TECHNICAL NOTE No. 44

DYNAMO

A ONE-DIMENSIONAL PRIMITIVE EQUATION MODEL

BY

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

\[ \frac{du}{dt} - fy + \frac{\partial \theta}{\partial x} = 0 \]  \hspace{1cm} (1)
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 derivatives of the momentum equations. The zonally averaged flow is assumed to be in geostrophic balance:

\[
\zeta = -\frac{\partial \Phi}{\partial y}
\]

where \( \zeta \) is taken as constant. We express the total flow as

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

where we note that all quantities other than \( \tilde{\Phi} \) 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

\[
\zeta_t + (u\zeta)_x + f\phi + \beta v = 0 \tag{5}
\]

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

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

\[
\zeta = v_x \quad ; \quad \delta = u_x
\]

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

\[
\Phi_t + (u\Phi)_x - f\bar{u}v + \Phi_{xx} = 0 \tag{7}
\]

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 $\phi$ on y, and scaling arguments can be used to show that this is small so that f and $\phi$ 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:

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

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

$$c(c+\beta/k^2)^2 - c(\hat{\phi}+(f/k)^2) - (\beta/k^2)\hat{\phi} = 0$$

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

$$c = c_R = - (\beta/k^2)/(1+f^2/k^2\hat{\phi})$$

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 \(|c| > |c_R|\) the constant term in (8) is negligible and we get the two roots:

\[
c = \pm \sqrt{ \left( \frac{\phi' f^2}{k^2} \right) }
\]

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

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

where \(h(x,t)\) is the total depth of the fluid. When the fluid surface is perfectly flat, \(h(x,t) = \bar{h}\), 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

\[
\int_{\bar{h}}^h \rho g (z - \bar{h}) \, dz = \frac{1}{2} \rho g (h - \bar{h})^2 = \frac{1}{2g} \rho \Phi^2
\]

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:
\[
\int_0^h \frac{1}{2}(u'^2+v'^2) \rho \, dz = \frac{1}{2}(u'^2+v'^2) \rho h = \frac{1}{2g} \rho (u'^2+v'^2) \theta. \tag{10}
\]

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 \(\rho \theta u'\), \((2)\) by \(\rho \theta v\) and \((3)\) by \(\rho \theta^2\); 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:

\[
\frac{d}{dt} \left[ \int \left[ \frac{1}{2} \rho (u'^2+v'^2) \theta + \frac{1}{2} \rho \theta^2 \right] \, dx \right] = - \int \left[ \rho v \left( \frac{1}{2} (u'^2+v'^2) + \theta^2 \right) \frac{\partial \theta}{\partial y} \right] \, dx \tag{11}
\]

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 \((\bar{u} = \bar{v} = 0)\) 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:

\[
K = K_\psi + K_x; \quad K_\psi = \frac{1}{2} \rho (\nabla \psi)^2 \theta; \quad K_x = \frac{1}{2} \rho (\nabla x)^2 \theta = \frac{1}{2} \rho u'^2 \theta.
\]

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 $fLV$ (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

$$ Ro = (V/fL) \ ; \ R_\beta = (\beta L/f) \sim (L/v) \ ; \ R_F = \tilde{\Theta}/(fL)^2 = (L_R/L)^2. $$

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, $L_R = \sqrt{\tilde{\Theta}/f}$. The equations of motion, (5), (6) and (7), may now be written in nondimensional form

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

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

$$ \Theta_t + Ro(u\Theta)_x - Rou_v + R_F \delta = 0 \quad (15) $$

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 role 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:

$$V = V_x - k \times \nabla \psi ; \nabla^2 \psi = \delta ; \nabla^2 \psi = \zeta.$$

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

$$X_{xx} = \delta ; \quad \psi_{xx} = \zeta$$

with periodic boundary conditions, and derive the velocities from

$$u = \chi_x ; \quad v = \psi_x.$$

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".

