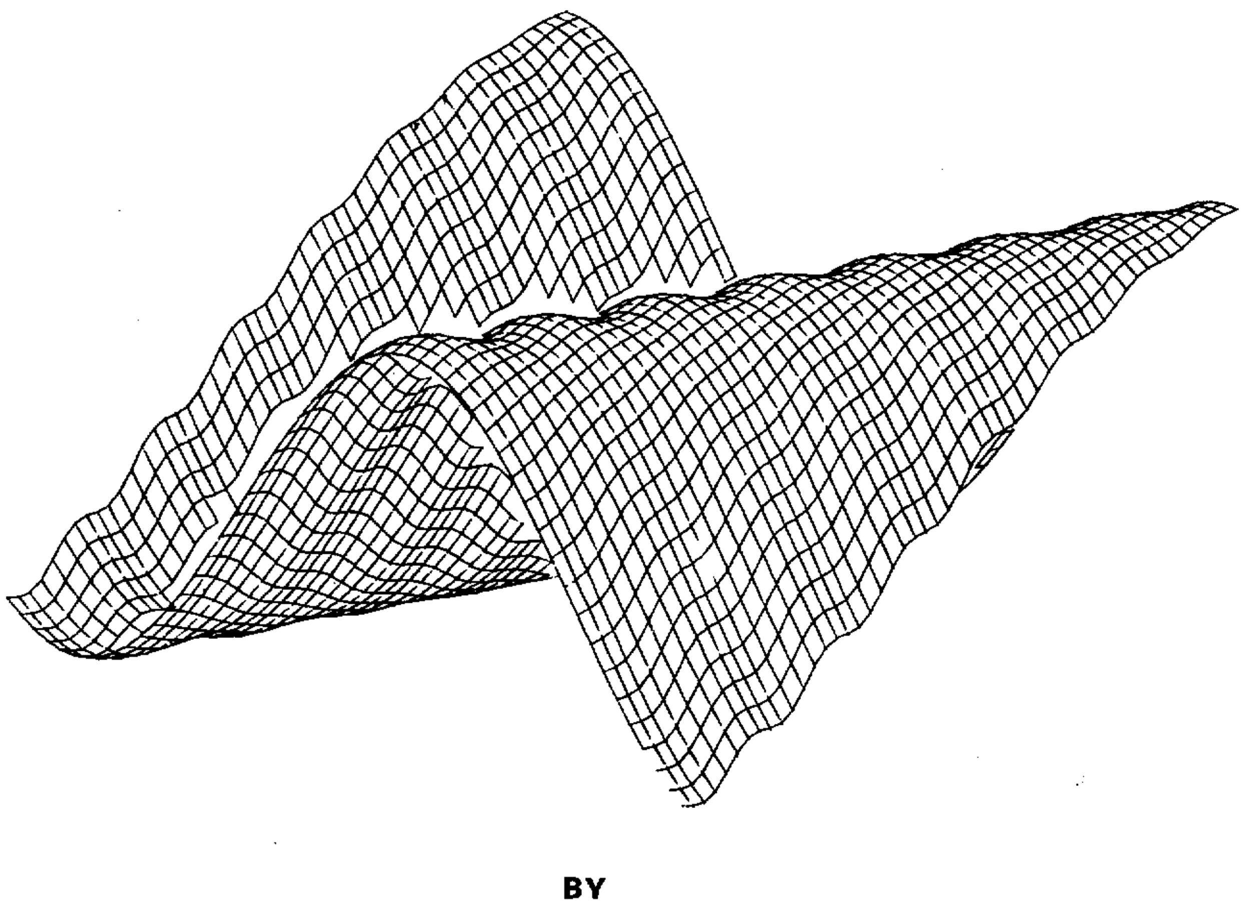
METEOROLOGICAL SERVICE

TECHNICAL NOTE No. 44

# DYNAMO

## A ONE-DIMENSIONAL PRIMITIVE EQUATION MODEL



BY

PETER LYNCH, M.Sc., Ph. D.

U.D.C. 551. 511.3:
681.3

GLASNEVIN HILL, DUBLIN 9
JANUARY 1984

Price; £2·20

# DYNAMO

*Abstract.*

A one-dimensional primitive equation model has been devised and programmed. Despite its great simplicity, it is capable of simulating several phenomena of importance in atmospheric dynamics, and should prove useful as a pedagogic aid and in meteorological research.

In this report the basic model equations are derived, their linear normal-mode solutions are investigated and their energetics are studied. The numerical formulation of the model is described, and the computer implementation is outlined. A few simple model runs are discussed, and suggestions for several other applications are offered.

## 1. INTRODUCTION

This note describes a simple numerical model which may be used to study the large-scale motions of the atmosphere. The model was originally designed to test initialization schemes, but it should have quite general applicability as a research tool and as a teaching aid: it may be used to simulate simple atmospheric flows; to investigate the structure and energetics of linear normal modes; to demonstrate the phenomenon of computational instability; to test various timestepping schemes, and other finite difference schemes. The model may easily be modified to filter gravity waves. Other effects such as orographic forcing can easily be incorporated. There is a possibility for growth of eddy motions in the presence of suitable mean flows (hydrodynamic instability), or of flow of energy back and forth between mean flow and eddies (vacillation).

The model is based on the primitive equations for an incompressible fluid in hydrostatic balance, *i.e.* the shallow water equations. The momentum equations are differentiated to form vorticity and divergence equations: this makes the $\beta$-effect explicit and all further latitudinal dependence can be supressed. Thus the model is one-dimensional in space. All spherical terms are neglected. There results a set of three prognostic equations for the vorticity, divergence and geopotential. After each timestep the horizontal velocity components can be calculated by solving two Poissón equations for the stream-function and velocity potential. The model has no external forcing, *i.e.* the bottom boundary is assumed to be flat.

The normal mode solutions of the model consist of rapidly travelling inertia-gravity waves, which move in both directions, and slow Rossby waves which move only westward (relative to the mean flow).

Since the equations are non-linear they cannot in general be solved analytically. They are expressed in terms of finite differences on a discrete grid and the resulting algebraic system is solved numerically. The boundary conditions are specified by assuming spatial periodicity for all dependent variables. The spatial domain is staggered, with different variables being evaluated at different points. Values not available directly are obtained by averaging. An Adams-Bashforth timestepping scheme is used, but this can easily be changed, *e.g.* to a leapfrog scheme.

The kinetic and available potential energy, as well as various other diagnostics, are calculated at each timestep; the total eddy energy is conserved in the absence of a mean flow; a non-vanishing mean flow *may* provide a source of energy for the growth of the eddy motions or for periodic exchange of energy between mean flow and eddies.

## 2. DERIVATION OF THE EQUATIONS

Since the Shallow Water Equations are derived in Pedlosky (1979), and discussed at length there, they will be set down here without further ado. For a shallow rotating layer of homogeneous incompressible and inviscid fluid above a plane and acted upon by gravity they take the form

$$\frac{du}{dt} - fv + \frac{\partial \Phi}{\partial x} = 0 \qquad (1)$$

$$\frac{dv}{dt} + f u + \frac{\partial \Phi}{\partial y} \quad = \quad 0 \tag{2}$$

$$\frac{d\Phi}{dt} + \Phi(u_x+v_y) \quad = \quad 0 \tag{3}$$

Here x and y are eastward and northward coordinates, u and v are the corresponding velocities, t is time, $\Phi = gh$ is the geopotential, where h is the depth of fluid above a flat surface, $f = f_0 + \beta y$ is the Coriolis parameter and $f_0$ and $\beta$ are assumed constant.

In order to eliminate the y-dependence while still retaining the $\beta$-effect, we derive vorticity and divergence equations by combining derivitives of the momentum equations. The zonally averaged flow is assumed to be in geostrophic balance:

$$f\bar{u} \quad = \quad -\bar{\Phi}_y \tag{4}$$

where $\bar{u}$ is taken as constant. We express the total flow as

$$u = \bar{u} + u'(x,t) \quad ; \quad v = v'(x,t) \quad ; \quad \Phi = \bar{\Phi}(y) + \Phi'(x,t)$$

where we note that all quantities other than $\bar{\Phi}$ are assumed to be independent of y. After subtracting the mean flow (4) from (2) the vorticity and divergence equations are derived by forming the combinations $((2)_x-(1)_y)$ and $((1)_x+(2)_y)$ respectively. The resulting equations can be written

$$\zeta_t + (u\zeta)_x + f\delta + \beta v \quad = \quad 0 \tag{5}$$

$$\delta_t + (u\delta)_x - f\zeta + \beta u' + \Phi_{xx} \quad = \quad 0 \tag{6}$$

where u' is the deviation from the mean zonal flow and the vorticity and divergence are given by the expressions

$$\zeta \quad = \quad v_x \quad ; \quad \delta \quad = \quad u_x.$$

Using (4) the continuity equation (3) can be written in the form

$$\Phi_t + (u\Phi)_x - f\bar{u}v + \bar{\Phi}\delta \quad = \quad 0 \tag{7}$$

The equations (5), (6) and (7) are the basic equations of the model. They form a set of three equations for the three independent variables vorticity, divergence and geopotential, with two independent variables, x and t. The only y-dependence is the parametric dependence of $f$ and $\bar{\Phi}$ on y, and scaling arguments can be used to show that this is small so that $f$ and $\bar{\Phi}$ may be assumed to be constant where they appear undifferentiated.

## 3. LINEAR NORMAL MODES

To investigate the simple types of wave-motion supported by the above system the equations are linearized about a state of rest and the perturbation quantities are assumed to be harmonic in x and t:

$$\begin{bmatrix} u' \\ v' \\ \Phi' \end{bmatrix} = \begin{bmatrix} \hat{u} \\ \hat{v} \\ \hat{\Phi} \end{bmatrix} \exp[ik(x-ct)]$$

Then (5), (6), and (7) become three homogeneous equations for the amplitudes $(\hat{u}, \hat{v}, \hat{\Phi})$. The condition for a non-trivial solution is that the system determinant should vanish. This gives a cubic equation for the phase-speed:

$$c(c+\beta/k^2)^2 - c(\bar{\Phi}+(f/k)^2) - (\beta/k^2)\bar{\Phi} = 0 \qquad (8)$$

The three roots are estimated by making simple assumptions about the magnitude of the phase-speed. These assumptions can then be justified *a posteriori*.

If $|c|$ is small the cubic term is neglected, giving

$$c = c_R \equiv -(\beta/k^2)/[1+f^2/k^2\bar{\Phi}]$$

This is the Rossby wave phase-speed. Equations (5) and (6) then tell us that this solution is in approximate geostrophic balance for v and that u is much smaller than v, *i.e.* the wave is quasi-nondivergent. The Rossby waves always travel westward relative to the mean flow.

If we assume that $|c| \gg |c_R|$ the constant term in (8) is negligible and we get the two roots:

$$c = \pm \sqrt{(\bar{\Phi} + f^2/k^2)}$$

which are the phase-speeds of the gravity-inertia waves. The gravity waves travel in both directions with relatively large phase-speeds. They are divergent motions and typically have fairly small vorticity.

The *quasi-geostrophic* shallow water equations are derived in Appendix A; these equations filter out the rapid gravity-inertia waves and allow only the slow rotational modes.

It is worth noting that if either u or v vanishes identically then the present model has no non-trivial linear solutions; thus, none of the normal modes are *purely* non-divergent or *purely* irrotational.

## 4. ENERGY CONSIDERATIONS

We consider the energy in a column of fluid of unit cross-section. The potential energy in the column is

$$\int_0^h \rho g z \, dz = \tfrac{1}{2}\rho g h^2 = \frac{1}{2g}\rho\Phi^2$$

where h(x,t) is the total depth of the fluid. When the fluid surface is perfectly flat, $h(x,t) \equiv \bar{h}$, constant, the system is in a state of minimum potential energy. Using this depth as the reference value, we define the *available potential energy* as

$$\int_{\bar{h}}^h \rho g (z-\bar{h}) \, dz = \tfrac{1}{2}\rho g (h-\bar{h})^2 = \frac{1}{2g}\rho\Phi'^2 \qquad (9)$$

This gives us a measure of the potential energy in the column which is available for conversion into kinetic energy.

The kinetic energy in the column can be partitioned into contributions due to the mean flow and to the eddies. The eddy kinetic energy is:

$$\int_0^h \tfrac{1}{2}(u'^2+v^2)\rho\, dz \;=\; \tfrac{1}{2}(u'^2+v^2)\rho h \;=\; \frac{1}{2g}\,\rho(u'^2+v^2)\Phi. \qquad (10)$$

We may note that this depends upon the total depth, whereas the available potential energy (9) depends only on the deviation from mean depth.

Energy equations are derived in the usual manner: equation (1) is multiplied by $\rho\Phi u'$. (2) by $\rho\Phi v$ and (3) by $\rho\Phi'$; they are then added together and integrated with respect to $x$. After some algebra we arrive at the equation for the energy budget of the eddy motion:

$$\frac{d}{dt}\int \left[\tfrac{1}{2}\rho(u'^2+v^2)\Phi+\tfrac{1}{2}\rho\Phi'^2\right] dx \;=\; -\int \left[\rho v[\tfrac{1}{2}(u'^2+v^2)+\Phi']\frac{\partial\bar{\Phi}}{\partial y}\right] dx \quad (11)$$

The left hand side is the temporal rate-of-change of the eddy kinetic plus available potential energy; the right hand side represents the conversion from mean flow energy to eddy energy; clearly, if the mean flow vanishes ($\bar{u}=\bar{\Phi}_y=0$) the total eddy energy remains constant.

