*CR — 1999 - 209231*

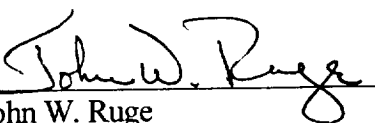# A Nonlinear Multigrid Solver for an Atmospheric General Circulation Model Based on Semi-implicit Semi-Lagrangian Advection of Potential Vorticity
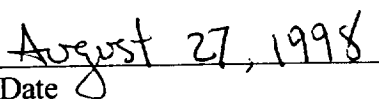
## Final Report

John W. Ruge
Principal Investigator
Front Range Scientific Computations, Inc.
1390 Claremont Dr.
Boulder, CO 80303

Final Report on Contract NAS596076

Submitted to
NASA/Goddard Space Flight Center
NASA Office of Mission to Planet Earth

John W. Ruge

August 27, 1998
Date

# 1. Introduction

This work represents part of a project to develop an atmospheric general circulation model based on the semi-Lagrangian advection of potential vorticity (PV), with divergence as the companion prognostic variable. The resulting model derived by Bates and Li is hence termed the PV-D model. The motivation for the use of potential vorticity as a prognostic variable is that it is the most basic dynamical quantity (see [5]) and is conserved along a trajectory in frictionless adiabatic flow. It is reasonable to assume that, particularly in situations of highly nonlinear flow, use of the evolution equation for PV as one of the governing equations will give a more accurate simulation of the PV evolution than a model using velocity components or vorticity and divergence as the basic prognostic variables. An important consideration in the development and time discretization of the PV-D model was the idea that a multigrid (MG) solver could be developed for efficient solution of nonlinear implicit equations obtained at each time step. The focus of this part of the project was the development and testing of the multigrid solver.

In this report, the continuous and discrete equations will be presented, ending with the system to be solved by multigrid at each time step. The development of the MG solver will be outlined, the algorithm described in detail, and results on model problems will be presented.

# 2. The Model Equations

Here, the equations will be presented. Much of the detail and motivation for choosing the model is outside the scope of this report and will not be described in detail. In the model, a generalized vertical coordinate $\xi$ is used, rather than an isentropic vertical coordinate. This hybrid coordinate basically is related to the ratio of pressure to surface pressure in the lower atmosphere (and thus is constant on the surface) while it is isentropic (follows layers of constant temperature) in the upper atmosphere. The reason for this is to eliminate problems at the lower boundary layers where stratification may become nearly neutral. Closely related to the above, potential vorticity is replaced by a quantity referred to as pseudo potential vorticity (PPV) and it is this quantity that is advected. In the upper layers, where the vertical coordinate is isentropic, this quantity is basically potential vorticity and is conserved. This

2

quantity is not exactly conserved in the lower layers. However, it can be argued that the isentropic PV does not have much dynamical significance there. Physics is also included in the equations as moisture.

## 2.1 The Continuous Equations.

A longitude-latitude coordinate system is used in combination with the vertical coordinate, giving a system $(\lambda, \phi, \xi)$. $\lambda \in [0,2\pi]$ is longitude, $\phi \in [-\pi/2,\pi/2]$ is latitude (with $\phi = -\pi/2$ as the south pole and $\phi = \pi/2$ as the north pole), and $\xi$ is the vertical coordinate, with bounds depending on the definition chosen. There are eight primary unknown functions defined on this domain:

| | |
|---|---|
| $p$ | Pressure. |
| $\psi$ | Streamfunction. |
| $\chi$ | Velocity Potential. |
| $M$ | Montgomery Streamfunction. |
| $\theta'$ | Potential Temperature perturbation. (Potential temperature $\theta$ is written as $\bar{\theta} + \theta$, here $\bar{\theta}(\xi)$ is a given mean potential temperature and $\theta'(\lambda, \phi, \xi)$ is the perturbation.) |
| $\dot{\xi}$ | Vertical velocity. |
| $\omega$ | Pseudo potential vorticity. |
| $q$ | Specific humidity. |

Physical constants that appear in the equations are defined as follows:

| | |
|---|---|
| $a$ | Radius of the earth ($6.371229 \cdot 10^6$ m). |
| $\Omega$ | Rotation rate of the earth ($7.292 \cdot 10^{-5}$ 1/s). |
| $C_\kappa$ | 0.28704/1.005 (non-dimensional). |
| $C_p$ | $1.005 \cdot 10^3$ J/kg·K. |
| $p_{00}$ | $10^5$ Pascal (N/m$^2$). |
| $R$ | Universal gas constant (287 J ($^\circ$K)$^{-1}$ Kg$^{-1}$. |
| $\kappa$ | $R/C_p$. |

Other variable quantities are:

$\Pi$      Defined as $C_p(p/p_{00})^\kappa$ and used in the equations for convenience.

$f$      Coriolis force ($2 \Omega \sin\phi$).

$\beta$      $1/a \, df/d\phi = 2\Omega \cos\phi /a$.

The quantity $\beta$ is related to the semi-Lagrangian discretization and will be discussed further in the next section. As noted below, the "continuous" form of the divergence equation given here is already discretized in time, and includes $\beta$.

In the following, $d/dt$ denotes the Lagrangian, or material, derivative (along a particle path). $\nabla$ is the horizontal gradient operator

$$\nabla = \left( \frac{1}{a\cos\phi}\frac{\partial \cdot}{\partial\lambda}, \frac{1}{a}\frac{\partial \cdot}{\partial\phi} \right)^T , \tag{2.1}$$

where $a$ is the radius of the sphere. $\nabla^2$ is the horizontal Laplacian:

$$\nabla^2 = \frac{1}{a^2\cos^2\phi}\frac{\partial^2}{\partial\lambda^2} + \frac{1}{a^2}\frac{\partial^2}{\partial\phi^2} \tag{2.2}$$

Although the more familiar quantities of velocity and divergence are not explicitly included in the equations here, they are important and it is useful to see how they relate to the unknowns used. A Helmholtz decomposition of horizontal velocity is used to obtain the streamfunction $\psi$ and velocity potential $\chi$, so that $u$ and $v$ satisfy

$$u = \frac{1}{a\cos\phi}\frac{\partial\chi}{\partial\lambda} - \frac{1}{a}\frac{\partial\psi}{\partial\phi},$$
$$v = \frac{1}{a}\frac{\partial\chi}{\partial\phi} + \frac{1}{a\cos\phi}\frac{\partial\psi}{\partial\lambda}. \tag{2.3}$$

The horizontal divergence $D$ satisfies

$$D = \nabla \cdot (u,v)^T = \nabla^2\chi. \tag{2.4}$$

Now, the set of continuous equations considered here can be presented.

*The PPV equation*

$$\frac{d\omega}{dt} = \omega\frac{\partial\dot{\xi}}{\partial\xi}$$

$$+\left(\frac{1}{\partial p/\partial\xi}\right)\left[\frac{1}{a\cos\phi}\frac{\partial\dot{\xi}}{\partial\lambda}\frac{\partial}{\partial\xi}\left(\frac{1}{a}\frac{\partial\chi}{\partial\phi}+\frac{1}{a\cos\phi}\frac{\partial\psi}{\partial\lambda}\right)-\frac{1}{a}\frac{\partial\dot{\xi}}{\partial\phi}\frac{\partial}{\partial\xi}\left(\frac{1}{a\cos\phi}\frac{\partial\chi}{\partial\lambda}-\frac{1}{a}\frac{\partial\psi}{\partial\phi}\right)\right]$$

$$-\left(\frac{1}{\partial p/\partial\xi}\right)\left(\frac{1}{a\cos\phi}\right)\left[\frac{\partial\Pi}{\partial\lambda}\frac{\partial(1+0.608q)\theta}{\partial\phi}-\frac{\partial\Pi}{\partial\phi}\frac{\partial(1+0.608q)\theta}{\partial\lambda}\right]$$

$$-\left(\frac{1}{\partial p/\partial\xi}\right)S_1$$

(2.5)

Here, $S_1$ is a term due to turbulent friction.

*The Divergence Equation*

The divergence equation is given here discretized in time. This is because the equation is derived by first performing a semi-Lagrangian time discretization of the horizontal momentum equations, then taking the divergence. An uncentering parameter is used and $\tau_1$ depends on this parameter and the timestep $\Delta t$. Superscript $n+1$ denotes values at the new time step.

$$\left(\nabla^2\chi\right)^{n+1}+\tau_1\left(\nabla^2 M\right)^{n+1}-\tau_1\Pi^{n+1}\left(\nabla^2\theta\right)^{n+1}-\tau_1(\nabla\Pi)^{n+1}\cdot(\nabla\theta)^{n+1}$$

$$+\tau_1 0.608\nabla(\theta q)^{n+1}\cdot(\nabla\Pi)^{n+1}+\tau_1 0.608(\theta q)^{n+1}\cdot\left(\nabla^2\Pi\right)^{n+1}$$

$$+\beta\tau_1\left(\frac{1}{a\cos\phi}\frac{\partial\chi}{\partial\lambda}-\frac{1}{a}\frac{\partial\psi}{\partial\phi}\right)^{n+1}-f\tau_1\left(\nabla^2\psi\right)^{n+1}$$

$$= forcing2 = r_2+S_2\Delta t.$$

(2.6)

*The Hydrostatic Equation*

$$\frac{\partial M}{\partial\xi}-\Pi\frac{\partial\theta}{\partial\xi}+0.608\left(\frac{R}{p_{00}}\right)\left[q\theta\left(\frac{p}{p_{00}}\right)^{\kappa-1}\right]\frac{\partial p}{\partial\xi}=0.$$

(2.7)

*The Continuity Equation*

$$\frac{d}{dt}\left(\frac{\partial p}{\partial\xi}\right)+\left(\frac{\partial p}{\partial\xi}\right)\left(\nabla^2\chi+\frac{\partial\dot{\xi}}{\partial\xi}\right)=0.$$

(2.8)

5

*The Thermodynamic Equation:*

$$\frac{d\theta}{dt} = \hat{Q}_1 = \frac{Q_1}{\mu_1 \Pi} + \mu_2.$$

(2.9)

*The Definition of $\xi$:*

$$\xi(p, p_s, \theta) = \xi.$$

(2.10)

*The Definition of $\omega$:*

$$\omega = -\frac{f + \nabla^2 \psi}{\partial p / \partial \xi}.$$

(2.11)

*The Moisture Equation:*

$$\frac{dq}{dt} = -\frac{Q_2}{L},$$

(2.12)

where $Q_2$ is the apparent moisture sink and $L$ is the latent heat of condensation for water vapor.

**Boundary Conditions.** The boundary conditions needed to close the system are as follows:

$$\dot{\xi}(\xi_0) = \dot{\xi}(\xi_{max}) = 0,$$

(2.13)

$$M(\xi_0) = \Phi(\lambda, \phi) + (\overline{\theta}(\xi_0) + \theta'(\xi_0))\Pi(\xi_0),$$

(2.14)

and

$$p(\xi_{max}) = p_T(\lambda, \phi).$$

(2.15)

Here, $p_T$ is a given function, generally taken as constant in tests here.

## 3. The SLSI Discretization

The form of the equations used (particularly the divergence equation) is tied to the use of a semi-Lagrangian vector discretization. The idea of semi-Lagrangian discretization is to express the material derivative of a function in terms of the difference of the function's value at a point at the current time step (the "arrival" point) and at the corresponding "departure" point at the previous time step, obtained by backtracking along the trajectory. In practice, the departure points cannot be determined exactly,

since these depend on the new, undetermined velocities, and exact integration along trajectories may not be possible. Instead, the departure points are approximated by assuming that the average velocity along a trajectory is its value at the midpoint, and obtain these midpoint values by spatial interpolation of time-extrapolated velocities from previous timesteps. To make this more precise, suppose that a time step $\Delta t$ has been chosen, and the solution is known at time $n\Delta t$. A two time-level semi-implicit scheme is used, giving equations relating the solutions at times $n\Delta t$ and $(n+1)\Delta t$. The time at which a quantity is defined is denoted by superscripts. Now, let $x$ be a point at time level $n+1$, and let $x_*$ denote the corresponding departure point at time level $n$. Using the trajectory midpoint approximation for velocity $V$ gives $x_*=x-\Delta t\ V^{n+1/2}((x+x_*)/2)$, where velocities at time step $n+1/2$ are obtained by forward extrapolation in time, with $V^{n+1/2}=1.5V^n-0.5V^{n-1}$. This implicitly defines $x_*$. Tthe superscript $n+1$ denotes function values at arrival points at time level $n+1$, while $(.)^n_*$ denotes the function values at the corresponding departure point at time level $n$. When the spatial discretization is introduced, the level $n+1$ values will be defined at the grid points, while the departure point values will be obtained by interpolation to the departure point at time level $n$.

A straightforward semi-Lagrangian discretization of the divergence equation leads to Rossby wave instability (see [2]). For this reason, as in previous work, instead of using the value of the Coriolis parameter $f$ at the departure point at time step $n$, we use a Taylor series expansion of $f$:

$$
\begin{aligned}
f &= f(\phi - \delta\phi) \\
&= f(\phi) - \beta(\phi)\tilde{v}\Delta t + f'
\end{aligned}
\tag{3.1}
$$

where $\beta(\phi) = (1/a)df/d\phi$, $\tilde{v}$ is the $\phi$-component of velocity approximated at the trajectory midpoint, and $f'$ represents the higher-order terms in the expansion. (Numerical experiments show that it is sufficient to keep only the second-order term in $f'$.) $v$ is approximated in the first-order term by the average $(v^{n+1} + v^n_*)/2$, and in $f'$ by $v^n_*$.

**The Grids:** Here, the computational and the physical grids are introduced. As noted previously, the computational domain is given by $[0,2\pi]\times[-\pi/2, \pi/2] \times[\xi_0,\xi_{max}]$, where $\xi_0$ and $\xi_{max}$ depend on the definition of the vertical coordinate chosen. Several examples for the definition of $\xi$ are given in Section 3. Vertical discretization is based on $K$ levels, which are prescribed so that $\xi_0 < \xi_1 < ... < \xi_K = \xi_{max}$. In the following, these are referred to as the "solid" levels (for historical reasons), and can be irregularly spaced. Intermediate levels ("dashed" levels) are defined as $\overline{\xi}_k = (\xi_{k-1} + \xi_k)/2$ for $k = 1,2,...,K$. For placement of one of the equations, "virtual" levels are also defined by $\overline{\overline{\xi}}_k = (\overline{\xi}_k + \overline{\xi}_{k+1})/2$ for $k = 1,2,...,K-1$, and $\overline{\overline{\xi}}_0 = \xi_0$. Note that for uniform spacing, the virtual levels and solid levels coincide.

Now, for each fixed $\xi$-level used, the horizontal discretization is defined. A rectangular discretization of the $(\lambda,\phi)$ plane is chosen as follows. Let $N_\lambda > 0$ and $N_\phi > 0$ be fixed integers, which define the mesh spacing on this grid by $h_\lambda = 2\pi/N_\lambda$ and $h_\phi = \pi/N_\phi$. The computational grid points are then defined to be $(\lambda_i,\phi_j, \xi)$, where $\lambda_i = i\cdot h_\lambda$ for $i=0, 1,..., N_\lambda$ and $\phi_j = -\pi/2+j\cdot h_\phi$ for $j=0, 1, ..., N_\phi$. (At times, fractional indices are used, and the meaning should be clear. For example, $\phi_{j+1/2} = (\phi_j +\phi_{j+1})/2$.) In the present work, $N_\phi = N$ is taken as some power of 2, and $N_\lambda = 2N$, resulting in a uniform horizontal computational mesh with $h_\lambda = h_\phi = h = \pi/N$. In the physical domain, each $\xi$-level is a sphere of radius $a$ (the increment due to height is neglected). Since $\lambda$ and $\phi$ represent longitude and latitude, respectively, $(\lambda_0,\phi_j, \xi)$ and $(\lambda_{2N},\phi_j,\xi)$ correspond to the same physical point for each $j$, and, for all $i$, $(\lambda_i,\phi_0, \xi)$ corresponds to the south pole while $(\lambda_i,\phi_N, \xi)$ corresponds to the north pole. Thus, on the sphere, the corresponding grid has uniform latitudinal (north/south) mesh spacing of size $a\cdot h$, while the longitudinal (east/west) spacing depends on the latitude and is given by $a\cdot h \cos \phi$, so that it equals the north/south spacing at the equator and approaches zero towards the poles. (Generally, it is not necessary to distinguish between the computational and physical grids, but keep in mind that the equations are defined on the sphere.)

