NASA Technical Memorandum 104567

A Fast Direct Solver for a Class of Two-Dimensional Separable Elliptic Equations on the Sphere

Shrinivas Moorthi and R. Wayne Higgins
Universities Space Research Association
NASA Goddard Space Flight Center
Greenbelt, Maryland

National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland 20771

1992

TABLE OF CONTENTS

Abstract

An efficient, direct, second order solver for the discrete solution of a class of two-dimensional separable elliptic equations on the sphere is presented. The method involves a Fourier transformation in longitude and a direct solution of the resulting coupled second order finite-difference equations in latitude. The solver is made efficient by vectorizing over longitudinal wavenumber and by using a vectorized fast Fourier transform routine. It is evaluated using a prescribed solution method and compared with a multigrid solver and the standard direct solver from FISHPAK.

I. Introduction

Numerical techniques used in global atmospheric models have evolved over the past several decades. Models with explicit time differencing generally require very small time steps in order to avoid linear computational instability associated with fast moving gravity waves, particularly because of the convergence of meridians near the pole. The introduction of semi-implicit time differencing (Robert, 1969) relaxed the requirement for linear computational stability and allowed larger time steps relative to explicit schemes. Further advances occurred through the introduction of semi-Lagrangian semi-implicit time differencing schemes (see Staniforth and Cote, 1991 and Bates et al., 1992 for a comprehensive review of the evolution of the semi-Lagrangian approach).

The implicit (or semi-implicit) time differencing, in general, leads to an elliptic equation (two- or three-dimensional and separable or non-separable depending on the formulation) on the sphere (Temperton and Staniforth, 1987; McDonald and Bates, 1989; Tanguay et al., 1989; Bates et al., 1990; Barros et al., 1989). Thus for the implicit schemes to be more economical compared to their explicit counterparts, efficient solvers are of paramount importance.

Algorithms for the direct solution of separable elliptic equations have been around for awhile. Swarztrauber (1974a) developed a method of direct solution of separable elliptic equations by extending the stabilized cyclic reduction algorithm. When the coefficients of the elliptic equations are independent of one of the dimensions (which is the class of equations we are considering here), the problem can be solved with the application of Fourier analysis. Hockney (1965) used this approach to obtain the direct solution of Poisson's equation. Lindzen and Kuo (1969) suggested the use of a Fourier transform in one direction combined with the direct solution of the resulting second order ordinary differential equations in the other, for more general elliptic equations. Le Bail (1972) applied fast Fourier transforms (FFT's) to solve a class of partial differential equations.

Until recently, fast direct solvers on the sphere have been available for the discrete Poisson or Helmholtz type elliptic equations (Swarztrauber and Sweet, 1973, 1975; Sweet, 1973, 1974; Swarztrauber, 1974; Adams et al., 1980). Thus many implicit time differencing schemes have been geared towards obtaining such elliptic equations (e.g. McDonald and Bates, 1989). Bates et al., (1990) (hereafter BSHB) were the first to obtain a more general elliptic equation for their vector semi-Lagrangian semi-implicit time differencing scheme in the global shallow water framework. Because no efficient direct solver was available at that time, they used the multigrid solver developed by Barros (1991); multigrid methods can be applied to more general elliptic equations and more complex domains (Phillips, 1984; Fulton et al.,1986; Barros et al., 1989; Bates et al., 1990; Barros, 1991). However, when the scheme of BSHB was extended to a global multi-level primitive equation model (Bates et al., 1992), it was soon realized that the original multigrid solver was not efficient enough. This led to the development of the FFT based direct solver presented here.

It should be emphasized that the idea of using FFT's is not new; they are routinely used to solve Poisson and Helmholtz equations on the sphere. Recently, Cote and Staniforth (1990) have also described and applied this approach to a more general elliptic problem.

Our solver is based on a latitude/longitude grid over the sphere. The solution method involves a Fourier decomposition in longitude to separate that dependency. This reduces the problem to a set of coupled ordinary differential equations in the latitudinal direction for each longitudinal wavenumber. These coupled equations are solved using the procedure described in Lindzen and Kuo (1969) and Chao (1979).

In Section II we present the elliptic equation, the separation of longitudinal dependency, the latitudinal discretization, the method of solution for two coupled ordinary differential equations and a brief description of the coding strategy and the calling sequence for the solver. In Section III, we validate the algorithm and compare it with the multigrid method used by BSHB. A summary is given in Section IV. The Appendix contains a listing of the source code for the solver.

II. Description of the solver

a) The elliptic equation

In this report we present a fast and direct algorithm to solve the following class of two-dimensional elliptic equations on the sphere:

$$\frac{c_1(\theta)}{a^2\cos^2\theta}\frac{\partial^2\phi}{\partial\lambda^2}+\frac{c_2(\theta)}{a^2\cos\theta}\frac{\partial^2\phi}{\partial\lambda\partial\theta}+\frac{1}{a^2\cos\theta}\frac{\partial}{\partial\theta}(c_3(\theta)\cos\theta\frac{\partial\phi}{\partial\theta})$$
$$+\frac{c_4(\theta)}{a\cos\theta}\frac{\partial\phi}{\partial\lambda}+\frac{1}{a\cos\theta}\frac{\partial}{\partial\theta}(c_5(\theta)\cos\theta\,\phi)+c_6(\theta)\phi=F \qquad (1)$$

where a is the radius of the sphere, λ and θ are the longitude and latitude, respectively, ϕ is the solution, and F is the forcing (which is known). Here the coefficients $c_1(\theta)$, $c_2(\theta)$, $c_3(\theta)$, $c_4(\theta)$, $c_5(\theta)$, and, $c_6(\theta)$ are at most functions of latitude. If any of these coefficients is also a function of longitude, then it is nearly impossible to write an efficient direct solver for such an equation in which case the multigrid method would be preferable (Phillips,1984).

b) Separation of longitudinal dependency

We solve (1) on a uniform longitude and latitude grid on the sphere. Any field defined on this grid can be expanded into a finite Fourier series in the longitudinal direction λ. Supposing that we have I equally spaced grid points along a latitude circle, any function ϕ can be expanded in a finite Fourier series as

$$\phi(\lambda_n)=\sum_{-I/2}^{I/2}\hat{\phi}(k)\,e^{ik\lambda_n}. \qquad (2)$$

Here $\hat{\phi}(k)$ is the complex amplitude for wavenumber k and $\hat{\phi}(-k)$ is the complex conjugate of $\hat{\phi}(k)$ where $i = \sqrt{-1}$. The complex amplitude can be obtained by

$$\hat{\phi}(k) = \frac{1}{I}\sum_{1}^{I} \phi(\lambda_n)\, e^{-ik\lambda_n}. \tag{3}$$

Since we are considering real data, for k =0 (i.e. the longitudinal mean part) and k =I/2, only the real part of the amplitude functions exist.

We first consider the case when $0<k<I/2$. For any wavenumber k, eq. (1) reduces to a second order ordinary differential equation in θ for the complex amplitude. Without loss of generality, let us consider a solution to (1) of the form

$$\phi = \hat{\phi} e^{ik\lambda}, \text{and } F = \hat{F} e^{ik\lambda} \tag{4}$$

where $\hat{\phi} = \phi^r + i\phi^i$, $\hat{F} = F^r + iF^i$. After substituting (4) into (1) we obtain

$$\frac{-k^2 c_1(\theta)}{a^2\cos^2\theta}\hat{\phi} + \frac{ikc_2(\theta)}{a^2\cos\theta}\frac{\partial\hat{\phi}}{\partial\theta} + \frac{1}{a^2\cos\theta}\frac{\partial}{\partial\theta}\left(c_3(\theta)\cos\theta\frac{\partial\hat{\phi}}{\partial\theta}\right)$$
$$+\frac{ikc_4(\theta)}{a\cos\theta}\hat{\phi} + \frac{1}{a\cos\theta}\frac{\partial}{\partial\theta}\left(c_5(\theta)\cos\theta\,\hat{\phi}\right) + c_6(\theta)\hat{\phi} = \hat{F}. \tag{5}$$

Separating (5) into real and imaginary parts, we obtain the following two coupled second order ordinary differential equations:

$$\frac{-k^2 c_1(\theta)}{a^2\cos^2\theta}\phi^r - \frac{kc_2(\theta)}{a^2\cos\theta}\frac{\partial\phi^i}{\partial\theta} + \frac{1}{a^2\cos\theta}\frac{\partial}{\partial\theta}\left(c_3(\theta)\cos\theta\frac{\partial\phi^r}{\partial\theta}\right)$$
$$-\frac{kc_4(\theta)}{a\cos\theta}\phi^i + \frac{1}{a\cos\theta}\frac{\partial}{\partial\theta}\left(c_5(\theta)\cos\theta\,\phi^r\right) + c_6(\theta)\phi^r = F^r, \tag{6}$$

and

$$\frac{-k^2 c_1(\theta)}{a^2\cos^2\theta}\phi^i + \frac{kc_2(\theta)}{a^2\cos\theta}\frac{\partial\phi^r}{\partial\theta} + \frac{1}{a^2\cos\theta}\frac{\partial}{\partial\theta}\left(c_3(\theta)\cos\theta\frac{\partial\phi^i}{\partial\theta}\right)$$
$$+\frac{kc_4(\theta)}{a\cos\theta}\phi^r + \frac{1}{a\cos\theta}\frac{\partial}{\partial\theta}\left(c_5(\theta)\cos\theta\,\phi^i\right) + c_6(\theta)\phi^i = F^i. \tag{7}$$

