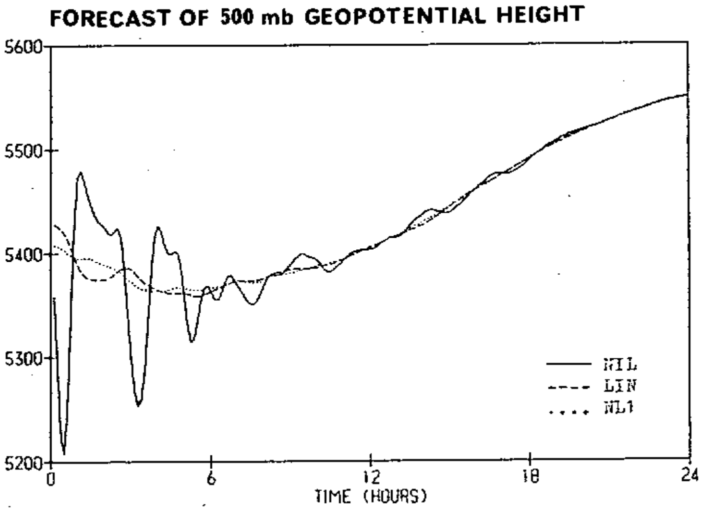
METEOROLOGICAL SERVICE

TECHNICAL NOTE No. 46

INITIALIZATION OF A BAROTROPIC LIMITED-AREA MODEL

USING THE

LAPLACE TRANSFORM TECHNIQUE



By
Peter Lynch M.Sc., Ph.D.

UDC 551.509.313

Price £2.20

GLASNEVIN HILL, DUBLIN 9
JULY 1984

Initialization of a Barotropic Limited-Area Model

using the

Laplace Transform Technique

The Laplace transform technique of initialization developed in Lynch (1984) is used to initialize the data for a barotropic forecasting model over a limited area. The model is described and the numerical formulation of the initialization technique is presented. The initialization is successful in suppressing high-frequency oscillations during the early forecast hours. It has negligible effect upon the resulting 24 hour forecast.

A variation of the linearization, wherein the Coriolis parameter is held constant, is investigated. It is found that the fields which result after a single nonlinear iteration of the modified scheme are almost identical to those resulting from the more general scheme. Since the horizontal variables are separable in the simplified case, the modified scheme is considerably more economical to run.

Table of Contents

1. Introduction

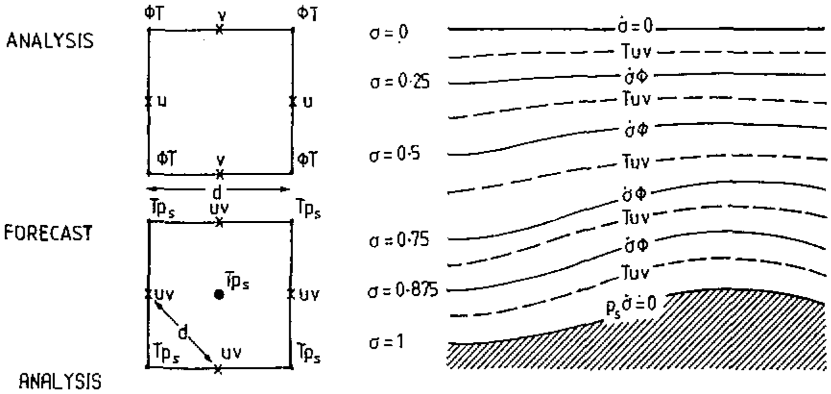
Initialization for limited-area models is a topic of considerable current interest. The nonlinear normal mode method (Machenhauer, 1977; Baer, 1977) provides a very effective method of defining initial conditions for global and hemispheric models. Adaptation of this method for limited-area models poses serious difficulties: it is difficult to determine appropriate normal modes, especially if the horizontal variables do not separate, and to allow for general boundary conditions. An alternative method, which uses a modified inversion formula for the Laplace transform, was proposed by Lynch (1984) and was shown to be effective in controlling high-frequency oscillations in a simple one-dimensional model with periodic boundary conditions.

In the present study the new initialization method will be applied in a more realistic context: we consider a one-level version of the operational forecasting model (LAPEM) used in the Irish Meteorological Service (see Box 1). The primitive equations which govern the flow reduce to a system isomorphic to the shallow water equations; they have linear solutions of both the low frequency rotational and high frequency gravity-inertia wave types. The forecast is performed over a limited area covering Europe, the North Atlantic and Eastern Canada (see *e.g.* Figure 2). In this study the boundary conditions are held constant, although there is no difficulty in principle to incorporate time-varying boundary conditions. The initial data for the model is taken from the standard 500 *mb* analysis. Since this analysis normally contains spuriously large gravity-wave components, the resulting forecast exhibits large amplitude high frequency oscillations during the early forecast hours. These oscillations are gradually dissipated by a light diffusive damping which is applied near the boundaries of the forecast area. (A model option to apply divergence damping (Sadourny, 1975) over the entire area is not used in the present study).

BOX 1

Irish Meteorological Service

Numerical Analysis and Forecasting System



ANALYSIS

Method	3-D univariate optimal interpolation
Analysed variables	Sea-level pressure, geopotential, temperature, wind components
Grid	Arakawa D-grid on polar stereographic projection; d≈300km at 60°N
Levels	1000, 850, 700, 500, 400, 300, 250 and 200 mb
Observation types	SYNOP,TEMP,PILOT,AIREP,SATOB
First guess	ECMWF 12 or 24 hour forecasts of geopotential and wind
Data cycle	12 hours

PREDICTION

Formulation	Primitive equations
Independent variables	λ, ϕ, σ, t
Dependent variables	T, u, v, p_s
Diagnostic variables	Φ, $\dot{\sigma}$
Grid	Arakawa E-grid, transformed lat/long, *d=1.4° (see figure); five levels in vertical*
Integration domain	Europe, North Atlantic, Eastern Canada
Finite difference scheme	Second order accuracy
Time-integration	Advection: semi-Lagrangian scheme (Δt=30min) Adjustment: foreward-backward for gravity wave terms; trapezoidal implicit for Coriolis terms (Δt=7.5min)
Parameterization	Orography: smooth Surface friction (Ekman) Vertical eddy momentum transport Horizontal diffusion: near boundaries Dry convective adjustment (applied every hour) Moisture: not included Radiation: not included
Boundaries	Upwind differencing scheme. Values from ECMWF forecasts updated every 6 hours.

To control the initial spurious oscillations the analysis (at 500*mb*) is initialized using the Laplace transform method. The forecast from the balanced initial fields evolves very smoothly, without any initial shock or subsequent large oscillations. In the specific case considered, the analysis for 00Z on 22nd November, 1982, the RMS (root-mean-square) difference between the original and balanced analysis is about 10 metres, with a maximum difference of less than 40 metres. The RMS difference in wind speed is about 2½ metres/second. The 24 hour forecasts resulting from the two analyses are virtually identical: the maximum differences in height and wind are 4 metres and ½ metre/second. Thus, the initialization process does not affect the final forecast, but it controls the high frequency oscillations by removing spuriously large gravity-wave components from the analysis.

Since gravity waves are ultimately dissipated by the in-built damping of the model, the quality of forecasts is neither enhanced nor impaired by the initialization process. However, the absence of noise in the early forecast hours means that short-range forecasts may be used as preliminary fields in a data assimilation cycle. It is also hoped that properly balanced initial fields will enable us to use longer timesteps in the semi-Lagrangian advection scheme of the operational model.

2. Outline of the Method

A description of the theoretical basis of the Laplace transform technique of initialization can be found in Lynch (1984) and only an outline is given here. We wish to adjust the initial conditions $\mathbf{X}^0$ to ensure the slow evolution of a system whose state $\mathbf{X}(t)$ is governed by the vector equation

$$\frac{d\mathbf{X}}{dt} + \mathbf{L}\mathbf{X} + \mathbf{N}(\mathbf{X}) = \mathbf{0} \qquad (1)$$

where $\mathbf{L}$ is a constant linear operator and $\mathbf{N}$ is a nonlinear vector function. The technique developed in Lynch (1984) may be summarised as follows: let $\mathbf{X}_n^0$ be the *n*th estimate of the balanced initial conditions; we approximate the nonlinear term by $\mathbf{N}(\mathbf{X}_n^0)$, assumed constant; then, the next estimate of the transformed solution is given by

$$\hat{\mathbf{X}}_{n+1} = (s\mathbf{I} + \mathbf{L})^{-1}[\mathbf{X}_n^o - \mathbf{N}(\mathbf{X}_n^o)/s] \qquad (2)$$

where $\mathbf{I}$ is the identity matrix and s the variable in the Laplace transform. The next estimate of the initial conditions is then given by

$$\mathbf{X}_{n+1}^0 = \mathcal{L}^*\{\hat{\mathbf{X}}_{n+1}\} \equiv \frac{1}{2\pi i}\oint_{C^*} \hat{\mathbf{X}}_{n+1}(s)\cdot ds \qquad (3)$$

where the contour $\mathbf{C}^*$ in the s-plane is a circle of radius γ centred at the origin. The value of γ is chosen to lie between the low frequencies (which we wish to preserve) and the high frequencies (which we wish to eliminate). Normally only one nonlinear iteration is required in the case of a one-level model.