**Positioning of the Unknowns and Equations.** The unknowns and equations are all defined along the same vertical lines (that is, there is no horizontal staggering of the grids used for each unknown). However, the vertical positioning differs from unknown to unknown and equation to equation. The

equation positioning will be specified later as the discrete equations are presented. The solid levels ($\xi_k$'s) are used for defining $p$, $\theta$ (and thus, $\overline{\theta}$ and $\theta'$), $\dot{\xi}$, and $q$. The dashed (intermediate) levels are used for defining $M$, $\chi$, $\psi$, and $\omega$. One exception is that $M_0$ is used to denote the value of $M$ at the surface (a solid level) although it is defined by a boundary condition there, rather than an equation. For those functions defined at solid levels, subscripts $i, j, k$ correspond to the value at point ($\lambda_i$, $\phi_j$, $\xi_k$), while for those defined at dashed levels, these subscripts indicate the function value at ($\lambda_i, \phi_j, \overline{\xi}_k$).

**The Discrete Operators.** First, the discrete form of the vertical derivatives used, along with some necessary notation, is introduced. An approximation to the vertical derivative $\partial \cdot / \partial \xi$ is used primarily in equations centered at the dashed levels. Letting $A$ be defined at the solid levels, this approximation is defined by:

$$\left(\delta_\xi A\right)_k = \frac{A_k - A_{k-1}}{\xi_k - \xi_{k-1}}. \tag{3.2}$$

A vertical derivative is also used at the virtual levels in (3.14) below. This time letting $A$ be defined at the solid levels, this approximation is defined by:

$$\left(\hat{\delta}_\xi A\right)_k = \frac{A_{k+1} - A_k}{\overline{\xi}_{k+1} - \overline{\xi}_k}. \tag{3.3}$$

Finally, because $M_0$ is defined at the surface (a solid level) and $M_1$ is defined at the first dashed level $\overline{\xi}_1$, the special placement of equation (3.14a) makes the following definition necessary:

$$\left(\tilde{\delta}_\xi M\right)_0 = \frac{M_1 - M_0}{\overline{\xi}_1 - \xi_0}. \tag{3.4}$$

At dashed levels, vertical averaging of some quantities at solid levels is also needed, and is denoted and defined by

$$\overline{A}_k = \frac{1}{2}\left(A_k + A_{k-1}\right). \tag{3.5}$$

The quantities denoted by $\overline{\overline{(.)}}_k$ are defined at the virtual levels $\overline{\overline{\xi}}_k$ by quadratic interpolation from solid levels as follows:

$$\overline{\overline{(.)}}_k = W_{k,-1}(.)_{k-1} + W_{k,0}(.)_k + W_{k,1}(.)_{k+1} \tag{3.6}$$

where

$$W_{k,-1} = \frac{\left(\bar{\bar{\xi}}_k - \xi_k\right)\left(\bar{\bar{\xi}}_k - \xi_{k+1}\right)}{\left(\xi_{k-1} - \xi_k\right)\left(\xi_{k-1} - \xi_{k+1}\right)}$$

$$W_{k,0} = \frac{\left(\bar{\bar{\xi}}_k - \xi_{k-1}\right)\left(\bar{\bar{\xi}}_k - \xi_{k+1}\right)}{\left(\xi_k - \xi_{k-1}\right)\left(\xi_k - \xi_{k+1}\right)} \qquad (3.7)$$

$$W_{k,1} = \frac{\left(\bar{\bar{\xi}}_k - \xi_{k-1}\right)\left(\bar{\bar{\xi}}_k - \xi_k\right)}{\left(\xi_{k+1} - \xi_{k-1}\right)\left(\xi_{k+1} - \xi_k\right)}$$

Note that with equal vertical spacing, $W_{k,0} = 1$ and $W_{k,-1} = W_{k,1} = 0$.

The horizontal derivatives are defined as in earlier work on shallow water equations (see [8]). These are fairly straightforward second-order finite differences away from the poles, and the integral definitions of the operators (where possible) at the poles. In the following, keep in mind that domain is periodic in the $\lambda$-direction, so that a subscript of $i,j,k$ is the same as $i\pm 2N,j,k$. Now, the first-order differences away from poles (that is, where $0 < j < N$) are defined by:

$$\left(\delta_\lambda A\right)_{i,j,k} = \frac{A_{i+1,j,k} - A_{i-1,j,k}}{2a\cos\phi\, h} \qquad (3.8)$$

and

$$\left(\delta_\phi A\right)_{i,j,k} = \frac{A_{i,j+1,k} - A_{i,j-1,k}}{2a h}. \qquad (3.9)$$

Note that, although $\lambda$ and $\phi$ are used in the notation, these are derivatives on the sphere, and not in the $(\lambda, \phi)$ plane. For the horizontal Laplacian $\nabla^2$, a conservative discretization is used in order to avoid problems over long time integrations. For the sake of simplicity (although at the risk of confusion) the same notation is used for both the continuous and discrete operator. Away from the poles, we define

$$\left(\nabla^2 A\right)_{i,j,k} = \frac{A_{i+1,j,k} - 2 A_{i,j,k} + A_{i-1,j,k}}{a^2 h^2 \cos^2\phi_j}$$
$$+ \frac{\cos\phi_{j+1/2}\left(A_{i,j+1,k} - A_{i,j,k}\right) - \cos\phi_{j-1/2}\left(A_{i,j,k} - A_{i,j-1,k}\right)}{a^2 h^2 \cos\phi_j}. \qquad (3.10)$$

10

At the poles, an approximation for the integral definition of $\nabla^2$ is used. Since the first index is meaningless there, it is omitted in the pole value itself. The operator at the north pole $(j=N)$ is defined as follows, with a similar definition for the south pole:

$$\left(\nabla^2 A\right)_{N,k} = -\frac{4}{a^2 h^2}\left(A_{N,k} - \frac{1}{2N}\sum_{i=1}^{2N} A_{i,N-1,k}\right). \tag{3.11}$$

In the equations, some quantities appear in the equations that involve derivatives with respect to $\lambda$ and $\phi$, which are not defined at the poles, although these can be well defined and exist in the limit. For such quantities, an average over points adjacent to the pole is used. Now, the discrete equations can be presented. For simplicity, the subscripts $i,j$ have been omitted, and it should be understood that these equations hold (with the appropriate discrete operators) for all $i,j$ and the indicated values of $k$. The forcing terms, labelled *forcing*1, ... , *forcing*6, are computed based on quantities at the previous timesteps.

*Pseudo Potential Vorticity Equation:* This equation, the discretization of (2.6), is defined at the dashed levels, and holds for $k$=1,2,...,$K$. Note that equation (2.11) has been used to eliminate $\omega$ from the (2.6), so that pseudo potential vorticity no longer explicitly appears in the equations at all. This simplifies the system somewhat. Here, $q_1$ and $q_2$ arise from semi-implicit treatment of the pseudo-potential vorticity term, and are given in terms of the solution at previous timesteps. It should be noted that $q_2 = 0$ at the poles.

$$
\begin{aligned}
&\left[1 - \tau_1^p(\delta_\xi \dot{\xi})_k\right](f + \nabla^2 \psi_k) - \tau_1^p\overline{\left[\delta_\lambda(\Pi)\delta_\phi\big((1+0.608q)(\overline{\theta}+\theta)\big)\right]}_k \\
&+\tau_1^p\overline{\left[\delta_\phi(\Pi)\delta_\lambda\big((1+0.608q)(\overline{\theta}+\theta)\big)\right]}_k \\
&+\tau_1^p\delta_\lambda\overline{\dot{\xi}}\cdot VERT_k^\nu - \tau_1^p\delta_\phi\overline{\dot{\xi}}\cdot VERT_k^\mu \\
&+\left[(q_1)_k + (q_2)_k\big(\delta_\lambda\varphi_k + \delta_\phi\chi_k\big)\right](\delta_\xi p) = (forcing1)_k.
\end{aligned}
\tag{3.12}
$$

11

*Divergence Equation:* The discretization of (2.7) is also defined at the dashed levels, and holds for $k=1,2,...,K$:

$$\nabla^2 \chi_k + \tau_1^d \nabla^2 M_k - \tau_1^d \overline{\left[\Pi \nabla^2 \theta\right]}_k - \tau_1^d \left[\delta_\lambda \Pi \cdot \delta_\lambda \theta + \delta_\phi \Pi \cdot \delta_\phi \theta\right]_k$$
$$\tau_1^d 0.608 \overline{\left[\delta_\lambda \Pi \cdot \delta_\lambda \left(q(\overline{\theta}+\theta)\right) + \delta_\phi \Pi \cdot \delta_\phi \left(q(\overline{\theta}+\theta)\right)\right]}_k \quad (3.13)$$
$$\tau_1^d 0.608 \overline{\left[q(\overline{\theta}+\theta)\nabla^2 \Pi\right]}_k + \beta \tau_1^d \left(\delta_\lambda \chi_k - \delta_\phi \varphi_k\right) - f \tau_1^d \nabla^2 \varphi_k = (forcing2)_k.$$

*Hydrostatic Equation (surface):* This equation, the discretization of (2.8) near the surface, is centered at $(0.75\xi_0 + 0.25\xi_1)$:

$$\left(\widetilde{\delta}_\xi M\right)_0 - \left(\frac{3}{4}\Pi_0 + \frac{1}{4}\Pi_1\right)\left[\frac{1}{2}\hat{\Gamma}_0 + \frac{1}{2}\Gamma_1 + \alpha_1 (\delta_\xi \theta)_1 + \alpha_2 (\delta_\xi \theta)_2\right]$$
$$+0.608\kappa\left[\frac{3}{4}\left(q(\overline{\theta}+\theta)\frac{\Pi}{p}\right)_0 + \frac{1}{4}\left(q(\overline{\theta}+\theta)\frac{\Pi}{p}\right)_1\right]\left[\alpha_1 (\delta_\xi p)_1 + \alpha_2 (\delta_\xi p)_2\right] = (forcing3)_0. \quad (3.14a)$$

*Hydrostatic Equation:* This equation, the discretization of (2.8) away from the surface, is centered at the virtual levels and is defined for $k=1,2,...,K-1$:

$$\left(\hat{\delta}_\xi M\right)_k - \frac{1}{2}\overline{\overline{\Pi}}_k\left[\Gamma_k + \Gamma_{k+1} + (\delta_\xi \theta)_k + (\delta_\xi \theta)_{k+1}\right]$$
$$+\frac{1}{2}0.608\kappa\overline{\overline{\left(q(\overline{\theta}+\theta)\frac{\Pi}{p}\right)}}_k\left[(\delta_\xi p)_k + (\delta_\xi p)_{k+1}\right] = (forcing3)_k. \quad (3.14b)$$

*Continuity Equation:* This equation, the discretization of (2.9), is centered at the dashed levels and is defined for for $k=1,2,...,K$.

$$\left[1 + \tau_1^d\left(\nabla^2 \chi_k\right) + \tau_1^d\left(\delta_\xi \dot{\xi}\right)_k\right]\left(\delta_\xi p\right) = (forcing4)_k. \quad (3.15)$$

*Thermodynamic Equation:* This equation, the discretization of (2.10), is centered at the solid levels and is defined for for $k=0,1,2,...,K$.

$$\left(\theta\right)_k + \tau_1^\theta \hat{\Gamma}_k \dot{\xi}_k = (forcing5)_k. \quad (3.16)$$

12

*Definition of vertical coordinate:* This equation, the discretization of (2.11), is centered at the solid levels and is defined for for $k=1,2,...,K-1$:

$$\xi(p_k, p_0, \overline{\theta} + \theta_k) = \xi_k. \tag{3.17}$$

*The Moisture Equation:* This equation, the discretization of (2.13), is defined for for $k=0,1,2,...,K$. Since this explicitly defines $q$ at the new timestep, $q$ can be taken as a known function in the remainder of the equations and this equation can be ignored in the development of the solver. In all tests, $q \equiv 0$.

$$q_k = (forcing6)_k. \tag{3.18}$$

In the above equations, a number of quantities have been used, which are defined here.

$$\alpha_1 = \frac{\frac{1}{2}\left(\xi_0 + \overline{\xi_1}\right) - \overline{\xi_2}}{\overline{\xi_1} - \overline{\xi_2}};$$

$$\alpha_2 = \frac{\frac{1}{2}\left(\xi_0 + \overline{\xi_1}\right) - \overline{\xi_1}}{\overline{\xi_1} - \overline{\xi_2}}; \tag{3.19}$$

The $VERT^u$ and $VERT^v$ terms in (3.12) are approximations to $\delta_\xi u$ and $\delta_\xi v$, respectively, and are defined as:

$$VERT_1^u = a_1\left[\hat{\delta}_\xi\left(\delta_\lambda\chi - \delta_\phi\psi\right)\right]_1 + b_1\left[\hat{\delta}_\xi\left(\delta_\lambda\chi - \delta_\phi\psi\right)\right]_2, \tag{3.22}$$

$$VERT_k^u = a_k\left[\hat{\delta}_\xi\left(\delta_\lambda\chi - \delta_\phi\psi\right)\right]_{k-1} + b_k\left[\hat{\delta}_\xi\left(\delta_\lambda\chi - \delta_\phi\psi\right)\right]_k, \tag{3.23}$$

$$VERT_K^u = a_K\left[\hat{\delta}_\xi\left(\delta_\lambda\chi - \delta_\phi\psi\right)\right]_{K-1} + b_K\left[\hat{\delta}_\xi\left(\delta_\lambda\chi - \delta_\phi\psi\right)\right]_{K-2}, \tag{3.24}$$

and

$$VERT_1^v = a_1\left[\hat{\delta}_\xi\left(\delta_\phi\chi + \delta_\lambda\psi\right)\right]_1 + b_1\left[\hat{\delta}_\xi\left(\delta_\phi\chi + \delta_\lambda\psi\right)\right]_2, \tag{3.25}$$

$$VERT_k^v = a_k\left[\hat{\delta}_\xi\left(\delta_\phi\chi + \delta_\lambda\psi\right)\right]_{k-1} + b_k\left[\hat{\delta}_\xi\left(\delta_\phi\chi + \delta_\lambda\psi\right)\right]_k, \tag{3.26}$$

$$VERT_K^v = a_K\left[\hat{\delta}_\xi\left(\delta_\phi\chi + \delta_\lambda\psi\right)\right]_{K-1} + b_K\left[\hat{\delta}_\xi\left(\delta_\phi\chi + \delta_\lambda\psi\right)\right]_{K-2}, \tag{3.27}$$

where the coefficients $a$ and $b$ are interpolation coefficients to the proper levels as follows:

$$a_1 = \frac{\overline{\xi_1} - \overline{\overline{\xi_2}}}{\overline{\xi_1} - \overline{\overline{\xi_2}}}; \qquad b_1 = \frac{\overline{\xi_1} - \overline{\overline{\xi_1}}}{\overline{\xi_2} - \overline{\xi_1}};$$

