FILE.

IOS

DEACON LABORATORY

A FREE SURFACE BRYAN-COX-SEMTNER MODEL

BY
P.D. KILLWORTH, D. STAINFORTH
D.J. WEBB & S.M. PATERSON

REPORT NO. 270

1989

Natural
Environment
Research
Council

INSTITUTE OF
OCEANOGRAPHIC SCIENCES
DEACON LABORATORY

INSTITUTE OF OCEANOGRAPHIC SCIENCES

DEACON LABORATORY

**Wormley, Godalming,
Surrey, GU8 5UB, U.K.**

**Telephone: 0428 79 4141
Telex: 858833 OCEANS G
Telefax: 0428 79 3066**

Director: Dr. C.P. Summerhayes

Natural Environment Research Council

INSTITUTE OF OCEANOGRAPHIC SCIENCES

DEACON LABORATORY

REPORT NO. 270

A free surface Bryan-Cox-Semtner model

P.D. Killworth[1+2], D. Stainforth[2], D. J. Webb[1]

and S. M. Paterson[3]

1 Institute of Oceanographic Sciences
Deacon Laboratory,
Wormley, Godalming, Surrey GU8 5UB

2 Hooke Institute for Atmospheric Research
Department of Atmospheric, Oceanic and
Planetary Physics, University of Oxford,
Parks Road, Oxford OX1 3PU

3 Data Connection Ltd.,
100 Church Street,
Enfield, Middlesex

1989

This work has been carried out with the support of the Procurement Executive, Ministry of Defence

DOCUMENT DATA SHEET

AUTHOR KILLWORTH, P.D., STAINFORTH, D., WEBB, D.J. & PATERSON, S.M.	PUBLICATION DATE 1989
TITLE *A free surface Bryan-Cox-Semtner model*	
REFERENCE *Institute of Oceanographic Sciences Deacon Laboratory Report, No.270,* (184pp)	

ABSTRACT

A version of the Cox numerical ocean general circulation model, adapted to include a free surface, is described and documented in detail. The model is designed for the following uses: tidal studies (a tidal option is explicitly included); assimilation of satellite altimetric data (since the surface elevation is now a prognostic variable); and in situations where accurate relaxation to obtain the stream function in the original model is too time-consuming.

The model has been produced using the Cox update scheme, making it simple to use by modellers already in a position to use the Cox model. A full listing is, however, also included. Example output is provided, and computations are compared with the equivalent Cox runs.

ISSUING ORGANISATION Institute of Oceanographic Sciences Deacon Laboratory Wormley, Godalming Surrey GU8 5UB. UK.	TELEPHONE 0428 79 4141 TELEX 858833 OCEANS G TELEFAX 0428 79 3066
KEYWORDS OCEAN MODEL MODELLING	CONTRACT PROJECT PRICE £40.00

Copies of this report are available from:
The Library, Institute of Oceanographic Sciences Deacon Laboratory.

CONTENTS

A. Introduction

The standard Oceanic General Circulation Model (OGCM) used by ocean modellers is that due originally to Bryan (1969) and programmed by Cox. A later adaptation of the code to vector processors by Semtner (1974) was used almost exclusively until recently when a combination of Cray and Cyber code, adjusted via a small update program, was introduced by Cox (1984). This has become the *de facto* standard for modellers, since it permits arbitrary bottom topography, coastline orientation, and forcing.

Any numerical model requires a set of design choices. Among the original choices made by Bryan and later co-workers was that the model should be integrable for times of order thousands of years, so that climate problems would be investigable. All finite-difference numerical models possess a series of constraints on the permissible timesteps. These typically take the form that advective and wave processes must not move more than one grid point per timestep (the Courant-Friedrichs-Lewy criterion, for example), or that diffusive processes must not spread more than a grid point per timestep. Bryan wished to minimise the degree of restraint in the system of equations for his model. The free surface equations permitted surface gravity waves, whose speed was of order $(gH)^{1/2}$ where g represents gravity and H some ocean depth. A typical value for this speed would be around 200 m s^{-1}. For a horizontal grid spacing of (30 km, 100 km, 250 km) this limits the timestep respectively to (150 s, 500 s, 1250 s) to within a factor of order unity. Unfortunately, with the exception of the widest grid spacing, the restrictions on the timestep given by other processes are far less stringent, often by a factor approaching 100.

Bryan therefore chose to eliminate the surface gravity waves from his system by replacing the free surface by a rigid lid. This well-known approximation has two effects: it increases the surface wave speed to infinity, and modifies certain other long wave dynamics (e.g. barotropic Rossby or planetary waves). However, this removed the strong constraint on timestep, and permitted steps of order several hours or longer to be made for the crude resolutions possible numerically in the early '70s.

A price had to be paid for this change. The barotropic flow had not been eliminated; it had merely been modified. To calculate it, a two-dimensional stream function had to be computed. The prognostic equation for the stream function involved the solution of a Poisson-like equation at each timestep. When the ocean basin has a simple shape, and the bottom topography has a simple structure, direct methods for solving for the stream function are available. For large models, with general coastlines and bottom topographies, relaxation methods have to be employed. Even with a good guess for the solution (acquired by extrapolating previous changes) the relaxation process has begun to take up steadily more and more of the total cpu time of the Cox model as grids have become finer, topographies more variable, and eddies more active. Instabilities (Killworth and Smith, 1984; Bonsell, 1989; Killworth, 1987) have begun to be found in the relaxation process.

Thus one rationale for instigating the development of a free surface model was the realisation that the necessity to make several hundred relaxations of a stream function per timestep might need about as much computation as would the taking of several hundred small, surface gravity wave limited, timesteps for the barotropic component of a free surface model.

A second rationale was assimilation of altimetric and other data into ocean models, which will play an important part in the generation of ocean prediction models. The rigid lid assumption on the Cox code means that the (effective) surface pressure -

and hence the model free surface - can only be obtained as a diagnostic after all the other fields have been computed. Furthermore, absolute surface elevation, and its temporal variability, cannot be obtained at all from the Cox model, so that no direct comparison with satellite sea level variability measurements can be made. There is no direct way to assimilate surface elevation observations into the Cox model, even using computationally fearsome methods such as adjoint or variational systems, which are not yet feasible.

A third rationale was to permit tidal investigations within the Cox model. Although internal tides can easily be studied, surface tides do not, of course, exist in the original model. In such cases, the need for a small timestep is not a restriction, since tidal studies could not permit most implicit methods in the numerical integration, since this would cause numerical damping of the very phenomenon studied. One option in the code provides explicitly for tidal integration.

The proposed modification itself imposes constraints on the system. On the plus side, islands are extremly simple to handle. On the minus side, another timestep has been added to the system (which already possesses a baroclinic and a tracer timestep, which may be different), so that the integrations of the free surface and barotropic mode must be integrated into the exisiting code smoothly. As shown later, grid-splitting, a known problem with the B-grid formulation of Bryan, can cause difficulties in cases of strong channel flow and steep topography. Changing to a C-grid creates new difficulties, however, and has not been done in this model.

The idea used throughout this work is to retain as much of the original Cox code as possible, and include all modifications using the Cox update scheme. This scheme provides a variety of alternative Fortran codes, nested together, with markers placed in the right margin. The user selects these markers (e.g. cyclic conditions versus non-cyclic; Cyber or Cray; etc.) and also uses the scheme to add in extra lines of his own code to tailor the program to his own needs. A small Fortran program then accepts the markers and inserts, plus the Cox code, and *generates* a new Fortran program which is then compiled in the usual way. This system removes the possibility of a permanent - and possibly fatal - modification being made to the code by any user; all changes in a given run are available afterwards in an easily accessible form. The coding has assumed Cray architecture (e.g., islands are stepped around), since we lacked Cyber testing facilities; the changes for other architectures would be simple.

The plan of this report is as follows. The equations of motion are briefly discussed, followed by a short discussion of general numerical problems and vertical modal structure. The new barotropic part of the model is then examined, followed by the changes to the baroclinic model. Details of program flow, the new variables used, new code options, and memory requirements are then given. A discussion of the behaviour of the model under various circumstances is given, as a guide to its use. A complete (and compilable) list of the updates is provided, with annotations explaining the new and changed features. A full listing of the revised code is given, and some sample runs shown, with comparisons with the original Cox runs.

Bonsell, J. J., 1989: Modelling heat fluxes in the ocean. D. Phil. Thesis, University of Oxford, 250 pp.

Bryan, K., 1969: A numerical method for the study of the circulation of the World Ocean. J. Comp. Physics, 4, 347.

Cox, M. D., 1984: A primitive equation, 3-dimensional model of the ocean. GFDL Ocean Group Technical Report No.1, Geophysical Fluid Dynamics Laboratory/NOAA, Princeton University, Princeton, N. J. 08542, U.S.A.

Killworth, P. D., 1987: Topographic instabilities in level model OGCMs. Ocean Modelling, 75.

Killworth, P. D., and J. M. Smith, 1984: Gradual instability of relaxation-extrapolation schemes. Dynamics of Atmospheres and Oceans, 8, 185-213.

Semtner, A. J., 1974: A general circulation model for the World Ocean. UCLA Dept. of Meteorology Tech. Rep. No. 8, 99 pp.

B. Continuous equations

The basic equations of the model follow those in Cox (1984) closely, and will merely be summarised here. Let

$$m = \sec \phi$$

$$n = \sin \phi$$

$$f = 2\Omega \sin \phi \qquad (1)$$

$$u = a\dot{\lambda}/m$$

$$v = a\dot{\phi}$$

where ϕ is latitude, λ is longitude, and a is the radius of the earth. Define an advective operator Γ by

$$\Gamma(\mu) = ma^{-1}[(u\mu)_\lambda + (v\mu m^{-1})_\phi] + (w\mu)_z \qquad (2)$$

Here μ represents any scalar quantity. The momentum equations then become

$$u_t + \Gamma(u) - fv = -ma^{-1}(p/\rho_o)_\lambda + F^u \qquad (3)$$

$$v_t + \Gamma(v) + fu = -a^{-1}(p/\rho_o)_\phi + F^v \qquad (4)$$

where ρ_o is a reference density for water. F^u, F^v represent turbulent effects, and are given below. The local pressure p is given by the hydrostatic relation

$$p = p_s + \int_z^0 g\rho dz \qquad (5)$$

where p_s is the pressure at z = 0. For the free surface model, we define this to be

$$p_s/\rho_o = g\eta(\lambda,\phi,t) \qquad (6)$$

where η is the free surface elevation. The other equations remain unchanged. The continuity equation is

$$\Gamma(1) = 0 \qquad (7)$$

and the conservation of tracer T (which can be temperature, salinity, or some passive tracer injected in the system) is

$$T_t + \Gamma(T) = F^T \qquad (8)$$

where F^T is a shorthand for the diffusive and other effects acting on T (e.g. radioactive decay, scavenging, etc.). The equation of state is

$$\rho = \rho(\theta, S, z) \qquad (9)$$

where θ is potential temperature, S is salinity. Opportunity is given in the standard code for a variety of formulations for (9); the normal case is to use the polynomial fit of Bryan and Cox (1972). The turbulent mixing is dealt with by defining

$$\nabla^2\mu = m^2\mu_{\lambda\lambda} + m(\mu_\phi/m)_\phi \qquad (10)$$

Then the turbulent effects are

$$F^u = A_{MV}u_{zz} + A_{MM}a^{-2}[\nabla^2 u + (1 - m^2n^2)u - 2nm^2v_\lambda] \qquad (11)$$

$$F^v = A_{MV}v_{zz} + A_{MM}a^{-2}[\nabla^2 v + (1 - m^2n^2)v + 2nm^2u_\lambda] \qquad (12)$$

$$F^T = [(A_{TV}/\delta)T_z]_z + A_{TH}a^{-2}\nabla^2 T \qquad (13)$$

where A_{ab} is the mixing coefficient corresponding to

a: M momentum
T tracer

b: V vertical
H horizontal.

Vertical mixing is handled by an infinitely rapid vertical adjustment when convective overturn could be expected. Define

$$\delta = \begin{cases} 1 & \rho''_z < 0 \\ 0 & \rho''_z > 0 \end{cases} \qquad (14)$$

where ρ''_z is the vertical gradient ignoring compressibility. The actual mixing is done as in the original Cox version; more accurate methods are to be preferred (cf. Killworth, 1989).

The lateral boundary conditions are

$$u = v = T_n = 0 \qquad (15)$$

where n is a normal coordinate to the wall. At the surface,

$$\rho_o A_{MV}(u_z, v_z) = (\tau^\lambda, \tau^\phi)$$

$$A_{TV}T_z = \text{(surface flux of tracer T)} \qquad z = 0 \qquad (16)$$

$$w = \eta_t + uma^{-1}\eta_\lambda + va^{-1}\eta_\phi$$

The τ terms represent wind stress. At the bottom,

$$\rho_0 A_{MV}(u_z, v_z) = (\tau_B{}^{\lambda}, \tau_B{}^{\phi})$$

$$T_z = 0 \qquad\qquad z = -H \qquad (17)$$

$$w = -mua^{-1}H_{\lambda} - va^{-1}H_{\phi}.$$

From here on, the method of solution diverges from that of Cox. We do, however, maintain a split into barotropic and baroclinic modes. (The definition of these perforce differs slightly.) Define

$$u = U/H + u'; \quad v = V/H + v' \qquad (18)$$

where (U,V) is the vertically integrated (barotropic) mass flux

$$U = \int_{-H}^{\eta} u\,dz \qquad V = \int_{-H}^{\eta} v\,dz \qquad (19)$$

and (u', v') is the baroclinic flow, which possesses no depth average:

$$\int_{-H}^{\eta} u'\,dz = \int_{-H}^{\eta} v'\,dz = 0 \qquad (20)$$

We first integrate the mass continuity equation (7) w.r.t. z, from -H to η. Using (19), (16) and (17), this becomes

$$\eta_t + a^{-1}[mU_{\lambda} + m(Vm^{-1})_{\phi}] = 0. \qquad (21)$$

This equation is familiar from linear gravity wave theory, and yields a prognostic equation for η. We now integrate the momentum equations (3), (4) w.r.t. z, from -H to η. If the boundary conditions on w are then applied, we obtain

$$U_t - fV = -ma^{-1}gH\eta_{\lambda} + X' \qquad (22)$$

$$V_t + fU = -a^{-1}gH\eta_{\phi} + Y' \qquad (23)$$

where (X', Y') are initial definitions of residual forcings, (to be modified below), namely

$$X' = -ma^{-1}(\partial/\partial\lambda)\int_{-H}^{\eta} u^2 dz - a^{-1}(\partial/\partial\phi)\int_{-H}^{\eta} uv\,dz - ma^{-1}\int_{-H}^{\eta} dz \int_{z}^{0} g\rho_{\lambda} dz + \int_{-H}^{\eta} F^u dz \qquad (24)$$

$$Y' = -ma^{-1}(\partial/\partial\lambda)\int_{-H}^{\eta} uv\,dz - a^{-1}(\partial/\partial\phi)\int_{-H}^{\eta} v^2 dz - a^{-1}\int_{-H}^{\eta} dz \int_{z}^{0} g\rho_{\phi} dz + \int_{-H}^{\eta} F^v dz \qquad (25)$$

The idea now is to remove the friction terms explicitly from (X', Y') and retain them in the integration that follows, thus providing some smoothing. There is freedom to choose the precise form of the friction terms; various forms have been tried, including no explicit friction at all, with few obvious differences.

We put

$$X' = X + A_{MH}a^{-2}H[\nabla^2(U/H) + (1-m^2n^2)U/H - 2nm^2(V/H)_\lambda] \qquad (26)$$

$$Y' = Y + A_{MH}a^{-2}H[\nabla^2(V/H) + (1-m^2n^2)V/H + 2nm^2(U/H)_\lambda] \qquad (27)$$

so that (22), (23) become

$$U_t - fV = -ma^{-1}gH\eta_\lambda + A_{MH}a^{-2}H[\nabla^2(U/H) + (1-m^2n^2)U/H - 2nm^2(V/H)_\lambda] + X \qquad (28)$$

$$V_t + fU = -a^{-1}gH\eta_\phi + A_{MH}a^{-2}H[\nabla^2(V/H) + (1-m^2n^2)V/H + 2nm^2(U/H)_\lambda] + Y \qquad (29)$$

which take the form of the linear barotropic free surface momentum equations, plus some horizontal smoothing through the A_{MH} terms.

Note that the wind stress forms a direct part of the forcing of these barotropic equations, through the integral of $A_{MV}(u_{zz}, v_{zz})$.

Bryan, K. and M. D. Cox, 1972: An approximate equation of state for numerical models of ocean circulation. J. Phys. Oceanogr., 2, 510-514.

Cox, M. D., 1984: see Section A.

Killworth, P. D., 1989: On the parameterisation of deep convection in ocean models. In: Proceedings of the 'Aha Huliko'a Winter Workshop, Hawaii Institute of Geophysics, ed. P. Muller.

C. Numerical implementation

The equations of Section B have been implemented by removing the Cox routine RELAX (which solved for the barotropic streamfunction), and replacing it by a new routine TROPIC, which integrates the barotropic flow field and surface elevation forward in time, using several small timesteps. These changes, and the others necessitated by the replacement, are now described. The Cox nomenclature and variable names have been retained as far as possible. A detailed discussion of each change in the code is given in Section J.

(a) Vertical normal modes

It is first necessary to recall properties of the linearised system of equations for a flat bottom. As is well known (e.g. Gill, 1982), the response can be described by a linear superposition of orthogonal vertical normal modes. The first of these is a barotropic mode; the rest are baroclinic. The properties of the baroclinic modes are almost identical whether or not the ocean possesses a rigid lid or a free surface; but the barotropic mode depends on the surface boundary condition and needs some discussion.

When a free surface is permitted, the barotropic mode has its traditional properties. The horizontal velocity is almost independent of depth, and the vertical velocity increases approximately linearly from zero at the floor to a maximum at the free surface. The density field also varies spatially and temporally (though weakly). With the rigid lid assumption, however, the barotropic mode's properties alter. Its flow is entirely horizontal, and exactly independent of depth. The flow is nondivergent (hence the use of a streamfunction in the Cox code), and the density field is invariant.

These two sets of properties are very different, and affect the behaviour of the entire system of equations. In the rigid lid case, the natural time scale for motions of the barotropic mode is that for nondivergent planetary waves; this time scale is very slow, so that timestep restrictions on the barotropic mode are weak. Furthermore, since the barotropic mode is exactly that obtained by vertical integration, and the baroclinic mode exactly the remainder, then depth-averaged and depth-dependent flows are uncoupled in the linear regime. This permits separate timestep restrictions in the two cases.

With a free surface, two separate features appear. First, surface gravity waves are permitted, with high speeds, which forces small timesteps to be taken. Second, and potentially somewhat dangerously, the depth integrated flow is not precisely the barotropic mode, and hence not exactly orthogonal to the depth dependent structure.Thus the Cox method of integration over depth does not now entirely separate barotropic and baroclinic modes in the linear regime (nor, indeed, can this separation easily be achieved by any other means). The result is a possible <u>leakage</u> of energy and information between barotropic and baroclinic modes by any integration method which treats the depth independent and dependent modes separately. In particular, the Cox code integrates the density field in time separately from the depth independent mode. Now, however, part of the density field is due to the depth independent mode. This corrects the square of the natural wave speed, gH, by the small amount Γ where

$$\Gamma = \int_{-H}^{0} dz \int_{0}^{z} N^2 (z + H)/H \, dz \qquad (30)$$

(cf. Gill, 1982, for example) The speed given by (30) is of order the first baroclinic wave speed.

Nonetheless, this possible leakage can present difficulties, since it couples rapidly temporally varying barotropic phenomena with more slowly varying baroclinic features. Under the right conditions, this can provide positive feedback between the modes and lead to numerical instability. How this is handled is discussed below.

(b) Coupling

Eqns. (21), (28), and (29) form a complete set of barotropic equations. Since the barotropic flow varies on rapid time scales, we assume (as in the original model) that the terms comprising (X, Y) vary on the slower time scale. These include the density field, the vertical friction (not present in the barotropic field by definition), the nonlinear terms (which include baroclinic-baroclinic and barotropic-barotropic interactions, whose contribution is assumed to vary only slowly; they are weak anyway for all reasonable oceanic flows save strong channel flows).

The rationale for the system is simple. The baroclinic fields and the tracer equations are stepped forward in time one leapfrog timestep precisely as in the Cox code. In the process, the terms (X, Y) can be computed - they are done so already in the present code, save for the horizontal viscous terms. (X, Y) are assumed constant over the time Δt, which is split into many small steps $\Delta t'$. The barotropic equations (21, 28, 29) are now stepped $(\Delta t/\Delta t')$ times, each timestep being numerically stable for surface wave propagation. The number of these small timesteps and the computation involved, are of the same order as the number of iterations and the computation involved for the streamfunction evaluation when there are many iterations. For coarse grids with few iterations, however, rather more resources are required for the free surface version.

(c) The numerical details of the barotropic system (subroutine TROPIC)

The barotropic system satisfies (21), (28), and (29). These equations are implemented in routine TROPIC using the same numerical techniques as in the original Cox code. The free surface η is stored at tracer points, and U, V at velocity points. It is assumed that the reader is familiar with the Cox code, and the standard notation for numerical operators.

Two different timestepping schemes have been implemented, with an option to select the appropriate one for the circumstances required. These are: full Euler backward (i.e. predictor-corrector) and Euler forward- backward. The former would be used for most problems, but the latter would be preferred for tidal investigations for reasons explained below. We shall first describe the full Euler backward scheme.

We first predict intermediate values of η, U and V, (denoted by primes) using

$$(\eta' - \eta)/\Delta t + a^{-1}m[\delta_\lambda(\overline{U})^{\phi} + \delta_\phi(\overline{V/m})^{\lambda}] = 0 \tag{31}$$

$$(U' - U)/\Delta t - f(V' + V)/2 = -ma^{-1}gH\delta_\lambda(\overline{\eta}^{\phi}) + F^u \tag{32}$$

$$(V' - V)/\Delta t + f(U' + U)/2 = -a^{-1}gH\delta_\phi(\overline{\eta}^{\lambda}) + F^v \tag{33}$$

where F^u, F^v, are now the frictional terms worked out exactly as in Cox (1984; eqns. 52, 53), plus the forcings (X,Y). A semi-implicit scheme is used for the Coriolis terms; U', V' may simply be obtained from (32), (33). These intermediate values are then used to predict the new values η'', U'' and V'':

$$(\eta'' - \eta)/\Delta t + a^{-1}m[\delta_\lambda(\overline{U'}^{\phi}) + \delta_\phi(\overline{V'/m}^{\lambda})] = 0 \quad (34)$$

$$(U'' - U)/\Delta t - f(V'' + V)/2 = -ma^{-1}gH\delta_\lambda(\overline{\eta'}^{\phi}) + F^u \quad (35)$$

$$(V'' - V)/\Delta t + f(U'' + U)/2 = -a^{-1}gH\delta_\phi(\overline{\eta'}^{\lambda}) + F^v \quad (36)$$

Then η, U and V are overwritten by η'', U'', and V'', and the cycle is repeated for the next (barotropic) timestep. It is straightforward to prove that the code conserves mass and energy.

Many schemes could have been chosen for the barotropic timestepping. Some, like the standard leapfrog method used elsewhere in the Cox code, are more efficient (since they need only one computation of right-hand-sides per timestep). The rationale was as follows. The system (21), (28) and (29), no matter how implemented, gives a linear algebraic scheme for predicting η'', U'', and V'' from η, U and V. This scheme has (at least) three eigenvalues/eigenvectors. For numerical stability, all eigenvalues must be less than unity in modulus, naturally. The advantage of the Euler backward scheme is that of the three eigenvalues, only one attains unity, corresponding to the steady geostrophic mode (including the forcing in the definition, and ignoring viscosity). Thus for a given forcing, on an infinite f-plane, and without baroclinic integration, continued application of the Euler backward scheme will selectively damp out the other time-varying modes, leaving only the geostrophic mode (which can exist at all horizontal wavelengths). This is not necessarily the case for most other schemes of integration.

This is relevant if the barotropic equations are linked back to the baroclinic structure. Recall that the density field is integrated with a (long) baroclinic timestep. If all the free surface modes are present, then even with scale-selective damping, the barotropic field will possess a full spectrum of motions. These will provide a feedback on the barotropic timescale to the density field, and can lead to instability. (To see this, model the oceanic response as a slowly-varying baroclinic mode, with a rapidly-varying barotropic mode superposed, with mutual feedback. The rapid variations of the barotropic mode can be considered as white noise for the baroclinic mode. This white noise can provide forcing with components always in phase with the baroclinic mode, which can give gradual instability.)

When the Euler backward scheme is used, the non-geostrophic barotropic modes decay (slowly for larger length scales, more rapidly for smaller length scales) during the barotropic integration. The divergence of the geostrophic mode is small (of order β, the variation of Coriolis parameter with distance north). The divergence terms then give a forcing of the density field proportional to $\beta\rho_\lambda$

$$\rho_t + (A/a)\beta\rho_\lambda + \ldots$$

for some A. This takes the form of a long baroclinic planetary wave. The phase speed is enough to limit the length of the tracer timestep (typically to a few hours, rather than days, for coarse resolution) but remains stable. (It should be noted that forcing by $\beta\rho_\lambda$ should, for perfect stability in the baroclinic leapfrog scheme, use the current value of $\beta\rho_\lambda$, whereas the form of the Cox code implies that this is computed at the previous timestep. The code was experimentally modified to take this into account, with negligible effect; since rather more computation was involved for this modification, it was removed again. There may conceivably be cases where this effect becomes important.)

The forward-backward Euler scheme is provided if this damping of nongeostrophic modes is explicitly not required, e.g. for tidal calculations. Then (31) is used to predict η', the new value of η, and this is substituted in (32), (33) to provide the pressure gradients necessary to give U', V'. This scheme does not damp the non-geostrophic motions, and thus gives an accurate rendering for tidal purposes. However, the forcing back on the density field now more resembles white noise, and smaller baroclinic timesteps become necessary to ensure stability. Some guidelines on this are given in Section I.

Early tests of the barotropic model showed that, as anticipated, under certain circumstances the B-grid can split into two checkerboard subgrids with a separate time development. The result is a +/- structure in space. A staggered coastline could trigger the effect when viscosity was small, for example. The problem is well known in the atmospheric literature, and leads to a preference for C-grids. (These, however, have their own problems on rigid boundaries; it was also not our intent to modify the main Cox code more than necessary.)

Grid-splitting of the barotropic mode was not encountered often during testing of the code, save when using the tidal option. The method used to control it is detailed in the Appendix. Since the method takes about 5% of the computations in the barotropic subroutine, there is little need to use it unless difficulties are encountered.

Another potential instability mechanism, never encountered in testing, is that strong (U, V) (e.g. channel flow) imply the importance of the barotropic advection terms. Because these are implicitly calculated in the forcing (X, Y), they are evaluated at the current baroclinic timestep, and so remain constant during the barotropic integrations. Theoretically this process is weakly unstable, as with any simple advection problem. It is simple to extract the mean advection terms from (X, Y) also, and include them explicitly in the TROPIC integrations. We can provide details of this if required; no instability due to this cause was ever found.

(d) Modifications to the remainder of the Cox code.

The majority of the other changes to the code are detailed later; here only a brief overview is given. Essentially the changes use the fact that the barotropic flow is known exactly at u,v points, rather than as differentials of a streamfunction. We also now assume that the flow is confined to the depth range (-H, 0), i.e. assume that the surface elevation $\eta << H$. Since elevations are seldom as much as 1 m, and depths are of order 4000 m, this may only be a problem on very shallow continental shelves, where the model is not ideally suitable. Thus no account of the free surface is made in computing interior flows.

The main changes occur in subroutine CLINIC, which integrates the baroclinic velocities. Although the surface pressure is now known explicitly, we have chosen to ignore this in order to keep code changes to a minimum. Thus the baroclinic velocities use the same pressure formulation as in Cox. It is necessary to replace the rigid lid condition on w with

$$w = -ma^{-1}[\delta_\lambda \overline{U}^{\lambda} + \delta_\phi(\overline{Vm^{-1}}^{\phi})], \; z = 0 \qquad (37)$$

and to define values of u, v, and T 'above' the surface to take their values just below the surface, to maintain conservation properties. (Of course, there is not necessarily any global conservation of tracers or velocities, since the system permits vertical flow at the surface, and hence out of the basin. In practice this is not a problem, since the small omitted depth range from 0 to η contains the missing quantities and could be included in global conservation if required.)

The treatment of advection differs somewhat. Since the barotropic flow is known at (U, V) points, there is no need to define auxiliary quantities (Cox eqns. 47, 48). Indeed, with one exception detailed below, advective fluxes of horizontal momentum are now defined using the actual velocity field, without modification for fluxes, baroclinic terms, etc.. This implies that interaction with topography is now somewhat different. Topographical features are not well handled in the Cox code, and do not appear to conserve momentum totally. At the base of a cliff, for example, momentum is advected horizontally into the cliff in the original code. The vertical velocity at the summit of the cliff is related - not perfectly - to its analytical value, but the vertical advection of momentum at that point cannot in general balance the horizontal advection into the cliff lower down. Thus volume integral of momentum advection yields a source or sink in the original code near topography.

The new code modifies this; reference to Fig. 1 will prove useful. First, the vertical velocity at the cliff summit is now $-1/2\mathbf{u}.\nabla h$ (the factor of a half relating, as in the Cox code, to the fact that the horizontal mass flux near a boundary is the average of the real flux and zero). The vertical velocity remains zero, as in the Cox code, at the base of the cliff as indicated. Second, the excess horizontal flux of horizontal momentum is now returned explicitly as an addition to the vertical flux of horizontal momentum at the cliff summit, thus avoiding source/sink terms in the momentum balance. (The black horizontal fluxes in Fig. 1 are added to the vertical flux above.) Note that the diagnostic terms such as 'nonlinear exchange error' have not had this correction made, and so will appear much larger than is actually the case.

The grid-splitting of the barotropic mode (see Appendix) can also, under some circumstances, affect the baroclinic modes. It has long been known that when there is strong horizontal flow, abrupt topography, (e.g. sudden steps of 1000 m), and low dissipation, evaluation of the vertical velocity at (u, v) points in the Cox code shows evidence of +/- behaviour even far from the topography; note that the code normally only prints w at tracer points, which tends to be a well-behaved quantity. This is believed to be caused by the injection into the model of an effective delta-function source of momentum at the topography. This delta-function has spatial Fourier coefficients at all horizontal wavenumbers, including the +/- structure of the grid-splitting. If the flow past the topography is sufficiently strong, and the dissipative mechanisms sufficiently weak, this erroneous forcing can produce a slowly growing instability. (Note also that topography can induce very large local cell Reynolds and Peclet numbers, which may also lead to strong +/- behaviour.)

Now the extra degrees of freedom in the free surface version permit w at (u, v) points to become rather larger in such conditions. This can lead to disastrously high levels of noise in the solution, although many features remain quite acceptable. Linking the two subgrids back together in the tracer equations (the baroclinic version of the smoothing discussed in the Appendix) would be arduous numerically. We have provided a partial solution as one of the code options. This evaluates w at (u, v) points not by centred differences like (37), but by taking a weighted 4-point average of the evaluations of w at tracer points. This rather brutal technique largely eliminates +/- behaviour at momentum points, at the cost of introducing it at tracer points; some conservation properties are also lost.

It must be stressed that these circumstances (rapid barotropic flow, abrupt topography, and low dissipation), which might be relevant for eddy-resolving channel investigations, may not yield acceptable results using the free surface model, even with the code option discussed above. We believe this to be an essential feature of the B-grid, not of the particular coding used here. Thus the model should be used with caution in the above circumstances.

(e) Integral constraints

Subject to the free surface condition, the new code satisfies the same integral constraints as Cox (1984), with the obvious replacement of Cox's streamfunction formulation for the barotropic mode by the explicit barotropic kinetic and potential energy (the latter coming from the free surface). In particular, the baroclinic contributions remain identical. The changes due to the free surface are minor, and are not described here. Note, however, that the 'nonlinear exchange error' reported by the code, which is small in the Cox code, will normally be somewhat larger in the free surface version since vertical advective terms evaluated at the surface will be nonzero.

Cox, M. D., 1984: see Section A.

Gill, A. E., 1982: Atmosphere-Ocean Dynamics. Academic Press, New York, 662pp.