Equations (6) and (7) determine the latitudinal structures of the amplitude for $k>0$. When $k=0$ (i.e. the longitudinal mean part) only the real part exists and we obtain a single second order ordinary differential equation. Denoting the longitudinal mean part by an overbar, we have

$$\frac{1}{a^2\cos\theta}\frac{\partial}{\partial\theta}(c_3(\theta)\cos\theta\frac{\partial\bar{\phi}}{\partial\theta})+\frac{1}{a\cos\theta}\frac{\partial}{\partial\theta}(c_5(\theta)\cos\theta\,\bar{\phi})+c_6(\theta)\bar{\phi}=\bar{F}. \tag{8}$$

Finally, since the imaginary part is zero for k =I/2, the problem reduces to a single second order ordinary differential equation governing the real part. This is obtained by ignoring the terms with imaginary part in (6).

c) Latitudinal discretization

We represent the distance from the south pole to the north pole of the sphere by a set of J equally spaced grid points. Each grid point is referred to by an integer index j, with j=1 and j=J representing the south and north poles, respectively. For interior points of the domain, we approximate the latitudinal derivatives in (6) and (7) by second order finite differences. Since the elliptic operator is not formally defined at the poles, we use integral definitions there, as detailed later. Thus for the interior, we assume

$$\frac{1}{a^2\cos\theta_j}\frac{\partial}{\partial\theta}(c_3\cos\theta\frac{\partial}{\partial\theta})\phi_j=\frac{(c_3\cos\theta)_{j+\frac{1}{2}}(\phi_{j+1}-\phi_j)-(c_3\cos\theta)_{j-\frac{1}{2}}(\phi_j-\phi_{j-1})}{a^2\cos\theta_j(\Delta\theta)^2}$$

and

$$\frac{1}{a\cos\theta_j}\frac{\partial}{\partial\theta}(c_5\cos\theta\,\phi)_j=\frac{(c_5\cos\theta)_{j+\frac{1}{2}}(\phi_{j+1}+\phi_j)-(c_5\cos\theta)_{j-\frac{1}{2}}(\phi_j+\phi_{j-1})}{a\cos\theta_j\,2\Delta\theta}.$$

Then from (6) and (7), we get

$$\left[\frac{(c_3\cos\theta)_{j+\frac{1}{2}}}{a^2\cos\theta_j(\Delta\theta)^2}+\frac{(c_5\cos\theta)_{j+\frac{1}{2}}}{a\cos\theta_j 2\Delta\theta}\right]\phi^r_{j+1}-\left[\frac{k^2(c_1)_j}{a^2\cos^2\theta_j}+\frac{1}{a^2\cos\theta_j(\Delta\theta)^2}\left\{(c_3\cos\theta)_{j+\frac{1}{2}}+(c_3\cos\theta)_{j-\frac{1}{2}}\right\}\right.$$
$$\left.-\frac{1}{a\cos\theta_j 2\Delta\theta}\left\{(c_5\cos\theta)_{j+\frac{1}{2}}-(c_5\cos\theta)_{j-\frac{1}{2}}\right\}-(c_6)_j\right]\phi^r_j+\left[\frac{(c_3\cos\theta)_{j-\frac{1}{2}}}{a^2\cos\theta_j(\Delta\theta)^2}-\frac{(c_5\cos\theta)_{j-\frac{1}{2}}}{a\cos\theta_j 2\Delta\theta}\right]\phi^r_{j-1} \quad (9)$$
$$-\frac{k(c_2)_j}{a^2\cos\theta_j 2\Delta\theta}\phi^i_{j+1}-\frac{k(c_4)_j}{a\cos\theta_j}\phi^i_j+\frac{k(c_2)_j}{a^2\cos\theta_j 2\Delta\theta}\phi^i_{j-1}=F^r_j$$

and

$$\left[\frac{(c_3\cos\theta)_{j+\frac{1}{2}}}{a^2\cos\theta_j(\Delta\theta)^2}+\frac{(c_5\cos\theta)_{j+\frac{1}{2}}}{a\cos\theta_j 2\Delta\theta}\right]\phi^i_{j+1}-\left[\frac{k^2(c_1)_j}{a^2\cos^2\theta_j}+\frac{1}{a^2\cos\theta_j(\Delta\theta)^2}\left\{(c_3\cos\theta)_{j+\frac{1}{2}}+(c_3\cos\theta)_{j-\frac{1}{2}}\right\}\right.$$
$$\left.-\frac{1}{a\cos\theta_j 2\Delta\theta}\left\{(c_5\cos\theta)_{j+\frac{1}{2}}-(c_5\cos\theta)_{j-\frac{1}{2}}\right\}-(c_6)_j\right]\phi^i_j+\left[\frac{(c_3\cos\theta)_{j-\frac{1}{2}}}{a^2\cos\theta_j(\Delta\theta)^2}-\frac{(c_5\cos\theta)_{j-\frac{1}{2}}}{a\cos\theta_j 2\Delta\theta}\right]\phi^i_{j-1} \quad (10)$$
$$+\frac{k(c_2)_j}{a^2\cos\theta_j 2\Delta\theta}\phi^r_{j+1}+\frac{k(c_4)_j}{a\cos\theta_j}\phi^r_j-\frac{k(c_2)_j}{a^2\cos\theta_j 2\Delta\theta}\phi^r_{j-1}=F^i_j,$$

where $1<j<J$. As shown by Barros (1991), the same discretization can be obtained through the finite volume discretization approach.

The centered differencing cannot be applied at the polar singularities. Therefore, at the poles we follow the integral approach used by Barros (1991) and BSHB. Since the poles are singular points, only the longitudinally symmetric component ($k=0$) is non zero there. Thus for the longitudinally asymmetric components ($k>0$), the boundary conditions at the poles simply become,

$$\phi^r_j=0 \text{ and } \phi^i_j=0 \quad (11)$$

To obtain the boundary conditions for the longitudinally symmetric part (which has only real part), we integrate (1) over the polar caps lying within $\Delta\theta/2$ from the poles. Then the terms with λ derivatives vanish due to cyclic continuity in the longitudinal direction. At the north pole, we then obtain,

$$\int_0^{2\pi}\int_{(\pi/2-\Delta\theta/2)}^{\pi/2}\frac{1}{\cos\theta}\frac{\partial}{\partial\theta}(c_3(\theta)\cos\theta\frac{\partial\phi}{\partial\theta})+\frac{a}{\cos\theta}\frac{\partial}{\partial\theta}(c_5(\theta)\cos\theta\,\phi)+a^2c_6(\theta)\phi]\cos\theta\,d\theta\,d\lambda$$

$$= \int_0^{2\pi} \int_{(\pi/2-\Delta\theta/2)}^{\pi/2} F\, a^2 \cos\theta \, d\theta \, d\lambda . \qquad (12)$$

From (12) we obtain

$$\int_0^{2\pi} [-(c_3 \cos\theta)_{J-\frac{1}{2}} (\frac{\partial \phi}{\partial \theta})_{J-\frac{1}{2}} - a(c_5 \cos\theta\, \phi)_{J-\frac{1}{2}}] \, d\lambda + \int_0^{2\pi} \int_{(\pi/2-\Delta\theta/2)}^{\pi/2} a^2 (c_6(\theta) - F) \cos\theta \, d\theta \, d\lambda = 0 \qquad (13)$$

Now the derivative of ϕ in the first term is evaluated by a centered difference and ϕ at J-1/2 is approximated by the average of its values at J and J-1. The second term in (13) is approximated by the mid-point rule applied to the spherical cap. Dividing by the approximation to the area of the spherical cap, $a^2 \cos\theta_{J-\frac{1}{2}} \Delta\theta(\pi/2)$, and denoting the longitudinal average by an overbar, we obtain

$$\bar{\phi}_J [(c_6)_J - \frac{4}{a^2(\Delta\theta)^2}(c_3)_{J-\frac{1}{2}} - \frac{2}{a\Delta\theta}(c_5)_{J-\frac{1}{2}}] + \bar{\phi}_{J-1} [\frac{4}{a^2(\Delta\theta)^2}(c_3)_{J-\frac{1}{2}} - \frac{2}{a\Delta\theta}(c_5)_{J-\frac{1}{2}}] = \bar{F}_J \qquad (14)$$

for the north pole and

$$\bar{\phi}_1 [(c_6)_1 - \frac{4}{a^2(\Delta\theta)^2}(c_3)_{1+\frac{1}{2}} + \frac{2}{a\Delta\theta}(c_5)_{1+\frac{1}{2}}] + \bar{\phi}_2 [\frac{4}{a^2(\Delta\theta)^2}(c_3)_{1+\frac{1}{2}} + \frac{2}{a\Delta\theta}(c_5)_{1+\frac{1}{2}}] = \bar{F}_1 . \qquad (15)$$

for the south pole. The above discretizations are second order accurate and maintain the conservative properties of the derivatives in the continuous form.

d) Solution of two simultaneous second order finite-difference equations

In the previous section, we reduced the problem to solving (9) and (10) subject to (11) (for $1 \le k \le I/2-1$), to solving the discrete version of (8) subject to (14) and (15) (for k=0) and to solving (9) without the imaginary part subject to (11) (for k =I/2). To solve (9) and (10) subject to (11), we follow the approach of Lindzen and Kuo (1969) and Chao (1979). For the sake of completeness, we repeat the method here as it is applied to our problem. Details of the method of

solution for k =0 and k =I/2 will not be presented here, however, the solutions can be obtained by following the procedure below (or see Lindzen and Kuo, 1969).

