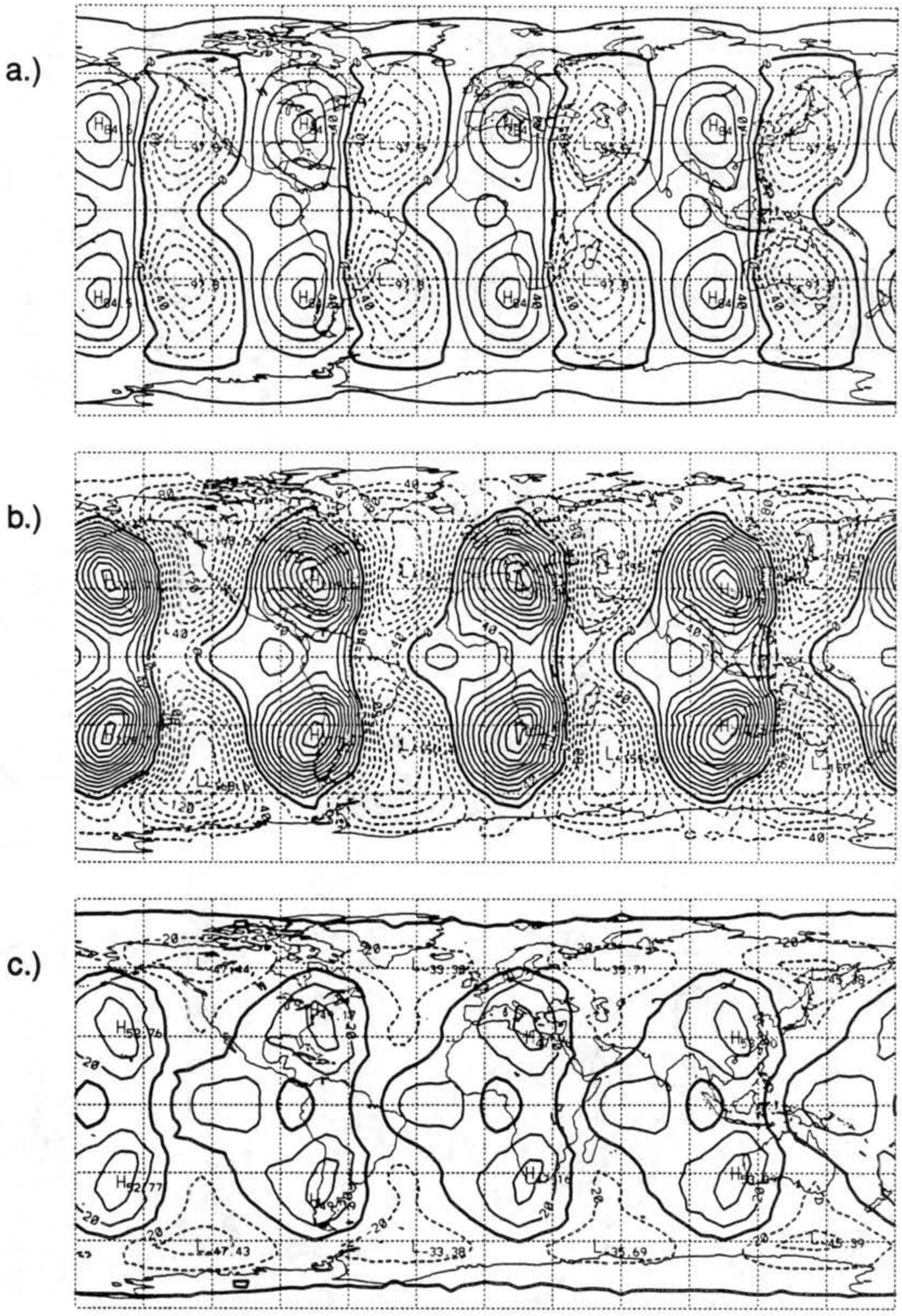

FIGURE 3.54: Test case 6. Difference in height field after 7 days. 20 m contour interval. a.) NCAR T106 model minus Arakawa-Lamb model. b.) NCAR T106 model minus TIG-2562 model. c.) NCAR T106 model minus TIG-10242 model.

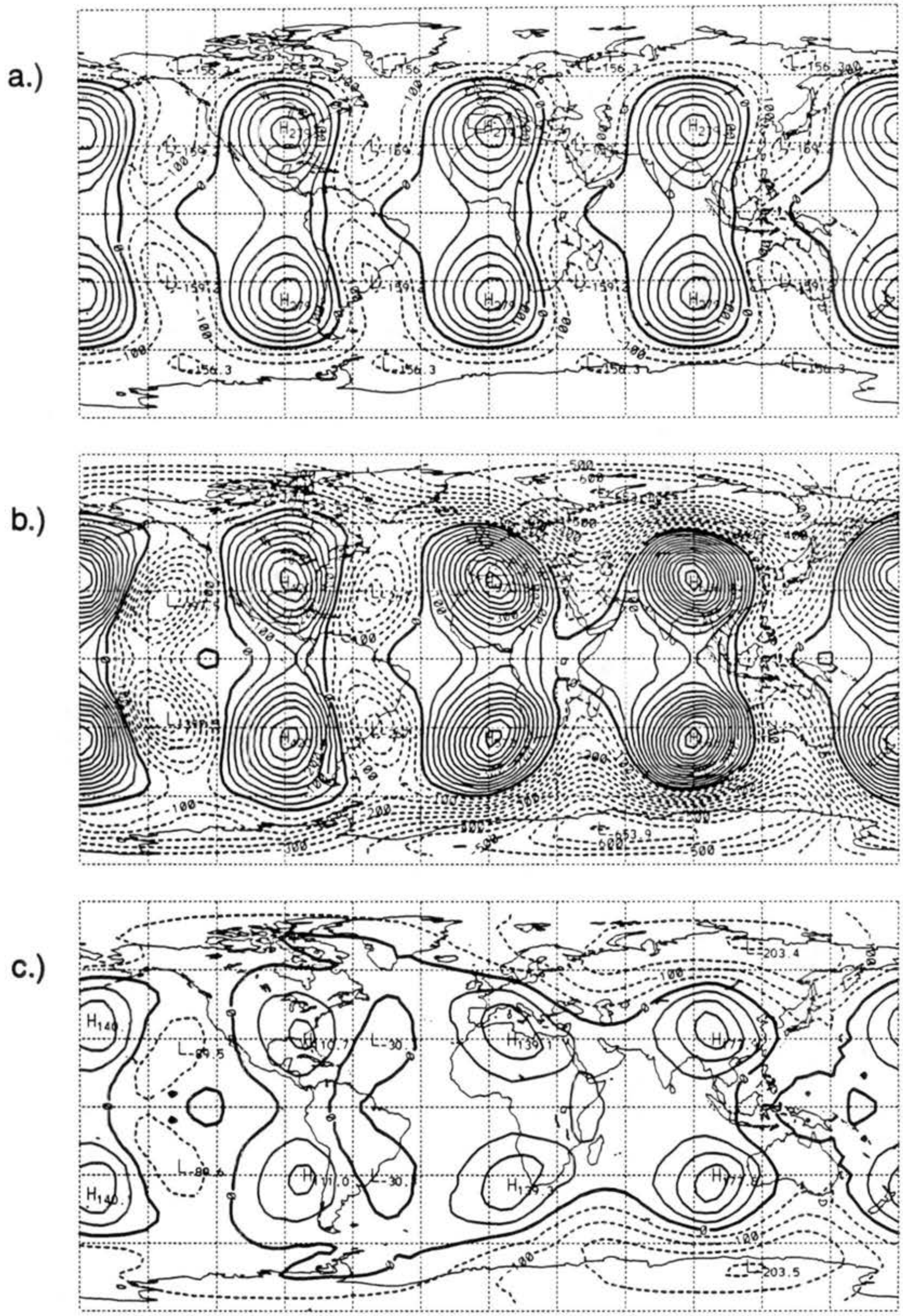

FIGURE 3.55: Test case 6. Difference in height field after 14 days. 50 m contour interval. a.) NCAR T106 model minus Arakawa-Lamb model. b.) NCAR T106 model minus TIG-2562 model. c.) NCAR T106 model minus TIG-10242 model.

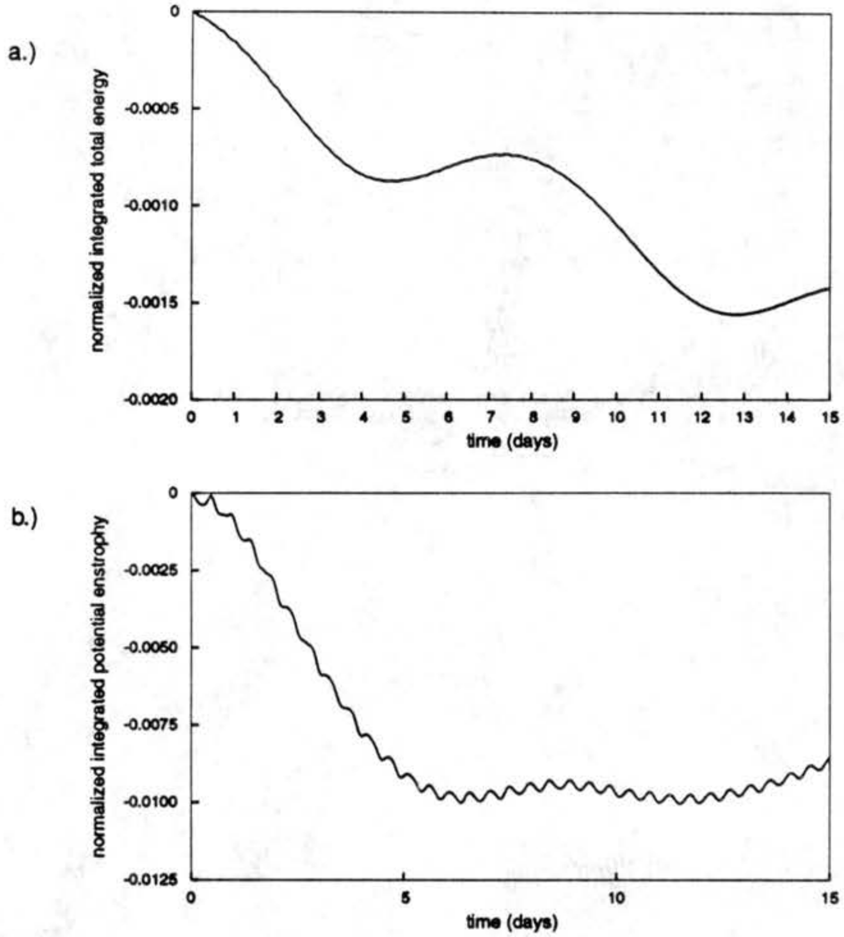

FIGURE 3.56: Test case 6. Arakawa-Lamb model. a.) Normalized integrated total energy. b.) Normalized integrated potential enstrophy