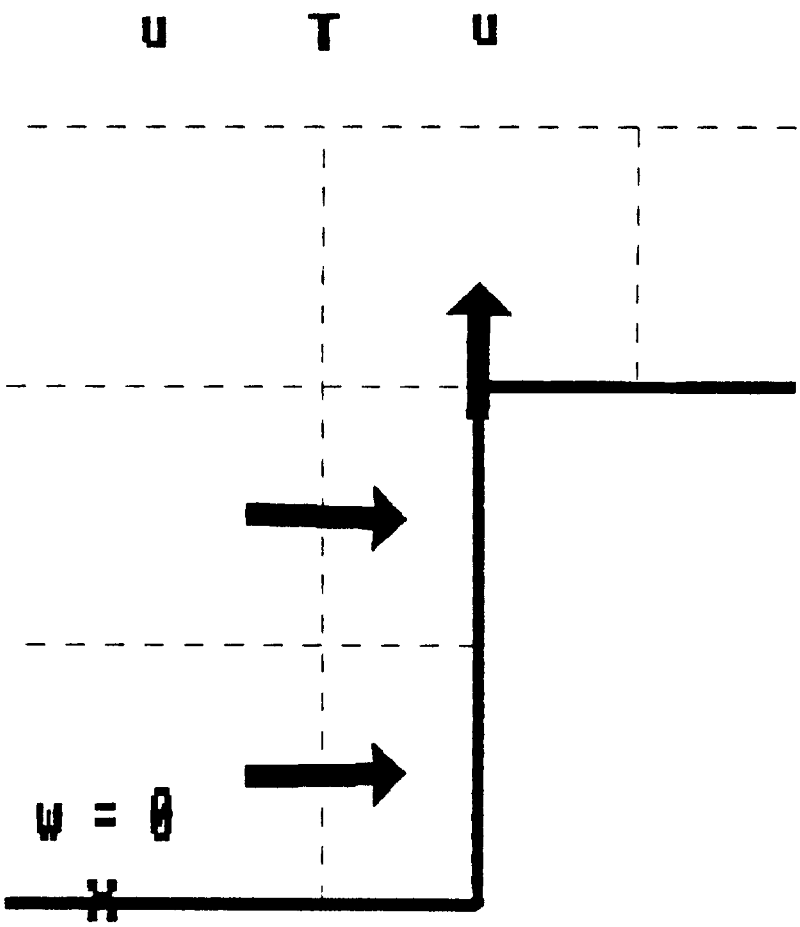


Fig. 1

D. Program flow

The FORTRAN code for the model still contains the main program OCEAN, together with seven main subroutines. The subroutine RELAX has been replaced by subroutine TROPIC. The functions of the subroutines remain identical to their Cox functions, except for the following changes.

CLINIC:

Called once per row by STEP, it computes the internal mode component of the u and v velocities as well as the initial estimates of the forcing for the barotropic mode for use by TROPIC later.

TROPIC:

Called once at the end of each timestep by STEP, it takes the barotropic forcing estimates computed in CLINIC, adjusts them so that the viscous terms become explicit, and integrates forwards a number of small barotropic time steps.

Although routines FINDEX and FILTER remain, no code has been provided for Fourier filtering in the new version of the code.

E. Disc I/O system

The disc input-output has been left as similar as possible to the system used by Cox. The slab data (cf. LABS in the code) and KONTRL are essentially unmodified, while KFLDS is now used for storing quantities mainly connected with subroutine TROPIC. The pattern of data flow during and between steps remains as before.

Some computer systems make it awkward to read and write from the same unit. To this end, a small modification has been made (no doubt already made by many users) so that data for restart are dumped at the end of a run to unit 23, not 21 as in the original.

An extra output unit, 8, is accessed in subroutine TROPIC, which gives a printout at specified intervals for the barotropic fields. It provides: timestep, kinetic energy, potential energy, total energy, and the basin-integrated surface elevation (which provides a mass error check, and should always be small).

F. Changes to the variable list

The Cox naming convention has been adhered to wherever possible, and the majority of the variables retain their original meanings and usage. Those relating to RELAX have disappeared. New variables concerning TROPIC have been added. A list is given below of new variables, etc., occurring in COMMON blocks; others which are used occasionally within the program are mainly self-explanatory.

1. Variables whose use has changed

NSSIF and NDICES, which serve as counters, have slightly modified uses; cf. the program listing in Section K.

2. New variables

(ISE, IEE)	: arrays holding the start/stop indices for ETA, for use in TROPIC
(ISU, IEU)	: as above, but for (U, V)
LSE	: array holding the number of start/stop segments for ETA at the given J
LSU	: as above, but for (U, V)
ISPL, JSPL	: arrays holding I, J land points for use with D option
NSPL	: number of I, J land points for use with D option
DTBT	: barotropic timestep (should be an integer fraction of DTUV)
WGHT	: the weighting ("α") used with the D option
XFBT, YFBT	: the barotropic forcing, computed in CLINIC
ETA	: the free surface elevation
UBT, VBT	: the (u, v) components of the barotropic flow
ZUENG, ZVENG	: arrays used to hold energetics of barotropic flow
FUWBT, FVNBT, FVSBT	: flux arrays used with M option
XF, YF	: the (u, v) components of the barotropic forcing, used in TROPIC
ETAD, ETAGRD	: intermediate arrays for timestepping ETA in TROPIC
UT, VT	: intermediate arrays for timestepping UBT, VBT in TROPIC
BHD, CHD, DHD, GHD, HHD	: arrays to speed viscous calculations in TROPIC
FUV	: array to compute semi-implicit Coriolis steps in TROPIC
SPLR	: array used in D option, for external ETA points

EM : mask array mainly used in D option smoothing

3. Variables removed

These include: FKMZ, ZTD, ZUSENG, ZVSENG, ZUNENG, ZVNENG, DTSF, C2DTSF, CRIT, SOR, P, PB. All WORKSPB space for RELAX has been changed for use in TROPIC.

G. Updating and code options

Cox's utility "UPDOC" has been used throughout the preparation of this program. Two versions of the code are provided. The first consists merely of the annotated, compilable updates (see Section J) which, if added to the other requisite input data, will convert the Cox code to a free surface version, compile and run it. The second is a complete listing of the new code, which can be used as the 'base' for further updates if preferred. The user should be aware that certain lines added later to the Cox code (e.g. those setting w to zero at the lowest grid point at (u, v) points) have had to be moved because of altered program flow logic, so that "standard" updates must be treated with caution!

The code options are loaded as in the Cox code, by a single record beginning in column one, of form

OPT=X,Y,Z,...

where X, Y, Z refer to possible single-letter option codes. Those supported are:

C : include comments

D : include the delplus-delcross smoothing; normally needed only with the tidal option G

G : 'tidal' option, using forward-backward barotropic integration to avoid any damping on free surface modes

K : run core contained - no disc I/O required

M : uses a 4-point average on w evaluations at (u, v) points. This option is an attempt to control grid-splitting in awkward cases, but is not totally efficient

O : cyclic conditions set east-west

S : symmetry conditions set at northern boundary

Code options which are <u>not</u> fully implemented are:

F : Fourier filtering. The Cox code for this remains, but no new code has been added to handle the new features

H : half-word mode running

Q : cyber-specific code

T : timing analyses

W : 'WHERE' statements used in place of IF

The code option "I" has now disappeared, since the free surface code handles islands quite normally, without needing to know their number (subject to certain arrays which hold I, J indices being filled).

The method of updating and modifying the code remains identical.

H. Memory requirements

The formulae here are computed in the same manner as Cox. The symbols used are:

IMT	: total number of east-west grid points
JMT	: total number of north-south grid points
KM	: total number of vertical grid points
LBC	: number of arrays kept as slab incidental data
NKFLDS	: number of IMTxJMT fields stored on unit 12 (normally 7)
C	: 0 if in disc I/O mode 1 if in core contained mode
D	: 0 if running nonsmoothed version (normal) 1 if running delplus-delcross smoothing

As before, the common block WORKSP is the largest single block of memory under most circumstances. WORKSP has two guises: the normal usage WORKSPA, in all routines but TROPIC, and WORKSPB, which occurs in TROPIC. The TROPIC usage can overwrite the previous values, since it is called but once by STEP, after all calls to CLINIC and TROPIC have been completed. The space taken by each is:

MEM(A) = (79+8*NT)*IMT*KM + (6+7*LBC)*IMT

MEM(B) = 7*IMT*JMT + 6*JMT + D*MSPL

where MSPL is the number of external points accessed in the ETA integration when the D option is selected. (The size of MEM(B) may be further reduced if the D option is not used, because the array EM is only used as a masking array in TROPIC; it is easy to provide an in-line statement using the depths at tracer points to do the same thing. Then one IMT*JMT array would be removed.)

The requirement for WORKSP is the maximum of MEM(A), MEM(B). (These figures are very similar to those for the Cox model; as a rough guide, if the Cox model fits into a given space, so will the free surface model if the D option is not chosen.) The other space requirements are the same as Cox's, e.g. the requirements for the virtual disc "BIG" are:

MEM(BIG)=20+NKFLDS*IMT*JMT+2*((NT+2)*IMT*KM)+LBC*IMT)*T

Thus the total memory requirement becomes

MEM(TOTAL) = Max[MEM(A), MEM(B)] + 4*IMT*JMT + C*MEM(BIG)

together with memory for object code, individual arrays, etc.

I. Comments on code and options

In the course of developing the free surface model, several test models were run under a variety of options and circumstances. Two of these are reported in Section L. A third, loosely termed an eddy-resolving model, was used to study channel geometry and abrupt topography. During these tests, a great deal was learned about the behaviour of the model, and this section discusses some of its features.

We consider first the options. Since the tidal option G is more restrictive on baroclinic timesteps than the default (Euler backward), there is no need to use it except in two cases.

The first case is if the resolution is sufficiently coarse that the 'double timestep' involved in Euler backward has not fully vectorised and the barotropic timestep is taking up a considerable fraction of the total computation. Then possibly having to reduce the tracer timestep a little, but only integrating a single timestep for the forward-backward scheme, may actually save on computation. The second case is, of course, when tidal calculations are required.

We have found that the G option is prone to grid-splitting (see Appendix), so that the D smoothing option must be applied in such cases. A small value of α around 0.1, is usually sufficient to suppress the splitting. No other case has been found where the D option is useful; indeed, in some cases it can generate a small amplitude +/- behaviour of its own. Thus the D option should be avoided except when running tidal cases.

The M option is provided for use in extreme cases. Such a case is the eddy-resolving model, which used a 32 x 32 x 5 grid in a channel geometry, with spacings of 0.25^o east and north, and a vertical spacing varying from 50 m at the surface to 4000 m at depth. The horizontal eddy viscosity and diffusion were (4, 1) x 10^6 cm^2s^{-1} respectively, with vertical coefficients both 0.1 cm^2s^{-1}. Timesteps were of half an hour (internal mode) and 45 seconds (barotropic mode). The topography was forced to be abrupt by the huge depth of the lowest grid point. East-west flows, along channel, reached about 30 cm s^{-1}.

The Cox model gave only marginally acceptable results on such an extreme case, with somewhat noisy tracer fields and very noisy w fields, especially at velocity points. Integrated quantities such as stream function were of course visually acceptable. One would expect vertical velocity to be poor, of course. If $w \sim uh_x \sim u\Delta z/\Delta x$, the topography used would give $w \sim 4.8$ cm s^{-1} - actually half that because of the way horizontal fluxes are computed. The Cox model tends to underestimate such values, but still produces large w values. These could eventually break the CFL criterion, but can also give very large vertical cell Peclet numbers $w\Delta z/A_{TV}$.

The free surface model gave even poorer results. This is not unexpected, since it possesses extra degrees of freedom. Without the M option, numerical overflow occurred some 2500 timesteps into the calculation. Although the barotropic fields were smooth and contourable, strong +/- behaviour was visible, even far from the topographic regions, in the baroclinic fields, before the overflow occurred. The worst case was the vertical velocity at (u, v) points, which is an accurate indicator of the degree of grid-splitting occurring, and is not printed in the Cox code. Adding the M option at least permitted the calculation to proceed, for at least 10000 steps, and strongly reduced the +/- structure at velocity points. However, a +/- behaviour now appeared at tracer points which had not previously existed. Although barotropic fields remained contourable, the results were clearly not acceptable.

We have devoted some little space to the discussion of an extreme, unacceptable case, because we feel that it is just as important for users of a program to recognise when it does not give good answers as when it does.

The original Cox manuscript does not provide guidance to timesteps and other limitations. The reader is referred to Killworth et al. (1984) for a compendium of restrictions for the B-grid. An additional restriction, not mentioned, is that depending on the term balance, too large a value of cell Peclet or Reynolds number (typically above 2) can yield a +/- behaviour, as mentioned above. That paper immediately gives a restriction on barotropic timestep as

$$\Delta t < \Delta x/2C$$

where Δx is the minimum of the horizontal grid spacings, and C is the largest phase speed of the surface gravity wave $(gH_{max})^{1/2}$.

When using the tidal option, the size of Δt_{uv} and Δt_{TS} must be much smaller than the normal case. As an example, when testing Cox model 2, a 6 hour timestep was possible for Δt_{TS}, while using a 2 hour value for Δt_{uv}. One would naturally require the uv and TS timesteps to be the same for tidal investigations. Experiments on Cox model 2 showed that 2 hours was unstable, for reasons discussed elsewhere. Rather than search parameter space to find an optimal value, a very short value of 12 minutes was chosen for Δt_{uv}, Δt_{TS}, which gave almost identical results to the Euler backward calculation. This necessity for a small timestep for tidal calculations is not likely to be a problem, since integrations longer than a few days are normally unnecessary.