The contour $\mathbf{C}^*$ is approximated by an inscribed polygon and the integral in (3) is calculated by evaluating the integrand at the centre, s_n, of each side Δs_n and forming a sum as follows

$$\oint_{C^*} \hat{f}(s)\cdot ds \doteq \sum_{n=1}^{N} \hat{f}(s_n)\cdot\Delta s_n. \tag{4}$$

A constant c has Laplace transform c/s. It is straightforward to show that the approximation (4) with $\hat{f}(s)=c/s$ overestimates c by a factor

$$\kappa = \tan(\pi/N)\,/\,(\pi/N)$$

where N is the order of the polygon. This is significant for small N, so we therefore correct the sum in (4) by dividing by κ. This gives excellent results with as few as eight points around C^* (an octogon).

When the original function $f(t)$ is real we have $\hat{f}(\bar{s})=\overline{\hat{f}(s)}$ and it is easy to show that

$$\frac{1}{2\pi i}\oint_{C^*} \hat{f}(s)\cdot ds = \frac{1}{\pi}\int_{C_1} Im[\hat{f}(s)\cdot ds] \tag{5}$$

where $Im[\cdot]$ is the imaginary part and C_1 is the upper half of C^*, traversed anti-clockwise. Since the dependent variables $X(t)$ are real in the present problem, the use of (5) halves the work required, and only four evaluations of the transformed function on C_1 are needed to give satisfactory results.

3. Description of the Forecasting Model.

The model used in this study is essentially a one-level version of the operational forecasting model of the Irish Meteorological Service. For the purposes of initialization the governing equations are nondimensionalized using length- and time-scales a and $(2\Omega)^{-1}$. The geopotential is split into a mean part, $\bar{\Phi}$, and a deviation therefrom, Φ'. The equations can then be written in the form

$$\frac{\partial\Phi}{\partial t} + (1/\epsilon)\nabla\cdot\mathbf{V} = -N_{\Phi} \qquad (6)$$

$$\frac{\partial u}{\partial t} - fv + \frac{\partial\Phi}{\partial x} = -N_u \qquad (7)$$

$$\frac{\partial v}{\partial t} + fu + \frac{\partial\Phi}{\partial y} = -N_v \qquad (8)$$

where $dx = \cos\phi\, d\lambda$, $dy = d\phi$ and $\epsilon = (2\Omega a)^2/\bar{\Phi}$. The nonlinear terms N_{Φ}, N_u and N_v have been collected on the right hand side; all other notation is conventional.

The equations are integrated over a limited area with a transformed latitude/longitude coordinate system: the North Pole of the transformed grid is at 30°N, 150°E, obtained by rotating the geographic grid through $\lambda_0 = -30°$ about the geographic polar axis and then through $\phi_0 = 60°$ in the plane of 30°W-150°E. In the transformed (λ,ϕ) coordinates the Coriolis parameter is of the form

$$f = [(\cos\phi_0)\sin\phi + (\sin\phi_0)\cos\lambda\cos\phi]$$

and is thus a function of both coordinates of the new system. This seriously complicates the linear analysis by making the horizontal variables non-separable.

The integration area is spanned by 81×51 gridpoints, with geopotential and winds being specified at alternate intersections of a $1^{o}\times 1^{o}$ mesh (Arakawa E-grid). Thus, the grid spacing between like points is 157 *km* at the model equator. The timestep is fixed at $\Delta t = 450\,s$ for both advection and adjustment terms; this ensures the stability of the gravity waves. A split explicit method is used to integrate the equations: the advection is handled using a multiply-upstream semi-Lagrangian scheme with biquadratic interpolation (Bates and McDonald, 1982); a forward-backward scheme is used for the gravity wave terms and a trapezoidal (pseudo-)implicit scheme for the Coriolis terms (Mesinger and Arakawa, 1976). The treatment of divergence in the continuity equation prevents the occurence of two-grid-interval noise.

The variables on the outermost boundary line are held constant and those on the first inner line are evaluated at each timestep by linear interpolation from the four surrounding points. Bilinear interpolation is also used for the Lagrangian advection scheme on the next three lines, which results in some damping. In addition, light diffusive damping is applied over the five outermost lines of the grid. Divergence damping (Sadourny, 1975) is optionally applied over the entire area; it is not used in this study.

4. Numerical Formulation of the Initialization Method.

For a one-dimensional model the application of the Laplace transform technique was straightforward (Lynch, 1984). For the present barotropic model the state vector **X**(t) contains some 6000 elements. Thus, the matrices **M**(s) = (sI+L) are enormous, and impossible to invert numerically, so the problem must be formulated to produce matrices of manageable size.

The Laplace transform of equations (6)-(8) may be written

$$s\hat{\Phi} + (1/\epsilon)[\hat{u}_x + \frac{(\hat{v}\cos\phi)_y}{\cos\phi}] = \Phi^0 - \hat{N}_\Phi \qquad (9)$$

$$s\hat{u} - f\hat{v} + \hat{\Phi}_x = u^0 - \hat{N}_u \qquad (10)$$

$$s\hat{v} + f\hat{u} + \hat{\Phi}_y = v^0 - \hat{N}_v \qquad (11)$$

where $\hat{\Phi}$ denotes the Laplace transform of Φ, *etc.* We discretise the domain and replace spatial derivatives by centred differences in the usual way. For the forecasting model the staggered grid is indexed by specifying I and J, where I is the grid-number in the x-direction and J numbers each pair of horizontal rows (see Figure 1a). To keep the matrices for the initialization scheme as small as possible it is convenient to re-label the grid by specifying M, the number of each triplet of values (Φ, u, v), and the row number N (see Figure 1b). Furthermore, since there are fewer points in the North-south than in the East-west direction, it is advantageous to assemble the vectors appearing below from columns (constant I) of the original grid.

The values of Φ, u and v on a single 'row' of the re-labled grid are collected in the vector

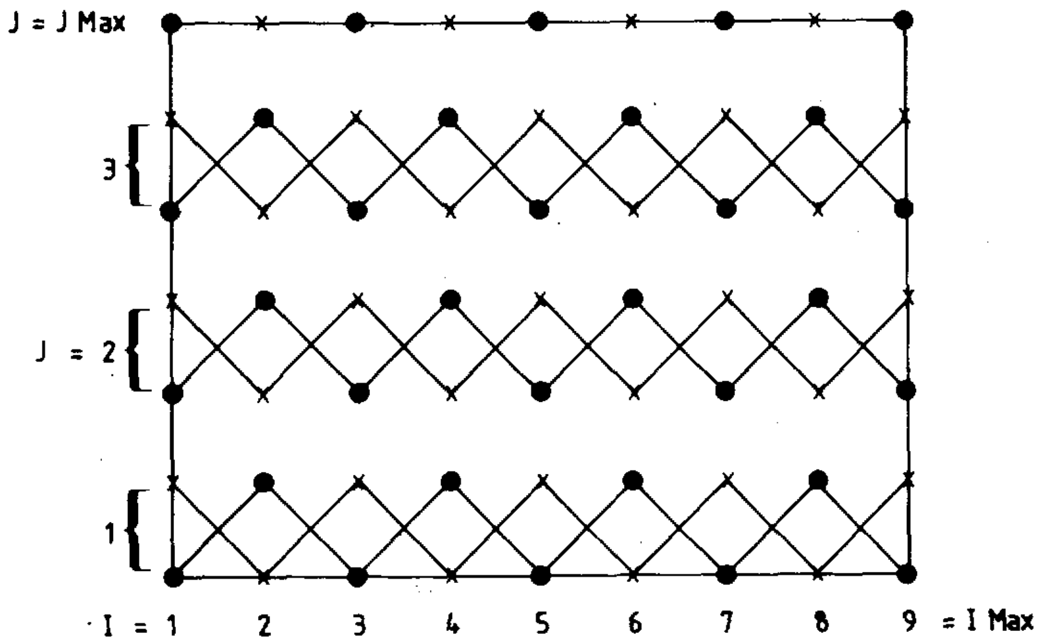


Figure 1a: Specification of the E-grid for the forecasting model. In this case there are in total 9 x 7 points so that I Max = 9 and J Max = 4. Geopotential and winds are given at alternate points: geopotential points are marked by dots and winds by crosses.