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

\[
(q_m)_x = \frac{(q_{m+1} - q_{m-1})}{2\Delta x} \quad ; \quad (q_m)_\beta = \frac{(q_{m+1} - q_{m-1})}{2\Delta x}
\]

Applying these operators successively we find that

\[
(q_m)_x = \frac{(q_{m+1} - q_{m-1})}{2\Delta x} \quad ; \quad (q_m)_\beta = \frac{(q_{m+1} - q_{m-1})}{(\Delta x)^2}
\]

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:

\[
\frac{\partial (uq)}{\partial x} \bigg|_m \rightarrow \frac{u_{m+\frac{1}{2}}(q_{m+1} - q_m) - u_{m-\frac{1}{2}}(q_{m+1} - q_m)}{\Delta x} = (\overline{uq})_x \bigg|_m
\]

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:

\[
\zeta_t = - (Ro(u\zeta)_x + \delta + R^\beta \overline{\nu}) \quad (18)
\]

\[
\delta_t = - (Ro(u\delta)_x - \zeta + R^\beta \overline{\nu} + \theta_{xx}) \quad (19)
\]

\[
\theta_t = - (Ro(u\theta)_x - Rou\overline{\nu} + R^\beta \delta) \quad (20)
\]

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 \( 1 + \rho^1 \) (where \( \rho = c_{\text{max}} \Delta t / Ax \)) 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 \( \rho < 1 \) 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.C.s.
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 CONTROLLING THE FLOW OF CALCULATIONS
LINEAR .TRUE. Ignore nonlinear terms
IDO .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, NHOV, NTHOV Control for Hovmöller diagrams.
VARIOUS CONSTANTS, PARAMETERS AND SCALES

\[ \pi = 3.14159265 \]

\[ \text{FCOR} = 1.0 \times 10^{-4} \text{ (Coriolis parameter)} \]

\[ \text{BETA} = 1.0 \times 10^{-11} \text{ (Beta parameter)} \]

\[ \text{UBAR} \text{ Mean Zonal Wind} \]

\[ \text{UO} \text{ Nondimensionalized UBAR} \]

\[ \text{FIBAR} \text{ Mean Geopotential} \]

\[ \text{FI0} \text{ Nondimensionalized FIBAR} \]

\[ \text{RF, RO, RB} \text{ Nondimensional numbers (Froude, Rossby, Beta; see text)} \]

\[ \text{GRAV} = 9.81 \text{ Gravitational acceleration} \]

\[ \text{SXL, SXT, SXV, SXDV, SXFI} \text{ Scales for length, time, velocity, vorticity (and divergence) and geopotential.} \]

INDEPENDENT VARIABLES, INCREMENTS, GRIDSPECS, ETC.

\[ \text{REAL X(0:NPX), T(0:NPT)} \text{ Spatial and temporal independent variables} \]

(\text{required for convenience in plotting results})

\[ \text{NX} \text{ Number of points in the spatial domain} \]

\[ \text{NXP1} = \text{NX} + 1 \]

\[ \text{NSTEPS} \text{ Number of timesteps in run} \]

\[ \Delta x, \text{ Grid distance} \]

\[ \Delta t, \text{ Timestep} \]

DEPENDENT VARIABLES

\[ \text{REAL U(0:NPX), V(0:NPX)} \text{ Horizontal velocities} \]

\[ \text{REAL FI(0:NPX), VORT(0:NPX), DIV(0:NPX)} \text{ Geopotential, vorticity, divergence} \]

\[ \text{REAL FI0L0D(0:NPX), VOROLD(0:NPX), DIV0L0D(0:NPX)} \text{ Old values of \( \Phi, \zeta, \delta \)} \]

(\text{Old values may not be required but are included for convenience})

\[ \text{REAL PSI(0:NPX), CHI(0:NPX)} \text{ Stream function, velocity potential} \]

RIGHT-HAND SIDES (AT TWO TIMES)

\[ \text{REAL RHS1V(0:NPX), RHS1D(0:NPX), RHS1C(0:NPX)} \text{ New values} \]

\[ \text{REAL RHS2V(0:NPX), RHS2D(0:NPX), RHS2C(0:NPX)} \text{ Old values} \]

(\text{Old values may not be required but are included for convenience})

ENERGY QUANTITIES

\[ \text{REAL KEROT(0:NPT), KEDIV(0:NPT), KETOT(0:NPT)} \]

Eddy rotational, divergent and total kinetic energy

\[ \text{REAL APE(0:NPT), APLUSK(0:NPT)} \]

Eddy available potential and total (APE+KE) energy

\[ \text{REAL SOURCE(0:NPT), DOTAPK(0:NPT)} \]

Conversion from zonal mean, and rate-of-change of eddy energy.
MID-POINT VALUES OF DEPENDENT VARIABLES (FOR PLOTTING)

REAL PTU(O:NPT),PTV(O:NPT),PTFI(O:NPT),PTVORT(O:NPI),PTDIV(O:NPT)

WORKING SPACE

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

ASCII STRING FOR INPUT FILE GUIDEDWORD

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-1 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 Δx = 200 km; Channel length L = 10000 km; Timestep Δt = 100 sec.

