Multigrid Solution of the Semigeostrophic Invertibility Relation

SCOTT R. FULTON

Department of Mathematics and Computer Science, Clarkson University, Potsdam, New York

(Manuscript received 28 November 1988, in final form 3 April 1989)

ABSTRACT

In semigeostrophic theory one can recover the balanced wind and mass fields from the potential vorticity by solving a nonlinear elliptic boundary-value problem. This paper describes the efficient solution of this invertibility relation in two dimensions by a multigrid algorithm. Numerical results show this method is competitive with a generalized Buneman algorithm for the linearized cases to which the latter applies. The fully nonlinear problem is solved to well below the level of truncation error 50–80 times faster than by simple relaxation on a single grid.

1. Introduction

Introducing an assumption of dynamical balance (such as geostrophic or gradient balance) into the equations governing fluid flow can simplify the dynamics, e.g., by filtering high-frequency waves. It also makes part of the flow field implicit. According to the invertibility principle of Hoskins et al. (1985), if the potential vorticity is known then the corresponding balanced flow can be determined. In practice, this usually means solving an elliptic problem.

Some linear invertibility relations can be solved by existing software; for more complicated problems, simple relaxation can be used (e.g., Schubert and Alworth 1987). However, relaxation is too slow when efficiency is an issue, such as in a predictive model where the problem must be solved at each time step. Multigrid methods hold promise for solving complicated elliptic problems quickly. For example, Ciesielski et al. (1986) used a multigrid method to solve the linear transverse circulation equation in a tropical cyclone.

The present paper describes the efficient solution of the nonlinear semigeostrophic invertibility relation in two dimensions by a multigrid method. Readers unfamiliar with multigrid methods should consult other sources (e.g., Brandt 1977; Fulton et al. 1986; Briggs 1987) for a general introduction, as only the details specific to the problem at hand are given here. In section 2 we review the problem and write it in a form suitable for numerical solution. The discretization and solution method are presented in section 3. Section 4 describes numerical results, and concluding remarks are given in section 5.

2. Problem formulation

We use the semigeostrophic equations as formulated by Schubert et al. (1989) in geostrophic and isentropic coordinates on an f-plane, i.e., using potential temperature $\theta$ as the vertical coordinate and transforming from physical coordinates $(x, y, \theta, \tau)$ to geostrophic coordinates $(X, Y, \Theta, T) = (x + u_g/f, y - u_g/f, \theta, \tau)$, where $u_g$ and $v_g$ are the geostrophic wind components. For frictionless flow, the system reduces to the single predictive equation

$$\frac{\partial \sigma^*}{\partial T} + \frac{\partial}{\partial X} (u_g \sigma^*) + \frac{\partial}{\partial Y} (v_g \sigma^*) + \frac{\partial}{\partial \Theta} (\sigma^*) = 0$$

(2.1)

for the potential pseudodensity (inverse potential vorticity)

$$\sigma^* = f \xi \sigma,$$

(2.2)

where $\sigma = -\partial p/\partial \theta$ is the pseudodensity, $p$ is pressure, and $\xi = f \partial(X, Y)/\partial(x, y)$. The ageostrophic part of the circulation is entirely implicit.

Balancing the simplicity of (2.1) is the complexity of the invertibility relation. If we know the Bernoulli function

$$M^* = \theta \Pi + \phi + \frac{1}{2} (u_g^2 + v_g^2),$$

(2.3)

where $\Pi = c_p (p/p_0)^*\phi$ is the Exner function and $\phi$ is the geopotential, we can recover the wind and mass fields through the geostrophic and hydrostatic relations

$$(f v_g, -f u_g, \Pi) = \left( \frac{\partial M^*}{\partial X}, \frac{\partial M^*}{\partial Y}, \frac{\partial M^*}{\partial \Theta} \right).$$

(2.4)

Obtaining $M^*$ from $\sigma^*$ requires solving a nonlinear
elliptic equation. For two-dimensional flow (i.e., \( \partial / \partial y = \partial / \partial x = 0 \), \( u_x = 0 \), \( v = v_y \)), this equation reduces to

\[
(\frac{f^2}{\partial x^2} - \frac{\partial^2 M^*}{\partial \Theta^2}) \frac{\partial^2 M^*}{\partial X \partial \Theta} + \left( \frac{\partial^2 M^*}{\partial X \partial \Theta} \right)^2 + f^2 \sigma^* = 0, \tag{2.5a}
\]

where \( \Gamma = d\Pi / dp \). Assuming that the top boundary \( \Theta = \Theta_T \) is isobaric \((p = p_T = \text{const})\), (2.4) leads to the condition

\[
\frac{\partial M^*}{\partial \Theta} = \Pi_T \quad \text{at} \quad \Theta = \Theta_T, \tag{2.5b}
\]