In the present, one dimensional, model the eddy kinetic energy can be split into contributions due to the rotational and divergent motions as follows:

$$K = K_\psi+K_\chi \;;\; K_\psi = \tfrac{1}{2}\rho(\nabla\psi)^2\Phi = \tfrac{1}{2}\rho v^2\Phi \;;\; K_\chi = \tfrac{1}{2}\rho(\nabla\chi)^2\Phi = \tfrac{1}{2}\rho u'^2\Phi.$$

The values of these, and various other, energy quantities are calculated at each timestep by the procedure ENERGY. Their evolution can give us valuable information about the dynamics of the motion being considered.

Note that equation (11) allows the *possibility* for growth of eddy energy with time, and this suggests that the mean flow may be unstable to small perturbations. You may wish to consider the linear normal modes in the presence of a mean flow to see if there are circumstances in which their phase speeds may become complex. No further discussion will be given here.

Another possibility is that energy may oscillate back and forth between the mean flow and the eddies, leading to a vacillating regime (see Holton and Mass, 1976). It is probable that a proper treatment of this phenomenon would require an extension of the present model to simulate the energetics of the mean flow; but such an extension would not be too difficult.

## 5. NONDIMENSIONALIZATION

In order to clarify the relative magnitude of the various terms in the equations of motion it is convenient to nondimensionalize the equations by defining characteristic scales for length, time and velocity. It is also convenient numerically to have the principal terms of order unity. Scale analysis is discussed in Holton (1972) and Haltiner and Williams (1980), so the treatment here will be brief. We introduce length and velocity scales L and V and scale time by $f^{-1}$ (alternatively we could use the advective time-scale (L/V)). The geopotential is scaled by $f$LV (suggested by the geostrophic relationship; we could have used $V^2$ or $(fL)^2$). Various nondimensional combinations *pop up* when we scale the equations: we define

$$Ro \equiv (V/fL) \; ; \; R_\beta \equiv (\beta L/f) \sim (L/a) \; ; \; R_F \equiv \bar{\Phi}/(fL)^2 \equiv (L_R/L)^2 .$$

Here $Ro$ is the Rossby number; $R_\beta$ is a measure of the importance of the $\beta$-effect, determined by the scale of the motion; $R_F$ is the reciprocal of the Froude number, and relates the length scale of the motion to the *Rossby radius of deformation*, $L_R \equiv \sqrt{\bar{\Phi}}/f$.

The equations of motion, (5), (6) and (7), may now be written in nondimensional form

$$\zeta_t + Ro(u\zeta)_x + \delta + R_\beta v = 0 \tag{13}$$

$$\delta_t + Ro(u\delta)_x - \zeta + R_\beta u' + \Phi_{xx} = 0 \tag{14}$$

$$\Phi_t + Ro(u\Phi)_x - Rou_0 v + R_F\delta = 0 \tag{15}$$

Note that if an advective timescale were chosen instead of $f^{-1}$, the time derivatives in these equations would be multiplied by $Ro$. Such a choice is made for deriving the quasi-geostrophic approximation to the above set of equations (see Appendix A).

## 6. NUMERICAL FORMULATION

The three equations (13), (14) and (15) provide a means for predicting $\zeta$, $\delta$ and $\Phi$, given their values at an initial time. Since these equations are nonlinear (and the nonlinear advection process plays a crucial rôle in atmospheric dynamics) they must be solved numerically. If the derivatives are approximated by finite differences in space and time the differential system is replaced by an algebraic system. The dependent variables are specified at points on a discrete grid in space and at isolated instants in time. From the values at (and prior to) a particular instant, t, the algebraic equations are used to predict their values at the next instant, $t+\Delta t$. This process is repeated until the required forecast length is reached.

The initial values normally involve specification of u, v and $\Phi$. The initial values of $\zeta$ and $\delta$ are obtained by finite differencing of the velocities. The equations are then used to step forward $\Delta t$. This gives us updated values for $\zeta$, $\delta$ and $\Phi$. The new velocities must be retrieved by solving for the velocity potential and stream- function:

$$\mathbf{V} = \nabla\chi + \mathbf{k}\times\nabla\psi \ ; \ \nabla^2\chi = \delta \ ; \ \nabla^2\psi = \zeta .$$

In the present, one-dimensional case we solve the equations

$$\chi_{xx} = \delta \quad ; \quad \psi_{xx} = \zeta \tag{16}$$

with periodic boundary conditions, and derive the velocities from

$$u = \chi_x \quad ; \quad v = \psi_x \tag{17}$$

This must be done at every timestep, since the velocities appear explicitly in the equations and are needed to perform the next timestep. The 1-D "Poisson" equations (16) are solved by a simple method described in Appendix B, and the velocities are obtained immediately from (17) by finite differencing.

The relationship between the velocities (u,v) and the prognostic variables ($\zeta$,$\delta$) suggests that we specify them at alternate points of a grid staggered in space. The velocities are specified at "half-points" and the vorticity, divergence and geopotential at "whole-points":

$$\cdots \quad u_{m-\frac{1}{2}}, v_{m-\frac{1}{2}} \qquad \zeta_m, \delta_m, \Phi_m \qquad u_{m+\frac{1}{2}}, v_{m+\frac{1}{2}} \qquad \zeta_{m+1}, \delta_{m+1}, \Phi_{m+1} \quad \cdots$$

$$\text{———}\times\text{———}\theta\text{———}\times\text{———}\theta\text{———}$$

$$m-\tfrac{1}{2} \qquad\qquad m \qquad\qquad m+\tfrac{1}{2} \qquad\qquad m+1$$

Velocities at whole points or $\zeta$, $\delta$, $\Phi$ at half points are obtained by averaging. We define some finite difference operators:

$$(q_m)_x = (q_{m+\frac{1}{2}} - q_{m-\frac{1}{2}})/\Delta x \quad ; \quad (\overline{q_m}) = \tfrac{1}{2}(q_{m-\frac{1}{2}} + q_{m+\frac{1}{2}})$$

Applying these operators successively we find that

$$(\overline{q_m})_x = (q_{m+1} - q_{m-1})/2\Delta x \quad ; \quad (q_m)_{xx} = (q_{m+1} - 2q_m + q_{m-1})/(\Delta x)^2$$

These forms are sufficient to approximate the derivatives on the staggered grid.

It is obvious in most cases how the finite differencing and averaging operators should be applied to approximate terms in the equations. However, in the case of the advection terms several possibilities present themselves; we choose the simplest form:

$$\left.\frac{\partial(uq)}{\partial x}\right|_m \longrightarrow \frac{u_{m+\frac{1}{2}} \cdot \frac{1}{2}(q_m + q_{m+1}) - u_{m-\frac{1}{2}} \cdot \frac{1}{2}(q_{m-1} + q_m)}{\Delta X} = \left.(u\bar{q})_x\right|_m$$

Other possibilities include using a double interval, or splitting up the derivative before differencing; the more complicated forms may have the advantage of *numerically* preserving various conservation properties of the continuous equations.

The spatially differenced equations may now be written in the form:

$$\zeta_t = - \left( Ro(u\bar{\zeta})_x + \delta + R_\beta \bar{v} \right) \tag{18}$$

$$\delta_t = - \left( Ro(u\bar{\delta})_x - \zeta + R_\beta \bar{u}' + \Phi_{xx} \right) \tag{19}$$

$$\Phi_t = - \left( Ro(u\bar{\Phi})_x - Ro u_0 \bar{v} + R_F \delta \right) \tag{20}$$

The time-differencing is done by an Adams-Bashforth scheme (Mesinger and Arakawa, 1976, [MA]). For the simple equation

$$dY/dt = F(Y,t)$$

the values of Y at the time-levels n and n+1 are related (exactly) by:

$$Y^{n+1} = Y^n + \int_{n\Delta t}^{(n+1)\Delta t} F(Y,t)\cdot dt$$

In the Adams-Bashforth scheme we approximate F(Y,t) by a value at the centre of the interval $\Delta t$ obtained by linear extrapolation using the known values $F^{n-1}$ and $F^n$. This gives

$$Y^{n+1} = Y^n + \Delta t\left(\frac{3}{2}F^n - \frac{1}{2}F^{n-1}\right).$$

The properties of the scheme are discussed in [MA]. It is of second order accuracy and has a computational mode which is damped. The amplification of the physical mode is $[1+p^4]$ (where $p = c_{max}\Delta t/\Delta x$) which implies marginal instability. This satisfies the Von Neumann necessary condition for boundedness of the solution for finite t, and experience shows that as long as $\Delta t$ is chosen sufficiently small the amplification is insignificant. Since the initial conditions refer to a single time we must begin the integration with a two level scheme; therefore, the first timestep is performed using an Euler forward scheme.

You may wish to experiment with other timestepping schemes. The leapfrog scheme is stable for $p < 1$ but its computational mode is neutral rather than damped; the trapezoidal scheme looks ideal (see [MA], figure 2·1) but it is implicit; there are numerous other options.

## 7. IMPLEMENTATION

A brief overview of the computer program which implements the model is given here. This section should be read in conjunction with the program listing in Appendix C, where some more details are given in comments within the code. Copies of the source code on disk are available on request.

The main program is called DYNAMO. The source version (in FORTRAN) is in the file DYNAMO.FOR; global variables are specified in the COMMON blocks in DYNAMO.COM; control parameters are read from DYNAMO.CDS and output goes to the file DYNAMO.LPT.

The main program contains calls to a number of routines whose purpose or function is described briefly here:

| | |
|---|---|
| GETPAR | Reads control cards and defines various constants |
| ICS | Sets up the initial fields for the run |
| LAPIN | Initializes the fields (not implemented) |
| STEPON | Performs a single timestep |
| OUTPUT | Prints out the final fields and various diagnostics. |

Various other subroutines are called; their purposes are given here:

| | |
|---|---|
| ENERGY | Calculation of various energy integrals |
| POIS1D | Solution of 1-D Poisson equations with periodic B.Cs. |
| DDXBAR, DDXF, DDXB, DDXX | Calculation of finite differences |
| XMEANF, XMEANB | Averaging operators |
| ENDS | Fill in end-values of a periodic array |
| MEMOVE | Move fields around in core |
| PLOTLN | Draw graphs on the lineprinter. |
| HOVMOL | Plot a zebra chart on the lineprinter |

The meanings of the more important variables and arrays are given below:

(0) MAXIMUM DIMENSIONS FOR ARRAYS

PARAMETER NPX=201,NPT=4001 Space and time array sizes.

(1) PARAMETERS CONTROLING THE FLOW OF CALCULATIONS

| | | |
|---|---|---|
| LINEAR | .TRUE. | Ignore nonlinear terms |
| IQG | .TRUE. | Use quasi-geostrophic equations (not implemented) |
| INIT | .TRUE. | Initialize the fields (not implemented) |
| IPRINT | .TRUE. | Print out various diagnostics. |
| NPRINT | Number of timesteps between printouts | |
| ICNUM | Indicator for the initial conditions. | |
| IHOV, NXHOV, NTHOV | Control for Hovmöller diagrams. | |

(2) VARIOUS CONSTANTS, PARAMETERS AND SCALES

PI=3.14159265

FCOR = 1.E-04 (Coriolis parameter) BETA = 1.E-11 (Beta parameter)

UBAR Mean Zonal Wind UO Nondimensionalized UBAR

FIBAR Mean Geopotential FIO Nondimensionalized FIBAR

RF,RO,RB Nondimensional numbers (Froude, Rossby, Beta; see text)

GRAV = 9.81 Gravitational acceleration

SXL,SXT,SXV,SXDV,SXFI Scales for length, time, velocity, vorticity (and divergence) and geopotential.

(3) INDEPENDENT VARIABLES, INCREMENTS, GRIDSPECS, ETC.

REAL X(0:NPX),T(0:NPT) Spatial and temporal independent variables (required for convenience in plotting results)

NX Number of points in the spatial domain NXP1 = NX + 1

NSTEPS Number of timesteps in run

DX $\Delta x$, Grid distance DT $\Delta t$, Timestep

(4) DEPENDENT VARIABLES

REAL U (0:NPX),V (0:NPX) Horizontal velocities

REAL FI(0:NPX),VORT(0:NPX),DIV(0:NPX) Geopotential, vorticity, divergence

REAL FIOLD (0:NPX),VOROLD(0:NPX),DIVOLD(0:NPX) Old values of $\Phi$, $\zeta$, $\delta$

(Old values may not be required but are included for convenience)

REAL PSI(0:NPX),CHI(0:NPX) Stream function, velocity potential

(5) RIGHT-HAND SIDES (AT TWO TIMES)

REAL RHS1V(0:NPX),RHS1D(0:NPX),RHS1C(0:NPX) New values

REAL RHS2V(0:NPX),RHS2D(0:NPX),RHS2C(0:NPX) Old values

(Old values may not be required but are included for convenience)

(6) ENERGY QUANTITIES

REAL KEROT(0:NPT),KEDIV(0:NPT),KETOT(0:NPT)

Eddy rotational, divergent and total kinetic energy

REAL APE(0:NPT),APLUSK(0:NPT)

Eddy available potential and total (APE+KE) energy

REAL SOURCE(0:NPT),DDTAPK(0:NPT)

Conversion from zonal mean, and rate-of-change of eddy energy.

(7) MID-POINT VALUES OF DEPENDENT VARIABLES (FOR PLOTTING)

REAL PTU(0:NPT),PTV(0:NPT),PTFI(0:NPT),PTVORT(0:NPT),PTDIV(0:NPT)

(8) WORKING SPACE

REAL WORK1(0:NPX),WORK2(0:NPX),WORK3(0:NPX),WORK4(0:NPX)

(9) ASCII STRING FOR INPUT FILE GUIDEWORD

DOUBLE PRECISION STRING ASCII string to describe control cards.