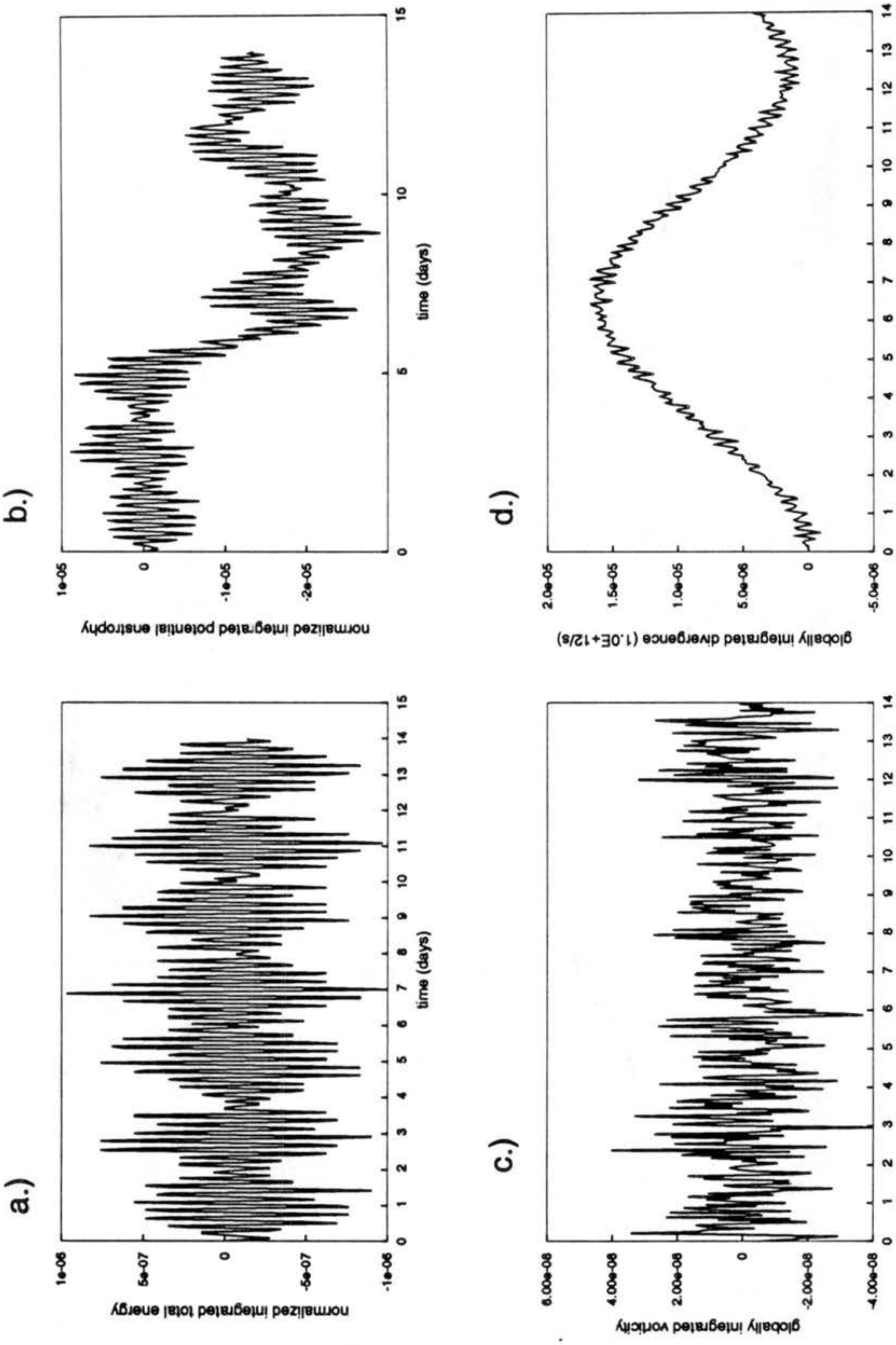

FIGURE 3.57: Test case 6. NCAR spectral model. a.) Normalized integrated total energy. b.) Normalized integrated potential enstrophy. c.) Integrated vorticity. d.) Integrated divergence.

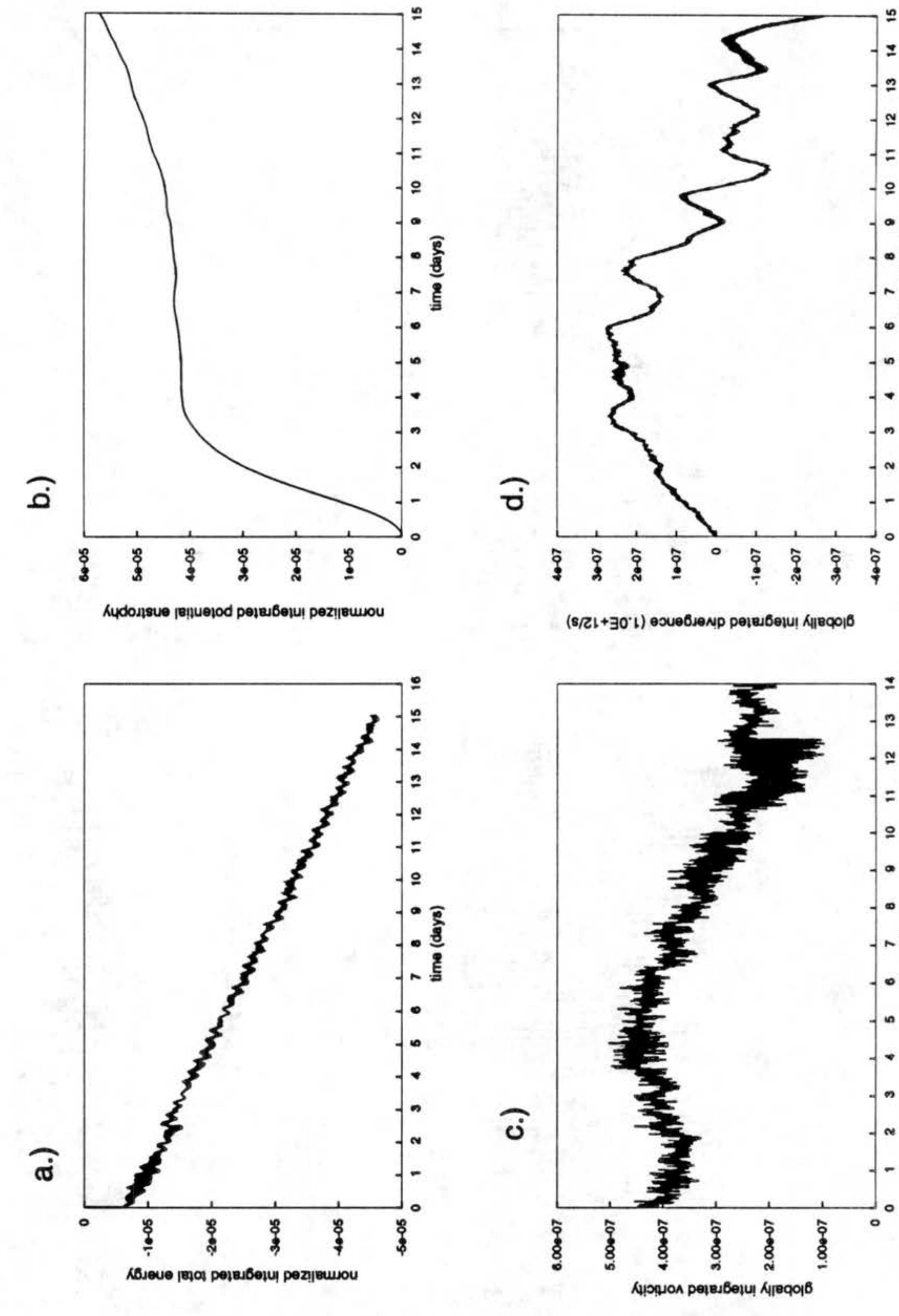

FIGURE 3.58: Test case 6. TIG-2562. a.) Normalized integrated total energy. b.) Normalized integrated potential enstrophy. c.) Integrated vorticity. d.) Integrated divergence

3.9 Test case 7: Analyzed 500 mb Height and Wind Field Initial Conditions

3.9.1 Description

The last test case initializes the models with observed 500 mb height and wind fields from 0000 GMT December 21, 1978. This set of initial conditions produces strong flow over the north pole, and so is a good test of each model's performance at the poles. It was first used by Ritchie (1988) to test a semi-Lagrangian model.

Other sets of observed initial conditions are available in netCDF format. These include 0000 GMT January 16, 1979 in which the initial pattern contains two cut off lows that develops into a blocking situation. Another case is derived from the strong zonal flow observed at 0000 GMT January 9, 1979. These two cases do not directly test the models at the poles, so they are not used in this study.

The initial conditions are contained in a netCDF formate file available from Rudy Jacobs at NCAR called REF0077.cdf. These data are truncated at T63. Truncation at this resolution should not remove any of the detail present in the original observations. The file includes the spectral transform coefficients of the u and v components of the wind, and the height field. The mean height field has been set to 10 km. The spectral coefficients for absolute vorticity and divergence are also contained in REF0077.cdf. Stream function and velocity potential can be computed by either solving Poisson's equations in spectral space, as discussed in Appendix C, or numerically solving in grid point space with the multigrid solver.

This data have been processed with nonlinear normal node initialization to prevent spurious gravity waves from contaminating the results. A set of initial conditions for this same situation without normal mode initialization is also available in a file called INI0077.cdf. Williamson (1992) suggests that a nonlinear normal mode initialization scheme consistent with the numerical scheme being tested should be used to process the initial data. However, for this study, we simply used the preprocessed data.

3.9.2 Numerical Results

REF0077.cdf also includes the daily spectral coefficients of the evolving pattern produced by the NCAR model. Fig. 3.59.a shows the data used to initialize the height field, Fig. 3.59.b shows the height field after 5 days, and Fig. 3.60 shows the four days in between. These plots are included to display the evolving pattern. Fig. 3.61 shows the

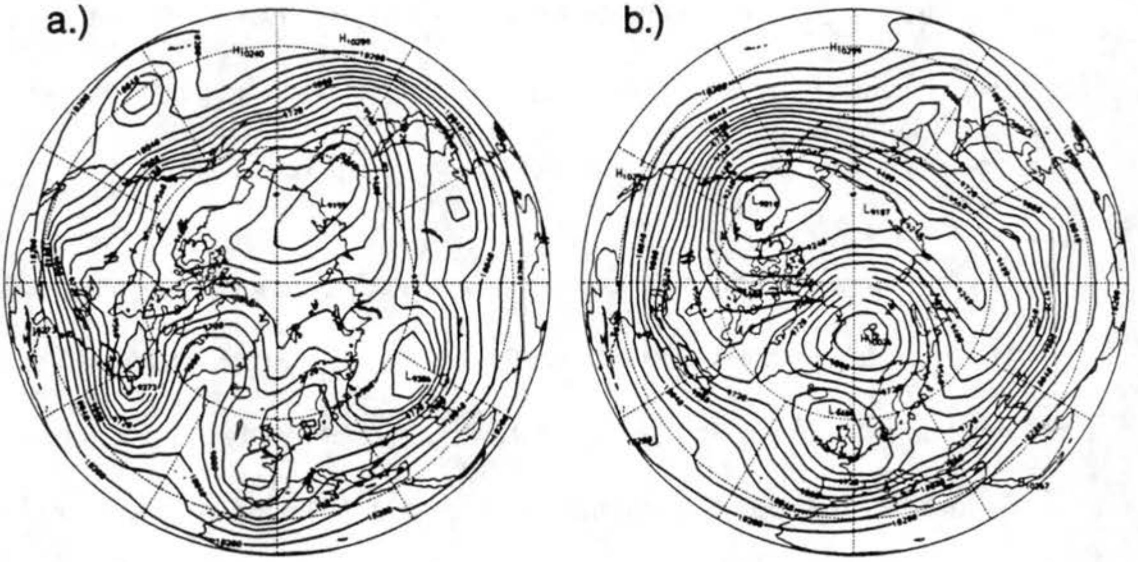

FIGURE 3.59: Test case 7. 80 m contour interval a.) Initial height field. b.) T106 height field after 5 days.