where \( \Pi_T = \Pi(p_T) \). At the bottom boundary \( \Theta = \Theta_B \), (2.3) leads to the condition

\[
M^* - \Theta \frac{\partial M^*}{\partial \Theta} - \frac{1}{2f^2} \left( \frac{\partial M^*}{\partial X} \right)^2 = \phi_B \quad \text{at} \quad \Theta = \Theta_B, \tag{2.5c}
\]

where the specified geopotential \( \phi_B \) describes the topography of the bottom boundary. Equations (2.5), together with appropriate lateral boundary conditions, constitute the invertibility relation to be solved. This problem is elliptic provided that \( \sigma^* > 0 \); the uniqueness of physically acceptable solutions (i.e., solutions with positive absolute vorticity) can be proved by the same argument used by Schubert and Alworth (1987, appendix A).

For lateral boundary conditions, suppose that the phenomenon being studied is confined in lateral extent so that in the far field (i.e., for \( |X| \) large), \( \sigma^* \) and \( \tilde{\sigma} \) are independent of \( X \) and \( \phi_B = 0 \). Then it can be shown from (2.5) that the corresponding \( M^* \) is independent of \( X \), so in fact \( \tilde{\sigma} = \tilde{f} \) and \( \sigma = \sigma^* \) in the far field. In this situation (2.5) reduces to a boundary-value problem in \( \Theta \) only, which can be solved for Dirichlet lateral boundary values of \( M^* \).

To isolate the dynamically significant portion of \( M^* \) we subtract out a horizontally uniform basic state which satisfies (2.5) when \( \sigma^* \) is independent of \( X \) and \( \phi_B = 0 \). Denoting the basic state variables by overbars, we specify \( \tilde{\sigma} \) as a function of \( \Theta \) (e.g., as the undisturbed far-field profile of \( \sigma^* \)) and obtain the other fields via

\[
\frac{dM^*}{d\Theta} = \bar{\Pi}, \quad \bar{\Pi} = c_p \left( \frac{\bar{p}}{p_0} \right), \tag{2.6}
\]

\[
\tilde{\Gamma} = \frac{d\bar{\Pi}}{d\bar{p}}, \quad \tilde{\sigma} = -\frac{d\bar{p}}{d\Theta}. \tag{2.6}
\]

Note that

\[
p_B = \bar{p}(\Theta_B) = p_T + \int_{\Theta_B}^{\Theta_T} \tilde{\sigma} d\Theta \tag{2.7}
\]

is a constant; for simplicity we take \( p_0 = p_B \) in the definition of the Exner function. It is convenient to express the problem for the deviation Bernoulli function \( M = M^* - \bar{M}^* \) in terms of the nondimensional coordinates

\[
x = \frac{\tilde{f} X}{c}, \quad z = \frac{\Theta - \Theta_B}{\Theta_T - \theta_B}, \tag{2.8}
\]

where \( c = [aR(\Theta_T - \Theta_B)]^{1/2} \) and \( a = (p_B - p_T)/p_B \). Note that \( x \) now denotes the nondimensional geostrophic coordinate, rather than the physical coordinate used before. Nondimensionalizing \( M \) and \( \phi_B \) by \( c^2 \), \( \sigma^* \) by \( e_0 = (p_B - p_T)/(\Theta_T - \Theta_B) \), \( \Gamma \) by \( \Gamma_0 = R/p_B \), \( \Pi \) by \( c_p \), and \( p \) by \( p_B \) (and similarly for basic state fields), the invertibility relation (2.5) reduces to

\[
\left( 1 - \frac{\partial^2 M}{\partial x^2} \right) \left( \bar{\Gamma} - \frac{\partial^2 M}{\partial z^2} \right) - \left( \frac{\partial^2 M}{\partial x \partial z} \right)^2 = \tilde{\Gamma} \sigma^*, \tag{2.9a}
\]

where \( b = \Theta_B/(\Theta_T - \Theta_B) \) and we have retained the same symbols for nondimensional quantities.

Several variants of (2.9) will also be considered. In the general case, \( \Gamma \) depends on \( M \) through

\[
\Gamma = \Pi^{(\tau - 1)/\kappa}, \quad \Pi = \bar{\Pi} + \kappa a \frac{d\bar{M}}{dz}. \tag{2.10}
\]

Since this dependence is relatively weak, we will also consider the anelastic \((\Gamma = \bar{\Gamma})\) and Boussinesq \((\Gamma = \Gamma = 1)\) cases; the latter corresponds to replacing the dimensional definition of \( \Pi \) by the linear approximation \( \Pi = c_p [1 + \kappa (p - p_0)/p_0] \). Also, the associated linearized problem

\[
-\Gamma \tilde{\sigma} \frac{\partial^2 M}{\partial x^2} - \frac{\partial^2 M}{\partial z^2} = \tilde{\Gamma} (\sigma^* - \tilde{\sigma}), \tag{2.11a}
\]

\[
\frac{\partial M}{\partial z} = 0 \quad \text{at} \quad z = 1, \tag{2.11b}
\]

\[
M - b \frac{\partial M}{\partial z} = \phi_B \text{ at } z = 0 \tag{2.11c}
\]

will be used to provide guidance for the numerical method. Note, however, that (2.11) is not useful in practice, since it effectively assumes \( v_y = 0 \), thus defeating the point of the geostrophic coordinate transformation.