13

$$a_k = \frac{\bar{\xi}_k - \bar{\bar{\xi}}_k}{\bar{\bar{\xi}}_{k-1} - \bar{\bar{\xi}}_k}; \qquad b_k = \frac{\bar{\xi}_k - \bar{\bar{\xi}}_{k-1}}{\bar{\bar{\xi}}_k - \bar{\bar{\xi}}_{k-1}}, \quad k = 2,3,...,K-1; \qquad (3.28)$$

$$a_K = \frac{\bar{\xi}_K - \bar{\bar{\xi}}_{K-2}}{\bar{\bar{\xi}}_{K-1} - \bar{\bar{\xi}}_{K-2}}; \quad b_K = \frac{\bar{\xi}_K - \bar{\bar{\xi}}_{K-1}}{\bar{\bar{\xi}}_{K-2} - \bar{\bar{\xi}}_{K-1}}.$$

*Boundary Conditions:* The boundary conditions for the discrete equations are:

$$\dot{\xi}_0 = \dot{\xi}_K = 0, \qquad (3.29)$$

$$M_0 = \Phi(\lambda, \phi) + (\bar{\theta}_0 + \theta_0)\Pi_0, \qquad (3.30)$$

and

$$p_K = p_T(\lambda, \phi). \qquad (3.31)$$

In summary, the system under consideration consists of the equations (3.12) - (3.17), with boundary conditions (3.29) - (3.31), and the set of unknowns being $(p, \theta', \dot{\xi}, \psi, \chi, M)$. In tests, two definitions for $\xi$ in (3.17) are used. The first, called the $\sigma$- coordinate, can be used for testing with with no orography, and is given by:

$$\xi = \sigma(p, p_0, p_T) = 1 - \frac{p - p_T}{p_0 - p_T}. \qquad (3.32)$$

The second, a hybrid coordinate, is defined in terms of $\sigma$ above, and is given by:

$$\xi = \theta_{min} + g(\sigma) \cdot [\theta - \theta_{min}] \quad \text{where } g(\sigma) = \frac{1 - e^{-\alpha\sigma}}{1 - e^{-\alpha}}. \qquad (3.33)$$

Here, $\theta_{min}$ is some value less than the maximum value of $\theta$, and in tests is set to 265. $\alpha$ is a value that controls the transition between the surface fitting $\xi$-coordinate and the upper isentropic layers, and in tests is taken variously as 5, 10, and 20. The values for $\tau$ are basically half the time step $\Delta t$, possibly modified by a mild uncentering parameter. In all tests, the $\tau$'s are taken as 1800 s. This completes the definition of the discrete equations.

## 4. The Multigrid Algorithm

In this section, the multigrid (MG) algorithm for the system given in the previous section is developed and described. It is assumed that the reader is fairly familiar with multigrid methods, and is referred to

[3], [4], [6], [7] and [9] for more detailed descriptions of some of the processes. It should be noted that the general philosophical approach taken here is to err initially on the side of robustness, with full efficiency as a secondary consideration.

In Section 4.1, some general terminology, notation, and algorithms are introduced, avoiding as much as possible the details of the multigrid components. Most of these details are given, along with the reasons for the choices, in Section 4.2. The choice of relaxation is quite involved. Some general decisions are given in Section 4.2, but the main task of analyzing the system and deciding the final form of relaxation to be used are contained in Section 4.3. Results contained in Section 5 show that for the given discretization, a reduced grid version of the algorithm may be needed. Details of this version, in which the fewer grid points are used along latitude lines near the poles, are given in Section 4.4. Finally, some comments on improving the algorithm, particularly with respect to efficiency, are given in Section 4.5.

### 4.1 Terminology and Notation

To describe the MG algorithm, it is helpful to first introduce some general terminology and notation. The terms "grid" and "level" will be used interchangeably. Superscripts will generally be used for the level number, with the finest level indicated by 1, and the coarsest by $M$. The level $l$ grid is denoted by $G^l$, and consists of the points $(\lambda_i^l, \phi_j^l, \xi_k^l)$, with $0 \le i \le N_\lambda^l$, $0 \le j \le N_\phi^l$, and $0 \le k \le K^l$. When the level is understood, the point $(\lambda_i^l, \phi_j^l, \xi_k^l)$ is also referred to as point $(i, j, k)$. Because of the diverse vertical positioning of the various equations and unknowns, it is more accurate to say that the level $l$ problem is defined *relative to* this grid rather than *on* this grid, but this should cause no confusion. The coarser grids will be defined later. Now, let $u^l$ denote a solution approximation on level $l$. Here, $u^l$ consists of the approximations (or corrections) to all the individual grid functions $p, \theta, \xi, \chi, \psi$, and $M$, defined at the appropriate points. Similarly, $f^l$ denotes a level $l$ right-hand side, corresponding to the different equations (3.12) - (3.17), and again defined at the appropriate points. Interpolation (prolongation) from level $l+1$ to level $l$ is denoted by $P^l$, while restriction from level $l$ to level $l+1$ is denoted by $R^{l+1}$. $P^l$ is used for interpolating corrections from coarser to finer levels, while $R^{l+1}$ is used for transferring residuals to coarser levels. In the FAS algorithms (see Section 4.2), a different restriction (generally

15

injection) may be used for transferring the values of the unknown functoions to coarser and is denoted by $\hat{R}^{l+1}$. In the FMG process, a higher-order interpolation (here, cubic) is used for interpolating the approximation to a new finer level. This is denoted here by $\tilde{P}^{l}$. Now, let $F^{l}$ denote the discrete approximation to the full nonlinear operator on level $l$, so that the fine grid problem can be written as:

$$F^{1}(u^{1}) = f^{1}. \tag{4.1}$$

In the MG algorithm, it will be necessary at times to look at grid functions and operators defined on only part of the grid in order to perform line or plane relaxation. Subscripts will be used to denote partial grid functions and operators. A line $(i,j)$ in $G^{l}$ is the vertical line of points $(i,j,k)$ with $0 \le k \le K^{l}$. This is also denoted by $G_{i,j}^{l}$. A $j$-plane is the vertical plane of points $(i,j,k)$ with $0 \le i \le N_{\lambda}^{l}$ and $0 \le k \le K^{l}$, and is denoted by $G_{j}^{l}$. Let $u_{i,j}^{l}$ denote the part of $u^{l}$ defined the vertical line $(i,j)$, and $u_{j}^{l}$ be the part of $u^{l}$ defined on the $j$-plane. For completeness, let $u_{N}^{l}$ be the part of the function lying on the vertical line at the north pole, and $u_{S}^{l}$ be the part at the south pole. Separating the operators is somewhat more complicated. For this purpose, the *complement* of a partial grid function is introduced. These are denoted by a "~" in front of the subscript, and indicates the remainder of the grid function. For example, consider the example of the $j$-plane (The following notation also applies to an $(i,j)$ line). The part of $u^{l}$ defined everywhere except on plane $j$ is denoted by $u_{-j}^{l}$, so that $u^{l}$ can be written as $(u_{j}^{l}, u_{-j}^{l})$. Now, let $F_{j}^{l}$ denote the part of $F^{l}$ defined on the plane $j$. This is actually defined on the full function $u^{l}$, but the primary interest is on the action of $F_{j}^{l}$ on $u_{j}^{l}$. Thus, $F_{j}^{l}(u_{j}^{l})$ should be interpreted more accurately as $F_{j}^{l}(u_{j}^{l}; u_{-j}^{l})$ with $u_{-j}^{l}$ frozen. For the plane relaxation, some linear approximation of $F_{j}^{l}$ around the current approximation $u_{-j}^{l}$ is needed. This is denoted by $L_{j}^{l}$, so that for a perturbation $\delta u_{j}^{l}$ of $u_{j}^{l}$, $L_{j}^{l} \delta u_{j}^{l} \approx F_{j}^{l}(u_{j}^{l} + \delta u_{j}^{l}; u_{-j}^{l})$. Here, of course, $L_{j}^{l}$ is generally a function of $u^{l}$. For convenience, setting this linear approximation will be denoted, somewhat inaccurately, by $L_{j}^{l} \approx F_{j}^{l}(u_{j}^{l}; u_{-j}^{l})$. In the current problem, the details of defining $L_{j}^{l}$ will be given later. For plane relaxation, the problem to be solved for the correction $\hat{u}_{j}^{l}$ to $u_{j}^{l}$ is then

$$L_{j}^{l} \hat{u}_{j}^{l} = \hat{f}^{l} = f_{j}^{l} - F_{j}^{l}(u_{j}^{l}, u_{-j}^{l}). \tag{4.2}$$

Similarly, $(i, j)$ line relaxation requires the solution of a problem of the form:

$$L_{i,j}^{l} \hat{u}_{i,j}^{l} = \hat{f}^{l} = f_{i,j}^{l} - F_{i,j}^{l}(u_{i,j}^{l}, u_{-i,j}^{l}) \tag{4.3}$$

for a correction $\hat{u}^l_{i,j}$ to $u^l_{i,j}$, where as above, $L^l_{i,j}$ is a linear approximation to $F^l_{i,j}(u^l_{i,j}, u^l_{-i,j})$ around $u^l$. Generally, a direct solver is used to solve (4.3).

For solution of the linear $j$-plane problem (4.2), a multigrid solver is used. The levels used here will be indicated by a second superscript, with the levels being denoted by $l$, $l+1$, ... , $M$. Note that the finest level number is the same as the global level number. This is somewhat less messy to write than starting again with 1, and is actually more convenient in programming. In the work here, the same coarsest level number $M$ is used for the global and plane MG solvers. Let the local grids be denoted by $G^{l,l}_j, G^{l,l+1}_j, ..., G^{l,M}_j$. Because of the coarsening chosen (see Sections 4.2 and 4.3), interpolation and restriction (written as matrices) between levels $m$ and $m+1$ depend only on $m$, and not on $l$ or $j$. Thus, these can be denoted by $P^m_p$ and $R^{m+1}_p$, respectively. A sequence of linear operators $L^{l,m}_j$ for levels $m = l$, $l+1$, ... , $M$, are defined, with $L^{l,l}_j = L^l_j$ and the coarser grid operator defined as some standard approximation to $L^l_j$. When $j$ and $l$ are understood, they are dropped from the notation.

## 4.2 Major Multigrid Choices

Now, there are a number of decisions to be made in constructing the multigrid algorithm. Although the choices are interrelated, they can often proceed from the most general (overall type of algorithm) to the specific (form of relaxation). In this section, the choices are outlined and explained. As noted, we start with the most general choice of algorithm type.

**Full Approximation Scheme (FAS).** To handle the nonlinearity, the full approximation scheme (FAS) [3] is used, rather than manipulating the problem to produce linear equations using some outer linearization scheme like Newton's method. In FAS the nonlinearity is incorporated into the cycle and taken to all coarser levels. In this way, the full nonlinear problem can be solved in essentially the same time it would take to solve one of the linearized problems. This requires some reformulation of the multigrid algorithm, both in relaxation and the form of the coarse grid correction. Rather than formulate a coarse grid equation for the correction (that is, instead of using the residual equation), the coarser grid problems approximate the full fine grid equation, with an appropriate correction added to

17

the right-hand side. The type of relaxation needed is determined by examination of the linearized problem, and then modified for the nonlinear case. In the notation given in Section 4.1, (1,1) V-cycles using FAS can be written recursively as follows. The relaxation used will be defined more precisely later in this section.

*FAS Multigrid Cycle:  FASMG* $(l; u^l; f^l)$

      If $l = M$:  Relax $n_c$ times on $F^l(u^l) = f^l$ (call $GR(M; u^M; f^M)$: See Section 4.3).

      Otherwise:

            1. Relax on $F^l(u^l) = f^l$ (call $GR(l; u^l; f^l)$).

            2. Set $u^{l+1} = \hat{R}^{l+1} u^l$ and $f^{l+1} = R^{l+1}(f^l - F^l(u^l)) + F^{l+1}(u^{l+1})$.

            3. Call *FASMG* $(l + 1; u^{l+1}; f^{l+1})$.

            4. Set $u^l \leftarrow u^l + P^l(u^{l+1} - \hat{R}^{l+1} u^l)$.

            5. Relax on $F^l(u^l) = f^l$ (call $GR(l; u^l; f^l)$).

**Full Multigrid (FMG).** For efficiency, it was decided to employ full multigrid [3], [4], [9], which is a process in which the first approximation on each level is obtained by interpolating the solution from the next coarser level (here, using cubic interpolation). In FMG, only a fixed reduction in the error (and therefore a small number of cycles) is needed on each succesively finer level to obtain a solution accurate to the level of discretization error, given that the coarser level solution is sufficiently accurate. Although the time discretization is separated from the solution of the nonlinear implicit equations, the time-dependent nature of the problem is important here. The FMG process is designed to solve the given implicit equations at each time step to the level of *spatial* truncation error. Since the discretization error also depends on $\Delta t$, the problem must be formulated to solve for the *change* in the solution over the previous timestep, rather than the full solution. Otherwise, error accumulation much larger than the discretization error can result. Solving for the incremental solution change is easily accomplished by using the solution from the previous timestep as an initial approximation, and by constructing the FMG process so that, on a given level, the initial approximation to this time increment is obtained by interpolation from a coarser level. This is done in the algorithm below.

18

Another consideration arising from the time dependent nature of the problem is the accuracy of the solution of the implicit equations that is required at each time step. It is easy to argue that, for an isolated equation or system, there is no need to solve below the level of discretization error since, beyond that point, the work invested does not result in a better approximation to the solution of the continuous problem. In time dependent problems, however, there may be accuracy or stability concerns because of error accumulation or possible feeding of nonlinear instabilities. Accuracy is addressed by the use of the FMG algorithm (designed to solve for the increment). This should ensure that overall error remains below the level of discretization error. Also, in this model, extrapolation of nonlinear terms, which is the (main) source of nonlinear instability, is avoided, so that no special care is needed to exclude or filter problem frequencies. Now, the FMG algorithm, based on $\nu$ FASMG cycles per level and written for the increment, follows:

*Full Multigrid Algorithm:* $FMG(u^1; f^1)$

  1. For $l = 2, 3, \ldots, M$, set $u^{l+1} = \hat{R}^{l+1} u^l$ and $f^{l+1} = R^{l+1}(f^l - F^l(u^l)) + F^{l+1}(u^{l+1})$.

  2. Set $l = M$.

  3. Call $FASMG(l; u^l; f^l)$ $\nu$ times.

  4. If $l = 1$, stop.

    Otherwise, set $u^l \leftarrow u^l + \tilde{P}^l(u^{l+1} - \hat{R}^{l+1} u^l)$, $l \leftarrow l{-}1$, and go to 3.


**Global Grid Coarsening.** The type of grid coarsening to use is a basic decision, here based primarily on experience and on knowledge of (and partly on ignorance of) the geometry of the mesh. A fairly obvious choice is to use standard coarsening in the $\lambda$ and $\phi$ directions. Such coarsening is typical in spherical models. Coarsening in $\xi$ is more problematic, since little is known *a priori* about the strength of connections in the vertical direction relative to that in the horizontal directions, particularly since it is anticipated that vertical mesh spacing can be nonuniform, with higher resolution near the surface and top boundaries. Since horizontal coarsening alone reduces the number if grid points by a factor of four, relaxation work in a V-cycle is 4/3 times that on the finest grid. Coarsening in the vertical direction also would only reduce that factor to 8/7, while possibly requiring much more complicated (and expensive) relaxation schemes to get the smoothing required. Thus, it was decided not to coarsen at all in the

vertical direction. In the notation introduced earlier, this means that, letting $N^l = 2^{-l+1}N$, the level $l$ grid is defined as follows:

$$G^l = \{(\lambda_i^l, \phi_j^l, \xi_k^l), \; i = 0,1,...,2N^l, \; j = 0,1,...,N^l, \; k = 0,1,...,K\},$$

where $\lambda_i^l = i \cdot \pi/N^l$, $\phi_j^l = -\pi/2 + j \cdot \pi/N^l$, and $\xi_k^l = \xi_k$. Note that $(\lambda_i^{l+1}, \phi_j^{l+1}, \xi_k^{l+1}) = (\lambda_{2i}^l, \phi_{2j}^l, \xi_k^l)$ for $l = 1$, 2, ..., $M$-1. In the following, superscripts on $\xi_k$ values will be dropped, since the same vertical gridpoints are used on all levels. In tests, it was found that coarsening to $N^l = 4$ was sufficient. Some effects of this grid on relaxation will be discussed in the next section.