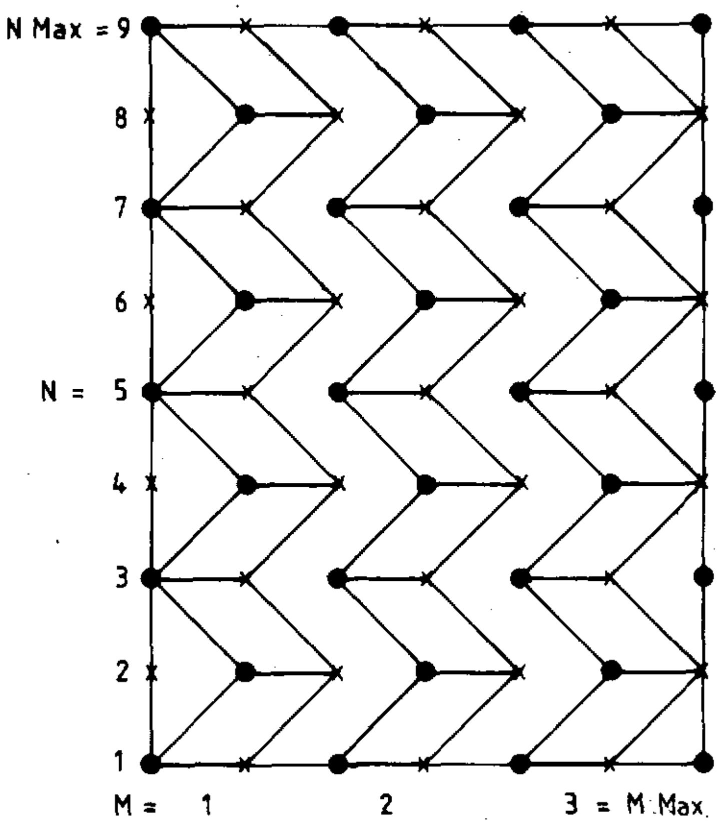


Figure 1b: Specification of the 'MN grid' used in the initialization. It may coincide with the E-grid, or (as in this case) be rotated through 90°. Here I Turn = 1, M Max = (J Max) -1 and N Max = I Max.

$$X_n = (\Phi_{1n}, u_{1n}, v_{1n}, \cdots, \Phi_{Mn}, u_{Mn}, v_{Mn})$$

and the transformed vector $\hat{X}_n$ is defined in a similar manner. Because of the grid staggering, some boundary points are included in X_n. We assume that these are constant and that points adjacent to the boundaries are defined by interpolation from the surrounding points. Thus, for example, we have

$$\dot{\Phi}_{1,2n-1} = 0$$

$$\dot{\Phi}_{1,2n} - \frac{1}{4}(\dot{\Phi}_{2,2n-1} + \dot{\Phi}_{2,2n+1}) = 0$$

which transform to give the following relations

$$s\hat{\Phi}_{1,2n-1} = \Phi^0_{1,2n-1}$$

$$s\hat{\Phi}_{1,2n} - \frac{s}{4}(\hat{\Phi}_{2,2n-1} + \hat{\Phi}_{2,2n+1}) = \Phi^0_{1,2n} - \frac{1}{4}(\Phi^0_{2,2n-1} + \Phi^0_{2,2n+1})$$

Similar relations hold for the geopotential on the 'right hand side', and for the velocity components on and near the boundary.

The system (9)-(11) may now be written in the form of a set of matrix equations

$$A_n\hat{X}_{n-1} + B_n\hat{X}_n + C_n\hat{X}_{n+1} = D_n \qquad (12)$$

where A_n, B_n and C_n are block-tridiagonal matrices whose elements depend upon the coefficients of the equations and D_n is a column vector of initial values plus transformed nonlinear terms. The lateral boundary values u_{1n}, u_{Mn}, etc. also occur in the vector D_n. The forms of the matrices are given in Appendix C.

Lindzen and Kuo (1969) have described a reliable direct method for solving systems of the form (12). We introduce a set of intermediate matrices and vectors, $\boldsymbol{\alpha}_n$ and $\boldsymbol{\beta}_n$, such that

$$\hat{X}_n = \boldsymbol{\alpha}_n\hat{X}_{n+1} + \boldsymbol{\beta}_n \tag{13}$$

When (13) is used to eliminate $\hat{X}_{n-1}$ from (12) a solution for $\boldsymbol{\alpha}_n$ and $\boldsymbol{\beta}_n$ is apparent in the form

$$\boldsymbol{\alpha}_n = -(M_n)^{-1}C_n \tag{14}$$

$$\boldsymbol{\beta}_n = -(M_n)^{-1}(A_n\boldsymbol{\beta}_{n-1}-D_n) \tag{15}$$

where $M_n=(A_n\boldsymbol{\alpha}_{n-1}+B_n)$. The boundary conditions for $n=1$ are used to obtain $\boldsymbol{\alpha}_1$, $\boldsymbol{\beta}_1$ and from (14)-(15) we then get $\boldsymbol{\alpha}_n$, $\boldsymbol{\beta}_n$. The boundary conditions at $n=N$ give us $\hat{X}_N$ and the solution $\hat{X}_n$ is then obtained from (13). The crucial point is that the matrices M_n are independent of the boundary values and of the forcing terms; therefore, they can be calculated *once for all* and stored on disk. The main computational effort is then in the matrix multiplications in (14) and (15); the sparse nature of A_n and C_n can be used to reduce this effort.

The first (linear) step in the initialization is performed after setting the nonlinear terms in D_n to zero. The system (12) is solved for $\hat{X}^1_n(s)$ and this is inverted on the modified contour C^* to give X^0_1, the linearly initialized fields. The nonlinear terms are evaluated either directly from these fields or by making a single timestep forecast. They are incorporated in the forcing vector D_n and the initialization cycle is repeated as often as required. In the case of a barotropic model a single nonlinear iteration is normally sufficient for convergence.

TABLE 1

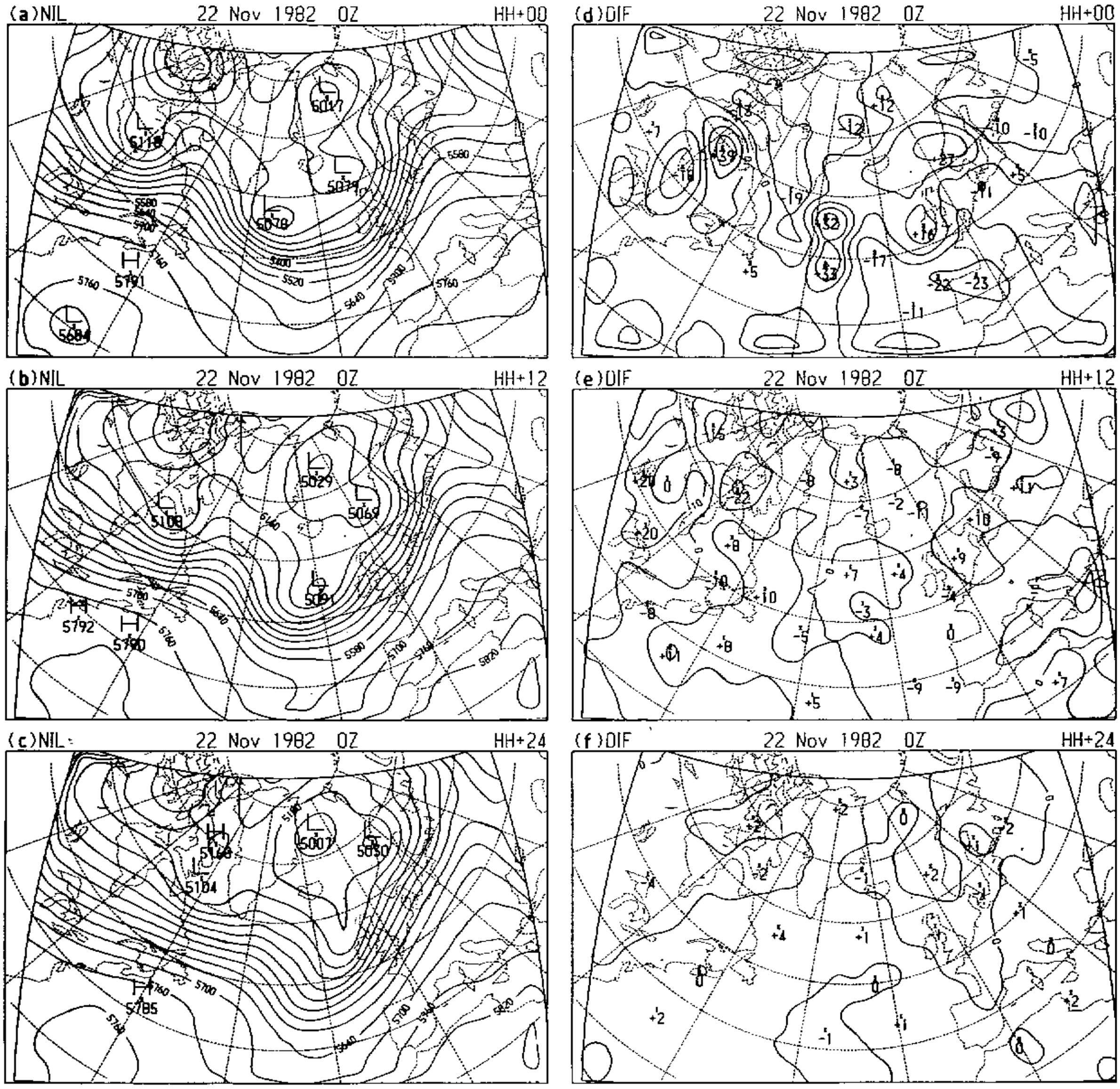
Root-mean-square (and maximum) changes to the geopotential height and wind fields due to each iteration of the initialization, and to the linear and first nonlinear iterations combined.

		z (*m*)	u (*m/s*)	v (*m/s*)
LIN	rms	16·9	2·56	2·68
	(max)	(77·5)	(15·33)	(12·72)
NL1	rms	10·9	0·23	0·11
	(max)	(46·9)	(0·85)	(0·32)
NL2	rms	0·5	0·02	0·02
	(max)	(1·9)	(0·12)	(0·09)
LIN + NL1	rms	9·5	2·53	2·64
	(max)	(38·9)	(15·32)	(12·78)

5. Results

Several test runs have been made with varying grid resolutions and other parameter values. The results described below are for the 500mb analysis valid at the initial time 00Z on 22nd November 1982. The grid resolution is $1^\circ \times 1^\circ$ (E-grid). The cut-off frequency for the inversion integral (3) is chosen by setting $\gamma = 0{\cdot}5$; this corresponds to eliminating all components with period less than 24 hours. The inversion contour is approximated by a regular octagon — thus, only four points are needed to integrate over the upper semi-circle. With the κ-correction described in Section 2 this is found to provide sufficient accuracy. One linear and one non-linear iteration of the initialization procedure are applied; it is found that the changes due to a second nonlinear iteration are very small (presumably further iterations would be needed in the baroclinic case where the equivalent depths are progressively smaller). The nonlinear terms may be evaluated directly within the initialization or by making a single timestep forecast of the model. In the present case it was found to be simpler to calculate them directly. The vectors $\mathbf{X}_n$ were assembled from vertical columns of values on the model grid (Figure 1). The size of the matrices is thus 75X75 as against 120X120 for horizontal sorting. Boundary values were held constant in all cases.