3. Solution method

The linear version of the invertibility relation—(2.11) with Dirichlet lateral boundary conditions—forms a linear elliptic problem, similar to the Poisson problem but with the variable coefficient \( \tilde{\Gamma}(z) \). However, the standard fast algorithms for such problems (e.g., the ADI, conjugate gradient, and Buneman
methods), do not generalize easily to the nonlinear problem (2.9). In this section we discretize the problem on a single grid by finite differences, formulate a simple relaxation scheme, and incorporate this relaxation into an efficient multigrid algorithm.

a. Discretization

The nondimensionalized problem is isotropic when \( \tilde{\sigma} = 1 \), and roughly so in the general case: for example, when \( \tilde{\sigma} = 1 \) then \( \tilde{\Gamma} = (1 - ax)^{-1} \) varies from 1 at \( z = 0 \) to about 3 at \( z = 1 \) for typical values of \( a \). Thus, it is reasonable to use equal mesh spacing \( h \) in \( x \) and \( z \). Using grid points \((x_j, z_k) = (jh, kh)\) for \( j = -N_x, -N_x + 1, \ldots, N_x \) and \( k = -1, 0, \ldots, N_z + 1 \) with \( h = 1/N_x \), we define the discrete approximation \( M_{jk} \) to the solution \( M(x_j, z_k) \) of (2.9) by the discrete equations

\[
[-\tilde{\sigma} D_x^2 M - D_z^2 M + (D_x^2 M)(D_z^2 M)]_{jk} - (D_x^2 M)^2)_{jk} = F_{jk}, \quad (3.1a)
\]

\[
[D_x^2 M]_{jk} = 0 \quad \text{at} \quad k = N_z, \quad (3.1b)
\]

\[
[M - b D_x^2 M - \frac{1}{2} (D_x^2 M)^2]_{jk} = [\phi_B]_j \quad \text{at} \quad k = 0. \quad (3.1c)
\]

Here \( D_x^2 \) and \( D_z^2 \) are the second-order centered difference operators

\[
[D_x^2 M]_{jk} = \frac{M_{j+1,k} - M_{j-1,k}}{2h}, \quad (3.2a)
\]

\[
[D_z^2 M]_{jk} = \frac{M_{j,k+1} - 2M_{j,k} + M_{j,k+1}}{h^2}, \quad (3.2b)
\]

\( D_x^2 \) and \( D_z^2 \) are analogous, and the right-hand side of (3.1a) is given by

\[
F_{jk} = \begin{cases} 
\sigma_{jk}^* - \tilde{\sigma}_k, & \text{Boussinesq} \\
\tilde{\Gamma}_k (\sigma_{jk}^* - \tilde{\sigma}_k), & \text{anelastic} \\
\Gamma_{jk} (\sigma_{jk}^* - \tilde{\sigma}_k), & \text{general}
\end{cases} \quad (3.3)
\]

Note that in the general case \( F_{jk} \) depends on the solution \( M \) through

\[
\Gamma_{jk} = \Pi_{jk}^{(s-1)/2}, \quad \Pi_{jk} = \Pi_k + \kappa a [D_x^2 M]_{jk}, \quad (3.4)
\]

which is the discrete analogue of (2.10). The interior equation (3.1a) is applied for \( j = -N_x, \ldots, N_x - 1 \) and \( k = 0, \ldots, N_z \), and the vertical boundary conditions (3.1b) and (3.1c) use centered differences with ghost points as \( k = -1 \) and \( k = N_z + 1 \). The discretization of the linear problem (2.11) is similar.

The lateral boundaries are located at \( x = \pm L \) with \( L = N_x h = N_x/N_z \), where we require that \( M \) be independent of \( x \) as discussed in section 2. Both the nonlinear and linear discrete problems then reduce to

\[
-[D_z^2 M]_{jk} = F_{jk}, \quad k = 0, \ldots, N_z, \quad (3.5a)
\]

\[
[M_{j,k} - b [D_x^2 M]_{jk} = [\phi_B], \quad k = 0, \quad (3.5b)
\]

at \( j = \pm N_x \). In the Boussinesq and anelastic cases (3.5) is a diagonally dominant tridiagonal linear system, which can be solved easily by Gaussian elimination for Dirichlet lateral boundary values of \( M \); independent of the interior equations. In the general case the system (3.5) is nonlinear due to coupling through (3.4), but can be solved by Newton iteration, again independent of the interior equations.