Example (1): A Rossby Wave

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

\[ \Phi = \cos(2\pi x/L) \quad u = 0 \quad v = \Phi_y \quad \text{(ICNUM=2)} \]

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
conditios are nondivergent. In figure 1a we show a plot of height against x and t (a so-called Hovmøller diagram). The westward movement of troughs and ridges is clear. The corresponding solution for the same initial conditions but with a mean zonal wind \( \bar{u} = 100 \text{ m/s}^{-1} \) 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): \( f = 1.1 \times 10^{-4} \text{s}^{-1}; \beta = 1.1 \times 10^{-11} \text{m}^{-1} \text{s}^{-1}; k = 2\pi/L, L = 1.1 \times 10^7 \text{ m}; \phi = 1.1 \times 10^5 \text{m}^2 \text{s}^{-2} \)). 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:

\[
\begin{align*}
(2a) & \quad \phi = \cos(2\pi x/L) \quad ; \quad u = 0 \quad ; \quad v = 0 \quad (\text{ICNUM} = 1) \\
(2b) & \quad \phi = \cos(2\pi x/L) \quad ; \quad u = 0 \quad ; \quad v = \phi_x \quad (\text{ICNUM} = 2) \\
(2c) & \quad \phi = \cos(2\pi x/L) \quad ; \quad u = \phi \quad ; \quad v = 0 \quad (\text{ICNUM} = 3)
\end{align*}
\]

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).
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 \( \bar{u} = 100 \text{ m s}^{-1} \); the initial conditions are (2a) above (ICNUM=1) and the model is integrated for 4000 timesteps (\( \Delta t = 100 \text{ s} \), so forecast length is about 4\( \frac{1}{2}\) 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.
Figure 3. Eddy energy for the initial conditions ICNUM = 3
(a) Kinetic energy  (b) Available potential energy
(c) Total eddy energy
Figure 4. Eddy energy in the presence of a zonal flow ($U_{BAR} = 100$ 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-inertial waves before your very eyes.
(2): Switch on IHOV to generate Hovmöller diagrams for these.
(3): Run with various values of At 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).

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
\[
\mathbf{V} = -\nabla_p + \mathbf{V}_I; \quad \nabla_p = k \times \nabla \phi; \quad \mathbf{V}_I = \mathbf{V}_x
\]
and ignore the latter where it appears undifferentiated. We now assume that
\( Ro \ll 1, \quad R_f \ll 1 \) 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):

\[
\begin{align*}
\frac{\partial \phi}{\partial t} + \bar{u} \frac{\partial \phi}{\partial x} + \beta v + f \phi &= 0 \\
-f \phi + \phi_{xx} &= 0 \\
\frac{\partial \phi}{\partial t} + \bar{u} \frac{\partial \phi}{\partial x} + f \bar{u} v + \phi \delta &= 0
\end{align*}
\]

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:
\[
\begin{align*}
\psi &= \phi^*/f; \quad v = \psi_x = \phi^*_x/f
\end{align*}
\]
Equations (A1), (A2) and (A3) are the quasi-geostrophic equations for our
one-dimensional model; although \( |\delta| \ll |\xi| \) 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:

\[
\left( \frac{\partial}{\partial t} + \bar{u} \frac{\partial}{\partial x} \right) \left[ \xi - (f/\phi) \phi^* \right] + \bar{v} \frac{\partial}{\partial y} \left[ f/\phi \right] = 0
\]

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):
\[
\delta_{xx} - (f^2/\phi) \delta = (\beta/\phi) \phi_x + (\bar{u}) \phi_{xxx}
\]
This is the quasi-geostrophic divergence equation and relates the divergence to
the geopotential. It is a \((1-\phi)\) 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 HELMID, analogous to POIS10 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 \( \chi \):

\[
\begin{align*}
\nabla^2 \psi &= \zeta \\
\nabla^2 \chi &= \delta
\end{align*}
\tag{B1}
\]

the velocities can be obtained by differentiation:

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

In the present case we assume that the dependent variables are independent of \( y \), are specified on the discrete grid \( \{x_0, x_1, x_2, \ldots, x_n\} \), and are periodic in \( x \). Thus we must solve two equations of the form

\[
\frac{d^2 \psi}{dx^2} = \rho \quad ; \quad \psi(0) = \psi(L)
\tag{B3}
\]

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

\[
\begin{align*}
-2\phi_1 + \phi_2 &= \rho_1 \\
\phi_1 - 2\phi_2 + \phi_3 &= \rho_2 \\
\phi_2 - 2\phi_3 &= \rho_3 \\
\vdots & \quad \vdots \\
\phi_n - 2\phi_{n+1} &= \rho_n \\
\phi_{n-1} + \phi_{n+1} &= \rho_n
\end{align*}
\tag{B4}
\]

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

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

does 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

$$u_n = \left( x_n - x_{n-1} \right) / \Delta x ; \quad v_n = \left( \psi_n - \psi_{n-1} \right) / \Delta x$$

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

References


STARTING FROM THE SHALLOW WATER EQUATIONS, THE VORTICITY AND DIVERGENCE EQUATIONS ARE DERIVED. THE BETA EFFECT THUS APPEARS EXPLICITLY AND SO ALL FURTHER Y-DEPENDENCE OF THE PERTURBATION QUANTITIES CAN BE SUPPRESSED. THE RESULT IS A SET OF THREE EQUATIONS WITH TWO INDEPENDENT VARIABLES, X AND T, AND THREE DEPENDENT VARIABLES, TO HIT,

\[ \frac{\partial \psi}{\partial x} = \text{VORTICITY} \]
\[ \frac{\partial \phi}{\partial x} = \text{DIVERGENCE} \]

PERIODIC BOUNDARY CONDITIONS ARE SPECIFIED, I.E. \( \psi(x+Lx_t, t) = \psi(x, t) \) FOR ALL PERTURBATION QUANTITIES \( \psi \).

INITIAL CONDITIONS ARE SET BY SPECIFYING THE VALUES OF \( \psi(x, 0) \) AND \( \phi(x, 0) \) FROM THESE THE INITIAL VALUES OF VORTICITY AND DIVERGENCE ARE IMMEDIATELY DERIVED. AFTER EACH FORWARD STEP WE MUST SOLVE THE POISSON EQUATIONS (WITH PERIODIC B.C.):

\[ \nabla^2 \psi = \text{VORTICITY} \]
\[ \nabla^2 \phi = \text{DIVERGENCE} \]

FROM THESE WE GET THE NEW VELOCITIES

\[ u = \frac{\partial \psi}{\partial x}, \quad v = \frac{\partial \psi}{\partial x} \]

THE SPATIAL GRID IS STAGGERED. THE VORTICITY AND DIVERGENCE ARE SPECIFIED AT "WHOLE POINTS", AS IS THE GEOPOTENTIAL. THE VELOCITIES ARE AT "HALF POINTS". VELOCITIES NEEDED AT WHOLE POINTS ARE GOT BY AVERAGING.

THE TIME-STEPPING IS DONE BY THE ADAMS-BASHFORTH SCHEME. THIS MEANS THAT WE NEED THE FORCING R.H.S.) TERMS AT TWO TIME-LEVELS.

INCLUDE 'DYNAMO.COM'

OPEN CONTROL (CARD) FILE AND OUTPUT (LINEPRINTER) FILE

OPEN(UNIT=5, DEVICE='DSK', ACCESS='SEQIN', MODE='ASCII', FILE='DYNAMO.CDS')
OPEN(UNIT=10, DEVICE='SCR', ACCESS='SEQOUT', MODE='ASCII', FILE='DYNAMO.LPT')

SET UP THE CONTROL PARAMETERS, CONSTANTS, ETC. FOR THE RUN

CALL GETPAR

SET UP THE INITIAL CONDITIONS

CALL ICS

CALL THE INITIALIZATION PROCEDURE

IF( INIT ) CALL LAPIN IST NOT IMPLEMENTED IN THIS VSN.
C DEFINE THE INDEPENDENT VARIABLES (FOR PLOT AXES)
DO 10 NN=0,NXPI
   X(NN) = (NN+1)*E+03 1 GRIDPOINTS (KM)
10 CONTINUE