The array storage of dependent variables works as follows: The geopotential, vorticity and divergence at point N are stored in FI(N), VORT(N) and DIV(N). The velocity potential and stream function at these points are stored in CHI(N) and PSI(N). The velocities at N-½ are stored in U(N) and V(N). Thus, to get U from CHI we call DDXB, the *backward* difference; to get the average of U at *whole* points we call XMEANF, the *forward* average; care is needed to difference and average in the correct direction. All the dependent variables are periodic, with period NX: Thus, we can set U(NX) = U(0) and U(NXP1) = U(1).

## 8. APPLICATIONS

*Sample Output*

In this section we present selected output from a few trial runs, which have been chosen to illustrate some simple phenomena which can be simulated by the model. The input parameters in all cases are as follows: Gridpoints NX = 50; Gridlength $\Delta x = 200$ *km*; Channel length L = 10000 *km*; Timestep $\Delta t = 100$ sec.

Example (1): *A Rossby Wave*

The initial conditions for the first example are chosen as follows:

$$\Phi = \cos(2\pi x/L) \quad ; \quad u = 0 \quad ; \quad v = \Phi_x \qquad \text{(ICNUM=2)}$$

*i.e.*, there is a wavenumber one geopotential disturbance, and the wind is in geostrophic balance with it. The main component of this initial field (ICNUM=2) is a Rossby wave. There are also small gravity wave components, since the *pure* Rossby wave has a small but non-vanishing divergence whereas these initial

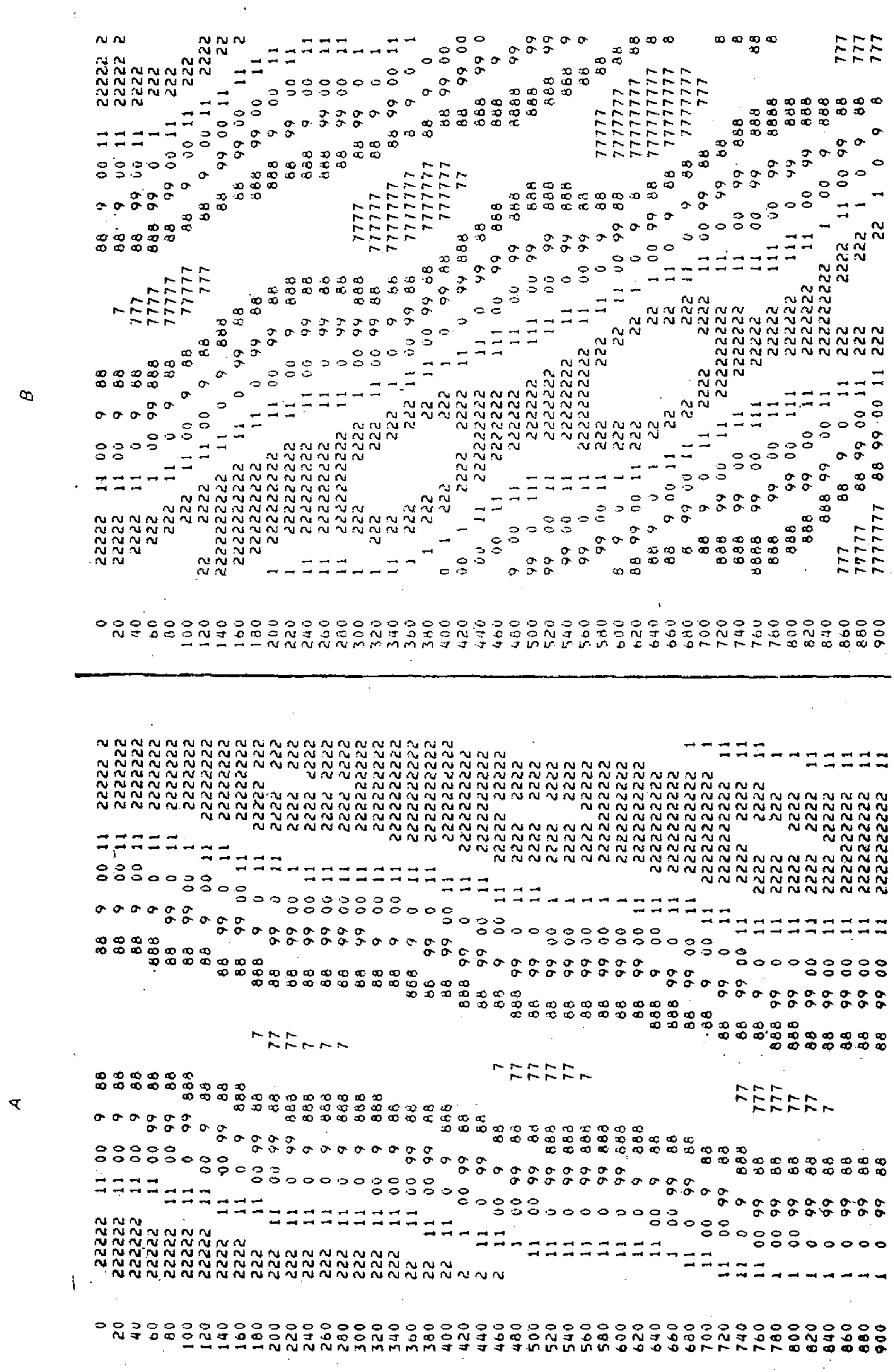



Figure 1. Hovmoeller diagram of geopotential. Horizontal axis $0 < X < 10,000$ KM.
Vertical axis: $0 < T < 900*100$ sec.
(a) Zero zonal flow (b) UBAR = 100 m/s.

conditions are nondivergent. In figure 1a we show a plot of height against x and t (a so-called Hovmöller diagram). The westward movement of troughs and ridges is clear. The corresponding solution for the same initial conditions but with a mean zonal wind $\bar{u} = 100\, m/s^{-1}$ is shown in figure 1b. The advective effect of the mean flow is clear. You may like to check the phase-speed of the solution against the theoretical Rossby phase speed (see the equation following (8): $f = 1.E\text{-}04\, s^{-1}$; $\beta = 1.E\text{-}11\, m^{-1}s^{-1}$; $k = 2\pi/L$, $L = 1.E\text{+}07$ $m$; $\bar{\Phi} = 1.E\text{+}05\, m^2 s^{-2}$). A three-dimensional plot of the Rossby wave solution (for $\bar{u} = 0$) is shown on the title page: note the high frequency ripples running along the ridge; these are due to the interference of the small amplitude gravity wave components present in the initial conditions.

Example (2): *Timescales of the Solutions*

The value of the geopotential at a central point of the grid, resulting from several different initial conditions, is plotted against time in figure 2. The initial conditions are:

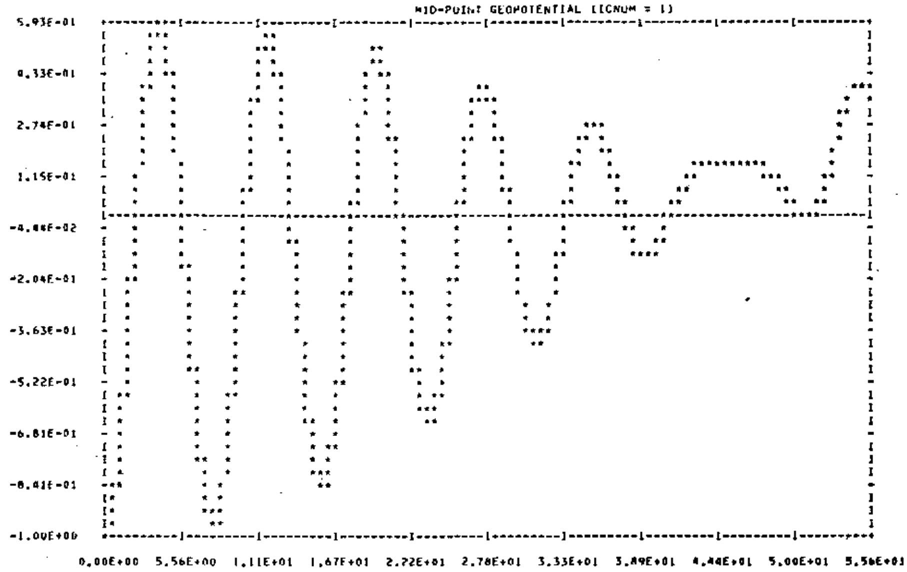
(2a) $\Phi = \cos(2\pi x/L)$ ; $u = 0$ ; $v = 0$ (ICNUM=1)

(2b) $\Phi = \cos(2\pi x/L)$ ; $u = 0$ ; $v = \Phi_x$ (ICNUM=2)

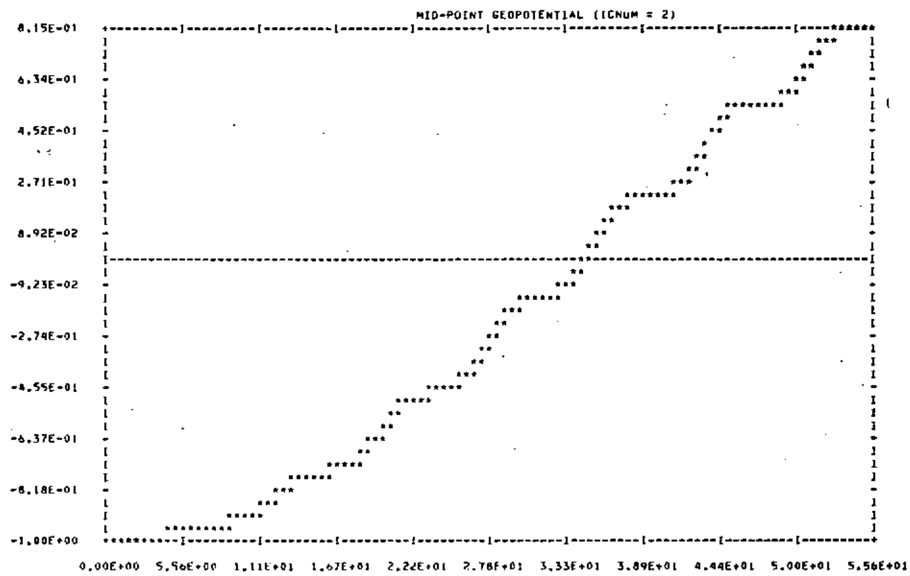
(2c) $\Phi = \cos(2\pi x/L)$ ; $u = \Phi$ ; $v = 0$ (ICNUM=3)

These initial conditions may be described as follows: (2a) represents a mixture of two gravity-inertia waves and a Rossby wave (no component is obviously dominant); (2b) is essentially a Rossby wave (with small G-I wave components); (2c) represents an eastward travelling gravity-inertia wave. The figure clearly shows the different timescales of the evolving geopotential for the differing types of motion. The rotational motion (2b) has a much slower evolution than the motion containing large gravity-wave components. It is the principal goal of the *initialization* process to remove the large, high frequency oscillations which arise from the presence of unrealistically large gravity-inertia components in the initial data used for numerical forecasts. (The changing amplitude of the geopotential, evident in figure 2a and 2c, is due to interference between different components; the total eddy energy of the disturbances remains constant, as we will see in the next example).

A




B




C

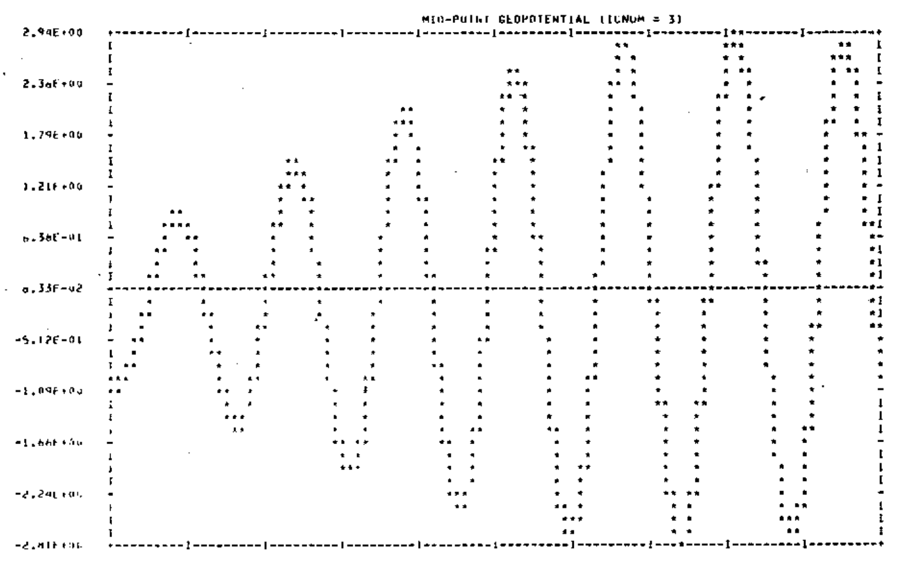



Figure 2. Time evolution of the geopotential at a central point
(a) ICNUM = 1 (b) ICNUM = 2 (c) ICNUM = 3

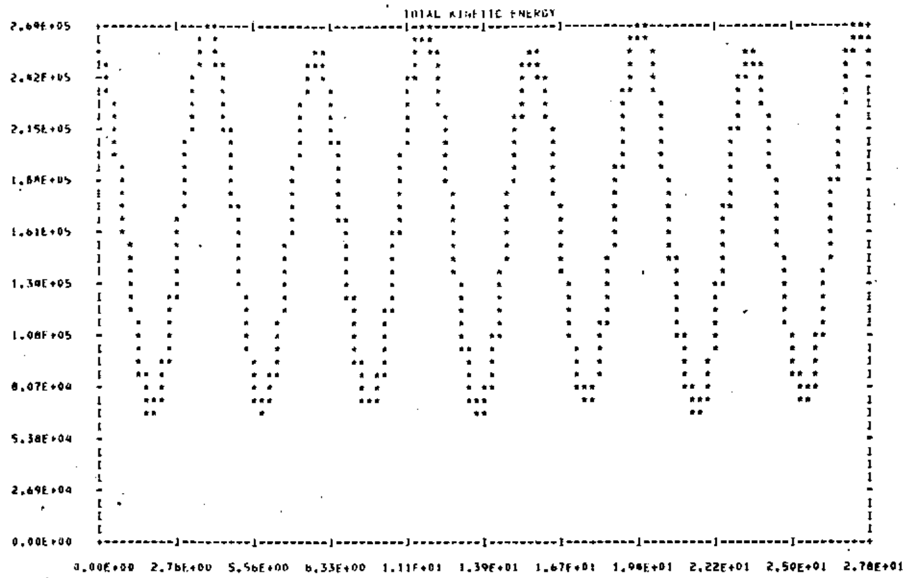
Example (3): *Conservation of Eddy Energy*