b. Relaxation

Gauss–Seidel relaxation is a simple iterative scheme for solving discrete systems of equations. For the linear discrete problem the details are well known. For the nonlinear discrete problem we take the same approach, sweeping through the grid point by point, at each point \((j, k)\) updating the approximate solution so the discrete equation at that point is satisfied using the current surrounding values. With some algebra, the discrete interior equation (3.1a) reduces to

\[
a_4 (M_{j,k} - A_{j,k})(M_{j,k} - B_{j,k}) = C_{j,k}, \quad (3.6)
\]

where

\[
A_{j,k} = \frac{1}{2} (M_{j-1,k} + M_{j+1,k} - h^2), \quad (3.7)
\]

\[
B_{j,k} = \frac{1}{2} (M_{j,k-1} + M_{j,k+1} - h^2 \tilde{\Gamma}_k \tilde{\sigma}_k), \quad (3.7)
\]

\[
C_{j,k} = h^4 (F_{jk} + \tilde{\Gamma}_k \tilde{\sigma}_k) + D_{jk}^2, \quad (3.7)
\]

\[
D_{jk} = \frac{1}{4} (M_{j+1,k+1} - M_{j+1,k-1}) - M_{j-1,k-1} - M_{j-1,k+1} - M_{j-1,k+1}). \quad (3.7)
\]

Regarding the surrounding values as known, (3.6) is a quadratic equation for \( M_{j,k} \); solving it gives the Gauss–Seidel relaxation equation

\[
M_{j,k} \leftarrow \frac{1}{2} \left\{ (A_{j,k} + B_{j,k}) \right. \right. 
+ \left. \left. \left[ (A_{j,k} - B_{j,k})^2 + C_{j,k} \right] \right)^{1/2} \right\}, \quad (3.8)
\]

where tildes denote quantities associated with the current approximation. The larger root is taken in (3.8) so that \( A_{j,k} - B_{j,k} > 0 \) as required for positive absolute vorticity [since \( f/\psi = 1 - (1/\psi) \partial v_y / \partial x \) in dimensional terms].

One relaxation sweep consists of relaxing the interior equations at \( j = -N_x, \ldots, N_x - 1 \) and \( k = 0, \ldots, N_z \) using (3.8) and updating the ghost-point values at \( k = -1 \) and \( k = N_z + 1 \) by imposing the vertical boundary conditions (3.1b) and (3.1c) (the ghost-point values could also be eliminated using the boundary conditions, but the coding would be more awkward). We consider two orderings of equations for relaxation.
With lexicographic (LEX) ordering, the interior equations are relaxed left to right first along row \( k = 0 \), then along row \( k = 1 \), etc., and the ghost-point values are updated after the interior sweep is completed. With red-black (RB) ordering, the values are updated first at the "red" points where \( j + k \) is even (relaxing the interior equations and then imposing the boundary conditions), and then at the "black" points where \( j + k \) is odd. In the general case where the right-hand side of (3.1) depends on the solution, we also update the values \( F_{jk} \) after each sweep from the new approximate solution using (3.3) and (3.4). With \( \kappa = \gamma_{r} \), the exponent in (3.4) is \(-\gamma_{r}\); computing this term using a square root rather than an exponent saves a significant amount of computer time, reducing the overall time per sweep by about 30%.

\[ \kappa = \gamma_{r}, \quad \text{exponent in (3.4) is } -\gamma_{r}; \text{computing this term using a square root rather than an exponent saves a significant amount of computer time, reducing the overall time per sweep by about 30%}. \]

\section{Multigrid processing}

As a solution method, Gauss-Seidel relaxation converges slowly, especially when the mesh size \( h \) is small. It can be accelerated by introducing a relaxation parameter \( \omega \) (yielding the SOR method), but even with \( \omega \) tuned to the problem (by trial and error) for fastest convergence, the convergence rate still deteriorates as \( h \) decreases. In contrast, multigrid processing gives much faster convergence—with a rate independent of \( h \)—by cycling between several grids with different mesh sizes and using relaxation to smooth the error efficiently on the scale of each grid. Multigrid methods were reviewed by Fulton et al. (1986); here we give only the details specific to this problem.

We define a sequence of grids (or levels), identifying the original grid on which the discrete problem is to be solved as level \( l = 1 \) and introducing coarser grids \( l = 2, 3, \ldots, m \) by successively doubling the mesh size. Thus, grid level \( l \) has \( N_x^{(l)} \) and \( N_z^{(l)} \) intervals in \( x \) and \( z \) with mesh spacing \( h_l = 2h_{l-1} = 1/N_z^{(l)} \). The coarsest grid \( l = m \) may have as few as two grid intervals in \( x \) and one in \( z \) (i.e., \( N_x^{(m)} > 1 \) and \( N_z^{(m)} > 1 \)). Associated with each grid level is a discrete problem

\[ L^h \mathbf{M}^h = F^h, \quad (3.9) \]

where the operator \( L^h \) represents the left-hand side of (3.1) for the mesh spacing \( h = h_l \) of that grid. Gauss-Seidel relaxation as described above is used to smooth the high-wavenumber error on each grid.