In Table 1 we show the RMS changes (and maximum changes) to the geopotential height and wind fields due to each iteration of the initialization and to the linear and first non-linear iterations combined. The changes of the height field are quite large for the linear (LIN) and first non-linear (NL1) iterations; the overall change due to the two (LIN+NL1) is somewhat less. The winds change markedly during the linear step but very little thereafter. In all cases there is hardly any change due to the second non-linear iteration (NL2); therefore, the results presented below refer to the case of a single non-linear iteration (LIN+NL1).



(a)NIL 22 Nov 1982 0Z HH+00

(d)DIF 22 Nov 1982 0Z HH+00

(b)NIL 22 Nov 1982 0Z HH+12

(e)DIF 22 Nov 1982 0Z HH+12

(c)NIL 22 Nov 1982 0Z HH+24

(f)DIF 22 Nov 1982 0Z HH+24

<u>Figure 2:</u> (a) Original 500 *mb* analysis valid at 00Z, 22nd November, 1982; (b), (c) 12 and 24 hour forecasts starting from this analysis; (d) Initialized *minus* original analysis; (e), (f) differences at 12 and 24 hours between forecasts from original and initialized analyses.

TABLE 2

Root-mean-square (and maximum) differences in the geopotential height and wind fields between the original and initialized fields (NL1 *minus* NIL) and between the 12 and 24 hour forecasts resulting from these fields.

		z (*m*)	u (*m/s*)	v (*m/s*)
HH+00	rms	9·5	2·53	2·64
	(max)	(+38·9)	(-15·32)	(+12·78)
HH+12	rms	6·1	0·25	0·21
	(max)	(-22·3)	(-0·97)	(-0·78)
HH+24	rms	1·0	0·11	0·09
	(max)	(+4·1)	(-0·55)	(+0·37)

Figure 2(a) shows the initial 500mb height analysis and 2(b) and 2(c) are the HH+12 and HH+24 hour forecasts resulting from this analysis. In Figure 2(d) we show the changes to the height field due to the initialization (LIN+NL1), and 2(e) and 2(f) show the differences, at 12 and 24 hours, between the forecasts starting from the two analyses (initialized *minus* original). The changes to the analysis, and also after 12 hours, are quite significant. The similarity between the two 24 hour forecasts is remarkable: *the maximum height difference is only 4 meters, and for practical purposes the forecasts are identical.* Further results are presented in Table 2, and they confirm the convergence between the two forecasts.

The effect of initialization on the evolution of the flow is indicated by several diagnostics. In Figure 3 we show the geopotential at a central point (I=37, J=9) resulting from the initial fields and after linear and non-linear initialization. The reduction of the initial oscillations in the linear case (LIN) is dramatic, and the evolution after NL1 is very smooth. Similar graphs of the evolution of the divergence (at the same central point) tell much the same story: the divergence fluctuates wildly if the initial fields are out of balance (Figure 4); this fluctuation is controlled by initialization.

The RMS divergence and global mean divergent kinetic energy give good overall measures of the noise in the evolution of the flow. The effects of the initialization upon these quantities are shown in Figures 5 and 6. In both cases there is a dramatic reduction of the noise in the foreacst when the fields are *initially balanced*.

All the above diagnostics confirm that the initialization (LIN+NL1) is successful in removing spurious oscillations from the early forecast and results in a noise-free evolution of the flow. The remarkable agreement between the 24 hour forecasts *before* and *after* (Figure 2(f)) demonstrates that the process is doing precisely what is required: removing high-frequency gravity waves without perturbing the development of the meteorological flow.

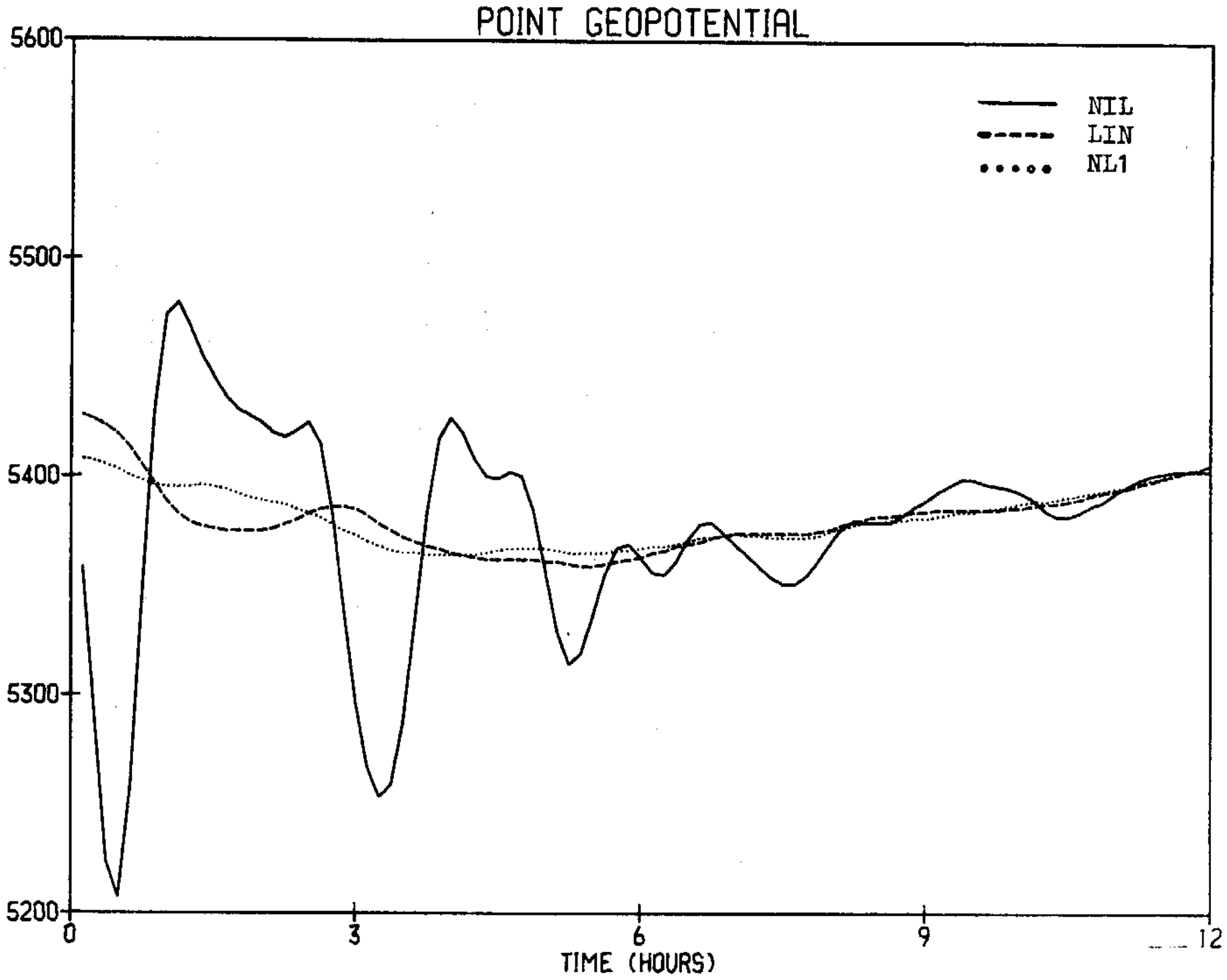


Figure 3: Geopotential height (*metres*) at the gridpoint I = 37, J = 9 for the first 12 hours of the forecast starting from uninitialized fields (solid), linearly initialized fields (dashed) and nonlinearly initialized fields (dotted).