J. Detailed listing of code updates, with comments

The updates and comments which follow use the colon option in UPDOC, so that the code provided will compile without further changes.

```
:
:       Additional input-output routines specific to Cray running out
:       of core
:
/*** CRAY ASYNCHRONOUS DIRECT ACCESS I/O **********************
C                                                                   632400010 -KC
C     TO SUPPRESS DISK I/O INITIALISE ISTAT TO ZERO                 632400020 -KC
C                                                                   632400030 -KC
      DATA ISTAT/9/                                                 632400040 -K
      NLB=NBLK/512+1                                                633500000 -K
      CALL WOPEN(LU,NLB,ISTAT)                                      633600000 -K
      CALL SEEK(LU,NFRST,NWDS)                                      634400000 -K
-634500000
      CALL GETWA(LU,A,NFRST,NWDS)                                   635400000 -K
-635500000
      CALL APUTWA(LU,A,NFRST,NWDS)                                  636400000 -K
-636500000
      CALL WCLOSE(LU)                                               637400000 -K
      IF(ISTAT.EQ.0)RETURN                                          637400010 -K
      IF(LU.EQ.15)CALL WSUMMARY(ISTAT)                              637500000 -K
      SUBROUTINE WSUMMARY(IST)                                      637900010 -K
      REWIND('$STATS')                                              637900020 -K
      CALL COPYD('$STATS','$OUT')                                   637900030 -K
      RETURN                                                        637900040 -K
      END                                                           637900050 -K
:
:       Changes to COMDECK to accommodate free surface adjustments
:
/                            !!! ... TROPIC MERGE UPDATES ( VERSION 2 ) ... !!!
     *,NSSIF=2*(NJTBFT+NJTBFU)*LSEGF*KM                             002000000 F
     *,NDICES=2*(2*JMT*LSEG+JMT+MSPL)+1                             002100000
     * ISE(JMT,LSEG),IEE(JMT,LSEG),ISU(JMT,LSEG),IEU(JMT,LSEG),     003500000
     * LSE(JMT),LSU(JMT),ISPL(MSPL),JSPL(MSPL),NSPL                  003600000
-003900000
      COMMON /SCALAR/ DTTS,DTUV,DTBT,C2DTTS,C2DTUV,AH,AM,FKPH,      004200000
     * FKPM,ACOR,WGHT,OMEGA,RADIUS,GRAV,RADIAN,PI,SWLDEG            004300000
     *,XFBT(IMT),YFBT(IMT)                                          004800000
      COMMON /FIELDS/                                               005600000
     * ETA(IMT,JMT),UBT(IMT,JMT),VBT(IMT,JMT),HR(IMT,JMT)           005610000
     * ZUENG(IMT),ZVENG(IMT)                                        008200000
     * XF(IMT,JMT),YF(IMT,JMT),ETAD(IMT,JMT)                        009100000
     * ,ETAGRD(IMT,JMT),UT(IMT,JMT),VT(IMT,JMT),                    009200000
     * BHD(JMT),CHD(JMT),DHD(JMT),GHD(JMT),HHD(JMT),FUV(JMT),EM(IMT,JMT)  009300000
     * ,SPLR(MSPL)                                                  009310000 D
:
:       ----------------------------------
:       I      Modifications to OCEAN     I
:       I                                 I
:       ----------------------------------
:
:       Just to keep the comments correct!
:
C                            ---> TROPIC                   ======== 101900000 C
:
:       Modifications to namelists and equivalences
:
      DIMENSION FKMP(IMT,JMT),FKMQ(IMT,JMT),FINS(3),EM(IMT,JMT)     104000000
      EQUIVALENCE (ETA,FKMP,FINS),(UBT,FKMQ)                        104400000
      REAL AMF,AHF,FKPMF,FKPHF,DTTSF,DTUVF,DTBTF,ACORF,WGHTF,       104600000
      NAMELIST /EDDY/ AMF,AHF,FKPMF,FKPHF /TSTEPS/ DTTSF,DTUVF,DTBTF 104800000
     * /PARMS/ ACORF                                                104900000
     * ,WGHTF                                                       104950000 D
-105200000
```

```
:
:       Altered namelist processing
:
      DTBT=DTBTF                                                        116400000
      WGHT=WGHTF                                                        116600000 D
-116700000
:
:       Housekeeping for external mode calculations,
:       as commented
:
-132700000
C-----------------------------------------------------------------------135700000 C
C  CALCULATE DO LOOP INDICES FOR ETA AND EXTERNAL MODE.                 135710000 C
C-----------------------------------------------------------------------135720000 C
C                                                                       135730000 C
      DO 748 N=1,NDICES                                                 135740000
        ISE(N,1)=0                                                      135750000
 748  CONTINUE                                                          135760000
C                                                                       135770000 C
      DO 750 J=2,JMTM1                                                  135780000
        L=0                                                             135790000
        IF ( FKMP(1,J).GT.0 .AND. FKMP(2,J).GT.0 ) THEN                 135800000
          L=L+1                                                         135810000
          ISE(J,L)=2                                                    135820000
        ENDIF                                                           135830000
        DO 752 I=2,IMTM1                                                135840000
          IF ( FKMP(I-1,J).EQ.0 .AND. FKMP(I,J).GT.0 ) THEN             135850000
            L=L+1                                                       135860000
            ISE(J,L)=I                                                  135870000
          ENDIF                                                         135880000
          IF ( FKMP(I,J).GT.0 .AND. FKMP(I+1,J).EQ.0 ) IEE(J,L)=I       135890000
 752    CONTINUE                                                        135900000
        IF ( FKMP(IMTM1,J).GT.0 .AND. FKMP(IMT,J).GT.0 ) IEE(J,L)=IMTM1 135910000
        LSE(J)=L                                                        135920000
 750  CONTINUE                                                          135930000
C                                                                       135940000 C
      MLSU=0                                                            135950000
      DO 760 J=2,JMTM2                                                  135960000 -S
      DO 760 J=2,JMTM1                                                  135965000 S
        L=0                                                             135970000
        IF ( FKMQ(1,J).GT.0 .AND. FKMQ(2,J).GT.0 ) THEN                 135980000
          L=L+1                                                         135990000
          ISU(J,L)=2                                                    136000000
        ENDIF                                                           136010000
        DO 762 I=2,IMUM1                                                136020000
          IF ( FKMQ(I-1,J).EQ.0 .AND. FKMQ(I,J).GT.0 ) THEN             136030000
            L=L+1                                                       136040000
            ISU(J,L)=I                                                  136050000
          ENDIF                                                         136060000
          IF ( FKMQ(I,J).GT.0 .AND. FKMQ(I+1,J).EQ.0 ) IEU(J,L)=I       136070000
 762    CONTINUE                                                        136080000
        IF ( FKMQ(IMUM1,J).GT.0 .AND. FKMQ(IMU,J).GT.0 ) IEU(J,L)=IMUM1 136090000
        LSU(J)=L                                                        136100000
        MLSU=MAX(LSU(J),MLSU)                                           136110000
 760  CONTINUE                                                          136120000
C                                                                       136130000 C
        IF ( MLSU.GT.LSEG ) THEN                                        136140000
          WRITE(6,9920) ' PARAMETER LSEG TOO SMALL!',                   136150000
     >                  ' MUST BE AT LEAST ',MLSU                       136160000
          STOP                                                          136170000
        ENDIF                                                           136180000
C                                                                       136190000 C
C-----------------------------------------------------------------------136200000 C
C  CALCULATE MASK FIELD AT ETA POINTS.                                  136210000 C
```

```
C-----------------------------------------------------------------      136220000 C
C                                                                       136230000 C
      DO 770 J=1,JMT                                                    136240000
      DO 770 I=1,IMT                                                    136250000
        IF ( FKMP(I,J).GT.0 ) THEN                                      136260000
          EM(I,J)=1.                                                    136270000
        ELSE                                                            136280000
          EM(I,J)=0.                                                    136290000
        ENDIF                                                           136300000
  770 CONTINUE                                                          136310000
C                                                                       136320000 CD
C-----------------------------------------------------------------      136330000 CD
C  CALCULATE INDEXING ARRAY FOR 'INTERIOR' ETA POINTS ON COAST.  THIS   136340000 CD
C  IS USED FOR THE CALCULATION OF THE DEL-SQUARED TERM ON ETA.          136350000 CD
C-----------------------------------------------------------------      136360000 CD
C                                                                       136370000 CD
      NSPL=0                                                            136380000 D
      DO 780 J=2,JMTM1                                                  136390000 D
      DO 780 I=2,IMTM1                                                  136400000 D
        IF ( EM(I,J).EQ.0 ) THEN                                        136410000 D
          SPL=EM(I,J+1)+EM(I+1,J)+EM(I-1,J)+EM(I,J-1)                   136420000 D
          IF ( SPL.NE.0 ) THEN                                          136430000 D
            NSPL=NSPL+1                                                 136440000 D
            ISPL(NSPL)=I                                                136450000 D
            JSPL(NSPL)=J                                                136460000 D
          ENDIF                                                         136470000 D
        ENDIF                                                           136480000 D
  780 CONTINUE                                                          136490000 D
C                                                                       136500000 CD
      IF ( NSPL.GT.MSPL ) THEN                                          136510000 D
        WRITE(6,9920) ' PARAMETER MSPL TOO SMALL!',                     136520000 D
     >                ' MUST BE AT LEAST ',NSPL                         136530000 D
        STOP                                                            136540000 D
      ENDIF                                                             136550000 D
C                                                                       136560000 C
C-----------------------------------------------------------------      136570000 C
C  PRINT OUT RUN DETAILS                                                136580000 C
C-----------------------------------------------------------------      136590000 C
C                                                                       136600000 C
      WRITE (6,9960)                                                    136610000
9960    FORMAT (//1X,'DO LOOP INDICES FOR SURFACE DISPLACEMENT  (ETA)'/) 136620000
      DO 9962 J=JMT,1,-1                                                136630000
        WRITE (6,9970) (J,(ISE(J,L),IEE(J,L),L=1,LSE(J)))               136640000
9962  CONTINUE                                                          136650000
C                                                                       136660000 C
      WRITE (6,9964)                                                    136670000
9964    FORMAT (//1X,'DO LOOP INDICES FOR TRANSPORTS  (UBT,VBT)'/)      136680000
      DO 9966 J=JMT,1,-1                                                136690000
        WRITE (6,9970) (J,(ISU(J,L),IEU(J,L),L=1,LSU(J)))               136700000
9966  CONTINUE                                                          136710000
C                                                                       136720000 C
      WRITE (6,'(/)')                                                   136730000
C                                                                       136740000 C
9920  FORMAT (5X,A/5X,A,I5)                                             136750000
9970  FORMAT (5X,I5,' ... ',20I5)                                       136760000
-136800000,138700000
:
:       Remove u, v filtering indices
:
-139900000,140000000
-140600000
:
:       Remove stream function indices printout, and island
:       definitions
```

```
:
-149300000,149900000
-150300000
-151200000,151300000
:
:       Initialisation and output of initial data
:
C  SEND RECIPROCAL DEPTH AND ETA MASK TO DISC                           157010000 C
C                                                                       157020000 C
      CALL OPUT(KFLDS,NWDS,6*NWDS+1,EM)                                 157030000
      CALL OPUT(KFLDS,NWDS,       1,HR)                                 157040000
C                                                                       157050000 C
C  SET INITIAL SURFACE DISPLACEMENT AND EXTERNAL TRANSPORTS TO ZERO.    157100000 C
        ETA(I,J)=0.0                                                    157500000
        UBT(I,J)=0.0                                                    157600000
        VBT(I,J)=0.0                                                    157610000
      CALL OPUT(KFLDS,NWDS,  NWDS+1,UBT)                                158100000
      CALL OPUT(KFLDS,NWDS,2*NWDS+1,VBT)                                158200000
      CALL OPUT(KFLDS,NWDS,3*NWDS+1,ETA)                                158300000
      CALL OPUT(KFLDS,NWDS,4*NWDS+1,UBT)                                158400000
      CALL OPUT(KFLDS,NWDS,5*NWDS+1,VBT)                                158500000
C       NOTE THAT "FINS" IS EQUIVALENCED WITH "ETA")                    158900000 C
        FINS(N)=ISE(N,1)                                                159200000
      CALL OPUT(KFLDS,NDICES,7*NWDS+1,FINS)                             160300000
:
:       Read in disc data to start run
:
      CALL OGET(KFLDS,NDICES,7*NWDS+1,FINS)                             162300000
        ISE(N,1)=FINS(N)                                                162500000
      CALL OGET(KFLDS,NWDS,3*NWDS+1,ETA)                                163300000
      CALL OGET(KFLDS,NWDS,4*NWDS+1,UBT)                                163400000
      CALL OGET(KFLDS,NWDS,5*NWDS+1,VBT)                                163500000
      CALL OGET(KFLDS,NWDS,       1, HR)                                163510000
:
:       Do not initialise ZTD
:
-164600000,164800000
:
:       At end of run, save data to unit 23, not 22,
:       for restart
:
      IF(ITT.EQ.NLAST .AND. NA.EQ.1) CALL OWRT(23)                      167200000 K
:
:       ----------------------------------
:       I      Modifications to STEP      I
:       I                                 I
:       ----------------------------------
:
:       Changes to mixing timestep adjustments
:
:
-207100000
-207600000,207800000
        CALL OPUT(KFLDS,NWDS,  NWDS+1,UBT)                              207900000
        CALL OPUT(KFLDS,NWDS,2*NWDS+1,VBT)                              208000000
:
:       Queue UBT, VBT for read in of slabs
:
        CALL OFIND(KFLDS,IMT,  NWDS+(J-1)*IMT+1)                        211610000
        CALL OFIND(KFLDS,IMT,2*NWDS+(J-1)*IMT+1)                        211620000
:
:       Initialise the (now one-dimensional) ZUENG, ZVENG
:
        DO 186 LL=1,8                                                   213200000
```

```
 186    CONTINUE                                                        213450000
          ZUENG(I)=0.0                                                  213600000
          ZVENG(I)=0.0                                                  213700000
:
:       Read UBT, VBT for slab 2
:
      CALL OGET(KFLDS,IMT,  NWDS+IMT+1,SFUB)                            216710000
      CALL OGET(KFLDS,IMT,2*NWDS+IMT+1,SFVB)                            216720000
:
:       Remove vorticity computation, and compute external
:       mode velocities for row 2
:
-224300000,225200000
-227500000,227600000
        SFUB(I)=SFUB(I)*HR(I,J+1)                                       227700000
        SFVB(I)=SFVB(I)*HR(I,J+1)                                       227800000
-228400000,228500000
        SFU(I)=UBT(I,J+1)*HR(I,J+1)                                     228600000
        SFV(I)=VBT(I,J+1)*HR(I,J+1)                                     228700000
:
:       Read J+1 slab barotropic velocities
:
        CALL OGET(KFLDS,IMT,  NWDS+J*IMT+1,SFUB)                        236810000
        CALL OGET(KFLDS,IMT,2*NWDS+J*IMT+1,SFVB)                        236820000
:
:       As indicated by comments, plus small housekeeping
:       modifications
:
C                                                                       264510000 C
C-----------------------------------------------------------------------264520000 C
C  INITIATE WRITEOUT OF NEWLY COMPUTED DATA FROM THE FINAL ROW          264530000 C
C-----------------------------------------------------------------------264540000 C
C                                                                       264550000 C
      CALL OPUT(LABS(NDISKA),NSLAB,(JMT-2)*NSLAB+1,TA)                  264560000
C                                                                       264570000 C
C-----------------------------------------------------------------------264580000 C
C  SOLVE FOR NEW EXTERNAL MODE                                          264590000 C
C-----------------------------------------------------------------------264600000 C
C                                                                       264610000 C
      IF ( MXP.EQ.0 ) CALL TROPIC                                       264620000
C                                                                       264630000 C
      PRINT 910,ITT,TTYEAR,TTDAY,EKTOT,DTABS(1),DTABS(2)                266100000
     *    1PE13.6,8H  DTEMP=,1PE13.6,8H  DSALT=,1PE13.6)                266300000
-275900000,277500000
:
:       If not second pass of Euler backward, save data in case
:       of abnormal stop
:
      IF ( MXP.EQ.0 ) THEN                                              281500000
        CALL OPUT(KFLDS,NWDS,3*NWDS+1,ETA)                              281510000
        CALL OPUT(KFLDS,NWDS,4*NWDS+1,UBT)                              281520000
        CALL OPUT(KFLDS,NWDS,5*NWDS+1,VBT)                              281530000
      ENDIF                                                             281540000
:
:       Remove printout of streamfunction
:
-281700000,282700000
:
:       ----------------------------------
:       I      Modifications to CLINIC    I
:       I                                 I
:       ----------------------------------
:
:       Set up new local variables for CLINIC and remove
```

```
:       calculation of advective coefficients
:
     *           ,FUWBT(IMT),FVNBT(IMT),FVSBT(IMT)                                302810000 M
     *           ,QU(IMT),QV(IMT)                                                 302820000
-304100000,307200000
:
:       New calculations of external mode velocities
:
-309700000,309800000
        SFUB(I)=SFUB(I)*HR(I,J+1)                                                 309900000
        SFVB(I)=SFVB(I)*HR(I,J+1)                                                 310000000
-310600000,310700000
        SFU(I)=UBT(I,J+1)*HR(I,J+1)                                               310800000
        SFV(I)=VBT(I,J+1)*HR(I,J+1)                                               310900000
:
:       Calculation of advective coefficients, with or
:       without 4-point average
:
C                                                                                 316510000 C
C------------------------------------------------------------------------         316520000 C
C  FIND ADVECTIVE COEFFICIENTS 'FUW' FOR WEST FACE  OF U,V BOX                    316530000 C
C                            & 'FVN' FOR NORTH FACE OF U,V BOX                    316540000 C
C------------------------------------------------------------------------         316550000 C
C                                                                                 316560000 C
      FX=DYU2R(J)*CSR(J)*CST(J+1)                                                 316570000
      DO 115 K=1,KM                                                               316580000
      DO 115 I=1,IMT                                                              316590000
        FUW(I,K)=(U(I,K)+U(I-1,K))*CSR(J)*0.5                                     316600000 -M
        FUW(I,K)=( ( ( U(I,K)*DXU(I)+U(I-1,K)*DXU(I-1) )*DYU(J)                   316610000 M
     *     +( UP(I,K)*DXU(I)+UP(I-1,K)*DXU(I-1) )*DYU(J+1) )*DYT2R(J+1)           316620000 M
     *   +( ( U (I,K)*DXU(I)+U (I-1,K)*DXU(I-1) )*DYU(J)                          316630000 M
     *     +( UM(I,K)*DXU(I)+UM(I-1,K)*DXU(I-1) )*DYU(J-1) )*DYT2R(J) )           316640000 M
     *     * DXT4R(I) * CSR(J)                                                    316650000 M
        FVN(I,K)=(VP(I,K)+V(I,K))*FX*0.5                                          316660000 -M
        FVN(I,K)= DYT4R(J+1) * FX *                                               316670000 M
     *  ( ( ( V (I,K)*DXU(I)+V (I-1,K)*DXU(I-1) )*DYU(J)                          316680000 M
     *     +( VP(I,K)*DXU(I)+VP(I-1,K)*DXU(I-1) )*DYU(J+1) )*DXT2R(I)             316690000 M
     *   +( ( V (I,K)*DXU(I)+V (I+1,K)*DXU(I+1) )*DYU(J)                          316700000 M
     *     +( VP(I,K)*DXU(I)+VP(I+1,K)*DXU(I+1) )*DYU(J+1) )*DXT2R(I+1))          316710000 M
 115  CONTINUE                                                                    316720000
C                                                                                 316730000 C
:
:       Calculate w at surface, with or without four-point average
:
C------------------------------------------------------------------------         316740000 SC
C                                                                                 320310000 MC
C CALCULATE FUWBT,FVNBT AND FVSBT READY FOR COMPUTATION OF W (SURFACE)            320320000 MC
C                                                                                 320330000 MC
      FX=DYUR(J)*CSR(J)*CST(J+1)                                                  320340000 M
      FXA=DYUR(J)*CSR(J)*CST(J)                                                   320350000 M
      DO 220 I=1,IMT                                                              320360000 M
       FUWBT(I)=( ( ( UBT(I,J)*DXU(I)+UBT(I-1,J)*DXU(I-1) )*DYU(J)                320370000 M
     * +(UBT(I,J+1)*DXU(I)+UBT(I-1,J+1)*DXU(I-1))*DYU(J+1) )*DYT2R(J+1)           320380000 M
     * +( ( UBT(I,J)*DXU(I)+UBT(I-1,J)*DXU(I-1) )*DYU(J)                          320390000 M
     * +(UBT(I,J-1)*DXU(I)+UBT(I-1,J-1)*DXU(I-1) )*DYU(J-1) )*DYT2R(J) )          320400000 M
     *     * DXT4R(I) * CSR(J)                                                    320410000 M
       FVNBT(I)= DYT4R(J+1) *  FX *                                               320420000 M
     *  ( ( ( VBT(I,J)*DXU(I)+VBT(I-1,J)*DXU(I-1) )*DYU(J)                        320430000 M
     * +(VBT(I,J+1)*DXU(I)+VBT(I-1,J+1)*DXU(I-1))*DYU(J+1) )*DXT2R(I)             320440000 M
     *   +( ( VBT(I,J)*DXU(I)+VBT(I+1,J)*DXU(I+1) )*DYU(J)                        320450000 M
     * +(VBT(I,J+1)*DXU(I)+VBT(I+1,J+1)*DXU(I+1))*DYU(J+1) )*DXT2R(I+1))          320460000 M
       FVSBT(I)= DYT4R(J-1) *  FXA *                                              320470000 M
     *  ( ( ( VBT(I,J)*DXU(I)+VBT(I-1,J)*DXU(I-1) )*DYU(J)                        320480000 M
     * +(VBT(I,J-1)*DXU(I)+VBT(I-1,J-1)*DXU(I-1))*DYU(J-1) )*DXT2R(I)             320490000 M
```

```
     *   +( ( VBT(I,J)*DXU(I)+VBT(I+1,J)*DXU(I+1) )*DYU(J)                      320500000 M
     * +(VBT(I,J-1)*DXU(I)+VBT(I+1,J-1)*DXU(I+1))*DYU(J-1) )*DXT2R(I+1))        320510000 M
 220  CONTINUE                                                                  320520000 M
C                                                                               320530000 C
C  CALCULATE VERTICAL VELOCITY AT THE SURFACE                                   320540000 C
-320900000
        W(I,1) = - ( DXUR(I)*( FUWBT(I+1)-FUWBT(I) )                            320910000 M
     >                   + ( FVNBT(I)-FVSBT(I) ) )                              320920000 M
        W(I,1) = -CSR(J)                                                        320930000 -M
     >            *( DXU2R(I)*(UBT(I+1,J)-UBT(I-1,J))                           320940000 -M
     >             + DYU2R(J)*( (VBT(I,J+1)+VBT(I,J))*CST(J+1)                  320950000 -M
     >                        - (VBT(I,J)+VBT(I,J-1))*CST( J ) ) )              320960000 -M
:
:       Additional comment
:
C  COMPUTE HORIZONTAL ADVECTION OF MOMENTUM                                      330700000 C
:
:       Calculate excess flux of horizontal momentum, and
:       remove vertical flux divergence calculation from here
:
C-----------------------------------------------------------------------        333600000 C
C COMPUTE QU,QV FOR CONSERVATION OF MOMENTUM OVER TOPOGRAPHY IN                  333700000 C
C VERTICAL FLUX DIVERGENCE LATER.                                               333800000 C
C-----------------------------------------------------------------------        333900000 C
C                                                                               334000000 C
      DO 307 I=1,IMT                                                            334100000
        QU(I)=0.0                                                               334200000
        QV(I)=0.0                                                               334300000
        DO 307 K=KMU(I)+1,KM                                                    334400000
          QU(I)=QU(I)+UA(I,K)*DZ(K)                                             334500000
          QV(I)=QV(I)+VA(I,K)*DZ(K)                                             334600000
 307  CONTINUE                                                                  334700000
-334800000,335000000
:
:       Re-ordered calculation of various terms, explained
:       in comments
C-----------------------------------------------------------------------        340400000 C
C  ADD IN PRESSURE TERM AND MASK OUT LAND                                       340500000 C
C-----------------------------------------------------------------------        340600000 C
C                                                                               340700000 C
      DO 350 K=1,KM                                                             340800000
      DO 350 I=1,IMT                                                            340900000
        UA(I,K)=GM(I,K)*(UA(I,K)-DPDX(I,K))                                     341000000
        VA(I,K)=GM(I,K)*(VA(I,K)-DPDY(I,K))                                     341100000
 350  CONTINUE                                                                  341200000
C                                                                               341300000 C
C-----------------------------------------------------------------------        341400000 C
C  CALCULATE VERTICALLY INTEGRATED FORCING                                      341500000 C
C-----------------------------------------------------------------------        341600000 C
C                                                                               341700000 C
      FX=0.0                                                                    341800000
      DO 355 I=1,IMT                                                            341900000
        XFBT(I)=FX                                                              342000000
        YFBT(I)=FX                                                              342100000
 355  CONTINUE                                                                  342200000
      DO 360 K=1,KM                                                             342300000
      DO 360 I=1,IMT                                                            342400000
        XFBT(I)=XFBT(I)+UA(I,K)*DZ(K)                                           342500000
        YFBT(I)=YFBT(I)+VA(I,K)*DZ(K)                                           342600000
 360  CONTINUE                                                                  342700000
C                                                                               342800000 C
C-----------------------------------------------------------------------        342900000 C
C  ADD IN VERTICAL ADVECTION OF MOMENTUM AND THEN                               342950000 C
C  ADD IN CORIOLIS FORCE (EVAL. ON TAU   TSTEP FOR EXPLICIT TRTMNT;             343000000 C
```

```
C                          EVAL. ON TAU-1 TSTEP FOR IMPLICIT TRTMNT              343100000 C
C                          WITH REMAINDER OF TERM TO BE ADDED LATER)            343200000 C
C-----------------------------------------------------------------------        343300000 C
C                                                                               343400000 C
      DO 340 K=1,KMP1                                                           343410000
      DO 340 I=1,IMT                                                            343420000
        TEMPA(I,K)=W(I,K)*(U(I,K-1)+U(I,K))                                     343430000
        TEMPB(I,K)=W(I,K)*(V(I,K-1)+V(I,K))                                     343440000
 340    CONTINUE                                                                343450000
      DO 341 I=1,IMT                                                            343460000
        TEMPA(I,1)=W(I,1)*2.0*U(I,1)                                            343470000
        TEMPB(I,1)=W(I,1)*2.0*V(I,1)                                            343480000
 341  CONTINUE                                                                  343490000
C                                                                               343500000 C
C  PATCH TO CONSERVE MOMENTUM OVER TOPOGRAPHY                                   343510000 C
C                                                                               343520000 C
      DO 3400 I=1,IMT                                                           343530000
C                                                                               343540000 C
C  FX IS USED IN PLACE OF AN 'IF' STATEMENT TO INCREASE VECTORISATION           343550000 C
C                                                                               343560000 C
        FX=MIN(1,MAX(0,KM-KMU(I)))                                              343570000
        FXA=1.-FX                                                               343580000
        TEMPA(I,KMU(I)+1)=TEMPA(I,KMU(I)+1)*FXA + 2.*QU(I)*FX                   343590000
        TEMPB(I,KMU(I)+1)=TEMPB(I,KMU(I)+1)*FXA + 2.*QV(I)*FX                   343600000
        XFBT(I)=XFBT(I)+QU(I)*FX                                                343610000
        YFBT(I)=YFBT(I)+QV(I)*FX                                                343620000
 3400 CONTINUE                                                                  343630000
C                                                                               343640000 C
      DO 343 K=1,KM                                                             343650000
      DO 343 I=1,IMT                                                            343660000
        UA(I,K)=UA(I,K)+(TEMPA(I,K+1)-TEMPA(I,K))*DZ2RQ(I,K)                    343670000
        VA(I,K)=VA(I,K)+(TEMPB(I,K+1)-TEMPB(I,K))*DZ2RQ(I,K)                    343680000
343   CONTINUE                                                                  343690000
C                                                                               343700000 C
C                                                                               343710000 C
      FX=2.0*OMEGA*SINE(J)                                                      343720000
      IF(ACOR.EQ.0.) THEN                                                       343730000
        DO 370 K=1,KM                                                           343740000
        DO 370 I=1,IMT                                                          343800000
          UA(I,K)=UA(I,K)+FX*V(I,K)                                             343900000
          VA(I,K)=VA(I,K)-FX*U(I,K)                                             344000000
 370    CONTINUE                                                                344100000
      ELSE                                                                      344200000
        DO 375 K=1,KM                                                           344300000
        DO 375 I=1,IMT                                                          344400000
          UA(I,K)=UA(I,K)+FX*VB(I,K)                                            344500000
          VA(I,K)=VA(I,K)-FX*UB(I,K)                                            344600000
 375    CONTINUE                                                                344700000
      ENDIF                                                                     344800000
C                                                                               344900000 C
C-----------------------------------------------------------------------        345000000 C
C  MASK OUT LAND AGAIN                                                          345100000 C
C-----------------------------------------------------------------------        345200000 C
C                                                                               345300000 C
      DO 380 K=1,KM                                                             345400000
      DO 380 I=1,IMT                                                            345500000
-345510000,345540000
        UA(I,K)=GM(I,K)*UA(I,K)                                                 345600000
        VA(I,K)=GM(I,K)*VA(I,K)                                                 345700000
 380  CONTINUE                                                                  345800000
-345900000,346200000
:
:       Incorporate external mode analysis with internal mode
:
```

```
      FXC=0.0                                                          346850000
-347000000
        ZUENG(I)=FX                                                    347200000
        ZVENG(I)=FX                                                    347300000
        ZUENG(I)=ZUENG(I)+UENG(I,K)*DZ(K)*HR(I,J)                      349000000
        ZVENG(I)=ZVENG(I)+VENG(I,K)*DZ(K)*HR(I,J)                      349100000
      DO 410 I=2,IMUM1                                                 349210000
        ZUENG(I)=ZUENG(I) - GRAV*CSR(J)*DXU2R(I)                       349220000
     >                    *( ETA(I+1,J+1) + ETA(I+1,J)                 349230000
     >                     - ETA(I  ,J+1) - ETA(I  ,J) )               349240000
        ZVENG(I)=ZVENG(I) - GRAV*DYU2R(J)                              349250000
     >                    *( ETA(I,J+1) + ETA(I+1,J+1)                 349260000
     >                     - ETA(I,J  ) - ETA(I+1,J  ) )               349270000
        ENGEXT(6)=ENGEXT(6)+(UBT(I,J)*ZUENG(I)                         349280000
     *                      +VBT(I,J)*ZVENG(I))*FX*DXU(I)              349290000
        ZUENG(I)=FXC                                                   349300000
        ZVENG(I)=FXC                                                   349310000
 410  CONTINUE                                                         349320000
C                                                                      349330000 C
        ZUENG(I)=ZUENG(I)+UENG(I,K)*DZ(K)*HR(I,J)                      350800000
        ZVENG(I)=ZVENG(I)+VENG(I,K)*DZ(K)*HR(I,J)                      350900000
      DO 440 I=2,IMUM1                                                 351010000
        ENGEXT(2)=ENGEXT(2)+(UBT(I,J)*ZUENG(I)                         351020000
     *                      +VBT(I,J)*ZVENG(I))*FX*DXU(I)              351030000
        ZUENG(I)=FXC                                                   351040000
        ZVENG(I)=FXC                                                   351050000
 440  CONTINUE                                                         351060000
       IF (K.EQ.1) THEN                                                351210000
        UENG(I,K)=GM(I,K)*(-(W(I,K  )*U(I,K)*2.0                       351220000
     *                 -W(I,K+1)*(U(I,K)+U(I,K+1)))*DZ2RQ(I,K))        351230000
        VENG(I,K)=GM(I,K)*(-(W(I,K  )*V(I,K)*2.0                       351240000
     *                 -W(I,K+1)*(V(I,K)+V(I,K+1)))*DZ2RQ(I,K))        351250000
       ELSE                                                            351260000
       ENDIF                                                           351610000
        ZUENG(I)=ZUENG(I)+UENG(I,K)*DZ(K)*HR(I,J)                      351900000
        ZVENG(I)=ZVENG(I)+VENG(I,K)*DZ(K)*HR(I,J)                      352000000
      DO 470 I=2,IMUM1                                                 352110000
        ENGEXT(3)=ENGEXT(3)+(UBT(I,J)*ZUENG(I)                         352120000
     *                      +VBT(I,J)*ZVENG(I))*FX*DXU(I)              352130000
        ZUENG(I)=FXC                                                   352140000
        ZVENG(I)=FXC                                                   352150000
 470  CONTINUE                                                         352160000
        ZUENG(I)=ZUENG(I)+UENG(I,K)*DZ(K)*HR(I,J)                      353900000
        ZVENG(I)=ZVENG(I)+VENG(I,K)*DZ(K)*HR(I,J)                      354000000
      DO 495 I=2,IMUM1                                                 354110000
        ENGEXT(4)=ENGEXT(4)+(UBT(I,J)*ZUENG(I)                         354120000
     *                      +VBT(I,J)*ZVENG(I))*FX*DXU(I)              354130000
        ZUENG(I)=FXC                                                   354140000
        ZVENG(I)=FXC                                                   354150000
 495  CONTINUE                                                         354160000
        ZUENG(I)=ZUENG(I)+UENG(I,1)*DZ(1)*HR(I,J)                      355000000
        ZVENG(I)=ZVENG(I)+VENG(I,1)*DZ(1)*HR(I,J)                      355100000
      DO 523 I=2,IMUM1                                                 355210000
        ENGEXT(7)=ENGEXT(7)+(UBT(I,J)*ZUENG(I)                         355220000
     *                      +VBT(I,J)*ZVENG(I))*FX*DXU(I)              355230000
        ZUENG(I)=FXC                                                   355240000
        ZVENG(I)=FXC                                                   355250000
 523  CONTINUE                                                         355260000
        ZUENG(I)=ZUENG(I)+UENG(I,KZ)*DZ(KZ)*HR(I,J)                    356200000
        ZVENG(I)=ZVENG(I)+VENG(I,KZ)*DZ(KZ)*HR(I,J)                    356300000
      DO 525 I=2,IMUM1                                                 356410000
        ENGEXT(8)=ENGEXT(8)+(UBT(I,J)*ZUENG(I)                         356420000
     *                      +VBT(I,J)*ZVENG(I))*FX*DXU(I)              356430000
        ZUENG(I)=FXC                                                   356440000
```

```
      ZVENG(I)=FXC                                                      356450000
 525  CONTINUE                                                          356460000
      ZUENG(I)=ZUENG(I)+UENG(I,K)*DZ(K)*HR(I,J)                         358100000
      ZVENG(I)=ZVENG(I)+VENG(I,K)*DZ(K)*HR(I,J)                         358200000
     DO 530 I=2,IMUM1                                                   358400000
      ENGEXT(5)=ENGEXT(5)+(UBT(I,J)*ZUENG(I)                            358420000
    *                   +VBT(I,J)*ZVENG(I))*FX*DXU(I)                   358430000
      ZUENG(I)=FXC                                                      358440000
      ZVENG(I)=FXC                                                      358450000
 530  CONTINUE                                                          358460000
 550  CONTINUE                                                          358470000
:
:       Set cyclic boundary conditions on external mode
:       forcing
:
-366100000,366400000
     XFBT(1)=XFBT(IMUM1)                                                366900000 O
     YFBT(1)=YFBT(IMUM1)                                                367000000 O
-367100000,367600000
:
:       As commented. plus removal of vorticity computation
:
C-----------------------------------------------------------------------367800000 C
C  WRITE OUT VALUES OF EXT. MODE FORCING FOR USE IN TROPIC              367900000 C
C-----------------------------------------------------------------------368000000 C
C                                                                       368100000 C
     IF ( MXP.EQ.0 ) THEN                                               368200000
      CALL OPUT(KFLDS,IMT,  NWDS+(J-1)*IMT+1,XFBT)                      368300000
      CALL OPUT(KFLDS,IMT,2*NWDS+(J-1)*IMT+1,YFBT)                      368400000
     ENDIF                                                              368500000
C                                                                       368600000 C
-368700000,384300000
:
:       Now no need for this transfer
:
-385400000,386400000
:
:       -----------------------------------
:       I      Modifications to TRACER    I
:       I                                 I
:       -----------------------------------
:
:       Calculate w at surface T points
:
C  1ST, COMPUTE VERTICAL VELOCITY AT THE SURFACE                        405500000 C
      W(I,1) = -CSTR(J)*DXT2R(I)*DYTR(J)                                405900000
    >          *(  UBT(I  ,J)*DYU(J)+UBT(I  ,J-1)*DYU(J-1)              405910000
    >            - (UBT(I-1,J)*DYU(J)+UBT(I-1,J-1)*DYU(J-1))            405920000
    >            + (VBT(I,J  )*DXU(I)+VBT(I-1,J  )*DXU(I-1))*CS( J )    405930000
    >            - (VBT(I,J-1)*DXU(I)+VBT(I-1,J-1)*DXU(I-1))*CS(J-1) )  405940000
:
:       Compute flux of tracer through the top ocean box
:
     DO 823 I=1,IMT                                                     414510000
      TEMPB(I,1)=W(I,1)*2.0*T(I,1,M)                                    414520000
 823  CONTINUE                                                          414530000
:
:       Correct analysis of tracer forcing to allow for
:       free surface
:
      TUP=T(I,K-1,M)                                                    424610000
      IF (K.EQ.1) TUP=T(I,K,M)                                          424620000
    *              -W(I,K  )*(TUP   +T(I,K  ,M)))*DZ2R(K)               425800000
:
```

```
:       Correct buoyancy forcing for free surface
:
        BUOY=BUOY-FX*DZZ(1)*W(I,1)*2.0*RHOS(I,1)                          428110000
        BUOY=BUOY-W(I,1)*ETA(I,J)*FX*2.0                                  428120000
:       -----------------------------------
:       I      Modifications to RELAX     I
:       I                                 I
:       -----------------------------------
:
:       Remove routine entirely
:
-500000000,563700000
:
:       -----------------------------------
:       I      Modifications to ODAM      I
:       I                                 I
:       -----------------------------------
:
:       Rewind restart unit if saving data before the
:       end of the run
:
      REWIND 22                                                           647850000 K
:
:       -----------------------------------
:       I      New routine TROPIC         I
:       I                                 I
:       -----------------------------------
:
:       The routine is fully commented below
:
*DECK TROPIC                                                              800000000
      SUBROUTINE TROPIC                                                   800100000
C                                                                         800200000 C
C======================================================================== 800300000 C
C                                                                  ====== 800400000 C
C         S. M. PATERSON (19/1/1988), D. STAINFORTH (6/4/1989)     ====== 800500000 C
C                                                                  ====== 800600000 C
C  TROPIC TAKES AS INPUT THE EXTERNAL MODE FORCING CALCULATED IN   ====== 800700000 C
C         "CLINIC" (XF,YF) AND BY TIME STEPPING THE BAROTROPIC EQUATIONS ====== 800800000 C
C         CALCULATES THE SURFACE DISPLACEMENT (ETA) AND THE BAROTROPIC ====== 800900000 C
C         TRANSPORTS (UBT,VBT).  THE INITIAL VALUES OF THE TRANSPORTS ====== 801000000 C
C         ARE WRITTEN TO DISK FOR USE IN THE NEXT BARCCLINIC TIME STEP. ====== 801100000 C
C                                                                  ====== 801200000 C
C         THIS VERSION DOES NOT INCLUDE ANY CODE FOR THE COX UPDATE ====== 801300000 C
C         OPTIONS 'F','H' OR 'T'.                                  ====== 801400000 C
C                                                                  ====== 801500000 C
C======================================================================== 801600000 C
C                                                                         801700000 C
C-----------------------------------------------------------------        801800000 C
C  DEFINE GLOBAL DATA                                                     801900000 C
C-----------------------------------------------------------------        802000000 C
C                                                                         802100000 C
*CALL PARAM                                                               802200000
*CALL FULLWD                                                              802300000
*CALL SCALAR                                                              802400000
*CALL ONEDIM                                                              802500000
*CALL FIELDS                                                              802600000
*CALL WRKSPB                                                              802700000
C                                                                         802800000 C
C-----------------------------------------------------------------        802900000 C
C  LOCAL DATA VARIABLE ETAD IS SET UP IN THE WRKSPB                       803000000 C
C-----------------------------------------------------------------        803100000 C
C                                                                         803200000 C
C-------------------------------------------------------------------      803300000 C
```

```
C  QUEUE UP DISK READS FOR TROPIC.
C-----------------------------------------------------------------------
C
      CALL OFIND(KFLDS,NWDS,6*NWDS+1)
      CALL OFIND(KFLDS,NWDS,  NWDS+1)
      CALL OFIND(KFLDS,NWDS,2*NWDS+1)
C
C-----------------------------------------------------------------------
C  SET ETAGRD TO ZERO TO AVOID UNSET VARIABLE ERROR
C-----------------------------------------------------------------------
C
      DO 20 J=1,JMT
      DO 20 I=1,IMT
        ETAGRD(I,J) = 0.
 20   CONTINUE
C
C-----------------------------------------------------------------------
C  CALCULATE ARRAYS USED IN THE HORIZONTAL DIFFUSION TERM AND ALSO THE
C  CORIOLIS PARAMETER.
C-----------------------------------------------------------------------
C
      DO 30 J=2,JMTM2
      DO 30 J=2,JMTM1
        BHD(J) = CSR(J)**2
        CHD(J) = DYUR(J)*CSR(J)*CST(J+1)
        DHD(J) = DYUR(J)*CSR(J)*CST(J)
        GHD(J) = (1.-TNG(J)**2)/(RADIUS**2)
        HHD(J) = SINE(J)/(RADIUS*CS(J)**2)
        FUV(J) = 2.*OMEGA*SINE(J)
 30   CONTINUE
C
C-----------------------------------------------------------------------
C   READ THE MASK EM.
C   READ FORCING TERMS XF,YF AND WRITE OUT CURRENT TRANSPORTS UBT,VBT.
C-----------------------------------------------------------------------
C
      CALL OGET(KFLDS,NWDS,6*NWDS+1,EM)
C
      CALL OGET(KFLDS,NWDS,  NWDS+1,XF)
      CALL OGET(KFLDS,NWDS,2*NWDS+1,YF)
C
C   SET Y FORCING ON SYMMETRY ROW TO ZERO
C
      DO 35 I=1,IMT
        YF(I,JMTM1)=0.0
 35   CONTINUE
C
      CALL OPUT(KFLDS,NWDS,  NWDS+1,UBT)
      CALL OPUT(KFLDS,NWDS,2*NWDS+1,VBT)
C
C-----------------------------------------------------------------------
C  CALCULATE THE ARRAY SPLR USED TO SET PSUEDO VALUES OF ETA ON LAND.
C-----------------------------------------------------------------------
C
      DO 40 L=1,NSPL
        I = ISPL(L)
        J = JSPL(L)
        SPLR(L) = 1./( EM(I+1,J) + EM(I-1,J) + EM(I,J+1) + EM(I,J-1) )
 40   CONTINUE
C
C-----------------------------------------------------------------------
C  CALCULATE BAROTROPIC VELOCITIES UT AND VT
C-----------------------------------------------------------------------
C
```

```
      DO 43  J=1,JMT
      DO 43  I=1,IMU
        UT(I,J) = UBT(I,J)*HR(I,J)
        VT(I,J) = VBT(I,J)*HR(I,J)
 43   CONTINUE
C
C-----------------------------------------------------------------------
C  START LOOP TO SUBTRACT FRICTION TERM FROM THE FORCING TERMS XF/YF
C-----------------------------------------------------------------------
C
      DO 45  J=2,JMTM2
      DO 45  J=2,JMTM1
      DO 45  L=1,LSU(J)
      DO 45  I=ISU(J,L),IEU(J,L)
C
C---------------------------------------------------------------
C  CALCULATE FRICTION TERM FOR UBT/VBT.  THEN SUBTRACT THIS FROM THE
C  APPROPRIATE FORCING TERM XF/YF
C---------------------------------------------------------------
C
        FRIC = BHD(J)*DXUR(I)*( DXTR(I+1)*( UT(I+1,J) - UT(I  ,J) )
     >                        - DXTR(I  )*( UT(I  ,J) - UT(I-1,J) ) )
     >       + CHD(J)*DYTR(J+1)*( UT(I,J+1) - UT(I,J  ) )
     >       - DHD(J)*DYTR(J  )*( UT(I,J  ) - UT(I,J-1) )
     >       + GHD(J)*UT(I,J)
     >       - HHD(J)*DXUR(I)*( VT(I+1,J) - VT(I-1,J) )
C
        BTFRIC  = AM*FRIC/HR(I,J)
C
C
          XF(I,J) = XF(I,J) - BTFRIC
C
C
        FRIC = BHD(J)*DXUR(I)*( DXTR(I+1)*( VT(I+1,J) - VT(I  ,J) )
     >                        - DXTR(I  )*( VT(I  ,J) - VT(I-1,J) ) )
     >       + CHD(J)*DYTR(J+1)*( VT(I,J+1) - VT(I,J  ) )
     >       - DHD(J)*DYTR(J  )*( VT(I,J  ) - VT(I,J-1) )
     >       + GHD(J)*VT(I,J)
     >       + HHD(J)*DXUR(I)*( UT(I+1,J) - UT(I-1,J) )
C
        BTFRIC = AM*FRIC/HR(I,J)
C
          YF(I,J) = YF(I,J) - BTFRIC
 45   CONTINUE
C
C---------------------------------------------------------------
C  SET ETAD EQUIVALENT TO ETA.
C---------------------------------------------------------------
      DO 80 J=1,JMT
      DO 80 I=1,IMT
        ETAD(I,J)=ETA(I,J)
 80   CONTINUE
C
C=======================================================================
C  TIME STEP BAROTROPIC EQUATIONS BETWEEN (LARGER) BAROCLINIC TIME STEPS  ======
C=======================================================================
C
      DO 500 ITBT = 1,NB
C
C
C---------------------------------------------------------------
C  FIRST SET PSEUDO ETA VALUES ON LAND FOR USE BY DLPL.
C---------------------------------------------------------------
C
```

```
      DO 50 L=1,NSPL                                                    816200000 D
        I = ISPL(L)                                                     816300000 D
        J = JSPL(L)                                                     816400000 D
        ETA(I,J) = ( EM(I+1,J)*ETA(I+1,J)                               816500000 D
     >             + EM(I-1,J)*ETA(I-1,J)                               816600000 D
     >             + EM(I,J+1)*ETA(I,J+1)                               816700000 D
     >             + EM(I,J-1)*ETA(I,J-1) )*SPLR(L)                     816800000 D
 50   CONTINUE                                                          816900000 D
C                                                                       817000000 CD
      DO 60 I=2,IMTM1                                                   817100000 D
        ETA(I,  1) = ETA(I,2)                                           817200000 D
        ETA(I,JMT) = ETA(I,JMTM1)                                       817300000 D
 60   CONTINUE                                                          817400000 D
C                                                                       817500000 CD
      DO 70 J=2,JMT                                                     817600000 DS
      DO 70 J=2,JMTM1                                                   817700000 D-S
        ETA(  1,J) = ETA(2,J)                                           817800000 D-O
        ETA(IMT,J) = ETA(IMTM1,J)                                       817900000 D-O
        ETA(  1,J) = ETA(IMTM1,J)                                       818000000 OD
        ETA(IMT,J) = ETA(2,J)                                           818100000 OD
 70   CONTINUE                                                          818200000 D
C                                                                       818300000 CD
C-----------------------------------------------------------------------818400000 CD
C  SET ETAD EQUIVALENT TO ETA.                                          818500000 CD
C-----------------------------------------------------------------------818600000 CD
      DO 80 J=1,JMT                                                     818700000 D
      DO 80 I=1,IMT                                                     818800000 D
        ETAD(I,J)=ETA(I,J)                                              818900000 D
 80   CONTINUE                                                          819000000 D
C                                                                       819100000 C
C-----------------------------------------------------------------------819200000 C
C  CALCULATE BAROTROPIC VELOCITIES UT AND VT                            819300000 C
C-----------------------------------------------------------------------819400000 C
C                                                                       819500000 C
      DO 140 J=1,JMT                                                    819600000 S
      DO 140 J=1,JMTM2                                                  819700000 -S
      DO 140 I=1,IMU                                                    819800000
        UT(I,J) = UBT(I,J)*HR(I,J)                                      819900000
        VT(I,J) = VBT(I,J)*HR(I,J)                                      820000000
 140  CONTINUE                                                          820100000
C                                                                       820200000 C-G
C-----------------------------------------------------------------------820300000 C-G
C  SET UP 'DO LOOP' TO PERFORM  EULER BACKWARD TIMESTEP.                820400000 C-G
C-----------------------------------------------------------------------820500000 C-G
C                                                                       820600000 C-G
      DO 90 ID=1,2                                                      820700000 -G
C                                                                       820800000 C
C-----------------------------------------------------------------------820900000 C
C  COMPUTE ETAGRD FROM ETA,UBT,VBT                                      821000000 C
C-----------------------------------------------------------------------821100000 C
C                                                                       821200000 C
      FX = DTBT*WGHT*GRAV*ZDZ(KM)                                       821300000 D
C                                                                       821400000 C
      DO 100 J=2,JMTM1                                                  821500000
      DO 100 L=1,LSE(J)                                                 821600000
      DO 100 I=ISE(J,L),IEE(J,L)                                        821700000
C                                                                       821800000 CD
        DLPL = ETA(I,J+1) + ETA(I+1,J)                                  821900000 D
     >       + ETA(I-1,J) + ETA(I,J-1) - 4.*ETA(I,J)                    822000000 D
C                                                                       822100000 CD
        DLCR = 0.5*( EM(I+1,J+1)*ETA(I+1,J+1)                           822200000 D
     >             + EM(I-1,J+1)*ETA(I-1,J+1)                           822300000 D
     >             + EM(I+1,J-1)*ETA(I+1,J-1)                           822400000 D
     >             + EM(I-1,J-1)*ETA(I-1,J-1)                           822500000 D
```

```
     >            - ( EM(I+1,J+1) + EM(I-1,J+1)
     >              + EM(I+1,J-1) + EM(I-1,J-1) )*ETA(I,J) )
C
C
        ETAGRD(I,J) = DXT2R(I)*DYTR(J)
     >                *( UBT(I  ,J)*DYU(J) + UBT(I  ,J-1)*DYU(J-1)
     >                 -( UBT(I-1,J)*DYU(J) + UBT(I-1,J-1)*DYU(J-1))
     >           + ( VBT(I,J)*DXU(I) + VBT(I-1,J)*DXU(I-1))*CS(J)
     >           -(VBT(I,J-1)*DXU(I) + VBT(I-1,J-1)*DXU(I-1))*CS(J-1))
     >             - FX*DXTR(I)*DYTR(J)*( DLPL - DLCR )
C
 100  CONTINUE
C
C-----------------------------------------------------------------------
C  IF TIDAL THEN COMPUTE ETA AND ETAD FROM ETAGRD AND ETA (OLD)
C-----------------------------------------------------------------------
C
        DO 110 J=1,JMT
        DO 110 I=1,IMT
          ETA(I,J) = EM(I,J)*( ETA(I,J) - DTBT*CSTR(J)*ETAGRD(I,J) )
          ETAD(I,J) = ETA(I,J)
 110    CONTINUE
C
        DO 115 I=1,IMT
          ETAD(I,JMT)=ETAD(I,JMTM1)
          ETA(I,JMT)=ETA(I,JMTM1)
 115    CONTINUE
C
        DO 120 J=2,JMT
          ETA(  1,J) = ETA(IMTM1,J)
          ETAD( 1,J) =ETA( 1,J)
          ETA(IMT,J) = ETA(2,J)
          ETAD(IMT,J)= ETA(IMT,J)
 120    CONTINUE
C-----------------------------------------------------------------------
C  START LOOP TO STEP UBT AND VBT FROM ETAD,UT AND VT.
C-----------------------------------------------------------------------
C
      DO 160 J=2,JMTM2
      DO 160 J=2,JMTM1
        FACTOR=FUV(J)*DTBT*0.5
        FAC2=1.0+FACTOR**2
      DO 160 L=1,LSU(J)
      DO 160 I=ISU(J,L),IEU(J,L)
C
C  CALCULATE H FROM HR(I,J) FOR USE BELOW.
C
      H = 1./HR(I,J)
C
C-------------------------------------------------------------------
C  CALCULATE FRICTION TERM FOR UBT AND THEN CALCULATE UBTGRD.
C-------------------------------------------------------------------
C
        FRIC = BHD(J)*DXUR(I)*( DXTR(I+1)*( UT(I+1,J) - UT(I  ,J) )
     >                        - DXTR(I  )*( UT(I  ,J) - UT(I-1,J) ) )
     >      + CHD(J)*DYTR(J+1)*( UT(I,J+1) - UT(I,J  ) )
     >      - DHD(J)*DYTR(J  )*( UT(I,J  ) - UT(I,J-1) )
     >      + GHD(J)*UT(I,J)
     >      - HHD(J)*DXUR(I)*( VT(I+1,J) - VT(I-1,J) )
C
        BTFRIC  = AM*H*FRIC
C
C
        UBTGRD = XF(I,J)
```

```
     >         + BTFRIC
     >         + H * ( FUV(J)*VT(I,J)*0.5
     >                - GRAV*CSR(J)*DXU2R(I)
     >                  *( ETAD(I+1,J+1) + ETAD(I+1,J)
     >                   - ETAD(I  ,J+1) - ETAD(I  ,J) ) )
C
       UBTDUM= ( UT(I,J) * H )  + DTBT*UBTGRD
C
C-------------------------------------------------------------------
C  CALCULATE FRICTION TERM FOR VBT AND THEN CALC VBTGRD.
C-------------------------------------------------------------------
C
       FRIC = BHD(J)*DXUR(I)*( DXTR(I+1)*( VT(I+1,J) - VT(I  ,J) )
     >                        - DXTR(I  )*( VT(I  ,J) - VT(I-1,J) ) )
     >       + CHD(J)*DYTR(J+1)*( VT(I,J+1) - VT(I,J  ) )
     >       - DHD(J)*DYTR(J  )*( VT(I,J  ) - VT(I,J-1) )
     >       + GHD(J)*VT(I,J)
     >       + HHD(J)*DXUR(I)*( UT(I+1,J) - UT(I-1,J) )
C
       BTFRIC = AM*H*FRIC
C
C
       VBTGRD = YF(I,J)
     >         + BTFRIC
     >         - H * ( FUV(J)*UT(I,J)*0.5
     >                + GRAV*DYU2R(J)
     >                  *( ETAD(I,J+1) + ETAD(I+1,J+1)
     >                   - ETAD(I,J  ) - ETAD(I+1,J  ) ) )
C
       VBTDUM =( VT(I,J) * H ) + DTBT*VBTGRD
C
C---------------------------------------------------------------------------
C  CALCULATE UBT AND VBT FROM UBTDUM AND VBTDUM.
C---------------------------------------------------------------------------
C
      UBT(I,J)=(UBTDUM+FACTOR*VBTDUM)/FAC2
      VBT(I,J)=(VBTDUM-FACTOR*UBTDUM)/FAC2
 160  CONTINUE
C
C--------------------------------------------------------------------
C  SET SYMMETRIC CONDITIONS ON UBT AND VBT.
C--------------------------------------------------------------------
      DO 165 I=1,IMT
        UBT(I,JMT)=UBT(I,JMTM2)
        VBT(I,JMTM1)=0.0
 165  CONTINUE
C
C-------------------------------------------------------------------
C  ALLOW FOR CYCLIC CONDITIONS
C-------------------------------------------------------------------
C
      DO 240 J=2,JMT
      DO 240 J=2,JMTM2
        UBT(  1,J) = UBT(IMUM1,J)
        VBT(  1,J) = VBT(IMUM1,J)
        UBT(IMU,J) = UBT(2,J)
        VBT(IMU,J) = VBT(2,J)
 240  CONTINUE
C
C---------------------------------------------------------------------------
C  IF NOT TIDAL THEN COMPUTE ETAD FROM ETAGRD AND ETA.
C---------------------------------------------------------------------------
C
       DO 170 J=1,JMT
```

```
          DO 170 I=1,IMT
            ETAD(I,J) = EM(I,J)*( ETA(I,J) - DTBT*CSTR(J)*ETAGRD(I,J) )
 170      CONTINUE
C
          DO 175 I=1,IMT
            ETAD(I,JMT)=ETAD(I,JMTM1)
 175      CONTINUE
C
          DO 180 J=2,JMT
            ETAD(   1,J) = ETAD(IMTM1,J)
            ETAD(IMT,J) = ETAD(2,J)
 180      CONTINUE
 90     CONTINUE
C
C---------------------------------------------------------------------------------
C IF NOT TIDAL THEN SET ETA EQUAL TO ETAD.
C---------------------------------------------------------------------------------
C
        DO 200 J=1,JMT
        DO 200 I=1,IMT
          ETA(I,J)=ETAD(I,J)
 200    CONTINUE
C
        ITTBT = (ITT-1)*NB + ITBT
C
C-----------------------------------------------------------------------
C  COMPUTE TOTAL RATE OF CHANGE OF KE FOR EXTERNAL MODE ON ENERGY T'STEPS
C-----------------------------------------------------------------------
        IF(NERGY.EQ.1 .AND. MXP.NE.1) THEN
        DO 330 J=2,JMTM2
        DO 330 J=2,JMTM1
         FX=CS(J)*DYU(J)/(NB*DTBT)
         IF(J.EQ.JMTM1) FX=FX*0.5
           DO 330 I=2,IMUM1
             ENGEXT(1)=ENGEXT(1)+( UBT(I,J)*(UBT(I,J)*HR(I,J)-UT(I,J))
     *                 +VBT(I,J)*(VBT(I,J)*HR(I,J)-VT(I,J)) )*FX*DXU(I)
 330    CONTINUE
        ENDIF
C
C-----------------------------------------------------------------------
C  PRINT OUT THE ENERGIES AND MASS ERROR AT GIVEN TIME STEPS.
C-----------------------------------------------------------------------
C
        IF ( MOD(ITTBT,NDW) .EQ. 0 ) THEN
C
        ETANRG = 0.
        DO 300 J=2,JMTM1
        DO 300 L=1,LSE(J)
        DO 300 I=ISE(J,L),IEE(J,L)
          ETANRG = ETANRG + DXT(I)*CST(J)*DYT(J)*ETA(I,J)**2
 300    CONTINUE
        ETANRG = 0.5*GRAV*ETANRG
C
        FX=1.0
        VELNRG = 0.
        DO 310 J=2,JMTM2
        DO 310 J=2,JMTM1
        IF (J.EQ.JMTM1) FX=0.5
        DO 310 L=1,LSU(J)
        DO 310 I=ISU(J,L),IEU(J,L)
          VELNRG = VELNRG
     >          + FX*DXU(I)*CS(J)*DYU(J)*HR(I,J)
     >            *( UBT(I,J)**2 + VBT(I,J)**2 )
 310    CONTINUE
```

```
      VELNRG = 0.5*VELNRG
C
      TOTNRG = ETANRG + VELNRG
C
      ERRMSS = 0.
      DO 320 J=2,JMTM1
      DO 320 L=1,LSE(J)
      DO 320 I=ISE(J,L),IEE(J,L)
        ERRMSS = ERRMSS + DXT(I)*CST(J)*DYT(J)*ETA(I,J)
 320  CONTINUE
C
      WRITE (8,900) ITTBT,VELNRG,ETANRG,TOTNRG,ERRMSS
C
      ENDIF
C
C-----------------------------------------------------------------------
C  PLOT FIELDS AT GIVEN TIME STEPS
C-----------------------------------------------------------------------
C
      IF ( MOD(ITTBT,NC) .EQ. 0 ) THEN
C
        ISTRT=1
        ISTOP=IMT
C
        SCL=0.
        WRITE (6,910) 'ETA SURFACE DISPLACEMENT  ',ITTBT
        CALL MATRIX(ETA,IMT,ISTRT,ISTOP,JMT,0,SCL)
C
        WRITE (6,910) 'UBT TRANSPORT  ',ITTBT
        CALL MATRIX(UBT,IMT,ISTRT,ISTOP,JMT,0,SCL)
C
        WRITE (6,910) 'VBT TRANSPORT  ',ITTBT
        CALL MATRIX(VBT,IMT,ISTRT,ISTOP,JMT,0,SCL)
C
C
      ENDIF
C
 500  CONTINUE
C
 900  FORMAT (1X,'AT TIME STEP ',I6,5X,1P,4E12.4)
 910  FORMAT (1X,A,'AT (BAROTROPIC) TIMESTEP',I7)
C
      RETURN
      END
```