**Interpolation and Restriction.** Given the semicoarsened mesh, fairly standard interpolation and restriction operators can be defined. Since there is no vertical coarsening and horizontal coarsening is uniform, simple bilinear interpolation for each unknown can be used within horizontal layers. Similarly, horizontal full weighting is used for restriction for the residuals of each equation.

### 4.3 Relaxation.

A final decision on relaxation details must be based on the discrete operators involved, since a basic MG principle is that relaxation must smooth the error in the coarsening directions. As will be verified later, the horizontal Laplacian ($\nabla^2$ in the equations) is a principle part of the operator and is involved in smoothing. Some decisions can be made based on this knowledge, the geometry, and the coarsening used. When point-type coarsening is used, smoothing generally occurs in the direction(s) of strong connections (large coefficients). When the problem is coarsened in a direction of weak connections, it is necessary to simultaneously relax blocks of points that are strongly connected to each other (e.g., line relaxation). Smoothing then depends on the size of the out-of-block connections, which are now relatively large. This is important here because of the uniform coarsening in the latitude-longitude directions. Ignoring the vertical connections for the moment, in the neighborhood of the poles, a uniform latitude-longitude grid gives much smaller mesh spacing in the $\lambda$ (east-west) direction than the $\phi$ (north-south) direction. Thus, for the $\nabla^2$ operator, near the poles the connections in the $\lambda$ direction are much stronger than those in the $\phi$ direction. For this reason, some form of coupled relaxation along

such latitude lines is necessary to obtain full horizontal smoothing of the error. The question of coupling of the equations in relaxation will be addressed later.

As noted previously, little is known about the strength of vertical connections relative to the horizontal connections that may occur in general circumstances. To allow for strong vertical coupling (as well as to forestall possible problems due to vertical central differencing and averaging) a robust approach in which relaxation couples the unknowns on each vertical line is chosen. This coupling also influenced the choice to eliminate the pseudo potential vorticity $\omega$ from (2.5) using the definition of $\omega$ (2.11), which only quantities within a vertical line.

Taken together, these choices indicate a multigrid approach in which $(i,j)$ vertical line relaxation is used away from the poles (and at the poles themselves), while a $j$-plane relaxation is used near the poles. These will be defined later, after analysis of the system.

**Relaxation Analysis.** In a complicated system such as this, it can be difficult to identify the *principle part* of the operator, which is the set of terms that actually determine the smoothing. Typically, this is done by writing the linearized system in operator form with coefficients frozen around the current solution approxiamtion, taking the determinant, and finding the dominant terms (see [4] and [10]). The operators contributing to these terms are used in relaxation (with other terms frozen, either for the entire sweep or just part of the sweep, such as relaxation on one line). For a problem such as this, the dominant terms can change according to the solution approximation itself (since the problem is nonlinear), the location within the domain, and the grid level, making a full analysis difficult. Several approaches were used to help in the identification of these terms, including systematic trial and error.

A technique was developed to facilitate identification of these dominant terms. For this, the Jacobian is computed locally using the given solution approximation. To see how this works, the discrete system of $n$ nonlinear equations can be written in a general form as follows (temporarily deviating from the notation introduced earlier by letting superscripts enclosed in parentheses indicate equation or unknown numbers):

$$\underline{F}^{(l)}(\underline{u}^{(1)},\underline{u}^{(2)},...,\underline{u}^{(n)}) = 0 \quad (l = 1,2,...,n), \tag{4.4}$$

where the $\underline{u}^{(l)}$ (the $l$'th unknown) is a grid function defined on the set of points $\Omega^l$ (identified by their indices $(i,j,k)$) and $\underline{F}^{(l)}$ (the $l$'th equation operator) is a vector valued function defined at the set of points $\Omega^{l'}$. Denote the components of $\underline{u}^{(l)}$ by $u^{(l)}_{i,j,k}$ and of $\underline{F}^{(l)}$ by $F^{(l)}_{i,j,k}$. The sets of points can be different for each unknown and equation, and, in general, a point identified by $(i,j,k)$ in one set need not necessarily coincide with the point with the same index in another set. However, all points and equations with the same $i,j$ lie on the same vertical line, and those with the same $k$ are close to each other (within half a vertical layer). Now, we are interested in the linearized operator around $(i'j'k')$. Since all discrete operators involved are local, a neighborhood $N_{i',j',k'}$ can be found such that each $F^{(l)}_{i',j',k'}$ only involves unknowns defined in this neighborhood. Thus, for each $l,m$ pair, a spatial array of values

$$J^{(l,m)}_{i',j',k'} = \left[ \frac{\partial F^{(l)}_{i',j',k'}}{\partial u^{(m)}_{i,j,k}} \right]_{(i,j,k)\in N_{i',j',k'}} \tag{4.5}$$

can be found (where the derivative is computed numerically, using calls to an existing residual calculation routine). This can be interpreted as the *stencil* of $\underline{u}^{(m)}$ in $F^l_{i',j',k'}$. For convenience, the subscripts are now dropped unless needed for . We are interested in how to smooth using the local operator, which we write symbolically as:

$$J = \begin{bmatrix} J^{(1,1)} & . & . & . & J^{(1,n)} \\ . & . & & & . \\ . & & . & & . \\ . & & & . & . \\ J^{(n,1)} & . & . & . & J^{(n,n)} \end{bmatrix}. \tag{4.6}$$

As stated previously, the terms responsible for smoothing are those that dominate the determinant of $J$. Rather than actually multiply the stencils involved, a faster (and rougher) idea of the dominant terms in the determinant can be obtained by simply looking at the sizes of the operators involved. For this, we define

$$\left| J^{(l,m)} \right| = \max_{(i,j,k)} \left| \frac{\partial F^{(l)}_{i',j',k'}}{\partial u^{(m)}_{i,j,k}} \right|, \tag{4.7}$$

the maximum stencil entry, and examine the resulting matrix

$$|J| = \begin{bmatrix} \left|J^{(1,1)}\right| & . & . & . & \left|J^{(1,n)}\right| \\ . & . & & & . \\ . & & . & & . \\ . & & & . & . \\ \left|J^{(n,1)}\right| & . & . & . & \left|J^{(n,n)}\right| \end{bmatrix}.$$

(4.8)

The determinant of this matrix is taken, term by term, and those products with the largest values are noted. Here, only a few dominate, while most are zero. From this list, the important $J^{(l,m)}$'s are identified. However, each such stencil can have more than one contributing term in the equation $\underline{F}^{(l)} = 0$. Generally, these can be easily identified by matching the form of the stencil with the known form of the discrete operators involved (that is, it is easy to tell whether the stencil corresponds to a horontal Laplacian, or a vertical first derivative, for example). This gives a good idea of the main terms in the discrete operator responsible for smoothing in the neighborhood of $(i', j', k')$. In addition, it can give information on whether or not some nonlinearities are important in relaxation, and whether coupling of equations is necessary. A note of caution is required here. At times, cancellation of what appear to be dominant operators can occur. This is not revealed in this analysis, so further examination of the equations is warranted if there are several large, nearly equal terms in the determinant. When such cancellation occurs, it may be important to couple relaxation of these equations/unknowns, and to also take into account the next largest determinant terms in relaxation.

When the dominant terms involve vertical derivatives, this process needs further refinement. It is necessary to take the terms involved into account in relaxation, but this also means that other terms may be responsible for horizontal smoothing. Thus, the stencils are summed in the vertical direction and the process is repeated. This has the effect of eliminating those terms containing vertical derivatives (although not vertical averaging). The remaining dominant terms terms must then be identified and also included for relaxation. This is somewhat more difficult, since the determinant of the corresponding system is generally zero, so large terms are identified by inspection.

Such a process was quite helpful in the current work, although in practice, there was quite a bit of testing and verification to ensure that smoothing is adequate. Another useful technique was to compare

the residual of the relaxation system (corresponding to a line or plane) with the full residual there when the global solution is corrected. If the full operator residual stagnates while the relaxation problem is solved well, this indicates that some important terms are being neglected.

For convenience here, equations (3.12)-(3.17) will be referred to as equations 1-6, and the unknowns $(p, \theta', \xi, \psi, \chi, M)$ as unknowns 1-6. Now, we can begin to analyze the smoothing behavior of a model system with no orography and a representative solution approximation. The mesh is defined by $N_\lambda = 64$, $N_\phi = 32$, and $K = 18$. A sigma vertical coordinate system (3.32) is used with uniform vertical mesh spacing (i.e., $\xi_k = k/K$). The matrices (4.8) are computed at several places in the domain. For illustration purposes, the point (32,17,2) is first examined. This lies on the equator near the surface. Now $|J|$ is given in the following matrix:

$$
\begin{vmatrix}
2.34E-10 & 4.34E-02 & 2.79E-09 & 3.29E-14 & 1.02E-11 & 0.00E+00 \\
2.99E-12 & 0.00E+00 & 9.10E-06 & 1.02E-11 & 3.29E-14 & 1.84E-08 \\
1.38E-01 & 0.00E+00 & 8.75E+03 & 0.00E+00 & 0.00E+00 & 1.80E+01 \\
1.80E+01 & 3.21E+09 & 0.00E+00 & 1.82E-03 & 0.00E+00 & 0.00E+00 \\
0.00E+00 & 8.32E+04 & 1.00E+00 & 0.00E+00 & 0.00E+00 & 0.00E+00 \\
1.01E-05 & 0.00E+00 & 0.00E+00 & 0.00E+00 & 0.00E+00 & 0.00E+00
\end{vmatrix}
$$

The determinant has six nonzero terms, given in descending order in the following table.

| Product Terms | | | | | | Size |
|---|---|---|---|---|---|---|
| 1,5 | 2,4 | 3,6 | 4,2 | 5,3 | 6,1 | 6.076E-17 |
| 1,5 | 2,3 | 3,6 | 4,4 | 5,2 | 6,1 | 2.554E-18 |
| 1,5 | 2,6 | 3,3 | 4,4 | 5,2 | 6,1 | 2.511E-18 |
| 1,4 | 2,5 | 3,6 | 4,2 | 5,3 | 6,1 | 6.327E-22 |
| 1,2 | 2,5 | 3,6 | 4,4 | 5,3 | 6,1 | 4.729E-22 |
| 1,3 | 2,5 | 3,6 | 4,4 | 5,2 | 6,1 | 2.532E-24 |

Each row of the table corresponds to one nonzero term of the determinant, where each term of the determinant is the product of one entry from row 1, one from row 2, etc. The pairs of numbers in the table are the row (equation) and column (unknown) numbers of the factors in the determinant term, while the "size" column is the term itself (with sign omitted). Thus, for example, the first row corresponds to the term

$$ \left|J^{1,5}\right| \cdot \left|J^{2,4}\right| \cdot \left|J^{3,6}\right| \cdot \left|J^{4,2}\right| \cdot \left|J^{5,3}\right| \cdot \left|J^{6,1}\right|. $$

This is the largest term. Examination of the actual stencils in the Jacobian shows that these terms correspond to the operators shown below (with coefficients omitted):

$$J^{(1,5)} \sim \nabla^2$$
$$J^{(2,4)} \sim \nabla^2$$
$$J^{(3,6)} \sim \delta_\xi$$
$$J^{(4,2)} \sim \delta_\xi$$
$$J^{(5,3)} \sim I$$
$$J^{(6,1)} \sim I$$

This identification process is repeated at a number of locations throughout the domain. At this stage of the work, rather than to include the different terms only where they tend to dominate, the choice has been to attempt to ensure robust smoothing and include all terms that dominate anywhere in the domain. Once this preliminary examination was performed and candidate terms for inclusion were identified, tests were done to ensure that line and plane relaxation did, in fact, work as expected. This was done by comparing the full nonlinear residuals for the lines and planes before and after relaxation to see that they were sufficiently reduced.

In testing plane relaxation, an FASMG solver was implemented, allowing for some nonlinearities. However, this was found to be unnecessary, and only a linear approximation to the equations was necessary. This linear systems used in line and plane relaxation can now be specified. The same terms are used in both types of relaxation, in all parts of the domain, and on all levels. The linearized equations are as follows, with the perturbations from the current solution designated by a single underline, and frozen coefficients designated by a double underline. All fully-frozen frozen terms are taken to the right-hand side, which is now denoted by $\tilde{f}$.

$$\left[1 - \tau_1^p(\delta_\xi \underline{\underline{\xi}})_k\right]\nabla^2 \underline{\psi}_k - \tau_1^p(f + \nabla^2 \underline{\underline{\psi}}_k)(\delta_\xi \underline{\dot{\xi}})_k + \left[(q_1)_k + (q_2)_k\left(\delta_\lambda \underline{\underline{\varphi}}_k + \delta_\phi \underline{\underline{\chi}}_k\right)\right](\delta_\xi \underline{p})_k = \tilde{f}_1 \quad (4.9)$$

$$\nabla^2 \underline{\chi}_k + \tau_1^d \nabla^2 \underline{M}_k - \tau_1^d\overline{\left[\underline{\underline{\Pi}}\nabla^2 \underline{\theta}\right]}_k - f\tau_1^d\nabla^2 \underline{\varphi}_k = \tilde{f}_2 \quad (4.10)$$

25

$$\left(\tilde{\delta}_\xi \underline{M}\right)_0 - \left(\frac{3}{4}\underline{\underline{\Pi}}_0 + \frac{1}{4}\underline{\underline{\Pi}}_1\right)\left[\alpha_1(\delta_\xi\underline{\theta})_1 + \alpha_2(\delta_\xi\underline{\theta})_2\right]$$

$$-\kappa\left[\frac{1}{2}\hat{\Gamma}_0 + \frac{1}{2}\Gamma_1 + \alpha_1(\delta_\xi\underline{\theta})_1 + \alpha_2(\delta_\xi\underline{\theta})_2\right]\left(\frac{3}{4}\frac{\Pi_0}{p_0}\underline{p}_0 + \frac{1}{4}\frac{\Pi_1}{p_1}\underline{p}_1\right) \tag{4.11a}$$

$$+0.608\kappa\left[\frac{3}{4}\left(\underline{q}(\overline{\theta}+\underline{\theta})\frac{\Pi}{p}\right)_0 + \frac{1}{4}\left(\underline{q}(\overline{\theta}+\underline{\theta})\frac{\Pi}{p}\right)_1\right]\left[\alpha_1(\delta_\xi\underline{p})_1 + \alpha_2(\delta_\xi\underline{p})_2\right] = \tilde{f}_0^3$$

$$\left(\hat{\delta}_\xi\underline{M}\right)_k - \frac{1}{2}\underline{\underline{\Pi}}_k\left[(\delta_\xi\underline{\theta})_k + (\delta_\xi\underline{\theta})_{k+1}\right]$$

$$-\frac{\kappa}{2}\left[\Gamma_k + \Gamma_{k+1} + (\delta_\xi\underline{\theta})_k + (\delta_\xi\underline{\theta})_{k+1}\right]\left[\overline{\overline{[\Pi/p]\underline{p}}}\right]_k \tag{4.11b}$$

$$+\frac{1}{2}0.608\kappa\left(\overline{\overline{\underline{q}(\overline{\theta}+\underline{\theta})\Pi/p}}\right)_k\left[(\delta_\xi\underline{p})_k + (\delta_\xi\underline{p})_{k+1}\right] = \tilde{f}_k^3$$

$$\tau_1^r\left(\delta_\xi\underline{p}\right)_k\left(\nabla^2\underline{\chi}_k\right) + \tau_1^r\left(\delta_\xi\underline{p}\right)_k\left(\delta_\xi\underline{\dot{\xi}}\right)_k + \left[1 + \tau_1^r\left(\nabla^2\underline{\chi}_k\right) + \tau_1^r\left(\delta_\xi\underline{\dot{\xi}}\right)_k\right]\left(\delta_\xi\underline{p}\right)_k = \tilde{f}_k^4 \tag{4.12}$$

$$\left(\underline{\theta}\right)_k + \tau_1^\theta\hat{\Gamma}_k\underline{\dot{\xi}}_k = \tilde{f}_k^5. \tag{4.13}$$

$$\frac{\partial\xi}{\partial p_k}\underline{p}_k + \frac{\partial\xi}{\partial p_0}\underline{p}_0 + \frac{\partial\xi}{\partial\theta_k}\underline{\theta}_0 = \tilde{f}_k^6. \tag{4.14}$$

For line or plane relaxation, the $\nabla^2$ operator is restricted to that line or plane. Now, given a line or plane, the linear operator $L$ corresponding to the linearized $F_{i,j}^l$ or $F_j^l$ can be constructed. Thus, relaxation at one line $(i,j)$ on level $l$ can now be defined:

*Nonlinear Line Solver:* $NLS(l,i,j,u^l,f^l)$

    1. Define $L \approx F_{i,j}^l(u_{i,j}^l;u_{-i,j}^l)$.

    2. Solve $L\hat{u} = \hat{f} = f_{i,j}^l - F_{i,j}^l(u_{i,j}^l;u_{-i,j}^l)$.

    3. Set $u_{i,j}^l \leftarrow u_{i,j}^l + \hat{u}$.

A direct linear solver is used for step 2. The efficiency of this step will be discussed later.