On the finest grid (\( l = 1 \)) the discrete problem (3.9) is simply the original problem (3.1). Since this is nonlinear, the equations on the coarser grids (\( l > 1 \)) must be defined using the full approximation scheme (FAS). Thus, we define the right-hand side of (3.9) on the coarser grids by

\[ F^{2h} = L^{2h}(\bar{I}_h^{2h}\bar{M}^h) + I_h^{2h}r^h. \quad (3.10a) \]

Here \( \bar{M}^h \) is the current approximate solution on a given grid \( h \), \( r^h = F^h - L^h \bar{M}^h \) is the corresponding residual, and \( \bar{I}_h^{2h} \) and \( I_h^{2h} \) represent the fine-to-coarse transfer of the solution and residual, respectively, which will be specified below. Since the boundary conditions are imposed after each relaxation sweep, the right-hand side of the coarse-grid boundary conditions is always zero. The coarse-grid solution \( \mathbf{M}^{2h} \) is then used to correct the fine-grid solution via

\[ \bar{M}^h \leftarrow \bar{M}^h + I_h^{2h}(\mathbf{M}^{2h} - \bar{I}_h^{2h}\bar{M}^h), \quad (3.10b) \]

where \( I_h^{2h} \) represents the coarse-to-fine transfer of the correction.

We use injection for the fine-to-coarse solution transfer operator \( \bar{I}_h^{2h} \) and bilinear interpolation for the coarse-to-fine correction transfer operator \( I_h^{2h} \). For the fine-to-coarse residual transfer operator \( I_h^{2h} \) we use full weighting; on the interior (0 < \( k < N_x^{(l)} \)) the standard weights

\[ \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \quad (3.11a) \]

apply, while at the upper and lower boundaries the appropriate weights are

\[ \frac{1}{16} \begin{bmatrix} 0 & 0 & 0 \\ 2 & 4 & 2 \\ 2 & 4 & 2 \end{bmatrix}, \quad \frac{1}{16} \begin{bmatrix} 2 & 4 & 2 \\ 0 & 0 & 0 \end{bmatrix}, \quad (3.11b) \]

respectively (Stüben and Trotttenberg 1982, §7.6). Two points deserve special mention. First, if we integrate (2.11a) over the domain we see that the integral of the right-hand side depends only on boundary terms involving \( \mathbf{M} \) and its derivatives; this integral is approximately preserved by the full weighting (3.11). In contrast, if we divide the interior equation by \( \bar{I} \) this property is lost, and significant errors are introduced by the grid transfers. Thus, it is important that \( L^h \) be based on (3.1) and not be divided by \( \bar{I} \). Second, during the initial stages of the multigrid processing, the right-hand side \( F^{2h} \) generated from (3.10a) may on some coarse grids correspond—through (3.3)—to negative \( \sigma \). If this occurs, we adjust the values of \( F^{2h} \) where necessary to correspond to \( \sigma^* = 0 \), thus guaranteeing that the argument of the square root in (3.8) is non-negative.

Three control algorithms were investigated. The simplest (for comparison purposes) is single-grid relaxation (SGR), which uses repeated relaxation sweeps on the original (finest) grid. The multigrid cycling (MGC) algorithm solves the problem by repeated multigrid V-cycles, starting on the finest grid. Finally, the full multigrid (FMG) algorithm starts by solving on the coarsest grid (by five relaxation sweeps), and then for each finer grid in turn obtains an initial approximation from the previous grid using bicubic interpolation and solves using one multigrid V-cycle. Initial approximations for the SGR and MGC algorithms (on the finest grid) and the FMG algorithm (on the coarsest grid) are constructed by linear interpolation in \( x \) between the lateral boundary values. For sim-
plicity, we use fixed numbers of relaxation sweeps in the downward (fine-to-coarse) and upward (coarse-to-
the next finer grid) branches of the V-cycles, chosen by numerical experimentation as discussed in the following section. Within a V-cycle, the solution on the coarsest grid is relaxed until the residual norm is less than 0.05 times the residual norm on the next finer grid.