DO 20 NN=0,NSTEPS
   T(NN) = NNDT/(60.*60.) 1 TIMESTEPS (HOURS)
20 CONTINUE

C DX = DX/SXL 1 NUNDIMENSIONAL

C READ MEAN FLOW PARAMETERS
READ(5,*) STRING,UBAR
READ(5,*) STRING,FIBAR
U0 = UBAR/SXL
F10 = FIBAR/SXI

C RO = SXV/(FIBOR*SXL) 1 ROSSBY NUMBER
C RB = BETA*SXL/FCOR 1 MEASURE OF BETA EFFECT
C RF = FIBOR/(FIBOR*SXL)**2 1 RECIPROCAL OF PROUDE NO.

C WRITE(6,907) UBAR,FIBAR,RO,RB,RF
907 FORMAT(6,907) UBAR,FIBAR,RO,RB,RF
   " MEAN ZONAL WIND UBAR = 'Fb.a', 'W/S'
   " MEAN GEOPOTENTIAL', 'Fb,0.' (M/S)**2 '/
   " ROSSBY NUMBER: ',1PE10,1/
   " RATIO RD 1 '1PE10.1/
   " RATIO RF 1 '1PE10.1/

C WRITE(6,9901) ICNUM
9901 FORMAT(' INITIAL CONDITIONS: ICs = ',12)
C WRITE(6,9902) ICNUM
9902 FORMAT(' HOVMOELLER DIAGRAMS ',1,2,4)

C RETURN END

C-------------------------------------------------------------
C SUBROUTINE ICS

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 KOUN1(20),PHASE(20),AMPL(20)
REAL COSN (0:201),SINN (0:201)
REAL COSNMH(0:201),SINNMH(0:201)

C NSPP = 0
C DEFINE THE INITIAL VELOCITIES AND GEOPOTENTIAL
C**** SOME ICS. ARE GIVEN HERE; IF YOU WANT TO DEFINE
C**** OTHER CONDITIONS, IDENTIFY THEM USING ICNUM > 4.

C=end
More accurate eastward gravity wave

Calculate the initial energies.

Calculate the initial values.

Calculate divergence and continuity equations.

More accurate westward gravity wave.

Save field values at a central point for plotting.

More accurate wind (initialization test).

Calculate the advection terms.

Perform single timestep for dynamo.

Include 'dynamo.com'.

Get rhs of vorticity, divergence and continuity equations at the present timelevel and put in rhsv, rhs1d and rhs3c (old values are in rms2 if nstep > 0).

Calculate the advection terms.

Calculate divergence and geopotential.

The Adams Bashforth scheme is used: the timestep is forward and the r.m.s. sides are estimated by linear extrapolation from the nth and (n+1)th timelevels.
C*** IF YOU WANT TO CHANGE THE TIME-SCHEME DO IT HERE.
DNL = 0.1
FN = 1.5
FNMI = -0.5
IF (NSTEP.GT.0) GO TO 250
FN = 1.0
FNMI = 0.0 1 FIRST STEP IS EULER FORWARD
250 CONTINUE

ACTUAL FORWARD STEP, EQUATIONS (18), (19), (20)
DO 300 NNX = 0, NXPI
   WORK1(NN) = VORT(NN) + DELT * (FN*RH31V(NN) + FNMI*RHS2V(NN))
   WORK2(NN) = DIV(NN) + DELT * (FN*RH31D(NN) + FNMI*RHS2D(NN))
   WORK3(NN) = FI(NN) + DELT * (FN*RH31C(NN) + FNMI*RHS2C(NN))
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK2(NN) = VORT(NN)
   WORK3(NN) = VORT(NN)
   WORK(NN) = VORT(NN)
   WORK(NN) = VOR...
CALCULATION OF VARIOUS FINITE DIFFERENCES

(1) CENTRED FIRST DIFFERENCE

SUBROUTINE DDXBAR(F,DIFF,N,DELTA)
DIMENSION F(0:N),DIFF(0:N)
DO 10 N=1,N
DIFF(N) = ( F(N+1) - F(N-1) ) / (2.*DELTA)
10 CONTINUE
CALL ENDS(DIFF)
RETURN