K. Listing of the new base code

For users who prefer to use a complete base code for modifications, we include a complete listing of the revised code.

```
*COMDECK PARAM
      PARAMETER (IMT=??,JMT=??,KM=??,NT=??,LSEG=??,MSPL=??,LBC=??
     *,LSEGF=??,JFRST=??,JFT0=??,JFT1=??,JFT2=??,JFU0=??,JFU1=??,JFU2=??
C
C  FILTER T TO YIELD EQUIV DX AT J=JFT0 FROM J=JFRST TO JFT1 AND
C       J=JFT2 TO JMTM1
C  FILTER U TO YIELD EQUIV DX AT J=JFU0 FROM J=JFRST TO JFU1 AND
C       J=JFU2 TO JMTM2
C
     *,IMU=IMT-1
     *,IMU=IMT
     *,IMTP1=IMT+1,IMTM1=IMT-1,IMTM2=IMT-2,IMUM1=IMU-1,IMUM2=IMU-2
     *,JMTP1=JMT+1,JMTM1=JMT-1,JMTM2=JMT-2,JSCAN=JMTM2
     *                                            +1
     *,KMP1=KM+1,KMP2=KM+2,KMM1=KM-1
     *,NSLAB=IMT*((NT+2)*KM+LBC),NWDS=IMT*JMT,NSWICH=LBC*IMT
     *,JSKPT=JFT2-JFT1,JSKPU=JFU2-JFU1
     *,NJTBFT=(JFT1-JFRST+1)+(JMTM1-JFT2+1)
     *,NJTBFU=(JFU1-JFRST+1)+(JMTM1-JFU2+1)
     *,NSSIF=2*(NJTBFT+NJTBFU)*LSEGF*KM                                 *
     *,NDICES=2*(2*JMT*LSEG+JMT+MSPL)+1                                 *
     *                         +NSSIF
     *,IMTKM=IMT*KM,NTMIN2=NT+1/NT,NKFLDS=7+(NDICES/NWDS))
      IMPLICIT HALF PRECISION(A-H,O-Z)
*COMDECK FULLWD
      REAL TTSEC,AREA,VOLUME,AKNTRL,PAD,ENGINT,ENGEXT,TTDTOT,BUOY,
     *     PLICIN,PLICEX,EKTOT,DTABS,TVAR
      LOGICAL EB
      COMMON /FULLWD/ ITT,TTSEC,AREA,VOLUME,AKNTRL(6),PAD(10),
     * ENGINT(8),ENGEXT(8),TTDTOT(6,NT),BUOY,PLICIN,PLICEX,EKTOT,
     * DTABS(NT),TVAR(NT),NFIRST,NLAST,NMIX,NNERGY,NWRITE,NA,NB,NC,
     * MXSCAN,NDISKB,NDISK,NDISKA,KONTRL,KFLDS,LABS(3),MIX,MXP,NERGY,
     * MSCAN,MSB,NDW,NTSI,KAR(KM),
     * KMT(IMT),KMTP(IMT),KMU(IMT),KMUP(IMT),EB,
     * ISE(JMT,LSEG),IEE(JMT,LSEG),ISU(JMT,LSEG),IEU(JMT,LSEG),        *
     * LSE(JMT),LSU(JMT),ISPL(MSPL),JSPL(MSPL),NSPL                     *
     *,ISTF(NJTBFT,LSEGF,KM), IETF(NJTBFT,LSEGF,KM),
     * ISUF(NJTBFU,LSEGF,KM), IEUF(NJTBFU,LSEGF,KM),
     * SPSIN(IMT), SPCOS(IMT)
*COMDECK SCALAR
      COMMON /SCALAR/ DTTS,DTUV,DTBT,C2DTTS,C2DTUV,AH,AM,FKPH,          *
     * FKPM,ACOR,WGHT,OMEGA,RADIUS,GRAV,RADIAN,PI,SWLDEG                *
*COMDECK ONEDIM
      COMMON /ONEDIM/
     * DXT  (IMT),DXTR (IMT),DXT2R(IMT),DXU  (IMT),DXUR (IMT),DXU2R(IMT)
     *,DXU4R(IMT),DXT4R(IMT),SFU  (IMT),SFUB (IMT),SFV  (IMT),SFVB (IMT)
     *,XFBT(IMT),YFBT(IMT)                                              *
     *,DYT  (JMT),DYTR (JMT),DYT2R(JMT),DYU  (JMT),DYUR (JMT),DYU2R(JMT)
     *,DYU4R(JMT),DYT4R(JMT),CS   (JMT),CSR  (JMT),CST  (JMT),CSTR (JMT)
     *,PHI  (JMT),PHIT (JMT),SINE (JMT),TNG  (JMT)
     *,C2DZ ( KM),DZ   ( KM),DZ2R ( KM),EEH  ( KM),EEM  ( KM),FFH  ( KM)
     *,FFM  ( KM),ZDZ  ( KM)
     *,DZZ (KMP1),DZZ2R(KMP1),ZDZZ(KMP1),TINIT(KM,NT)
*COMDECK FIELDS
      COMMON /FIELDS/                                                   *
     * ETA(IMT,JMT),UBT(IMT,JMT),VBT(IMT,JMT),HR(IMT,JMT)               *
*COMDECK WRKSPA
      COMMON /WORKSP/
     *TA (IMT,KM,NT),UA (IMT,KM),VA (IMT,KM),BCON(IMT,LBC),
     *TBP(IMT,KM,NT),UBP(IMT,KM),VBP(IMT,KM),
     *  FKMUP(IMT),WSYP(IMT),
     *TP (IMT,KM,NT),UP (IMT,KM),VP (IMT,KM),
     *  FKMTP(IMT),WSXP(IMT),
```

```
     *TB (IMT,KM,NT),UB (IMT,KM),VB (IMT,KM),
     *  FKMU (IMT),WSY (IMT),
     *T  (IMT,KM,NT),U  (IMT,KM),V  (IMT,KM),
     *  FKMT (IMT),WSX (IMT),
     *TBM(IMT,KM,NT),UBM(IMT,KM),VBM(IMT,KM),
     *  FKMUM(IMT),WSYM(IMT),
     *TM (IMT,KM,NT),UM (IMT,KM),VM (IMT,KM),
     *  FKMTM(IMT),WSXM(IMT)
      COMMON /WORKSP/
     * UCLIN(IMT,KM),VCLIN(IMT,KM),USAV (IMT,KM),VSAV (IMT,KM),
     * RHON (IMT,KM),RHOS (IMT,KM),FUW  (IMT,KM),FVN  (IMT,KM),
     * FVSU (IMT,KM),FVST (IMT,KM),
     * FMM  (IMT,KM),FM   (IMT,KM),FMP (IMT,KM),
     *              GM   (IMT,KM),
     *    UOVER(IMT),UDIF (IMT,KM),UUNDER(IMT),
     *    VOVER(IMT),VDIF (IMT,KM),VUNDER(IMT),
     * W(IMT,KMP1),TEMPA(IMT,KMP1),TEMPB(IMT,KMP1),
     * TDIF(IMT,KMP2,NTMIN2),
     * ZUENG(IMT),ZVENG(IMT)
      COMMON /WORKSP/
     * DXTQ  (IMT,KM),DXUQ  (IMT,KM),DXT4RQ(IMT,KM),DXU2RQ(IMT,KM),
     * DZ2RQ (IMT,KM),DZZQ  (IMT,KM),DZZ2RQ(IMT,KM),C2DZQ (IMT,KM),
     * EEHQ  (IMT,KM),EEMQ  (IMT,KM),FFHQ  (IMT,KM),FFMQ  (IMT,KM),
     * CQ (IMT,KM,9  ),TOQ (IMT,KM  ),SOQ (IMT,KM  ),
     * CIQ(IMT,KM,9,2),TOIQ(IMT,KM,2),SOIQ(IMT,KM,2)
*COMDECK WRKSPB
      COMMON /WORKSP/
     * XF(IMT,JMT),YF(IMT,JMT),ETAD(IMT,JMT)
     * ,ETAGRD(IMT,JMT),UT(IMT,JMT),VT(IMT,JMT),
     * BHD(JMT),CHD(JMT),DHD(JMT),GHD(JMT),HHD(JMT),FUV(JMT),EM(IMT,JMT)
     * ,SPLR(MSPL)
*COMDECK BITVEC
      BIT KEVENBV(IMT,KM),KODDBV(IMT,KM),KALTBV(IMT,KM,2)
      EQUIVALENCE (KODDBV,KALTBV(1,1,1)),(KEVENBV,KALTBV(1,1,2))
      COMMON /BITVEC/ KALTBV
*COMDECK TIME
      REAL T0,T1,T2,T3,TIME
      COMMON /TIME/ TIME(10)
```

```
*DECK OCEAN
      PROGRAM OCEAN
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C======================================================================
C
C  OCEAN IS THE PRIMARY CALLING ROUTINE.  IT PERFORMS ALL
C        OPERATIONS WHICH NEED BE DONE ONLY ONCE AT THE
C        BEGINNING OF EACH RUN OF THE MODEL, CALLS STEP
C        ONCE PER TIMESTEP, AND ATTENDS TO OPERATIONS
C        WHICH MUST BE DONE ONLY AT THE END OF EACH RUN.
C
C        THE CALLING SEQUENCE OF THE CODE IS AS FOLLOWS:
C
C                              ---> CLINIC - STATE
C        ODAM     ODAM      /
C          \        \      /
C    -->  OCEAN --> STEP -- - ---> TRACER - STATEC
C                   /      \
C                MATRIX     \
C                              ---> TROPIC
C
C======================================================================
C
C---------------------------------------------------------------------
C  DEFINE GLOBAL DATA
C---------------------------------------------------------------------
C
*CALL PARAM
*CALL FULLWD
*CALL SCALAR
*CALL ONEDIM
*CALL FIELDS
*CALL WRKSPA
*CALL BITVEC
*CALL TIME
C
C---------------------------------------------------------------------
C  DEFINE AND EQUIVALENCE LOCAL DATA; DEFINE NAMELIST INPUT
C---------------------------------------------------------------------
C
      DIMENSION FKMP(IMT,JMT),FKMQ(IMT,JMT),FINS(3),EM(IMT,JMT)
      CHARACTER*1 DOT,BLK,ABT
      DIMENSION ABT(IMT)
      DIMENSION KPR(IMT)
      EQUIVALENCE (ETA,FKMP,FINS),(UBT,FKMQ)
      REAL SUMDY
      REAL AMF,AHF,FKPMF,FKPHF,DTTSF,DTUVF,DTBTF,ACORF,WGHTF,
     * TINITF(KM,NT)
      NAMELIST /EDDY/ AMF,AHF,FKPMF,FKPHF /TSTEPS/ DTTSF,DTUVF,DTBTF
     * /PARMS/ ACORF
     * ,WGHTF
     */CONTRL/ NFIRST,NLAST,NNERGY,NMIX,NWRITE,NDW,NTSI,NA,NB,NC
      NAMELIST /TSPROF/ TINITF
      DATA DOT/'.'/,BLK/' '/
C
C---------------------------------------------------------------------
C  BEGIN EXECUTABLE CODE
C---------------------------------------------------------------------
C
C======================================================================
C  BEGIN INTRODUCTORY SECTION WHICH IS NEEDED FOR EACH RUN  ============
C       (INCLUDING RESTARTS)                                ============
```

```
C=====================================================================
C
C
      DO 10 LTIME=1,10
 10     TIME(LTIME)=0.
      CALL GETIME(T3,T1)
C
C---------------------------------------------------------------------
C  SET THE TYPE OF MIXING TIMESTEP.  IF EB=.TRUE., AN EULER
C  BACKWARD STEP IS DONE; IF EB=.FALSE., A FORWARD STEP IS DONE.
C---------------------------------------------------------------------
C
      EB=.TRUE.
C
C---------------------------------------------------------------------
C  IN HALFWORD MODE, NWDS AND NSLAB MUST BE EVEN FOR I/O REASONS
C---------------------------------------------------------------------
C
      IF(MOD(NWDS,2).NE.0 .OR. MOD(NSLAB,2).NE.0) THEN
        PRINT 9831
        STOP 9831
      ENDIF
 9831 FORMAT(1X,'NWDS & NSLAB MUST BE EVEN FOR QDAM IN HALFWORD MODE')
C
C---------------------------------------------------------------------
C  INITIALIZE VARIOUS QUANTITIES
C---------------------------------------------------------------------
C
      NWRITE=1000000
      NDW=1000000
      NTSI=1
      NA=0
      NB=0
      NC=0
      KONTRL=11
      KFLDS=12
      LABS(1)=13
      LABS(2)=14
      LABS(3)=15
      PI=3.1415927
      OMEGA=3.1415927/43082.
      RADIUS=6370.E5
      RADIAN=57.29578
      GRAV=980.6
C
C---------------------------------------------------------------------
C  SET THE LATITUDE (IN DEGREES) OF THE SOUTHERN WALL
C---------------------------------------------------------------------
C
      SWLDEG=??
C
C---------------------------------------------------------------------
C  SET Y DIMENSION OF BOXES IN DEGREES AND CONVERT TO CENTIMETERS
C---------------------------------------------------------------------
C
      DO 52 J=2,JMTM1
        DYT(J)=??
        DYT(J)=DYT(J)*RADIUS/RADIAN
 52   CONTINUE
      DYT(1)=DYT(2)
      DYT(JMT)=DYT(JMTM1)
C
C---------------------------------------------------------------------
C  SET X DIMENSION OF BOXES IN DEGREES AND CONVERT TO CENTIMETERS
```

```
C---------------------------------------------------------------------
C
      DO 57 I=2,IMTM1
        DXT(I)=??
        DXT(I)=DXT(I)*RADIUS/RADIAN
 57   CONTINUE
      DXT(1)=DXT(2)
      DXT(IMT)=DXT(IMTM1)
C
C  SET CYCLIC CONDITIONS ON DXT
C
      DXT(1)=DXT(IMTM1)
      DXT(IMT)=DXT(2)
C
C---------------------------------------------------------------------
C  SET Z DIMENSION OF BOXES (IN CENTIMETERS)
C---------------------------------------------------------------------
C
      DZ(1)=??
C
C---------------------------------------------------------------------
C  READ IN RUN PARAMETERS
C---------------------------------------------------------------------
C
      READ (5,CONTRL)
      WRITE(6,CONTRL)
      READ (5,EDDY)
      WRITE(6,EDDY)
      READ (5,TSTEPS)
      WRITE(6,TSTEPS)
      READ (5,PARMS)
      WRITE(6,PARMS)
      AM=AMF
      AH=AHF
      FKPM=FKPMF
      FKPH=FKPHF
      DTTS=DTTSF
      DTUV=DTUVF
      DTBT=DTBTF
      ACOR=ACORF
      WGHT=WGHTF
C
C---------------------------------------------------------------------
C  COMPUTE AUXILIARY ARRAYS BASED UPON THE SPACING SPECIFIED ABOVE
C---------------------------------------------------------------------
C
      DO 100 K=1,KM
        C2DZ(K)=2.0*DZ(K)
        DZ2R(K)=1.0/C2DZ(K)
 100  CONTINUE
      DZZ(1)=0.5*DZ(1)
      ZDZ(1)=DZ(1)
      DO 110 K=2,KM
        DZZ(K)=0.5*(DZ(K-1)+DZ(K))
        ZDZ(K)=ZDZ(K-1)+DZ(K)
 110  CONTINUE
      DZZ(KM+1)=0.5*DZ(KM)
      DZZ2R(KMP1)=0.5/DZZ(KMP1)
      ZDZZ(1)=DZZ(1)
      DO 120 K=1,KM
        DZZ2R(K)=.5/DZZ(K)
        ZDZZ(K+1)=ZDZZ(K)+DZZ(K+1)
        EEH(K)=FKPH/(DZ(K)*DZZ(K))
        FFH(K)=FKPH/(DZ(K)*DZZ(K+1))
```

```
        EEM(K)=FKPM/(DZ(K)*DZZ(K))
        FFM(K)=FKPM/(DZ(K)*DZZ(K+1))
        KAR(K)=K
 120  CONTINUE
      PHI(1)=SWLDEG/RADIAN
      PHIT(1)=PHI(1)-.5*DYT(1)/RADIUS
      SUMDY=PHI(1)
      DYU(JMT)=DYT(JMT)
      DO 130 J=1,JMT
        IF(J.NE.JMT) DYU(J)=.5*(DYT(J)+DYT(J+1))
        DYTR(J)=1./DYT(J)
        DYT2R(J)=.5/DYT(J)
        DYT4R(J)=.25/DYT(J)
        DYUR(J)=1./DYU(J)
        DYU2R(J)=.5/DYU(J)
        DYU4R(J)=.25/DYU(J)
        IF(J.NE.JMT) SUMDY=SUMDY+DYT(J+1)/RADIUS
        IF(J.NE.JMT) PHI(J+1)=SUMDY
        IF(J.NE.1) PHIT(J)=.5*(PHI(J-1)+PHI(J))
        CST(J)=COS(PHIT(J))
        CS (J)=COS(PHI (J))
        SINE(J)=SIN(PHI(J))
        CSTR(J)=1.0/CST(J)
        CSR(J)=1.0/CS(J)
        TNG(J)=SINE(J)/CS(J)
 130  CONTINUE
      DXU(IMT)=DXT(IMT)
      DXU(IMT)=.5*(DXT(2)+DXT(3))                                          O
      DO 140  I=1,IMT
        IF(I.NE.IMT) DXU(I)=.5*(DXT(I)+DXT(I+1))
        DXTR(I)=1./DXT(I)
        DXT2R(I)=.5/DXT(I)
        DXT4R(I)=.25/DXT(I)
        DXUR(I)=1./DXU(I)
        DXU2R(I)=.5/DXU(I)
        DXU4R(I)=.25/DXU(I)
  140 CONTINUE
C                                                                          WC
C------------------------------------------------------------------        WC
C  CREATE A BIT ARRAY FOR LATER WHERE STATEMENTS                           WC
C  (..NOTE.. "KEVENBV" AND "KODDBV" ARE EQUIVALENCED WITH "KALTBV")        WC
C------------------------------------------------------------------        WC
C                                                                          WC
      LENV=2*IMT*KM                                                        W
      KALTBV(1,1,1;LENV)=B'0'                                              W
      KALTBV(1,1,1;LENV)=.NOT.KALTBV(1,1,1;LENV)                           W
      DO 145 K=1,KM                                                        W
        IF (K-2*(K/2).NE.0) THEN                                           W
          KEVENBV(1,K;IMT)=B'0'                                            W
        ELSE                                                               W
          KODDBV(1,K;IMT)=B'0'                                             W
        ENDIF                                                              W
 145  CONTINUE                                                             W
C                                                                          FC
C------------------------------------------------------------------        FC
C  COMPUTE SIN AND COS VALUES FOR VECTOR CORRECTION BEFORE FILTER          FC
C------------------------------------------------------------------        FC
C                                                                          FC
      FX=1.0E-10                                                           F
      FXA=DXT(1)/RADIUS                                                    F
      DO 670 I=2,IMUM1                                                     F
        FXB=FXA*FLOAT(I-2)                                                 F
        SPSIN(I)=SIN(FXB)                                                  F
        SPCOS(I)=COS(FXB)                                                  F
```

```
        IF(ABS(SPSIN(I)).LT.FX)SPSIN(I)=0.0
        IF(ABS(SPCOS(I)).LT.FX)SPCOS(I)=0.0
  670 CONTINUE
      SPSIN(1)=0.0
      SPCOS(1)=0.0
      SPSIN(IMU)=0.0
      SPCOS(IMU)=0.0
C
C-----------------------------------------------------------------------
C  PRINT GRID GEOMETRY ARRAYS
C-----------------------------------------------------------------------
C
      PRINT 9701
 9701 FORMAT(50H0 GRID BOX THICKNESS  'DZ'                         )
      PRINT 970, DZ
      PRINT 9702
 9702 FORMAT(50H0 GRID POINT SEPARATION  'DZZ'                     )
      PRINT 970, DZZ
      PRINT 9703
 9703 FORMAT(50H0 DEPTH OF BOX BOTTOM  'ZDZ'                       )
      PRINT 970, ZDZ
      PRINT 9704
 9704 FORMAT(50H0 DEPTH OF GRID POINT  'ZDZZ'                      )
      PRINT 970, ZDZZ
      PRINT 9705
 9705 FORMAT(50H0 LATITUDE OF T,S POINTS (RADIANS)  'PHIT'         )
      PRINT 970, PHIT
      PRINT 9706
 9706 FORMAT(50H0 LATITUDE OF U,V POINTS (RADIANS)  'PHI'          )
      PRINT 970, PHI
      PRINT 9707
 9707 FORMAT(50H0 COSINE OF T,S LATITUDE  'CST'                    )
      PRINT 970, CST
      PRINT 9708
 9708 FORMAT(50H0 COSINE OF U,V LATITUDE  'CS'                     )
      PRINT 970, CS
      PRINT 9709
 9709 FORMAT(50H0 SINE OF U,V LATITUDE  'SINE'                     )
      PRINT 970, SINE
 970  FORMAT(1X,10E13.5)
C
C-----------------------------------------------------------------------
C  OPEN THE DISK DATASETS
C-----------------------------------------------------------------------
C
      CALL OSTART(KONTRL,20,20,1)
      CALL OSTART(KFLDS,NKFLDS*NWDS,NWDS,1)
      NBUF=2
      CALL OSTART(LABS(1),JMT*NSLAB,NSLAB,NBUF)
      CALL OSTART(LABS(2),JMT*NSLAB,NSLAB,NBUF)
      CALL OSTART(LABS(3),JMT*NSLAB,NSLAB,NBUF)
      IF(NFIRST.EQ.0) CALL ORD(21)
C
C=======================================================================
C  END INTRODUCTORY SECTION  ===========================================
C=======================================================================
C
C=======================================================================
C  BEGIN SECTION WHICH IS EXECUTED ONLY WHEN STARTING  =================
C       A RUN FROM SCRATCH                             =================
C=======================================================================
C
      IF (NFIRST.EQ.0) GO TO 160
C
```

```
C-------------------------------------------------------------------
C  SET MAXIMUM LEVEL INDICATORS FOR TOPOGRAPHY
C-------------------------------------------------------------------
C
      DO 690 J=1,JMT
      DO 690 I=1,IMT
        FKMP(I,J)=0
        FKMQ(I,J)=0
  690 CONTINUE
C
C  1ST, SET NUMBER OF VERTICAL LEVELS FOR T POINTS
C
      DO 700 J=2,JMTM1
      DO 700 I=2,IMTM1
        FKMP(I,J)=??
  700 CONTINUE
C
C  SET CYCLIC BOUNDARY CONDITIONS
C
      DO 728 J=1,JMT
        FKMP(  1,J)=FKMP(IMTM1,J)
        FKMP(IMT,J)=FKMP(    2,J)
  728 CONTINUE
C
C  2ND, COMPUTE NUMBER OF VERTICAL LEVELS AT EACH U,V POINT
C
      DO 730 J=1,JMTM1
      DO 730 I=1,IMTM1
       FKMQ(I,J)=MIN(FKMP(I,J),FKMP(I+1,J),FKMP(I,J+1),FKMP(I+1,J+1))
  730 CONTINUE
C
C  SET CYCLIC CONDITIONS
C
      DO 732 J=1,JMT
        FKMQ(IMT,J)=FKMQ(2,J)
  732 CONTINUE
C
C--------------------------------------------------------------------
C  CALCULATE DO LOOP INDICES FOR ETA AND EXTERNAL MODE.
C--------------------------------------------------------------------
C
      DO 748 N=1,NDICES
        ISE(N,1)=0
  748 CONTINUE
C
      DO 750 J=2,JMTM1
        L=0
        IF ( FKMP(1,J).GT.0 .AND. FKMP(2,J).GT.0 ) THEN
          L=L+1
          ISE(J,L)=2
        ENDIF
        DO 752 I=2,IMTM1
          IF ( FKMP(I-1,J).EQ.0 .AND. FKMP(I,J).GT.0 ) THEN
            L=L+1
            ISE(J,L)=I
          ENDIF
          IF ( FKMP(I,J).GT.0 .AND. FKMP(I+1,J).EQ.0 ) IEE(J,L)=I
  752   CONTINUE
        IF ( FKMP(IMTM1,J).GT.0 .AND. FKMP(IMT,J).GT.0 ) IEE(J,L)=IMTM1
        LSE(J)=L
  750 CONTINUE
C
      MLSU=0
      DO 760 J=2,JMTM2
```

```
      DO 760 J=2,JMTM1
        L=0
        IF ( FKMQ(1,J).GT.0 .AND. FKMQ(2,J).GT.0 ) THEN
          L=L+1
          ISU(J,L)=2
        ENDIF
        DO 762 I=2,IMUM1
          IF ( FKMQ(I-1,J).EQ.0 .AND. FKMQ(I,J).GT.0 ) THEN
            L=L+1
            ISU(J,L)=I
          ENDIF
          IF ( FKMQ(I,J).GT.0 .AND. FKMQ(I+1,J).EQ.0 ) IEU(J,L)=I
 762    CONTINUE
        IF ( FKMQ(IMUM1,J).GT.0 .AND. FKMQ(IMU,J).GT.0 ) IEU(J,L)=IMUM1
        LSU(J)=L
        MLSU=MAX(LSU(J),MLSU)
 760  CONTINUE
C
        IF ( MLSU.GT.LSEG ) THEN
          WRITE(6,9920) ' PARAMETER LSEG TOO SMALL!',
     >                  ' MUST BE AT LEAST ',MLSU
          STOP
        ENDIF
C
C-----------------------------------------------------------------------
C  CALCULATE MASK FIELD AT ETA POINTS.
C-----------------------------------------------------------------------
C
      DO 770 J=1,JMT
      DO 770 I=1,IMT
        IF ( FKMP(I,J).GT.0 ) THEN
          EM(I,J)=1.
        ELSE
          EM(I,J)=0.
        ENDIF
 770  CONTINUE
C
C-----------------------------------------------------------------------
C  CALCULATE INDEXING ARRAY FOR 'INTERIOR' ETA POINTS ON COAST.  THIS
C  IS USED FOR THE CALCUATION OF THE DEL-SQUARED TERM ON ETA.
C-----------------------------------------------------------------------
C
      NSPL=0
      DO 780 J=2,JMTM1
      DO 780 I=2,IMTM1
        IF ( EM(I,J).EQ.0 ) THEN
          SPL=EM(I,J+1)+EM(I+1,J)+EM(I-1,J)+EM(I,J-1)
          IF ( SPL.NE.0 ) THEN
            NSPL=NSPL+1
            ISPL(NSPL)=I
            JSPL(NSPL)=J
          ENDIF
        ENDIF
 780  CONTINUE
C
      IF ( NSPL.GT.MSPL ) THEN
        WRITE(6,9920) ' PARAMETER MSPL TOO SMALL!',
     >                ' MUST BE AT LEAST ',NSPL
        STOP
      ENDIF
C
C-----------------------------------------------------------------------
C  PRINT OUT RUN DETAILS
C-----------------------------------------------------------------------
```

```
C                                                                      *136600> 136600000 C
      WRITE (6,9960)                                                   *<610000 136610000
9960    FORMAT (//1X,'DO LOOP INDICES FOR SURFACE DISPLACEMENT  (ETA)'/)*<620000 136620000
      DO 9962 J=JMT,1,-1                                               *<630000 136630000
        WRITE (6,9970) (J,(ISE(J,L),IEE(J,L),L=1,LSE(J)))              *<640000 136640000
9962  CONTINUE                                                         *<650000 136650000
C                                                                      *<660000 136660000 C
      WRITE (6,9964)                                                   *<670000 136670000
9964    FORMAT (//1X,'DO LOOP INDICES FOR TRANSPORTS  (UBT,VBT)'/)     *<680000 136680000
      DO 9966 J=JMT,1,-1                                               *<690000 136690000
        WRITE (6,9970) (J,(ISU(J,L),IEU(J,L),L=1,LSU(J)))              *136700> 136700000
9966  CONTINUE                                                         *<710000 136710000
C                                                                      *<720000 136720000 C
      WRITE (6,'(/)')                                                  *<730000 136730000
C                                                                      *<740000 136740000 C
9920  FORMAT (5X,A/5X,A,I5)                                            *<750000 136750000
9970  FORMAT (5X,I5,' ... ',20I5)                                      *<760000 136760000
C                                                                       138800> 138800000 FC
C-----------------------------------------------------------------------138900> 138900000 FC
C  FIND AND PRINT START & END INDICES FOR FILTERING                     139000> 139000000 FC
C-----------------------------------------------------------------------139100> 139100000 FC
C                                                                       139200> 139200000 FC
      PRINT 833                                                         139300> 139300000 F
      IF (LSEGF.GT.11) PRINT 834                                        139400> 139400000 F
      PRINT 835                                                         139500> 139500000 F
      CALL FINDEX(FKMP,NJTBFT,KM,JFT1,JFT2,IMT,ISTF,IETF)               139600> 139600000 F
      PRINT 836                                                         139700> 139700000 F
      CALL FINDEX(FKMQ,NJTBFU,KM,JFU1,JFU2,IMU,ISUF,IEUF)               139800> 139800000 F
  833 FORMAT (1H1,'START AND END INDICES FOR FOURIER FILTERING:'/)      140100> 140100000 F
  834 FORMAT (1X,'ONLY 11 SETS OF INDICES FIT ACCROSS THE PAGE.',       140200> 140200000 F
     *       '  OTHERS WILL NOT BE PRINTED.'/)                          140300> 140300000 F
  835 FORMAT (///1X,'FILTERING INDICES FOR T,S:')                       140400> 140400000 F
  836 FORMAT (///1X,'FILTERING INDICES FOR U,V:')                       140500> 140500000 F
C                                                                       140700> 140700000 C
C-----------------------------------------------------------------------140800> 140800000 C
C  COMPUTE FIELD OF RECIPROCAL DEPTH                                    140900> 140900000 C
C-----------------------------------------------------------------------141000> 141000000 C
C                                                                       141100> 141100000 C
      DO 790 J=1,JMT                                                    141200> 141200000
      DO 790 I=1,IMT                                                    141300> 141300000
        HR(I,J)=0.0                                                     141400> 141400000
        IF(FKMQ(I,J).NE.0) HR(I,J)=1./ZDZ(INT(FKMQ(I,J)))               141500> 141500000
 790  CONTINUE                                                          141600> 141600000
C                                                                       141700> 141700000 SC
C  SET SYMMETRY CONDITIONS                                              141800> 141800000 SC
C                                                                       141900> 141900000 SC
      DO 792 I=1,IMT                                                    142000> 142000000 S
        HR(I,JMT)=HR(I,JMTM2)                                           142100> 142100000 S
 792  CONTINUE                                                          142200> 142200000 S
C                                                                       142300> 142300000 C
C-----------------------------------------------------------------------142400> 142400000 C
C  COMPUTE THE SURFACE AREA AND VOLUME OF THE OCEAN                     142500> 142500000 C
C-----------------------------------------------------------------------142600> 142600000 C
C                                                                       142700> 142700000 C
      AREA=0.0                                                          142800> 142800000
      VOLUME=0.0                                                        142900> 142900000
      DO 800 J=2,JMTM1                                                  143000> 143000000
      DO 800 I=2,IMTM1                                                  143100> 143100000
        IF(FKMP(I,J).GT.0) THEN                                         143200> 143200000
          AREA=AREA+CST(J)*DXT(I)*DYT(J)                                143300> 143300000