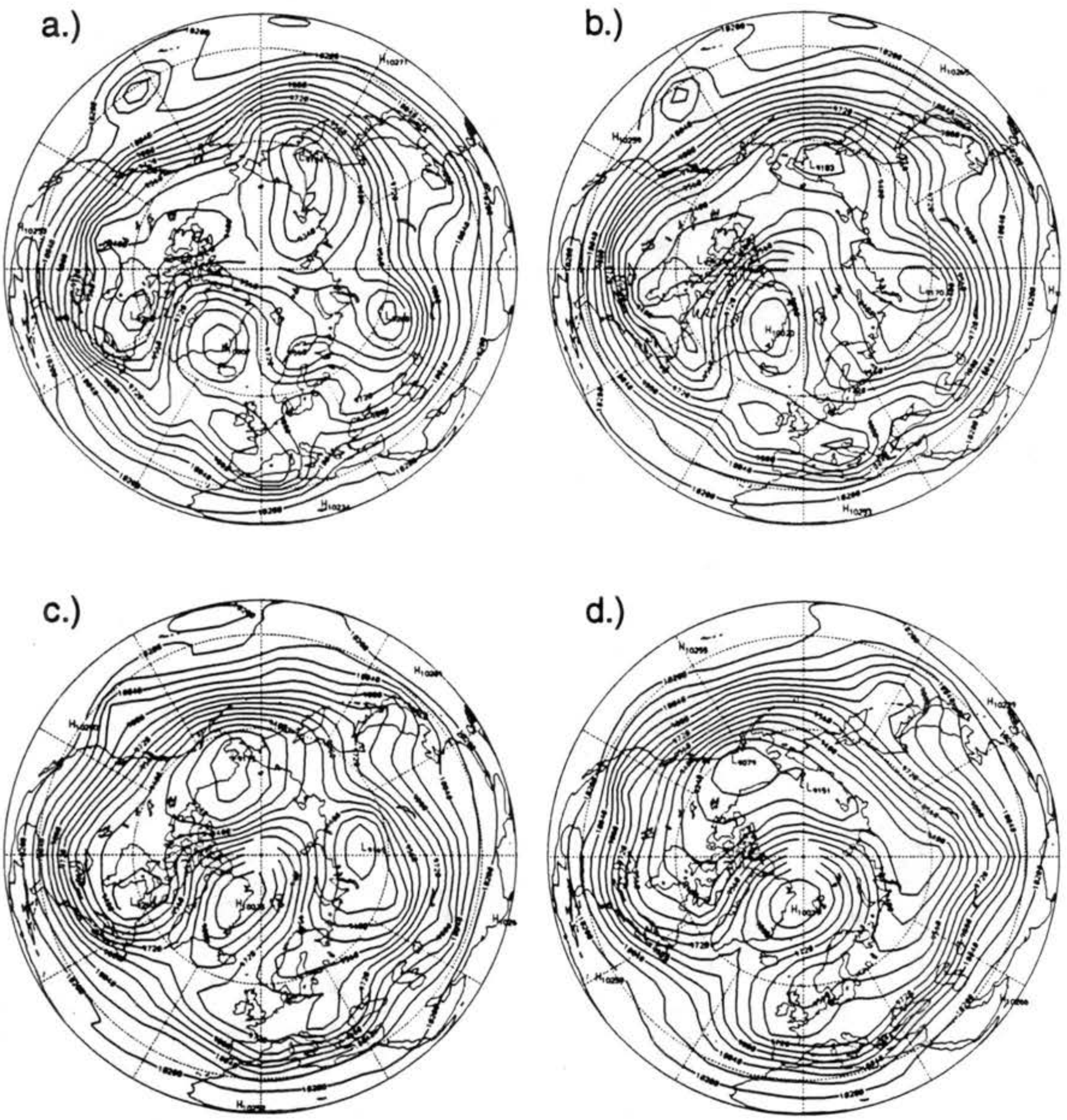

FIGURE 3.60: Test case 7. Height field of T106 model. 80 m contour interval. a.) Day1. b.) Day 2. c.) Day 3. d.) Day 4.

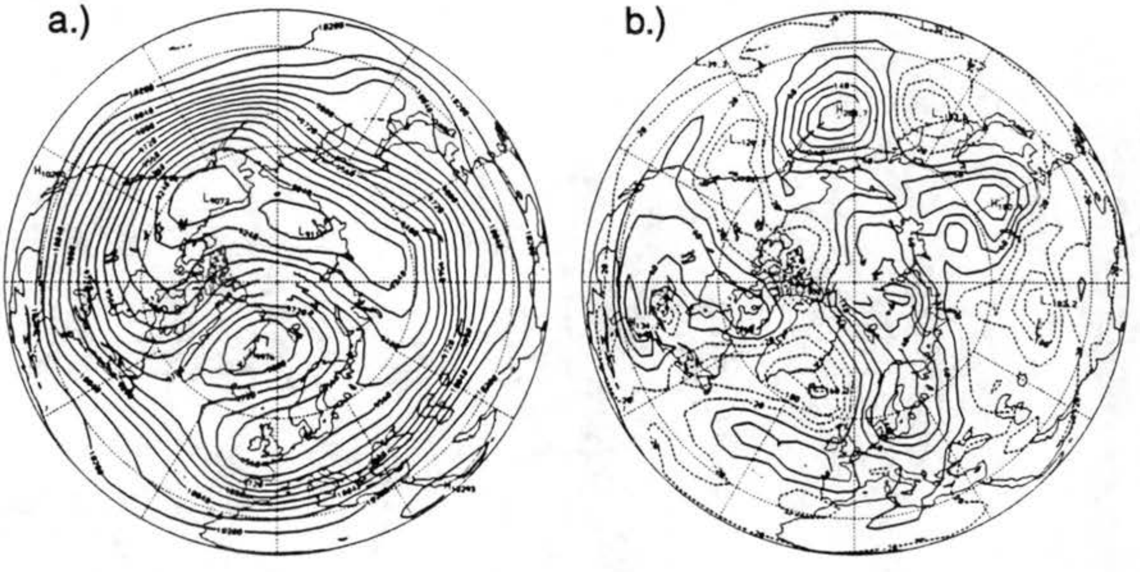

FIGURE 3.61: Test case 7. a.) Height field of Arakawa-Lamb model after 5 days. 80 m contour interval. b.) T106 height field minus AL height field after 5 days. 40 m contour interval.

Arakawa-Lamb model after 5 days and the difference between the T106 model and the AL model. Fig. 3.62 show the TIG-2562 model after 5 days, and the T106 height field

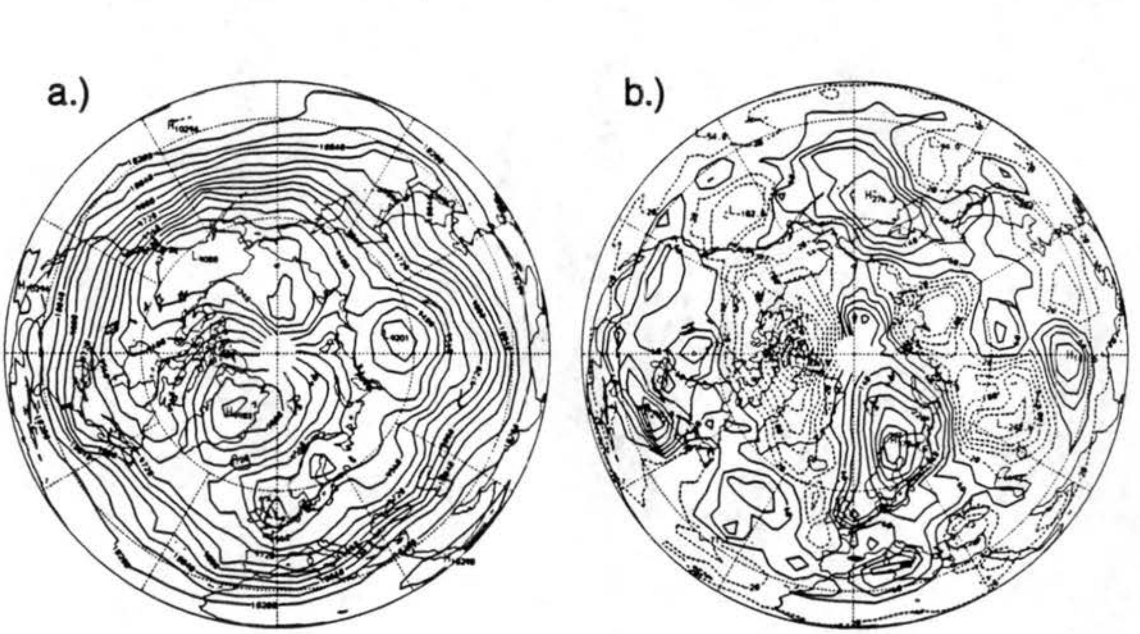

FIGURE 3.62: Test case 7. a.) Height field of TIG-2562 model after 5 days. 80 m contour interval. b.) T106 height field minus TIG-2562 height field after 5 days. 40 m contour interval.

minus the TIG-2562 height field after 5 days. Finally Fig. 3.63 shows the TIG-10242

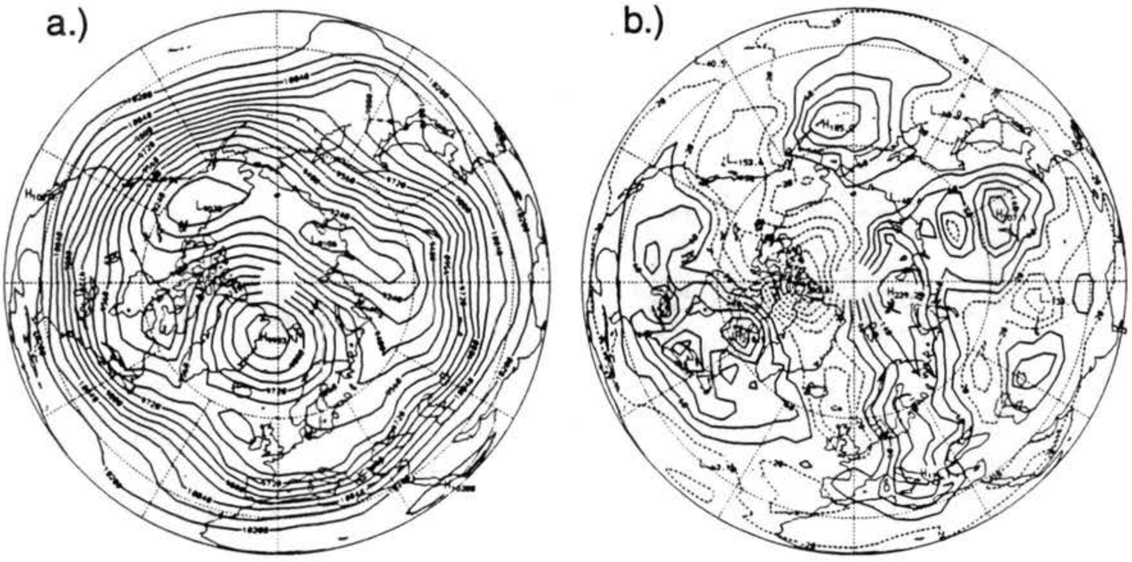

FIGURE 3.63: Test case 7. a.) Height field of TIG-10242 model after 5 days. 80 m contour interval. b.) T106 height field minus TIG-10242 height field after 5 days. 40 m contour interval.

height field after 5 days, and the T106 height field minus the TIG-10242 height field after 5 days. Looking at Fig. 3.60, we can see the flow over the pole is created by the region of high pressure that moves north-east along Greenland. The TIG-2562 model fails to move this region as far east as the Arakawa-Lamb or TIG-10242 model. It is also more noisy than the other models. The Arakawa-Lamb model and the TIG-10242 models move the region of high pressure further east. In the TIG-10242 model, the region is more built-up, and cut-off than in the NCAR spectral or the Arakawa-Lamb model.