For simplicity, we rewrite (9) and (10) as

$$A_j^r\phi_{j-1}^r + B_j^r\phi_j^r + C_j^r\phi_{j+1}^r + A_j^i\phi_{j-1}^i + B_j^i\phi_j^i + C_j^i\phi_{j+1}^i = F_j^r \tag{16}$$

and

$$P_j^r\phi_{j-1}^r + Q_j^r\phi_j^r + R_j^r\phi_{j+1}^r + P_j^i\phi_{j-1}^i + Q_j^i\phi_j^i + R_j^i\phi_{j+1}^i = F_j^i, \tag{17}$$

for j=2, 3, ..., J-1. The coefficients in (16) and (17) are defined from (9) and (10). We seek the solutions to (16) and (17) in the form

$$\phi_j^r = \alpha_j^r\phi_{j+1}^r + \alpha_j^i\phi_{j+1}^i + \beta_j^r \tag{18}$$

and

$$\phi_j^i = \gamma_j^r\phi_{j+1}^r + \gamma_j^i\phi_{j+1}^i + \beta_j^i \tag{19}$$

where the α's, β's and γs are new variables yet to be determined. Evaluating (18) and (19) at j-1 and substituting into (16) and (17), we obtain

$$a_j^r\phi_j^r + a_j^i\phi_j^i + C_j^r\phi_{j+1}^r + C_j^i\phi_{j+1}^i = f_j^r \tag{20}$$

and

$$p_j^r\phi_j^r + p_j^i\phi_j^i + R_j^r\phi_{j+1}^r + R_j^i\phi_{j+1}^i = f_j^i \tag{21}$$

where

$$a_j^r = A_j^r\alpha_{j-1}^r + A_j^i\gamma_{j-1}^r + B_j^r, \tag{22}$$

$$a_j^i = A_j^r\alpha_{j-1}^i + A_j^i\gamma_{j-1}^i + B_j^i, \tag{23}$$

$$f_j^r = F_j^r - A_j^r\beta_{j-1}^r - A_j^i\beta_{j-1}^i \tag{24}$$

$$p_j^r = P_j^r\alpha_{j-1}^r + P_j^i\gamma_{j-1}^r + Q_j^r, \tag{25}$$

$$p_j^i = P_j^r\alpha_{j-1}^i + P_j^i\gamma_{j-1}^i + Q_j^i, \tag{26}$$

and

$$f_j^i = F_j^i - P_j^r\beta_{j-1}^r - P_j^i\beta_{j-1}^i. \tag{27}$$

Eliminating ϕ_j^i from (20) and (21) and comparing the result with (18), we obtain

$$\alpha_j^r = -(C_j^r p_j^i - R_j^r a_j^i)/(a_j^r p_j^i - p_j^r a_j^i), \tag{28}$$

$$\alpha_j^i = -(C_j^i p_j^i - R_j^i a_j^i)/(a_j^r p_j^i - p_j^r a_j^i), \tag{29}$$

$$\beta_j^r = (f_j^r p_j^i - f_j^i a_j^i)/(a_j^r p_j^i - p_j^r a_j^i). \tag{30}$$

Similarly,eliminating ϕ_j^r from (20) and (21) and comparing the result with (19), we obtain

$$\gamma_j^r = (C_j^r p_j^r - R_j^r a_j^r)/(a_j^r p_j^i - p_j^r a_j^i), \tag{31}$$

$$\gamma_j^i = (C_j^i p_j^r - R_j^i a_j^r)/(a_j^r p_j^i - p_j^r a_j^i), \tag{32}$$

and

$$\beta_j^i = -(f_j^r p_j^r - f_j^i a_j^r)/(a_j^r p_j^i - p_j^r a_j^i). \tag{33}$$

Thus, if we know $\alpha_1^r, \alpha_1^i, \beta_1^r, \beta_1^i, \gamma_1^r$, and γ_1^i (which can be obtained from the lower boundary condition), then all of the α's, β's and γ's are readily obtained for each j. In the present problem, the boundary condition (11) implies that all $\alpha_1^r = \alpha_1^i = \beta_1^r = \beta_1^i = \gamma_1^r = \gamma_1^i = 0$.

Equations (18) and (19) can then be used to obtain ϕ_j^r and ϕ_j^i at all j=J-1, J-2,...1, provided ϕ_J^r and ϕ_J^i are known, which can be obtained from the upper boundary condition. In the present problem, $\phi_J^r = \phi_J^i = 0$.

e) coding strategy

The first step of the solution procedure involves the use of a forward FFT to obtain the complex amplitudes of the forcing function F that depend only on latitude. Then for each wavenumber, the complex amplitude of the solution is obtained by solving the coupled second order difference equations. Finally we use a backward FFT to obtain the solution.

We made this procedure very efficient by using the CRAY subroutine RFFTMLT to transform (both forward and backward) all latitudes simultaneously . In addition, we wrote the

code to solve two coupled second order difference equations by vectorizing over longitudinal wavenumbers. This vectorization coupled with the vectorized FFT package makes our direct solver very efficient.

It should be pointed out that if the longitudinal derivatives are approximated by finite differences, then the definitions of k and k^2 should be appropriately changed in all of the equations above . Also the FFT has a restriction on I, namely that $I= 2^p x 3^q x 5^r$, where p, q and r are integers. However, there is no restriction on the choice of J.

The complete listing of the FORTRAN code for the solver is given in the Appendix. The calling sequence for the solver is as follows:

```
CALL ELLSOL(IMA, IM, JM, AE, INIT1, INIT2, DIFF, C1, C2, C3, C4, C5, C6, FI, FO
                                                    , WSV, WRK, IX)
```

where

IMA is the leading dimension of input and output arrays FI and FO.

IM is the longitudinal domain over which the solution is desired.

(IM should be less than or equal to IMA)

JM is the number of grid points from the south pole to the north pole.

AE is the radius of the sphere.

INIT1 is a logical variable, true only the first time for a given IM, JM and AE.

INIT2 is a logical variable, used only when INIT1 is false. It is true whenever the coefficients of the elliptic equation change (except for C6).

DIFF is a logical variable, true if wavenumber k is based on finite-difference, false otherwise.

C1, C2, C3, C4, C5, C6 are the coefficients of equation (1) and have dimension JM.

C1, C2, C4, and C6 are defined at the grid points j and C3 and C5 are defined at the midpoints between j and j+1. C6 can change any time. If any other coefficients change, then INIT2 should be true.

FI is an input array of dimension (IMA,JM) containing the forcing F in the locations I=1,IM and J=1,JM.

FO is an output array of dimension (IMA,JM) containing the solution ϕ in the locations I=1,IM and J=1,JM.

WSV is an array whose dimension is at least [4*IM+9*JM+7]. This array stores some constants to be used in subsequent calls to ELLSOL. It should not be overwritten unless the next call to ELLSOL has INIT1 true.

WRK is a work array whose dimension is at least [14 * IM * JM]

IX is an integer array of dimension IM needed for the FFT (it should not be overwritten)

III. Evaluation of the solver

In this section, we present results from some tests to evaluate the solver. Since it is difficult to find analytical solutions to (1) against which we can compare the numerical solutions, we adopt the following "prescribed solution" procedure. Under this procedure, we assume a solution *apriori* and apply the differential operator on the left hand side of (1) to obtain the forcing F. Then we obtain the numerical solution for this forcing and compare the results with the original assumed solution. Here we consider the following simple function for ϕ:

$$\phi_{i,j} = 5.0\text{x}10^4 + 1.0\text{x}10^3 \cos\theta_j \cos(2\pi i / I) \tag{34}$$

where i and j are longitudinal and latitudinal indices, respectively. A contour map of this function is shown in Fig. 1. The forcing $F_{i,j}$ is computed using a second order accurate finite-difference operator corresponding to the left hand side of (1). In the following, we consider two cases. In case 1 we apply the solver to the elliptic equation of BSHB and compare the results with those obtained from the multigrid solver. In case 2 we apply it to the Poisson equation and compare the results with those obtained from FISHPAK.

case 1

The elliptic equation of BSHB is obtained by setting

$$c_1(\theta) = G, \qquad c_2(\theta) = 0.0, \qquad c_3(\theta) = G$$

$$c_4(\theta) = -\frac{\partial(GF)}{a\partial\theta}, \qquad c_5(\theta) = 0.0, \qquad c_6(\theta) = -[(\Delta t/2)^2 \bar{\phi}]^{-1}$$

where

$$G = [1 + F^2]^{-1}, \qquad \text{and} \qquad F = \Delta t \Omega \sin\theta.$$

Here $\bar{\phi}$=50000 is a mean value, Δt=3600 s, and Ω is the earth's rotation rate. The 2D multigrid solver of BSHB is used in the Full-Multigrid mode. We use a single V-cycle with one relaxation sweep both before and after the coarse grid correction and eight relaxation steps on the coarsest grid. These are identical to those used in BSHB and more details are available in that paper.

We solved the above problem for various resolutions ranging from (I,J) = (48,25) to (I,J)=(768,385). Figures 2a and 2b show the numerical solution for the direct method and the multigrid method, respectively with (I,J)=(96,49). The solutions appear to be almost identical to that shown in Fig. 1. However, the accuracy of the solution is revealed in Figs. 3a and 3b which show the difference between the exact and the numerical solutions for both solvers. Notice that in Fig. 3a the error is multiplied by a factor of 10^8 while in Fig. 3b, it is multiplied by a factor of 10. Thus in this case the direct solver is several orders of magnitude more accurate. Similar results were also found at other resolutions (not shown). Furthermore, the accuracy of the multigrid solver did not improve significantly when both the number of V-cycles and the number of relaxation sweeps were increased. However, we do recognize that the level of accuracy of the spatial discretization should be the level of accuracy desired for any problem. Nevertheless, an efficient direct solver is always preferable since its solution is close to machine accuracy.

We next examine the efficiency of the solvers. For this purpose, we present in Fig. 4 the CPU time (t) taken for 500 calls to the solver on a single processor of the CRAY YMP as a

function of the total number (N) of grid points (i.e. IxJ). In Fig. 4a the axes are linear while in Fig. 4b they are logarithmic. The timings are also presented in Table 1. Clearly the direct solver is faster at all resolutions examined. Also, note that for both solvers, the time (t) is almost a linear function of the total number of grid points (N).