          VOLUME=VOLUME+CST(J)*DXT(I)*DYT(J)*ZDZ(INT(FKMP(I,J)))        143400> 143400000
        ENDIF                                                           143500> 143500000
 800  CONTINUE                                                          143600> 143600000
C                                                                       143700> 143700000 C
```

```
C-----------------------------------------------------------------------
C  PRINT TOPOGRAPHY MAP
C  (..NOTE.. THE NUMBER OF LEVELS ARE PRINTED IN HEX;
C           A DOT SUPERIMPOSED ===> ADD AN ADDITIONAL 16)
C-----------------------------------------------------------------------
C
      PRINT 950
  950 FORMAT(50H1 NUMBER OF LEVELS AT T,S POINTS AND U,V POINTS    )
      DO 810 IBK=1,IMT,65
        PRINT 960
  960   FORMAT(/)
        ISP=IBK
        IEPT=IBK+64
        IEPU=IBK+64
        IF(IEPT.GT.IMT) IEPT=IMT
        IF(IEPU.GT.IMU) IEPU=IMU
      DO 810 JREV=1,JMT
        J=JMT-JREV+1
        IF(J.NE.JMT) THEN
          DO 968 I=1,IMT
            KPR(I)=FKMQ(I,J)
  968     CONTINUE
          PRINT 972, (KPR(I),I=ISP,IEPU)
  972     FORMAT(2X,65(1X,Z1))
          DO 969 I=1,IMT
            ABT(I)=BLK
  969     CONTINUE
          DO 952 I=ISP,IEPU
            IF(KPR(I).GT.15)ABT(I)=DOT
  952     CONTINUE
          PRINT 971,(ABT(I),I=ISP,IEPU)
  971     FORMAT(2H+ ,65(1X,A1))
        ENDIF
        DO 953 I=1,IMT
          KPR(I)=FKMP(I,J)
  953   CONTINUE
        PRINT 982, (KPR(I),I=ISP,IEPT)
  982   FORMAT(1X,65(1X,Z1))
        DO 979 I=1,IMT
          ABT(I)=BLK
  979   CONTINUE
        DO 954 I=ISP,IEPT
          IF(KPR(I).GT.15)ABT(I)=DOT
  954   CONTINUE
        PRINT 981,(ABT(I),I=ISP,IEPT)
  981   FORMAT(1H+,65(1X,A1))
  810 CONTINUE
C
C-----------------------------------------------------------------------
C  PRINT AREA AND VOLUME OF THE OCEAN, AS WELL AS START & END
C  INDICES FOR THE STREAM FUNCTION CALCULATION
C-----------------------------------------------------------------------
C
      PRINT 940, AREA,VOLUME
  940 FORMAT(//,15H SURFACE AREA =,1PE13.6,5X,9H VOLUME =,1PE13.6)
C
C-----------------------------------------------------------------------
C  READ IN INITIAL TRACER VALUES
C-----------------------------------------------------------------------
C
      READ (5,TSPROF)
      WRITE(6,TSPROF)
      DO 832 M=1,NT
      DO 832 K=1,KM
```

```
          TINIT(K,M)=TINITF(K,M)
  832   CONTINUE
C
C-----------------------------------------------------------------------
C  INITIALIZE SLAB DATA ON DISK
C-----------------------------------------------------------------------
C
        DO 880 J=1,JMT
          DO 840 I=1,IMT
C
C  SET WIND STRESS TO SPECIFIED DISTRIBUTION
C
            WSX(I)=??
            WSY(I)=??
C
C  SET MAXIMUM LEVEL INDICATORS TO VALUES COMPUTED ABOVE
C
            FKMT(I)=FKMP(I,J)
            FKMU(I)=FKMQ(I,J)
 840      CONTINUE
          DO 842 K=1,KM
          DO 842 I=1,IMT
C
C  SET INTERNAL MODE VELOCITIES TO ZERO
C
            UB(I,K)=0.0
            U (I,K)=0.0
            VB(I,K)=0.0
            V (I,K)=0.0
          DO 842 M=1,NT
            TB(I,K,M)=0.0
            T (I,K,M)=0.0
 842      CONTINUE
C
C  SET TRACERS OVER OCEAN POINTS TO SPECIFIED VALUES
C
          DO 870 I=1,IMT
            KZ=FKMP(I,J)
            IF(KZ.NE.0) THEN
              DO 860 K=1,KZ
              DO 860 M=1,NT
                TB(I,K,M)=TINIT(K,M)
                IF(M.EQ.2) TB(I,K,M)=TINIT(K,M)-0.035
                T(I,K,M)=TB(I,K,M)
 860          CONTINUE
            ENDIF
 870      CONTINUE
C
C  SEND THE INITIAL SLABS TO DISK
C
          CALL OPUT(LABS(1),NSLAB,(J-1)*NSLAB+1,TB)
          CALL OPUT(LABS(2),NSLAB,(J-1)*NSLAB+1,T )
          CALL OPUT(LABS(3),NSLAB,(J-1)*NSLAB+1,T )
 880    CONTINUE
C
C-----------------------------------------------------------------------
C   INITIALIZE REMAINDER OF DISK
C-----------------------------------------------------------------------
C
C  SEND RECIPROCAL DEPTH AND ETA MASK TO DISC
C
        CALL OPUT(KFLDS,NWDS,6*NWDS+1,EM)
        CALL OPUT(KFLDS,NWDS,       1,HR)
C
```

```
C  SET INITIAL SURFACE DISPLACEMENT AND EXTERNAL TRANSPORTS TO ZERO.
C
      DO 890 J=1,JMT
      DO 890 I=1,IMT
        ETA(I,J)=0.0
        UBT(I,J)=0.0
        VBT(I,J)=0.0
 890  CONTINUE
C
C  SEND INITIALIZED DATA TO DISC
C
      CALL OPUT(KFLDS,NWDS,  NWDS+1,UBT)
      CALL OPUT(KFLDS,NWDS,2*NWDS+1,VBT)
      CALL OPUT(KFLDS,NWDS,3*NWDS+1,ETA)
      CALL OPUT(KFLDS,NWDS,4*NWDS+1,UBT)
      CALL OPUT(KFLDS,NWDS,5*NWDS+1,VBT)
C
C  CONVERT START AND END INDICES TO REAL (THIS IS DONE TO ACCOMODATE
C        THE OPTION OF RUNNING THE MODEL IN HALFWORD MODE; ALSO,
C       NOTE THAT "FINS" IS EQUIVALENCED WITH "ETA")
C
      DO 164 N=1,NDICES
        FINS(N)=ISE(N,1)
 164  CONTINUE
C
C  SET TIMESTEP COUNTER AND TOTAL ELAPSED TIME TO ZERO
C
      ITT=0
      TTSEC=0.0
C
C  SEND REMAINDER OF INITIALIZED DATA TO DISC
C
      CALL OPUT(KONTRL,20,1,ITT)
      CALL OPUT(KFLDS,NDICES,7*NWDS+1,FINS)
C
C=========================================================================
C  END SECTION TO START FROM SCRATCH  ===================================
C=========================================================================
C
C=========================================================================
C  BEGIN SECTION TO TIMESTEP THE MODEL  =================================
C=========================================================================
C
C-----------------------------------------------------------------------
C  READ DISK DATA INTO MEMORY FOR STARTUP
C-----------------------------------------------------------------------
C
C  READ IN TIMESTEP COUNTER, TOTAL ELAPSED TIME, AREA AND VOLUME
C
 160  CALL OGET(KONTRL,20,1,ITT)
C
C  READ IN START AND END INDICES AND CONVERT TO INTEGERS
C
      CALL OGET(KFLDS,NDICES,7*NWDS+1,FINS)
      DO 165 N=1,NDICES
        ISE(N,1)=FINS(N)
 165  CONTINUE
C
C  COMPUTE PERMUTING DISC INDICATORS AND READ IN 2 LEVELS OF
C       STREAM FUNCTION AS WELL AS RECIPROCAL DEPTH.
C
      NDISK =MOD(ITT+0,3)+1
      NDISKA=MOD(ITT+1,3)+1
      CALL OGET(KFLDS,NWDS,3*NWDS+1,ETA)
```

```
      CALL OGET(KFLDS,NWDS,4*NWDS+1,UBT)
      CALL OGET(KFLDS,NWDS,5*NWDS+1,VBT)
      CALL OGET(KFLDS,NWDS,       1, HR)
C
C-----------------------------------------------------------------------
C  INITIALIZE SEVERAL VARIABLES TO ZERO TO AVOID AN "UNINITIALIZED
C       VARIABLE" TYPE OF ERROR LATER WHERE, FOR PURPOSES OF VEC-
C       TORIZATION, THE COMPUTATION PROCEEDS ACROSS LAND POINTS
C-----------------------------------------------------------------------
C
      DO 168 I=1,IMT
        UUNDER(I)=0.0
        VUNDER(I)=0.0
        DO 167 K=1,KMP1
          TEMPA(I,K)=0.0
          TEMPB(I,K)=0.0
 167    CONTINUE
      DO 168 M=1,NT
      DO 168 K=1,KMP2
        TDIF(I,K,M)=0.0
 168  CONTINUE
C
      CALL GETIME(T0,TIME(1))
C
C-----------------------------------------------------------------------
C  PROCEED WITH TIMESTEPPING UNTIL THE SPECIFIED NUMBER
C       OF STEPS HAVE BEEN TAKEN
C-----------------------------------------------------------------------
C
 200  CALL STEP
C
C-----------------------------------------------------------------------
C  SAVE RESTART DATA EVERY NWRITE TIMESTEPS, AND AT END OF RUN IF NA=1
C-----------------------------------------------------------------------
C
      IF(MOD(ITT,NWRITE).EQ.0) CALL OWRT(22)
      IF(ITT.EQ.NLAST .AND. NA.EQ.1) CALL OWRT(23)
      IF(ITT.LT.NLAST) GO TO 200
C
C=======================================================================
C  END TIMESTEPPING OF THE MODEL  ======================================
C=======================================================================
C
C-----------------------------------------------------------------------
C  COMPLETE THE TIMING ANALYSIS AND PRINT IT
C-----------------------------------------------------------------------
C
      CALL GETIME(T1,T2)
      T3=100./(T1-T3)
      TIME(2)=T1-T0-TIME(3)-TIME(4)-TIME(5)
      TIME(3)=TIME(3)-TIME(6)
      TIME(4)=TIME(4)-TIME(7)
      TIME(5)=TIME(5)-TIME(8)
      DO 3325 LTIME=1,10
 3325 TIME(LTIME)=TIME(LTIME)*T3
      PRINT 9601
 9601 FORMAT('0TIMING ANALYSIS',/,7X,'MAIN',6X,'STEP',4X,'CLINIC',1X,
     *'STATE(CL)',4X,'TRACER',1X,'STATE(TR)',5X,'RELAX',5X,'SCANS')
      PRINT 9602,TIME(1),TIME(2),TIME(3),TIME(6),TIME(4),TIME(7),
     *TIME(5),TIME(8)
 9602 FORMAT(1X,13(F9.2,'%'))
      PRINT 9603,TIME(9)
 9603 FORMAT('0% OF TOTAL TIME IN FILTER:',F10.2)
C
```

```
C-----------------------------------------------------------------------
C  CLOSE THE DISC DATASETS
C-----------------------------------------------------------------------
C
      CALL OCLOSE(KONTRL)
      CALL OCLOSE(KFLDS)
      CALL OCLOSE(LABS(1))
      CALL OCLOSE(LABS(2))
      CALL OCLOSE(LABS(3))
      STOP 13000
      END
```

```
*DECK STEP
      SUBROUTINE STEP
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C=====================================================================
C                                                                 ====
C  STEP IS CALLED ONCE PER TIMESTEP.  IT INITIALIZES VARIOUS      ====
C       QUANTITIES, BOOTSTRAPS THE BASIC ROW BY ROW COMPUTATION   ====
C       OF PROGNOSTIC VARIABLES, MANAGES THE I/O FOR THE LATTER,  ====
C       AND PERFORMS VARIOUS ANALYSIS PROCEDURES ON THE PROGRESSING ====
C       SOLUTION.                                                 ====
C                                                                 ====
C=====================================================================
C
C---------------------------------------------------------------------
C  DEFINE GLOBAL DATA
C---------------------------------------------------------------------
C
*CALL PARAM
*CALL FULLWD
*CALL SCALAR
*CALL ONEDIM
*CALL FIELDS
*CALL WRKSPA
*CALL TIME
C
C---------------------------------------------------------------------
C  DEMENSION AND EQUIVALENCE LOCAL DATA
C---------------------------------------------------------------------
C
      DIMENSION TBSLAB(NSLAB),TSLAB(NSLAB),VBR(KM),TBRN(KM,NT),
     *          TBRS(KM,NT),TTN(8,JMT,NTMIN2),TMT(JMT,KM)
      EQUIVALENCE (TBSLAB,TB),(TSLAB,T)
      CHARACTER C8TIME*8,CTIME*10
C
C---------------------------------------------------------------------
C  BEGIN EXECUTABLE CODE
C---------------------------------------------------------------------
C
      CALL GETIME(T0,T1)
      CALL Q5TIME('MASTER=',C8TIME)
      PRINT 110,T0,C8TIME
 110  FORMAT (' ACC TIME(SEC)=',F10.2,5X,Z16)
C
C=====================================================================
C  BEGIN SECTION FOR THE INITIALIZATION OF  ==========================
C  VARIOUS QUANTITIES ON EACH TIMESTEP      ==========================
C=====================================================================
C
C---------------------------------------------------------------------
C  UPDATE TIMESTEP COUNTER AND TOTAL ELAPSED TIME
C---------------------------------------------------------------------
C
      ITT=ITT+1
      TTSEC=TTSEC+DTTS
C
C---------------------------------------------------------------------
C  UPDATE PERMUTING DISC I/O UNITS
C---------------------------------------------------------------------
C
      NDISKB= MOD(ITT+2,3)+1
      NDISK = MOD(ITT+0,3)+1
      NDISKA= MOD(ITT+1,3)+1
```

```
C
C-----------------------------------------------------------------------
C  ADJUST VARIOUS QUANTITIES FOR MIXING TIMESTEP
C-----------------------------------------------------------------------
C
      MIX=0
      MXP=0
      C2DTTS=2.0*DTTS
      C2DTUV=2.0*DTUV
      IF(MOD(ITT,NMIX).EQ.1) THEN
        MIX=1
        C2DTTS=DTTS
        C2DTUV=DTUV
        CALL OPUT(KFLDS,NWDS,  NWDS+1,UBT)
        CALL OPUT(KFLDS,NWDS,2*NWDS+1,VBT)
      ENDIF
 182  CONTINUE
C
C-----------------------------------------------------------------------
C  ESTABLISH OVER DIMENSIONED ARRAYS FOR VECTORIZATION
C-----------------------------------------------------------------------
C
      DO 184 K=1,KM
      DO 184 I=1,IMT
        DXTQ  (I,K)=DXT  (I)
        DXT4RQ(I,K)=DXT4R(I)
        DXUQ  (I,K)=DXU  (I)
        DXU2RQ(I,K)=DXU2R(I)
        DZZQ  (I,K)=DZZ  (K)
        DZ2RQ (I,K)=DZ2R (K)
        DZZ2RQ(I,K)=DZZ2R(K)
        C2DZQ (I,K)=C2DZ (K)
        EEHQ  (I,K)=EEH  (K)
        EEMQ  (I,K)=EEM  (K)
        FFHQ  (I,K)=FFH  (K)
        FFMQ  (I,K)=FFM  (K)
 184  CONTINUE
C
C-----------------------------------------------------------------------
C  LOAD COEFFICIENT ARRAYS FOR SUBSEQUENT CALLS TO "STATE" AND "STATEC"
C-----------------------------------------------------------------------
C
      CALL STINIT
C
C-----------------------------------------------------------------------
C  QUEUE UP DISK READS FOR THIS TIMESTEP
C-----------------------------------------------------------------------
C
      DO 120 J=2,JMTM1
        CALL OFIND(LABS(NDISKB),NSLAB,(J-1)*NSLAB+1)
        CALL OFIND(LABS(NDISK),NSLAB,(J-1)*NSLAB+1)
        CALL OFIND(KFLDS,IMT,  NWDS+(J-1)*IMT+1)
        CALL OFIND(KFLDS,IMT,2*NWDS+(J-1)*IMT+1)
 120  CONTINUE
C
C-----------------------------------------------------------------------
C  INITIALIZE VARIOUS QUANTITIES USED FOR ANALYSIS OF THE SOLUTION
C-----------------------------------------------------------------------
C
      EKTOT=0.0
      DO 130 M=1,NT
        DTABS(M)=0.0
        TVAR(M)=0.0
 130  CONTINUE
```

```
      NERGY=0
      IF(MOD(ITT,NNERGY).EQ.0) NERGY=1
      IF(NERGY.EQ.1 .AND. MXP.EQ.0) THEN
        BUOY=0.0
        DO 186 LL=1,8
          ENGINT(LL)=0.0
          ENGEXT(LL)=0.0
 186    CONTINUE
        DO 190 I=1,IMT
          ZUENG(I)=0.0
          ZVENG(I)=0.0
 190    CONTINUE
        DO 192 M=1,NT
        DO 192 LL=1,6
          TTDTOT(LL,M)=0.0
 192    CONTINUE
        DO 194 J=1,JMT
          DO 193 M=1,NTMIN2
          DO 193 LL=1,8
            TTN(LL,J,M)=0.0
 193      CONTINUE
        DO 194 K=1,KM
          TMT(J,K)=0.0
 194    CONTINUE
      ENDIF
C
C=====================================================================
C  END OF SECTION FOR INITIALIZATION  ==================================
C=====================================================================
C
C=====================================================================
C  BEGIN A BOOTSTRAP PROCEDURE TO PREPARE FOR THE  =====================
C  ROW-BY-ROW COMPUTATION OF PROGNOSTIC VARIABLES  =====================
C=====================================================================
C
C---------------------------------------------------------------------
C  FETCH DATA FOR ROW 2 FROM THE DISC
C---------------------------------------------------------------------
C
      CALL OGET(LABS(NDISKB),NSLAB,NSLAB+1,TBP)
      CALL OGET(LABS(NDISK ),NSLAB,NSLAB+1,TP )
      CALL OGET(KFLDS,IMT,  NWDS+IMT+1,SFUB)
      CALL OGET(KFLDS,IMT,2*NWDS+IMT+1,SFVB)
C
C---------------------------------------------------------------------
C  SWITCH SLAB INCIDENTAL DATA INTO CORRECT SLAB AFTER READIN
C---------------------------------------------------------------------
C
      IF(MOD(ITT,2)+MXP.NE.1) THEN
        DO 220 N=1,NSWICH
          BCON(N,1)=FKMUP(N)
          FKMUP(N)=FKMTP(N)
          FKMTP(N)=BCON(N,1)
 220    CONTINUE
      ENDIF
C
C---------------------------------------------------------------------
C  CONVERT MAXIMUM LEVEL INDICATORS TO INTEGERS
C---------------------------------------------------------------------
C
      DO 222 I=1,IMT
        KMTP(I)=FKMTP(I)
        KMUP(I)=FKMUP(I)
 222  CONTINUE
```

```
C
C-----------------------------------------------------------------------
C  MOVE TAU-1 DATA TO TAU LEVEL ON A MIXING TIMESTEP
C-----------------------------------------------------------------------
C
      IF(MIX.EQ.1) THEN
        DO 224 M=1,NT
        DO 224 K=1,KM
        DO 224 I=1,IMT
          TBP(I,K,M)=TP(I,K,M)
 224    CONTINUE
        DO 226 K=1,KM
        DO 226 I=1,IMT
          UBP(I,K)=UP(I,K)
          VBP(I,K)=VP(I,K)
 226    CONTINUE
      ENDIF
C
C-----------------------------------------------------------------------
C  INITIALIZE ARRAYS FOR FIRST CALLS TO CLINIC AND TRACER
C-----------------------------------------------------------------------
C
      FX=0.0
      DO 240 N=1,NSLAB
        TBSLAB(N)=FX
        T SLAB(N)=FX
 240  CONTINUE
      DO 250 K=1,KM
      DO 250 I=1,IMT
        FVST(I,K)=FX
        RHOS(I,K)=FX
        FMM (I,K)=FX
        FM  (I,K)=FX
 250  CONTINUE
C
C-----------------------------------------------------------------------
C  CONSTRUCT MASK ARRAY FOR ROW 2 TRACERS
C-----------------------------------------------------------------------
C
      DO 254 K=1,KM
      DO 254 I=1,IMT
        IF(KMTP(I).GE.KAR(K)) THEN
          FMP(I,K)=1.0
        ELSE
          FMP(I,K)=0.0
        ENDIF
      FXA=0.0
      FXB=1.0
      WHERE (KMTP(1;IMT).GE.KAR(K))
         FMP(1,K;IMT)=FXB
      OTHERWISE
         FMP(1,K;IMT)=FXA
      END WHERE
 254  CONTINUE
C
C-----------------------------------------------------------------------
C  SAVE INTERNAL MODE VELOCITIES FOR ROW 2
C  AND COMPUTE ADVECTIVE COEFFICIENT FOR SOUTH FACE OF ROW 2 U,V BOXES
C-----------------------------------------------------------------------
C
      FX=DYU2R(2)*CSR(2)*CST(2)*0.5
      DO 260 K=1,KM
      DO 260 I=1,IMT
        UCLIN(I,K)=UP(I,K)
```

```
        VCLIN(I,K)=VP(I,K)
        FVSU(I,K)=(VP(I,K)+V(I,K))*FX
  260  CONTINUE
C
C-----------------------------------------------------------------------
C  COMPUTE EXTERNAL MODE VELOCITIES FOR ROW 2
C-----------------------------------------------------------------------
C
C  1ST, COMPUTE FOR TAU-1 TIME LEVEL
C
      J=1
      DO 270 I=1,IMTM1
        SFUB(I)=SFUB(I)*HR(I,J+1)
        SFVB(I)=SFVB(I)*HR(I,J+1)
  270  CONTINUE
C
C  2ND, COMPUTE FOR TAU TIME LEVEL
C
      DO 280 I=1,IMTM1
        SFU(I)=UBT(I,J+1)*HR(I,J+1)
        SFV(I)=VBT(I,J+1)*HR(I,J+1)
  280  CONTINUE
C
C  3RD, SET CYCLIC BOUNDARY CONDITIONS
C
      SFUB(IMT)=SFUB(2)
      SFVB(IMT)=SFVB(2)
      SFU (IMT)=SFU (2)
      SFV (IMT)=SFV (2)
C
C-----------------------------------------------------------------------
C  ADD EXTERNAL MODE TO INTERNAL MODE FOR ROW 2  (OCEAN PTS. ONLY)
C-----------------------------------------------------------------------
C
      DO 300 K=1,KM
      DO 300 I=1,IMU
        IF(KMUP(I).GE.KAR(K)) THEN
          UBP(I,K)=UBP(I,K)+SFUB(I)
          VBP(I,K)=VBP(I,K)+SFVB(I)
          UP (I,K)=UP (I,K)+SFU (I)
          VP (I,K)=VP (I,K)+SFV (I)
        ENDIF
        WHERE (KMUP(1;IMU).GE.KAR(K))
          UBP(1,K;IMU)=UBP(1,K;IMU)+SFUB(1;IMU)
          VBP(1,K;IMU)=VBP(1,K;IMU)+SFVB(1;IMU)
          UP (1,K;IMU)=UP (1,K;IMU)+SFU (1;IMU)
          VP (1,K;IMU)=VP (1,K;IMU)+SFV (1;IMU)
        ENDWHERE
  300  CONTINUE
C
C-----------------------------------------------------------------------
C  ACCUMULATE KINETIC ENERGY FROM ROW 2 EVERY NTSI TIMESTEPS
C-----------------------------------------------------------------------
C
      IF(MOD(ITT,NTSI).EQ.0) THEN
        DO 305 K=1,KM
          FX=0.5*CS(J+1)*DYU(J+1)*DZ(K)
        DO 305 I=2,IMUM1
          EKTOT=EKTOT+(UP(I,K)*UP(I,K)+VP(I,K)*VP(I,K))*FX*DXU(I)
  305    CONTINUE
      ENDIF
C
C-----------------------------------------------------------------------
C  COMPUTE DENSITY OF ROW 2
```

```
C-----------------------------------------------------------------------
C
      CALL STATE(TP(1,1,1),TP(1,1,2),RHOS,TDIF(1,1,1),TDIF(1,1,2))
C
C  SET CYCLIC BOUNDARY CONDITIONS
C
      DO 310 K=1,KM
        RHOS(IMT,K)=RHOS(2,K)
 310  CONTINUE
C
C======================================================================
C  END OF BOOTSTRAP PROCEDURE  ========================================
C======================================================================
C
C======================================================================
C  BEGIN ROW-BY-ROW COMPUTATION OF PROGNOSTIC VARIABLES  ==============
C======================================================================
C
      DO 380 J=2,JMTM1
C
C-----------------------------------------------------------------------
C  MOVE ALL SLAB DATA DOWN ONE ROW
C-----------------------------------------------------------------------
C
      DO 320 N=1,NSLAB
        TBM(N,1,1)=TB (N,1,1)
        TM (N,1,1)=T  (N,1,1)
        TB (N,1,1)=TBP(N,1,1)
        T  (N,1,1)=TP (N,1,1)
 320  CONTINUE
C
C-----------------------------------------------------------------------
C  COMPLETE READIN OF J+1 SLAB (EXCEPT LAST ROW)
C-----------------------------------------------------------------------
C
      IF(J.NE.JMTM1) THEN
        CALL OGET(LABS(NDISKB),NSLAB,J*NSLAB+1,TBP)
        CALL OGET(LABS(NDISK ),NSLAB,J*NSLAB+1,TP )
        CALL OGET(KFLDS,IMT,  NWDS+J*IMT+1,SFUB)
        CALL OGET(KFLDS,IMT,2*NWDS+J*IMT+1,SFVB)
      ENDIF
C
C-----------------------------------------------------------------------
C  INITIATE WRITEOUT OF NEWLY COMPUTED DATA FROM PREVIOUS ROW
C-----------------------------------------------------------------------
C
      IF(J.GT.2) CALL OPUT(LABS(NDISKA),NSLAB,(J-2)*NSLAB+1,TA)
C
C-----------------------------------------------------------------------
C  SWITCH SLAB INCIDENTAL DATA INTO CORRECT SLAB AFTER READIN
C-----------------------------------------------------------------------
C
      IF(MOD(ITT,2)+MXP.NE.1) THEN
        DO 332 N=1,NSWICH
          BCON(N,1)=FKMUP(N)
          FKMUP(N)=FKMTP(N)
          FKMTP(N)=BCON(N,1)
 332    CONTINUE
      ENDIF
C
C-----------------------------------------------------------------------
C  SHIFT MAXIMUM LEVEL INDICATORS DOWN ONE ROW AND SET J+1 VALUES
C-----------------------------------------------------------------------
C
```

```
      DO 334 I=1,IMT
        KMT (I)=KMTP (I)
        KMU (I)=KMUP (I)
        KMTP(I)=FKMTP(I)
        KMUP(I)=FKMUP(I)
 334  CONTINUE
C
C-----------------------------------------------------------------------
C  SET SYMMETRY BOUNDARY CONDITIONS ON LAST ROW
C-----------------------------------------------------------------------
C
      IF(J.EQ.JMTM1) THEN
        DO 335 I=1,IMT
          KMTP(I)=FKMT (I)
          KMUP(I)=FKMUM(I)
 335    CONTINUE
        DO 336 M=1,NT
        DO 336 K=1,KM
        DO 336 I=1,IMT
          TBP(I,K,M)=TB(I,K,M)
          T P(I,K,M)=T (I,K,M)
 336    CONTINUE
      ENDIF
C
C-----------------------------------------------------------------------
C  MOVE TAU-1 DATA TO TAU LEVEL ON A MIXING TIMESTEP
C-----------------------------------------------------------------------
C
      IF(MIX.EQ.1) THEN
        DO 337 M=1,NT
        DO 337 K=1,KM
        DO 337 I=1,IMT
          TBP(I,K,M)=TP(I,K,M)
 337    CONTINUE
        DO 338 K=1,KM
        DO 338 I=1,IMT
          UBP(I,K)=UP(I,K)
          VBP(I,K)=VP(I,K)
 338    CONTINUE
      ENDIF
C
C-----------------------------------------------------------------------
C  SHIFT MASKS DOWN ONE ROW AND COMPUTE NEW MASKS
C-----------------------------------------------------------------------
C
      DO 345 K=1,KM
      DO 345 I=1,IMT
        FMM(I,K)=FM (I,K)
        FM (I,K)=FMP(I,K)
 345  CONTINUE
      DO 354 K=1,KM
      DO 354 I=1,IMT
        IF(KMTP(I).GE.KAR(K)) THEN
           FMP(I,K)=1.0
        ELSE
           FMP(I,K)=0.0
        ENDIF
        IF(KMU(I).GE.KAR(K)) THEN
           GM(I,K)=1.0
        ELSE
           GM(I,K)=0.0
        ENDIF
        FXA=0.0
        FXB=1.0
```

```
        WHERE (KMTP(1;IMT).GE.KAR(K))
           FMP(1,K;IMT)=FXB
        OTHERWISE
           FMP(1,K;IMT)=FXA
        ENDWHERE
        WHERE (KMU(1;IMT).GE.KAR(K))
           GM(1,K;IMT)=FXB
        OTHERWISE
           GM(1,K;IMT)=FXA
        ENDWHERE
 354  CONTINUE
C
      CALL GETIME(T0,T2)
C
C-----------------------------------------------------------------------
C  CALL THE MAIN COMPUTATIONAL ROUTINES TO UPDATE THE ROW
C-----------------------------------------------------------------------
C
      IF(J.NE.JMTM1) CALL CLINIC(J)
C
      CALL CLINIC(J)
C
      CALL GETIME(T1,T2)
      TIME(3)=TIME(3)+T1-T0
C
      CALL TRACER(J)
C
      CALL GETIME(T0,T2)
      TIME(4)=TIME(4)+T0-T1
C
C-----------------------------------------------------------------------
C  PRINT THE PROGRESSING SOLUTION AT SPECIFIED ROWS ON ENERGY TSTEP
C-----------------------------------------------------------------------
C
      MTEST=MOD(J,??)
      IF(NERGY.EQ.0.OR.MXP.EQ.1) GO TO 339
      IF(MTEST.NE.0) GO TO 8090
      IPRT=??
C
C  DETERMINE INDEX OF FIRST T OCEAN POINT
C
      DO 430 I=1,IMT
        ISTRT=I
        IF(KMT(I).NE.0) GO TO 431
 430  CONTINUE
 431  CONTINUE
      ISTOP=ISTRT+IPRT-1
      IF(ISTOP.GT.IMT) ISTOP=IMT
      DO 8015 M=1,NT
        IF(M.EQ.1) PRINT 8001,J,ITT
        IF(M.EQ.2) PRINT 8002,J,ITT
        IF(M.EQ.3) PRINT 8003,J,ITT
        IF(M.EQ.4) PRINT 8004,J,ITT
 8001   FORMAT(20H TEMPERATURE FOR J =,I4,12H AT TIMESTEP,I7)
 8002   FORMAT(20H SALINITY    FOR J =,I4,12H AT TIMESTEP,I7)
 8003   FORMAT(20H TRACER 1    FOR J =,I4,12H AT TIMESTEP,I7)
 8004   FORMAT(20H TRACER 2    FOR J =,I4,12H AT TIMESTEP,I7)
        SCL=1.0
        IF(M.EQ.2) SCL=1.E-3
        CALL MATRIX(T(1,1,M),IMT,ISTRT,ISTOP,0,KM,SCL)
 8015 CONTINUE
      PRINT 8011,J,ITT
 8011 FORMAT(20H W VELOCITY  FOR J =,I4,12H AT TIMESTEP,I7)
C
```

```
C  SET CYCLIC BOUNDARY CONDITION ON W BEFORE PRINTING
C
      DO 433 K=1,KMP1