4. Numerical results

In this section we show that the multigrid method described above solves the problem correctly, and compare its efficiency with that of relaxation on a single grid.

a. Test case

To test the solution method we specify the analytical solution

$$M(x, z) = \frac{\cos(\lambda(1-z))}{\cosh(\lambda x)}, \quad (4.1)$$

where $\lambda$ solves $h \tan \lambda = 1$. The vertical dependence of (4.1) corresponds to a vertical normal mode of the linear problem (2.11) with $\beta a = 1$; the horizontal dependence is such that $M$ decays for large $|x|$ with the corresponding $\epsilon$-folding width $1/\lambda$. We derive data $\sigma^*$ and $\phi_B$ by substituting (4.1) into the continuous equations (2.9) or (2.11) and evaluating the right-hand side analytically. We then measure how well the numerical method can reconstruct (4.1) and its derivatives from this data. For all results reported here we use the values $p_B = 10^5$ Pa, $p_T = 2 \times 10^4$ Pa, $\Theta_B = 290$ K, $\Theta_T = 330$ K, and $f = 10^{-4}$ s$^{-1}$, which give $a = 0.8$, $b = 7.25$, and $c \approx 95.83$ m s$^{-1}$, and choose the second vertical mode with $\lambda \approx 6.3051$ and $M_0 = 0.02$ (higher values of $M_0$ give $\sigma^* \leq 0$ in parts of the domain). Execution times reported are for single precision on a SUN 3/50 workstation with MC68881 coprocessor.

b. Convergence factors

To measure convergence it is convenient to use the norm of the dynamic residual, i.e., the residual computed during a relaxation sweep. For the relaxation scheme (3.8) the value of the dynamic residual at a point is

$$\tilde{r}_{jk} = \frac{1}{h^4} \left[ \tilde{\phi}_{jk} - 4(\tilde{M}_{jk} - \tilde{A}_{jk})(\tilde{M}_{jk} - \tilde{B}_{jk}) \right]. \quad (4.2)$$

We measure the size of $\tilde{r}$ (or any function $\psi$ defined on the grid) by the discrete $l_2$ norm

$$\| \psi \| = \left( \frac{1}{2N_x N_z} \sum_{j=0}^{N_x-1} \sum_{k=0}^{N_z-1} \psi_{jk}^2 \right)^{1/2}. \quad (4.3)$$

For relaxation on a single grid we define the convergence factor per sweep $\mu_z$ for the residuals associated with two successive sweeps. Likewise, for multigrid cycling we define the convergence factor per cycle $\mu_c$ as the ratio of the residual norms associated with the final fine-grid relaxation sweep of two successive cycles, and define the effective convergence factor per sweep $\tilde{\mu}_z$ as $\mu_z^{1/n}$, where $n$ is the number of fine-grid sweeps per cycle.

Table 1 shows the observed asymptotic convergence factor per sweep $\mu_z$ for multigrid cycling using lexicographic (LEX) and red-black (RB) ordering; the numbers in parentheses are the numbers of relaxation sweeps in the downward and upward branches of the V-cycles. The values shown are for the test case (4.1) with $N_x = 64$ and $N_z = 32$, but nearly identical values were obtained for other grid sizes, other vertical modes, and for the squall-line problem studied in section 3 of Schubert et al. (1989). For comparison, we also list the theoretical multigrid smoothing factor $\mu$. The linear Boussinesq problem is equivalent to the Poisson problem, for which $\mu$ is well known (e.g., Stib and Trottenberg 1982). For the non-Boussinesq cases, $\mu$ is given for the linearized problem with $\tilde{\Gamma}$ frozen at 3 using the smoothing analysis described in Brandt (1984, §3.1). The numerical results agree well with the theoretical values, and show that the multigrid method converges reasonably fast. In contrast, relaxation on a single grid gives $\mu_z > 0.99$ for all cases reported in the table (both with LEX and RB ordering).

For the multigrid method, smaller $\tilde{\mu}$ and $\mu_z$ values could be obtained in the non-Boussinesq cases by z-line relaxation, but the added complexity (and computational cost) of solving a nonlinear system for each line would make the gain in efficiency marginal at best. A more reasonable approach would be to use slightly different mesh spacing in $x$ and $z$ to render the problem more nearly isotropic on the grid. Again, however, the possible gains in efficiency would be small.

c. Accuracy and efficiency

Asymptotic convergence factors may be misleading, since for the first few sweeps or cycles $\mu_z$ is typically smaller than its asymptotic value (until the error mode which limits the convergence rate dominates). Table 2 lists the time required to solve the test problem (4.1)
with \( N_x = 64 \) and \( N_z = 32 \) by the FMG method using RB(1, 2) relaxation. The increased time required for the nonlinear cases is due primarily to the additional terms in the equations; updating the value of \( \Gamma \) each sweep in the general case also increases the execution time. Also shown is the norm of the final (dynamic) residual, normalized by the norm of the local truncation error \( \tau^h = F^h - L^h M^h \) on the finest grid, where \( M^h \) is the analytical solution (4.1). We see that the FMG algorithm with RB(1, 2) relaxation solves the problem well below the level of truncation error. Other choices of relaxation and numbers of sweeps (results not shown) gave comparable (or occasionally larger) errors but consistently larger residuals, by factors of about 3, 5, and 4 for the LEX(1, 2), LEX(2, 1), RB(1, 1), and RB(2, 1) schemes, respectively. Execution times were essentially the same, except for the RB(1, 1) scheme, which took about 25% less time but failed to solve to below truncation error. Thus, for all further results we use only the RB(1, 2) scheme.