CHAPTER 4

Model Performance

4.1 The TIG Model

To test the computational performance of the model, test case 5 from chapter 3 was run for 500 time steps. The TIG model was run with grid resolutions of 642, 2562, 10242 and 40962 cells. For each increase in resolution, the time step was cut in half, so the time steps were 1200, 600, 300 and 150 seconds, respectively. Since the model spends most of its time executing the procedures `relaxation`, `interpolate`, `inject_residual`, `check_convergence`, `flux_divergence`, `Jacobian`, `Laplacian`, `wrap`, we examine the individual performance of these procedures. The first four procedures in the above list are associated with the multigrid solver, and the last four are associated with time stepping the equations. The model was run on a CRAY Y-MP C90, and the performance results were obtained using the CRAY utility `perfview`. The results are shown in Table 4.1. All times are measured in seconds. The row labeled time/call is the average CPU time spent executing the procedure. The last column shows the total CPU time to run the program and the average megaflop rate.

Table 4.1: TIG Model performance for four different grid resolutions

Resolution		Results by procedure								Total for program
		relaxation	interpolate	inject_ residual	check_con vergence	flux_ divergence	Jacobian	Laplacian	wrap	
642 cells	CPU time	3.72E+00	2.16E-01	1.10E-01	5.46E-01	5.43E-01	2.84E-01	1.38E-01	1.56E+00	8.10E+00
	time/call	3.37E-05	6.88E-05	3.52E-05	3.55E-05	2.17E-04	1.42E-04	9.17E-05	1.30E-05	
	execution %	45.9	2.7	1.4	6.7	6.7	3.5	1.7	19.3	
	Mflop	53.0	43.9	43.2	14.5	91.8	114.6	126.0	0.0	42.5
2562 cells	CPU time	6.56E+00	4.76E-01	2.39E-01	8.55E-01	1.25E+00	5.98E-01	2.99E-01	2.13E+00	1.42E+01
	time/call	4.51E-05	9.55E-05	4.80E-05	4.23E-05	4.98E-04	2.99E-04	1.99E-04	1.34E-05	
	execution %	46.3	3.4	1.7	6.0	8.8	4.2	2.1	15.0	
	Mflop	101.3	75.6	76.5	20.0	159.3	215.2	231.4	0.0	86.9
10242 cells	CPU time	1.25E+01	9.60E-01	4.77E-01	1.43E+00	3.20E+00	1.41E+00	7.06E-01	2.88E+00	2.81E+01
	time/call	6.28E-05	1.50E-04	7.46E-05	5.06E-05	1.28E-03	7.02E-04	7.46E-04	1.33E-05	
	execution %	44.6	3.4	1.7	5.1	11.4	5.0	2.5	10.3	
	Mflop	195.1	139.8	143.9	29.7	248.5	365.0	392.7	0.0	166.2
40960 cells	CPU time	2.75E+01	2.24E+00	1.08E+00	2.64E+00	9.47E+00	4.02E+00	2.00E+00	4.02E+00	6.73E+01
	time/call	1.02E-04	2.41E-04	1.16E-04	6.76E-05	3.78E-03	2.00E-03	1.33E-03	1.37E-05	
	execution %	40.8	3.3	1.6	3.9	14.1	6.0	3.0	6.0	
	Mflop	347.0	235.8	251.8	51.8	335.8	510.9	552.8	0.0	272.8

4.2 The Arakawa-Lamb Model

The Arakawa-Lamb model was run with the same conditions at three grid resolutions. The first resolution has 72 grid points longitudinally and 44 latitudinally, the second has 152 grid points longitudinally and 94 latitudinally, and the third 252 grid points longitudinally and 154 latitudinally. The CPU time and average megaflop rate are shown in Table 4.2.

Table 4.2: Model performance of three grid resolutions of the Arakawa-Lamb Model

Resolution	CPU time	Mflop
72×44	2.73E+00	350.7
152×94	1.75E+01	430.3
252×154	6.09E+01	520.0

It is clear that the Arakawa-Lamb model is considerably faster than the TIG model in terms of megaflop rate. At the lower resolutions, the TIG model suffers from vectors too short to be efficiently processed by the vector hardware. This is evident in the dramatic increase in megaflop rate as the resolution is increased. Also, the procedures `flux_divergence`, `Jacobian`, `Laplacian`, which only deal with the finest grid resolution, are much faster than the procedures `relaxation`, `interpolate`, `inject_residual` which deal with all grid resolutions. The fault lies in the way in which the scalar fields are distributed in computer memory. This could be done in a more efficient fashion that would increase the model's performance. A more efficient representation of scalar fields in memory would also eliminate the need for the procedure `wrap`. Despite efforts to improve it, this procedure is very inefficient and accounts for a considerable percentage of the execution time. The procedure `check_convergence` is another bottleneck because it has several `if` tests within loops that do not vectorize.

Fig. 4.1 shows the time needed to execute the program per cell per number of time steps. The data points with circles are the four resolutions of the TIG model in Table 4.1, and the data points with squares are the three Arakawa-Lamb resolutions in Table 4.2. The Arakawa-Lamb model uses more time per grid points as the resolution increases because the polar filtering requires more computational effort. It should be noted that the relatively inefficient algorithm currently used in the polar filter subroutine is to be replaced with more efficient Fast Fourier Transform.

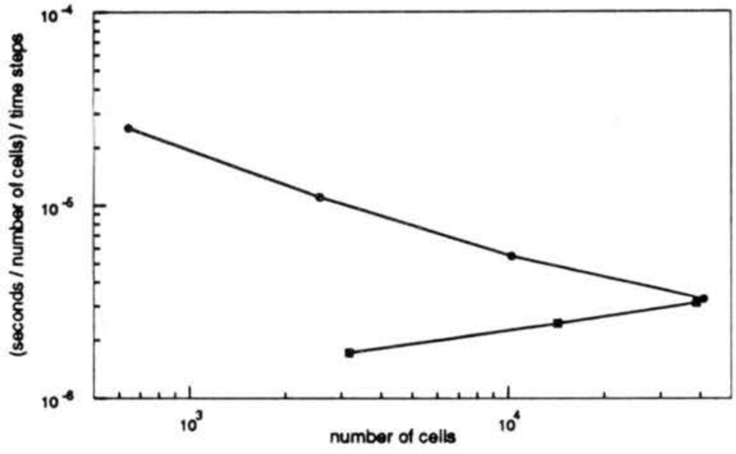

FIGURE 4.1: Time per number of cells as a function of number of cells.

CHAPTER 5

Conclusions and Future Plans

5.1 Conclusions

The TIG shallow water model has advantages, as well as disadvantages, when compared to the Arakawa-Lamb and NCAR spectral transform shallow water models. We can discuss the advantages and disadvantages by comparing the different grids and discretization schemes used in each model. Let us consider the grids first.

The spherical geodesic grid homogeneously covers the sphere, whereas the latitude-longitude grids used in the Arakawa-Lamb model and the NCAR spectral model concentrate grid points near the poles. This results in the need for polar filtering in the Arakawa-Lamb model and unnecessary computational effort in both models. Test case 1 in Chapter 3 shows that the homogeneity of the grid is beneficial when advecting material over the poles of the grid. The latitude-longitude grid in the Arakawa-Lamb model alters the shape of the advected cosine bell depending on the angle between the polar axis of the grid and the advecting current. On the other hand, the shape remains more-or-less the same for all angles in the TIG model. The cases 2 and 3 also indicate the behavior of the flow is not effected by the relative orientations of the TIG grid and the flow.

The test cases point out some shortcomings of the TIG grid as well. As was shown in Test case 3, the sharp gradient in grid cell area can effect the flow. A model using the TIG grid is more difficult to code than a model using a latitude-longitude grid. As was shown at the end of Chapter 2, a complicated data structure is needed to represent a scalar field in computer memory. As a result, more code is required to manipulate the numbers. The data structure prevents long vectors needed for efficient vector processing, and much of the allocated memory is not used. Also, in many subroutines, special code is needed for the pentagons. Some solutions to these problems are discussed below. On the other hand, a scalar field on a latitude-longitude grid can be stored in a simple two-dimensional array of memory.

Consider now the discretization schemes used in each model. The use of the SFVP form of the shallow water equations has the advantage that all the fields in the model are true scalars, independent of a coordinate system, and avoids the problems associated the a spherical coordinate system. This carries over to the discretization scheme in that the scheme depends only upon the distance between cell centers and the length of cell walls. Indeed, the SFVP form of the shallow water equations makes possible the use of the spherical geodesic grid.

On the down side, the SFVP form of the shallow water equations requires solving a set of elliptic equations each time step. Also, this form does not directly predict the u and v components of the wind. The stream function and velocity potential must be interpolated to a latitude-longitude grid where meridional and zonal derivatives are taken to approximate the wind. Of course, the u and v components of the wind are only for diagnostic purposes in the SFVP form. The equations in spherical coordinates have the advantage that high-order schemes exist which conserve total energy and potential enstrophy. The equations in this form are relatively easy to code, since the only special grid points are those at the highest and lowest latitudes.

The spectral method of the NCAR model has advantages and disadvantages, as well. The spherical harmonics isotropically cover of the sphere, and there is no computational dispersion under linear advection. On the other hand, the computationally expensive Legendre transform may limit the use of very high resolution spectral models. In addition there are problems not manifest in the shallow water equations, for example, failure to conserve and maintain positively quantities such as water vapor.

5.2 Future Plans

Future modification to the model will address the data structure problem and the grid area problem. The data structures to store scalar fields can be more efficient if they do not store all grid resolutions within the same block of memory. For example, an alternative way to store just the grid with 642 cells is shown in Fig. 5.1. This structure will allow longer vectors and eliminates some of the wrapping.

The sharp gradient in grid areas can be smoothed using the ideas of the wandering electron grid presented at the end of Chapter 1. If the grid points are allowed to move a little, being repelled away from regions of high grid point density, the result should be a grid that more evenly distributes area.

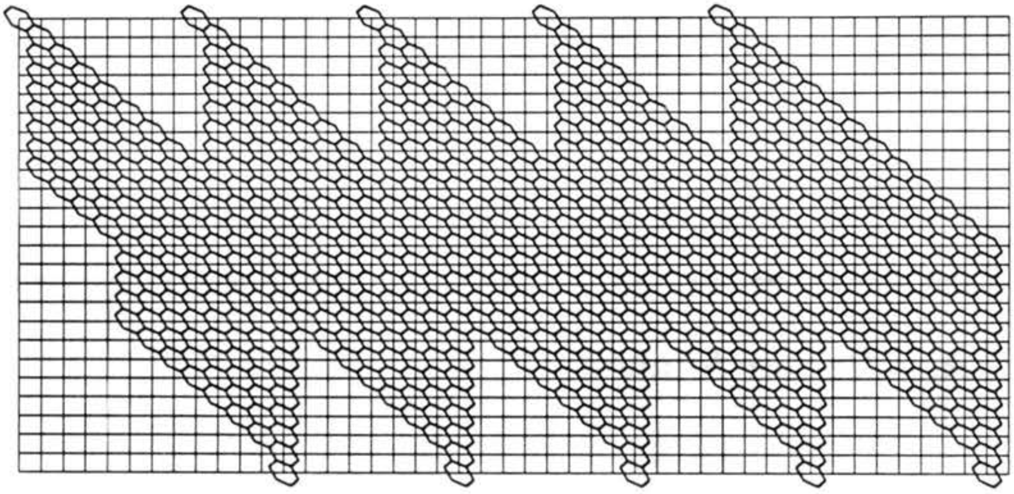

FIGURE 5.1: Data structure to store 642 cell grid.