When the mean zonal wind vanishes, there is no physical source of energy which might enable a disturbance to grow with time: the total eddy energy remains constant (see equation (11)). We have not made any effort to ensure that the numerical scheme reflects this conservation property; however, if the model is to be of any use for simulating atmospheric phenomena, the energy budget must be properly represented. In this example we start from initial conditions (2c) above (ICNUM=3) and calculate the eddy kinetic, available potential and total energy at each timestep. Plots of these as functions of time are shown in figure 3. These clearly show how the energy may flow back and forth between the kinetic and potential forms, due to interaction between different wave components. Figure 3c demonstrates the conservation of total eddy energy. (N.B. The Adams Bashforth scheme is marginally unstable; if an extended integration is carried out the energy will eventually begin to grow in an unacceptable way; you may like to try this, and then rerun with a shorter timestep)

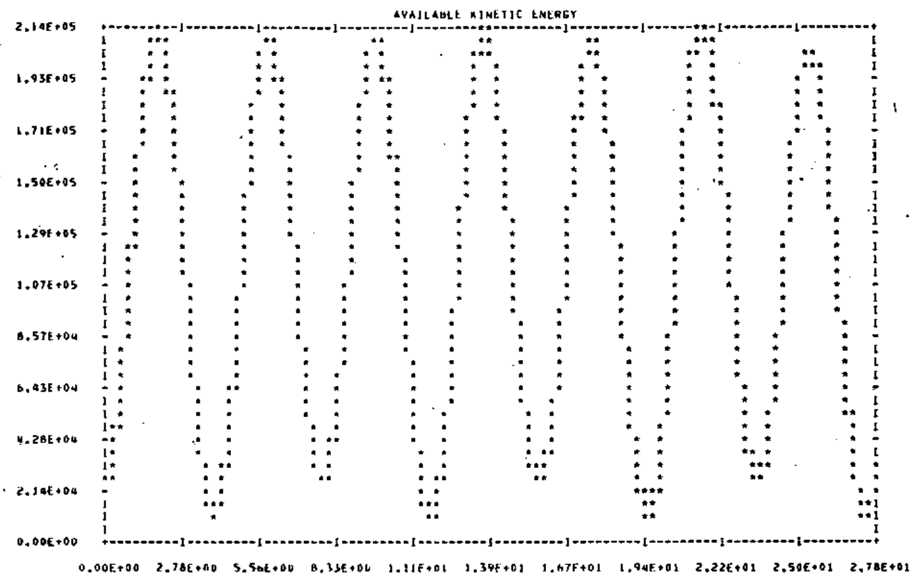
Example (4): *Wave - Mean Flow Interaction*

Only a very simple case is considered here: the zonal mean wind is taken as $\bar{u} = 100\,m\,s^{-1}$; the initial conditions are (2a) above (ICNUM=1) and the model is integrated for 4000 timesteps ($\Delta t = 100\,s$, so forecast length is about 4½ days). Equation (11) shows that the eddy energy may change in time when there is a non-vanishing zonal flow. The total eddy energy (KE+APE) is shown in figure 4a; we see that there is a quasi-periodic exchange of energy between the mean flow and the eddy motion, and that it has two timescales (it is reminiscent of the phenomenon of beats between waves with nearly equal frequencies). The right hand side of equation (11) is calculated at each timestep (it is called SOURCE in the code) and plotted in figure 4b; the time rate of change of the eddy energy is obtained by finite differencing of the values shown in figure 4a, and the result is plotted in figure 4c; it is in excellent agreement with the energy source values in figure 4b.

A




B




C

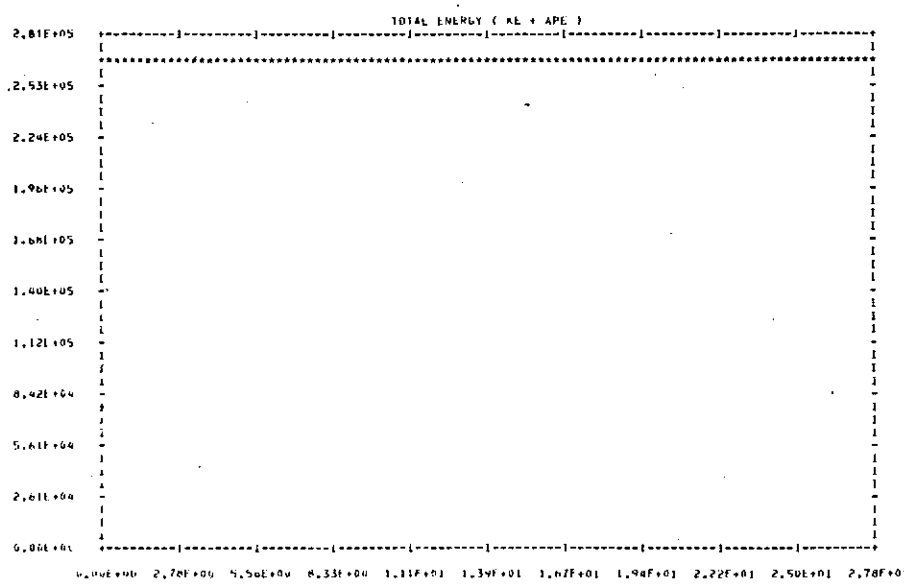



Figure 3. Eddy energy for the initial conditions ICNUM = 3
(a) Kinetic energy (b) Available potential energy
(c) Total eddy energy

A

B

C

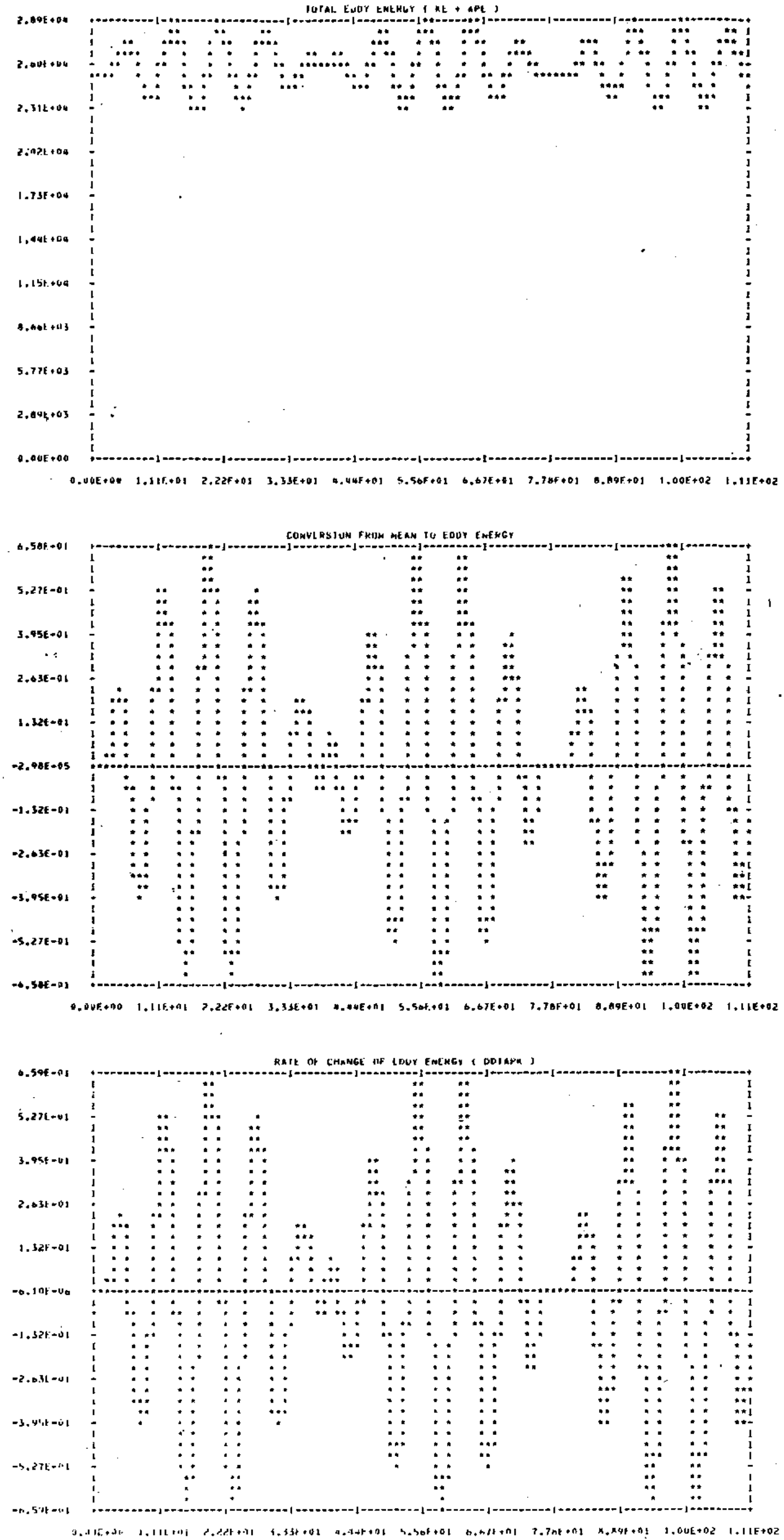



Figure 4. Eddy energy in the presence of a zonal flow (UBAR = 100 m/s)
(a) Total eddy energy (b) Source term in EQN (11)
(c) Time rate-of-change of total eddy energy

It would appear that this model is capable of simulating more complicated interactions between the mean flow and the eddy motions. No further consideration of this matter is presented here. More specifically, the question of hydrodynamic instability of the eddy motion has not been addressed, and is left for your consideration.

*Suggestions for Further Applications*

There follows a hastily assembled list of suggestions for making use of the model DYNAMO. I would be grateful to hear of your experiences with any new tests, or any bugs found, etc.

First some trivial runs to gain familiarity:

(1): Run the model with various initial conditions (ICNUM) to produce the Rossby and Gravity-inertia waves *before your very eyes*.

(2): Switch on IHOV to generate Hovmöller diagrams for these.

(3): Run with various values of $\Delta t$ to illustrate computational instability when the CFL criterion is violated.

(4): Run for extended period to illustrate the ultimate breakdown due to the marginal instability of the Adams-Bashforth scheme; *cure* this by reducing the timestep.

More Advanced Applications:

(1): Change the timestepping scheme, *e.g.* use Euler forward (unstable!), Euler backward *or* Matsuno, Leapfrog, Trapezoidal (implicit); these are all discussed in Mesinger and Arakawa (1976).

(2): Split the integration and use a semi-Lagrangian scheme for advection (Bates and McDonald, 1982).

(3) Perform extended runs (perhaps with leapfrog scheme) to check for non-linear instability; see if this can be *cured* by reformulating the finite differencing of the advection terms. Does non-linear instability occur with a semi-Lagrangian scheme?

(4): Modify the model to use either the primitive equations or the filtered equations (see Appendix A). Compare the handling of Rossby waves by the two methods.

(5): Derive the equation expressing *Conservation of Potential Vorticity*; calculate this quantity numerically and see if the model is conserving it properly.

(6): It is easy to incorporate *mountains* into the model. What sort of motion is forced by orography? How does it effect the energy balance of the eddy motion? Note the very different response for small and large mean flow.

(7): Extend the model to predict the mean flow. Calculate the energetics of the mean flow and investigate the phenomenon of vacillation (this is an essentially non-linear phenomenon, and there is no simple analytical description of it).

(8): Investigate the possibilities for hydrodynamic instability of the eddy motion in the presence of a non-zero zonal mean flow. What are the energetics of the instability? What (if any) is its geophysical relevance?

In doing any of the above experiments, do not be afraid to make even major changes to the model code. Clean copies of the original code are there for the asking (*e.g.* you can send me a request using the MAIL facility on the DEC 20-50).

======== 00000000 ========

## APPENDIX A: THE FILTERED EQUATIONS.

The quasi-geostrophic approximation to the shallow water equations is derived here (for more details see Haltiner and Williams, chapter 3). Equations (5), (6) and (7) are nondimensionalized as in section 5 but with an advective timescale. They take a form similar to (13), (14) and (15) but with the time derivatives multiplied by *Ro*. Since the divergence is small compared to the vorticity we separate the wind into rotational and divergent parts

$$\mathbf{V} = \mathbf{V}_\psi + \mathbf{V}_\chi \quad ; \quad \mathbf{V}_\psi = \mathbf{k} \times \nabla\psi \quad ; \quad \mathbf{V}_\chi = \nabla\chi$$

and ignore the latter where it appears undifferentiated. We now assume that $Ro \ll 1$, $R_\beta \lesssim 1$ and $R_f \sim 1$, and drop all terms which are of order *Ro* or smaller. The resulting equations may be written (in dimensional form):

$$\frac{\partial\zeta}{\partial t} + \bar{u}\frac{\partial\zeta}{\partial x} + \beta v + f\delta = 0 \qquad \text{(A1)}$$

$$-f\zeta + \Phi_{xx} = 0 \qquad \text{(A2)}$$

$$\frac{\partial\Phi}{\partial t} + \bar{u}\frac{\partial\Phi}{\partial x} - f\bar{u}v + \bar{\Phi}\delta = 0 \qquad \text{(A3)}$$

Note that the divergence equation has become a diagnostic equation (*i.e.* it has no time derivative); it shows that the vorticity is geostrophic and gives the rotational wind:

$$\psi = \Phi'/f \quad ; \quad v = \psi_x = \Phi'_x/f \qquad \text{(A4)}$$

Equations (A1), (A2) and (A3) are the quasi-geostrophic equations for our one-dimensional model; although $|\delta| \ll |\zeta|$ the divergence terms in (A1) and (A3) are of the same magnitude as the other terms.