Also shown in Table 2 are the corresponding normalized errors \( \| \mathcal{M} - \mathcal{M}^h \| / \| \mathcal{M} \|, \| \nu \| - \nu^h \| / \| \nu \| \) and \( \| \Pi - \Pi^h \| / \| \Pi \| \). Here \( \mathcal{M} \) is the analytical solution (4.1), \( \nu \) is the analytical solution, \( \partial \mathcal{M} / \partial \mathcal{x} \), and \( \Pi = k \alpha \partial \mathcal{M} / \partial \mathcal{z} \), while the corresponding discrete fields are \( \mathcal{M}^h, \nu^h = \partial \mathcal{M}^h / \partial \mathcal{x}, \) and \( \Pi^h = k \alpha \partial \mathcal{M}^h / \partial \mathcal{z} \) (only the deviation of \( \Pi \) from the basic state is considered). The errors in \( \nu \) and \( \Pi \) are nearly identical to those due to discretization errors alone (as determined by further iteration). In contrast, the errors in \( \mathcal{M} \) are in most cases somewhat larger than the discretization errors, reflecting the fact that the mean value of \( \mathcal{M} \) is only weakly determined by the governing equations (through the lower boundary condition). Since only the derivatives of \( \mathcal{M} \) are of dynamical significance, the errors in \( \mathcal{M} \) are less important; they could be reduced if desired by an additional V-cycle following the FMG algorithm.

To put the execution times in perspective, we also solved the linear cases by the generalized Buneman algorithm of Sweet (1977). This direct solver gives zero residuals (neglecting roundoff error), but the solution errors were essentially the same as those of the FMG method (except for the \( \mathcal{M} \) error in the anelastic case, which was 0.003). Asymptotically, the Buneman method requires more operations than the FMG method \( \mathcal{O}(N_x N_z \log_2 N_z) \) versus \( \mathcal{O}(N_x N_z) \), but for this problem size it required 3.8 seconds, or about 40% less time than the FMG method. However, the Buneman method works only for the linear cases.

Results for the MGC method are given in Table 3, using the residual norm determined by the FMG method as the convergence tolerance. Although the resulting residuals are somewhat smaller (since only complete cycles can be made), the resulting errors are comparable, thus illustrating that solving much below the level of truncation error is not necessary. Since at least three cycles are required, the execution time is 2 to 3.5 times that required by the FMG method. The nonlinear cases require more cycles, an effect not predicted by the asymptotic convergence factors given in Table 1. This occurs because during the first few cycles, the FAS right-hand side on the coarser grids corresponds to \( \alpha^* \leq \alpha^* \) at some points as discussed in section 3; adjusting these values allows the solution to converge but degrades the convergence rate.

Finally, Table 4 shows corresponding results for the SGR method (with RB ordering), again using the residual norm determined by the FMG method as the convergence tolerance. The resulting errors in the solution are comparable to those of the FMG method, but the execution time is 30 to 80 times greater.

d. Error estimates

A dividend of the FAS mode of multigrid processing is the ability to generate estimates of the truncation

---

Table 2. Accuracy and efficiency for FMG solution.

<table>
<thead>
<tr>
<th></th>
<th>Time (sec)</th>
<th>Normalized residual</th>
<th>Normalized errors</th>
<th>Normalized errors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>( M )</td>
<td>( \nu )</td>
<td>II</td>
</tr>
<tr>
<td>Linear</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boussinesq</td>
<td>6.53</td>
<td>0.222</td>
<td>0.004</td>
<td>0.007</td>
</tr>
<tr>
<td>anelastic</td>
<td>6.59</td>
<td>0.293</td>
<td>0.034</td>
<td>0.010</td>
</tr>
<tr>
<td>Nonlinear</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boussinesq</td>
<td>12.73</td>
<td>0.194</td>
<td>0.056</td>
<td>0.016</td>
</tr>
<tr>
<td>anelastic</td>
<td>12.30</td>
<td>0.242</td>
<td>0.075</td>
<td>0.018</td>
</tr>
<tr>
<td>general</td>
<td>16.56</td>
<td>0.394</td>
<td>0.061</td>
<td>0.015</td>
</tr>
</tbody>
</table>

Table 3. Accuracy and efficiency for MGC solution.