(2) FORWARD FIRST DIFFERENCE

ENTRY DDXF(F,DIFF,N,DELTA)
DO 20 N=1,N
DIFF(N) = ( F(N+1) - F(N) ) / DELTA
20 CONTINUE
CALL ENDS(DIFF)
RETURN

(3) BACKWARD FIRST DIFFERENCE

ENTRY DDXB(F,DIFF,N,DELTA)
DO 30 N=1,N
DIFF(N) = ( F(N) - F(N-1) ) / DELTA
30 CONTINUE
CALL ENDS(DIFF)
RETURN

(4) CENTRED SECOND DIFFERENCE

ENTRY DDXS(F,DIFF,N,DELTA)
DO 40 N=1,N
DIFF(N) = ( F(N+1) - 2.*F(N) + F(N-1) ) / (DELTA**2)
40 CONTINUE
CALL ENDS(DIFF)
RETURN

SUBROUTINE POISIO(PHI,RHO,N,DELTA)

SOLVE A POISSON EQUATION IN ONE DIMENSION WITH PERIODIC BOUNDARY CONDITIONS
LAP(PHI) = RHO
WHERE LAP IS THE LAPLACIAN OPERATOR

REAL PHI(0:N),RHO(0:N)
C ASSUME THE END VALUES VANISH (THEY ARE ARBITRARY).
PHI(0) = 0.
CALCULATE THE FIRST VALUE DELSQ = DELTA**2
SUM = 0.0
DO 100 N=1,N
SUM = SUM + RHO(N)*N
100 CONTINUE
PHI(1) = (SUM/N)*DELSQ

LOOP FOR THE OTHER VALUES DO 200 N=2,N
PHI(N) = RHO(N)*N*DELSQ + PHI(N-1) - PHI(N-2)
200 CONTINUE

CHECK R.H. END IS APPRX ZERO (MEASURE OF ROUNDOFF)
TYPE 99961, ( N,NPHI(0:N),NNN1,N )
99961 FORMAT(* PHISO/*(15,1PE1.2))
FILL IN END VALUES
CALL ENDS(PHI)
RETURN

END

AVERAGING OPERATORS (FORWARD AND BACKWARD)

SUBROUTINE XMEAN(F,FBARX,N)
DIMENSION F(0:N),FBARX(0:N)
DO 10 N=1,N
FBARX(N) = ( F(N) + F(N+1) ) / 2.0
10 CONTINUE
CALL ENDS(FBARX)
RETURN

ENTRY XMEAN(F,FBARX,N)
DO 20 N=1,N
FBARX(N) = ( F(N) + F(N-1) ) / 2.0
20 CONTINUE
CALL ENDS(FBARX)
RETURN

SUBROUTINE ENDS(A,N)
FILL END VALUES OF A PERIODIC ARRAY
REAL A(0:N)
A(0) = A(N)
A(N+1) = A(1)
RETURN

END
SUBROUTINE VECCONIC(A,B,1:1:12)
MULTIPLY A REAL VECTOR A(1:1:12) BY A
CONSTANT C AND STORE IN MATRIX B.

REAL A(1:1:12), B(1:1:12)
DO 10 K=1:12
   B(K) = C*A(K)
10 CONTINUE
RETURN
END

SUBROUTINE MEMOVE(FROM,TO,NN,NX)
MOVE AN ARRAY IN CORE
REAL FROM(NN:NX), TO(NN:NX)
DO 100 K=1:NN,NX
   100 TO(K)=FROM(K)
RETURN
END

SUBROUTINE PLOTLN(PX,PY,K1,K2,PYMIN,PYMAX,MESAGE)
LINE PRINTER PLOT OF PY(I) VS. PX(I) FOR K1,K2
WRITTEN BY J HAMILTON; MODIFIED BY PETER LYNCH.

DIMENSION PX(K1:K2),PY(K1:K2),MESAGE(2)
DIMENSION ZXSCALE(100),ZYSCALE(100)
DIMENSION IGRAPH(100:103)
DATA IBLANK/*4H,/*3STAR/*4H,/*4//,/*4//,*4//,*4//,*4I
DATA IXMIN/2/,IXMAX/102/,IYMIN/2/,IYMAX/42/

KCNAME=I
ICHANL=KCNAME
IF(ICHANL.LT.0) ICHANL=ICHANL
WRITE(KCHANL,99999) MESAGE
99999 FORMAT(/'70X,2A3)