The equations (A1) and (A3) can be used to forecast the vorticity and geopotential. Alternatively, we can eliminate $\delta$ between them and get the quasi-geostrophic potential vorticity equation :

$$\left(\frac{\partial}{\partial t} + \bar{u}\frac{\partial}{\partial x}\right)\left[\zeta - (f/\bar{\Phi})\Phi'\right] + v\bar{\Phi}\frac{\partial}{\partial y}\left[f/\bar{\Phi}\right] = 0 \qquad \text{(A5)}$$

With the use of (A2) and (A4) this can be written in terms of $\Phi'$ alone. A diagnostic equation for the divergence is obtained by eliminating the time derivatives between (A1) and (A3):

$$\delta_{xx} - (f^2/\bar{\Phi})\,\delta = (\beta/\bar{\Phi})\,\Phi_x + (\bar{u})\,\Phi_{xxx} \qquad \text{(A6)}$$

This is the *quasi-geostrophic divergence equation* and relates the divergence to the geopotential. It is a (1-D) Helmholtz-equation for $\delta$ when $\Phi$ is known.

The system (A1), (A3) and (A6) are sometimes called the *Filtered Equations:* they allow only the slow quasigeostrophic motion; the fast gravity inertia modes are filtered out; you may check this by deriving the linear normal mode solutions of these equations and comparing the results with those (in section 3) for the primitive equations. The model DYNAMO may easily be modified to use the filtered equations. The LOGICAL variable IOG should be used to switch from one system to the other. A routine HELM1D, analogous to POIS1D but for a Helmholtz equation, will have to be written. You could, for example, try a successive overrelaxation (SOR) method; let me know how you get on.

## APPENDIX B: SOLUTION OF POISSON'S EQUATION.

After each forward step the new values of the velocities $u$ and $v$ must be derived from the vorticity $\zeta$ and divergence $\delta$. If we solve two Poisson equations for the stream-function $\psi$ and velocity potential $\chi$:

$$\nabla^2\psi = \zeta \qquad\qquad \nabla^2\chi = \delta \tag{B1}$$

the velocities can be obtained by differentiation:

$$\mathbf{V} = \mathbf{k} \times \nabla\psi + \nabla\chi \tag{B2}$$

In the present case we assume that the dependent variables are independent of $y$, are specified on the discrete grid $\{x_0=0, x_1, x_2, \cdots, x_N=L\}$, and are periodic in $x$. Thus we must solve two equations of the form

$$d^2\phi/dx^2 = \rho \quad ; \quad \phi(0)=\phi(L) \tag{B3}$$

Since the reference potential is arbitrary we can choose $\phi(0)=0$. The discrete equations (taking $\Delta x=1$) can then be written

$$\begin{aligned}
-2\phi_1 + \phi_2 \qquad\qquad\qquad &= \rho_1 \\
\phi_1 - 2\phi_2 + \phi_3 \qquad\quad &= \rho_2 \\
\phi_2 - 2\phi_3 \qquad\quad &= \rho_3 \\
&\;\;\vdots \\
\phi_{N-2} - 2\phi_{N-1} &= \rho_{N-1} \\
\phi_1 \qquad\qquad + \phi_{N-1} &= \rho_N
\end{aligned} \tag{B4}$$

The first equation is multiplied by one, the second by two, etc., and the resulting equations added up to obtain

$$N\phi_1 = \sum_{n=1}^{N} n\rho_n$$

this gives us $\phi_1$; the first of (B4) then gives $\phi_2$, the second $\phi_3$, and so forth until the full solution is obtained. The value of $\phi_N$, which should be zero, can be used to check the effects of roundoff error. The entire algorithm is coded in the procedure POIS1D. When the stream-function and velocity potential have been derived the velocities on the staggered grid are obtained from the equations

$$u_n = (\chi_n - \chi_{n-1})/\Delta x \quad ; \quad v_n = (\psi_n - \psi_{n-1})/\Delta x$$

by calling the procedure DDXB. (The method of solution described here was found in Hockney and Eastwood, 1981.)

## *References*


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.

Haltiner, G.J. and R.T. Williams, 1980: Numerical Prediction and Dynamic Meteorology. John Wiley & Sons, New York, 477pp.

Hockney, R.W. and J.W Eastwood, 1981: Computer Simulation Using Particles. McGraw-Hill Inc., New York, 540pp.

Holton, J.R., 1972: An Introduction to Dynamic Meteorology. Academic Press, New York, 319pp.

Holton, J.R. and C. Mass, 1976: Stratospheric vacillation cycles. *J. Atmos. Sci.*, **33**, 2218-2225.

Mesinger, F. and A. Arakawa, 1976: Numerical Methods Used in Atmospheric Models. WMO, GARP Publication Series, No. 17., Vol. 1.

Pedlosky, J., 1979: *Geophysical Fluid Dynamics.* Springer-Verlag, New York, Heidelberg, Berlin. 624pp.

```
      PROGRAM DYNAMO
C**********************************************************************
C*      SHALLOW WATER EQUATION MODEL - 1 DIMENSION.                   *
C*              " D Y N A M O "                                       *
C**********************************************************************
C*      STARTING FROM THE SHALLOW WATER EQUATIONS, THE                *
C*      VORTICITY AND DIVERGENCE EQUATIONS ARE DERIVED.               *
C*      THE BETA EFFECT THUS APPEARS EXPLICITLY AND SO                *
C*      ALL FURTHER Y-DEPENDENCE OF THE PERTURBATION                  *
C*      QUANTITIES CAN BE SUPRESSED. THE RESULT IS A SET              *
C*      OF THREE EQUATIONS WITH TWO INDEPENDENT VARIABLES,            *
C*      X AND T, AND THREE DEPENDENT VARIABLES, TO WIT,               *
C*              DV/DX = VORTICITY                                     *
C*              DU/DX = DIVERGENCE                                    *
C*              FI    = GEOPOTENTIAL.                                 *
C*      PERIODIC BOUNDARY CONDITIONS ARE SPECIFIED, I.E.              *
C*              Q(X+LX,T) = Q(X,T)                                    *
C*      FOR ALL PERTURBATION QUANTITIES Q.                            *
C*      INITIAL CONDITIONS ARE SET BY SPECIFYING THE VALUES OF        *
C*      U(X,0), V(X,0), FI(X,0). FROM THESE THE INITIAL VALUES        *
C*      OF VORTICITY AND DIVERGENCE ARE IMMEDIATELY DERIVED.          *
C*      AFTER EACH FORWARD STEP WE MUST SOLVE TWO POISSON             *
C*      EQUATIONS (WITH PERIODIC B.CS.)                               *
C*              LAP(PSI) = VORTICITY                                  *
C*              LAP(CHI) = DIVERGENCE.                                *
C*      FROM THESE WE GET THE NEW VELOCITIES:                         *
C*              U = D(CHI)/DX                                         *
C*              V = D(PSI)/DX.                                        *
C*      THE SPATIAL GRID IS STAGGERED; THE VORTICITY AND              *
C*      DIVERGENCE ARE SPECIFIED AT "WHOLE POINTS", AS IS THE         *
C*      GEOPOTENTIAL; THE VELOCITIES ARE AT "HALF POINTS".            *
C*      VELOCITIES NEEDED AT WHOLE POINTS ARE GOT BY AVERAGING.       *
C*      THE TIME-STEPPING IS DONE BY AN ADAMS-BASHFORTH SCHEME;       *
C*      THIS MEANS THAT WE NEED THE FORCING (R.H.S.) TERMS AT         *
C*      TWO TIME-LEVELS.                                              *
C*                                                                    *
C**********************************************************************
C
      INCLUDE 'DYNAMO.COM'
C
C----------------------------------------------------------------------
C     OPEN CONTROL (CARD) FILE AND OUTPUT (LINEPRINTER) FILE
      OPEN(UNIT=5,DEVICE='DSK',ACCESS='SEQIN',
     X    MODE='ASCII',FILE='DYNAMO.CDS')
      OPEN(UNIT=6,DEVICE='DSK',ACCESS='SEQOUT',
     X    MODE='ASCII',FILE='DYNAMO.LPT')
C

C     OPEN OUTPUT FILE FOR DIAGNOSTICS, ETC. (ON SCRATCH)
      OPEN(UNIT=10,DEVICE='SCR',ACCESS='SEQOUT',
     X    MODE='BINARY',FILE='DYNAMO.DAT')
C----------------------------------------------------------------------
C     SET UP THE CONTROL PARAMETERS, CONSTANTS, ETC, FOR THE RUN
      CALL GETPAR
C
C     SET UP THE INITIAL CONDITIONS
      CALL ICS
C
C     CALL THE INITIALIZATION PROCEDURE
C     IF( INIT ) CALL LAPIN  ! NOT IMPLIMENTED IN THIS VSN.
C
C     PERFORM THE INTEGRATION
      DO 1000 NS=1,NSTEPS
        NSTEP = NS
        CALL STEPON(NSTEP)
1000  CONTINUE
C
C     PRINT OUT THE FINAL RESULTS
      CALL OUTPUT
C
      STOP
      END
C----------------------------------------------------------------------
      SUBROUTINE GETPAR
C
C     DEFINE CONSTANTS AND PARAMETERS AND SET UP THE
C     GRID. INITIALIZE FLOW CONTROL VARIABLES FOR THE RUN
C
      INCLUDE 'DYNAMO.COM'
      DOUBLE PRECISION STRING
C
C     SOME PARAMETERS
      DATA PI/3.14159265/, FCOR/1.E-04/,BETA/1.6E-11/,GRAV/9.81/
C     SOME SCALES
      DATA SXL/1.E+06/,SXV/10./
C

      SXT = 1./FCOR          !  TIME SCALE
      SXDV = SXV/SXL         !  SCALE FOR DIVERGENCE AND VORTICITY
      SXFI = SXL*SXV/SXT     !  SCALE FOR GEOPOTENTIAL
C
C     READ CONTROL PARAMETERS FOR THE FLOW OF COMPUTATIONS
      READ(5,*) STRING,LINEAR
      READ(5,*) STRING,IQG
      READ(5,*) STRING,INIT
      READ(5,*) STRING,IPRINT
      READ(5,*) STRING,NPRINT
      WRITE(6,900) LINEAR,IQG,INIT,IPRINT,NPRINT
900   FORMAT(/' DYNAMO '/' RUN CONTROL PARAMETERS '/
     X    ' LINEAR ',L1,' QG ',L1,' INIT ',L1,' IPRINT ',L1/
     X    ' PRINT OUT INTERMEDIATE RESULTS EVERY ',I5,' STEPS'/)
C
C     SET UP THE INDEPENDENT VARIABLE DISCRETIZATION
      READ(5,*) STRING,NX      ! NUMBER OF POINTS IN X-DIRECTION
      READ(5,*) STRING,DX      ! GRIDLENGTH IN X-DIRECTION (METRES)
      READ(5,*) STRING,NSTEPS  ! NUMBER OF TIMESTEPS FOR RUN
      READ(5,*) STRING,DT      ! TIME INCREMENT (SECONDS)
      NXP1 = NX+1
      XLEN = NX*DX
      LENX = XLEN/1000.    ! CHANNEL LENGTH IN KM.
      TLEN = NSTEPS*DT
      HLEN = TLEN/(60.*60.)  ! FORECAST LENGTH IN HOURS
      WRITE(6,901) NX,LENX,DX,NSTEPS,HLEN,DT
      TYPE    901, NX,LENX,DX,NSTEPS,HLEN,DT
901   FORMAT('   GRIDPTS: ',I4,3X,
     X       ' CHANNEL LENGTH: ',I6,'KM',
     X       '  GRIDLENGTH DX: ',-3PF6.0,'KM'/
     X       ' TIMESTEPS: ',I4,3X,
     X    ' FORECAST LENGTH:',0PF5.1,' HOURS  ',
     X    ' TIMESTEP: ',0PF6.0,' SEC.'/)
C
```