**Plane Relaxation.** For test purposes, a direct solver for the plane problems based on cyclic reduction was initially implemented. Not surprisingly, it proved to be very expensive, and therefore impractical. However, exact solution of the plane problems is not necessary, since the goal of relaxation is simply to reduce oscillatory off-plane error components. This can be done much more efficiently using a plane MG solver. Some problems were encountered here, and will be described in the next section. Consider the plane $j$ on global level $l$. Using the notation introduced in Section 4.1, the grid is denoted by $G_j^l$, and consists of the points $(\lambda_i^l, \phi_j^l, \xi_k)$, with $0 \le i \le 2N^l$ and $0 \le k \le K$. As in the global MG solver, this grid is not coarsened in the vertical direction. Standard coarsening is used in the $\lambda$-direction, giving a 2D semicoarsened grid. As before, denote this fine grid by level $l$ and denote the grids used in this plane solver by $G_j^{l,m}$, with $m = l, \dots, M$. Thus, $G_j^{l,m}$ consists of the points $(\lambda_i^m, \phi_j^l, \xi_k)$, with $0 \le i \le 2N^m$ and $0 \le k \le K$. Note that, due to the coarsening used and the grid numbering chosen, level $m$ uses the same values for $\lambda_i^m$ and $N^m$ for the global grids and all plane grids, independent of $l$. Because of the semicoarsening, vertical line relaxation is used in the smoothing on each level, which is done in red/black order. Linear interpolation is used in the $\lambda$-direction only. Similarly, $\lambda$-direction full weighting is used for residual transfer to coarser levels. The coarsest grid problem (generally four vertical lines) is solved by several relaxation sweeps. The operators $L^m$ for $m = l, \dots, M$ are defined here using the differences given in Section 3 in the linearized equations in (4.9)-(4.14). In the next section, an alternative is presented. Now, the plane MG (1,1) V-cycle has a form independent of $j$ and $l$, and is defined as follows:

*The Plane Multigrid V-cycle:* $PMG(m, u^m, f^m, L^m, \dots, L^M)$

    If $m = M$: Relax $n_c$ times on $L^m u^m = f^m$ (vertical line relaxation, red/black ordering).
    Otherwise:

        1. Relax on $L^m u^m = f^m$ (vertical line relaxation, red/black ordering).

        2. Set $u^{m+1} = 0$ and $f^{m+1} = R^{m+1}(f^m - L^m u^m)$.

        3. Call $PMG(m+1, u^{m+1}, f^{m+1}, L^{m+1}, \dots, L^M)$.

        4. Set $u^m \leftarrow u^m + P_p^m u^{m+1}$.

        5. Relax on $L^m u^m = f^m$ (vertical line relaxation, red/black ordering).

Plane relaxation is defined in terms of the plane MG cycle above, and is given here:

*Plane Relaxation:* $PR(l,j,u^l,f^l)$.

    1. Define $L_j^{l,l} \approx F_j^l(u_j^l;u_{-j}^l)$, and coarser grid operators $L_j^{l,l+1},...,L_j^{l,M}$.

    2. Set $\hat{u}^l = 0$ and $\hat{f}^l = f_j^l - F_j^l(u_j^l;u_{-j}^l)$.

    3. Call $PMG(l,\hat{u}^l,\hat{f}^l,L_j^{l,l},...,L_j^{l,M})$.

    4. Set $u_j^l \leftarrow u_j^l + \hat{u}^l$.

It should be noted that the linear systems used in line relaxation of the global sweeps and in the line relaxation used within the plane solver are basically identical. As noted, these line problems are solved directly, and the work there constitutes the major part of the overall solution algorithm. Efficiency considerations will be discussed further in Section 4.5.

At the poles, relaxation uses vertical line solves very similar to the $(i,j)$ line solves away from the poles. The only real difference is in the discretizations used in the equations (4.9)-(4.14). For convenience, pole relaxation will be denoted using the same notation as the nonlinear line solver, with $(i,j) = (1,0)$ for the south pole and $(1,N^l)$ for the north pole. Now, the full global relaxation sweep can be defined in terms of $\phi_{plane}$, a parameter that specifies where line and plane relaxation are to be performed:

*Global Relaxation Sweep:* $GR(l;u^l;f^l)$

    1. Call $NLS(l,1,0,u^l,f^l)$ and $NLS(l,1,N^l,u^l,f^l)$    (pole relaxation).

    2. For $j= 2, 4, ..., N^l-2, 1, 3, ..., N^l-1$,

        If $|\phi_j| \geq \phi_{plane}$, call $PR(l,j,u^l,f^l)$.    (plane relaxation)

    3. For i+j odd, then i+j even $(0 \leq i \leq 2N^l, 1 \leq j \leq N^l -1)$,

        If $|\phi_j| < \phi_{plane}$, call $NLS(l,i,j,u^l,f^l)$.    (line relaxation)

This completes the specification of the basic MG solver.

## 4.4 The Reduced Grid.

Because of problems with divergence using the current discretization discussed in Section 5, it is sometimes necessary to work with *reduced grids*. The basic idea is that the small mesh size in the $\lambda$-direction near the poles is really an artifact of the uniform longitude-latitude grid, and does not contribute to overall accuracy. Thus, it is possible to seek a solution whose smoothness along latitude lines (from one grid point to the next) depends on the latitude line $j$. To make this more precise, consider the fine grid $G^1$, with mesh parameter $N^1$. As before, the grid consists of the points $(\lambda_i^1, \phi_j^1, \xi_k)$, with $0 \le i \le 2N^1$, $0 \le j \le N^1$, and $0 \le k \le K^1$. For this type of reduced grid, for each $j$, a number $\rho_j$ is chosen, generally a power of 2, so that the $\lambda$ values actually used along line $j$ are $\lambda_i$, $i = 0$, $\rho_j$, $2\rho_j$, ... , $2N^1$. Let $N_j^1 = N^1/\rho_j$. Function values at intermediate points can be defined (if needed) by interpolation in the $\lambda$-direction. In the work here, linear interpolation was used. This affects the discretizations for the gradient and horizontal Laplacian operators given in (3.8) and (3.10) as follows:

$$\left(\delta_\lambda A\right)_{i,j,k} = \frac{A_{i+\rho_j,j,k} - A_{i-\rho_j,j,k}}{2a\rho_j \cos\phi\, h} \tag{4.15}$$

and

$$\left(\nabla^2 A\right)_{i,j,k} = \frac{A_{i+\rho_j,j,k} - 2 A_{i,j,k} + A_{i-\rho_j,j,k}}{a^2 \rho_j^2 h^2 \cos^2\phi_j}$$
$$+ \frac{\cos\phi_{j+1/2}\left(A_{i,j+1,k} - A_{i,j,k}\right) - \cos\phi_{j-1/2}\left(A_{i,j,k} - A_{i,j-1,k}\right)}{a^2 h^2 \cos\phi_j}. \tag{4.16}$$

Note that this decreases the size of the connections in the $\lambda$-direction, and by choosing appropriate $\rho_j$'s, it may be possible to eliminate the need for plane relaxation. However, the approach taken here has been only to take the smallest $\rho_j$ needed for convergence and retain the plane relaxation. The actual choice of the $\rho_j$'s will be discussed further in Section 5. Here, it will be assumed that $\rho_j = 1$ except at a few $j$-planes near the north and south poles.

So far, only the finest global reduced grid has been defined. The method of coarsening such grids must be specified. For this, a superscript is added to indicate level number. That is, define $\rho_j^1 = \rho_j$. Now, the coarser grid $\rho_j$'s can be defined recursively. Remember that plane $j$ (with $j$ even) on level $l$ corresponds

29

to the plane $j/2$ on level $l+1$, the next coarser level. Let $\bar{G}^l$ indicate the reduced level $l$ grid. Rather than "fully" coarsening each grid (so that the number of grid points along all even latitude lines, including those already reduced, is halved on coarser levels), we take $\bar{G}^{l+1} = \bar{G}^l \cap G^{l+1}$. That is, if a plane $j$ on level $l$ contains fewer than $2N^l$ points in the $\lambda$-direction, the number of points is *not* reduced further on going to the next coarser level. This can also be interpreted as $\rho_j^{l+1} = \max(1, \rho_{2j}^l / 2)$ (or, alternatively, that $N_j^{l+1} = \min(N^{l+1}, N_{2j}^l / 2)$).

Ideally, the overall solution process (including the outer, time step iteration) should be defined using this reduced grid. However, as an intermediate step in testing the MG solution method, the approach taken has been to find some "reduced-grid" solution approximation for the full problem. For this, the original grids are used, but it is desired to keep an approximation that lies in the corresponding reduced space. That is, the solution approximation is defined on all of $G^l$, but has interpolated values on those points not in $\bar{G}^l$. The goal, perhaps somewhat artificial, has been to modify the MG algorithm so that the solution naturally remains in the reduced space while maintaining good MG convergence, as measured by the fine grid residual. That is, if the solution to the actual problem lies in this space, the MG method should converge to that solution with usual MG efficiency. To this end, the first step was to start with the full problem, and during plane relaxation near the poles, simply skip relaxation on levels finer than the new, reduced resolution there. That is, if $\rho_j > 1$, then within the *PMG* algorithm, do not perform relaxation sweeps on levels $m = 1, \ldots, \log_2 \rho_j$. If the initial approximation on the plane is contained in this reduced solution space, then the corrected solution after the plane MG cycle will also be in this space.

Using the above algorithm, one further requirement on the grid reduction is given by the global coarsening for practical reasons. Horizontal bilinear interpolation from level $l+1$ to level $l$ should give a solution approximation that stays in the reduced space. That is, along each $\lambda$-line in a particular $j$-plane, it should be a linear interpolant from $N_j^l$ points along that line. If $j$ is even, this is automatic, since the correction obtained from level $l+1$ is the interpolant from $N_{j/2}^{l+1} \le N_j^l$ points. If $j$ is odd, however, since plane $j$ interpolates from planes $j/2$ and $j/2 +1$ (where $j/2$ actually indicates the greatest integer in $j/2$),

the correction appears as a linear interpolant from $\max(N^{l+1}_{j/2}, N^{l+1}_{j/2+1})$ points. This requires that

$N^l_j \geq \max(N^{l+1}_{j/2}, N^{l+1}_{j/2+1})$. Taking the first value, this means that

$$N^l / \rho^l_j = N^l_j \geq N^{l+1}_{j/2} = \min\left(N^{l+1}, N^l_{j-1}/2\right)$$
$$= \min\left(N^l/2, N^l/2\rho^l_{j-1}\right)$$

Thus, $\rho^l_j \leq 2\rho^l_{j-1}$. Similarly, $\rho^l_j \leq 2\rho^l_{j+1}$. This basically precludes large jumps in grid reduction, at least from an even to an odd plane. (The poles represent a "naturally" reduced plane with $\rho^l_N = \rho^l_S = N^l$, so there is no restriction in jumps from the poles to adjacent planes.) In the test cases here, a factor of 2 reduction from one plane to the next was natural, so no problems were encountered. It should be noted that a factor of 2 from one plane to the next on level $l$ corresponds to a factor of 4 on level $l+1$, so for fine enough horizontal meshes, problems could be encountered and further restrictions would be necessary. However, in the meshes used and the reduction required here, the conditions above were met on all levels.

It was noted that, in this approach, one goal was to maintain good MG convergence as measured by reduction of the fine grid residual. This is quite restrictive, and as noted, somewhat artificial. A more realistic method would be to simply measure the residual on the reduced grid problem, where the right-hand side is fixed. In the approach taken here, simply skipping relaxation on finer levels in plane relaxation results in recomputation of the right-hand sides of the "target" (coarser grid) plane problem. In testing convergence of the linear plane problems, it was found that, even when the finest grid right-hand side was set to zero (so that the solution was zero) and the initial error was constructed to lie in the reduced space, convergence per cycle degraded with the number of levels skipped in relaxation. Using a variational definition of the coarser grid linear operators as follows restored good convergence factors, as seen in Section 5.2.

$$L^{m+1} = R^{m+1}_p L^m P^m_p \quad \text{for } m = 1, ..., M. \tag{4.17}$$

This has been implemented in the linear plane solvers only, and not in the global coarse grid discretizations. It is an open question whether the variational definition of the operators will actually be necessary when more realistic convergence measures for the reduced problem are introduced.

31

## 4.5 Efficiency Considerations.

As the focus on work to date has been to obtain a working algorithm, concentrating on robustness rather than efficiency, timing comparisons for the method have not been included. Currently, the algorithm is probably much slower than comparable explicit or semi-implicit $u$-$v$ models. In the algorithm presented here, the main computational work is in relaxation. Computation of the residuals is the second most expensive process. Some improvement can be gained here by more careful programming. However, the largest gains will be made in relaxation. This can be done in two ways. In both the line and plane relaxation, the main work is involved in the vertical line solves. The line problem is solved as a fully coupled system, so that the size of the matrix involved is approximately $6K$ rows and columns. A direct solver is used that takes little advantage of the basic banded structure (although there is coupling at all layers to the surface pressure through the definition of $\xi$). A more efficient solver can be written for this. The second method for making relaxation more efficient is to determine the real coupling of equations necessary. Often in MG applications, relaxation of the different unknowns can be done in separate sweeps. As a trivial example, consider the case of two grid functions (unknowns) defined on the mesh, in which each satisfies a Poisson equation. Since there are two independent problems defined, relaxation for each can be separate, but the type of relaxation grouping (line, plane, etc.) is determined by the geometry. More often, the connection between unknowns is of lower-triangular form, so that the unknowns can be relaxed in sequence, smoothing the error in each separately without adversely affecting residuals of previously relaxed equations. As noted, the current algorithm was written for robustness rather than efficiency, and more system analysis is required to determine the minimal amount of coupling necessary for relaxation (that is, which equations or groups of equations can be relaxed separately for the different quantities). Complete separation (which is unrealistic) would result in the direct solution of 6 $K{\times}K$ matrix problems per vertical line solve, rather than one $6K{\times}6K$ matrix problem, resulting in considerable savings. More realistically, it can be expected that some coupling will be necessary. However, any decoupling, combined with improved matrix solution routines tailored for the task, will improve efficiency greatly.