Table 1: CPU time (t) in seconds for 500 calls to the direct solver, the multigrid solver and the solver from FISHPAK on a single processor of the CRAY YMP as a function of resolution.

(I,J)	direct solver	multigrid solver	FISHPAK
(48,25)	0.513	2.323	3.14
(72,46)	1.110	-	-
(96,49)	1.401	7.008	12.615
(144,91)	4.062	-	-
(192,97)	5.253	23.301	53.914
(288,181)	14.464	-	-
(384,193)	19.563	85.588	241.579
(576,361)	55.889	-	-
(768,385)	77.998	326.955	-

case 2

Now we consider the Poisson equation which is obtained by setting $c_1(\theta) = c_3(\theta) = 1$ and $c_2(\theta) = c_4(\theta) = c_5(\theta) = c_6(\theta) = 0$. It should be pointed out that when $c_6(\theta) = 0$ we cannot determine the constant part and thus there is no unique solution. In addition, if the forcing has a

global mean component, then there is no solution to the problem. Therefore, in our solver we remove the global mean from the forcing F whenever $c_6(\theta) = 0$. This is equivalent to the perturbation method of Swarztrauber (1974b).

We used the forcing as in case 1 to compare the numerical solution with the assumed one. The errors in the numerical solution (not shown) are of the order 10^{-6} or less. Similar results are found for other resolutions.

For this case, we compare the efficiency of our direct solver with the direct solver from the FISHPAK package (Adams et al., 1980). As before, we show the CPU time (t) taken for 500 calls to both solvers as a function of the total number (N) of grid points (Fig. 5). Again, our direct solver is significantly faster at all resolutions and there is a noticeable divergence between the two curves as the resolution increases.

We also confirmed that our solver works equally well for a general case in which all of the coefficients in (1) are nonzero functions of θ. Finally, we repeated each case above by assuming a prescribed solution made up of random numbers. We found for all cases that the errors were comparable to those discussed above.

IV. Summary

An efficient, direct solver for the discrete solution of a class of two-dimensional separable elliptic equations on the sphere has been discussed. It is based on a Fourier decomposition in longitude and a direct solution of the resulting coupled second order finite-difference equations in latitude. These equations are solved following the approach of Lindzen and Kuo (1969) and Chao (1979).

For the elliptic equation of BSHB we find that the direct solver is both more efficient and accurate than the multigrid solver at all resolutions. For the special case of Poisson's equation we find, at all resolutions, that the direct solver presented here is more efficient than that available in FISHPAK (Adams et al.,1980).

Thus our solver is both accurate and efficient and general enough that it could be used for any separable elliptic equation on the sphere with coefficients independent of longitude. It can also be applied to limited areas on the sphere if cyclic boundary conditions are invoked in longitude and if appropriate boundary conditions are used in latitude. It is currently being used in the global multi-level primitive equation semi-Lagrangian semi-implicit model of Bates et al. (1992) at the Goddard Laboratory for Atmospheres and in the adjoint model development (Li et al., 1991) at the Florida State University. While the solver is accurate and flexible, it takes only a few percent of the CPU time taken by the multi-level model dynamics.

V. Acknowledgments

We thank Dr. J. R. Bates for valuable guidance and encouragement during the course of this work. We also thank Professors I. M. Navon and F. H. M. Semazzi for their useful suggestions. This work was supported by the Climate and Hydrologic Systems Modeling and Data Analysis Program of NASA headquarters under Grant 578-41-16-20.

VI. References

Adams, J., P.N. Swarztrauber, and R. Sweet, 1980 : "FISHPAK: A package of FORTRAN subprograms for the solution of separable elliptic partial differential equations", (Version 3.1, October 1980), National Center for Atmospheric Research, Boulder, CO 80307.

Barros, S. R. M. 1991: Multigrid methods for two- and three-dimensional Poisson-type equations on the sphere. *J. Comp. Phys.*, **92**, 313-348.

Barros, S. R. M., D. P. Dee and F. Dickstein, 1989: A multi-grid solver for semi-implicit global shallow water models. *Atmos Ocean*, **28**, 24-47.

Bates, J. R., F. H. M. Semazzi, R. W. Higgins and S. R. M. Barros, 1990: Integration of the shallow water equations on the sphere using a vector semi-Lagrangian scheme with a multi-grid solver. *Mon. Wea. Rev.* **118**, 1615-1627.

Bates, J. R., S. Moorthi and R. W. Higgins, 1992: A global multilevel atmospheric model using a vector semi-Lagrangian finite difference scheme. Part I: Adiabatic Formulation. *Mon. Wea. Rev.* (in press).

Chao, W. C., 1979: "A study of conditional instability of second kind and a numerical simulation of intertropical convergence zone and easterly waves" *Ph. D. Dissertation*, University of California, Los Angeles, 253pp.

Cote, J., A. Staniforth, 1990: An accurate and efficient finite-element global model of the shallow-water primitive equations. *Mon. Wea. Rev*, **118**, 2707-2717.

Fulton, S. R., P. E. Ciesielski and W. H. Schubert, 1986: Multi-grid methods for elliptic problems: A review. *Mon. Wea.Rev.* **115**, 943-959.

Hockney, R. W., 1965: A fast direct solution of Poisson's equation using Fourier analysis. *J. Assoc. Comput. Mach.*, **8**, 95-113.

Le Bail, R. C., 1972: Use of fast Fourier transforms for solving partial differential equations in physics. *J. Comp. Phys.* **9**, 440-465.

Li, Y., I. M. Navon, S. Moorthi, and R. W. Higgins, 1991: The 2-D semi-implicit semi-Lagrangian global model: direct solver, vectorization and adjoint model development. Report # FSU-SCRI-91-158, The Florida State University, Florida, 56pp

Lindzen, R. S., and H. L. Kuo, 1969: A reliable method for the numerical integration of a large class of ordinary and partial differential equations. *Mon. Wea. Rev.*, **97**, 732-734.

McDonald, A., and J. R. Bates, 1989: Semi-Lagrangian integration of a gridpoint shallow water model on the sphere. *Mon. Wea.Rev*, **117**, 130-137.

Phillips, T. N., 1984: Numerical solution of a coupled pair of elliptic equations from solid state electronics. *J. Comp. Phys.*, **53**, 472-483.

Robert, A., 1969: The integration of a spectral model of the atmosphere by the implicit method. *Proc. WMO/IUGG Symposium on Numerical Weather Prediction*, Tokyo (Nov-Dec., 1968), [Japan Meteorological Agency, Tokyo], p VII-19 - VII-24.

Staniforth, A., and J. Cote, 1991: Semi-Lagrangian integration schemes for atmospheric models - a review. *Mon. Wea. Rev.*, **119**, 2206-2223.

Sweet, R., 1973: Direct methods for the solution of Poisson's equation on a staggered grid. *J. Comp. Phys.* **12**, 422-428.

Sweet, R., 1974: A generalized cyclic reduction algorithm. *SIAM J. Numer. Anal.*. **11**, 506-520.

Swarztrauber, P. N., 1974a: A direct method for the discrete solution of separable elliptic equations, *SIAM J. Numer. Anal.*. **11**, 1136-1150.

Swarztrauber, P. N., 1974b: The direct solution of the discrete Poisson equation on the surface of a sphere. *J. Comp. Phys.* **15**, 46-54.

Swarztrauber, P. N. and R. Sweet, 1973: The direct solution of the discrete Poisson equation on a disk. *SIAM J. Numer. Anal.*. **10**, 900-907.

Swarztrauber, P. N. and R. Sweet, 1975: Efficient FORTRAN subprograms for the solution of elliptic partial differential equations, NCAR Technical Note, NCAR-TN/IA-109, 139pp.

Tanguay, M., A. Simrad and A. Staniforth, 1989: A three dimensional semi-Lagrangian scheme for Canadian regional finite-element forecast model. *Mon. Wea. Rev.* **117**, 1861-1871.

Temperton, C. and A. Staniforth, 1987: An efficient two-time-level semi-Lagrangian semi-implicit integration scheme. *Q. J. R. M. S.* **113**,1025-1039.