- 25 -

Appendix C

```
C       CALCULATE THE INDEPENDENT VARIABLES (FOR PLOT AXES)
        DO 10 NN=0,NXP1
          X(NN) = NN*DX/1.E+03    ! GRIDPOINTS (KM)
10      CONTINUE

        DO 20 NN=0,NSTEPS
          T(NN) = NN*DT/(60.*60.)    ! TIMSTEPS (HOURS)
20      CONTINUE
C
        DX = DX/SXL    !
        DT = DT/SXT    !    NONDIMENSIONAL
C
C       READ MEAN FLOW PARAMETERS
        READ(5,*) STRING,UBAR
        READ(5,*) STRING,FIBAR
        U0 = UBAR/SXV
        FI0 = FIBAR/SXFI
C
        RO = SXV/(FCOR*SXL)            !    ROSSBY NUMBER
        RB = BETA*SXL/FCOR             !    MEASURE OF BETA EFFECT
        RF = FIBAR/(FCOR*SXL)**2       !    RECIPROCAL OF FROUDE NO.
C
        WRITE(6,907) UBAR,FIBAR,RO,RB,RF
907     FORMAT(' MEAN ZONAL WIND UBAR = ',F6.0,' M/S'/
     X          ' MEAN GEOPOTENTIAL:',F8.0,' (M/S)**2 '/
     X          ' ROSSBY NUMBER: ',1PE10.1/
     X          ' RATIO RB      :  ',1PE10.1/
     X          ' RATIO RF      :  ',1PE10.1/)
C
C       READ INDICATOR FOR INITIAL CONDITIONS
        READ(5,*) STRING, ICNUM
        TYPE 9901,ICNUM
        WRITE(6,9901) ICNUM
9901    FORMAT(' INITIAL CONDITIONS:  ICS = ',I2/)
C
C       READ INDICATORS FOR HOVMOELLER DIAGRAMS
        READ(5,*) STRING, IHOV,NXHOV,NTHOV
        TYPE 9902,IHOV,NXHOV,NTHOV
        WRITE(6,9902) IHOV,NXHOV,NTHOV
9902    FORMAT(' HOVMOELLER DIAGRAMS  ',L2,2I4/)
C
        RETURN
        END
C-----------------------------------------------------------------
        SUBROUTINE ICS
C
C       DEFINE THE INITIAL CONDITIONS. THE VALUES SPECIFIED ARE
C       FOR U, V, AND FI: THE INITIAL VALUES OF VELOCITY ARE
C       SPECIFIED AT HALF POINTS (STARTING AT I=-1/2); THE VALUES
C       OF FI ARE GIVEN AT WHOLE POINTS (STARTING AT I=0).
C
        INCLUDE 'DYNAMO.COM'

        REAL KOUNT(20),PHASE(20),AMPL(20)
        REAL COSN   (0:201),SINN   (0:201)
        REAL COSNMH(0:201),SINNMH(0:201)
C
        NSTEP = 0
C       DEFINE THE INITIAL VELOCITIES AND GEOPOTENTIAL
C*****     SOME SAMPLE I.CS. ARE GIVEN HERE; IF YOU WANT TO DEFINE
C*****     OTHER CONDITIONS, IDENTIFY THEM USING ICNUM > 4.
```

```
        AM = 1.    !   ZONAL WAVENUMBER ONE
C
        FIAMP = 1.E+03
        FIAMP = FIAMP/SXFI   !   NON-DIM AMPLITUDE OF GEOPOTENTIAL
C
        XNX = FLOAT(NX)
        DO 100 NN=0,NXP1
          XNN = FLOAT(NN)
          XNMH = FLOAT(NN)-0.5
          U (NN) = 0.
          V (NN) = 0.
          FI(NN) = 0.
C
          COSN  (NN) = COS(AM*2.*PI*XNN /XNX)
          COSNMH(NN) = COS(AM*2.*PI*XNMH/XNX)
          SINN  (NN) = SIN(AM*2.*PI*XNN /XNX)
          SINNMH(NN) = SIN(AM*2.*PI*XNMH/XNX)
100     CONTINUE
C
C       CALCULATE SOME PHASE-SPEEDS (SEE REPORT, SECTION 3)
        XLEN = NX*(DX*SXL)
        WLEN = XLEN / AM
        ZK = 2.*PI/WLEN
        CGRAV = SQRT( FIBAR + (FCOR/ZK)**2 )
        CROSB = - (BETA/ZK**2) / (1.+(FCOR**2/(FIBAR*ZK**2)))
        WRITE(6,92221) CGRAV,CROSB
        TYPE    92221, CGRAV,CROSB
92221           FORMAT(' WAVESPEEDS:   CG, CR   ' 1P2E12.3 /)
C

        GO TO (101,102,103,104,105,106,107,108),ICNUM
        STOP ' ICNUM OUTSIDE RANGE '
C
C       PURE GEOPOTENTIAL PERTURBATION: COMBINATION OF THREE WAVES.
101     CALL VECCON(FIAMP,COSN,FI,0,NXP1)    ;   GO TO 1000
C
C       APPROX ROSSBY WAVE
102     CALL VECCON(FIAMP,COSN,FI,0,NXP1)
        CALL DDXB(FI,V,NX,DX)       ;   GO TO 1000
C
C       APPROX EASTWARD-TRAVELLING G-I WAVE
103     CALL VECCON(FIAMP,COSN,FI,0,NXP1)
        UAMP = FIAMP
        CALL VECCON(UAMP,COSNMH,U,0,NXP1)      ;   GO TO 1000
C
C       APPROX WESTWARD-TRAVELLING G-I WAVE
104     CALL VECCON(FIAMP,COSN,FI,0,NXP1)
        UAMP = - FIAMP
        CALL VECCON(UAMP,COSNMH,U,0,NXP1)   ;   GO TO 1000
C
C       MORE ACCURATE ROSSBY WAVE
105     CONTINUE
        C = CROSB
        UAMP = (C/FIBAR)*FIAMP*(SXFI/SXV)
        VAMP = ZK*FCOR/(ZK**2*C+BETA) * UAMP
        CALL VECCON(FIAMP,COSN,FI,0,NXP1)
        CALL VECCON(UAMP,COSNMH,U,0,NXP1)
        CALL VECCON(VAMP,SINNMH,V,0,NXP1)
        CONTINUE                  ;   GO TO 1000
C
```

```
C       MORE ACCURATE EASTWARD GRAVITY WAVE
106     CALL VECCON(FIAMP,COSN,FI,0,NXP1)
        C = CGRAV
        UAMP = (C/FIBAR)*FIAMP*(SXFI/SXV)
        CALL VECCON(UAMP,COSNMH,U,0,NXP1)        ;   GO TO 1000
C
C       MORE ACCURATE WESTWARD GRAVITY WAVE
107     CALL VECCON(FIAMP,COSN,FI,0,NXP1)

        C = - CGRAV
        UAMP = (C/FIBAR)*FIAMP*(SXFI/SXV)
        CALL VECCON(UAMP,COSNMH,U,0,NXP1)        ;   GO TO 1000
C
C 108   CONTINUE       ;   GO TO 1000
C#######################################################################
C       COMBINATION OF WAVES WITH A MINUS-FIVE-THIRDS SPECTRUM
C       AND GEOSTROPHIC WIND (INITIALIZATION TEST)
108     CONTINUE
        FIAMP = 1.E+03
        FIAMP = FIAMP/SXFI
        KMAX = 3
        POWER = -(5./3.)
        POWER = 0.
        WRITE(6,9071) KMAX,POWER
9071    FORMAT(/' ICNUM 8  KMAX: ',I4,' WAVES, POWER SPECTRUM: ',F8.4/)
        DO 166 KK=1,KMAX
          KOUNT(KK) = KK
          AMPL(KK) = FLOAT(KK)**POWER
          PHASE(KK) = RAN(1.)*2.*PI
166     CONTINUE
        CALL PLOTLN(KOUNT,AMPL ,1,KMAX,0.,0.,'   AMPL   ')
C       CALL PLOTLN(KOUNT,PHASE,1,KMAX,0.,0.,'  PHASE   ')
        XNX = FLOAT(NX)
        DO 1006 NN=0,NXP1
          XNN = FLOAT(NN)
          U (NN) = 0.
          V (NN) = 0.
          SUM = 0.0
          DO 10006 KK=1,KMAX
            XKK = FLOAT(KK)
            SUM = SUM + AMPL(XKK) * COS(XKK*2.*PI*XNN/XNX + PHASE(XKK) )
10006     CONTINUE
          FI(NN) = SUM*FIAMP
1006    CONTINUE
        CALL DDXB(FI,V,NX,DX)
        CONTINUE                      ;   GO TO 1000
C#######################################################################
C
C*****
C
1000    CONTINUE
C
C       CALCULATE DIVERGENCE AND VORTICITY (AT WHOLE POINTS)
        CALL DDXF(U, DIV,NX,DX)
        CALL DDXF(V,VORT,NX,DX)
C

C       PLOT INITIAL VALUES OF DEPENDENT VARIABLES
        IF(IPRINT) CALL PLOTLN(X,U    ,0,NX,-2.,+2.,'INITIAL U ')
        IF(IPRINT) CALL PLOTLN(X,V    ,0,NX,-2.,+2.,'INITIAL V ')
        IF(IPRINT) CALL PLOTLN(X,FI   ,0,NX,-2.,+2.,'INITIAL FI')
C       IF(IPRINT) CALL PLOTLN(X,VORT,0,NX, 0., 0.,'VORTICITY ')
C       IF(IPRINT) CALL PLOTLN(X,DIV ,0,NX, 0., 0.,'DIVERGENCE')
```



```
C
C       CALCULATE THE INITIAL ENERGIES.
        CALL ENERGY(NSTEP)
C
C       SAVE FIELD VALUES AT A CENTRAL POINT FOR PLOTTING
        NXH = NX/2
        PTU(NSTEP) = U(NXH)
        PTV(NSTEP) = V(NXH)
        PTFI(NSTEP) = FI(NXH)
        PTDIV(NSTEP) = DIV(NXH)
        PTVORT(NSTEP) = VORT(NXH)
C
C       SAVE VALUES FOR THE HOVMOELLER DIAGRAM OF FI.
        WRITE (10) FI
C
        RETURN
        END
C-----------------------------------------------------------------------
        SUBROUTINE STEPON(NSTEP)
C
C       PERFORM SINGLE TIMESTEP FOR DYNAMO
C
        INCLUDE 'DYNAMO.COM'
C
C       GET RHS OF VORTICITY, DIVERGENCE AND CONTINUITY EQUATIONS
C       AT THE PRESENT TIMELEVEL AND PUT IN RHS1V, RHS1D AND RHS1C
C       (OLD VALUES ARE IN RHS2 IF NSTEP > 0).
C
        XNL = 1.0                 !  NON-LINEAR FACTOR
        IF(LINEAR) XNL = 0.0
C
C       CALCULATE THE ADVECTION TERMS

        CALL XMEANB(VORT,WORK2,NX) !
        CALL XMEANB(DIV ,WORK3,NX) ! AVERAGE TO VELOCITY POINTS
        CALL XMEANB(FI  ,WORK4,NX) !
        DO 100 NN=0,NXP1
          WORK2(NN) = (U0 + XNL*U(NN))*WORK2(NN)
          WORK3(NN) = (U0 + XNL*U(NN))*WORK3(NN)
          WORK4(NN) = (U0 + XNL*U(NN))*WORK4(NN)
100     CONTINUE
        CALL DDXF(WORK2,WORK1,NX,DX) !
        CALL DDXF(WORK3,WORK2,NX,DX) ! ADVECTION TERMS
        CALL DDXF(WORK4,WORK3,NX,DX) !
C
        CALL XMEANF(U,U,NX)  !  U AND V NO LONGER NEEDED AT HALF