In the future we would like to write a parallel version of the model. Many of the shortcomings of the model written for a serial computer would be eliminated in a parallel environment. For example, there would no longer be the need for long vector lengths to run efficiently on a vector computer.

$$= \hat{\boldsymbol{k}} \bullet [\nabla \eta \times \boldsymbol{v} + \eta (\nabla \times \boldsymbol{v})] - \nabla^2 \left[gh + \frac{\boldsymbol{v} \bullet \boldsymbol{v}}{2} \right] \tag{A24}$$

$$= \hat{\boldsymbol{k}} \bullet \nabla \times (\eta \boldsymbol{v}) - \nabla^2 \left[gh + \frac{\boldsymbol{v} \bullet \boldsymbol{v}}{2} \right]. \tag{A25}$$

A.3 Stream Function, Velocity Potential Equations

The horizontal velocity is given by $\boldsymbol{v} = \hat{\boldsymbol{k}} \times \nabla \psi + \nabla \chi$. The Coriolis parameter does not depend on time, $\partial f / \partial t = 0$, so, $\partial \eta / \partial t = \partial \zeta / \partial t$. For arbitrary scalar functions A and B, define the Jacobian $J(A, B)$ as follows: $J(A, B) \equiv \hat{\boldsymbol{k}} \bullet (\nabla A \times \nabla B)$. The absolute vorticity equation is given by

$$\frac{\partial \eta}{\partial t} = -\nabla \bullet (\eta \boldsymbol{v}) \tag{A26}$$

$$= -\nabla \bullet [\eta (\hat{\boldsymbol{k}} \times \nabla \psi + \nabla \chi)] \tag{A27}$$

$$= -\nabla \bullet [\eta (\hat{\boldsymbol{k}} \times \nabla \psi)] - \nabla \bullet [\eta \nabla \chi] \tag{A28}$$

$$= -\nabla \bullet [\eta \nabla \chi] - \eta \nabla \bullet (\hat{\boldsymbol{k}} \times \nabla \psi) - \nabla \eta \bullet (\hat{\boldsymbol{k}} \times \nabla \psi) \tag{A29}$$

$$= -\nabla \bullet [\eta \nabla \chi] + \hat{\boldsymbol{k}} \bullet (\nabla \eta \times \nabla \psi) \tag{A30}$$

$$= -\nabla \bullet [\eta \nabla \chi] + J(\eta, \psi). \tag{A31}$$

The kinetic energy term K in the equation for divergence is given by

$$K = \frac{\boldsymbol{v} \bullet \boldsymbol{v}}{2} \tag{A32}$$

$$= \frac{1}{2} [(\hat{\boldsymbol{k}} \times \nabla \psi + \nabla \chi) \bullet (\hat{\boldsymbol{k}} \times \nabla \psi + \nabla \chi)] \tag{A33}$$

$$= \frac{1}{2} [(\hat{\boldsymbol{k}} \times \nabla \psi) \bullet (\hat{\boldsymbol{k}} \times \nabla \psi) + (\hat{\boldsymbol{k}} \times \nabla \psi) \bullet \nabla \chi \tag{A34}$$

$$+\nabla\chi\bullet(\hat{\boldsymbol{k}}\times\nabla\psi)+\nabla\chi\bullet\nabla\chi\,]$$

$$=\frac{1}{2}\left[(\hat{\boldsymbol{k}}\times\nabla\psi)\bullet(\hat{\boldsymbol{k}}\times\nabla\psi)+\nabla\bullet(\chi\nabla\chi)-\chi\nabla^2\chi\right] \tag{A35}$$

$$+\nabla\chi\bullet(\hat{\boldsymbol{k}}\times\nabla\psi)$$

$$=\frac{1}{2}\left[\,[(\hat{\boldsymbol{k}}\times\nabla\psi)\times\hat{\boldsymbol{k}}]\bullet\nabla\psi+\nabla\bullet(\chi\nabla\chi)-\chi\nabla^2\chi\right] \tag{A36}$$

$$+\hat{\boldsymbol{k}}\bullet(\nabla\psi\times\nabla\chi)$$

$$=\frac{1}{2}\left[-[\hat{\boldsymbol{k}}\times(\hat{\boldsymbol{k}}\times\nabla\psi)\,]\bullet\nabla\psi+\nabla\bullet(\chi\nabla\chi)-\chi\nabla^2\chi\right]-J(\psi,\chi) \tag{A37}$$

$$=\frac{1}{2}\left[-[\hat{\boldsymbol{k}}(\nabla\psi\bullet\hat{\boldsymbol{k}})-\nabla\psi(\hat{\boldsymbol{k}}\bullet\hat{\boldsymbol{k}})\,]\bullet\nabla\psi+\nabla\bullet(\chi\nabla\chi)-\chi\nabla^2\chi\right] \tag{A38}$$

$$-J(\psi,\chi)$$

$$=\frac{1}{2}\left[\nabla\psi\bullet\nabla\psi+\nabla\bullet(\chi\nabla\chi)-\chi\nabla^2\chi\right]-J(\psi,\chi) \tag{A39}$$

$$=\frac{1}{2}\left[(\nabla\bullet(\psi\nabla\psi)-\psi\nabla^2\psi)+\nabla\bullet(\chi\nabla\chi)-\chi\nabla^2\chi\right]-J(\psi,\chi)\,. \tag{A40}$$

The divergence equation is given by

$$\frac{\partial\delta}{\partial t}=\hat{\boldsymbol{k}}\bullet\nabla\times(\eta\boldsymbol{v})-\nabla^2(gh+K) \tag{A41}$$

$$=\hat{\boldsymbol{k}}\bullet\nabla\times[\eta(\hat{\boldsymbol{k}}\times\nabla\psi+\nabla\chi)\,]-\nabla^2(gh+K) \tag{A42}$$

$$=\hat{\boldsymbol{k}}\bullet\nabla\times[\eta(\hat{\boldsymbol{k}}\times\nabla\psi)+\eta\nabla\chi]-\nabla^2(gh+K) \tag{A43}$$

$$=\hat{\boldsymbol{k}}\bullet\nabla\times[\eta(\hat{\boldsymbol{k}}\times\nabla\psi)\,]+\hat{\boldsymbol{k}}\bullet\nabla\times(\eta\nabla\chi)-\nabla^2(gh+K) \tag{A44}$$

$$=\hat{\boldsymbol{k}}\bullet\{\nabla\eta\times(\hat{\boldsymbol{k}}\times\nabla\psi)+\eta\,[\nabla\times(\hat{\boldsymbol{k}}\times\nabla\psi)\,]\,\} \tag{A45}$$

$$+\hat{\boldsymbol{k}}\bullet[\nabla\eta\times\nabla\chi+\eta\,(\nabla\times\nabla\chi)\,]-\nabla^2(gh+K)$$

$$=\hat{\boldsymbol{k}}\bullet[\hat{\boldsymbol{k}}(\nabla\psi\bullet\nabla\eta)-\nabla\psi(\nabla\eta\bullet\hat{\boldsymbol{k}})\,] \tag{A46}$$

$$+\hat{\boldsymbol{k}} \bullet \{\eta\,[\hat{\boldsymbol{k}}\,(\nabla\bullet\nabla\,\psi) - \nabla\,\psi\,(\nabla\bullet\hat{\boldsymbol{k}}) - (\hat{\boldsymbol{k}}\bullet\nabla)\,\nabla\,\psi + (\nabla\,\psi\bullet\nabla)\,\hat{\boldsymbol{k}}]\,\}$$

$$+\hat{\boldsymbol{k}} \bullet (\nabla\,\eta \bullet \nabla\,\chi) - \nabla^2\,(gh + K)$$

$$= \nabla\,\psi \bullet \nabla\,\eta + \eta\nabla^2\,\psi + J\,(\eta, \chi) - \nabla^2\,(gh + K) \tag{A47}$$

$$= \nabla\bullet\,(\eta\nabla\,\psi) + J\,(\eta, \chi) - \nabla^2\,(gh + K)\,. \tag{A48}$$

And finally, the continuity equation is given by

$$\frac{\partial}{\partial t}h^* = -\nabla\bullet\,(h^*\,\boldsymbol{v}) \tag{A49}$$

$$= -\nabla\bullet\,[h^*\,(\hat{\boldsymbol{k}}\times\nabla\,\psi + \nabla\,\chi)\,] \tag{A50}$$

$$= -\nabla\bullet\,[h^*\,(\hat{\boldsymbol{k}}\times\nabla\,\psi)\,] - \nabla\bullet\,(h^*\,\nabla\,\chi) \tag{A51}$$

$$= -\nabla\bullet\,(h^*\,\nabla\,\chi) - [h^*\,\nabla\bullet\,(\hat{\boldsymbol{k}}\times\nabla\,\psi) + \nabla\,h^* \bullet (\hat{\boldsymbol{k}}\times\nabla\,\psi)\,] \tag{A52}$$

$$= -\nabla\bullet\,(h^*\,\nabla\,\chi) + \hat{\boldsymbol{k}} \bullet (\nabla\,h^* \times \nabla\,\psi) \quad \text{(by A.1 and A.5)} \tag{A53}$$

$$= -\nabla\bullet\,(h^*\,\nabla\,\chi) + J\,(h^*, \psi)\,. \tag{A54}$$

Appendix B: The Arakawa-Lamb Shallow Water Model

Arakawa and Lamb (1981) derived a second order potential enstrophy and energy conserving scheme for the shallow water equations. As an extension to this scheme, Takano (1981) derived a fourth-order scheme which conserves potential enstrophy and total energy, and, in addition, gives fourth-order accuracy for the advection of potential vorticity in the case of nondivergent flow.

B.3 Properties of the Continuous Equations

To begin, we define potential vorticity $q \equiv \eta / h^{*} = (f + \zeta) / h^{*}$, where η is absolute vorticity, and ζ is relative vorticity. This is a slightly different definition than was presented in chapter 2. In addition, we define mass flux $\boldsymbol{v}^{*} \equiv \boldsymbol{v} h^{*}$, kinetic energy $K \equiv (\boldsymbol{v} \bullet \boldsymbol{v}) / 2$ and geopotential $\phi \equiv gh$, where h^{*} is the depth of the fluid and h is height of the free surface above some reference level. With these definitions, we can rewrite the momentum equation (1.38) as

$$\frac{\partial \boldsymbol{v}}{\partial t} + q\boldsymbol{k} \times \boldsymbol{v}^{*} + \nabla (\phi + K) = 0. \qquad \text{(B.1)}$$

The height equation (1.33) can be written

$$\frac{\partial}{\partial t}h^* + \nabla \bullet v^* = 0. \tag{B.2}$$

Next, we make some observations about this system of equations which aid in formulating the finite difference representation. We can derive equations for change in kinetic energy by evaluating $h^* [v \bullet (\mathrm{B.1})] + K(\mathrm{B.2})$, and for potential energy by evaluating $\phi(\mathrm{B.2})$. These equations are

$$\frac{\partial}{\partial t}(h^* K) + \nabla \bullet (v^* K) + v^* \bullet \nabla \phi = 0 \tag{B.3}$$

$$\frac{\partial}{\partial t}\left(\frac{gh^{*2}}{2} + gh^* h_s\right) + \nabla \bullet (v^* \phi) - v^* \bullet \nabla \phi = 0. \tag{B.4}$$

Adding (B.3) and (B.4) gives

$$\frac{\partial}{\partial t}\left[h^* \left(K + \frac{gh^*}{2} + gh_s\right)\right] + 2\nabla \bullet (v^* \phi) = 0. \tag{B.5}$$

When (B.5) is integrated over a closed domain, such as the surface of a sphere, or an infinite domain we get

$$\int_{\Sigma} \frac{\partial}{\partial t}\left[h^* \left(K + \frac{gh^*}{2} + gh_s\right)\right] d\Sigma = 0, \tag{B.6}$$

where the domain is denoted by Σ. According to (B.6) total energy is conserved. This was also shown when deriving the finite difference approximation of the stream function, velocity potential form of the shallow water equations. Since total energy is conserved in the continuous setting it should also be conserved in the finite difference setting. Also, the potential vorticity q is not involved in the change of total energy. This too should be true of the finite difference analog of the continuous kinetic energy equation.