        W(1  ,K)=W(IMTM1,K)
        W(IMT,K)=W(2    ,K)
 433  CONTINUE
      SCL=1.E-3
      CALL MATRIX(W,IMT,ISTRT,ISTOP,0,KMP1,SCL)
C
C  DETERMINE INDEX OF FIRST U,V OCEAN POINT
C
      DO 440 I=1,IMTM1
        ISTRT=I
        IF(KMU(I+1).NE.0) GO TO 441
 440  CONTINUE
 441  CONTINUE
      ISTOP=ISTRT+IPRT-1
      IF(ISTOP.GT.IMT) ISTOP=IMT
      PRINT 8021, J,ITT
 8021 FORMAT(20H U VELOCITY  FOR J =,I4,12H AT TIMESTEP,I7)
      SCL=1.0
      CALL MATRIX(U,IMT,ISTRT,ISTOP,0,KM,SCL)
      PRINT 8022, J,ITT
 8022 FORMAT(20H V VELOCITY  FOR J =,I4,12H AT TIMESTEP,I7)
      CALL MATRIX(V,IMT,ISTRT,ISTOP,0,KM,SCL)
C
C-----------------------------------------------------------------------
C  COMPUTE THE NORTHWARD TRANSPORT OF EACH TRACER QUANTITY
C  AS WELL AS THE ZONALLY INTEGRATED MERIDIONAL MASS TRANSPORT
C-----------------------------------------------------------------------
C
 8090 IF(J.EQ.JMTM1) GO TO 8190
      DO 8092 K=1,KM
        VBR(K)=0.0
      DO 8092 M=1,NT
        TBRS(K,M)=TBRN(K,M)
        TBRN(K,M)=0.0
 8092 CONTINUE
      IF(J.GT.2) GO TO 8110
      DO 8094 M=1,NT
      DO 8094 K=1,KM
        TBRS(K,M)=0.0
 8094 CONTINUE
      DO 8102 K=1,KM
        TOTDX=0.0
        DO 8100 I=2,IMTM1
          TOTDX=TOTDX+DXT(I)*(FM(I,K))
        DO 8100 M=1,NT
          TBRS(K,M)=TBRS(K,M)+T(I,K,M)*FM(I,K)*DXT(I)
 8100   CONTINUE
        IF(TOTDX.NE.0.0) THEN
          DO 8101 M=1,NT
            TBRS(K,M)=TBRS(K,M)/TOTDX
 8101     CONTINUE
        ENDIF
 8102 CONTINUE
 8110 CCTJ=AH*DYUR(J)
      DO 8130 K=1,KM
        TOTDX=0.0
        DO 8120 I=2,IMTM1
          TOTDX=TOTDX+DXT(I)*(FMP(I,K))
          VBR(K)=VBR(K)+V(I,K)*DXU(I)*CS(J)
        DO 8120 M=1,NT
          TBRN(K,M)=TBRN(K,M)+TP(I,K,M)*FMP(I,K)*DXT(I)
```

```
 8120   CONTINUE
        IF(TOTDX.NE.0.0) THEN
          DO 8122 M=1,NT
            TBRN(K,M)=TBRN(K,M)/TOTDX
 8122     CONTINUE
        ENDIF
        IF(K.EQ.1) TMT(J,1)=VBR(1)*DZ(1)
        IF(K.GT.1) TMT(J,K)=TMT(J,K-1)+VBR(K)*DZ(K)
      DO 8130 M=1,NT
        TTN(1,J,M)=TTN(1,J,M)+VBR(K)*(TBRN(K,M)+TBRS(K,M))*0.5*DZ(K)
      DO 8130 I=2,IMTM1
        TTN(6,J,M)=TTN(6,J,M)+(V(I,K)*DXU(I)+V(I-1,K)*DXU(I-1))*
     *             (T(I,K,M)+TP(I,K,M))*CS(J)*0.25*DZ(K)
        TTN(7,J,M)=TTN(7,J,M)-CCTJ*FM(I,K)*FMP(I,K)*
     *             (TP(I,K,M)-T(I,K,M))*DXT(I)*CS(J)*DZ(K)
 8130 CONTINUE
      DO 8140 M=1,NT
      DO 8140 I=2,IMTM1
        TOTDZ=0.0
        VBRZ=0.0
        TBRZ=0.0
        IKM=I
        IF(KMU(I-1).GT.KMU(I)) IKM=I-1
        KZ=KMU(IKM)
        IF(KZ.EQ.0) GO TO 8140
        DO 8136 K=1,KZ
          VBRZ=VBRZ+(V(I,K)*DXU(I)+V(I-1,K)*DXU(I-1))*DZ(K)
          TBRZ=TBRZ+(T(I,K,M)+TP(I,K,M))*DZ(K)
          TOTDZ=TOTDZ+DZ(K)
 8136   CONTINUE
        TBRZ=TBRZ/TOTDZ
        TTN(3,J,M)=TTN(3,J,M)+VBRZ*TBRZ*CS(J)*0.25
        TTN(5,J,M)=TTN(5,J,M)-(WSX(I)*DXU(I)+WSX(I-1)*DXU(I-1))*
     *             (T(I,1,M)+TP(I,1,M)-TBRZ)*CS(J)/(8.0*OMEGA*SINE(J))
 8140 CONTINUE
      DO 8150 M=1,NT
        TTN(2,J,M)=TTN(6,J,M)-TTN(1,J,M)
        TTN(4,J,M)=TTN(6,J,M)-TTN(3,J,M)-TTN(5,J,M)
        TTN(8,J,M)=TTN(6,J,M)+TTN(7,J,M)
 8150 CONTINUE
 8190 CONTINUE
 339  CONTINUE
C
C-----------------------------------------------------------------------
C  PUT SLAB INCIDENTAL DATA INTO CORRECT SLAB FOR WRITEOUT
C-----------------------------------------------------------------------
C
      IF(MOD(ITT,2).EQ.0) THEN
        DO 340 N=1,NSWICH
          BCON(N,1)=FKMT(N)
 340    CONTINUE
      ELSE
        DO 360 N=1,NSWICH
          BCON(N,1)=FKMU(N)
 360    CONTINUE
      ENDIF
 380  CONTINUE
C
C=======================================================================
C  END ROW-BY-ROW COMPUTATION  ========================================
C=======================================================================
C
C-----------------------------------------------------------------------
C  INITIATE WRITEOUT OF NEWLY COMPUTED DATA FROM THE FINAL ROW
```

```
C-----------------------------------------------------------------------
C
      CALL OPUT(LABS(NDISKA),NSLAB,(JMT-2)*NSLAB+1,TA)
C
C-----------------------------------------------------------------------
C  SOLVE FOR NEW EXTERNAL MODE
C-----------------------------------------------------------------------
C
      IF ( MXP.EQ.0 ) CALL TROPIC
C
C---------------------------------------------------------------------
C  PRINT ONE LINE OF TIMESTEP INFORMATION ON SPECIFIED TIMESTEPS
C---------------------------------------------------------------------
C
      IF(EB.AND.MIX.EQ.1) GO TO 390
      IF(MOD(ITT,NTSI).EQ.0) THEN
        EKTOT=EKTOT/VOLUME
        DO 381 M=1,NT
          DTABS(M)=DTABS(M)/VOLUME
 381    CONTINUE
        DAYSYR=365.25
        TTYEAR=TTSEC/(3600.*24.*DAYSYR)
        TTDAY=TTSEC/(3600.*24.)
        TTDAY=MOD(TTDAY,DAYSYR)
      PRINT 910,ITT,TTYEAR,TTDAY,EKTOT,DTABS(1),DTABS(2)
 910    FORMAT(4H TS=,I6,7H  YEAR=,F7.2,6H  DAY=,F5.1,9H  ENERGY=,
    *     1PE13.6,8H  DTEMP=,1PE13.6,8H  DSALT=,1PE13.6)
      ENDIF
C
C---------------------------------------------------------------------
C  COMPLETE AND PRINT THE ON-LINE INTEGRALS ON ENERGY TIMESTEPS
C---------------------------------------------------------------------
C
      IF(NERGY.EQ.0) GO TO 390
C
C  1ST, NORMALIZE PREVIOUSLY COMPUTED INTEGRALS BY VOLUME
C
      DO 382 LL=1,8
        ENGINT(LL)=ENGINT(LL)/VOLUME
        ENGEXT(LL)=ENGEXT(LL)/VOLUME
 382  CONTINUE
      DO 383 M=1,NT
        TVAR(M)=TVAR(M)/VOLUME
      DO 383 LL=1,6
        TTDTOT(LL,M)=TTDTOT(LL,M)/VOLUME
 383  CONTINUE
      BUOY=BUOY/VOLUME
C
C  2ND, COMPUTE RESIDUAL TERMS
C
      PLICIN=ENGINT(1)-ENGINT(2)-ENGINT(3)-ENGINT(4)
    *       -ENGINT(5)-ENGINT(6)-ENGINT(7)-ENGINT(8)
      PLICEX=ENGEXT(1)-ENGEXT(2)-ENGEXT(3)-ENGEXT(4)
    *       -ENGEXT(5)-ENGEXT(6)-ENGEXT(7)-ENGEXT(8)
      DO 384 M=1,NT
        TTDTOT(6,M)=TTDTOT(1,M)-TTDTOT(2,M)-TTDTOT(3,M)
    *              -TTDTOT(4,M)-TTDTOT(5,M)
 384  CONTINUE
C
C  3RD, PRINT THE INTEGRALS
C
      PRINT 9100
      PRINT 9101,ENGINT(1),ENGEXT(1),TTDTOT(1,1),TTDTOT(1,2)
      PRINT 9102,ENGINT(2),ENGEXT(2),TTDTOT(2,1),TTDTOT(2,2)
```

```
      PRINT 9103,ENGINT(3),ENGEXT(3),TTDTOT(3,1),TTDTOT(3,2)
      PRINT 9104,ENGINT(4),ENGEXT(4),TTDTOT(4,1),TTDTOT(4,2)
      PRINT 9105,ENGINT(5),ENGEXT(5),TTDTOT(5,1),TTDTOT(5,2)
      PRINT 9106,ENGINT(6),ENGEXT(6),TTDTOT(6,1),TTDTOT(6,2)
      PRINT 9109,PLICIN,PLICEX,TVAR(1),TVAR(2)
      PRINT 9107,ENGINT(7),ENGEXT(7)
      PRINT 9108,ENGINT(8),ENGEXT(8)
 9100 FORMAT( 1X,50HWORK BY:              INTERNAL MODE  EXTERNAL MODE,
     *       10X,50H                       TEMPERATURE     SALINITY    )
 9101 FORMAT( 1X,20HTIME RATE OF CHANGE ,2(1PE15.6),
     *       10X,20HTIME RATE OF CHANGE ,2(1PE15.6))
 9102 FORMAT( 1X,20HHORIZONTAL ADVECTION,2(1PE15.6),
     *       10X,20HHORIZONTAL ADVECTION,2(1PE15.6))
 9103 FORMAT( 1X,20HVERTICAL ADVECTION  ,2(1PE15.6),
     *       10X,20HVERTICAL ADVECTION  ,2(1PE15.6))
 9104 FORMAT( 1X,20HHORIZONTAL FRICTION ,2(1PE15.6),
     *       10X,20HHORIZONTAL DIFFUSION,2(1PE15.6))
 9105 FORMAT( 1X,20HVERTICAL FRICTION   ,2(1PE15.6),
     *       10X,20HSURFACE FLUX        ,2(1PE15.6))
 9106 FORMAT( 1X,20HPRESSURE FORCES     ,2(1PE15.6),
     *       10X,20HTRUNCATION ERROR    ,2(1PE15.6))
 9107 FORMAT( 1X,20HWORK BY WIND        ,2(1PE15.6))
 9108 FORMAT( 1X,20HBOTTOM DRAG         ,2(1PE15.6))
 9109 FORMAT( 1X,20HIMPLICIT EFFECTS    ,2(1PE15.6),
     *       10X,20HCHANGE OF VARIANCE  ,2(1PE15.6))
      TVAR(1)=BUOY-ENGINT(6)-ENGEXT(6)
      DTABS(1)=ENGINT(2)+ENGINT(3)+ENGEXT(2)+ENGEXT(3)
      PRINT 9110,BUOY,TVAR(1),DTABS(1)
 9110 FORMAT(1X,25HWORK BY BUOYANCY FORCES  ,1PE15.6,5X,25HENERGY CONVER
     *SION ERROR  ,1PE15.6,5X,25HNONLINEAR EXCHANGE ERROR ,1PE15.6)
C
C-----------------------------------------------------------------------
C  PRINT THE NORTHWARD TRANSPORT OF HEAT AND SALT
C-----------------------------------------------------------------------
C
      PRINT 8195
 8195 FORMAT(/,' NORTHWARD TRANSPORT OF HEAT (X10**15 WATTS)',24X,'NORTH
     *WARD TRANSPORT OF SALT (X10**10 CM**3/SEC)',/,6X,'X MEAN  X EDDY
     *Z MEAN  Z EDDY   EKMAN TOT ADV  DIFFUS   TOTAL   X MEAN  X EDDY  Z
     * MEAN  Z EDDY   EKMAN TOT ADV  DIFFUS   TOTAL')
C
C  CONVERT HEAT TRANSPORT TO PEDAWATTS, SALT TRNSPT TO 10**10 CM**3/SEC
C
      DO 8198 J=1,JMT
      DO 8198 LL=1,8
        TTN(LL,J,1)=TTN(LL,J,1)*4.186E-15
        TTN(LL,J,2)=TTN(LL,J,2)*1.E-10
 8198 CONTINUE
      DO 8197 JJ=2,JMTM2
        J=JMT-JJ
        PRINT 8196,J,(TTN(I,J,1),I=1,8),(TTN(I,J,2),I=1,8)
 8196   FORMAT(I4,8F8.3,1X,8F8.3)
 8197 CONTINUE
      PRINT 8194
 8194 FORMAT(/,' MERIDIONAL MASS TRANSPORT')
      SCL=1.E12
      CALL MATRIX(TMT,JMT,1,JMT,0,KM,SCL)
 390  CONTINUE
C
C-----------------------------------------------------------------------
C  IF THIS IS THE END OF THE 1ST PASS OF AN EULER BACKWARD TIMESTEP,
C  SET THE INPUT DISC UNITS SO THAT THE PROPER LEVELS ARE FETCHED ON
C  THE NEXT PASS.  THE OUTPUT FOR THE 2ND PASS WILL BE PLACED ON THE
C  UNUSED UNIT ("NDISKB") AND TRANSFERRED TO THE PROPER UNIT ("NDISKA")
```

```
C  LATER.  RETURN TO THE TOP OF "STEP" TO DO THE 2ND PASS.
C-----------------------------------------------------------------
C
      IF(MIX.EQ.1 .AND. EB) THEN
        MIX=0
        MXP=1
        NDISKX=NDISKB
        NDISKB=NDISK
        NDISK=NDISKA
        NDISKA=NDISKX
        GO TO 182
      ENDIF
C
C-----------------------------------------------------------------
C  IF THIS IS THE END OF THE 2ND PASS OF AN EULER BACKWARD TIMESTEP,
C  TRANSFER THE DATA WRITTEN TEMPORARILY TO "NDISKX" TO ITS FINAL
C  DESTINATION (THE ORIGINAL "NDISKA").
C-----------------------------------------------------------------
C
      IF(MXP.EQ.1) THEN
        NDISKA=NDISK
        NDISK=NDISKB
        DO 394 J=2,JMTM1
          CALL OGET(LABS(NDISKX),NSLAB,(J-1)*NSLAB+1,TA)
          CALL OPUT(LABS(NDISKA),NSLAB,(J-1)*NSLAB+1,TA)
 394    CONTINUE
      ENDIF
C
C-----------------------------------------------------------------
C  FOR PURPOSES OF RECOVERING FROM THE DISC AFTER AN ABNORMAL STOP,
C  NORMALLY INACTIVE DISC UNITS ARE BROUGHT UP TO DATE HERE.
C-----------------------------------------------------------------
C
      IF ( MXP.EQ.0 ) THEN
        CALL OPUT(KFLDS,NWDS,3*NWDS+1,ETA)
        CALL OPUT(KFLDS,NWDS,4*NWDS+1,UBT)
        CALL OPUT(KFLDS,NWDS,5*NWDS+1,VBT)
      ENDIF
      CALL OPUT(KONTRL,20,1,ITT)
C
      RETURN
      END
```

```
*DECK CLINIC
      SUBROUTINE CLINIC(J)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C=======================================================================
C                                                                   ====
C  CLINIC COMPUTES, FOR ONE ROW, THE INTERNAL MODE COMPONENT OF     ====
C         THE U AND V VELOCITIES, AS WELL AS THE VORTICITY DRIVING  ====
C         FUNCTION FOR USE BY "RELAX" LATER IN DETERMINING THE      ====
C         EXTERNAL MODES, WHERE:                                    ====
C           J=THE ROW NUMBER                                        ====
C                                                                   ====
C=======================================================================
C
C-----------------------------------------------------------------------
C  DEFINE GLOBAL DATA
C-----------------------------------------------------------------------
C
*CALL PARAM
*CALL FULLWD
*CALL SCALAR
*CALL ONEDIM
*CALL FIELDS
*CALL WRKSPA
C
C-----------------------------------------------------------------------
C  DEFINE AND EQUIVALENCE LOCAL DATA
C-----------------------------------------------------------------------
C
      DIMENSION DPDX(IMT,KM),DPDY(IMT,KM),UENG(IMT,KM),VENG(IMT,KM)
     *          ,FUWBT(IMT),FVNBT(IMT),FVSBT(IMT)                       *
     *          ,QU(IMT),QV(IMT)                                        *
      EQUIVALENCE (TDIF(1,1,1),DPDX(1,1),UENG(1,1)),
     *            (TDIF(1,1,2),DPDY(1,1),VENG(1,1))
C
C-----------------------------------------------------------------------
C  BEGIN EXECUTABLE CODE
C-----------------------------------------------------------------------
C
C=======================================================================
C  BEGIN INTRODUCTORY SECTION, PREPARING VARIOUS    ====================
C  ARRAYS FOR THE COMPUTATION OF THE INTERNAL MODES ====================
C=======================================================================
C
C
C-----------------------------------------------------------------------
C  SAVE INTERNAL MODE VELOCITIES
C-----------------------------------------------------------------------
C
      DO 140 K=1,KM
      DO 140 I=1,IMT
        USAV(I,K)=UCLIN(I,K)
        VSAV(I,K)=VCLIN(I,K)
        UCLIN(I,K)=UP(I,K)
        VCLIN(I,K)=VP(I,K)
 140  CONTINUE
C
C  IF LAST ROW, NO NEED TO PERFORM OPERATIONS ON J+1 ROW
C
      IF(J.EQ.JMTM1) GO TO 176
C
C-----------------------------------------------------------------------
C  COMPUTE EXTERNAL MODE VELOCITIES FOR ROW J+1
```

```
C-----------------------------------------------------------------
C
C  1ST, COMPUTE FOR TAU-1 TIME LEVEL
C
      DO 150 I=1,IMTM1
        SFUB(I)=SFUB(I)*HR(I,J+1)
        SFVB(I)=SFVB(I)*HR(I,J+1)
 150  CONTINUE
C
C  2ND, COMPUTE FOR TAU TIME LEVEL
C
      DO 155 I=1,IMTM1
        SFU(I)=UBT(I,J+1)*HR(I,J+1)
        SFV(I)=VBT(I,J+1)*HR(I,J+1)
 155  CONTINUE
C
C  3RD, SET CYCLIC BOUNDARY CONDITIONS
C
      SFUB(IMT)=SFUB(2)
      SFVB(IMT)=SFVB(2)
      SFU (IMT)=SFU (2)
      SFV (IMT)=SFV (2)
C
C-----------------------------------------------------------------
C  ADD EXTERNAL MODE TO INTERNAL MODE FOR ROW J+1  (OCEAN PTS. ONLY)
C-----------------------------------------------------------------
C
      DO 170 K=1,KM
      DO 170 I=1,IMU
        IF(KMUP(I).GE.KAR(K)) THEN
          UBP(I,K)=UBP(I,K)+SFUB(I)
          VBP(I,K)=VBP(I,K)+SFVB(I)
          UP (I,K)=UP (I,K)+SFU (I)
          VP (I,K)=VP (I,K)+SFV (I)
        ENDIF
        WHERE (KMUP(1;IMU).GE.KAR(K))
          UBP(1,K;IMU)=UBP(1,K;IMU)+SFUB(1;IMU)
          VBP(1,K;IMU)=VBP(1,K;IMU)+SFVB(1;IMU)
          UP (1,K;IMU)=UP (1,K;IMU)+SFU (1;IMU)
          VP (1,K;IMU)=VP (1,K;IMU)+SFV (1;IMU)
        ENDWHERE
 170  CONTINUE
C
C-----------------------------------------------------------------
C  ACCUMULATE KINETIC ENERGY FROM ROW J+1 EVERY NTSI TIMESTEPS
C-----------------------------------------------------------------
C
      IF(MOD(ITT,NTSI).EQ.0) THEN
        FX=0.25*CS(J+1)*DYU(J+1)
C
C  WEIGHT SYMMETRY ROW BY ONE HALF
C
        IF(J.EQ.JMTM2) FX=FX*0.5
        DO 173 K=1,KM
        DO 173 I=1,IMT
          UENG(I,K)=(FX*(UP(I,K)*UP(I,K)+VP(I,K)*VP(I,K)))
     *              *C2DZQ(I,K)*DXUQ(I,K)
 173    CONTINUE
        DO 175 K=1,KM
        DO 175 I=2,IMUM1
          EKTOT=EKTOT+UENG(I,K)
C*****************************************************************
C       DO 175 I=2,IMUM1
C         EKTOT=EKTOT+UENG(I,K)
```

```
C  ***** ABOVE 2 STMNTS REPLACED BY FOLLOWING FOR VECTORIZATION ******   316000> 316000000 QC
          EKTOT=EKTOT+Q8SSUM(UENG(2,K;IMUM2))                             316100> 316100000 Q
C**********************************************************************  316200> 316200000 QC
 175    CONTINUE                                                         316300> 316300000
      ENDIF                                                              316400> 316400000
 176  CONTINUE                                                           316500> 316500000
C                                                                       *<510000 316510000 C
C-----------------------------------------------------------------------*<520000 316520000 C
C  FIND ADVECTIVE COEFFICIENTS 'FUW' FOR WEST FACE  OF U,V BOX          *<530000 316530000 C
C                            & 'FVN' FOR NORTH FACE OF U,V BOX          *<540000 316540000 C
C-----------------------------------------------------------------------*<550000 316550000 C
C                                                                       *<560000 316560000 C
      FX=DYU2R(J)*CSR(J)*CST(J+1)                                       *<570000 316570000
      DO 115 K=1,KM                                                     *<580000 316580000
      DO 115 I=1,IMT                                                    *<590000 316590000
        FUW(I,K)=(U(I,K)+U(I-1,K))*CSR(J)*0.5                           *316600> 316600000 -M
        FUW(I,K)=( ( ( U(I,K)*DXU(I)+U(I-1,K)*DXU(I-1) )*DYU(J)         *<610000 316610000 M
     *     +( UP(I,K)*DXU(I)+UP(I-1,K)*DXU(I-1) )*DYU(J+1) )*DYT2R(J+1) *<620000 316620000 M
     *   +( ( U (I,K)*DXU(I)+U (I-1,K)*DXU(I-1) )*DYU(J)                *<630000 316630000 M
     *     +( UM(I,K)*DXU(I)+UM(I-1,K)*DXU(I-1) )*DYU(J-1) )*DYT2R(J) ) *<640000 316640000 M
     *     * DXT4R(I) * CSR(J)                                          *<650000 316650000 M
        FVN(I,K)=(VP(I,K)+V(I,K))*FX*0.5                                *<660000 316660000 -M
        FVN(I,K)= DYT4R(J+1) * FX *                                     *<670000 316670000 M
     *  ( ( ( V (I,K)*DXU(I)+V (I-1,K)*DXU(I-1) )*DYU(J)                *<680000 316680000 M
     *     +( VP(I,K)*DXU(I)+VP(I-1,K)*DXU(I-1) )*DYU(J+1) )*DXT2R(I)   *<690000 316690000 M
     *   +( ( V (I,K)*DXU(I)+V (I+1,K)*DXU(I+1) )*DYU(J)                *316700> 316700000 M
     *     +( VP(I,K)*DXU(I)+VP(I+1,K)*DXU(I+1) )*DYU(J+1) )*DXT2R(I+1))*<710000 316710000 M
 115  CONTINUE                                                          *<720000 316720000
C                                                                       *<730000 316730000 C
C-----------------------------------------------------------------------*<740000 316740000 SC
C  SET SYMMETRY CONDITIONS ON THE LAST ROW                               316800> 316800000 SC
C----------------------------------------------------------------------  316900> 316900000 SC
C                                                                        317000> 317000000 SC
      IF(J.EQ.JMTM1) THEN                                                317100> 317100000 S
        DO 178 K=1,KM                                                    317200> 317200000 S
        DO 178 I=1,IMT                                                   317300> 317300000 S
          FVN(I,K)=-FVSU(I,K)                                            317400> 317400000 S
          UBP(I,K)= UBM (I,K)                                            317500> 317500000 S
          UP (I,K)= UM  (I,K)                                            317600> 317600000 S
 178    CONTINUE                                                         317700> 317700000 S
C                                                                        317800> 317800000 SC
C  ON 1ST PASS OF MIXING TSTEP, REPLACE TAU-1 U VEL. WITH TAU U VEL.     317900> 317900000 SC
C                                                                        318000> 318000000 SC
        IF(MIX.NE.0) THEN                                                318100> 318100000 S
          DO 179 K=1,KM                                                  318200> 318200000 S
          DO 179 I=1,IMT                                                 318300> 318300000 S
            UBP(I,K)=UP(I,K)                                             318400> 318400000 S
 179      CONTINUE                                                       318500> 318500000 S
        ENDIF                                                            318600> 318600000 S
      ENDIF                                                              318700> 318700000 S
C                                                                        318800> 318800000 C
C----------------------------------------------------------------------  318900> 318900000 C
C  COMPUTE DENSITY OF ROW J+1                                            319000> 319000000 C
C----------------------------------------------------------------------  319100> 319100000 C
C                                                                        319200> 319200000 C
      CALL STATE(TP(1,1,1),TP(1,1,2),RHON,TDIF(1,1,1),TDIF(1,1,2))       319300> 319300000
C                                                                        319400> 319400000 OC
C  SET CYCLIC BOUNDARY CONDITIONS                                        319500> 319500000 OC
C                                                                        319600> 319600000 OC
      DO 232 K=1,KM                                                      319700> 319700000 O
        RHON(IMT,K)=RHON(2,K)                                            319800> 319800000 O
 232  CONTINUE                                                           319900> 319900000 O
C                                                                        320000> 320000000 C
C----------------------------------------------------------------------  320100> 320100000 C
```

```
C  COMPUTE VERTICAL VELOCITY IN U,V COLUMNS                              320200> 320200000 C
C------------------------------------------------------------------      320300> 320300000 C
C                                                                        *<310000 320310000 MC
C CALCULATE FUWBT,FVNBT AND FVSBT READY FOR COMPUTATION OF W (SURFACE)   *<320000 320320000 MC
C                                                                        *<330000 320330000 MC
      FX=DYUR(J)*CSR(J)*CST(J+1)                                         *<340000 320340000 M
      FXA=DYUR(J)*CSR(J)*CST(J)                                          *<350000 320350000 M
      DO 220 I=1,IMT                                                     *<360000 320360000 M
       FUWBT(I)=( ( ( UBT(I,J)*DXU(I)+UBT(I-1,J)*DXU(I-1) )*DYU(J)       *<370000 320370000 M
     * +(UBT(I,J+1)*DXU(I)+UBT(I-1,J+1)*DXU(I-1))*DYU(J+1) )*DYT2R(J+1) *<380000 320380000 M
     * +( ( UBT(I,J)*DXU(I)+UBT(I-1,J)*DXU(I-1) )*DYU(J)                 *<390000 320390000 M
     * +(UBT(I,J-1)*DXU(I)+UBT(I-1,J-1)*DXU(I-1) )*DYU(J-1) )*DYT2R(J) )*320400> 320400000 M
     *      * DXT4R(I) * CSR(J)                                          *<410000 320410000 M
       FVNBT(I)= DYT4R(J+1) *  FX *                                      *<420000 320420000 M
     *  ( ( ( VBT(I,J)*DXU(I)+VBT(I-1,J)*DXU(I-1) )*DYU(J)               *<430000 320430000 M
     * +(VBT(I,J+1)*DXU(I)+VBT(I-1,J+1)*DXU(I-1))*DYU(J+1) )*DXT2R(I)    *<440000 320440000 M
     *   +( ( VBT(I,J)*DXU(I)+VBT(I+1,J)*DXU(I+1) )*DYU(J)               *<450000 320450000 M
     * +(VBT(I,J+1)*DXU(I)+VBT(I+1,J+1)*DXU(I+1))*DYU(J+1) )*DXT2R(I+1))*<460000 320460000 M
       FVSBT(I)= DYT4R(J-1) *  FXA *                                     *<470000 320470000 M
     *  ( ( ( VBT(I,J)*DXU(I)+VBT(I-1,J)*DXU(I-1) )*DYU(J)               *<480000 320480000 M
     * +(VBT(I,J-1)*DXU(I)+VBT(I-1,J-1)*DXU(I-1))*DYU(J-1) )*DXT2R(I)    *<490000 320490000 M
     *   +( ( VBT(I,J)*DXU(I)+VBT(I+1,J)*DXU(I+1) )*DYU(J)               *320500> 320500000 M
     * +(VBT(I,J-1)*DXU(I)+VBT(I+1,J-1)*DXU(I+1))*DYU(J-1) )*DXT2R(I+1))*<510000 320510000 M
 220  CONTINUE                                                           *<520000 320520000 M
C                                                                        *<530000 320530000 C
C  CALCULATE VERTICAL VELOCITY AT THE SURFACE                            *<540000 320540000 C
C                                                                         320600> 320600000 C
      FX=0.0                                                              320700> 320700000
      DO 240 I=1,IMT                                                      320800> 320800000
        W(I,1) = - ( DXUR(I)*( FUWBT(I+1)-FUWBT(I) )                     *<910000 320910000 M
     >                    + ( FVNBT(I)-FVSBT(I) ) )                      *<920000 320920000 M
        W(I,1) = -CSR(J)                                                 *<930000 320930000 -M
     >            *( DXU2R(I)*(UBT(I+1,J)-UBT(I-1,J))                    *<940000 320940000 -M
     >             + DYU2R(J)*( (VBT(I,J+1)+VBT(I,J))*CST(J+1)           *<950000 320950000 -M
     >                        - (VBT(I,J)+VBT(I,J-1))*CST( J ) ) )       *<960000 320960000 -M
 240  CONTINUE                                                            321000> 321000000
C                                                                         321100> 321100000 C
C  2ND, COMPUTE CHANGE OF W BETWEEN LEVELS                                321200> 321200000 C
C                                                                         321300> 321300000 C
      DO 250 K=1,KM                                                       321400> 321400000
      DO 250 I=1,IMT                                                      321500> 321500000
        W(I,K+1)=C2DZQ(I,K)*((FUW(I+1,K)-FUW (I,K))*DXU2RQ(I,K)           321600> 321600000
     *                         +FVN(I   ,K)-FVSU(I,K))                    321700> 321700000
 250  CONTINUE                                                            321800> 321800000
C                                                                         321900> 321900000 C
C  3RD, INTEGRATE DOWNWARD FROM THE SURFACE                               322000> 322000000 C
C                                                                         322100> 322100000 C