## 5. Numerical Results.

In this section, numerical results are presented showing the behavior of the algorithm on several sample problems at single timesteps. Such problems are still being examined, and no results are presented showing the behavior of the algorithm in the full time-dependent model. In Section 5.1, the several model problems are presented, and the results with the MG algorithm are shown. There are problems for which the basic method of Section 4 does not converge. These are presented in Section 5.2. The difficulties with one of these problems is examined in detail, and a modified algorithm, using the reduced grids presented in Section 4.4, is shown to converge there.

### 5.1 Initial Tests.

The first set of problems considered are as follows:

*Problem 1.* This example is a model system with no orography and a representative solution approximation. The mesh is defined by $N_\lambda = 64$, $N_\phi = 32$, and $K = 18$. A sigma vertical coordinate system (3.32) is used with uniform vertical mesh spacing (i.e., $\xi_k = k/K$).

*Problem 2.* This example is a model system with no orography and a representative solution approximation. The mesh is defined by $N_\lambda = 64$, $N_\phi = 32$, and $K = 18$. A hybrid vertical coordinate system (3.33) is used with $\alpha = 20$ and $\theta_{min} = 265$. Different resolution is used in the lower and upper layers, with $\xi_0 = 265$, $\xi_k - \xi_{k-1} = 6$, for $k = 1, 2, \ldots 15$, and $\xi_k - \xi_{k-1} = 40$ otherwise, so that $\xi_{max} = 475$.

*Problem 3.* This example is the same as the previous problem with the exception of the definition of the vertical coordinate. The hybrid coordinate (3.33) is still used, but with $\alpha = 10$ and $\theta_{min} = 265$. This time, $\xi_0 = 265$, $\xi_k - \xi_{k-1} = 8\,^2/_3$ for $k = 1, 2, \ldots 15$, and $\xi_k - \xi_{k-1} = 26\,^2/_3$ otherwise, so that again $\xi_{max} = 475$.

First, the behavior of line and plane relaxation is examined (for Problems 1 and 2 only). The following tables show the behavior of the plane solvers used. Tables 1-3 contain results for Problem 1. For

33

| j (plane) | Eq. 1 | Eq. 2 | Eq. 3 | Eq. 4 | Eq. 5 | Eq. 6 |
|---|---|---|---|---|---|---|
| 32 | 0.380 | 0.380 | 0.382 | 0.380 | 0.384 | 0.380 |
| 31 | 0.092 | 0.089 | 0.087 | 0.092 | 0.092 | 0.085 |
| 30 | 0.089 | 0.084 | 0.078 | 0.089 | 0.085 | 0.080 |
| 25 | 0.052 | 0.052 | 0.037 | 0.052 | 0.052 | 0.038 |
| 17 | 0.030 | 0.031 | 0.018 | 0.031 | 0.031 | 0.017 |

**Table 1.** Average convergence factors for MG plane solver (linear problem) for Problem 1.

| j (plane) | Eq. 1 | Eq. 2 | Eq. 3 | Eq. 4 | Eq. 5 | Eq. 6 |
|---|---|---|---|---|---|---|
| 32 | 0.957 | 0.979 | 0.979 | 0.957 | 1.001 | 0.966 |
| 31 | 0.966 | 0.928 | 0.928 | 0.966 | 0.947 | 0.890 |
| 30 | 0.869 | 0.850 | 0.850 | 0.869 | 0.861 | 0.774 |
| 25 | 0.445 | 0.445 | 0.445 | 0.445 | 0.447 | 0.299 |
| 17 | 0.241 | 0.250 | 0.250 | 0.250 | 0.250 | 0.146 |

**Table 2.** Average convergence factors for relaxation only (linear problem) for Problem 1.

| j (plane) | Eq. 1 | Eq. 2 | Eq. 3 | Eq. 4 | Eq. 5 | Eq. 6 |
|---|---|---|---|---|---|---|
| 32 | 0.377 | 0.263 | 0.891 | 0.377 | 1.000 | 1.006 |
| 31 | 0.091 | 0.090* | 0.994 | 0.084 | 1.012 | 1.001 |
| 30 | 0.080 | 0.083* | 1.003 | 0.083 | 1.008 | 0.994 |
| 25 | 0.053* | 0.052* | 1.005 | 0.050* | 0.996 | 0.989 |
| 17 | 0.031* | 0.031* | 1.000 | 0.031* | 0.992 | 0.975 |

**Table 3.** Average convergence factors for MG plane solver (nonlinear problem) for Problem 1.

| j (plane) | Eq. 1 | Eq. 2 | Eq. 3 | Eq. 4 | Eq. 5 | Eq. 6 |
|---|---|---|---|---|---|---|
| 32 | 0.083 | 0.061 | 0.079 | 0.081 | 0.082 | 0.083 |
| 31 | 0.082 | 0.089 | 0.083 | 0.072 | 0.087 | 0.090 |
| 30 | 0.074 | 0.084 | 0.082 | 0.086 | 0.079 | 0.081 |
| 25 | 0.055 | 0.052 | 0.035 | 0.055 | 0.051 | 0.052 |
| 17 | 0.031 | 0.031 | 0.017 | 0.032 | 0.032 | 0.031 |

**Table 4.** Average convergence factors for MG plane solver (linear problem) for Problem 2.

| j (plane) | Eq. 1 | Eq. 2 | Eq. 3 | Eq. 4 | Eq. 5 | Eq. 6 |
|---|---|---|---|---|---|---|
| 32 | 0.956 | 0.979 | 0.965 | 0.925 | 0.972 | 0.967 |
| 31 | 0.978 | 0.928 | 0.877 | 0.970 | 0.948 | 0.955 |
| 30 | 0.870 | 0.850 | 0.755 | 0.865 | 0.865 | 0.862 |
| 25 | 0.445 | 0.445 | 0.285 | 0.445 | 0.445 | 0.440 |
| 17 | 0.250 | 0.250 | 0.141 | 0.250 | 0.251 | 0.251 |

**Table 5.** Average convergence factors for relaxation only (linear problem) for Problem 2.

| j (plane) | Eq. 1 | Eq. 2 | Eq. 3 | Eq. 4 | Eq. 5 | Eq. 6 |
|---|---|---|---|---|---|---|
| 32 | 0.085 | 0.191 | 1.053 | 0.084 | 0.979 | 0.812 |
| 31 | 0.087 | 0.130 | 1.009 | 0.083 | 1.049 | 0.000 |
| 30 | 0.078 | 0.084 | 1.025 | 0.081 | 0.979 | 0.000 |
| 25 | 0.052 | 0.055* | 0.739 | 0.036* | 1.075 | 0.909 |
| 17 | 0.032* | 0.033* | 0.984 | 0.017* | 1.055 | 0.922 |

**Table 6.** Average convergence factors for MG plane solver (nonlinear problem) for Problem 2.


V-cycle results for the full problems are given in Tables 7, 8, and 9, for problems 1, 2, and 3, respectively. These contain initial residuals for each equation and residuals after each cycle, and an effective convergence factor (the maximum convergence factor for equations 1, 2, and 4). For reasons discussed below, the norm of the right-hand side of each equation is also given. In both cases, convergence factors appear satisfactory for a few cycles, then the residuals stagnate. This type of behavior can often be attributed to one (or a combination) of three causes: The level of roundoff error can be reached; the solver may not converge for certain error components; or the right-hand side may not be in the range of the operator. The level of roundoff error has clearly not been reached, since that should ideally give residuals of a size $10^{-16} \cdot \|f\|$. Here, the third appears the most likely. It is clear that the operator itself is singular, since there are no horizontal boundary conditions and all $\chi$ and $\psi$ only occur in conjunction with horizontal derivatives. Thus, arbitrary constants can be added to $\chi$ and $\psi$ at each vertical level without affecting the residual. This means that the range of the operator cannot be the set of all possible right-hand sides, and some consistency condition must be met if the problem is to have a solution. For these test problems, all grid functions used (the right-hand sides, the initial solution approximation, and various coefficients) were given to six decimal places only, so it seems reasonable to assume that the residual could be reduced to approximately $10^{-6} \cdot \|f\|$ without consistency problems. Comparison of the largest residual (equation 4) with $\|f_4\|$ shows that this appears to be the case. (For smaller residuals, particularly equation 2, it is likely that lack of convergence of the other equations could interfere there, so that reduction of those residuals do not reach the expected levels.)

Unfortunately, for a complicated nonlinear system such as this, it is difficult to determine the proper consistency condition and to somehow enforce it. As a test, 20 cycles were performed, and a new right-hand side computed consistent with the solution obtained (that is, the right-hand side was simply set to the operator applied to the solution). Then, the original initial approximation was restored and

36

| Cycle | Eq. 1 | Eq. 2 | Eq. 3 | Eq. 4 | Eq. 5 | Eq. 6 | Factor |
|---|---|---|---|---|---|---|---|
| $\|f\|$ | 0.00E+00 | 1.52E-09 | 0.00E+00 | 4.76E+02 | 4.76E-02 | 0.00E+00 | - |
| Initial | 4.03E-10 | 8.75E-10 | 4.16E-02 | 3.60E-01 | 3.04E-04 | 6.88E-09 | - |
| 1 | 9.19E-11 | 1.16E-10 | 1.71E-05 | 3.17E-02 | 3.18E-18 | 1.96E-16 | 0.088 |
| 2 | 1.14E-11 | 1.08E-11 | 1.33E-08 | 1.90E-03 | 3.22E-18 | 2.43E-19 | 0.060 |
| 3 | 1.71E-12 | 1.51E-12 | 1.22E-11 | 2.73E-04 | 3.19E-18 | 2.41E-19 | 0.144 |
| 4 | 2.76E-13 | 7.71E-13 | 2.22E-12 | 1.27E-04 | 3.20E-18 | 2.39E-19 | 0.465 |
| 5 | 6.99E-14 | 7.57E-13 | 2.23E-12 | 1.23E-04 | 3.20E-18 | 2.41E-19 | 0.969 |
| 6 | 4.93E-14 | 7.57E-13 | 2.22E-12 | 1.23E-04 | 3.19E-18 | 2.44E-19 | 1.000 |
| 7 | 4.83E-14 | 7.57E-13 | 2.23E-12 | 1.23E-04 | 3.21E-18 | 2.40E-19 | 1.000 |
| 8 | 4.83E-14 | 7.57E-13 | 2.22E-12 | 1.23E-04 | 3.17E-18 | 2.44E-19 | 1.000 |
| 9 | 4.83E-14 | 7.57E-13 | 2.22E-12 | 1.23E-04 | 3.21E-18 | 2.42E-19 | 1.000 |
| 10 | 4.83E-14 | 7.57E-13 | 2.21E-12 | 1.23E-04 | 3.23E-18 | 2.41E-19 | 1.000 |

**Table 7.** Convergence history for Problem 1 (original forcing terms).

| Cycle | Eq. 1 | Eq. 2 | Eq. 3 | Eq. 4 | Eq. 5 | Eq. 6 | Factor |
|---|---|---|---|---|---|---|---|
| $\|f\|$ | 0.00E+00 | 9.12E-10 | 0.00E+00 | 5.74E+00 | 3.30E-02 | 0.00E+00 | - |
| Initial | 1.21E-11 | 1.96E-09 | 3.27E-04 | 5.65E-05 | 6.73E-09 | 2.09E-03 | - |
| 1 | 1.71E-09 | 2.39E-10 | 1.89E-06 | 1.96E-03 | 2.15E-18 | 2.82E-08 | >1 |
| 2 | 5.94E-10 | 7.58E-11 | 9.09E-09 | 1.95E-04 | 1.99E-18 | 1.40E-12 | 0.347 |
| 3 | 1.91E-10 | 2.64E-11 | 3.24E-11 | 4.71E-05 | 1.89E-18 | 3.01E-14 | 0.348 |
| 4 | 6.07E-11 | 1.11E-11 | 3.87E-13 | 2.46E-05 | 1.94E-18 | 2.51E-15 | 0.522 |
| 5 | 1.92E-11 | 8.32E-12 | 1.17E-13 | 1.95E-05 | 1.93E-18 | 8.76E-16 | 0.793 |
| 6 | 6.03E-12 | 8.10E-12 | 6.47E-14 | 1.86E-05 | 1.89E-18 | 5.86E-16 | 0.954 |
| 7 | 1.90E-12 | 8.11E-12 | 5.26E-14 | 1.83E-05 | 1.94E-18 | 5.23E-16 | 0.984 |
| 8 | 6.25E-13 | 8.12E-12 | 4.92E-14 | 1.83E-05 | 1.91E-18 | 5.00E-16 | 1.000 |
| 9 | 2.70E-13 | 8.12E-12 | 4.78E-14 | 1.83E-05 | 1.95E-18 | 4.99E-16 | 1.000 |
| 10 | 1.98E-13 | 8.12E-12 | 4.79E-14 | 1.82E-05 | 1.95E-18 | 4.97E-16 | 0.995 |

**Table 8.** Convergence history for Problem 2 (original forcing terms).

| Cycle | Eq. 1 | Eq. 2 | Eq. 3 | Eq. 4 | Eq. 5 | Eq. 6 | Factor |
|---|---|---|---|---|---|---|---|
| $\|f\|$ | 0.00E+00 | 6.48E-10 | 0.00E+00 | 4.62E+00 | 2.85E-02 | 0.00E+00 | - |
| Initial | 2.30E-10 | 1.14E-09 | 2.91E-04 | 1.90E-03 | 1.38E-05 | 3.57E-04 | - |
| 1 | 4.70E-11 | 1.01E-10 | 1.11E-07 | 3.43E-04 | 1.54E-18 | 1.12E-10 | 0.204 |
| 2 | 9.55E-12 | 8.39E-12 | 2.78E-10 | 1.96E-05 | 1.51E-18 | 6.02E-14 | 0.203 |
| 3 | 2.73E-12 | 1.37E-12 | 9.57E-13 | 2.85E-06 | 1.54E-18 | 3.86E-16 | 0.286 |
| 4 | 7.62E-13 | 1.06E-12 | 1.37E-14 | 1.61E-06 | 1.52E-18 | 1.58E-16 | 0.774 |
| 5 | 2.06E-13 | 1.04E-12 | 1.32E-14 | 1.68E-06 | 1.52E-18 | 1.58E-16 | 1.043 |
| 6 | 5.53E-14 | 1.04E-12 | 1.33E-14 | 1.69E-06 | 1.53E-18 | 1.60E-16 | 1.006 |
| 7 | 2.34E-14 | 1.04E-12 | 1.32E-14 | 1.69E-06 | 1.54E-18 | 1.56E-16 | 1.000 |
| 8 | 2.20E-14 | 1.04E-12 | 1.34E-14 | 1.69E-06 | 1.54E-18 | 1.59E-16 | 1.000 |
| 9 | 2.27E-14 | 1.04E-12 | 1.33E-14 | 1.69E-06 | 1.53E-18 | 1.60E-16 | 1.032 |
| 10 | 2.29E-14 | 1.04E-12 | 1.32E-14 | 1.69E-06 | 1.51E-18 | 1.56E-16 | 1.009 |

**Table 9.** Convergence history for Problem 3 (original forcing terms).

the problem re-solved. Results for problems 1, 2, and 3 are shown in Tables 10, 11, and 12, respectively. Convergence now appears quite good. This cannot be viewed as conclusive, however, since this test may be construed as cheating in a way. If there are error components that are untouched by the MG solution process, it could be simply that the new problem is manufactured so that these error components are not present. However, this seems unlikely. This type of behavior is consistent with that for simpler linear systems in which the operator is singular, such as the Poisson problem with full Neumann boundary conditions. There, the same type of singularity exists, and in an inconsistent problem, MG will converge well to a solution to the consistent problem, although as measured by the residual, convergence stagnates. The fact that the modified problem continues to converge well long after the original solution process appears to stagnate indicates that something similar is occurring here, and stagnation is due to inconsistency of the right-hand side.