C*CHECK THERE ARE SUFFICIENT POINTS IN THE ARRAYS FOR PLOTTING
 IPINDEX=(K2-K1+1)
 IF(IPINDEX.LT.0) GO TO 15
 WRITE(ICHANL,10)
10 FORMAT(/'54H5,'ERROR IN CALL TO **PLOTLN** NOT ENOUGH POINTS')
RETURN

C*FIND MAXIMUM AND MINIMUM VALUES OF "Y"
ZXMAX = PX(K1)
ZYMIN = PX(K1)
DO 20 J=1:2
   IF(PY(J).GT.ZYMAX) ZYMAX=PY(J)
   IF(PY(J).LT.ZYMIN) ZYMIN=PY(J)
20 CONTINUE

C*FIND MAXIMUM AND MINIMUM VALUES OF "X"
25 CONTINUE
ZXMAX = PY(K1)
ZYMIN = PY(K1)
DO 25 J=1:2
   IF(PY(J).GT.ZYMAX) ZYMAX=PY(J)
   IF(PY(J).LT.ZYMIN) ZYMIN=PY(J)
25 CONTINUE

C*RANGES OF X AND Y = PRINT MESSAGE
IF(ZYMIN.NE.ZYMAX) AND (ZYMIN.NE.ZYMAX) GOTO 35
WRITE(ICHANL,30) ZYMIN,ZYMAX,ZYMIN,ZYMAX
30 FORMAT('DUPLICATE CALL TO **PLOTLN**')
   1 RANGE OF X: /*1PE12.5,E12.3
2 RANGE OF Y: /*1PE12.5,E12.3
RETURN