      DO 255 K=1,KM                                                       322200> 322200000
      DO 255 I=1,IMT                                                      322300> 322300000
        W(I,K+1)=W(I,K)+W(I,K+1)                                          322400> 322400000
 255  CONTINUE                                                            322500> 322500000
C                                                                         322600> 322600000 C
C------------------------------------------------------------------       322700> 322700000 C
C  COMPUTE HYDROSTATIC PRESSURE GRADIENT                                  322800> 322800000 C
C------------------------------------------------------------------       322900> 322900000 C
C                                                                         323000> 323000000 C
C  1ST, COMPUTE IT AT THE FIRST LEVEL                                     323100> 323100000 C
C                                                                         323200> 323200000 C
      FXA=GRAV*DZZ(1)*CSR(J)                                              323300> 323300000
      FXB=GRAV*DZZ(1)*DYU2R(J)                                            323400> 323400000
      DO 260 I=1,IMT                                                      323500> 323500000
        UDIF(I,1)=RHON(I+1,1)-RHOS(I   ,1)                                323600> 323600000
        VDIF(I,1)=RHON(I   ,1)-RHOS(I+1,1)                                323700> 323700000
        DPDX(I,1)=((UDIF(I,1)-VDIF(I,1))*FXA)*DXU2R(I)                    323800> 323800000
```

```
        DPDY(I,1)= (UDIF(I,1)+VDIF(I,1))*FXB
  260   CONTINUE
C
C  2ND, COMPUTE THE CHANGE IN PRESSURE GRADIENT BETWEEN LEVELS
C
      FXA=GRAV*CSR(J)*0.5
      FXB=GRAV*DYU4R(J)
      DO 270 K=2,KM
      DO 270 I=1,IMT
        DPDX(I,K)=RHON(I,K-1)+RHON(I,K)
        DPDY(I,K)=RHOS(I,K-1)+RHOS(I,K)
  270 CONTINUE
      DO 273 K=2,KM
      DO 273 I=1,IMT
        UDIF(I,K)=DPDX(I+1,K)-DPDY(I  ,K)
        VDIF(I,K)=DPDX(I  ,K)-DPDY(I+1,K)
        DPDX(I,K)=(FXA*(UDIF(I,K)-VDIF(I,K)))*DZZQ(I,K)*DXU2RQ(I,K)
        DPDY(I,K)=(FXB*(UDIF(I,K)+VDIF(I,K)))*DZZQ(I,K)
  273 CONTINUE
C
C  3RD, INTEGRATE DOWNWARD FROM THE FIRST LEVEL
C
      DO 275 K=1,KMM1
      DO 275 I=1,IMT
        DPDX(I,K+1)=DPDX(I,K)+DPDX(I,K+1)
        DPDY(I,K+1)=DPDY(I,K)+DPDY(I,K+1)
  275 CONTINUE
C
C-------------------------------------------------------------------
C  SET BOUNDARY CONDITIONS FOR THE COMPUTATION OF
C       VERTICAL DIFFUSION OF MOMENTUM
C-------------------------------------------------------------------
C
C  1ST, TRANSFER INTERIOR POINTS INTO DIFFUSION COMPUTATION ARRAYS
C
      DO 280 K=1,KM
      DO 280 I=1,IMT
        UDIF(I,K)=UB(I,K)
        VDIF(I,K)=VB(I,K)
  280 CONTINUE
C
C  2ND, SET K=0 ELEMENTS OF DIFF. COMP. ARRAYS TO REFLECT WIND STRESS
C
      FX=DZZ(1)/FKPM
      DO 290 I=1,IMT
        UOVER(I)=UB(I,1)+WSX(I)*FX
        VOVER(I)=VB(I,1)+WSY(I)*FX
  290 CONTINUE
C
C  3RD, SET FIRST LAND LEVEL IN EACH COLUMN TO REFLECT BOTTOM CONDITION
C
      DO 295 I=1,IMT
        KZ=KMU(I)
        UDIF(I,KZ+1)=UB(I,KZ)
        VDIF(I,KZ+1)=VB(I,KZ)
  295 CONTINUE
C
C=====================================================================
C  END INTRODUCTORY SECTION  ==========================================
C=====================================================================
C
C=====================================================================
C  BEGIN COMPUTATION OF THE INTERNAL MODES.              =============
C  THE NEW VALUES "UA" AND "VA", WILL FIRST BE LOADED WITH  ==========
```

```
C  THE TIME RATE OF CHANGE, AND THEN UPDATED.          =============
C=====================================================================
C
C--------------------------------------------------------------------
C  COMPUTE HORIZONTAL ADVECTION OF MOMENTUM
C--------------------------------------------------------------------
C
C  1ST, COMPUTE FLUX THROUGH WEST FACE OF U,V BOX
C
      DO 300 K=1,KM
      DO 300 I=1,IMT
        TEMPA(I,K)=FUW(I,K)*(U(I-1,K)+U(I,K))
        TEMPB(I,K)=FUW(I,K)*(V(I-1,K)+V(I,K))
 300  CONTINUE
C
C  2ND, COMPUTE ZONAL FLUX DIVERGENCE
C
      DO 303 K=1,KM
      DO 303 I=1,IMT
        UA(I,K)=(TEMPA(I,K)-TEMPA(I+1,K))*DXU2RQ(I,K)
        VA(I,K)=(TEMPB(I,K)-TEMPB(I+1,K))*DXU2RQ(I,K)
 303  CONTINUE
C
C  3RD, ADD IN MERIDIONAL FLUX DIVERGENCE
C
      DO 305 K=1,KM
      DO 305 I=1,IMT
        UA(I,K)=UA(I,K)-FVN (I,K)*(UP(I,K)+U (I,K))
     *                 +FVSU(I,K)*(U (I,K)+UM(I,K))
        VA(I,K)=VA(I,K)-FVN (I,K)*(VP(I,K)+V (I,K))
     *                 +FVSU(I,K)*(V (I,K)+VM(I,K))
 305  CONTINUE
C
C---------------------------------------------------------------------
C COMPUTE QU,QV FOR CONSERVATION OF MOMENTUM OVER TOPOGRAPHY IN
C VERTICAL FLUX DIVERGENCE LATER.
C---------------------------------------------------------------------
C
      DO 307 I=1,IMT
        QU(I)=0.0
        QV(I)=0.0
        DO 307 K=KMU(I)+1,KM
          QU(I)=QU(I)+UA(I,K)*DZ(K)
          QV(I)=QV(I)+VA(I,K)*DZ(K)
 307  CONTINUE
C
C--------------------------------------------------------------------
C  ADD IN HORIZONTAL DIFFUSION OF MOMENTUM (EVAL. AT TAU-1 TSTEP)
C--------------------------------------------------------------------
C
C  1ST, COMPUTE SEVERAL COEFFICIENTS DEPENDENT ONLY ON LATITUDE
C
      BBUJ=8.0*AM*CSR(J)*CSR(J)
      CCUJ=AM*CST(J+1)*DYTR(J+1)*DYUR(J)*CSR(J)
      DDUJ=AM*CST(J  )*DYTR(J  )*DYUR(J)*CSR(J)
      GGUJ=AM*(1.0-TNG(J)*TNG(J))/(RADIUS*RADIUS)
      HHUJ=2.0*AM*SINE(J)/(RADIUS*CS(J)*CS(J))
C
C  2ND, COMPUTE GRADIENTS AT WEST FACE OF U,V BOX
C
      DO 320 K=1,KM
      DO 320 I=1,IMT
        TEMPA(I,K)=DXT4RQ(I,K)*(UB(I,K)-UB(I-1,K))
        TEMPB(I,K)=DXT4RQ(I,K)*(VB(I,K)-VB(I-1,K))
```

```
 320  CONTINUE
C
C  3RD, ADD IN FINAL CONTRIBUTION FROM HOR. DIFF. OF MOMENTUM
C
      DO 323 K=1,KM
      DO 323 I=1,IMT
        UA(I,K)=UA(I,K)+BBUJ*(DXU2RQ(I,K)*(TEMPA(I+1,K)-TEMPA(I,K)))
     *          +CCUJ*(UBP(I,K)-UB(I,K))+DDUJ*(UBM(I,K)-UB(I,K))
     *          +GGUJ*UB(I,K)-HHUJ*DXU2RQ(I,K)*(VB(I+1,K)-VB(I-1,K))
        VA(I,K)=VA(I,K)+BBUJ*(DXU2RQ(I,K)*(TEMPB(I+1,K)-TEMPB(I,K)))
     *          +CCUJ*(VBP(I,K)-VB(I,K))+DDUJ*(VBM(I,K)-VB(I,K))
     *          +GGUJ*VB(I,K)+HHUJ*DXU2RQ(I,K)*(UB(I+1,K)-UB(I-1,K))
 323  CONTINUE
C
C-----------------------------------------------------------------------
C  ADD IN VERTICAL DIFFUSION OF MOMENTUM
C-----------------------------------------------------------------------
C
C  1ST, COMPUTE GRADIENTS AT TOP OF U,V BOX
C
      DO 345 K=1,KMP1
      DO 345 I=1,IMT
        TEMPA(I,K)=UDIF(I,K-1)-UDIF(I,K)
        TEMPB(I,K)=VDIF(I,K-1)-VDIF(I,K)
 345  CONTINUE
C
C  2ND, ADD IN FINAL CONTRIBUTION FROM VERT. DIFF. OF MOMENTUM
C
      DO 348 K=1,KM
      DO 348 I=1,IMT
        UA(I,K)=UA(I,K)+EEMQ(I,K)*TEMPA(I,K)-FFMQ(I,K)*TEMPA(I,K+1)
        VA(I,K)=VA(I,K)+EEMQ(I,K)*TEMPB(I,K)-FFMQ(I,K)*TEMPB(I,K+1)
 348  CONTINUE
C
C-----------------------------------------------------------------------
C  ADD IN PRESSURE TERM AND MASK OUT LAND
C-----------------------------------------------------------------------
C
      DO 350 K=1,KM
      DO 350 I=1,IMT
        UA(I,K)=GM(I,K)*(UA(I,K)-DPDX(I,K))
        VA(I,K)=GM(I,K)*(VA(I,K)-DPDY(I,K))
 350  CONTINUE
C
C-----------------------------------------------------------------------
C  CALCULATE VERTICALLY INTEGRATED FORCING
C-----------------------------------------------------------------------
C
      FX=0.0
      DO 355 I=1,IMT
        XFBT(I)=FX
        YFBT(I)=FX
 355  CONTINUE
      DO 360 K=1,KM
      DO 360 I=1,IMT
        XFBT(I)=XFBT(I)+UA(I,K)*DZ(K)
        YFBT(I)=YFBT(I)+VA(I,K)*DZ(K)
 360  CONTINUE
C
C-----------------------------------------------------------------------
C  ADD IN VERTICAL ADVECTION OF MOMENTUM AND THEN
C  ADD IN CORIOLIS FORCE (EVAL. ON TAU   TSTEP FOR EXPLICIT TRTMNT;
C                        EVAL. ON TAU-1 TSTEP FOR IMPLICIT TRTMNT
C                        WITH REMAINDER OF TERM TO BE ADDED LATER)
```

```
C-----------------------------------------------------------------------
C
      DO 340 K=1,KMP1
      DO 340 I=1,IMT
        TEMPA(I,K)=W(I,K)*(U(I,K-1)+U(I,K))
        TEMPB(I,K)=W(I,K)*(V(I,K-1)+V(I,K))
 340   CONTINUE
      DO 341 I=1,IMT
        TEMPA(I,1)=W(I,1)*2.0*U(I,1)
        TEMPB(I,1)=W(I,1)*2.0*V(I,1)
 341  CONTINUE
C
C  PATCH TO CONSERVE MOMENTUM OVER TOPOGRAPHY
C
      DO 3400 I=1,IMT
C
C  FX IS USED IN PLACE OF AN 'IF' STATEMENT TO INCREASE VECTORISATION
C
        FX=MIN(1,MAX(0,KM-KMU(I)))
        FXA=1.-FX
        TEMPA(I,KMU(I)+1)=TEMPA(I,KMU(I)+1)*FXA + 2.*QU(I)*FX
        TEMPB(I,KMU(I)+1)=TEMPB(I,KMU(I)+1)*FXA + 2.*QV(I)*FX
        XFBT(I)=XFBT(I)+QU(I)*FX
        YFBT(I)=YFBT(I)+QV(I)*FX
 3400 CONTINUE
C
      DO 343 K=1,KM
      DO 343 I=1,IMT
        UA(I,K)=UA(I,K)+(TEMPA(I,K+1)-TEMPA(I,K))*DZ2RQ(I,K)
        VA(I,K)=VA(I,K)+(TEMPB(I,K+1)-TEMPB(I,K))*DZ2RQ(I,K)
343   CONTINUE
C
C
      FX=2.0*OMEGA*SINE(J)
      IF(ACOR.EQ.0.) THEN
        DO 370 K=1,KM
        DO 370 I=1,IMT
          UA(I,K)=UA(I,K)+FX*V(I,K)
          VA(I,K)=VA(I,K)-FX*U(I,K)
 370    CONTINUE
      ELSE
        DO 375 K=1,KM
        DO 375 I=1,IMT
          UA(I,K)=UA(I,K)+FX*VB(I,K)
          VA(I,K)=VA(I,K)-FX*UB(I,K)
 375    CONTINUE
      ENDIF
C
C-----------------------------------------------------------------------
C  MASK OUT LAND AGAIN
C-----------------------------------------------------------------------
C
      DO 380 K=1,KM
      DO 380 I=1,IMT
        UA(I,K)=GM(I,K)*UA(I,K)
        VA(I,K)=GM(I,K)*VA(I,K)
 380  CONTINUE
C
C-----------------------------------------------------------------------
C  DO ANALYSIS OF INTERNAL MODE FORCING ON ENERGY TIMESTEP
C  ALSO, FORM VERT AVE. FOR USE LATER IN EXT. MODE ANALYSIS
C-----------------------------------------------------------------------
C
      FXC=0.0
```

```
      FX=0.0
      DO 395 I=1,IMT
        ZUENG(I)=FX
        ZVENG(I)=FX
 395  CONTINUE
C
C  1ST, COMPUTE KE CHANGE DUE TO PRESSURE TERM
C
      IF(NERGY.EQ.0.OR.MXP.EQ.1) GO TO 550
      FX=CS(J)*DYU(J)
C
C  (WEIGHT SYMMETRY ROW BY ONE HALF)
C
      IF(J.EQ.JMTM1) FX=FX*0.5
      DO 400 K=1,KM
      DO 400 I=2,IMUM1
        UENG(I,K)=GM(I,K)*(-DPDX(I,K))
        VENG(I,K)=GM(I,K)*(-DPDY(I,K))
        ENGINT(6)=ENGINT(6)+(USAV(I,K)*UENG(I,K)
     *                     +VSAV(I,K)*VENG(I,K))*FX*DXU(I)*DZ(K)
        ZUENG(I)=ZUENG(I)+UENG(I,K)*DZ(K)*HR(I,J)
        ZVENG(I)=ZVENG(I)+VENG(I,K)*DZ(K)*HR(I,J)
 400  CONTINUE
      DO 410 I=2,IMUM1
        ZUENG(I)=ZUENG(I) - GRAV*CSR(J)*DXU2R(I)
     >                   *( ETA(I+1,J+1) + ETA(I+1,J)
     >                    - ETA(I  ,J+1) - ETA(I  ,J) )
        ZVENG(I)=ZVENG(I) - GRAV*DYU2R(J)
     >                   *( ETA(I,J+1) + ETA(I+1,J+1)
     >                    - ETA(I,J  ) - ETA(I+1,J  ) )
        ENGEXT(6)=ENGEXT(6)+(UBT(I,J)*ZUENG(I)
     *                     +VBT(I,J)*ZVENG(I))*FX*DXU(I)
        ZUENG(I)=FXC
        ZVENG(I)=FXC
 410  CONTINUE
C
C  2ND, COMPUTE KE CHANGE DUE TO ADVECTION OF MOMENTUM
C
      DO 430 K=1,KM
      DO 430 I=2,IMUM1
        UENG(I,K)=GM(I,K)*((-FUW (I+1,K)*(U (I+1,K)+U (I   ,K))
     *                     +FUW (I   ,K)*(U (I   ,K)+U (I-1,K)))*DXU2R(I)
     *                     -FVN (I   ,K)*(UP(I   ,K)+U (I   ,K))
     *                     +FVSU(I   ,K)*(U (I   ,K)+UM(I   ,K)))
        VENG(I,K)=GM(I,K)*((-FUW (I+1,K)*(V (I+1,K)+V (I   ,K))
     *                     +FUW (I   ,K)*(V (I   ,K)+V (I-1,K)))*DXU2R(I)
     *                     -FVN (I   ,K)*(VP(I   ,K)+V (I   ,K))
     *                     +FVSU(I   ,K)*(V (I   ,K)+VM(I   ,K)))
        ENGINT(2)=ENGINT(2)+(USAV(I,K)*UENG(I,K)
     *                     +VSAV(I,K)*VENG(I,K))*FX*DXU(I)*DZ(K)
        ZUENG(I)=ZUENG(I)+UENG(I,K)*DZ(K)*HR(I,J)
        ZVENG(I)=ZVENG(I)+VENG(I,K)*DZ(K)*HR(I,J)
 430  CONTINUE
      DO 440 I=2,IMUM1
        ENGEXT(2)=ENGEXT(2)+(UBT(I,J)*ZUENG(I)
     *                     +VBT(I,J)*ZVENG(I))*FX*DXU(I)
        ZUENG(I)=FXC
        ZVENG(I)=FXC
 440  CONTINUE
      DO 460 K=1,KM
      DO 460 I=2,IMUM1
       IF (K.EQ.1) THEN
        UENG(I,K)=GM(I,K)*(-(W(I,K  )*U(I,K)*2.0
     *               -W(I,K+1)*(U(I,K)+U(I,K+1)))*DZ2RQ(I,K))
```

```
        VENG(I,K)=GM(I,K)*(-(W(I,K  )*V(I,K)*2.0
     *               -W(I,K+1)*(V(I,K)+V(I,K+1)))*DZ2RQ(I,K))
       ELSE
        UENG(I,K)=GM(I,K)*(-(W(I,K  )*(U(I,K-1)+U(I,K  ))
     *                     -W(I,K+1)*(U(I,K  )+U(I,K+1)))*DZ2RQ(I,K))
        VENG(I,K)=GM(I,K)*(-(W(I,K  )*(V(I,K-1)+V(I,K  ))
     *                     -W(I,K+1)*(V(I,K  )+V(I,K+1)))*DZ2RQ(I,K))
       ENDIF
        ENGINT(3)=ENGINT(3)+(USAV(I,K)*UENG(I,K)
     *                     +VSAV(I,K)*VENG(I,K))*FX*DXU(I)*DZ(K)
        ZUENG(I)=ZUENG(I)+UENG(I,K)*DZ(K)*HR(I,J)
        ZVENG(I)=ZVENG(I)+VENG(I,K)*DZ(K)*HR(I,J)
 460  CONTINUE
      DO 470 I=2,IMUM1
        ENGEXT(3)=ENGEXT(3)+(UBT(I,J)*ZUENG(I)
     *                     +VBT(I,J)*ZVENG(I))*FX*DXU(I)
        ZUENG(I)=FXC
        ZVENG(I)=FXC
 470  CONTINUE
C
C  3RD, COMPUTE KE CHANGE DUE TO HOR. DIFFUSION OF MOMENTUM
C
      DO 490 K=1,KM
      DO 490 I=2,IMUM1
        UENG(I,K)=GM(I,K)*(
     *          +BBUJ*DXU2R(I)*(DXT4R(I+1)*(UB(I+1,K)-UB(I,K))
     *                         +DXT4R(I  )*(UB(I-1,K)-UB(I,K)))
     *          +CCUJ*(UBP(I,K)-UB(I,K))+DDUJ*(UBM(I,K)-UB(I,K))
     *          +GGUJ*UB(I,K)-HHUJ*DXU2R(I)*(VB(I+1,K)-VB(I-1,K)))
        VENG(I,K)=GM(I,K)*(
     *          +BBUJ*DXU2R(I)*(DXT4R(I+1)*(VB(I+1,K)-VB(I,K))
     *                         +DXT4R(I  )*(VB(I-1,K)-VB(I,K)))
     *          +CCUJ*(VBP(I,K)-VB(I,K))+DDUJ*(VBM(I,K)-VB(I,K))
     *          +GGUJ*VB(I,K)+HHUJ*DXU2R(I)*(UB(I+1,K)-UB(I-1,K)))
        ENGINT(4)=ENGINT(4)+(USAV(I,K)*UENG(I,K)
     *                     +VSAV(I,K)*VENG(I,K))*FX*DXU(I)*DZ(K)
        ZUENG(I)=ZUENG(I)+UENG(I,K)*DZ(K)*HR(I,J)
        ZVENG(I)=ZVENG(I)+VENG(I,K)*DZ(K)*HR(I,J)
 490  CONTINUE
      DO 495 I=2,IMUM1
        ENGEXT(4)=ENGEXT(4)+(UBT(I,J)*ZUENG(I)
     *                     +VBT(I,J)*ZVENG(I))*FX*DXU(I)
        ZUENG(I)=FXC
        ZVENG(I)=FXC
 495  CONTINUE
C
C  4TH, COMPUTE KE CHANGE DUE TO WIND STRESS
C
      DO 522 I=2,IMUM1
        UENG(I,1)=GM(I,1)*EEM(1)*(UOVER(I)-UDIF(I,1))
        VENG(I,1)=GM(I,1)*EEM(1)*(VOVER(I)-VDIF(I,1))
        ENGINT(7)=ENGINT(7)+(USAV(I,1)*UENG(I,1)
     *                     +VSAV(I,1)*VENG(I,1))*FX*DXU(I)*DZ(1)
        ZUENG(I)=ZUENG(I)+UENG(I,1)*DZ(1)*HR(I,J)
        ZVENG(I)=ZVENG(I)+VENG(I,1)*DZ(1)*HR(I,J)
 522  CONTINUE
      DO 523 I=2,IMUM1
        ENGEXT(7)=ENGEXT(7)+(UBT(I,J)*ZUENG(I)
     *                     +VBT(I,J)*ZVENG(I))*FX*DXU(I)
        ZUENG(I)=FXC
        ZVENG(I)=FXC
 523  CONTINUE
C
C  5TH, COMPUTE KE CHANGE DUE TO BOTTOM DRAG
```

```
C
      DO 524 I=2,IMUM1
        KZ=KMU(I)
        UENG(I,KZ)=GM(I,KZ)*FFM(KZ)*(UDIF(I,KZ+1)-UDIF(I,KZ))
        VENG(I,KZ)=GM(I,KZ)*FFM(KZ)*(VDIF(I,KZ+1)-VDIF(I,KZ))
        ENGINT(8)=ENGINT(8)+(USAV(I,KZ)*UENG(I,KZ)
     *                    +VSAV(I,KZ)*VENG(I,KZ))*FX*DXU(I)*DZ(KZ)
        ZUENG(I)=ZUENG(I)+UENG(I,KZ)*DZ(KZ)*HR(I,J)
        ZVENG(I)=ZVENG(I)+VENG(I,KZ)*DZ(KZ)*HR(I,J)
 524  CONTINUE
      DO 525 I=2,IMUM1
        ENGEXT(8)=ENGEXT(8)+(UBT(I,J)*ZUENG(I)
     *                    +VBT(I,J)*ZVENG(I))*FX*DXU(I)
        ZUENG(I)=FXC
        ZVENG(I)=FXC
 525  CONTINUE
C
C  6TH, COMPUTE KE CHANGE DUE TO VERT. DIFFUSION OF MOMENTUM
C
      DO 520 I=2,IMUM1
        KZ=KMU(I)
      DO 520 K=1,KZ
        FXA=1.0
        FXB=1.0
        IF(K.EQ.1) FXA=0.0
        IF(K.EQ.KZ) FXB=0.0
        UENG(I,K)=GM(I,K)*( FXA*EEM(K)*(UDIF(I,K-1)-UDIF(I,K  ))
     *                    -FXB*FFM(K)*(UDIF(I,K  )-UDIF(I,K+1)))
        VENG(I,K)=GM(I,K)*( FXA*EEM(K)*(VDIF(I,K-1)-VDIF(I,K  ))
     *                    -FXB*FFM(K)*(VDIF(I,K  )-VDIF(I,K+1)))
        ENGINT(5)=ENGINT(5)+(USAV(I,K)*UENG(I,K)
     *                    +VSAV(I,K)*VENG(I,K))*FX*DXU(I)*DZ(K)
        ZUENG(I)=ZUENG(I)+UENG(I,K)*DZ(K)*HR(I,J)
        ZVENG(I)=ZVENG(I)+VENG(I,K)*DZ(K)*HR(I,J)
 520  CONTINUE
      DO 530 I=2,IMUM1
        ENGEXT(5)=ENGEXT(5)+(UBT(I,J)*ZUENG(I)
     *                    +VBT(I,J)*ZVENG(I))*FX*DXU(I)
        ZUENG(I)=FXC
        ZVENG(I)=FXC
 530  CONTINUE
 550  CONTINUE
C
C-----------------------------------------------------------------
C  COMPUTE NEW VELOCITIES (WITH INCORRECT VERTICAL MEANS)
C  ALSO, ADD IN REMAINDER OF CORIOLIS TERM IF TREATED IMPLICITLY
C-----------------------------------------------------------------
C
      IF(ACOR.EQ.0.) THEN
        DO 560 K=1,KM
        DO 560 I=1,IMT
          UA(I,K)=UB(I,K)+C2DTUV*UA(I,K)
          VA(I,K)=VB(I,K)+C2DTUV*VA(I,K)
 560    CONTINUE
      ELSE
        FX=C2DTUV*ACOR*2.0*OMEGA*SINE(J)
        DETMR=1.0/(1.0+FX*FX)
        DO 565 K=1,KM
        DO 565 I=1,IMT
          UDIF(I,K)=(UA(I,K)+FX*VA(I,K))*DETMR
          VDIF(I,K)=(VA(I,K)-FX*UA(I,K))*DETMR
          UA(I,K)=UB(I,K)+C2DTUV*UDIF(I,K)
          VA(I,K)=VB(I,K)+C2DTUV*VDIF(I,K)
 565    CONTINUE
```

```
      ENDIF
C
C-----------------------------------------------------------------------
C  DETERMINE THE INCORRECT VERTICAL MEANS OF THE NEW VELOCITIES
C-----------------------------------------------------------------------
C
      FX=0.0
      DO 575 I=1,IMT
        SFU(I)=FX
        SFV(I)=FX
 575  CONTINUE
      DO 580 K=1,KM
      DO 580 I=1,IMT
        SFU(I)=SFU(I)+UA(I,K)*DZ(K)
        SFV(I)=SFV(I)+VA(I,K)*DZ(K)
 580  CONTINUE
      DO 590 I=1,IMT
        SFU(I)=SFU(I)*HR(I,J)
        SFV(I)=SFV(I)*HR(I,J)
 590  CONTINUE
C
C-----------------------------------------------------------------------
C  SUBTRACT INCORRECT VERTICAL MEAN TO GET INTERNAL MODE
C-----------------------------------------------------------------------
C
      DO 600 K=1,KM
      DO 600 I=1,IMT
        UA(I,K)=UA(I,K)-SFU(I)
        VA(I,K)=VA(I,K)-SFV(I)
 600  CONTINUE
      DO 602 K=1,KM
      DO 602 I=1,IMT
        UA(I,K)=GM(I,K)*UA(I,K)
        VA(I,K)=GM(I,K)*VA(I,K)
 602  CONTINUE
C
C-----------------------------------------------------------------------
C  COMPUTE TOTAL CHANGE OF K.E. OF INTERNAL MODE ON ENERGY TIMESTEP
C-----------------------------------------------------------------------
C
      IF(NERGY.EQ.1 .AND. MXP.NE.1) THEN
      DO 605 K=1,KM
        FX=CS(J)*DYU(J)*DZ(K)/C2DTUV
        IF(J.EQ.JMTM1)FX=FX*0.5
        DO 605 I=2,IMUM1
          ENGINT(1)=ENGINT(1)+(USAV(I,K)*(UA(I,K)-UB(I,K))
     *                        +VSAV(I,K)*(VA(I,K)-VB(I,K)))*FX*DXU(I)
 605    CONTINUE
      ENDIF
C
C=========================================================================
C  END COMPUTATION OF INTERNAL MODES  ===================================
C=========================================================================
C
C-----------------------------------------------------------------------
C  SET CYCLIC BOUNDARY CONDITIONS ON EXT. MODE FORCING FUNCTIONS
C-----------------------------------------------------------------------
C
      XFBT(1)=XFBT(IMUM1)
      YFBT(1)=YFBT(IMUM1)
C
C-----------------------------------------------------------------------
C  WRITE OUT VALUES OF EXT. MODE FORCING FOR USE IN TROPIC
C-----------------------------------------------------------------------
```

```
C
      IF ( MXP.EQ.0 ) THEN
        CALL OPUT(KFLDS,IMT,  NWDS+(J-1)*IMT+1,XFBT)
        CALL OPUT(KFLDS,IMT,2*NWDS+(J-1)*IMT+1,YFBT)
      ENDIF
C
C-----------------------------------------------------------------------
C  TRANSFER QUANTITIES COMPUTED TO THE NORTH OF THE PRESENT ROW
C  TO BE DEFINED TO THE SOUTH IN THE COMPUTATION OF THE NEXT ROW
C-----------------------------------------------------------------------
C
      FX=CS(J)*DYU(J)*CSR(J+1)*DYUR(J+1)
      DO 644 K=1,KM
      DO 644 I=1,IMT
        FVSU(I,K)=FVN(I,K)*FX
 644  CONTINUE
C
C-----------------------------------------------------------------------
C  SET CYCLIC BOUNDARY CONDITIONS ON NEWLY COMPUTED INTERNAL MODE
C-----------------------------------------------------------------------
C
      DO 662 K=1,KM
        UA(1,K)=UA(IMUM1,K)
        VA(1,K)=VA(IMUM1,K)
        UA(IMU,K)=UA(2,K)
        VA(IMU,K)=VA(2,K)
 662  CONTINUE
C
C-----------------------------------------------------------------------
C  SET MERIDIONAL COMPONENT OF INTERNAL MODE TO ZERO ON SYMMETRY ROW
C-----------------------------------------------------------------------
C
      IF(J.EQ.JMTM1) THEN
        FX=0.0
        DO 850 K=1,KM
        DO 850 I=1,IMT
          VA(I,K)=FX
 850    CONTINUE
      ENDIF
C
      RETURN
      END
```

```
*DECK TRACER                                                           400000> 400000000
      SUBROUTINE TRACER(J)                                             400100> 400100000
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                              <150000 400150000
C                                                                      400200> 400200000 C
C=====================================================================400300> 400300000 C
C                                                                 ====400400> 400400000 C
C  TRACER COMPUTES, FOR ONE ROW, THE NT TRACERS, WHERE:           ====400500> 400500000 C
C           J=THE ROW NUMBER                                      ====400600> 400600000 C
C                                                                 ====400700> 400700000 C
C=====================================================================400800> 400800000 C
C                                                                      400900> 400900000 C
C--------------------------------------------------------------------- 401000> 401000000 C
C  DEFINE GLOBAL DATA                                                  401100> 401100000 C
C--------------------------------------------------------------------- 401200> 401200000 C
C                                                                      401300> 401300000 C
*CALL PARAM                                                            401400> 401400000
*CALL FULLWD                                                           401500> 401500000
*CALL SCALAR                                                           401600> 401600000
*CALL ONEDIM                                                           401700> 401700000
*CALL FIELDS                                                           401800> 401800000
*CALL WRKSPA                                                           401900> 401900000
*CALL BITVEC                                                           402000> 402000000 W
C                                                                      402100> 402100000 HC
C--------------------------------------------------------------------- 402200> 402200000 HC
C  DEMENSION LOCAL DATA                                                402300> 402300000 HC
C--------------------------------------------------------------------- 402400> 402400000 HC
C                                                                      402500> 402500000 HC
      REAL TFULL(IMT,KM)                                               402600> 402600000 H
C                                                                      402700> 402700000 C
C--------------------------------------------------------------------- 402800> 402800000 C
C  BEGIN EXECUTABLE CODE                                               402900> 402900000 C
C--------------------------------------------------------------------- 403000> 403000000 C
C                                                                      403100> 403100000 C
C=====================================================================403200> 403200000 C
C  BEGIN INTRODUCTORY SECTION, PREPARING VARIOUS  =======================403300> 403300000 C
C  ARRAYS FOR THE COMPUTATION OF THE TRACERS      =======================403400> 403400000 C
C=====================================================================403500> 403500000 C
C                                                                      403600> 403600000 C
C--------------------------------------------------------------------- 403700> 403700000 C
C  FIND ADVECTIVE COEFFICIENT 'FUW' FOR WEST  FACE OF T BOX            403800> 403800000 C
C                           & 'FVN' FOR NORTH FACE OF T BOX            403900> 403900000 C
C--------------------------------------------------------------------- 404000> 404000000 C
C                                                                      404100> 404100000 C
      FXA=CSTR(J)*DYTR(J)                                              404200> 404200000
      FXB=FXA*CS(J)                                                    404300> 404300000
      DO 690 K=1,KM                                                    404400> 404400000
      DO 690 I=1,IMT                                                   404500> 404500000
        FUW(I,K)=(U(I-1,K)*DYU (J  )+UM(I-1,K)*DYU (J-1  ))*FXA        404600> 404600000
        FVN(I,K)=(V(I   ,K)*DXUQ(I,K)+V (I-1,K)*DXUQ(I-1,K))*FXB       404700> 404700000
     *            *DXT4RQ(I,K)                                         404800> 404800000
 690  CONTINUE                                                         404900> 404900000
C                                                                      405000> 405000000 C