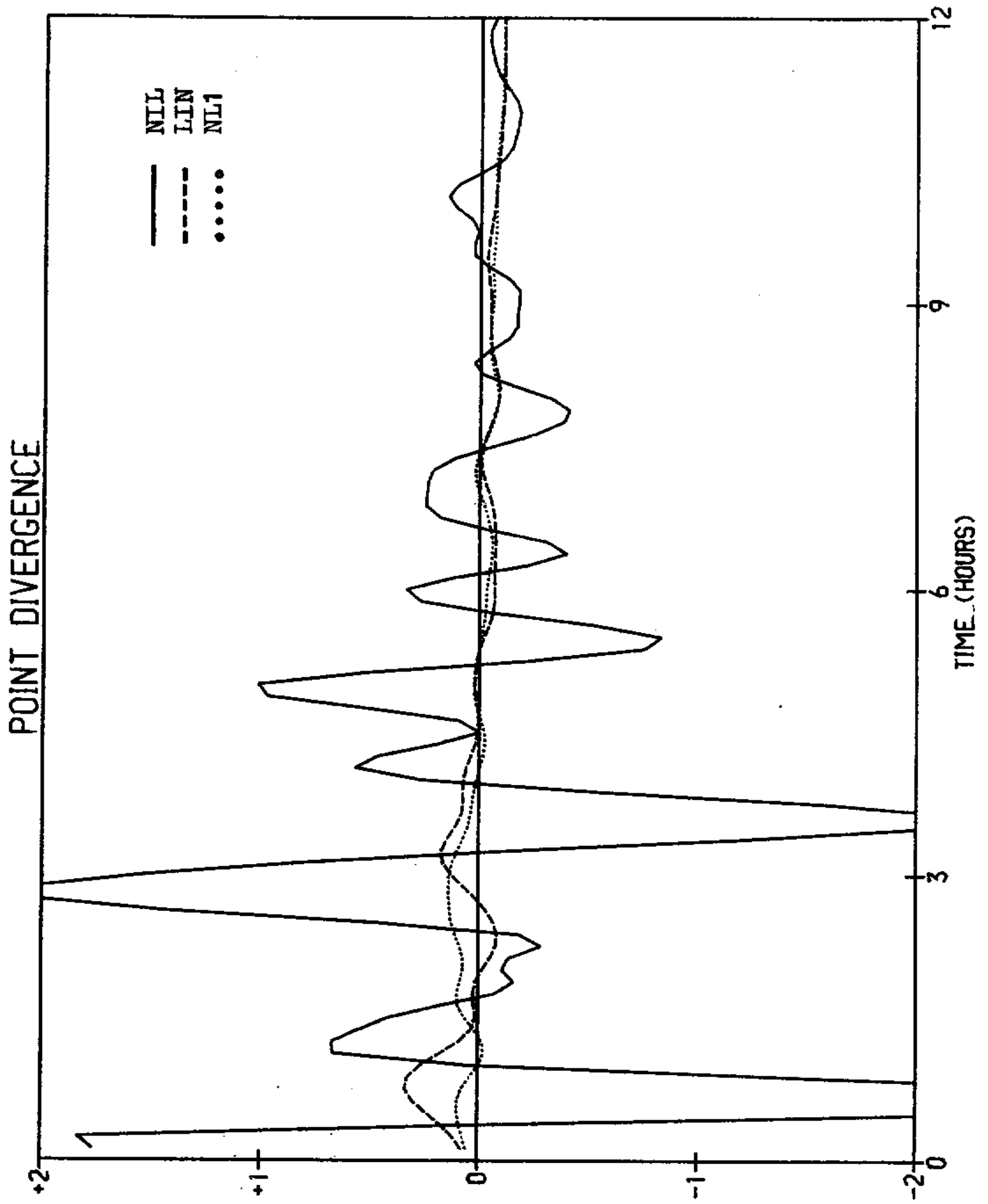


Figure 4: Divergence at a central point, I = 37, J = 9 (units $\times 10^5\ s^{-1}$) for the first 12 hours of the forecast starting from uninitialized fields (solid), linearly initialized fields (dashed) and nonlinearly initialized fields (dotted).

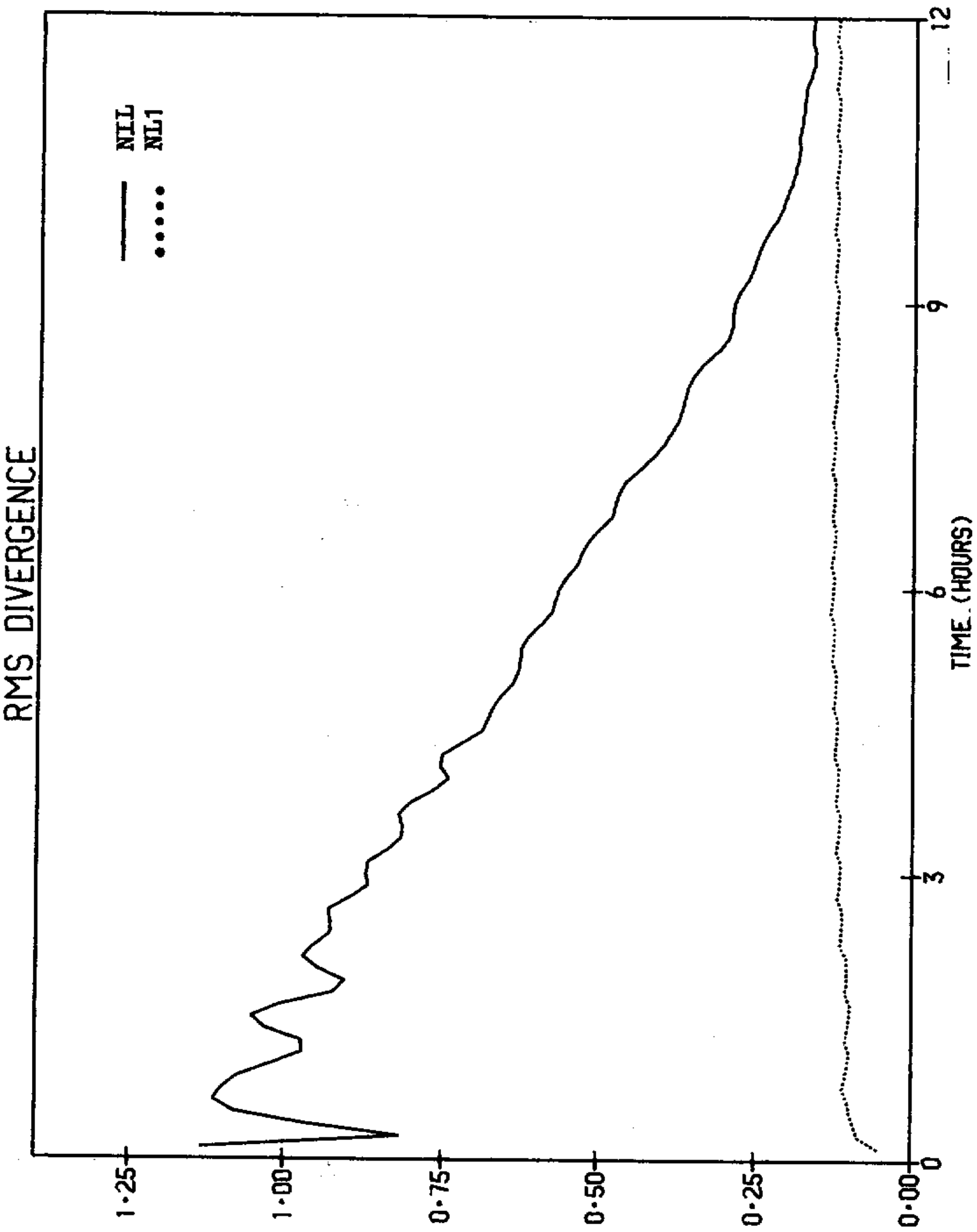


Figure 5: Root-mean-square divergence (units $\times 10^5$ s^{-1}) for the first 12 hours of the forecast starting from uninitialized fields (solid) and nonlinearly initialized fields (dotted).