VII. Appendix

```
      SUBROUTINE ELLSOL(IMA, IM, JM, AE, INIT1, INIT2, DIFF
     *,                  C1, C2, C3, C4, C5, C6, FI, FO, WSV, WRK, IX)
C
C*************************************************************************
C*                                                                       *
C*    (IMA, JM) is the dimension of the forcing FI (input) and the       *
C*    solution FO (output).  IM is less than or equal to IMA and         *
C*    represents the physical domain (in longitude) over which the       *
C*    solution is computed. AE is the radius of the sphere.              *
C*                                                                       *
C*    INIT1, INIT2 and DIFF are logical variables.                       *
C*    When INIT1 is true, the value passed for INIT2 is ignored          *
C*                                                                       *
C*    C1, C2,...C6 are arrays of dimension JM containing the coefficients*
C*    of the elliptic equation.                                          *
C*                                                                       *
C*    WSV, WRK and IX are other miscellaneous work arrays.               *
C*                                                                       *
C*    The dimension of WSV should be at least [4*IM + 9*JM + 7].         *
C*    This array stores constants needed in subsequent calls to ELLSOL.  *
c*    It should not be overwritten unless INIT1 is true for the          *
C*    next call to ELLSOL.                                               *
C*                                                                       *
C*    The dimension of WRK should be at least [14 * IM*JM].              *
C*                                                                       *
C*    The dimension of IX should be at least IM and it should not be     *
C*    overwritten unless INIT1 is true for the next call to ELLSOL.      *
C*                                                                       *
C*************************************************************************
C
      DIMENSION    FI(IMA,JM), FO(IMA,JM)
     *,            C1(JM),     C2(JM), C3(JM), C4(JM), C5(JM), C6(JM)
     *,            WSV(1),     WRK(1), IX(IM)
C
      LOGICAL   INIT1, INIT2, DIFF
C
      ITRG = 3*(IM+2) + 1
      IMB2 = IM / 2
      IJM  = (IMB2-1) * JM
      IJMM = IJM*22 + JM
      IJM2 = IJMM + (IM+2)*(JM-2)
C
      CALL DIRSOL(IMA, IM, JM, IMB2, ITRG, AE, INIT1, INIT2, DIFF
     *,           C1, C2, C3, C4, C5, C6, FI, FO, IX
C
     *,           WSV(1),       WSV(JM+1),    WSV(JM*2+1), WSV(JM*3+1)
     *,           WSV(JM*4+1), WSV(JM*5+1), WSV(JM*6+1), WSV(JM*7+1)
     *,           WSV(JM*9+1), WSV(JM*9+IMB2+1), WSV(JM*9+IM+1)
C
     *,     WRK(1),        WRK(IJM+1),    WRK(IJM*2+1),  WRK(IJM*3+1)
     *,     WRK(IJM*4+1), WRK(IJM*5+1),  WRK(IJM*6+1),  WRK(IJM*7+1)
     *,     WRK(IJM*8+1), WRK(IJM*9+1),  WRK(IJM*10+1), WRK(IJM*11+1)
     *,     WRK(IJM*12+1),WRK(IJM*13+1), WRK(IJM*14+1), WRK(IJM*15+1)
     *,     WRK(IJM*16+1),WRK(IJM*17+1), WRK(IJM*18+1), WRK(IJM*19+1)
     *,     WRK(IJM*20+1),WRK(IJM*21+1), WRK(IJM*22+1), WRK(IJMM+1)
     *,     WRK(IJM2+1),  WRK(1),        WRK(JM+1),     WRK(JM*2+1)
     *,     WRK(JM*3+1),  WRK(JM*4+1))
C
      RETURN
      END
```

```
      SUBROUTINE DIRSOL(IMA, IM, JM, IMB2, ITRG, AE, INIT1, INIT2, DIFF
     *,                 C1, C2, C3, C4, C5, C6, FI, FO, IX
C
     *,                 CDS1, CDN1, CDS2, CDN2, CDC1, CDC2, CDC3, CPH
     *,                 WV,   WVSQ, TR

     *,                 AN1, BN1, CN1, DN1, FN1, PN1, QN1, RN1
     *,                 AN2, BN2, CN2, DN2, FN2, PN2, QN2, RN2
     *,                 AL1, AL2, GM1, GM2, BT1, BT2
     *,                 FM, A, B, AN,  BN,  CN,  DN, FN)
C
C     ITRG SHOULD AT LEAST BE 3*(IM+2) + 1
C
      INTEGER   FORWARD, BACKWARD
C
      PARAMETER (FORWARD=-1,BACKWARD=1)
      PARAMETER(PI=3.1415926535898, TWOPI=PI+PI, PIO2=0.5*PI)
C
      DIMENSION FI(IMA,JM), FO(IMA,JM)
     *,         C1(JM),     C2(JM), C3(JM), C4(JM), C5(JM), C6(JM)
     *,         TR(ITRG),   IX(IM), WVSQ(IMB2), WV(IMB2)
C
      DIMENSION AN1(IMB2-1,JM), BN1(IMB2-1,JM), CN1(IMB2-1,JM)
     *,         AN2(IMB2-1,JM), BN2(IMB2-1,JM), CN2(IMB2-1,JM)
     *,         QN1(IMB2-1,JM), PN1(IMB2-1,JM), RN1(IMB2-1,JM)
     *,         QN2(IMB2-1,JM), PN2(IMB2-1,JM), RN2(IMB2-1,JM)
     *,         DN1(IMB2-1,JM), DN2(IMB2-1,JM)
     *,         FN1(IMB2-1,JM), FN2(IMB2-1,JM)
C
     *,         AL1(IMB2-1,JM), AL2(IMB2-1,JM)
     *,         GM1(IMB2-1,JM), GM2(IMB2-1,JM)
     *,         BT1(IMB2-1,JM), BT2(IMB2-1,JM)
C
      DIMENSION AN(JM),   BN(JM), CN(JM), DN(JM), FN(JM)
     *,         CDN1(JM), CDS1(JM), CDN2(JM), CDS2(JM)
     *,         CDC1(JM), CDC2(JM), CDC3(JM), FM(JM),   CPH(JM,2)
C
      DIMENSION A(IM+2,JM-2), B(IM*2,JM-2)
C
      LOGICAL   INIT1, INIT2, DIFF, FIRST, SETC0
C
      DATA JMM1, JMM2, IMP1, IMP2, LEN/5*0/, AREA/0.0/
      DATA FIM, DLM, DPH, RDLM, RDPH, DY, AESQ, CDNP, CDSP/9*0.0/
      DATA FIRST/.FALSE./
C
      SAVE
C
C   INITIALIZATION ON FIRST CALL
C
      IF (INIT1) THEN
         CALL FFTFAX(IM, IX, TR)
C
         JMM1 = JM - 1
         JMM2 = JM - 2
         IMP1 = IM + 1
         IMP2 = IM + 2
         LEN  = IMB2 - 1
         FIM  = 1.0 / FLOAT(IM)
         DLM  = TWOPI * FIM
         DPH  = PI / FLOAT(JMM1)
C
         RDLM  = 1.0 / DLM
         RDPH  = 1.0 / DPH
```

```
C
      DY    = AE * DPH
      AESQ  = AE * AE
C
      DO 10 J=1,JM
      TEM       = -PIO2  + (J-1)*DPH
      CPH(J,1) = COS(TEM + 0.5*DPH)
      CPH(J,2) = COS(TEM)
10    CONTINUE
C
      CPH(1,2)  = 0.0
      CPH(JM,2) = 0.0
      FIRST     = .TRUE.
      INIT2     = .TRUE.
    ENDIF
C
    IF (INIT2) THEN
      DO 20 J=2,JMM1
      TEM1    = 1.0  / (CPH(J,2) * 2.0 * DY)
      TEM2    = 1.0  / (CPH(J,2) * DY*DY)
C
      CDN1(J) = C5(J)    * CPH(J,1)    * TEM1
      CDS1(J) = C5(J-1)  * CPH(J-1,1)  * TEM1
      CDN2(J) = C3(J)    * CPH(J,1)    * TEM2
      CDS2(J) = C3(J-1)  * CPH(J-1,1)  * TEM2
C
      CDC1(J) = C1(J) / (CPH(J,2) * CPH(J,2) * AESQ)
      CDC2(J) = C4(J) / (AE * CPH(J,2))
      CDC3(J) = C2(J) / (AE * CPH(J,2) * 2.0 * DY)
C
20    CONTINUE
      CDC1(1)  = 4.0 * C3(1)    / (DY*DY)
      CDC1(JM) = 4.0 * C3(JMM1) / (DY*DY)
      CDNP     = C5(JMM1) * 2.0 / DY
      CDSP     = C5(1   ) * 2.0 / DY
C
      IF (DIFF) THEN
        DO 30 I=1,IMB2
        WVSQ(I) = (SIN(0.5*I*DLM) * (RDLM * 2.0)) ** 2
        WV  (I) = SIN(I*DLM) * RDLM
30      CONTINUE
      ELSE
        DO 40 I=1,IMB2
        WVSQ(I) = I * I
        WV  (I) = I
40      CONTINUE
      ENDIF
    ENDIF
C
    DO 25 J=1,JM
    IF (C6(J) .NE. 0.0) GO TO 26
25  CONTINUE