Problem 1 and 3 convergence histories generally look reasonable, while problem 2 residuals show a jump at the first cycle. This may indicate that the solution process actually converges to a different solution than the desired one. The main difference between problems 2 and 3 is the definition of $\alpha$ used in the hybrid coordinate. This needs closer examination, and awaits further study.

| Cycle | Eq. 1 | Eq. 2 | Eq. 3 | Eq. 4 | Eq. 5 | Eq. 6 | Factor |
|-------|-------|-------|-------|-------|-------|-------|--------|
| $\|f\|$ | 0.00E+00 | 1.52E-09 | 0.00E+00 | 4.76E+02 | 4.76E-02 | 0.00E+00 | - |
| Initial | 4.03E-10 | 8.75E-10 | 4.16E-02 | 3.60E-01 | 3.04E-04 | 6.88E-09 | - |
| 1 | 9.19E-11 | 1.16E-10 | 1.71E-05 | 3.16E-02 | 3.21E-18 | 3.21E-18 | 0.124 |
| 2 | 1.14E-11 | 1.08E-11 | 1.33E-08 | 1.90E-03 | 3.19E-18 | 3.19E-18 | 0.150 |
| 3 | 1.71E-12 | 1.33E-12 | 1.22E-11 | 2.43E-04 | 3.22E-18 | 3.22E-18 | 0.159 |
| 4 | 2.72E-13 | 1.66E-13 | 2.24E-12 | 3.00E-05 | 3.21E-18 | 3.21E-18 | 0.192 |
| 5 | 5.21E-14 | 2.22E-14 | 2.24E-12 | 4.01E-06 | 3.21E-18 | 3.21E-18 | 0.228 |
| 6 | 1.19E-14 | 3.19E-15 | 2.23E-12 | 5.59E-07 | 3.22E-18 | 3.22E-18 | 0.250 |
| 7 | 2.98E-15 | 5.53E-16 | 2.22E-12 | 8.82E-08 | 3.23E-18 | 3.23E-18 | 0.258 |
| 8 | 7.70E-16 | 1.19E-16 | 2.23E-12 | 1.70E-08 | 3.23E-18 | 3.23E-18 | 0.261 |
| 9 | 2.01E-16 | 2.92E-17 | 2.22E-12 | 3.89E-09 | 3.21E-18 | 3.21E-18 | 0.262 |
| 10 | 5.26E-17 | 7.50E-18 | 2.21E-12 | 9.75E-10 | 3.20E-18 | 3.20E-18 | 0.262 |
| 11 | 1.38E-17 | 1.95E-18 | 2.24E-12 | 2.52E-10 | 3.18E-18 | 3.18E-18 | 0.262 |
| 12 | 3.61E-18 | 5.12E-19 | 2.11E-12 | 6.58E-11 | 3.20E-18 | 3.20E-18 | 0.262 |
| 13 | 9.46E-19 | 1.37E-19 | 2.00E-12 | 1.72E-11 | 3.19E-18 | 3.19E-18 | 0.265 |
| 14 | 2.48E-19 | 4.42E-20 | 1.82E-12 | 4.55E-12 | 3.23E-18 | 3.23E-18 | 0.629 |
| 15 | 6.49E-20 | 2.78E-20 | 1.57E-12 | 1.28E-12 | 3.19E-18 | 3.19E-18 | 0.942 |

**Table 10.** Convergence history for Problem 1 (modified forcing terms).

| Cycle | Eq. 1 | Eq. 2 | Eq. 3 | Eq. 4 | Eq. 5 | Eq. 6 | Factor |
|---|---|---|---|---|---|---|---|
| $\|f\|$ | 0.00E+00 | 9.12E-10 | 0.00E+00 | 5.74E+00 | 3.30E-02 | 0.00E+00 | - |
| Initial | 1.21E-11 | 1.96E-09 | 3.27E-04 | 5.93E-05 | 6.73E-09 | 2.09E-03 | - |
| 1 | 1.71E-09 | 2.39E-10 | 1.89E-06 | 1.97E-03 | 2.16E-18 | 2.82E-08 | >1 |
| 2 | 5.94E-10 | 7.60E-11 | 9.08E-09 | 1.87E-04 | 1.94E-18 | 1.40E-12 | 0.095 |
| 3 | 1.91E-10 | 2.58E-11 | 3.24E-11 | 4.06E-05 | 1.96E-18 | 2.54E-14 | 0.217 |
| 4 | 6.07E-11 | 8.31E-12 | 1.98E-13 | 1.30E-05 | 1.93E-18 | 1.23E-15 | 0.320 |
| 5 | 1.92E-11 | 2.63E-12 | 2.79E-14 | 4.10E-06 | 1.91E-18 | 2.02E-16 | 0.315 |
| 6 | 6.01E-12 | 8.24E-13 | 1.95E-14 | 1.29E-06 | 1.96E-18 | 1.50E-16 | 0.315 |
| 7 | 1.87E-12 | 2.55E-13 | 1.92E-14 | 4.03E-07 | 1.89E-18 | 1.48E-16 | 0.312 |
| 8 | 5.76E-13 | 7.85E-14 | 1.90E-14 | 1.25E-07 | 1.95E-18 | 1.43E-16 | 0.310 |
| 9 | 1.76E-13 | 2.39E-14 | 1.91E-14 | 3.84E-08 | 1.95E-18 | 1.39E-16 | 0.307 |
| 10 | 5.34E-14 | 7.22E-15 | 1.90E-14 | 1.17E-08 | 1.95E-18 | 1.30E-16 | 0.305 |
| 11 | 1.60E-14 | 2.16E-15 | 1.88E-14 | 3.53E-09 | 1.97E-18 | 1.24E-16 | 0.302 |
| 12 | 4.78E-15 | 6.42E-16 | 1.90E-14 | 1.05E-09 | 1.94E-18 | 1.17E-16 | 0.297 |
| 13 | 1.41E-15 | 1.89E-16 | 1.89E-14 | 3.13E-10 | 2.00E-18 | 1.14E-16 | 0.298 |
| 14 | 4.14E-16 | 5.52E-17 | 1.90E-14 | 9.19E-11 | 1.97E-18 | 1.09E-16 | 0.294 |
| 15 | 1.20E-16 | 1.60E-17 | 1.93E-14 | 2.68E-11 | 1.95E-18 | 1.04E-16 | 0.292 |

**Table 11.** Convergence history for Problem 2 (modified forcing terms).

| Cycle | Eq. 1 | Eq. 2 | Eq. 3 | Eq. 4 | Eq. 5 | Eq. 6 | Factor |
|---|---|---|---|---|---|---|---|
| $\|f\|$ | 2.30E-14 | 6.48E-10 | 1.33E-14 | 4.62E+00 | 2.85E-02 | 0.00E+00 | - |
| Initial | 2.30E-10 | 1.14E-09 | 2.91E-04 | 1.90E-03 | 1.38E-05 | 3.57E-04 | - |
| 1 | 4.70E-11 | 1.01E-10 | 1.11E-07 | 3.41E-04 | 1.56E-18 | 1.12E-10 | 0.204 |
| 2 | 9.56E-12 | 8.35E-12 | 2.78E-10 | 2.07E-05 | 1.54E-18 | 6.02E-14 | 0.203 |
| 3 | 2.74E-12 | 8.86E-13 | 9.57E-13 | 1.88E-06 | 1.53E-18 | 4.12E-16 | 0.287 |
| 4 | 7.70E-13 | 1.29E-13 | 1.37E-14 | 2.17E-07 | 1.51E-18 | 1.59E-16 | 0.281 |
| 5 | 2.13E-13 | 2.91E-14 | 1.33E-14 | 4.96E-08 | 1.52E-18 | 1.58E-16 | 0.277 |
| 6 | 5.83E-14 | 7.87E-15 | 1.33E-14 | 1.27E-08 | 1.53E-18 | 1.60E-16 | 0.274 |
| 7 | 1.59E-14 | 2.17E-15 | 1.32E-14 | 3.45E-09 | 1.54E-18 | 1.56E-16 | 0.276 |
| 8 | 4.35E-15 | 5.95E-16 | 1.32E-14 | 9.38E-10 | 1.53E-18 | 1.49E-16 | 0.274 |
| 9 | 1.19E-15 | 1.63E-16 | 1.32E-14 | 2.56E-10 | 1.50E-18 | 1.40E-16 | 0.274 |
| 10 | 3.24E-16 | 4.45E-17 | 1.33E-14 | 6.97E-11 | 1.52E-18 | 1.23E-16 | 0.273 |
| 11 | 8.83E-17 | 1.21E-17 | 1.32E-14 | 1.90E-11 | 1.53E-18 | 1.09E-16 | 0.273 |
| 12 | 2.41E-17 | 3.31E-18 | 1.32E-14 | 5.17E-12 | 1.53E-18 | 9.72E-17 | 0.274 |
| 13 | 6.56E-18 | 9.01E-19 | 1.28E-14 | 1.41E-12 | 1.52E-18 | 8.66E-17 | 0.273 |
| 14 | 1.79E-18 | 2.47E-19 | 1.25E-14 | 3.84E-13 | 1.55E-18 | 7.67E-17 | 0.274 |
| 15 | 4.86E-19 | 7.20E-20 | 1.23E-14 | 1.04E-13 | 1.56E-18 | 7.03E-17 | 0.291 |

**Table 12.** Convergence history for Problem 3 (modified forcing terms).

## 5.2 Further Tests and the Reduced grid Problem

During the development of the MG solver, difficulties with convergence led to generating test problems in which the atmosphere was in a state of rest (that is, $u = v = \dot{\xi} = 0$). Here, the solution should remain unchanged, and the purpose of these problems was to make sure that the MG solver maintained this state of rest. Problems arose with the more highly refined mesh, . The problem, and the approach taken to deal with it, are discussed in this section.

Problem 4. This example is a model system with no orography. The mesh is defined by $N_\lambda = 64$, $N_\phi = 32$, and $K = 18$. A sigma vertical coordinate system (3.32) is used with uniform vertical mesh spacing (i.e., $\xi_k = k/K$).

Problem 5. This example is a model system with no orography. The mesh is defined by $N_\lambda = 64$, $N_\phi = 32$, and $K = 18$. A hybrid vertical coordinate system (3.33) is used with $\alpha = 5$ and $\theta_{min} = 260$. Uniform vertical mesh spacing is used with $\xi_0 = 260$ and $\xi_{max} = 550$.

Problem 6. This example is a model system with no orography. Here, a more highly refined horizontal mesh is used, with with $N_\lambda = 128$, $N_\phi = 64$, and $K = 18$. A sigma vertical coordinate system (3.32) is used with uniform vertical mesh spacing (i.e., $\xi_k = k/K$).

For problems 4 and 5, the solver did not diverge, maintaining the state of rest. However, The solver failed on Problem 6, when a negative value of pressure was obtained (which makes the computation of $\Pi$ impossible). It was found that the plane MG solver was diverging near the poles. This is demonstrated Table 13, which shows the behavior of plane relaxation on the linear problem for $j = 63$ (adjacent to the north pole), 62, 61, 60, 59, 48 (45° N), and 32 (the equator).

| j (plane) | Eq. 1 | Eq. 2 | Eq. 3 | Eq. 4 | Eq. 5 | Eq. 6 |
|---|---|---|---|---|---|---|
| 63 | 120.3 | 39.43 | 122.2 | 120.3 | 123.2 | 121.1 |
| 62 | 21.50 | 0.945 | 20.74 | 21.51 | 21.52 | 4.989 |
| 61 | 1.667 | 0.947 | 1.642 | 1.667 | 1.555 | 0.924 |
| 60 | 0.839 | 0.923 | 0.881 | 0.838 | 0.977 | 0.904 |
| 59 | 0.891 | 0.889 | 0.853 | 0.891 | 0.935 | 0.859 |
| 48 | 0.478 | 0.444 | 0.404 | 0.478 | 0.456 | 0.397 |
| 32 | 0.250 | 0.250 | 0.221 | 0.269 | 0.258 | 0.218 |

**Table 13.** Average convergence factors for line relaxation only, Problem 6, full linear problem.

Trial and error was used to determine the equations and terms responsible for this (that is, to identify the simplest subsystem that still exhibiting this divergent behavior in the linear solver). The resulting system retained Equations (3.13), (3.14b), (3.15), and (3.16), and the unknowns ($\theta'$, $\dot{\xi}$, $\chi$, $M$). Note that $M_0$, and (through the boundary condition (3.30)) $p_0$, have been excluded. The resulting equations, showing the terms retained, are as follows,

Equation (3.13) for $k=1, 2, ..., K$:

$$\left(\nabla^2 \chi\right)_k + \tau\left(\nabla^2 M\right)_k - \tau\overline{\left[\Pi\nabla^2\theta\right]}_k = f_k^2 \tag{5.1}$$

Equation (3.14b) for $k=1, 2, ..., K-1$:

$$\left(\hat{\delta}_\xi M\right)_k - \frac{1}{2}\Pi_k\left[(\delta_\xi\theta)_k + (\delta_\xi\theta)_{k+1}\right] = f_k^3 \tag{5.2}$$

Equation (3.15) for $k=1, 2, ..., K$:

$$\left(\nabla^2 \chi\right)_k + \left(\delta_\xi\dot{\xi}\right)_k = f_k^4 \tag{5.3}$$

Equation (3.15) for $k=0, 1, 2, ..., K$:

$$\left(\theta\right)_k + \tau\left(\hat{\Gamma}\dot{\xi}\right)_k = f_k^5. \tag{5.4}$$

Here, $\Pi$ is frozen using the current values of $p$, all values of $\tau$ are equal, and vertical mesh spacing is uniform (so that solid and virtual levels coincide). Now, repeating the plane relaxation tests on this reduced system gives the results shown in Table 14. Note that both diverge for $j = 61, 62,$ and $63$, while they converge elsewhere.