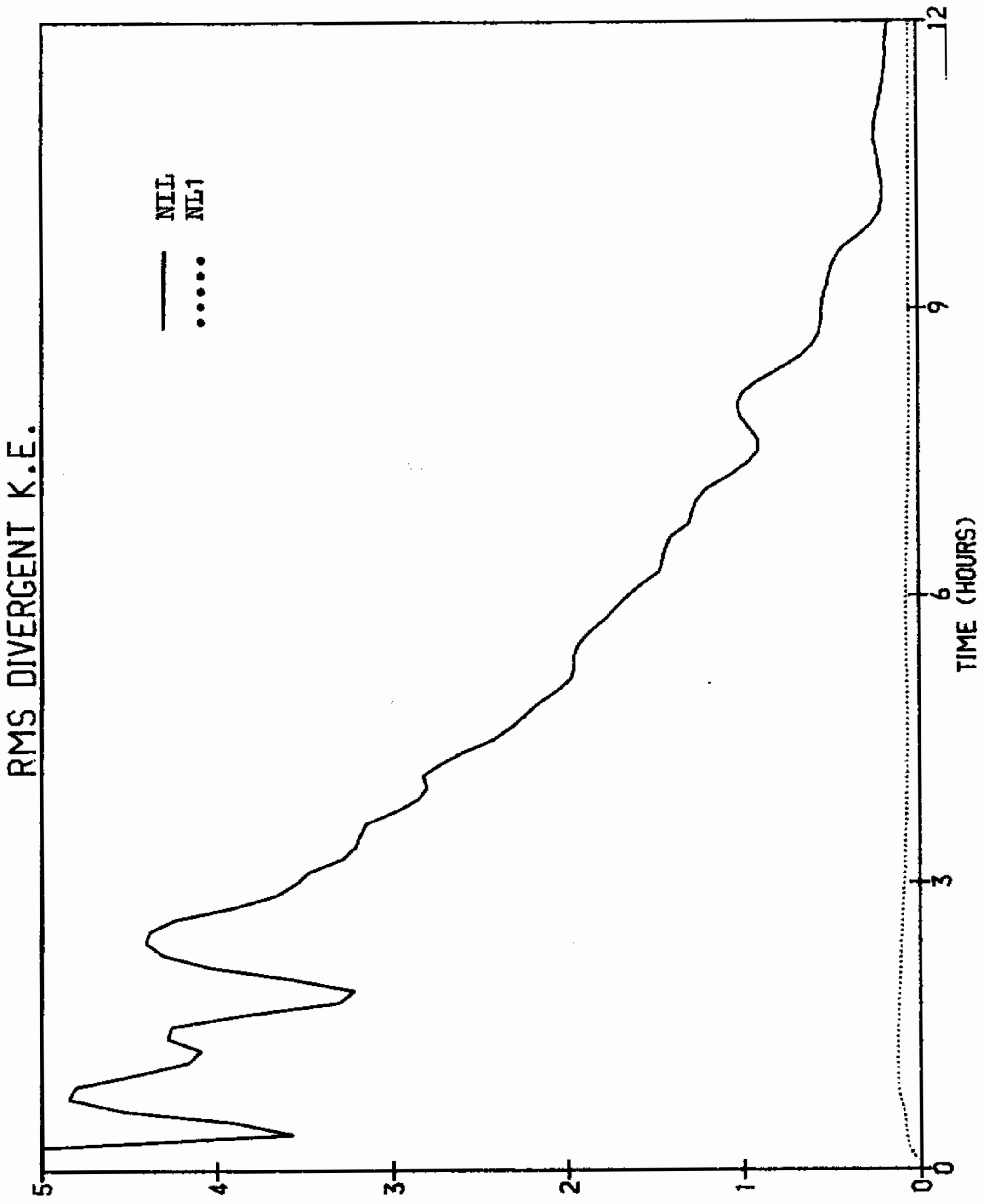


Figure 6: Root-mean-square divergent kinetic energy per unit mass ($m^2 s^{-2}$) for the first 12 hours of the forecast starting from uninitialized fields (solid) and nonlinearly initialized fields (dotted).

6. A Simplified Linearization

The development of the present method of initialization was guided by an intuitive feeling that it is important to include the full variation of the Coriolis parameter (the β-effect) in the linearized equations. Ballish (1979) showed, in the context of a one-dimensional model, that the omission of the β-terms from the eigenvector analysis leads to larger oscillations than if they are included. It seemed likely that this would also be true for a more general model. The question is examined below.

The initialization procedure was modified in the following way: The Coriolis parameter f occuring in the linear terms of equations (7) and (8) was replaced by its mean value f_0, and its variation was accounted for by including the factors $-(f-f_0)v$ and $+(f-f_0)u$ in the nonlinear terms N_u and N_v of these equations. Thus, the nonlinear equations to be solved are unchanged, but they are split into linear and nonlinear parts in a different way.

The difference between the original ($f=f(\lambda,\phi)$) and simplified ($f=f_0$, constant) initialization schemes can be seen from Figure 7. In Figure 7(a) we show the difference in the 500 *mb* analyses resulting from the original and simplified schemes after linear initialization. Since the linear equations used in the two cases differ, it is hardly surprising that the two analyses differ by as much as 30 *metres*, with an RMS difference of 10 *metres*. In contrast to this, Figure 7(b) shows that after a single nonlinear iteration the two schemes produce very similar analyses: the maximum difference is 1·6 *metres*, and the RMS difference only 0·5 *metres*. The maximum difference in the corresponding wind analyses is only 0·14 *metres/second*. For practical purposes the two analyses are identical. The noise profiles produced by the forecasts from the two analyses are indistinguishable.

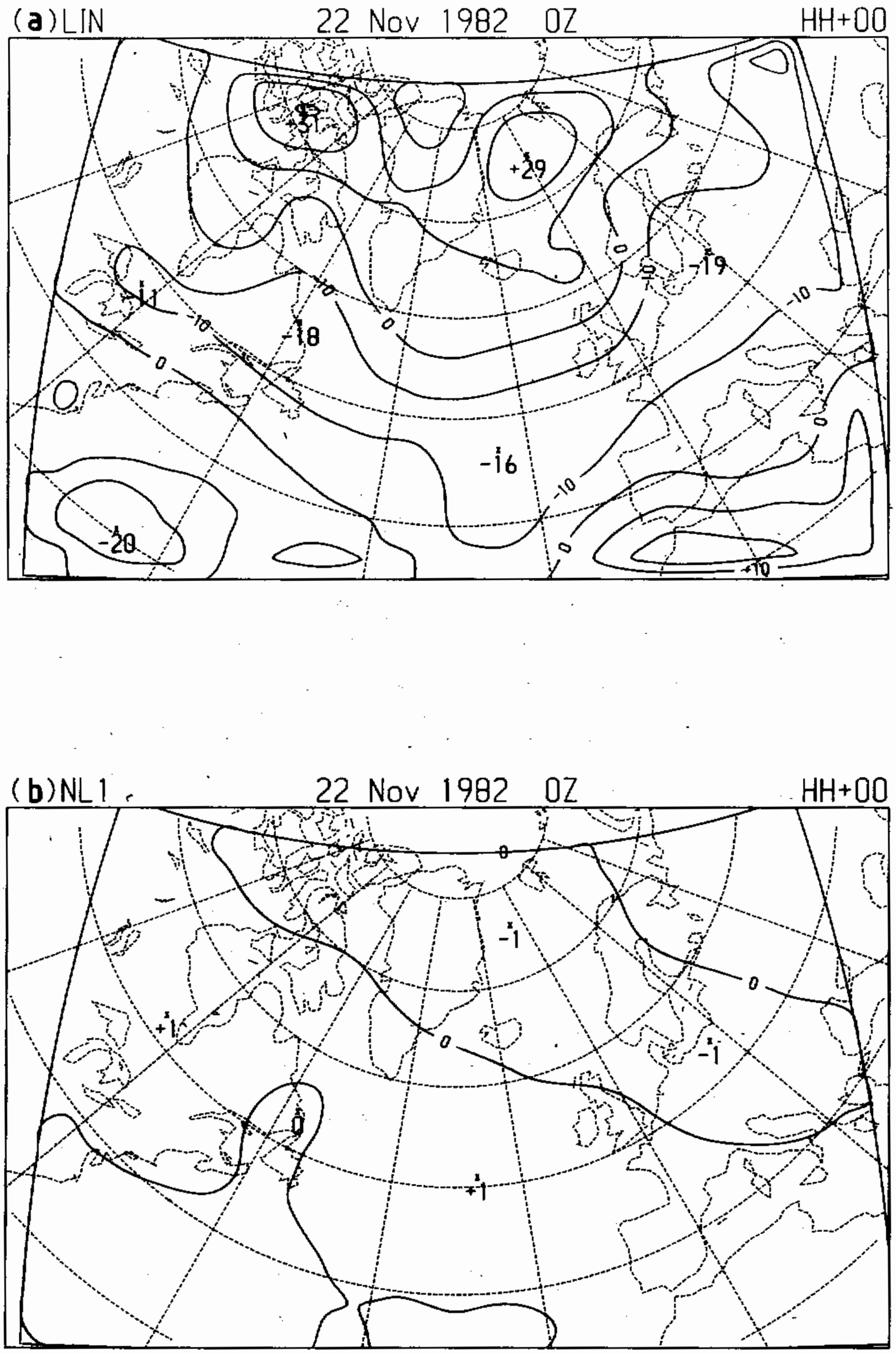


Figure 7: Difference between the initialized 500 mb analyses resulting from the original and simplified (f constant) schemes, (a) after linear initialization and (b) after one nonlinear iteration.

The simplified scheme gives results equivalent to the original method, and results in linearized equations in which separation of the horizontal variables obtains. It is therefore possible with this scheme to develop a more efficient initialization procedure. Furthermore, the amount of disk storage should be considerably reduced.

It seems highly probable that the equivalence of the two methods will also hold in the case of a baroclinic model. However, the geographical extent of the analysis area must be taken into account. It is not clear how important the inclusion of the β-terms in the linear equations may become if the area extends to or straddles the equator. This question will be addressed elsewhere.

7. Summary.

The Laplace transform technique has been applied in the context of a one-level limited-area model. The results have shown that the method is capable of removing spurious gravity-wave noise without having any adverse affect on the resulting forecast. There do not appear to be any problems associated with the boundaries.

The simplification of the linearization, in which the Coriolis parameter is taken as constant, is found to produce results which are almost identical to those obtained with the more general scheme. This fact may allow us to develop a more efficient procedure for operational purposes.

It is intended to extend the initialization method for application to the operational baroclinic model with variable boundary conditions. The separation of vertical structure can be done in exactly the same way as for the nonlinear normal mode method (see, *e.g.*, Kasahara and Puri, 1981). The results of this work will be reported in a future Technical Note.

APPENDIX A

The Relationship between the Laplace Transform Technique and Nonlinear Normal Mode Initialization.