    SETC0 = .TRUE.
    CALL REMGLM(IMA, IM, JM, FI, CPH, FIRST, AREA)
    GO TO 27
26  SETC0 = .FALSE.
27  CONTINUE
C
    DO 42 J=2,JMM1
    FM(J) = 0.0
    DO 42 I=1,IM
    FM(J) = FM(J) + FI(I,J)
42  CONTINUE
```

```
      DO 43 J=2,JMM1
      FM(J) = FM(J) * FIM
   43 CONTINUE
      DO 45 J=2,JMM1
      DO 45 I=1,IM
      A(I,J-1) = FI(I,J) - FM(J)
C     A(I,J-1) = FI(I,J)
   45 CONTINUE
C
      DO 50 J=1,JMM2
      A(IM+1,J) = A(1,J)
      A(IM+2,J) = A(2,J)
   50 CONTINUE
C
      CALL RFFTMLT(A,B,TR,IX,1,IMP2,IM,JMM2,FORWARD)
C
      DO 60 J=2,JMM1
      DO 60 I=1,LEN
      AN1(I,J) =   CDS2(J) - CDS1(J)
      BN1(I,J) = -(CDS2(J) + CDN2(J) + WVSQ(I) * CDC1(J)
     *           + CDS1(J) - CDN1(J) - C6(J))
      CN1(I,J) =   CDN2(J) + CDN1(J)
      AN2(I,J) =   CDC3(J) * WV(I)
      BN2(I,J) = - CDC2(J) * WV(I)
      CN2(I,J) = - CDC3(J) * WV(I)
C
      DN1(I,J) =   A(I+I+1,J-1)
C
      QN1(I,J) =   CN2(I,J)
      PN1(I,J) = - BN2(I,J)
      RN1(I,J) =   AN2(I,J)
      QN2(I,J) =   AN1(I,J)
      PN2(I,J) =   BN1(I,J)
      RN2(I,J) =   CN1(I,J)
C
      DN2(I,J) = A(I+I+2,J-1)
   60 CONTINUE
C
      CALL SO2ODE(LEN, JM,  AN1, BN1, CN1, DN1, AN2, BN2, CN2, DN2
     *,                  QN1, PN1, RN1, QN2, PN2, RN2, FN1, FN2
     *,                  AL1, AL2, GM1, GM2, BT1, BT2)
C
      DO 70 J=2,JMM1
      DO 70 I=1,LEN
      A(I+I+1,J-1) = FN1(I,J)
      A(I+I+2,J-1) = FN2(I,J)
   70 CONTINUE
C
      DO 80 J=2,JMM1
      AN(J)  = CDS2(J) - CDS1(J)
      BN(J)  = -(CDS2(J) + CDN2(J) + CDS1(J) - CDN1(J)
     *                             + WVSQ(IMB2) * CDC1(J) - C6(J))
      CN(J)  = CDN2(J) + CDN1(J)
      DN(J)  = A(IMP1,J-1)
   80 CONTINUE
C
      CALL SO1ODE(JM,  AN, BN, CN, DN, FN, AL1, BT1)
C
      DO 90 J=2,JMM1
      A(IMP1,J-1) = FN(J)
C     BN(J)       = -(AN(J) + CN(J) - C6(J))
      BN(J)  = -(CDS2(J) + CDN2(J) + CDS1(J) - CDN1(J) - C6(J))
C###  DN(J)       = A(1,J-1)
      DN(J)       = FM(J)
   90 CONTINUE
```

```
      BN(1)  = - (-C6(1)  + CDC1(1)  - CDSP)
      BN(JM) = - (-C6(JM) + CDC1(JM) + CDNP)
      AN(JM) = CDC1(JM)   - CDNP
      CN(1)  = CDC1(1)    + CDSP
      DN(1)  = FI(1,1)
      DN(JM) = FI(1,JM)
C
      CALL SO1OD2(JM,  AN, BN, CN, DN, FN, AL1, BT1)
C
      IF (SETC0) THEN
         FLDM = (FN(1) + FN(JM)) * (AREA*CPH(1,1)*0.25)
         DO 92 J=2,JMM1
         FLDM = FLDM + FN(J) * (CPH(J,2)*AREA)
   92    CONTINUE
         DO 94 J=1,JM
         FN(J) = FN(J) - FLDM
   94    CONTINUE
      ENDIF
C
      DO 100 J=2,JMM1
C     A(1,   J-1) = FN(J)
      A(2,   J-1) = 0.0
      A(IMP2,J-1) = 0.0
  100 CONTINUE
C
      CALL RFFTMLT(A,B,TR,IX,1,IMP2,IM,JMM2,BACKWARD)
C
      DO 120 J=1,JMM2
      DO 110 I=1,IM
C     FO(I,J+1) = A(I,J)
      FO(I,J+1) = A(I,J) + FN(J+1)
  110 CONTINUE
  120 CONTINUE
C
      DO 130 I=1,IM
      FO(I,1)  = FN(1)
      FO(I,JM) = FN(JM)
  130 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE SO2ODE(LEN, JM,  AN1, BN1, CN1, DN1, AN2, BN2, CN2, DN2
     *,                 QN1, PN1, RN1, QN2, PN2, RN2, FN1, FN2
     *,                 AL1, AL2, GM1, GM2, BT1, BT2)
C
      DIMENSION AN1(LEN,JM), BN1(LEN,JM), CN1(LEN,JM)
     *,         AN2(LEN,JM), BN2(LEN,JM), CN2(LEN,JM)
     *,         QN1(LEN,JM), PN1(LEN,JM), RN1(LEN,JM)
     *,         QN2(LEN,JM), PN2(LEN,JM), RN2(LEN,JM)
     *,         DN1(LEN,JM), DN2(LEN,JM), FN1(LEN,JM), FN2(LEN,JM)
C
      DIMENSION AL1(LEN,JM), AL2(LEN,JM), GM1(LEN,JM), GM2(LEN,JM)
     *,         BT1(LEN,JM), BT2(LEN,JM)
C
      JMM1 = JM - 1
C
      DO 10 I=1,LEN
      AL1(I,1) = 0.0
      AL2(I,1) = 0.0
      BT1(I,1) = 0.0
      BT2(I,1) = 0.0
      GM1(I,1) = 0.0
      GM2(I,1) = 0.0
   10 CONTINUE
C
```

```
      DO 30 J=2,JMM1
      JM1 = J - 1
C
      DO 20 I=1,LEN
      SA1 = AN1(I,J) * AL1(I,JM1) + AN2(I,J) * GM1(I,JM1) + BN1(I,J)
      SA2 = AN1(I,J) * AL2(I,JM1) + AN2(I,J) * GM2(I,JM1) + BN2(I,J)
C
      SB1 = QN1(I,J) * AL1(I,JM1) + QN2(I,J) * GM1(I,JM1) + PN1(I,J)
      SB2 = QN1(I,J) * AL2(I,JM1) + QN2(I,J) * GM2(I,JM1) + PN2(I,J)
C
      SD1 = DN1(I,J) - AN1(I,J) * BT1(I,JM1) - AN2(I,J) * BT2(I,JM1)
      SD2 = DN2(I,J) - QN1(I,J) * BT1(I,JM1) - QN2(I,J) * BT2(I,JM1)
C
      RM  = 1.0 / (SA1*SB2 - SB1*SA2)
C
      AL1(I,J) = - RM * (CN1(I,J) * SB2 - RN1(I,J) * SA2)
      AL2(I,J) = - RM * (CN2(I,J) * SB2 - RN2(I,J) * SA2)
      BT1(I,J) =   RM * (SD1      * SB2 - SD2      * SA2)
C
      GM1(I,J) =   RM * (CN1(I,J) * SB1 - RN1(I,J) * SA1)
      GM2(I,J) =   RM * (CN2(I,J) * SB1 - RN2(I,J) * SA1)
      BT2(I,J) = - RM * (SD1      * SB1 - SD2      * SA1)
C
   20 CONTINUE
   30 CONTINUE
C
      DO 40 I=1,LEN
      FN1(I,JM) = 0.0
      FN2(I,JM) = 0.0
   40 CONTINUE
C
      DO 60 J=JMM1,1,-1
      DO 50 I=1,LEN
      FN1(I,J) = AL1(I,J)*FN1(I,J+1) + AL2(I,J)*FN2(I,J+1) + BT1(I,J)
      FN2(I,J) = GM1(I,J)*FN1(I,J+1) + GM2(I,J)*FN2(I,J+1) + BT2(I,J)
   50 CONTINUE
   60 CONTINUE
C
      RETURN
      END

      SUBROUTINE SO1ODE(JM,  AN, BN, CN, DN, FN, AL, BT)
C
      DIMENSION AN(JM), BN(JM), CN(JM), DN(JM), FN(JM)
C
      DIMENSION AL(JM), BT(JM)
C
      JMM1 = JM - 1
C
      AL(1)  = 0.0
      BT(1)  = 0.0
      FN(JM) = 0.0
C
      DO 10 J=2,JMM1
      JM1   = J - 1
      SA1   = 1.0 / (AN(J) * AL(JM1) + BN(J))
      AL(J) = - CN(J) * SA1
      BT(J) = (DN(J) - AN(J) * BT(JM1)) * SA1
   10 CONTINUE
C
      DO 20 J=JMM1,1,-1
      FN(J) = AL(J)*FN(J+1) + BT(J)
   20 CONTINUE
C
      RETURN
      END
C
```

```
      SUBROUTINE SO1OD2(JM,  AN, BN, CN, DN, FN, AL, BT)
C
      DIMENSION AN(JM), BN(JM), CN(JM), DN(JM), FN(JM)
C
      DIMENSION AL(JM), BT(JM)
C
      JMM1 = JM - 1
C
      TEM = 1.0 / BN(1)
      AL(1) = - CN(1) * TEM
      BT(1) =   DN(1) * TEM
C
      DO 10 J=2,JMM1
      JM1    = J - 1
      SA1    = 1.0 / (AN(J) * AL(JM1) + BN(J))
      AL(J)  = - CN(J) * SA1
      BT(J)  = (DN(J) - AN(J) * BT(JM1)) * SA1
   10 CONTINUE
C
      FN(JM) = (DN(JM) - AN(JM)*BT(JMM1)) / (BN(JM) + AN(JM)*AL(JMM1))
C
      DO 20 J=JMM1,1,-1
      FN(J) = AL(J)*FN(J+1) + BT(J)
   20 CONTINUE
C
      RETURN
      END

      SUBROUTINE REMGLM(IMA, IM, JM, FLD, CPH, FIRST, AREA)
C
      DIMENSION FLD(IMA,JM), CPH(JM,2)
      LOGICAL FIRST
      DATA JMM1/0/, FIM/0.0/
      SAVE
C
      IF (FIRST) THEN
         JMM1   = JM - 1
         FIM    = 1.0 / FLOAT(IM)
         AREA   = 0.0
         DO 10 J=2,JMM1
         AREA   = AREA + CPH(J,2)
   10    CONTINUE
         AREA   = 1.0 / (AREA + CPH(1,1) * 0.5)
         FIRST  = .FALSE.
      ENDIF
C
      FLDM = (FLD(1,1) + FLD(1,JM)) * (AREA*CPH(1,1)*0.25)
      DO 20 J=2,JMM1
      DO 20 I=1,IM
      FLDM = FLDM + FLD(I,J) * (CPH(J,2)*AREA*FIM)
   20 CONTINUE
C
      DO 30 I=1,IMA*JM
      FLD(I,1) = FLD(I,1) - FLDM
   30 CONTINUE
C
      RETURN
      END
```