The vorticity equation (1.39) can be written

$$\frac{\partial}{\partial t}(h^* q) + \nabla \bullet (v^* q) = 0. \tag{B.7}$$

Using (B.2), (B.7) can be written as the potential vorticity advection equation

$$\frac{\partial q}{\partial t} + v \bullet \nabla q = 0. \tag{B.8}$$

So, when there are no spatial gradients of q there should be no change in q. This property is also used in the design of the scheme. It is important to treat advection of potential vorticity well in order to accurately represent flow over steep topography.

Define potential enstrophy $Z \equiv (h^* q^2)/2$. Again, this is a slightly different definition than that presented in Chapter two. Evaluating $h^* q(\mathrm{B8}) + (q^2(\mathrm{B2}))/2$ gives a equation for the change in potential enstrophy

$$\frac{\partial}{\partial t}\left(\frac{h^* q^2}{2}\right) + \nabla \bullet \left(\frac{v^* q^2}{2}\right) = 0. \tag{B.9}$$

Integrating over a closed or infinite domain gives

$$\int_{\Sigma} \frac{\partial}{\partial t}\left(\frac{h^* q^2}{2}\right) d\Sigma = 0. \tag{B.10}$$

So, potential enstrophy is conserved in the continuous setting. This is an important quantity to conserve because, as Arakawa (1977) showed, conservation of energy alone does not prevent spurious energy cascades. That is, energy does not slowly creep into the higher wavenumbers as the integration progresses. However, conservation of potential enstrophy, coupled with conservation of total energy, prevents a false energy cascade.

Qualitatively, since enstrophy is the square of vorticity, an increase of energy on smaller scales would result in an increase in vorticity squared and an increase in enstrophy. Hence, if enstrophy is conserved, then energy cannot cascade.

B.4 The Discretized Height Equation

The variables u, v, h^* and q are discretized on a staggered grid known as the C grid. Arakawa (1977) showed this configuration of variables best simulates the geostrophic adjustment process. The distribution of variables on this grid is shown in Fig. B1.

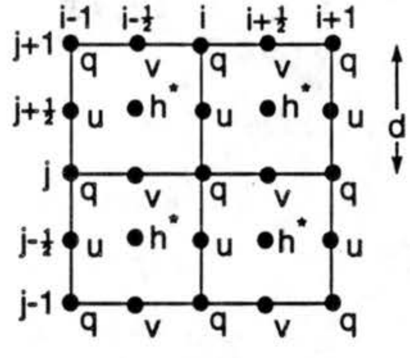

FIGURE B1: Distribution of variables on the Arakawa C grid.

Leaving time derivatives in continuous form, the second order discretized height equation has the form

$$\frac{\partial h^*_{i+1/2,j+1/2}}{\partial t} + (\nabla \bullet v^*)_{i+1/2,j+1/2} = 0, \qquad \textbf{(B.11)}$$

where

$$(\nabla \bullet v^*)_{i+1/2,j+1/2} \equiv \frac{1}{d}\left[u^*_{i+1,j+1/2} - u^*_{i,j+1/2} + v^*_{i+1/2,j+1} - v^*_{i+1/2,j}\right], \qquad \textbf{(B.12)}$$

and,

$$u^*_{i,j+1/2} \equiv [h^{(u)}u]_{i,j+1/2} \text{ and, } v^*_{i+1/2,j} \equiv [h^{(v)}v]_{i+1/2,j}, \tag{B.13}$$

where $h^{(u)}$ is height defined at u points and $h^{(v)}$ is height at v points. For an arbitrary quantity α, define the difference operators $\overline{\alpha}^i$ and $\overline{\alpha}^j$ as follows

$$(\overline{\alpha}^i)_{i,j} \equiv \frac{(\alpha_{i+1/2,j} + \alpha_{i-1/2,j})}{2} \text{ and, } (\overline{\alpha}^j)_{i,j} \equiv \frac{(\alpha_{i,j+1/2} + \alpha_{i,j-1/2})}{2}. \tag{B.14}$$

We will set

$$h^{(u)} \equiv (\overline{h}^i)_{i,j+1/2} \text{ and, } h^{(v)} \equiv (\overline{h}^j)_{i+1/2,j}. \tag{B.15}$$

B.5 The Discretized Momentum Equations

We have seen several features present in the continuous equations that should be built into the finite difference equation. We wish to construct a finite difference scheme which conserves total energy and potential enstrophy in the presence of a divergent mass flux. In addition, when q is constant in space, it should not change. We can write down a very general finite difference scheme in the plane for (B.1) with several undetermined parameters, and then fix the parameters by enforcing the above conservation constraints. The general fourth-order scheme for (B.1) can be written

$$\begin{aligned}
&\frac{\partial u_{i,j+1/2}}{\partial t} - \alpha_{i,j+1/2} v^*_{i+1/2,j+1} - \beta_{i,j+1/2} v^*_{i-1/2,j+1} \\
&\quad - \gamma_{i,j+1/2} v^*_{i-1/2,j} - \delta_{i,j+1/2} v^*_{i+1/2,j} \\
&\quad + \varepsilon_{i+1/2,j+1/2} u^*_{i+1,j+1/2} - \varepsilon_{i-1/2,j+1/2} u^*_{i-1,j+1/2} \\
&\quad + \lambda_{i,j+1} u^*_{i,j+3/2} - \lambda_{i,j} u^*_{i,j-1/2} \\
&+ [(K+\phi)_{i+1/2,j+1/2} - (K+\phi)_{i-1/2,j+1/2}]/d = 0,
\end{aligned} \tag{B.16}$$

and,

$$\begin{aligned}\frac{\partial v_{i+1/2,j}}{\partial t} &+ \gamma_{i+1,j+1/2}u^{*}{}_{i+1,j+1/2} + \delta_{i,j+1/2}u^{*}{}_{i,j+1/2}\\ &+ \alpha_{i,j-1/2}u^{*}{}_{i,j-1/2} + \beta_{i+1,j-1/2}u^{*}{}_{i+1,j-1/2}\\ &+ \varphi_{i+1/2,j+1/2}v^{*}{}_{i+1/2,j+1} - \varphi_{i+1/2,j-1/2}v^{*}{}_{i+1/2,j-1}\\ &+ \mu_{i+1,j}v^{*}{}_{i+3/2,j} - \mu_{i,j}v^{*}{}_{i-1/2,j}\\ &+ [(K+\phi)_{i+1/2,j+1/2} + (K+\phi)_{i+1/2,j-1/2}]/d = 0, \end{aligned} \tag{B.17}$$

where K at the h points is defined

$$K_{i+1/2,j+1/2} \equiv \frac{1}{2}\left[\overline{u^2}^{i} + \overline{v^2}^{j}\right]_{i+1/2,j+1/2}. \tag{B.18}$$

It is the addition of the μ and λ terms which distinguish this derivation from Arakawa (1981) and insure fourth-order advection of potential vorticity. From the form of (B.1), we see that the variables α, β, γ, δ, ε, φ, λ and μ must be linear combinations of q defined at the h points. Arakawa and Lamb (1981) show that this choice of K, (B.18), conserves total energy for a divergent mass flux, and that the finite difference form of the kinetic energy equation does not involve q. The forms of α, β, γ, δ, ε, φ, λ and μ are determined by applying the remaining constrains. These include the constraint that $\partial q/\partial t$ vanish when q is set equal to a constant, and the constraint that the scheme conserve potential enstrophy with a divergent mass flux. The derivation is certainly nontrivial, and the conclusions are stated here without proof:

$$\alpha_{i,j+1/2} = \frac{1}{24}[2q_{i+1,j+1} + 3q_{i,j+1} + 2q_{i,j} + q_{i+1,j} - q_{i,j+2} - q_{i-1,j+1}] \tag{B19}$$

$$\beta_{i,j+1/2} = \frac{1}{24}[3q_{i,j+1} + 2q_{i-1,j+1} + q_{i-1,j} + 2q_{i,j} - q_{i+1,j+1} - q_{i,j+2}] \tag{B20}$$

$$\gamma_{i,j+1/2} = \frac{1}{24}[2q_{i,j+1} + q_{i-1,j+1} + 2q_{i-1,j} + 3q_{i,j} - q_{i,j-1} - q_{i+1,j}] \quad \textbf{(B21)}$$

$$\delta_{i,j+1/2} = \frac{1}{24}[q_{i+1,j+1} + 2q_{i,j+1} + 3q_{i,j} + 2q_{i+1,j} - q_{i-1,j} - q_{i,j-1}] \quad \textbf{(B22)}$$

$$\varepsilon_{i+1/2,j+1/2} = \frac{1}{24}[q_{i+1,j+1} + q_{i,j+1} - q_{i,j} - q_{i+1,j}] \quad \textbf{(B23)}$$

$$\varphi_{i+1/2,j+1/2} = \frac{1}{24}[-q_{i+1,j+1} + q_{i,j+1} + q_{i,j} - q_{i+1,j}] \quad \textbf{(B24)}$$

$$\lambda_{i,j} = \frac{1}{24}[q_{i+1,j} - q_{i-1,j}] \quad \textbf{(B25)}$$

$$\mu_{i,j} = \frac{1}{24}[q_{i,j-1} - q_{i,j+1}] \quad \textbf{(B26)}$$

This derivation has been done on the plane. The scheme can also extended to a latitude/longitude spherical grid and still maintain conservation of total kinetic energy and potential enstrophy for a divergent mass flux.

Appendix C: The NCAR Spectral Transform Shallow Water Model

C.6 The Galerkin Spectral Method

In this section we briefly establish a framework in which to describe the spectral method. Consider a mixed initial-boundary value problem of the form

$$\frac{\partial}{\partial t}u(x,t) = L(x,t)u(x,t) \quad x \in \Omega, \tag{C.1}$$

$$B(x)u(x,t) = 0 \quad x \in \partial\Omega, \tag{C.1}$$

$$u(x,0) = g(x), \tag{C.2}$$

where Ω is a closed domain with boundary $\partial\Omega$. L is a linear differential operator and B is a linear boundary operator. Let the function $u(x,t)$ be a member of a Hilbert space denoted H. The Hilbert space is a set of functions coupled with an inner product (α, β) and a norm $\|\alpha\|$ for arbitrary $\alpha, \beta \in H$. Let $b \subset H$ be the set of all functions $u \in H$ which satisfy the boundary condition, $Bu = 0$. Let $\varphi_1, \varphi_2, \varphi_3, \ldots \in b$ be an orthonormal set of functions which span b. For numerical purposes, it is practical to use only a subset $b_N = \{\varphi_1, \varphi_2, \ldots, \varphi_N\}$ of b such that $b_N \subset b_M$ when $N < M$.

We can approximate u with a truncated series u_N of the form

$$u_N = \sum_{n=1}^{N} a_n(t)\varphi_n(x), \tag{C.3}$$

where the expansion coefficients are the orthogonal projection of u onto the subspace spanned by $\{\varphi_1, \varphi_2, \ldots, \varphi_N\}$. That is,

$$a_n = (u, \varphi_n) . \tag{C.4}$$

Generally, $Lu_N \notin b_N$, so we define L_N to be a linear operator from H to b_N. Now, we can write equation (C.(C.1)) as

$$\frac{\partial u_N}{\partial t} = L_N u_N. \tag{C.5}$$

Equation (C.5) is known as the truncated equation. Using (C.4) and (C.5) we get an expression for the evolution of the expansion coefficients $a_n(t)$ know as the are Galerkin equations

$$\frac{\partial}{\partial t}(u_N, \varphi_n) = (Lu_N, \varphi_n) \quad n = 1, \ldots, N. \tag{C.6}$$

Equation (C.6) can be written

$$\sum_{m=1}^{N} (\varphi_m, \varphi_n) \frac{da_m}{dt} = \sum_{m=1}^{N} a_m (L\varphi_m, \varphi_n). \tag{C.7}$$

Since, $\{\varphi_1, \varphi_2, \ldots, \varphi_N\}$ are assumed orthonormal, (C.7) can be written

$$\frac{da_n}{dt} = \sum_{m=1}^{N} a_m (L\varphi_m, \varphi_n). \tag{C.8}$$

In solving (C.8), we are really performing a least squares minimization to find the quantities da_n/dt $(n = 1, \ldots, N)$ which minimize

$$\left(\frac{\partial u_N}{\partial t} - Lu_N, \frac{\partial u_N}{\partial t} - Lu_N\right) . \tag{C.9}$$

Hence, the main idea behind spectral methods is to decompose the system (C.(C.1)), (C.1) and (C.2) into a system of ordinary differential equations.