Suppose that the linear normal modes of the system governed by equation (1) are known and that they span the space $\mathcal{X}$. Then X may be expressed as a sum of these modes and L becomes a diagonal matrix Λ:

$$\Lambda \;=\; \mathrm{diag}(\lambda_1,\lambda_2,\lambda_3,\;\cdots\;) \;=\; (\Lambda^*,\Lambda^\dagger)$$

where the eigenfrequencies are arranged in order of ascending absolute value and split into slow modes (Λ^*) and fast modes ($\Lambda^\dagger$). We can define projection operators onto the slow and fast subspaces of $\mathcal{X}$ (see *e.g.* Halmos, 1958):

$$\mathcal{X} = P^*\mathcal{X} \oplus P^\dagger\mathcal{X} = \mathcal{X}^* \oplus \mathcal{X}^\dagger$$

$$P^* = \mathcal{L}^*\mathcal{L} \;\;;\;\; P^\dagger = \mathcal{L}^\dagger\mathcal{L} = (\mathcal{L}^{-1}-\mathcal{L}^*) \qquad ; \;\; P^* + P^\dagger = I$$

and separate the solution into slow and fast parts:

$$X \;=\; P^*X + P^\dagger X \;=\; \begin{bmatrix} X^* \\ \\ X^\dagger \end{bmatrix}$$

If the nonlinear terms are decomposed in the same way

$$N(X) \;=\; P^*N + P^\dagger N \;=\; \begin{bmatrix} N^*(X^*,X^\dagger) \\ \\ N^\dagger(X^*,X^\dagger) \end{bmatrix}$$

we can separate the system (1) into slow and fast parts:

$$\begin{aligned} \dot{X}^* \;+\; \Lambda^*X^* \;+\; N^* \;&=\; 0 \\ \dot{X}^\dagger \;+\; \Lambda^\dagger X^\dagger \;+\; N^\dagger \;&=\; 0 \end{aligned} \qquad \text{(A1)}$$

The initialization is iterated by using equation (2), which here becomes

$$\hat{X}_{n+1} \;=\; (sI+\Lambda)^{-1}[X_n^*(0) - N_n/s] \qquad \text{(A2)}$$

Using partial fractions to split the diagonal elements we can write the matrix inversion:

$$(sI+\Lambda)^{-1}/s \;=\; -\,(sI+\Lambda)^{-1}\,\Lambda^{-1} \;+\; \Lambda^{-1}/s. \qquad (A3)$$

Equation (A2) can then be written in the form

$$\hat{X}_{n+1} \;=\; (sI+\Lambda)^{-1}[X_n^*(0)+\Lambda^{-1}N_n] \;-\; \Lambda^{-1}N_n/s \qquad (A4)$$

To get the new initial conditions we apply $\mathcal{L}^*$ (at $t=0$) to (A4). Since the matrices are diagonal, the components can be considered individually. For slow modes $|\lambda| < \gamma$, the cutoff frequency, and all the terms in (A4) contribute to the solution. After cancellation we find

$$X^*_{n+1}(0) \;=\; X^*_n(0) \qquad (A5)$$

Thus, the coefficients of the slow modes remain unchanged. For the fast modes $|\lambda| > \gamma$ and the terms in (A4) involving $(sI+\Lambda)^{-1}$ contribute nothing to the integral $\mathcal{L}^*$. Thus, we have

$$X^{\dagger}_{n+1}(0) \;=\; -\,(\Lambda^{\dagger})^{-1}N^{\dagger}_n \qquad (A6)$$

This is precisely the iterative solution of the equation

$$\Lambda^{\dagger}X^{\dagger} \;+\; N^{\dagger}(X^*,X^{\dagger}) \;=\; 0$$

which is obtained by setting $\dot{X}^{\dagger}=0$ in (A1). Equations (A5) and (A6) show the equivalence between the Laplace transform technique and the nonlinear normal mode method as formulated by Machenhauer (1977).

Appendix B: Outline of the Program FILTER

A brief description of the computer implementation of the initialization scheme is given here (extensive comments are also to be found in the code). The main program is called FILTER and the source version is in the file FILTER.E2D. The global variables and arrays are in COMMON blocks defined in FILTER.COM. Control of the flow of computations is through the card file FILTER.CDS and output messages and diagnostics go to FILTER.LPT. The fields to be initialized are in HINTRP.(B)IN and the balanced fields are written in the same format to HINTRP.OUT.

The program runs in one of two modes determined by the LOGICAL variable MODEL: if MODEL is .TRUE. the nonlinear terms are obtained by running the forecast model (TRIAL) for one timestep. In this case each iteration of the initialization involves a separate run of the program FILTER, and these runs alternate with one-step runs of TRIAL. If MODEL is .FALSE. the nonlinear terms are calculated directly within FILTER, and several iterations can be made in the same run.

The flow of computations is outlined below.

(1) GETPAR reads the control cards, sets up the parameters for the flow of computations and defines various constants and variables.

(2) GETICS opens the input file HINTRP.(B)IN and reads in the mass and wind fields. If MODEL is .TRUE. and LINEAR is .FALSE. the forecast fields after one timestep are also read in. All fields are then nondimensionalised.

(3) GETBCS extracts boundary values from the fields and stores them in vectors (it is not required in the current version).

(4) SHUFLE rearranges the dependent variables FI(IM,JM), U(IM,JM) and V(IM,JM), which are stored in the zig-zag array form used in the model, into vectors of initial valued stored in XO(MMAX,NMAX). The vectors consist of values in horizontal or vertical rows of the original grid, depending on the value of ITURN (0 or 1). The Coriolis parameter and trigonometric factors are also shuffled to the MN grid.

(5) GETMAT constructs the matrices A_n, B_n and C_n which express the discretization of equation (12). A_n and C_n are sparse, with at most one nonzero element per row, and are stored as vectors (end values are set in KMTXSL, *etc.*). B_n is defined for each point on the inversion contour C^* (IS-loop) and for each row of the grid (N-loop). The inverse of the matrix $M_n = (A_n\alpha_{n-1}+B_n)$ is then calculated and written to a disk file FILTER.DE2. If REPEAT is

.TRUE. the disk file is already available and no matrix inversions are necessary.

(6) SOLVER is the routine which actually solves the system (12). The method of Lindzen and Kuo (1969) is used; an upward sweep through the grid is made to define the intermediate quantities α_n and β_n, followed by a downward sweep to get the (transformed) solution XHAT(M,N). This is done for each point on the inversion contour and the values are then used to calculate the inversion integral (4). The values are then moved to the solution array X(M,N) and the correction factor κ (see equation following (3)) is applied. The right hand side forcing terms are obtained by calling RHSIDE. If LINEAR is .FALSE. the nonlinear terms are calculated by the routines GETNL (if MODEL is .TRUE.) or NONLIN (if MODEL is .FALSE.).

(7) SHUFLE is called again to rearrange the fields into the format of the model grid.

(8) OUTPUT redimensionalizes the initialized fields and writes them to the output file HINTRP.OUT. Various maps are plotted if required, diagnostics are printed and statistics of the changes due to the initialization are calculated and printed.

Appendix C: Form of the Matrices in Equation (12).

Because of the grid staggering, the matrices $\mathbf{A}_n$, $\mathbf{B}_n$ and $\mathbf{C}_n$ occuring in Equation (12) have somewhat differing forms for odd and even values of *n*. For simplicity of description we give their forms here in the case $M=3$; thus, they are 9X9 matrices. The generalization to larger M is obvious: the central row of blocks is repeated as often as necessary, shifted each time three columns to the right.

The matrices $\mathbf{A}_n$ and $\mathbf{C}_n$ are sparse, with at most one nonzero element per row. We define certain quantities which depend upon whether the MN-grid is unrotated or rotated: in the former case we have $d_{m'n}=\cos\phi_n\cdot\Delta\lambda$, $d_n=\Delta\phi$, $c_{m'}=1$ and $c_n=\cos\phi_n$ while in the latter case we have $d_{m'}=\Delta\phi$, $d_{m'n}=\cos\phi_{m'}\cdot\Delta\lambda$, $c_{m'}=\cos\phi_{m'}$, and $c_n=1$, where m' is equal to $(2m-1)$ or $2m$ according as *n* is odd or even. For odd *n* the matrix $\mathbf{A}_n$ then takes the form:

$$\begin{bmatrix}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \sigma & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \sigma & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & a_{2n} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & a'_{2n} & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & a_{3n} & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \sigma & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \sigma & \cdot & \cdot & \cdot
\end{bmatrix}$$

where $\sigma=-s/4$, $a_{mn}=c_{n+1}/(c_n\epsilon d_{mn})$ and $a'_{mn}=1/d'_{mn}$. For even n the matrix $\mathbf{A}_n$ takes the alternative form

$$\begin{bmatrix}
\cdot & \cdot & \cdot & \sigma & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & a'_{1n} & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & a_{2n} & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & a'_{2n} & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \sigma & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{bmatrix}$$

where now 2m replaces (2m-1) and (2m+1) replaces 2m in the definition of a_{mn} and a'_{mn}.

The matrices $\mathbf{C}_n$ are formally identical provided that we redefine $a_{mn} = -c_{n-1}/(c_n \epsilon d_{mn})$ and $a'_{mn} = -1/d_{mn}$. Note that, apart from the σ-elements, $\mathbf{A}_n$ and $\mathbf{C}_n$ are real; thus, they may be stored as real vectors and the σ-elements allowed for explicitly when they are used in calculations (in the procedures KMTXSL, KMTXSR and KVCXSL).