VIII. List of Figures

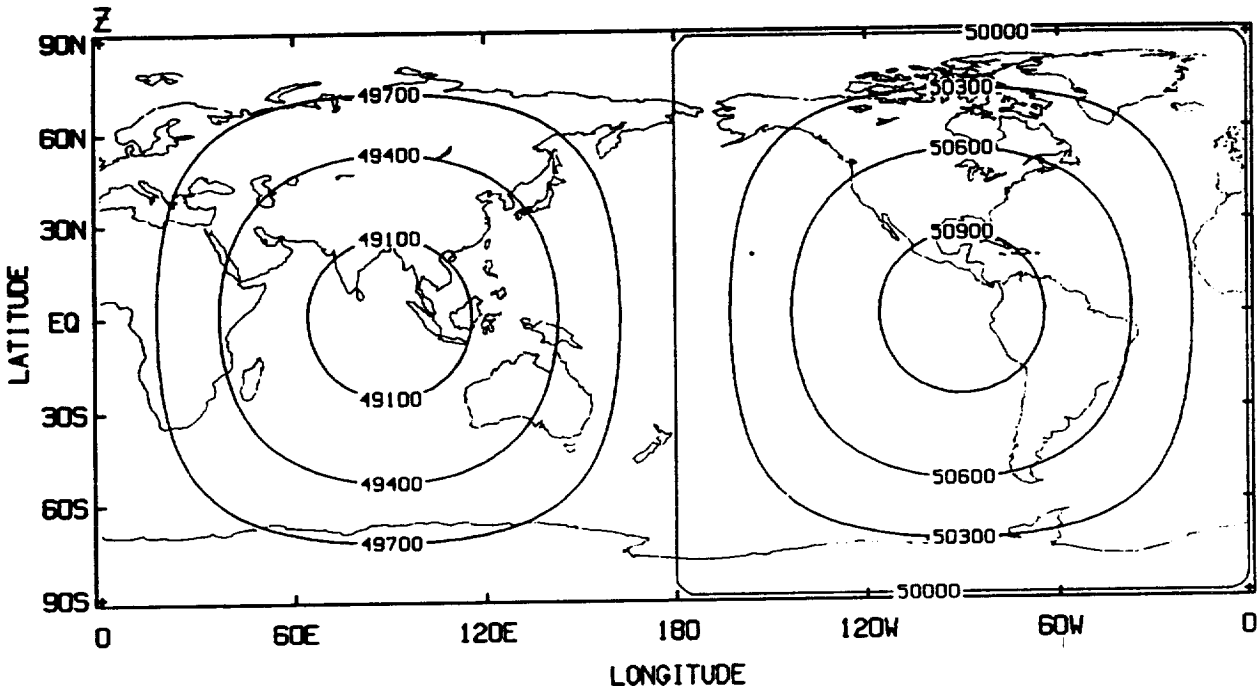
Z
90N
60N
30N
EQ
30S
60S
90S
LATITUDE
0
60E
120E
180
120W
60W
0
LONGITUDE
49700
49400
49100
49100
49400
49700
50000
50300
50600
50900
50600
50300
50000

Fig. 1

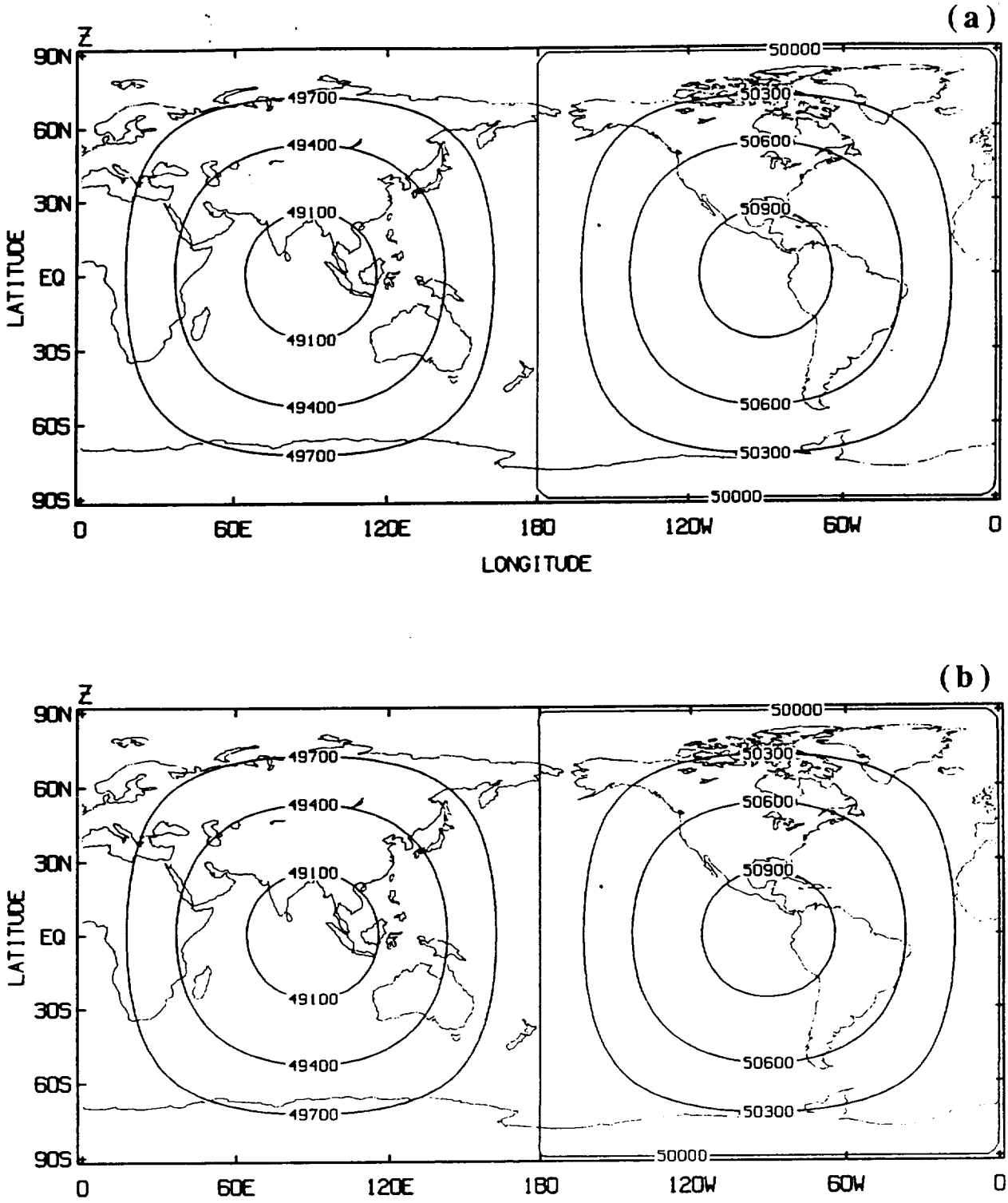
(a)
Z
90N
60N
30N
EQ
30S
60S
90S
LATITUDE
49700
49400
49100
49100
49400
49700
50000
50300
50600
50900
50600
50300
50000
0
60E
120E
180
120W
60W
0
LONGITUDE
(b)
Z
90N
60N
30N
EQ
30S
60S
90S
LATITUDE
49700
49400
49100
49100
49400
49700
50000
50300
50600
50900
50600
50300
50000
0
60E
120E
180
120W
60W
0

Fig. 2

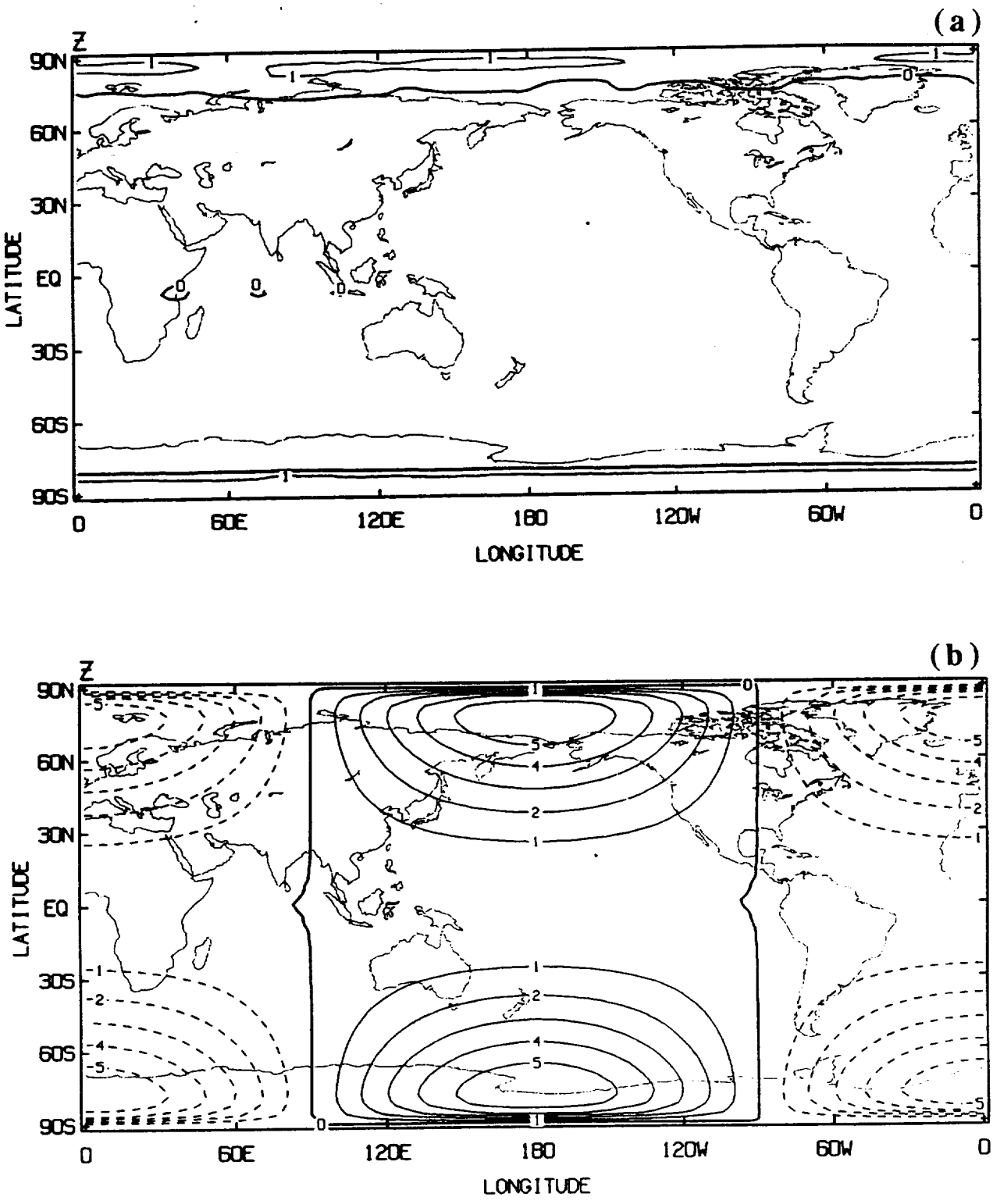
(a)
Z
90N
60N
30N
EQ
30S
60S
90S
LATITUDE
0
60E
120E
180
120W
60W
0
LONGITUDE
(b)
Z
90N
60N
30N
EQ
30S
60S
90S
LATITUDE
0
60E
120E
180
120W
60W
0
LONGITUDE