        CALL XMEANF(V,V,NX)  !  POINTS BUT AT WHOLE POINTS.
C
        CALL DDXX(FI,WORK4,NX,DX) ! PRESSURE GRADIENT TERM
C
C       GET R.H.S. OF EQUATIONS (18), (19), AND (20)
        DO 200 NN=0,NXP1
          RHS1V(NN) = - ( RO*WORK1(NN)+ DIV(NN)+RB*V(NN) )
          RHS1D(NN) = - ( RO*WORK2(NN)-VORT(NN)+RB*U(NN)+WORK4(NN) )
          RHS1C(NN) = - ( RO*WORK3(NN)-HO*U0*V(NN)+RF*DIV(NN) )
200     CONTINUE
C
C       STEP FORWARD FOR VORTICITY, DIVERGENCE AND GEOPOTENTIAL
C       THE ADAMS BASHFORTH SCHEME IS USED; THE TIMESTEP IS FORWARD
C       AND THE R.H.S. SIDES ARE ESTIMATED BY LINEAR EXTRAPOLATION
C       FROM THE NTH AND (N-1)TH TIMELEVELS.
```

```
C***    IF YOU WANT TO CHANGE THE TIME-SCHEME DO IT HERE.
        DELT = DT
        FN   =  1.5
        FNM1 = -0.5
        IF(NSTEP.GT.0) GO TO 250
        FN   = 1.0
        FNM1 = 0.0    !  FIRST STEP IS EULER FORWARD
250     CONTINUE
C
C       ACTUAL FORWARD STEP, EQUATIONS (18),(19),(20)

        DO 300 NN=0,NXP1
          WORK1(NN) = VORT(NN) + DELT * (FN*RHS1V(NN)+FNM1*RHS2V(NN))
          WORK2(NN) =  DIV(NN) + DELT * (FN*RHS1D(NN)+FNM1*RHS2D(NN))
          WORK3(NN) =   FI(NN) + DELT * (FN*RHS1C(NN)+FNM1*RHS2C(NN))
300     CONTINUE
C
C       IF THE TIMESTEPPING SCHEME IS TWO-LEVEL WE ONLY NEED
C       TO KEEP THE MOST RECENT VALUES; IF IT IS A THREE-LEVEL
C       SCHEME (E.G. LEAPFROG OR ADAMS BASHFORTH) THEN SOME OLD
C       VALUES WILL BE NEEDED. THE EMPHASIS IN THIS 1-D MODEL
C       IS ON SIMPLICITY OF CODING RATHER THAN ECONOMY OF CORE.
C
C       SAVE THE OLD VALUES AND RELOCATE THE UPDATED ONES
CCC     CALL MEMOVE(VORT,VOROLD,0,NXP1)     !
CCC     CALL MEMOVE(DIV ,DIVOLD,0,NXP1)     !  NOT REQUIRED THIS VSN.
CCC     CALL MEMOVE(FI  , FIOLD,0,NXP1)     !
C
        CALL MEMOVE(WORK1,VORT,0,NXP1)  !
        CALL MEMOVE(WORK2,DIV ,0,NXP1)  !  MOVE IN NEW VALUES
        CALL MEMOVE(WORK3,FI  ,0,NXP1)  !
C
C       CALCULATE THE VELOCITIES AT THE NEW TIME
        CALL POIS1D(CHI, DIV,NX,DX)  !  VELOCITY POTENTIAL
        CALL POIS1D(PSI,VORT,NX,DX)  !      STREAM FUNCTION
        CALL DDXB(CHI,U,NX,DX)    !        ZONAL VELOCITY
        CALL DDXB(PSI,V,NX,DX)    !  MERIDIONAL VELOCITY
C
C       CALCULATE THE ENERGETICS
        CALL ENERGY(NSTEP)
C
C       SHIFT R.H.S TERMS FOR NEXT CYCLE
        CALL MEMOVE(RHS1V,RHS2V,0,NXP1)
        CALL MEMOVE(RHS1D,RHS2D,0,NXP1)
        CALL MEMOVE(RHS1C,RHS2C,0,NXP1)
C
C       SAVE FIELD VALUES AT A CENTRAL POINT (FOR PLOTTING)
        NXH = NX/2
        PTU(NSTEP) = U(NXH)
        PTV(NSTEP) = V(NXH)
        PTFI(NSTEP) = FI(NXH)
        PTDIV(NSTEP) = DIV(NXH)
        PTVORT(NSTEP) = VORT(NXH)
C
C       SAVE VALUES FOR THE HOVMOELLER DIAGRAM OF FI.
        IF((NSTEP/NTHOV*NTHOV).EQ.NSTEP) WRITE (10) FI
C
        RETURN
        END
C------------------------------------------------------------------
```

```
        SUBROUTINE ENERGY(NSTEP)
C
C       CALCULATE THE ZONALLY AVERAGED ENERGIES (SEE REPORT, SEC 4)
C
        INCLUDE 'DYNAMO.COM'
C
        XK = SXV**2*SXFI/(2.*GRAV)
        XA = SXFI**2/(2.*GRAV)
        XNX = NX
        ROTKE = 0.
        DIVKE = 0.
        AVPE = 0.
        SOURC = 0.
        DO 100 NN=1,NX
          ROTKE = ROTKE + (V(NN)**2)*(FI0+(FI(NN-1)+FI(NN))/2)*XK
          DIVKE = DIVKE + (U(NN)**2)*(FI0+(FI(NN-1)+FI(NN))/2)*XK
          AVPE  = AVPE  + ( FI(NN)**2 )*XA
          SOURC = SOURC + ( V(NN)*0.5*(U(NN)**2+V(NN)**2)*SXV**3
     X                  +   V(NN)*(FI(NN-1)+FI(NN))/2*SXV*SXFI  )
     X                   *  FCOR*U0*SXV/GRAV
100     CONTINUE
        EDDYKE = ROTKE+DIVKE
        EDYTOT = EDDYKE + AVPE
        KEROT(NSTEP) = (1/XNX)*ROTKE       !   ROTATIONAL KINETIC ENERGY
        KEDIV(NSTEP) = (1/XNX)*DIVKE       !    DIVERGENT KINETIC ENERGY

        KETOT(NSTEP) = (1/XNX)*EDDYKE      !   TOTAL EDDY KINETIC ENERGY
        APE  (NSTEP) = (1/XNX)*AVPE        !  AVAILABLE POTENTIAL ENERGY
        APLUSK(NSTEP) = (1/XNX)*EDYTOT     !           TOTAL EDDY ENERGY
        SOURCE(NSTEP) = (1/XNX)*SOURC      !            SOURCE OF ENERGY
C
C       CALCULATE RATE OF CHANGE OF EDDY ENERGY BY FINITE DIFFERENCE
        IF(NSTEP.GT.0)
     X     DDTAPK(NSTEP)=(APLUSK(NSTEP)-APLUSK(NSTEP-1))/(DT*SXT)
C
C       PRINT/TYPE THE ENERGIES AT EACH TIMESTEP: THEY GIVE A GOOD
C       INDICATION OF NUMERICAL STABILITY OR INSTABILITY
C       WRITE(6,99901) NSTEP,ROTKE,DIVKE,EDDYKE,AVPE,EDYTOT,SOURC
        TYPE    99901, NSTEP,ROTKE,DIVKE,EDDYKE,AVPE,EDYTOT,SOURC
99901   FORMAT(' NSTEP: ',I4,' KR,KD,KT,AP,TE,SC ',1P6E12.3)
C
        RETURN
        END
C------------------------------------------------------------------
        SUBROUTINE OUTPUT
C
C       PRINT OUT THE RESULTS OF THE RUN
C
        INCLUDE 'DYNAMO.COM'
C
C       PLOT THE FINAL FIELD VALUES
        WRITE(6,99991)
99991   FORMAT(/' **********      FINAL FIELD VALUES     **********'/)
        IF(IPRINT) CALL PLOTLN(X,U   ,0,NX,-2.,+2.,'    U      ')
        IF(IPRINT) CALL PLOTLN(X,V   ,0,NX,-2.,+2.,'    V      ')
        IF(IPRINT) CALL PLOTLN(X,FI  ,0,NX,-2.,+2.,'    FI     ')
C       IF(IPRINT) CALL PLOTLN(X,VORT,0,NX, 0., 0.,'VORTICITY ')
C       IF(IPRINT) CALL PLOTLN(X,DIV ,0,NX, 0., 0.,'DIVERGENCE')
C
```

```
C       PLOT THE FIELD VALUES AT THE MIDDLE POINT
        WRITE(6,99992)
99992   FORMAT(/' **********       POINT TENDENCIES       **********'/)
        IF(IPRINT) CALL PLOTLN(T,PTU    ,0,NSTEPS,00.,00.,'MIDPT U   ')
        IF(IPRINT) CALL PLOTLN(T,PTV    ,0,NSTEPS,00.,00.,'MIDPT V   ')

        IF(IPRINT) CALL PLOTLN(T,PTFI   ,0,NSTEPS,00.,00.,'MIDPT FI  ')
C       IF(IPRINT) CALL PLOTLN(T,PTDIV  ,0,NSTEPS,00.,00.,'MIDPT DIV ')
C       IF(IPRINT) CALL PLOTLN(T,PTVORT,0,NSTEPS,00.,00.,'MIDPT VORT')
C
C       GRAPH THE EVOLUTION OF THE ENERGY QUANTITIES
        WRITE(6,99993)
99993   FORMAT(/' **********      ENERGY QUANTITIES      **********'/)
        IF(IPRINT) CALL PLOTLN(T,KEROT,0,NSTEPS, 0., 1.,' ROT KE   ')
        IF(IPRINT) CALL PLOTLN(T,KEDIV,0,NSTEPS, 0., 1.,' DIV KE   ')
        IF(IPRINT) CALL PLOTLN(T,KETOT,0,NSTEPS, 0., 1.,' TOT KE   ')
        IF(IPRINT) CALL PLOTLN(T, APE ,0,NSTEPS, 0., 1.,'  APE     ')
        IF(IPRINT) CALL PLOTLN(T,APLUSK,0,NSTEPS, 0., 1.,' APLUSK   ')
        IF(IPRINT) CALL PLOTLN(T,SOURCE,0,NSTEPS, 0., 0.,' SOURCE   ')
        IF(IPRINT) CALL PLOTLN(T,DDTAPK,0,NSTEPS, 0., 0.,' DDTAPK   ')
C
C       PLOT HOVMOELLER DIAGRAM OF GEOPOTENTIAL
        IF( IHOV ) CALL HOVMOL
C
        RETURN
        END
C-----------------------------------------------------------------------
        SUBROUTINE POIS1D(PHI,RHO,N,DELTA)
C
C       SOLVE A POISSON EQUATION IN ONE DIMENSION
C       WITH PERIODIC BOUNDARY CONDITIONS
C               LAP(PHI) = RHO
C       WHERE LAP IS THE LAPLACIAN OPERATOR
C       [REF: HOCKNEY, R.W. AND J.W. EASTWOOD, 1981:
C       "COMPUTER SIMULATION USING PARTICLES"
C       MC GRAW-HILL INC., NEW YORK, 540PP.]
C
        REAL PHI(0:N),RHO(0:N)
C
C       ASSUME THE END VALUES VANISH (THEY ARE ARBITRARY).
        PHI(0) = 0.
C       CALCULATE THE FIRST VALUE
        DELSQ = DELTA**2
        SUM = 0.0
        DO 100 NN=1,N
          SUM = SUM + NN*RHO(NN)
100     CONTINUE
        PHI(1) = (SUM/N)*DELSQ
C       LOOP FOR THE OTHER VALUES
        DO 200 NN=2,N
          PHI(NN) = RHO(NN-1)*DELSQ+2.*PHI(NN-1)-PHI(NN-2)
200     CONTINUE
C
C       CHECK R.H. END IS APPROX ZERO (MEASURE OF ROUNDOFF)
C       TYPE 99901, ( NN,PHI(NN),NN=1,N )
99901   FORMAT(/' POIS1D'/(I5,1PE12.3))
C       FILL IN END VALUES
        CALL ENDS(PHI,N)
        RETURN
        END
C-----------------------------------------------------------------------
```