The matrices $\mathbf{B}_n$ also have two forms, depending upon the parity of n. We define $b_{m'n} = \cos\phi_{m'+1}/(\epsilon c_{m'} d_{m'n})$ and $b'_n = 1/d_{m'n}$ and the matrix $\mathbf{B}_n$ can be written, in the case of odd n, in the form

$$\begin{bmatrix}
s & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & s & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & s & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & -b_{mn} & \cdot & s & b_{mn} & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & -b_n & s & -f & b'_n & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & f & s & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & -b_n & \cdot & s & b_n & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & s & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & s
\end{bmatrix}$$

and, in the case of even n, in the form

$$\begin{bmatrix}
s & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
-b'_n & s & -f & b'_n & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & f & s & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & -b_{mn} & \cdot & s & b_{mn} & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & -b'_n & s & -f & b'_n & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & f & s & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & s & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & s & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & s
\end{bmatrix}$$

A listing of the subroutine GETMAT, which defines the matrices $\mathbf{A}_n$, $\mathbf{B}_n$ and $\mathbf{C}_n$, is given below.

```
C****************************************************************
        SUBROUTINE GETMAT
C
C       CONSTRUCT THE MATRICES A, B AND C WHICH EXPRESS
C       THE DISCRETIZATION OF THE EQUATIONS.
C       CALCULATE THE ALFA AND INVERSE MATRICES AND WRITE
C       THE LATTER TO DISK.
C       A AND C DO NOT DEPEND ON S; THEY ARE CALCULATED ONLY ONCE
C       AND HELD IN CORE. MATRIX INVERSIONS ARE DONE ONLY ON THE
C       FIRST RUN AND STORED ON DISK. MATRIX B IS NOT NEEDED ON
C       REPEAT RUNS.
C
        INCLUDE 'FILTER.COM'
C
        TYPE 8883
8883    FORMAT(/'    GETMAT  '/)
C
        NMAXM1 = NMAX-1
        NMAXM2 = NMAX-2
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C         GET THE SPARSE MATRICES A AND C. EACH HAS AT MOST ONE
C         NONZERO ELEMENT PER ROW, AND THEY ARE STORED BY ROWS
C         AND FOR EACH DOUBLY INTERIOR LINE
C         EDGE VALUES ARE ALLOWED FOR IN KMTXSL, ETC.
C
        DO 500 N=3,NMAXM2
          NEVEN = (N/2)*2
          CSNP1 = COSN(N+1)
          CSN   = COSN(N  )
          CSNM1 = COSN(N-1)
          DO 520 M=1,MMAX
            M3 = 3*(M-1)+1
            M2 = 2*M-1
            IF(NEVEN.EQ.N) M2 = 2*M
            COSZ = COSM(M2)*CSN
            CSM = COSM(M2+1)
            IF(M.EQ.1    .AND.NEVEN.NE.N) GO TO 512
            IF(M.EQ.1    .AND.NEVEN.EQ.N) GO TO 511
            IF(M.EQ.MMAX.AND.NEVEN.EQ.N) GO TO 512
             C(M3  ,N) =  RE*CSNP1/(COSZ*DELN)
             A(M3  ,N) = -RE*CSNM1/(COSZ*DELN)
511         CONTINUE
            IF(M.EQ.MMAX.AND.NEVEN.NE.N) GO TO 512
             C(M3+2,N) =  1./(CSM*DELN)
             A(M3+2,N) = -1./(CSM*DELN)
512         CONTINUE
520       CONTINUE

500     CONTINUE
C
        IF(REPEAT) RETURN
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C       OPEN THE DISK FILES FOR STORING THE INVERSE MATRICES
        IRKSIZ = M3MAX*M3MAX*2
        OPEN(UNIT=IDATA,DEVICE='SCR',ACCESS='RANDOM',MODE='BINARY',
    X        RECORD SIZE=IRKSIZ,FILE='FILTER.DE2')
C
C       LOOP FOR EACH VALUE OF S
        DO 10000 IS=1,NS
                        TYPE 9501,IS
9501                    FORMAT(' S VALUE IS: ',I4)
        SMID = ( S(IS-1)+S(IS) ) / 2.
        SS = SMID
C
C       LOOP FOR EACH INTERIOR LINE, CALCULATING THE
C       MATRICES B(N), -ALFA(N) AND THE INVERSE. STORE THE INVERSE.
```

```
C       ALFA(1) IS IDENTICALLY ZERO SINCE THE BOTTOM B.C.
C       DOES NOT INVOLVE ANY DERIVATIVES
        CALL ZEROKA(QMAT1,MAXM3,MAXM3)
C
        DO 650 N=2,NMAXM1
          NCOPY = N
           TYPE *,N
          NEVEN = (N/2)*2
C
C         GET THE MATRIX B
          CALL ZEROKA(B,MAXM3,MAXM3)
          CSN = COSN(N)
          DO 610 M=1,MMAX
            M3 = 3*(M-1)+1
            M2 = 2*M-1
            IF(NEVEN.EQ.N) M2 = 2*M
            COSZ = COSM(M2)*CSN
            CORR = COR(M,N)
C
            B(M3  ,M3  ) = SS
            IF(N.EQ.2.OR.N.EQ.NMAXM1   ) GO TO 601
            IF(M.EQ.1                  ) GO TO 601  !  B.C.
            IF(M.EQ.MMAX.AND.NEVEN.EQ.N) GO TO 601  !  B.C.
            B(M3  ,M3+1) = +RE*COSM(M2+1)/(COSZ*DELM)
            B(M3  ,M3-2) = -RE*COSM(M2-1)/(COSZ*DELM)
C
601         B(M3+1,M3+1) = SS
            IF(N.EQ.2.OR.N.EQ.NMAXM1   ) GO TO 602

            IF(M.EQ.1   .AND.NEVEN.NE.N) GO TO 602  !  B.C.
            IF(M.EQ.MMAX               ) GO TO 602  !  B.C.
            B(M3+1,M3  ) = -1./(CSN*DELM)
            B(M3+1,M3+2) = -CORR
            IF(M.EQ.MMAX) GO TO 602  !  BRING BOUNDARY TERM TO RHS
            B(M3+1,M3+3) =  1./(CSN*DELM)
C
602         B(M3+2,M3+2) = SS
            IF(N.EQ.2.OR.N.EQ.NMAXM1   ) GO TO 603
            IF(M.EQ.1   .AND.NEVEN.NE.N) GO TO 603  !  B.C.
            IF(M.EQ.MMAX               ) GO TO 603  !  B.C.
            B(M3+2,M3+1) =  CORR
C
603         CONTINUE
610       CONTINUE
C
C         GET THE INVERSE MATRIX M(N) AND STORE IT
C *** *** ***   BEWARE OF HEAVY COMPUTATION   *** *** *** *** *** ***
C         LEFT-MULTIPLY -ALFA(N-1) BY A(N)
          CALL KMTXSL(A(1,N),QMAT1,QMAT2,MMAX,M3MAX,NCOPY,NMAX,SS)
C         TAKE AWAY FROM MATRIX B(N)
          CALL KMTSUB(B,QMAT2,QMAT1,M3MAX,M3MAX)
C         INVERT TO GET THE MATRIX M(N)
          CALL KMTINV(QMAT1,QMAT2,M3MAX,M3MAX)
C         WRITE OUT M(N) TO DISK FOR USE LATER
          IRECNO = (IS-1)*(NMAX-2) + (N-1)
          WRITE(IDATA'IRECNO) QMAT2
C         RIGHT-MULTIPLY M(N) BY C(N) TO GET -ALFA(N)
          CALL KMTXSR(C(1,N),QMAT2,QMAT1,MMAX,M3MAX,NCOPY,NMAX,SS)
C *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
C
650     CONTINUE
C
10000   CONTINUE
C
C       CLOSE THE FILE WITH INVERSE MATRICES
        CLOSE(UNIT=IDATA,DISPOSE='SAVE')
C
        RETURN
        END
C*****************************************************************
```

References

Baer, F., 1977: Adjustment of initial conditions required to supress gravity oscillations in nonlinear flows. *Beitr. Phys. Atmos.*, **50**, 350-366.

Bellish, B., 1979: Comparison of some nonlinear initialization techniques. *Preprints of the fourth conference on numerical weather prediction*, Amercian Meteorological Society, Boston, pp9-12.

Bates, J.R. and A. McDonald, 1982: Multiply-upstream semi-Lagrangian advective schemes: analysis and application to a multi-level primitive equation model. *Mon. Weather Rev.*, **110**, 1831-1842.

Daley, R., 1981: Normal mode initialization. *Rev. Geophys. Space Phys.*, **19**, 450-468.

Halmos, P.R., 1958: *Finite-Dimensional Vector Spaces*. Van Nostrand, Princeton, New Jersey. 200 pp.

Kasahara, A. and K. Puri, 1981: Spectral representation of three dimensional data by expansion in normal mode functions. *Mon. Weather Rev.*,**109**, 37-51.

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.

Lynch, P., 1984: Initialization using Laplace transforms. Tech. Note No. 45, Irish Meteorological Service, Dublin.

Machenhauer, B., 1977: On the dynamics of gravity oscillations in a shallow water model, with applications to normal mode initialization. *Beitr. Phys. Atmos.*, **50**, 253-271.

Mesinger, F. and A. Arakawa, 1976: Numerical methods used in atmospheric models. WMO, GARP Publication Series, No 17., Vol 1.

Sadourny, R., 1975: The dynamics of finite-difference models of the shallow water equations. *J. Atmos. Sci.*, **32**, 680-689.