41

| j (plane) | Eq. 1 | Eq. 2 | Eq. 3 | Eq. 4 | Eq. 5 | Eq. 6 |
|-----------|-------|-------|-------|-------|-------|-------|
| 63 | - | 21.09 | 21.03 | 21.07 | 21.06 | - |
| 62 | - | 44.73 | 44.71 | 44.77 | 44.77 | - |
| 61 | - | 1.440 | 2.335 | 2.339 | 2.335 | - |
| 60 | - | 0.928 | 0.929 | 0.929 | 0.926 | - |
| 59 | - | 0.892 | 0.895 | 0.891 | 0.893 | - |
| 48 | - | 0.445 | 0.441 | 0.443 | 0.441 | - |
| 32 | - | 0.250 | 0.250 | 0.249 | 0.250 | - |

**Table 14.** Average convergence factors for line relaxation only, Problem 6, reduced linear problem.

To see why divergence occurs, this system is first simplified as follows:

1. Eliminate $\chi$ in Equation (5.1) using (5.3).

2. Eliminate $\theta$ in Equations (5.1) and (5.2) using Equation (5.4).

3. Apply $\delta_\xi$ to Equation (5.1), $\tau \nabla^2$ to Equation (5.2), and subtract to eliminate $M$.

Explicitly writing the vertical differences and averages, the resulting equation in $\xi$ has the form:

$$-\frac{\xi_{k+1} - 2\xi_k + \xi_{k-1}}{h_\xi^2} + \frac{\tau^2}{2}\left[\frac{\Pi_{k+1} - \Pi_k}{h_\xi}\hat{\Gamma}_{k+1}\nabla^2\xi_{k+1} + \frac{\Pi_k - \Pi_{k-1}}{h_\xi}\hat{\Gamma}_{k-1}\nabla^2\xi_{k-1}\right] = \hat{f} \qquad (5.5)$$

for $k = 1, 2, ..., K-1$. The boundary conditions are as before, with $\xi_0 = \xi_K = 0$. Now, restricted to a $\phi$-plane $j$, the operator $\nabla^2$ has the form

$$\left(\nabla^2 A\right)_{i,j,k} = c_j^w A_{i-1,j,k} + c_j^c A_{i,j,k} + c_j^e A_{i-1,j,k} , \qquad (5.6)$$

where

$$c_j^w = c_j^e = \frac{1}{a^2 h^2 \cos^2 \phi_j} \quad \text{and} \quad c_j^c = -(c_j^w + c_j^e) - \frac{\cos\phi_{j+1/2} + \cos\phi_{j-1/2}}{a^2 h^2 \cos\phi_j}. \qquad (5.7)$$

Thus, the operator can be scaled and written in stencil form as:

$$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 0 \end{bmatrix} + c_k^+ \begin{bmatrix} -1 & 2+\varepsilon & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + c_k^- \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & 2+\varepsilon & -1 \end{bmatrix} \qquad (5.8)$$

where

$$c_k^+ = \frac{\tau^2 h_\xi^2 \hat{\Gamma}_{k+1}}{2a^2 h^2 \cos^2 \phi_j} \frac{\Pi_{k+1} - \Pi_k}{h_\xi},$$

$$\qquad (5.9)$$

$$c_k^- = \frac{\tau^2 h_\xi^2 \hat{\Gamma}_{k-1}}{2a^2 h^2 \cos^2 \phi_j} \frac{\Pi_k - \Pi_{k-1}}{h_\xi},$$

and

$$\varepsilon = \cos\phi_j(\cos\phi_{j+1/2} + \cos\phi_{j-1/2}).\qquad(5.10)$$

As before, $h = \pi/N$. Also, note that at grid lines adjacent to the poles ($j = 1$ and $N$-1), $\cos\phi_j \approx h$, so that $\varepsilon \approx 2h^2$. Tests show that line relaxation on a model system (5.5) with $c_k^+ = c_k^- = c$ for all $k$ and $\varepsilon = 0$ diverges for $c > 0.5$. In particular, note that, for this simplified model, if $c = 0.5$, the resulting stencil is:

$$\begin{bmatrix} -\frac{1}{2} & 0 & -\frac{1}{2} \\ 0 & 2 & 0 \\ -\frac{1}{2} & 0 & -\frac{1}{2} \end{bmatrix}$$

This is the so-called skewed Laplace operator, which has an MG convergence factor of 1.0 when semicoarsening (or standard coarsening, for that matter) is used, since there is no smoothing in the left-right direction. In practice, there is some convergence due to relaxation because of the Dirichlet boundary conditions at $k = 0$ and $k = K$. For acceptable convergence, the value should actually be somewhat less than 0.5. In Problem 6, the $c_k$'s increase with $\xi$, and decrease according to the square of the distance from the pole. The maximum value of $c_k$ is 7.05 at $j = 63$, 1.76 at $j = 62$, 0.78 at $j = 61$, 0.31 at $j = 60$, and 0.29 at $j = 59$. Thus, divergence for this simplified model corresponds to that for the plane relaxation on the reduced equations and the full system. It should be noted that results are symmetrical about the equator, so the same behavior is observed at $j = 1$ as at $j = 63$, for example.

It appears that the stencil is some discretization of $\partial^2/\partial\xi^2 + c\nabla^2$, which is nicely elliptic, but that the vertical positioning of the equations and unknowns in the discretization leads to a vertical averaging for the $c\nabla^2$ term that introduces non-ellipticity near the poles. Several methods were tested for dealing with this problem, all based on the fact that the grid is basically over-resolved in the $\lambda$-direction near the poles. Various $\lambda$-direction averaging methods were tested, both averaging the solution to reduce high frequency error and introducing using averaging in constructing the horizontal differences. These were unsuccessful, however. Another method was then used in which longer-range larger stencils were used for the difference operators in the $\lambda$-direction, which is described in Section 4.4.

Note that the denominators in the first terms in (5.9) correspond to twice the square of the horizontal mesh size on the spherical mesh, so that when the reduced grid method is used, the $c_k$'s for plane $j$ are are reduced by a factor of $\rho_j^2$. Using the values for $c_k$ obtained for Problem 6, minimum values of $\rho_j$

needed for convergence are using $\rho_1 = \rho_{63} = 4$, $\rho_2 = \rho_{62} = 2$, and $\rho_j = 1$ otherwise. Convergence of plane MG relaxation for $j = 63, ..., 59$ for values of $\rho_j = 1, 2, 4$, and 8 are given in Table 15 for the reduced system (5.1)-(5.4), and for the full system (3.12)-(3.17) in Table 16. Here, standard coarse grid discretizations for the operators are used. Note that divergence does occur where predicted. However, as noted in Section 4.4, there appears to be a limit to convergence factors that degrades as $\rho_j$ is increased. The convergence factors are measured on the finest level, not the actual reduced grid level, so this could be an artifact of this. Nevertheless, it was found that the use of variational operators (4.17) improved convergence, avoiding this limitation. Corresponding results for the variational operators are shown in Table 17 for the reduced system (5.1)-(5.4), and in Table 18 for the full system (3.12)-(3.17).

| j (plane) | $\rho_j = 1$ | $\rho_j = 2$ | $\rho_j = 4$ | $\rho_j = 8$ |
|---|---|---|---|---|
| 63 | 1474. | 1502. | 0.600 | 0.630 |
| 62 | 1091. | 0.483 | 0.600 | 0.630 |
| 61 | 4.445 | 0.483 | 0.600 | 0.630 |
| 60 | 0.104 | 0.483 | 0.600 | 0.630 |
| 59 | 0.095 | 0.483 | 0.600 | 0.630 |

**Table 15.** Overall average convergence factors for plane MG cycles for Problem 6, reduced system (5.1)-(5.4), standard CG operators, using the reduced grids defined by $\rho_j$.

| j (plane) | $\rho_j = 1$ | $\rho_j = 2$ | $\rho_j = 4$ | $\rho_j = 8$ |
|---|---|---|---|---|
| 63 | 8096. | 301.0 | 0.599 | 0.626 |
| 62 | 222.7 | 0.481 | 0.601 | 0.626 |
| 61 | 2.508 | 0.480 | 0.600 | 0.626 |
| 60 | 0.101 | 0.481 | 0.600 | 0.626 |
| 59 | 0.096 | 0.481 | 0.600 | 0.626 |

**Table 16.** Overall average convergence factors for plane MG cycles for Problem 6, full system (3.12)-(3.17), standard CG operators, using the reduced grids defined by $\rho_j$.

| j (plane) | $\rho_j = 1$ | $\rho_j = 2$ | $\rho_j = 4$ | $\rho_j = 8$ |
|---|---|---|---|---|
| 63 | >9999 | 6253. | 8.796 | 0.087 |
| 62 | 1071. | 3.309 | 0.077 | 0.087 |
| 61 | 4.452 | 0.124 | 0.069 | 0.087 |
| 60 | 0.106 | 0.077 | 0.069 | 0.087 |
| 59 | 0.087 | 0.068 | 0.069 | 0.087 |

**Table 17.** Overall average convergence factors for plane MG cycles for Problem 6, reduced system (5.1)-(5.4), variational CG operators, using the reduced grids defined by $\rho_j$.

| j (plane) | $\rho_j = 1$ | $\rho_j = 2$ | $\rho_j = 4$ | $\rho_j = 8$ |
|---|---|---|---|---|
| 63 | >9999 | >9999 | 6.168 | 0.087 |
| 62 | 271.3 | 2.437 | 0.077 | 0.087 |
| 61 | 2.490 | 0.104 | 0.069 | 0.087 |
| 60 | 0.094 | 0.077 | 0.069 | 0.089 |
| 59 | 0.090 | 0.068 | 0.069 | 0.087 |

**Table 18.** Overall average convergence factors for plane MG cycles for Problem 6, full system (3.12)-(3.17), variational CG operators, using the reduced grids defined by $\rho_j$.

One thing to note is that, with the variational discretization, the plane problems diverge for $\rho_{63} = 4$ and $\rho_{62} = 2$, while convergence was obtained with standard discretizations. That is due to the fact that, for the vertical derivatives, the variational definition uses a horizontal averaging over three vertical grid lines, which affects smoothing and places somewhat stricter requirements the $c_k$'s. Thus, $\rho_{63} = 8$ and $\rho_{62} = 4$, are required for convergence of the plane MG solver.

Thus, the reduced grid method was implemented using for the full Problem 6 using values $\rho_1 = \rho_{63} = 8$, $\rho_2 = \rho_{62} = 4$, $\rho_3 = \rho_{61} = 2$, and $\rho_j = 1$ otherwise. Convergence factors for the first five cycles are shown in the Table 19 below. The divergence problem has been solved, and it appears that the solution as given is accurate to close to the level of roundoff error.

| Cycle | Eq. 1 | Eq. 2 | Eq. 3 | Eq. 4 | Eq. 5 | Eq. 6 |
|---|---|---|---|---|---|---|
| $\|f\|$ | 0.00E+00 | 0.00E+00 | 0.00E+00 | 2.49E+02 | 0.00E+00 | 0.00E+00 |
| Initial | 5.69E-17 | 4.66E-18 | 4.57E-08 | 1.69E-09 | 5.58E-27 | 7.91E-19 |
| 1 | 1.57E-17 | 1.60E-18 | 1.37E-10 | 5.47E-09 | 3.24E-27 | 1.18E-19 |
| 2 | 7.15E-18 | 7.33E-19 | 9.34E-11 | 3.45E-09 | 3.44E-28 | 1.26E-19 |
| 3 | 5.91E-18 | 7.42E-19 | 8.81E-11 | 3.39E-09 | 3.65E-28 | 1.34E-19 |
| 4 | 5.48E-18 | 7.92E-19 | 8.31E-11 | 3.36E-09 | 3.84E-28 | 1.41E-19 |
| 5 | 5.28E-18 | 8.42E-19 | 7.84E-11 | 3.34E-09 | 4.11E-28 | 1.47E-19 |

**Table 19.** Convergence history for Problem 6.

A note of warning about the general use of reduced grids is needed here. For fixed $\tau$ and $h_\xi$, it is possible to reduce $h$ enough so that the desired $\rho_j$ is larger than $N$. This may not be a practical problem, since higher accuracy will usually require both increased vertical and horizontal resolution, as well as decreased timesteps.

Finally, it should be noted that the method failed on a test problem with orography, a sigma vertical coordinate system (3.32) with uniform vertical mesh spacing (i.e., $\xi_k = k/K$), and the mesh is defined by $N_\lambda = 64$, $N_\phi = 32$, and $K = 18$. The reason for this is still under study.

## 6. Conclusions.

An FAS FMG solver has been developed which shows promise for the atmospheric general circulation model based on the semi-Lagrangian advection of potential vorticity (PV) presented here. Additional development of the solver is needed, both for robustness and efficiency reasons, as there are test problems for which the method was unable to converge. An important remaining step is to test the solver in the full model in order. Since $\xi$-line relaxation dominates the computation, both in the global sweeps away from the poles and within the MG plane solvers, this is where the most speedup is to be gained. Further system analysis to determine the needed coupling of equations necessary in relaxation and the development of tailored linear solvers are expected to improve efficiency.

# References:

[1] Bates, J.R., F.H.M. Semazzi, R.W. Higgins and S.R.M. Barros, *Integration of the shallow water equations on a sphere using a vector semi-Lagrangian scheme with a multigrid solver.* Mon. Weather Rev., 118 (1990), pp. 1615-1627.

[2] Bates, J.R., Y. Li, A. Brandt, S. F. McCormick and J. Ruge, *A global shallow water numerical model based on the semi-Lagrangian advection of potential vorticity,* Quart. J. Roy. Met. Soc., (in press).

[3] Brandt, A., *Multi-level adaptive solutions to boundary value problems,* Math. Comp., 31 (1977), pp. 333-390.

[4] Brandt, A., *Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics,* GMD Studien 85, Gesellschaft fuer Mathematik und Datenverarbeitung mbH Bonn, Postfach 1240, Schloss Birlinghoven, D-5205 St. Augustin 1, Germany, 1985.

[5] Hoskins, B. J., M. E. McIntyre and A. W. Robertson, *On the use and significance of isentropic potential vorticity maps.* Quart. J. Roy. Met. Soc., 111, 877-946.

[6] McCormick, S.F. (ed.), *Multigrid Methods,* Society for Industrial and Applied Mathematics, Philadelphia, PA (1987).

[7] McCormick, S.F., *Multilevel Adaptive Methods for Partial Differential Equations.* Society for Industrial and Applied Mathematics. Philadelphia, PA (1989).

[8] Ruge, J., Y. Li, S. F. McCormick, A. Brandt and J. R. Bates, *A nonlinear multigrid solver for a semi-Lagrangian potential vorticity vbased shallow water model on the sphere.* SIAM J. Cci. Comput. (in Press).

[9] Stueben, K. and U. Trottenberg, *Multigrid methods: Fundamental algorithms, model problem analysis and applications.* In: Multigrid Methods, W. Hackbusch and U. Trottenberg (Eds), Lecture Notes in Mathematics 960, Springer-Verlag, Berlin, 1982.

[10] Yavneh, I., *A method for devising efficient multigrid smoothers for complicated PDE systems,* SIAM J. Sci. Comput., 14 (1993), 1437-1463.

[11] Yavneh, I., *On red black SOR smoothing in multigrid,* SIAM J. Sci. Comput., 17 (1996), 180-192.

# REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

| 1. AGENCY USE ONLY (Leave blank) | 2. REPORT DATE August 1998 | 3. REPORT TYPE AND DATES COVERED Contractor Report |
|---|---|---|

**4. TITLE AND SUBTITLE**

Final Report: A Nonlinear Multigrid Solver for an Atmospheric General Circulation Model Based on Semi-implicit Semi-Lagrangian Advection of Potential Vorticity

**5. FUNDING NUMBERS**

NAS596076

**6. AUTHOR(S)**

S. McCormick and J. Ruge

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS (ES)**

Front Range Scientific Computations, Inc.
1390 Claremont Dr.
Boulder, CO 80303

**8. PEFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS (ES)**

National Aeronautics and Space Administration
Washington, DC 20546-0001

**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

CR—1999–209231

**11. SUPPLEMENTARY NOTES**

**12a. DISTRIBUTION / AVAILABILITY STATEMENT**

Unclassified - Unlimited
Subject Category:46
Report available from the NASA Center for AeroSpace Information,
Parkway Center/7121 Standard Drive, Hanover, Maryland 21076-1320

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** (Maximum 200 words)

This work represents a part of a project to develop an atmospheric general circulation model based on the semi-Lagrangian advection of potential vorticity (PC) with divergence as the companion prognostic variable.

**14. SUBJECT TERMS**

atmospheric circulation model, semi-Lagrangian, potential vorticity

**15. NUMBER OF PAGES** 47

**16. PRICE CODE**

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | UL |