-29-

```
C       CALCULATION OF VARIOUS FINITE DIFFERENCES
C         (1) CENTRED FIRST DIFFERENCE
      SUBROUTINE DDXBAR(F,DIFF,N,DELTA)
        DIMENSION F(0:N),DIFF(0:N)
        DO 10 NN=1,N
          DIFF(NN) = ( F(NN+1)-F(NN-1) ) / (2.*DELTA)
10      CONTINUE
        CALL ENDS(DIFF,N)
        RETURN
C         (2) FORWARD FIRST DIFFERENCE
      ENTRY DDXF(F,DIFF,N,DELTA)
        DO 20 NN=1,N
          DIFF(NN) = ( F(NN+1)-F(NN) ) / DELTA
20      CONTINUE
        CALL ENDS(DIFF,N)
        RETURN
C         (3) BACKWARD FIRST DIFFERENCE
      ENTRY DDXB(F,DIFF,N,DELTA)
        DO 30 NN=1,N
          DIFF(NN) = ( F(NN)-F(NN-1) ) / DELTA
30      CONTINUE
        CALL ENDS(DIFF,N)
        RETURN
C         (4) CENTRED SECOND DIFFERENCE
      ENTRY DDXX(F,DIFF,N,DELTA)
        DO 40 NN=1,N
          DIFF(NN) = (F(NN+1)+F(NN-1)-2.*F(NN))/(DELTA**2)
40      CONTINUE
        CALL ENDS(DIFF,N)
        RETURN

        END
C
C       AVERAGING OPERATORS (FORWARD AND BACKWARD)
      SUBROUTINE XMEANF(F,FBARX,N)
        DIMENSION F(0:N),FBARX(0:N)
        DO 10 NN=1,N
          FBARX(NN) = ( F(NN)+F(NN+1) ) / 2.0
10      CONTINUE
        CALL ENDS(FBARX,N)
        RETURN
      ENTRY XMEANB(F,FBARX,N)
        DO 20 NN=N,1,-1
          FBARX(NN) = ( F(NN-1)+F(NN) ) / 2.0
20      CONTINUE
        CALL ENDS(FBARX,N)
        RETURN
        END
C
        SUBROUTINE ENDS(A,N)
C       FILL END VALUES OF A PERIODIC ARRAY
        REAL A(0:N)
        A(0) = A(N)
        A(N+1) = A(1)
        RETURN
        END
C
```

```
      SUBROUTINE VECCON(C,A,B,I1,I2)
C     MULTIPLY A REAL VECTOR A(I1:I2) BY A
C     CONSTANT C AND STORE IN MATRIX B.
C
      REAL A(I1:I2),B(I1:I2)
C
      DO 10 K=I1,I2
          B(K) = C*A(K)
10    CONTINUE
      RETURN
      END
C-------------------------------------------------------------------
      SUBROUTINE MEMOVE(FROM,TO,NN,NX)
C        MOVE AN ARRAY IN CORE
      REAL FROM(NN:NX),TO(NN:NX)
      DO 100 K=NN,NX
  100 TO(K) = FROM(K)
      RETURN
      END
C-------------------------------------------------------------------
      SUBROUTINE PLOTLN (PX,PY,K1,K2,PYMIN,PYMAX,MESAGE)
C
C       LINE PRINTER PLOT OF PY(I) VS. PX(I)    ; I=K1,K2
C       ------------------------------------------------
C       WRITTEN BY J HAMILTON; MODIFIED BY PETER LYNCH.
C
C       LIMITS OF Y-AXIS ARE PYMIN,PYMAX BUT IF THESE ARE
C       TOO SMALL THE RANGE IS EXTENDED. MESAGE IS A
C       SMALL ARRAY VARIABLE TO PRINT A SHORT HEADING.
C
      DIMENSION PX(K1:K2),PY(K1:K2),MESAGE(2)
      DIMENSION ZXSCAL(11),ZYSCAL(11)
      DIMENSION IGRAPH(103,43)
      DATA IBLANK/4H    /,ISTAR /4H*   /,IXLINE/4H-   /,IYLINE/4HI   
      DATA IDOT  /4H.   /,ICROSS/4H+   /
      DATA IXMIN/2/,IXMAX/102/,IYMIN/2/,IYMAX/42/
C
      KCHANL=6
      ICHANL=KCHANL
      IF(ICHANL.LT.0) ICHANL=-ICHANL
C
      WRITE(KCHANL,99999) MESAGE
99999 FORMAT(//70X,2A5)
C
C*CHECK THAT THERE ARE SUFFICIENT POINTS IN THE ARRAYS FOR PLOT
      IPOINT=K2-K1+1
      IF(IPOINT.GT.0) GOTO 15
      WRITE(ICHANL,10)
   10 FORMAT(' ERROR IN CALL TO ==PLOTLN== NOT ENOUGH POINTS ')
      RETURN
C
C*FIND MAXIMUM AND MINIMUM VALUES OF "X"
   15 CONTINUE
      ZXMAX = PX(K1)
      ZXMIN = PX(K1)
      DO 20 I=K1,K2
         IF (PX(I) .GT. ZXMAX)  ZXMAX=PX(I)
         IF (PX(I) .LT. ZXMIN)  ZXMIN=PX(I)
   20    CONTINUE
C
C*FIND MAXIMUM AND MINIMUM VALUES OF "Y"
      ZYMAX = PY(K1)
      ZYMIN = PY(K1)
      DO 25 I=K1,K2
         IF (PY(I) .GT. ZYMAX)  ZYMAX=PY(I)
         IF (PY(I) .LT. ZYMIN)  ZYMIN=PY(I)
   25    CONTINUE
C
C*RANGES OF X AND Y - PRINT MESSAGE
      IF((ZXMIN.NE.ZXMAX).AND.(ZYMIN.NE.ZYMAX))  GOTO 35
      WRITE(ICHANL,30) ZXMIN,ZXMAX,ZYMIN,ZYMAX
   30 FORMAT(' REDUNDANT CALL TO ==PLOTLN=='/
     1   ' RANGE OF X: ',1PE12.3,E12.3/
     2   ' RANGE OF Y: ',1PE12.3,E12.3)
      RETURN
C
C*SCALING FACTORS
   35 CONTINUE
      IF(PYMIN.EQ.PYMAX) GO TO 36
      IF(PYMIN.LT.ZYMIN) ZYMIN=PYMIN
      IF(PYMAX.GT.ZYMAX) ZYMAX=PYMAX
36    ZXFACT = 100.0 / (ZXMAX-ZXMIN)
      ZYFACT = 40.0 / (ZYMAX-ZYMIN)
C
C*X AXIS SCALE
      ZXAXES=(ZXMAX-ZXMIN)/10.0
      DO 40 I=1,11
         ZXSCAL(I)=FLOAT(I-1)*ZXAXES + ZXMIN
   40    CONTINUE
C
C*Y AXIS SCALE
      ZYAXES=(ZYMAX-ZYMIN)/10.0
      DO 45 I=1,11
         IREV=12-I
         ZYSCAL(IREV)=FLOAT(I-1)*ZYAXES + ZYMIN
   45    CONTINUE
C

C*CLEAR THE GRAPHICAL AREA
      DO 50 IY=2,42
         DO 50 IX=2,102
            IGRAPH (IX,IY)=IBLANK
   50       CONTINUE
C
C PRINT X-AXIS AT ZERO (IF POSSIBLE)
      IF((ZYMIN.GT.0.0).OR.(ZYMAX.LT.0.0)) GO TO 60
      YY = 42.0 - (0.0-ZYMIN)*ZYFACT
      IY = YY+0.5
      IF(IY.LT.IYMIN) IY=IYMIN
      IF(IY.GT.IYMAX) IY=IYMAX
      DO 55 JX=2,102
         IGRAPH (JX,IY) = IXLINE
   55    CONTINUE
C
C PRINT Y-AXIS AT ZERO (IF POSSIBLE)
   60 CONTINUE
      IF((ZXMIN.GT.0.0).OR.(ZXMAX.LT.0.0)) GO TO 70
      XX = (0.0-ZXMIN)*ZXFACT+2.0
      IX=XX+0.5
      IF(IX.LT.IXMIN) IX=IXMIN
      IF(IX.GT.IXMAX) IX=IXMAX
```

```
      DO 65 JY=2,42
         IGRAPH (IX,JY) = IYLINE
   65    CONTINUE
C
C*X AXIS (TOP AND BOTTOM)
   70 CONTINUE
      DO 75 IX=2,102
         IGRAPH(IX, 2)=IXLINE
         IGRAPH(IX,42)=IXLINE
   75    CONTINUE
      DO 80 IX=2,102,10
         IGRAPH(IX, 2)=IYLINE
         IGRAPH(IX,42)=IYLINE
   80    CONTINUE
C

C#Y-AXIS (LEFT AND RIGHT)
      DO 85 IY=2,42
         IGRAPH(  2,IY)=IYLINE
         IGRAPH(102,IY)=IYLINE
   85    CONTINUE
      DO 90 IY=2,42,4
         IGRAPH(  2,IY)=IXLINE
         IGRAPH(102,IY)=IXLINE
   90    CONTINUE
C
C#CORNERS
      IGRAPH(  2, 2)=ICROSS
      IGRAPH(102, 2)=ICROSS
      IGRAPH(  2,42)=ICROSS
      IGRAPH(102,42)=ICROSS
C
C*PLOT THE POINTS (USE NEAREST POINT IN ARRAY)
      DO 135 IPT=K1,K2
         XX = (PX(IPT)-ZXMIN)*ZXFACT+2.0
         IX = XX+0.5
         IF(IX.LT.IXMIN) IX=IXMIN
         IF(IX.GT.IXMAX) IX=IXMAX
         YY = 42.0 - (PY(IPT)-ZYMIN)*ZYFACT
         IY = YY+0.5
         IF(IY.LT.IYMIN) IY=IYMIN
         IF(IY.GT.IYMAX) IY=IYMAX
         IGRAPH (IX,IY) = ISTAR
C
C#JOIN UP THE POINTS WITH LINES
         IF(IPT.EQ.K1)    GOTO 130
         IF(KCHANL.LT.0)  GOTO 130
         IX1=IX
         IY1=IY
         IX2=IXOLD
         IY2=IYOLD
         XX1 = IX1
         YY1 = IY1
         XX2 = IX2
         YY2 = IY2
C
         IDELX = (IX2-IX1)*(IX2-IX1)
         IDELY = (IY2-IY1)*(IY2-IY1)
         IF (IDELY.GT.IDELX)  GOTO 110
C
         SLOPE = 0.0
         TEST  = (XX2-XX1) * (XX2-XX1)
         IF (TEST.GE.1.0E-6) SLOPE= (YY2-YY1) / (XX2-XX1)
         IF (IX1.GT.IX2) GOTO 95
         IMIN = IX1
         IMAX = IX2
         GOTO 100
   95    IMIN = IX2
         IMAX = IX1
  100    CONTINUE
         DO 105 JX = IMIN,IMAX
            XX = JX
            YY = YY1 + SLOPE * (XX - XX1)
            JY = YY
            IF(IGRAPH(JX,JY).EQ.IBLANK) IGRAPH(JX,JY)=IDOT
  105       CONTINUE
         GOTO 130
C
  110    CONTINUE
         SLOPE = 0.0
         TEST  = (YY2-YY1) * (YY2-YY1)
         IF (TEST.GT.1.0E-6) SLOPE= (XX2-XX1) / (YY2-YY1)
         IF (IY1.GT.IY2) GOTO 115
         IMIN = IY1
         IMAX = IY2
         GOTO 120
  115    IMIN = IY2
         IMAX = IY1
  120    CONTINUE
         DO 125 JY = IMIN,IMAX
            YY = JY
            XX = XX1 + SLOPE * (YY - YY1)
            JX = XX
            IF(IGRAPH(JX,JY).EQ.IBLANK) IGRAPH(JX,JY)=IDOT
  125       CONTINUE
C
C#SAVE CURRENT POINT FOR PROCESSING NEXT TIME AROUND
  130    CONTINUE
         IXOLD=IX
         IYOLD=IY
  135    CONTINUE

C
C#OUTPUT THE GRAPH
      DO 150 IYQRT=1,10
         IY1=4*IYQRT-2
         IY2=IY1+1
         IY3=IY2+1
         IY4=IY3+1
         WRITE(ICHANL,140)(ZYSCAL(IYQRT),(IGRAPH(IX,IY1),IX=2,102))
  140    FORMAT(10X,1PE10.2,3X,101(A1))
         WRITE(ICHANL,145)(IGRAPH(IX,IY2),IX=2,102)
         WRITE(ICHANL,145)(IGRAPH(IX,IY3),IX=2,102)
         WRITE(ICHANL,145)(IGRAPH(IX,IY4),IX=2,102)
  145    FORMAT(23X,101(A1))
  150    CONTINUE
      WRITE(ICHANL,140)(ZYSCAL(11),(IGRAPH(IX,42),IX=2,102))
      WRITE(ICHANL,155)(ZXSCAL(I),I=1,11)
  155 FORMAT(/18X,11(1PE10.2)/)
C
      RETURN
```

```
      SUBROUTINE HOVMOL
C
C     PRINT OUT A HOVMOELLER DIAGRAM FROM THE
C     VALUES STORED ON FILE 10
C
      INCLUDE 'DYNAMO,COM'
      REAL ZEBRA(0:121)
C
      DIMENSION CHAR(21)
      DATA CHAR /'0',' ','1',' ','2',' ','3',' ','4',' ','5',
     X              ' ','6',' ','7',' ','8',' ','9',' ','0'/
C
      NXPTS = NXP1/NXHOV
      IF(NXPTS.GT.121) RETURN
C
      CLOSE (UNIT=10)
      OPEN(UNIT=10,DEVICE='SCR',ACCESS='SEQIN',
     X          MODE='BINARY',FILE='DYNAMO.DAT',
     X                  DISPOSE='DELETE')
C
      DO 1000 NSTEP = 0,NSTEPS,NTHOV
        READ(10) FI
        IF(NSTEP.GT.0) GO TO 200
C       DEFINE ZERO AND SPACING
        ZMAX = ABS(FI(0))
        DO 100 NN=1,NX,NXHOV
          ABSFI = ABS(FI(NN))
          IF(ZMAX.LT.ABSFI) ZMAX=ABSFI
100     CONTINUE
        ZERO = 0.
        SPACE = ZMAX/5.         ! CHANGE AS REQUIRED
        IF(SPACE.LT.1.E-30) SPACE = 1.
        WRITE(6,99901)
99901   FORMAT(///)
C
C       DERIVE PRINT CHARACTERS FOR THE ROW
200     CONTINUE
        DO 400 NN=0,NXP1,NXHOV
          FINN = FI(NN)
          IND = MOD(INT((FINN-ZERO)/SPACE+2000.),20)+1
          ZEBRA(NN) = CHAR(IND)
400     CONTINUE
C
C       PRINT THE ROW OF CHARACTERS
        WRITE(6,999) NSTEP,(ZEBRA(NN),NN=0,NXPTS)
        TYPE    999, NSTEP,(ZEBRA(NN),NN=0,NXPTS)
999     FORMAT(1X,I6,4X,121A1)
C
1000  CONTINUE
      WRITE(6,99901)
C
      RETURN
      END
C----------------------------------------------------------------------
```

```
C----------------------------------------------------------------------
C       'DYNAMO.COM'
C       COMMON BLOCKS FOR DYNAMO
C
C       SET MAXIMUM DIMENSIONS FOR ARRAYS
        PARAMETER NPX=201,NPT=4001
C
C       PARAMETERS CONTROLING THE FLOW OF CALCULATIONS
        LOGICAL LINEAR,IQG,INIT,IPRINT,IHOV
        INTEGER NPRINT, ICNUM, NXHOV,NTHOV
        COMMON /FLOW/ LINEAR,IQG,INIT,IPRINT,NPRINT, ICNUM,
     X                IHOV,NXHOV,NTHOV
C
C       VARIOUS CONSTANTS, PARAMETERS AND SCALES
        COMMON /PARAMS/ PI, FCOR,BETA, UBAR,U0,FIBAR,FI0, RF,RO,RB,
     X                  GRAV, SXL,SXT,SXV,SXDV,SXFI
C
C       INDEPENDENT VARIABLES, INCREMENTS, GRIDSPECS, ETC.
        REAL X(0:NPX),T(0:NPT)
        COMMON /IVARS/ NX,NXP1,NSTEPS,DX,DT,X,T
C
C       DEPENDENT VARIABLES (TWO TIME LEVELS), HORIZONTAL VELOCITIES,
C       GEOPOTENTIAL, VORTICITY AND DIVERGENCE.
        REAL U (0:NPX),V (0:NPX)
        REAL FI    (0:NPX),VORT  (0:NPX),DIV   (0:NPX)
C       REAL FIOLD (0:NPX),VOROLD(0:NPX),DIVOLD(0:NPX)
        COMMON /DVARS/ U,V,FI,VORT,DIV
C    X       ,      FIOLD ,VOROLD,DIVOLD     ! NOT REQD THIS VSN
C
C       STREAM FUNCTION AND VELOCITY POTENTIAL
        REAL PSI(0:NPX),CHI(0:NPX)
        COMMON /PSICHI/ PSI,CHI
C
C       RIGHT HAND SIDES (AT TWO TIMES)
        REAL RHS1V(0:NPX),RHS1D(0:NPX),RHS1C(0:NPX)
        REAL RHS2V(0:NPX),RHS2D(0:NPX),RHS2C(0:NPX)
        COMMON /RHS/ RHS1V,RHS1D,RHS1C,RHS2V,RHS2D,RHS2C
C
C       ENERGY QUANTITIES
        REAL KEROT(0:NPT),KEDIV(0:NPT),KETOT(0:NPT)
        REAL APE(0:NPT),APLUSK(0:NPT)
        REAL SOURCE(0:NPT),DDTAPK(0:NPT)
        COMMON / NRG / KEROT,KEDIV,KETOT,APE,APLUSK,TOTEN
     X          ,       SOURCE,DDTAPK
C
C       POINT VALUES OF DEPENDENT VARIABLES
        REAL PTU(0:NPT),PTV(0:NPT),PTFI(0:NPT)
        REAL PTVORT(0:NPT),PTDIV(0:NPT)
        COMMON /POINTS/ PTU,PTV,PTFI,PTVORT,PTDIV
C
C       WORKING SPACE
        REAL WORK1(0:NPX),WORK2(0:NPX),WORK3(0:NPX),WORK4(0:NPX)
        COMMON /WORK/ WORK1,WORK2,WORK3,WORK4
C
C----------------------------------------------------------------------
```