<table>
<thead>
<tr>
<th></th>
<th>Time (sec)</th>
<th>Multigrid cycles</th>
<th>Normalized residual</th>
<th>Normalized errors</th>
<th>Normalized errors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>( M )</td>
<td>( \nu )</td>
<td>II</td>
</tr>
<tr>
<td>Linear</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boussinesq</td>
<td>13.1</td>
<td>3</td>
<td>0.067</td>
<td>0.003</td>
<td>0.008</td>
</tr>
<tr>
<td>anelastic</td>
<td>13.2</td>
<td>3</td>
<td>0.136</td>
<td>0.006</td>
<td>0.008</td>
</tr>
<tr>
<td>Nonlinear</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boussinesq</td>
<td>44.1</td>
<td>5</td>
<td>0.132</td>
<td>0.073</td>
<td>0.019</td>
</tr>
<tr>
<td>anelastic</td>
<td>34.1</td>
<td>4</td>
<td>0.228</td>
<td>0.112</td>
<td>0.026</td>
</tr>
<tr>
<td>general</td>
<td>54.7</td>
<td>5</td>
<td>0.129</td>
<td>0.019</td>
<td>0.010</td>
</tr>
</tbody>
</table>
Table 4. Accuracy and efficiency for SGR solution.

<table>
<thead>
<tr>
<th></th>
<th>Time (sec)</th>
<th>Relaxation sweeps</th>
<th>Normalized residual</th>
<th>Normalized errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boussinesq</td>
<td>219</td>
<td>308</td>
<td>0.218</td>
<td>0.002</td>
</tr>
<tr>
<td>anelastic</td>
<td>419</td>
<td>588</td>
<td>0.293</td>
<td>0.030</td>
</tr>
<tr>
<td>Nonlinear</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boussinesq</td>
<td>1007</td>
<td>705</td>
<td>0.194</td>
<td>0.109</td>
</tr>
<tr>
<td>anelastic</td>
<td>957</td>
<td>670</td>
<td>0.241</td>
<td>0.105</td>
</tr>
<tr>
<td>general</td>
<td>767</td>
<td>354</td>
<td>0.393</td>
<td>0.083</td>
</tr>
</tbody>
</table>

error. This is accomplished by computing the relative local truncation error

\[ \tau_h^{2h} = F_h^{2h} - f_h^{2h} F_h^h, \]  

using injection for the transfer of \( F \), and approximating the truncation error \( \tau_h \) by \( \frac{1}{2} \tau_h^{2h} \) (Brandt 1984, sec. 8.4). Table 5 shows the norm of the estimated truncation error (normalized by the norm of the actual truncation error) for the test problem (4.1), with size \( N_x \times N_z \) as indicated. The estimate is accurate to within 15% in all cases, and usually is much closer. These estimates could be used to give a stopping tolerance for multigrid iteration, although the FMG method automatically solves below truncation error as seen in Table 2. Alternatively, estimates of the local truncation error could be used in mesh refinement criteria (e.g., Skamarock 1989) or to provide high-order accuracy through defect corrections (e.g., Brandt 1984, sec 10.1).

In the FMG method one can similarly estimate the solution error by comparing the final solution on a level \( l + 1 \) (before interpolating to level \( l \)) to the final solution on level \( l \) at points common to the two grids. Values of the norm of the estimated solution error (normalized by the norm of the actual solution error) are also given in Table 5. These results and further experimentation with other mesh sizes and test problems indicate that the solution error estimates are less reliable than the truncation error estimates, but may still be valuable as a rough guide. More reliable estimates can be obtained, e.g., by using two V-cycles to solve on each grid level in the FMG method, but only at the cost of increasing the execution time substantially.

5. Concluding remarks

The multigrid method described here solves the semigeostrophic invertibility relation efficiently. For the linear cases it is competitive with the generalized Buneman algorithm and much faster than simple relaxation. The nonlinear cases are solved with the same efficiency (taking into account the extra operations needed to simply express the nonlinear discrete equations.) Among the variations tested, the FMG algorithm with RB(1, 2) relaxation produced the best results, solving well below the level of truncation error and generating useful error estimates.

In the procedure described here we assumed the lower boundary to be an isentropic surface. Andrews (1983) has shown that this restriction can be removed by picturing the isentropic surfaces which intersect the ground as continuing just below the ground, giving \( z^* = 0 \) there. In this case the invertibility relation (2.9) would remain the same except that the lower boundary of the domain would become a known function \( z = z_g(x) \), at which (2.9c) would apply. Since the boundary would not coincide with a grid line, the discretization and grid transfers there would be more complicated, but the overall approach should remain the same.

The techniques used here should generalize to the corresponding three-dimensional case, and should apply equally well to similar nonlinear elliptic problems. A fast solver for the invertibility relation opens the door to using the semigeostrophic equations in a predictive (rather than diagnostic) sense in a reasonable amount of computer time.

Acknowledgments. The author had many valuable discussions with Wayne H. Schubert, who proposed the problem treated in this paper. Thanks also go to Rolf Hertenstein and Michael Felland for helpful comments. This work was supported by the National Science Foundation under Grants ATM-8618079 and ATM-8814541.
REFERENCES