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 ZYFACT = 100.0/(ZMAX-ZMIN)
   ZYFACT = 40.0/(ZMAX-ZMIN)
C*X AXIS SCALE
   ZXMAX=(ZMAX-ZXMIN)/10.0
   DO 40 J=1,11
      ZXSCALE(J)=FLOAT(J)*ZXMAX + ZXMIN
40 CONTINUE
C*Y AXIS SCALE
   ZYMAX=(ZMAX-ZYMIN)/10.0
   DO 45 J=1,11
      IREV=12-J
      ZYSCALE(J)=FLOAT(J)*ZYMAX + ZYMIN
45 CONTINUE
C*CLEAR THE GRAPHICAL AREA
50 DO 50 JX=1,102
   IGRAPH(JX,JY)=1BLANK
50 CONTINUE
C*PRINT X AXIS AT ZERO (IF POSSIBLE)
IF((ZYMIN.GT.0.0).OR.(ZYMAX.LT.0.0)) GO TO 60
   YY = (-2.0-ZYMIN)*ZYFACT
   JY = YY/0.5
   IF(JY.LT.IYMIN) YMAX=IYMIN
   IF(JY.GT.IYMAX) IYMAX=JY
   DO 55 JX=1,102
      IGRAPH(JX,JY)=1ILINE
55 CONTINUE
C*PRINT Y AXIS AT ZERO (IF POSSIBLE)
60 CONTINUE
IF((ZXMIN.GT.0.0).OR.(ZXMAX.LT.0.0)) GO TO 70
   XX = (-2.0-ZXMIN)*ZXFACT+2.0
   JX=XX/0.5
   IF(JX.LT.IXMIN) IXMAX=IXMIN
   IF(JX.GT.IXMAX) IXMIN=JX
   DO 75 JY=1,102
      IGRAPH(JX,JY)=1I LINE
DO 65 JY=2,42
   IGRAPH(1X,JY) = IYLINE
65 CONTINUE
C
C#AXIS (TOP AND BOTTOM)
70 CONTINUE
   DO 75 IX=2,102
      IGRAPH(IX, 2)=IYLINE
      IGRAPH(IX,42)=IYLINE
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)
90 CONTINUE
   DO 85 IY=2,42
      IGRAPH( 2, IY)=IYLINE
      IGRAPH(102, IY)=IYLINE
85 CONTINUE
   DO 90 IY=2,42,4
      IGRAPH( 2, IY)=IYLINE
      IGRAPH(102,1Y)=IYLINE
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=0,K
      XX = (PX(IPT)-ZMIN)*ZFACT+2.0
      IF(IX,L1,IXMIN) IX=IXMIN
      IF(IY,LT,IXMAX) IX=IXMAX
      YY = (PY(IPT)-ZMIN)*ZFACT
      IF(IY,LT,YY+0.5) YY=Y+0.5
      IF(IY,LT,YYMAX) YY=YYMAX
      IGRAPH (IX,YY) = ISTAR
135 CONTINUE
C
C#JOIN UP THE POINTS WITH LINES
   IF(IPT,GT,K1) GOTO 130
   IF(ICHANL,LT,0) GOTO 130
   IX=IX
   IY=IY
   TX2=IXOLD
   IY2=IYOLD
   XX1 = IX1
   YY1 = IY1
   XX2 = IX2
   YY2 = IY2
   IDELA = (IX2-IX1)*(IX2-IX1)
   IDELY = (IY2-IY1)*(IY2-IY1)
   IF (IDELA,GT,100) GOTO 110
C
SLOPE = 0.0
TEST = ((XX2-XX1) = (XX2-XX1)
IF (TEST,GT,0) SLOPE = (YY2-YY1) / (XX2-XX1)
IF (IX1,LT,XX2) 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
      IGRAPH(JX,JY),EQ,1BLANK
   IGRAPH(JX,JY)=IDDOT
105 CONTINUE
GOTO 130
C
110 CONTINUE
SLOPE = 0.0
TEST = ((YY2-YY1) = (YY2-YY1)
IF (TEST,GT,0) SLOPE = (XX2-XX1) / (YY2-YY1)
IF (IX1,LT,XX2) GOTO 115
IMIN = IY1
IMAX = IY2
GOTO 120
115 IMIN = IY2
IMAX = IY1
120 CONTINUE
   DO 125 JY = IMIN,IMAX
      YY = YY1
      XX = XX1 + SLOPE * (YY = YY1)
      JX = JX
      IGRAPH(JX,JY),EQ,1BLANK
   IGRAPH(JX,JY)=IDDOT
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 IYOR=1,10
      IF(4*IYOR-2) IY2=IY1+1
      IY2=IY1+1
      IY2=IY1+1
      WRITE([CHANL,40],[YSCALE(YOR)],(GRAPH(1X,1Y1),1X=2,100))
      FORMAT((I10,[P16,2,5X,10(I1))))
      WRITE([CHANL,145],[GRAPH(1X,Y2),1X=2,102])
      WRITE([CHANL,145],[GRAPH(1X,Y2),1X=2,102])
      WRITE([CHANL,145],[GRAPH(1X,Y2),1X=2,102])
      I10 FORMAT(2X,10(I1))
150 CONTINUE
   WRITE([CHANL,140],[YSCALE(11)],(GRAPH(1X,42),1X=2,102))
   WRITE([CHANL,155],[XSCALE(11),1X=11,11])
   I15 FORMAT(10X,11([P16,0,4]))
C
RETURN
SUBROUTINE HOVMOL
PRINT OUT A HOVMOELLER DIAGRAM FROM THE
VALUES STORED ON FILE 10
DIMENSION CHAR(I)
DATA CHAR(/,O,,.,'l,,','2,,','3',','4',','5*,
'x .',),'b',','7.,','8,,','9',','0/)
NXPTS = NXPI/NXHOV
IF(NXPTS.GT.121)
RETURN
CLOSE (UNIT=10)
OPEN(UNIT=10,DEVICE='SCRH',ACCES='SEQIN',MODE='BINARY',FILE='DYNADO,DAT',DISPOSE='DELETE')
DO 1000 NSTEP = 0,NSTEPS,NXHOV
READ(10) FI
IF(NSTEP.GT.0) GO TO 200
DEFINE ZERO AND SPACING
ZMAX = ABS(FL0)
DO 100 NN = 0,NX,NXHOV
ABSFI = ABS(FI(NN))
IF(ZMAX.LT.ABSFI) ZMAX = ABSFI
CONTINUE
ZERO = 0.
SPACE = ZMAX/5. CHANGE AS REQUIRED
IF(SPACE,LT,1,E-30) SPACE = 1.
WRITE(6,99001)
99001 FORMAT(/)
DO 200 NN = 0,NXPI,NXHOV
FINN = F(NN)
IND = MOD(INT((FINN-ZERO)/SPACE+2000.,20)+1
ZEBRA(NN) = CHAR(IND)
CONTINUE
WRITE(6,99001)
999 FORMAT(1X,16,12:1A1)
CONTINUE
WRITE(6,99001)
RETURN
END