C.7 Description of the Model

C.7.1 The Equations Used in The Model

The equations integrated in the model are the vorticity and divergence equations, (1.39) and (1.40), from Chapter 1:

$$\frac{\partial \zeta}{\partial t} = -\nabla \bullet (\eta \boldsymbol{v}), \tag{C.10}$$

$$\frac{\partial \delta}{\partial t} = \hat{\boldsymbol{k}} \bullet \nabla \times (\eta \boldsymbol{v}) - \nabla^2 \left(gh + \frac{\boldsymbol{v} \bullet \boldsymbol{v}}{2}\right). \tag{C.11}$$

In the continuity equation, we will partition the geopotential into two parts, a time-independent spatial mean denoted by $\bar{\phi} = g\bar{h}$, and a time-dependent perturbation part denoted ϕ', so, $\phi = \phi' + \bar{\phi}$. Then (1.32) is given by

$$\frac{\partial}{\partial t}\phi' = -\nabla \bullet (\phi \boldsymbol{v}) = -\nabla \bullet (\phi' \boldsymbol{v}) - \bar{\phi}\delta. \tag{C.12}$$

In order to satisfy a linear CFL condition, the explicit time step is limited by the phase speed of the fastest wave in the model, in this case, the phase speed of the gravity waves. However, this partitioning allows the higher-phase-speed gravity waves to be time stepped implicitly, and only the slower-phase-speed meteorological waves to be time stepped explicitly. A much longer time step can be used without violating the CFL condition. This procedure is discussed further in Hack (1992).

Define $U \equiv u\cos\theta$, $V \equiv v\cos\theta$ and $\mu \equiv \sin\theta$. With these definitions, rewriting (C.10), (C.11) and (C.12) in spherical coordinates gives

$$\frac{\partial\eta}{\partial t} = -\frac{1}{a(1-\mu^2)}\frac{\partial}{\partial\lambda}(U\eta) - \frac{1}{a}\frac{\partial}{\partial\mu}(V\eta), \tag{C.13}$$

$$\frac{\partial\delta}{\partial t} = \frac{1}{a(1-\mu^2)}\frac{\partial}{\partial\lambda}(V\eta) - \frac{1}{a}\frac{\partial}{\partial\mu}(U\eta) - \nabla^2\left(\phi + \frac{U^2+V^2}{2(1-\mu^2)}\right), \tag{C.14}$$

$$\frac{\partial\phi}{\partial t} = \frac{1}{a(1-\mu^2)}\frac{\partial}{\partial\lambda}(U\phi) - \frac{1}{a}\frac{\partial}{\partial\mu}(V\phi) - \bar{\phi}\delta. \tag{C.15}$$

C.7.2 Transformations to Spectral Space

An arbitrary scalar quantity $\xi(\theta, \lambda)$ can be approximated with a truncated series of spherical harmonics as follows:

$$\xi(\theta, \lambda) \approx \sum_{m=-M}^{M} \sum_{n=|m|}^{N(m)} \xi_n^m P_n^m(\mu) e^{im\lambda}. \tag{C.16}$$

Here, $P_n^m(\mu)$ is the associated Legendre polynomial as discussed in chapter 2. The NCAR spectral model allows for both triangular and rhomboidal truncation.

Analogous to equation (C.4), the spectral components, ξ_n^m, are determined by

$$\xi_n^m = \int_{-1}^{1} \frac{1}{2\pi} \int_{0}^{2\pi} \xi(\mu, \lambda)\, e^{-im\lambda} d\lambda P_n^m(\mu) d\mu. \tag{C.17}$$

The zonal integral is a Fourier transform which can be approximated with a discrete Fourier transform as follows:

$$\xi^m(\mu) = \frac{1}{2\pi}\int_0^{2\pi} \xi(\mu, \lambda)\, e^{-im\lambda} d\lambda \approx \frac{1}{I}\sum_{I=1}^{I} \xi(\mu, \lambda_i)\, e^{im\lambda_i}, \qquad \text{(C.18)}$$

where $\lambda_i = 2\pi i/I$ and I is the number of grid points in the zonal direction. The discrete Fourier transform can be efficiently computed with a Fast Fourier Transform. The meridional integral is

$$\xi_n^m = \int_{-1}^{1} \xi^m(\mu) P_n^m(\mu) d\mu \approx \sum_{j=1}^{J} \xi^m(\mu_j) P_n^m(\mu_j) w_j. \qquad \text{(C.19)}$$

This integral is approximated with Gaussian quadrature where J is the number of Gauss points from pole to pole. The Gauss points, μ_j, are the roots of the associated Legendre polynomial $P_J^0(\mu)$, and the weights, w_j, are determined by

$$w_j = \frac{2(1-\mu_j^2)}{[Jp_{J-1}(\mu_j)]^2}. \qquad \text{(C.20)}$$

C.7.3 Treatment of Quadratic Terms

To this point in the discussion, we have only dealt with linear operations on the field variables. Equations (C.13) and (C.14) contain several quadratic terms. Two approaches are used to treat such products of terms. Initially, spectral models used a method based on interaction coefficients. This method can be explained by considering the product two scalar fields $C = AB$. Spectral expansions of A and B can be written

$$A(\mu, \lambda) = \sum_{m=-M}^{M} \sum_{n=|m|}^{N(m)} A_n^m P_n^m(\mu)\, e^{im\lambda}, \qquad \text{(C.21)}$$

$$B(\mu,\lambda) = \sum_{m=-M}^{M} \sum_{n=|m|}^{N(m)} B_n^m P_n^m(\mu)\, e^{im\lambda}. \tag{C.22}$$

Using (C.17), we can write

$$C_n^m = \int_{-1}^{1} \frac{1}{2\pi} \int_0^{2\pi} ABe^{-im\lambda} d\lambda P_n^m(\mu) d\mu. \tag{C.23}$$

Substitution of (C.21) and (C.22) into (C.23) gives

$$C_n^m = \sum_{p=-M}^{M} \sum_{q=|p|}^{N(p)} \sum_{r=-M}^{M} \sum_{s=|r|}^{N(r)} I_{qsn}^{prm} A_q^p B_s^r, \tag{C.24}$$

where the terms I_{qsn}^{prm} are called interaction coefficients. Even though the I_{qsn}^{prm} can be computed and stored prior to running the model, the method is extremely costly. In fact, Jarraud (1983) claims that the number of operations per time step increases as $O(N^5)$. The method of interaction of coefficients seemed, initially, to make spectral models impractical. However, the second method, the transform method, proved to work much better.

In the transform method, the fields A and B are transformed from spectral space via (C.16). They are then simply multiplied together in grid point space and transformed back to spectral space. In this process, care must taken to avoid aliasing errors. That is, by using a sufficiently large number of grid points in the zonal and meridional directions, the transforms (C.18) and (C.19) can be performed to give nonaliased results for quadratic terms. The number of grid points in the east-west direction I must satisfy $I \geq 3M+1$, and for triangular truncation in the north-south directions J must satisfy

$J \geq (3M+1)/2$. The transform method requires $O(N^3)$ operations per time step, a substantial savings over the interaction coefficient method.

C.7.4 Differentiation of Terms

Equations (C.13), (C.14) and (C.15) contain longitudinal and latitudinal differentiation, as well as the Laplacian operator. In this section, we derive expressions for the expansion coefficients of these operators applied to an arbitrary function. With (C.18) and integration by parts using cyclic boundary conditions, we can write

$$\left(\frac{\partial\xi}{\partial\lambda}\right)^m = \frac{1}{2\pi}\int_0^{2\pi}\frac{\partial\xi}{\partial\lambda}e^{-im\lambda}d\lambda = im\left[\frac{1}{2\pi}\int_0^{2\pi}\xi e^{-im\lambda}d\lambda\right] = im\xi^m. \tag{C.25}$$

The computational sequence for longitudinal differentiation first Fourier transforms and then multiplies by im in (C.25), and lastly does a meridional transform to give

$$\left(\frac{1}{a(1-\mu^2)}\frac{\partial\xi}{\partial\lambda}\right)^m_n = \sum_{j=1}^{J} im\xi^m(\mu_j)\frac{P_n^m(\mu_j)}{a(1-u_j^2)}w_j. \tag{C.26}$$

Unfortunately, the spherical harmonics are not eigenfunctions of $\partial(\ )/\partial\mu$. We can, however, use a recurrence relation for the associated Legendre polynomials to write

$$H_n^m(\mu) \equiv (1-\mu^2)\frac{dP_n^m}{d\mu} = (n+1)\varepsilon_n^m P_{n-1}^m - n\varepsilon_{n+1}^m P_{n+1}^m, \tag{C.27}$$

where

$$\varepsilon_n^m = \sqrt{\frac{n^2-m^2}{4n^2-1}}. \tag{C.28}$$

Analogous to (C.25), assuming $\xi^m(\mu) = 0$ at the poles, meridional differentiation can be written

$$\left(\frac{1}{a}\frac{\partial \xi}{\partial \lambda}\right)_n^m = \int_{-1}^{1} \frac{1}{a}\frac{\partial \xi^m}{\partial \mu} P_n^m(\mu) d\mu = -\int_{-1}^{1} \frac{1}{a} \xi^m \frac{H_n^m(\mu)}{1-\mu^2} d\mu. \quad \text{(C.29)}$$

Using (C.19), we get

$$\left(\frac{1}{a}\frac{\partial \xi}{\partial \mu}\right)_n^m = -\sum_{j=1}^{J} \xi^m(\mu_j) \frac{H_n^m(\mu_j)}{a(1-\mu^2)} w_j. \quad \text{(C.30)}$$

As shown in Chapter 1, the spherical harmonics are eigenfunctions of the Laplacian operator on the sphere, that is,

$$\nabla^2 P_n^m(\mu) e^{im\lambda} = -\frac{n(n+1)}{a^2} P_n^m(\mu) e^{im\lambda}. \quad \text{(C.31)}$$

So, in the same way (C.30) is derived for the meridional derivative, the expansion coefficients of the Laplacian of a function are given by

$$(\nabla^2 \xi)_n^m = -\frac{n(n+1)}{a^2} \sum_{j=1}^{J} \xi^m(\mu_j) P_n^m(\mu_j) w_j. \quad \text{(C.32)}$$

C.7.5 Spectral Representation of Equations

Following Hack (1992), we can redefine the quadratic terms in equations (C.13), (C.14) and (C.15) as follows

$$A \equiv U\eta,\ B \equiv V\eta,\ C \equiv U\phi,\ D \equiv V\phi \text{ and } E \equiv \frac{U^2+V^2}{2(1-\mu^2)}. \quad \text{(C.33)}$$

So, substitution of (C.33) into (C.13), (C.14) and (C.15) gives

$$\frac{\partial \eta}{\partial t} = -\frac{1}{a\,(1-\mu^2)}\frac{\partial A}{\partial \lambda} - \frac{1}{a}\frac{\partial B}{\partial \mu}, \tag{C.34}$$

$$\frac{\partial \delta}{\partial t} = \frac{1}{a\,(1-\mu^2)}\frac{\partial B}{\partial \lambda} - \frac{1}{a}\frac{\partial A}{\partial \mu} - \nabla^2\,(\phi + E)\,, \tag{C.35}$$

$$\frac{\partial \phi}{\partial t} = -\frac{1}{a\,(1-\mu^2)}\frac{\partial C}{\partial \lambda} - \frac{1}{a}\frac{\partial D}{\partial \mu} - \bar{\phi}\delta. \tag{C.36}$$

Finally, analogous to (C.8), and using (C.26), (C.30) and (C.32), we can rewrite equations (C.34), (C.35) and (C.36) as follows:

$$\frac{\partial \eta_n^m}{\partial t} = -\sum_{j=1}^{J}\frac{w_j}{a\,(1-\mu_j^2)}\,[imA^m(\mu_j)P_n^m(\mu_j) - B^m(\mu_j)H_n^m(\mu_j)]\,, \tag{C.37}$$

$$\frac{\partial \zeta_n^m}{\partial t} = -\sum_{j=1}^{J}\frac{w_j}{a\,(1-\mu_j^2)}\,[-imB^m(\mu_j)P_n^m(\mu_j) - A^m(\mu_j)H_n^m(\mu_j)] \tag{C.38}$$

$$+\frac{n\,(n+1)}{a^2}\sum_{j=1}^{J}P_n^m(\mu_j)w_j\,[E^m(\mu_j) + \phi^m(\mu_j)]\,,$$

$$\frac{\partial \phi_n^m}{\partial t} = -\sum_{j=1}^{J}\frac{w_j}{a\,(1-\mu_j^2)}\,[imC^m(\mu_j)P_n^m(\mu_j) - D^m(\mu_j)H_n^m(\mu_j)] - \bar{\phi}\delta_n^m, \tag{C.39}$$

where $A^m(\mu_j)$, $B^m(\mu_j)$, $C^m(\mu_j)$, $D^m(\mu_j)$ and $E^m(\mu_j)$ are given by (C.18).