Fig. 3

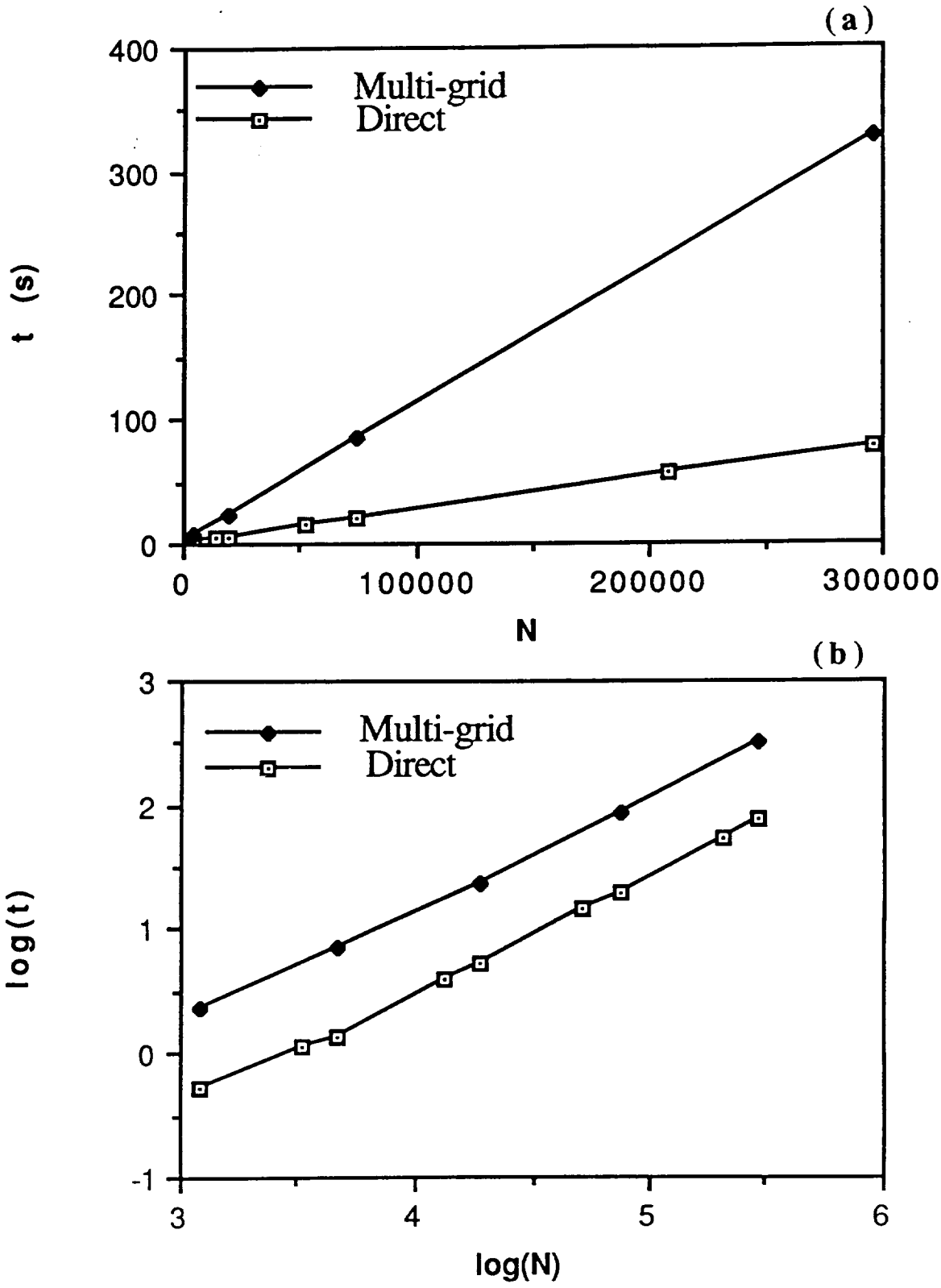
(a)
400
300
200
100
0
t (s)
Multi-grid
Direct
0
100000
200000
300000
N
(b)
3
2
1
0
-1
log(t)
Multi-grid
Direct
3
4
5
6
log(N)

Fig. 4

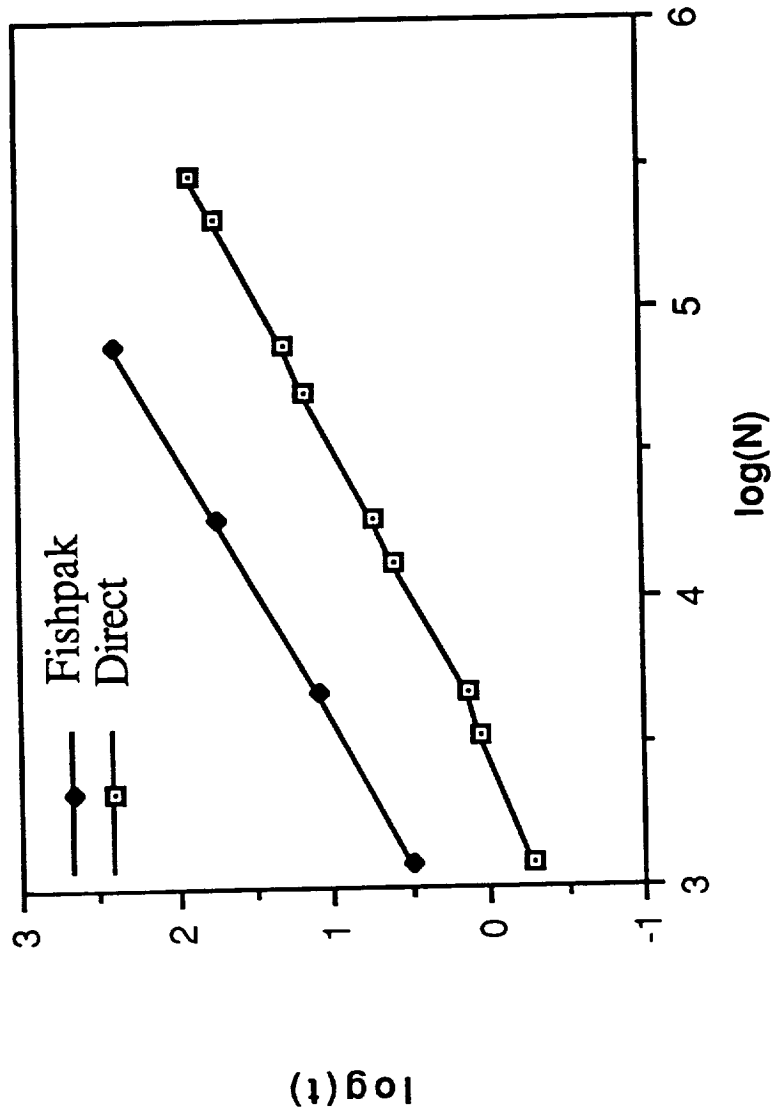

Fig. 5

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
May 1992

3. REPORT TYPE AND DATES COVERED
Technical Memorandum

4. TITLE AND SUBTITLE
A Fast Direct Solver for a Class of Two-Dimensional Separable Elliptic Equations on the Sphere

5. FUNDING NUMBERS

6. AUTHOR(S)

Shrinivas Moorthi and R. Wayne Higgins

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

NASA-Goddard Space Flight Center
Data Assimilation Office
NASA Goddard Space Flight Center
Greenbelt, Maryland 20771

8. PERFORMING ORGANIZATION REPORT NUMBER

92B00092
Code 910.3

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

National Aeronautics and Space Administration
Washington, D.C. 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

TM-104567

11. SUPPLEMENTARY NOTES

Shrinivas Moorthi and R. Wayne Higgins: Resident Research Associates, Universities Space Research Association

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 64

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

An efficient, direct, second-order solver for the discrete solution of a class of two-dimensional separable elliptic equations on the sphere is presented. The method involves a Fourier transformation in longitude and a direct solution of the resulting coupled second-order finite-difference equations in latitude. The solver is made efficient by vectorizing over longitudinal wavenumber and by using a vectorized fast Fourier transform routine. It is evaluated using a prescribed solution method and compared with a multigrid solver and the standard direct solver from FISHPAK.

14. SUBJECT TERMS

separable elliptic equations, direct solver, fast Fourier Transform (FFT), discrete solution

15. NUMBER OF PAGES
39

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	Unlimited