C.7.6 Diagnostic Equations

Equations (C.34), (C.35) and (C.36) do not predict new values for U and V, so we need diagnostic equations to get U and V from η and δ. This is done by first computing two intermediate variables, stream function and velocity potential. In continuous form ψ and χ are related to η and δ by

$$\nabla^2\psi = \eta - f, \tag{C.40}$$

$$\nabla^2 \chi = \delta. \tag{C.41}$$

Then U and V are derived from ψ and χ with $V = \hat{k} \times \nabla \psi + \nabla \chi$. That is,

$$U = \frac{1}{a}\frac{\partial \chi}{\partial \lambda} - \frac{(1-\mu^2)}{a}\frac{\partial \psi}{\partial \mu}, \tag{C.42}$$

$$V = \frac{1}{a}\frac{\partial \psi}{\partial \lambda} + \frac{(1-\mu^2)}{a}\frac{\partial \chi}{\partial \mu}. \tag{C.43}$$

According to (C.16), ψ and χ can be approximated with a truncated series of spectral harmonics as follows

$$\psi(\mu, \lambda) \approx \sum_{m=-M}^{M} \sum_{n=|m|}^{N(m)} \psi_n^m P_n^m(\mu) e^{im\lambda}, \tag{C.44}$$

$$\chi(\mu, \lambda) \approx \sum_{m=-M}^{M} \sum_{n=|m|}^{N(m)} \chi_n^m P_n^m(\mu) e^{im\lambda}. \tag{C.45}$$

Since only the derivatives of stream function and velocity potential are ever needed, the global averages of these quantities ψ_0^0 and χ_0^0 can be arbitrarily chosen. Equations (C.44) and (C.45) can be rewritten using (C.40), (C.41) and (C.31), and the fact that the spectral harmonics are orthogonal as

$$\psi(\mu, \lambda) \approx - \sum_{m=-M}^{M} \sum_{n=|m|}^{N(m)} \frac{a^2}{n(n+1)} (\eta_n^m - f_n^m) P_n^m(\mu) e^{im\lambda}, \tag{C.46}$$

$$\chi(\mu, \lambda) \approx - \sum_{m=-M}^{M} \sum_{n=|m|}^{N(m)} \frac{a^2}{n(n+1)} \delta_n^m P_n^m(\mu) e^{im\lambda}. \tag{C.47}$$

It can be shown that a spectral representation of the Coriolis parameter is given by

$$f_n^m = \begin{cases} \dfrac{\Omega}{\sqrt{0.375}} & \text{for } n=1 \text{ and } m=0 \\ 0 & \text{otherwise.} \end{cases} \quad \textbf{(C.48)}$$

Next, we derive an expression for U and V in term of the expansion coefficients η_n^m and δ_n^m. Zonal and meridional differentiation of (C.44) and (C.45) gives

$$\frac{\partial\psi}{\partial\mu} \approx -\sum_{m=-M}^{M} \sum_{n=|m|}^{N(m)} \frac{a^2}{n(n+1)} (\eta_n^m - f_n^m) \frac{\partial}{\partial\mu} P_n^m(\mu) e^{im\lambda}, \quad \textbf{(C.49)}$$

$$\frac{\partial\chi}{\partial\mu} \approx -\sum_{m=-M}^{M} \sum_{n=|m|}^{N(m)} \frac{a^2}{n(n+1)} \delta_n^m \frac{\partial}{\partial\mu} P_n^m(\mu) e^{im\lambda}, \quad \textbf{(C.50)}$$

$$\frac{\partial\psi}{\partial\lambda} \approx -\sum_{m=-M}^{M} \sum_{n=|m|}^{N(m)} im \frac{a^2}{n(n+1)} (\eta_n^m - f_n^m) P_n^m(\mu) e^{im\lambda}, \quad \textbf{(C.51)}$$

$$\frac{\partial\chi}{\partial\lambda} \approx -\sum_{m=-M}^{M} \sum_{n=|m|}^{N(m)} im \frac{a^2}{n(n+1)} \delta_n^m P_n^m(\mu) e^{im\lambda}. \quad \textbf{(C.52)}$$

Substitution of (C.49), (C.50), (C.51), and (C.52) into (C.42) and (C.43) produces

$$U(u_j, \lambda_i) = -\sum_{m=-M}^{M} \sum_{n=|m|}^{N(m)} \frac{a}{n(n+1)} [im\delta_n^m P_n^m(\mu_j) - (\eta_n^m - f_n^m) H_n^m(\mu_j)] e^{im\lambda}, \quad \textbf{(C.53)}$$

$$V(u_j, \lambda_i) = -\sum_{m=-M}^{M} \sum_{n=|m|}^{N(m)} \frac{a}{n(n+1)} [im(\eta_n^m - f_n^m) P_n^m(\mu_j) + \delta_n^m H_n^m(\mu_j)] e^{im\lambda}. \quad \textbf{(C.54)}$$

These representations of U and V in grid point space can now be used to compute the quadratic terms in (C.33) with the spectral transform method. The products are then transformed to spectral space via Fast Fourier Transform and Gaussian quadrature as in (C.18) and (C.19). All spatial derivatives are evaluated in spectral space, and (C.37), (C.38) and (C.39) are time stepped. The new η_n^m and δ_n^m are used in (C.53) and (C.54), and so on.

C.8 Software Availability

The software for the NCAR spectral transform shallow water model is available via anonymous FTP from the machine `ftp.ucar.edu`, IP address 128.117.64.4. The source code for the model resides in subdirectory `/chammp/shallow/src`. In the subdirectory `/chammp/shallow` there is a document called `descrition.txt` that is very useful for compiling and running the model. The source code contains a `makefile` that enables the program to be compiled on variety of different machines. The netCDF files described in Chapter 3 are also available from Rudy Jakob at NCAR.

REFERENCES

Arakawa, A., and V. R. Lamb, 1977: Computational design of the basic dynamical processes of the UCLA general circulation model. *Methods in Computational Physics*, **17,** Academic Press, 173-265.

-----, and -----, 1981: A Potential Enstrophy and Energy Conserving Scheme for the Shallow Water Equations. *Mon. Wea. Rev.*, **109,** 18-36.

Augenbaum, J. M., and C.S. Peskin , 1985: On the Construction of the Voronoi Mesh on a Sphere. *J. Comput. Phy.*, **14,** 177-192.

Beyer, W. H., 1984: *CRC Standard Mathematical Tables, 27th Edition*. CRC Press Inc., 301-305.

Brandt, A., 1977: Multi-Level Adaptive Solution to Boundary-Value Problems. *Math Comp.*, **31,** 333-390.

Browning, G. L., J. J. Hack, and P. N. Swarztrauber, 1989: A Comparison of Three Numerical Methods for Solving Differential Equations on the Sphere. *Mon. Wea. Rev.*, **117,** 1058-1075.

Canuto, C., M. Y. Hussaini, A. Quarteroni and T. A. Zang, 1988: *Spectral Methods in Fluid Dynamics*. Springer-Verlag.

Chung, F., and S. Sternberg, 1993: Mathematics and the Buckyball, *American Scientist*, **81,** 56-71.

Fulton, S., P. Ciesielshi, and W. Schubert, 1986: Multigrid Methods for Elliptic Problems: A Review. *Mon. Wea. Rev.*, **114,** 943-957.

Gill, A., 1982: *Atmosphere-Ocean Dynamics*. Academic Press.

Gottlieb, D., and S. A. Orszag, 1977: *Numerical Analysis of Spectral Methods: Theory and Applications*. S.I.A.M.

Hoskins, B. J., and A. Hollingsworth, 1973: On the Simplest Example of thee Baratropic Instability of Rossby Wave Motion. *J. Atmos. Sci.*, **30,** 150-153.

Jarraud, M., and A. J. Simmons, 1983: The Spectral Method. Seminar 1983, Numerical Methods for Weather Prediction, European Centre for Medium Range Weather Forecasts.

Kurihara, Y., 1969: A Finite Difference Scheme by Making Use of the Primitive Equations of a Spherical Grid. *Mon. Wea. Rev.*, **93,** 399-415.

Lorenz, E. N., 1972: Barotropic Instability of Rossby Wave Motion. *J. Atmos. Sci.*, **29,** 258-264.

Masuda, Y., and H. Ohnishi, 1986: An Integration Scheme of the Primitive Equations Model with an Icosahedral-Hexagonal Grid System and its Application to the Shallow Water Equations. *Short- and Medium-Range Numerical Weather Prediction.* Japan Meteorological Society, Tokyo, 317-326.

Pedlosky, J., 1986: *Geophysical Fluid Dynamics.* Springer-Verlag.

Sadourny, R., A. Arakawa, and Y. Mintz, 1968: Integration of the Nondivergent Barotropic Vorticity Equation with an Icosahedral-Hexagonal Grid for the Sphere. *Mon. Wea. Rev.*, **96,** 351-356.

-----, and P. Morel, 1969: A Finite Difference Approximation of the Primitive Equations for a Hexagonal Grid on a Plane. *Mon. Wea. Rev.*, **97,** 439-445.

Takacs, L. L., 1988: Effects of Using a Posteriori Methods for the Conservation of Integral Invariants. *Mon. Wea. Rev.*, **116,** 525-545.

Takano, K., 1982: A Fourth-Order Energy and Potential Enstrophy Conserving Difference Scheme. *Air Force Geophysical Laboratory.* Report Number AFGL-TR-82-0205.

Williamson, D. L., 1968: Integration of the Barotropic Vorticity Equation on a Spherical Geodesic Grid. *Tellus,* **20,** 642-653.

-----, 1970: Integration of the Primitive Barotropic Model over a Spherical Geodesic Grid. *Mon. Wea. Rev.*, **98,** 512-520.

-----, Numerical Methods Used in Atmospheric Models, GARP Pub. Ser. No. 17 (JOC, WMO, Geneva, 1979), Chap. 2, 51-120.

-----, J. B. Drake, J. J. Hack, R. Jakob, and P. N. Swarztrauber, 1992: A Standard Test Set for Numerical Approximatios to the Shallow Water Equations in Spherical Geometry. *J. Comput. Phy.*, **102,** 221-224.

Wolfram, S., 1991: *Mathematica: A System for Doing Mathematics by Computer.* Addison-Wesley.