C--------------------------------------------------------------------- 405100> 405100000 C
C  COMPUTE VERTICAL VELOCITY IN T COLUMNS                              405200> 405200000 C
C--------------------------------------------------------------------- 405300> 405300000 C
C                                                                      405400> 405400000 C
C  1ST, COMPUTE VERTICAL VELOCITY AT THE SURFACE                      *405500> 405500000 C
C                                                                      405600> 405600000 C
      FX=0.0                                                           405700> 405700000
      DO 700 I=1,IMT                                                   405800> 405800000
        W(I,1) = -CSTR(J)*DXT2R(I)*DYTR(J)                            *405900> 405900000
     >           *(  UBT(I   ,J)*DYU(J)+UBT(I   ,J-1)*DYU(J-1)        *<910000 405910000
     >            - (UBT(I-1,J)*DYU(J)+UBT(I-1,J-1)*DYU(J-1))         *<920000 405920000
```

```
     >             + (VBT(I,J  )*DXU(I)+VBT(I-1,J  )*DXU(I-1))*CS( J )  *
     >             - (VBT(I,J-1)*DXU(I)+VBT(I-1,J-1)*DXU(I-1))*CS(J-1) )*
 700  CONTINUE
C
C  2ND, COMPUTE CHANGE OF W BETWEEN LEVELS
C
      DO 710 K=1,KM
      DO 710 I=1,IMT
        W(I,K+1)=C2DZQ(I,K)*((FUW(I+1,K)-FUW (I,K))*DXT4RQ(I,K)
     *                      +FVN(I  ,K)-FVST(I,K))
 710  CONTINUE
C
C  3RD, INTEGRATE DOWNWARD FROM THE SURFACE
C
      DO 712 K=1,KM
      DO 712 I=1,IMT
        W(I,K+1)=W(I,K)+W(I,K+1)
 712  CONTINUE
C
C-------------------------------------------------------------------
C  SET BOUNDARY CONDITIONS FOR VERTICAL DIFFUSION OF TRACERS
C-------------------------------------------------------------------
C
C  1ST, TRANSFER INTERIOR POINTS INTO DIFFUSION COMPUTATION ARRAY
C
      DO 720 M=1,NT
      DO 720 K=1,KM
      DO 720 I=1,IMT
        TDIF(I,K+1,M)=TB(I,K,M)
 720  CONTINUE
C
C  2ND, SET TOP POINT OF THE COLUMN TO REFLECT SURFACE FLUX,
C          BOTTOM POINT OF THE COLUMN TO REFLECT INSULATION.
C  (THE ROUNDOFF ERROR IN W AT THE BOTTOM IS ALSO ELIMINATED HERE)
C
      FX=0.0
      DO 730 I=1,IMT
        KZ=KMT(I)
        W(I,KZ+1)=FX
      DO 730 M=1,NT
        TDIF(I,1   ,M)=TB(I,1 ,M)
        TDIF(I,KZ+2,M)=TB(I,KZ,M)
 730  CONTINUE
C
C======================================================================
C  END INTRODUCTORY SECTION  ==========================================
C======================================================================
C
C======================================================================
C  BEGIN COMPUTATION OF THE TRACERS.                ====================
C  THE NEW VALUES "TA", WILL FIRST BE LOADED WITH   ====================
C  THE TIME RATE OF CHANGE, AND THEN UPDATED.       ====================
C======================================================================
C
      DO 855 M=1,NT
C
C-------------------------------------------------------------------
C  COMPUTE TOTAL ADVECTION OF TRACERS
C-------------------------------------------------------------------
C
C  1ST, COMPUTE FLUX THROUGH WEST FACE OF T BOX
C
      DO 810 K=1,KM
      DO 810 I=1,IMT
```

```
        TEMPA(I,K)=FUW(I,K)*(T(I,K,M)+T(I-1,K,M))
  810 CONTINUE
C
C  2ND, COMPUTE ZONAL FLUX DIVERGENCE
C
      DO 815 K=1,KM
      DO 815 I=1,IMT
        TA(I,K,M)=(TEMPA(I,K)-TEMPA(I+1,K))*DXT4RQ(I,K)
  815 CONTINUE
C
C  3RD, ADD IN MERIDIONAL FLUX DIVERGENCE
C
      DO 820 K=1,KM
      DO 820 I=1,IMT
        TA(I,K,M)=TA(I,K,M)-FVN (I,K)*(TP(I,K,M)+T (I,K,M))
     *                     +FVST(I,K)*(T (I,K,M)+TM(I,K,M))
  820 CONTINUE
C
C  4TH, COMPUTE FLUX THROUGH TOP OF T BOX
C
      DO 822 K=1,KMP1
      DO 822 I=1,IMT
        TEMPB(I,K)=W(I,K)*(T(I,K-1,M)+T(I,K,M))
  822 CONTINUE
      DO 823 I=1,IMT
        TEMPB(I,1)=W(I,1)*2.0*T(I,1,M)
  823 CONTINUE
C
C  5TH, ADD IN VERTICAL FLUX DIVERGENCE
C
      DO 824 K=1,KM
      DO 824 I=1,IMT
        TA(I,K,M)=TA(I,K,M)+(TEMPB(I,K+1)-TEMPB(I,K))*DZ2RQ(I,K)
  824 CONTINUE
C
C-----------------------------------------------------------------------
C  ADD IN HORIZONTAL DIFFUSION OF TRACERS (EVAL. AT TAU-1 TIMESTEP)
C-----------------------------------------------------------------------
C
C  1ST, COMPUTE SEVERAL COEFFICIENTS DEPENDENT ONLY ON LATITUDE
C
      BBTJ=8.0*AH*CSTR(J)*CSTR(J)
      CCTJ=AH*CS(J  )*DYUR(J  )*DYTR(J)*CSTR(J)
      DDTJ=AH*CS(J-1)*DYUR(J-1)*DYTR(J)*CSTR(J)
C
C  2ND, COMPUTE GRADIENTS AT WEST FACE OF T BOX
C
      DO 838 K=1,KM
      DO 838 I=1,IMT
        TEMPA(I,K)=DXU2RQ(I-1,K)*(TB(I,K,M)-TB(I-1,K,M))
  838 CONTINUE
C
C  3RD, ADD IN FINAL CONTRIBUTION FROM HOR. DIFF. OF TRACERS.
C  (TO PROVIDE FOR INSULATED WALLS, EACH GRADIENT IS MULTIPLIED BY
C  THE MASK OF THE POINT IN ITS RESPECTIVE DIRECTION, THUS
C  CAUSING IT TO BE ZERO IF IT IS TAKEN ACROSS A WALL)
C
      DO 840 K=1,KM
      DO 840 I=1,IMT
        TA(I,K,M)=TA(I,K,M)+BBTJ*DXT4RQ(I,K)*
     *           (FM(I+1,K)*TEMPA(I+1,K)-FM(I-1,K)*TEMPA(I,K))
     *           +CCTJ*FMP(I,K)*(TBP(I,K,M)-TB(I,K,M))
     *           +DDTJ*FMM(I,K)*(TBM(I,K,M)-TB(I,K,M))
  840 CONTINUE
```

```
C
C-----------------------------------------------------------------------
C  ADD IN VERTICAL DIFFUSION OF TRACERS
C-----------------------------------------------------------------------
C
C  1ST, COMPUTE GRADIENTS AT TOP OF T BOX
C
      DO 842 K=1,KMP1
      DO 842 I=1,IMT
        TEMPB(I,K)=TDIF(I,K,M)-TDIF(I,K+1,M)
 842  CONTINUE
C
C  2ND, ADD IN FINAL CONTRIBUTION FROM VERT. DIFF. OF TRACERS
C
      DO 844 K=1,KM
      DO 844 I=1,IMT
        TA(I,K,M)=TA(I,K,M)+EEHQ(I,K)*TEMPB(I,K  )
     *                         -FFHQ(I,K)*TEMPB(I,K+1)
 844  CONTINUE
C
C-----------------------------------------------------------------------
C  COMPUTE NEW TRACERS, RESETTING LAND POINTS TO ZERO
C-----------------------------------------------------------------------
C
      DO 850 K=1,KM
      DO 850 I=1,IMT
        TA(I,K,M)=(TB(I,K,M)+C2DTTS*TA(I,K,M))*FM(I,K)
C
C  FORCE UPDATING IN FULL PRECISION TO ASSURE CONSERVATION OF TRACERS
C
      TFULL(I,K)=TB(I,K,M)+EXTEND(C2DTTS)*TA(I,K,M)
 850  CONTINUE
C
C  ROUND TO HALF PRECISION (NOTE.. DOES NOT TRUNCATE)
C
      CALL Q8RCONV(B'00000000',0,TFULL(1,1;IMTKM),,,0,TA(1,1,M;IMTKM))
      TA(*,*,M)=TA(*,*,M)*FM(*,*)
C
 855  CONTINUE
C
C-----------------------------------------------------------------------
C  SET SALINITY TO 45 PPT OVER LAND TO STOP CONVECTION THERE
C  (..NOTE THAT THIS IS .01 IN MODEL UNITS -- (PPT-35)/1000..)
C-----------------------------------------------------------------------
C
      IF(NT.GE.2) THEN
        FXA=0.01
        FXB=1.0
        DO 860 K=1,KM
        DO 860 I=1,IMT
          TA(I,K,2)=TA(I,K,2)+FXA*(FXB-FM(I,K))
 860    CONTINUE
      ENDIF
C
C-----------------------------------------------------------------------
C  DO ANALYSIS OF TRACER FORCING ON ENERGY TIMESTEP
C-----------------------------------------------------------------------
C
      IF(NERGY.EQ.0.OR.MXP.EQ.1) GO TO 920
      DO 910 I=2,IMTM1
        KZ=KMT(I)
        IF (KZ.EQ.0) GOTO 910
        DO 900 M=1,NT
        DO 900 K=1,KZ
```

```
      TUP=T(I,K-1,M)
      IF (K.EQ.1) TUP=T(I,K,M)
        BOXVOL = CST(J)*DXT(I)*DYT(J)*DZ(K)
C
C  1ST, COMPUTE TRACER CHANGE DUE TO ADVECTION
C
        TTDTOT(2,M)=TTDTOT(2,M)+BOXVOL*
     *             ((-FUW (I+1,K)*(T (I+1,K,M)+T (I   ,K,M))
     *               +FUW (I   ,K)*(T (I   ,K,M)+T (I-1,K,M)))*DXT4R(I)
     *               -FVN (I   ,K)*(TP(I   ,K,M)+T (I   ,K,M))
     *               +FVST(I   ,K)*(T (I   ,K,M)+TM(I   ,K,M)))
        TTDTOT(3,M)=TTDTOT(3,M)+BOXVOL*
     *               (W(I,K+1)*(T(I,K   ,M)+T(I,K+1,M))
     *               -W(I,K   )*(TUP    +T(I,K   ,M)))*DZ2R(K)
C
C  2ND, COMPUTE TRACER CHANGE DUE TO HORIZONTAL DIFFUSION
C
        TTDTOT(4,M)=TTDTOT(4,M)+BOXVOL*
     *   (BBTJ*DXU2R(I   )*DXT4R(I)*FM(I+1,K)*(TB(I+1,K,M)-TB(I,K,M))
     *   +BBTJ*DXU2R(I-1)*DXT4R(I)*FM(I-1,K)*(TB(I-1,K,M)-TB(I,K,M))
     *           +CCTJ*FMP(I,K)*(TBP(I,K,M)-TB(I,K,M))
     *           +DDTJ*FMM(I,K)*(TBM(I,K,M)-TB(I,K,M)))
C
C  3RD, COMPUTE TRACER CHANGE DUE TO VERTICAL DIFFUSION
C
        TTDTOT(5,M)=TTDTOT(5,M)+BOXVOL*
     *               (EEH(K)*(TDIF(I,K,M)-TDIF(I,K+1,M))
     *               -FFH(K)*(TDIF(I,K+1,M)-TDIF(I,K+2,M)))
 900    CONTINUE
C
C  4TH, COMPUTE TOTAL ENERGY EXCHANGE BETWEEN POTENTIAL AND KINETIC
C
        IF(KZ.LT.2) GO TO 910
        FX=CST(J)*DXT(I)*DYT(J)*GRAV*0.5
        DO 905 K=2,KZ
          BUOY=BUOY-FX*DZZ(K)*W(I,K)*(RHOS(I,K-1)+RHOS(I,K))
 905    CONTINUE
        BUOY=BUOY-FX*DZZ(1)*W(I,1)*2.0*RHOS(I,1)
        BUOY=BUOY-W(I,1)*ETA(I,J)*FX*2.0
 910  CONTINUE
 920  CONTINUE
C
C-----------------------------------------------------------------
C  CONVECTIVELY ADJUST WATER COLUMN IF GRAVITATIONALLY UNSTABLE
C-----------------------------------------------------------------
C
C  SET NCON FOR NUMBER OF PASSES THROUGH CONVECTION LOOP
C  KS=1: COMPARE LEV. 1 TO 2; 3 TO 4; ETC. AND ADJUST IF NECESSARY
C  KS=2: COMPARE LEV. 2 TO 3; 4 TO 5; ETC. AND ADJUST IF NECESSARY
C
      NCON=1
      DO 965 N=1,NCON
      DO 965 KS=1,2
C
C  1ST, FIND DENSITY FOR ENTIRE SLAB FOR STABILITY DETERMINATION
C
        CALL STATEC(TA,TA(1,1,2),TEMPA,TDIF,TDIF(1,1,2),KS)
C
C  2ND, FOR EACH TRACER, MIX ADJOINING LEVELS IF UNSTABLE
C
        DO 960 K=KS,KMM1,2
        DO 960 I=2,IMTM1
          IF(TEMPA(I,K).GT.TEMPA(I,K+1)) THEN
            DO 948 M=1,NT
```

```
           TA(I,K,M)=(DZ(K)*TA(I,K,M)+DZ(K+1)*TA(I,K+1,M))*DZZ2R(K+1)
           TA(I,K+1,M)=TA(I,K,M)
 948     CONTINUE
       ENDIF
 960   CONTINUE
       FX=0.5
       LN=(2*((KMM1-KS)/2)+1)*IMT
       DO 960 M=1,NT
         TEMPB(1,1;IMTKM)=FX*C2DZQ(1,1;IMTKM)*TA(1,1,M;IMTKM)
         WHERE (KALTBV(1,KS,KS;LN)
     *         .AND. TEMPA(1,KS;LN).GT.TEMPA(1,KS+1;LN))
           TA(1,KS,M;LN)=(TEMPB(1,KS;LN)+TEMPB(1,KS+1;LN))
     *                   *DZZ2RQ(1,KS+1;LN)
           TA(1,KS+1,M;LN)=TA(1,KS,M;LN)
         ENDWHERE
 960   CONTINUE
 965  CONTINUE
C
C======================================================================
C  END COMPUTATION OF THE TRACERS  ====================================
C======================================================================
C
C---------------------------------------------------------------------
C  INTEGRATE TOTAL CHANGES IN T,S AND SQUARED T,S ON ENERGY TIMESTEP
C---------------------------------------------------------------------
C
      IF(NERGY.EQ.1 .AND. MXP.NE.1) THEN
        DO 970 M=1,NT
        DO 970 K=1,KM
         FX=CST(J)*DYT(J)*DZ(K)/C2DTTS
        DO 970 I=2,IMTM1
          TTDTOT(1,M)=TTDTOT(1,M)+(TA(I,K,M)   -TB(I,K,M)   )*FX*DXT(I)
          TVAR(M)    =TVAR(M)    +(TA(I,K,M)**2-TB(I,K,M)**2)*FX*DXT(I)
 970   CONTINUE
      ENDIF
C
C---------------------------------------------------------------------
C  FOURIER FILTER TRACERS AT HIGH LATITUDES
C---------------------------------------------------------------------
C
      IF((J.GT.JFT1.AND.J.LT.JFT2).OR.J.LT.JFRST)GO TO 1190
      JJ=J-JFRST+1
      IF (J.GE.JFT2) JJ=JJ-JSKPT+1
C
C  IF PREVIOUS STRIPS WERE OF SAME LENGTH, DONT RECOMPUTE FOURIER COEFFS
C
      ISAVE=0
      IEAVE=0
      DO 1140 L=1,LSEGF
        DO 1135 K=1,KM
          IF(ISTF(JJ,L,K).EQ.0) GO TO 1135
          IS=ISTF(JJ,L,K)
          IE=IETF(JJ,L,K)
          IREDO=0
          IF(IS.NE.ISAVE .OR. IE.NE.IEAVE) THEN
            IREDO=-1
            ISAVE=IS
            IEAVE=IE
            IM=IE-IS+1
            M=1
            N=IFIX(IM*CST(J)*CSTR(JFTO)+.5)
            IF(IM.NE.IMTM2.OR.KMT(1).LT.K) THEN
              M=1
              N=IFIX(IM*CST(J)*CSTR(JFTO)+.5)
```

```
          ELSE
            M=3
            N=IFIX(IM*CST(J)*CSTR(JFTO)*.5+.5)
          ENDIF
        ENDIF
        DO 1130 MM=1,NT
          IDX=IREDO+MM
          DO 1100 II=IS,IE
            I=MOD(II-2,IMTM2)+2
            TDIF(II+1-IS,K,1)=TA(I,K,MM)
 1100     CONTINUE
C
          CALL FILTER(TDIF(1,K,1),IM,M,N,IDX)
C
          DO 1120 II=IS,IE
            I=MOD(II-2,IMTM2)+2
            TA(I,K,MM)=TDIF(II+1-IS,K,1)
 1120     CONTINUE
 1130   CONTINUE
 1135  CONTINUE
 1140 CONTINUE
 1190 CONTINUE
C
C-------------------------------------------------------------------
C  ACCUMULATE INTEGRATED ABSOLUTE CHANGES IN T EVERY NTSI TIMESTEPS
C-------------------------------------------------------------------
C
      IF(MOD(ITT,NTSI).EQ.0) THEN
        FX=0.5*CST(J)*DYT(J)/C2DTTS
        DO 983 M=1,NT
        DO 983 K=1,KM
        DO 983 I=1,IMT
          TDIF(I,K,M)=ABS(TA(I,K,M)-TB(I,K,M))*C2DZQ(I,K)*FX*DXTQ(I,K)
 983    CONTINUE
        DO 985 M=1,NT
        DO 985 K=1,KM
        DO 985 I=2,IMTM1
          DTABS(M)=DTABS(M)+TDIF(I,K,M)
C *******************************************************************
C       DO 985 I=2,IMTM1
C         DTABS(M)=DTABS(M)+TDIF(I,K,M)
C  ****  ABOVE 2 STMNTS REPLACED BY FOLLOWING FOR VECTORIZATION  *****
          DTABS(M)=DTABS(M)+Q8SSUM(TDIF(2,K,M;IMTM2))
C *******************************************************************
 985    CONTINUE
      ENDIF
C
C-------------------------------------------------------------------
C  TRANSFER QUANTITIES COMPUTED TO THE NORTH OF THE PRESENT ROW
C  TO BE DEFINED TO THE SOUTH IN THE COMPUTATION OF THE NEXT ROW
C-------------------------------------------------------------------
C
      FX=CST(J)*DYT(J)*CSTR(J+1)*DYTR(J+1)
      DO 990 K=1,KM
      DO 990 I=1,IMT
        FVST(I,K)=FVN(I,K)*FX
        RHOS(I,K)=RHON(I,K)
 990  CONTINUE
C
C-------------------------------------------------------------------
C  SET CYCLIC BOUNDARY CONDITIONS ON NEWLY COMPUTED TRACERS
C-------------------------------------------------------------------
C
      DO 992 M=1,NT
```

```
      DO 992 K=1,KM
        TA(1   ,K,M)=TA(IMTM1,K,M)
        TA(IMT,K,M)=TA(2    ,K,M)
 992  CONTINUE
C
C-----------------------------------------------------------------------
C  SET NEW VELOCITIES AT NORTHERN WALL TO ZERO SINCE NO PASS THROUGH
C  CLINIC IS MADE FOR THIS ROW
C-----------------------------------------------------------------------
C
      IF(J.EQ.JMTM1) THEN
        FX=0.0
        DO 680 K=1,KM
        DO 680 I=1,IMT
          UA(I,K)=FX
          VA(I,K)=FX
 680    CONTINUE
      ENDIF
      RETURN
      END
```

```
*DECK STATE
      SUBROUTINE STATE(TX,SX,RHO,TQ,SQ)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C=======================================================================
C                                                                   ====
C  STATE COMPUTES ONE ROW OF NORMALIZED DENSITIES BY USING A 3RD    ====
C        ORDER POLYNOMIAL FIT TO THE KNUDSEN FORMULA, LEVEL BY      ====
C        LEVEL, WHERE:                                              ====
C          TX =THE INPUT ROW OF TEMPERATURES                        ====
C          SX =THE INPUT ROW OF SALINITIES (UNITS: (PPT-35)/1000)   ====
C          RHO=THE RETURNED ROW OF NORMALIZED DENSITIES             ====
C          TQ =ONE ROW OF WORK SPACE PROVIDED BY THE CALLING ROUTINE ====
C          SQ =ONE ROW OF WORK SPACE PROVIDED BY THE CALLING ROUTINE ====
C                                                                   ====
C=======================================================================
C
C-----------------------------------------------------------------------
C  DEFINE GLOBAL DATA
C-----------------------------------------------------------------------
C
*CALL PARAM
*CALL SCALAR
*CALL WRKSPA
*CALL TIME
C
C-----------------------------------------------------------------------
C  DEMENSION LOCAL DATA
C-----------------------------------------------------------------------
C
      DIMENSION TX(IMT,KM),SX(IMT,KM),RHO(IMT,KM),TQ(IMT,KM),SQ(IMT,KM)
      DIMENSION TO(KM),SO(KM),C(KM,9)
C
C-----------------------------------------------------------------------
C  ENTER NORMALIZING TEMPERATURES AND SALINITIES,
C  AND COEFFICIENTS GENERATED BY THE PROGRAM ("KNUDSN") WHICH
C  FITS 3RD ORDER POLYNOMIALS TO THE KNUDSEN FORMULA, LEVEL BY LEVEL.
C-----------------------------------------------------------------------
C
      DATA TO/??/
      DATA SO/??/
      DATA  C/??/
C
C-----------------------------------------------------------------------
C  BEGIN EXECUTABLE CODE
C-----------------------------------------------------------------------
      CALL GETIME(T0,T2)
C
C-----------------------------------------------------------------------
C  SUBTRACT NORMALIZING CONSTANTS FROM TEMPERATURE AND SALINITY
C  AND COMPUTE POLYNOMIAL APPROXIMATION OF KNUDSEN DENSITY.
C  (..NOTE.. FOR PRECISION PURPOSES, THERE IS A CONSTANT SUBTRACTED
C   FROM THE DENSITY RETURNED BY THIS ROUTINE. THIS MAKES NO DIFFERENCE
C   HOWEVER, SINCE ONLY HORIZONTAL GRADIENTS ARE USED BY THE MODEL.)
C-----------------------------------------------------------------------
C
      DO 100 K=1,KM
      DO 100 I=1,IMT
        TQ(I,K)=TX(I,K)-TOQ(I,K)
        SQ(I,K)=SX(I,K)-SOQ(I,K)
        RHO(I,K)=(CQ(I,K,1)+(CQ(I,K,4)+CQ(I,K,7)*SQ(I,K))*SQ(I,K)
     *  +(CQ(I,K,3)+CQ(I,K,8)*SQ(I,K)+CQ(I,K,6)*TQ(I,K))*TQ(I,K))
     *  *TQ(I,K)+(CQ(I,K,2)+(CQ(I,K,5)+CQ(I,K,9)
```

```
     *  *SQ(I,K))*SQ(I,K))*SQ(I,K)
 100  CONTINUE
      CALL GETIME(T1,T2)
      TIME(6)=TIME(6)+T1-T0
      RETURN
C
      ENTRY STATEC(TX,SX,RHO,TQ,SQ,IND)
C
C========================================================================
C                                                                    ====
C  STATEC COMPUTES, FOR ONE ROW, THE NORMALIZED DENSITIES BY USING   ====
C        A 3RD ORDER POLYNOMIAL FIT TO THE KNUDSEN FORMULA, FOR      ====
C        PURPOSES OF CHECKING VERTICAL STABILITY BETWEEN ADJACENT    ====
C        LEVELS.  THE REFERENCE DEPTH FOR PRESSURE DEPENDENCE IN     ====
C        THE KNUDSEN FORMULA MUST BE HELD CONSTANT FOR THIS PURPOSE.====
C        THAT LEVEL IS DETERMINED BY "IND".  THE ARGUMENTS ARE:      ====
C      TX =THE INPUT ROW OF TEMPERATURES                             ====
C      SX =THE INPUT ROW OF SALINITIES (UNITS: (PPT-35)/1000)        ====
C      RHO=THE RETURNED ROW OF NORMALIZED DENSITIES                  ====
C      TQ =ONE ROW OF WORK SPACE PROVIDED BY THE CALLING ROUTINE     ====
C      SQ =ONE ROW OF WORK SPACE PROVIDED BY THE CALLING ROUTINE     ====
C      IND=1 FOR COMPARING LEVELS 1 TO 2, 3 TO 4, ETC.               ====
C          --COEFFICIENTS FOR THE LOWER OF THE 2 LEVELS ARE USED     ====
C      IND=2 FOR COMPARING LEVELS 2 TO 3, 4 TO 5, ETC.               ====
C          --COEFFICIENTS FOR THE UPPER OF THE 2 LEVELS ARE USED     ====
C                                                                    ====
C========================================================================
C
      CALL GETIME(T0,T2)
      DO 1100 L=1,KM
      DO 1100 I=1,IMT
      TQ(I,L)=TX(I,L)-TOIQ(I,L,IND)
      SQ(I,L)=SX(I,L)-SOIQ(I,L,IND)
      RHO(I,L)=(CIQ(I,L,1,IND)+(CIQ(I,L,4,IND)+CIQ(I,L,7,IND)
     1*SQ(I,L))*SQ(I,L)+(CIQ(I,L,3,IND)+CIQ(I,L,8,IND)*SQ(I,L)
     2+CIQ(I,L,6,IND)*TQ(I,L))*TQ(I,L))*TQ(I,L)+(CIQ(I,L,2,IND)
     3+(CIQ(I,L,5,IND)+CIQ(I,L,9,IND)*SQ(I,L))*SQ(I,L))*SQ(I,L)
 1100 CONTINUE
      CALL GETIME(T1,T2)
      TIME(7)=TIME(7)+T1-T0
      RETURN
C
      ENTRY STINIT
C
C========================================================================
C                                                                    ====
C  STINIT LOADS THE APPROPRIATE NORMALIZATION CONSTANTS AND COEF-    ====
C        FICIENTS INTO ARRAYS OF PROPER DIMENSION TO PERMIT VEC-     ====
C        TORIZATION IN THE SUBSEQUENT CALLS TO "STATE" AND "STATEC"  ====
C                                                                    ====
C========================================================================
C
C-----------------------------------------------------------------------
C  LOAD COEFFICIENTS FOR USE IN STATE
C-----------------------------------------------------------------------
C
      DO 10 N=1,9
      DO 10 K=1,KM
      DO 10 I=1,IMT
        CQ(I,K,N)=C(K,N)
 10   CONTINUE
      DO 20 K=1,KM
      DO 20 I=1,IMT
        TOQ(I,K)=TO(K)
```

```
      SOQ(I,K)=SO(K)
 20   CONTINUE
C
C-----------------------------------------------------------------------
C  LOAD COEFFICIENTS FOR USE IN STATEC.
C  DETERMINE THE REFERENCE LEVEL INDICATOR, "KREF" IN ACCORD WITH
C  COMMENT ON "IND" IN INTRODUCTORY STATEMENT FOR ENTRY STATEC.
C-----------------------------------------------------------------------
C
      DO 70 JND=1,2
        DO 52 K=1,KM,2
          IF (JND.EQ.1) THEN
            KREF=K+1
            IF(KREF.GT.KM) KREF=KM
          ELSE
            KREF=K
          ENDIF
          DO 50 I=1,IMT
            TOIQ(I,K,JND)=TOQ(I,KREF)
            SOIQ(I,K,JND)=SOQ(I,KREF)
 50       CONTINUE
          DO 52 N=1,9
          DO 52 I=1,IMT
            CIQ(I,K,N,JND)=CQ(I,KREF,N)
 52     CONTINUE
        DO 62 K=2,KM,2
          IF (JND.EQ.2) THEN
            KREF=K+1
            IF(KREF.GT.KM) KREF=KM
          ELSE
            KREF=K
          ENDIF
          DO 60 I=1,IMT
            TOIQ(I,K,JND)=TOQ(I,KREF)
            SOIQ(I,K,JND)=SOQ(I,KREF)
 60       CONTINUE
          DO 62 N=1,9
          DO 62 I=1,IMT
            CIQ(I,K,N,JND)=CQ(I,KREF,N)
 62     CONTINUE
 70   CONTINUE
      RETURN
      END
```

```
*DECK MATRIX
      SUBROUTINE MATRIX(ARRAY,IRDIM,ISTRT,IM,JM,KK,SCALE)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C==========================================================================
C                                                                     ====
C  MATRIX IS A GENERAL TWO-DIMENSIONAL ARRAY PRINTING ROUTINE,        ====
C      WHERE:                                                         ====
C      ARRAY=THE ARRAY TO BE PRINTED                                  ====
C      IRDIM=THE 1ST DIMENSION OF ARRAY                               ====
C      ISTRT=THE 1ST ELEMENT OF THE 1ST DIMENSION TO BE PRINTED       ====
C      IM   =THE LAST ELEMENT OF THE 1ST DIMENSION TO BE PRINTED      ====
C      JM   =THE LAST ELEMENT OF THE 2ND DIMENSION TO BE PRINTED      ====
C            --THE ROWS ARE PRINTED IN DESCENDING ORDER--             ====
C            (IF THIS IS 0, KK IS USED)                               ====
C      KK   =THE LAST ELEMENT OF THE 2ND DIMENSION TO BE PRINTED      ====
C            --THE ROWS ARE PRINTED IN ASCENDING ORDER--              ====
C            (IF THIS IS 0, JM IS USED)                               ====
C      SCALE=A SCALING FACTOR BY WHICH ARRAY IS DIVIDED BEFORE        ====
C            PRINTING.  (IF THIS IS ZERO, NO SCALING IS DONE.)        ====
C            IF SCALE=0, 10 COLUMNS ARE PRINTED ACROSS IN E FORMAT    ====
C            IF SCALE>0, 20 COLUMNS ARE PRINTED ACROSS IN F FORMAT    ====
C                                                                     ====
C==========================================================================
C
C-----------------------------------------------------------------------
C  DEFINE GLOBAL DATA
C-----------------------------------------------------------------------
C
*CALL PARAM
C
C-----------------------------------------------------------------------
C  DEFINE LOCAL DATA
C-----------------------------------------------------------------------
C
      DIMENSION ARRAY(IRDIM,1),NUM(20),PLINE(20)
C
C-----------------------------------------------------------------------
C  BEGIN EXECUTABLE CODE
C-----------------------------------------------------------------------
C
      IF(SCALE.NE.0) GO TO 500
      DO 251 IS=ISTRT,IM,10
      IE=IS+9
      IF(IE.GT.IM) IE=IM
      IDIF=IE-IS+1
      DO 100 I=1,IDIF
 100  NUM(I)=IS+I-1
      PRINT 9990, (NUM(I),I=1,IDIF)
 9990 FORMAT(10I13,/)
      JMORKM=JM+KK
      DO 252 JORK=1,JMORKM
      IF(JM.NE.0) L=JMORKM+1-JORK
      IF(KK.NE.0) L=JORK
      PRINT 9966, L, (ARRAY(I,L),I=IS,IE)
 252  CONTINUE
      PRINT 9984
 251  CONTINUE
 9966 FORMAT(1X,I2,10(1PE13.5))
 9984 FORMAT(//)
      RETURN
 500  CONTINUE
      SCALER=1.0/SCALE
```

```
      DO 751 IS=ISTRT,IM,20
      IE=IS+19
      IF(IE.GT.IM) IE=IM
      IDIF=IE-IS+1
      DO 600 I=1,IDIF
600   NUM(I)=IS+I-1
      PRINT 9991, (NUM(I),I=1,IDIF)
9991  FORMAT(3X,20I6,/)
      JMORKM=JM+KK
      DO 752 JORK=1,JMORKM
      IF(JM.NE.0) L=JMORKM+1-JORK
      IF(KK.NE.0) L=JORK
      DO 753 I=1,IDIF
      PLINE(I)=ARRAY(IS+I-1,L)*SCALER
753   CONTINUE
      PRINT 9967,L,(PLINE(I),I=1,IDIF)
752   CONTINUE
      PRINT 9984
751   CONTINUE
9967  FORMAT(1X,I3,1X,20F6.2)
      RETURN
      END
```

```
*DECK ODAM                                                                     630000> 630000000
      SUBROUTINE ODAM                                                          630100> 630100000 -K
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                      <150000 630150000 -K
C                                                                              630200> 630200000 -KC
C========================================================================630300> 630300000 -KC
C                                                                          ====630400> 630400000 -KC
C  ODAM (OCEAN DIRECT ACCESS MANAGER) IS A SET OF ROUTINES                 ====630500> 630500000 -KC
C       WHICH IS RESPONSIBLE FOR HANDLING THE TRANSFER OF DATA             ====630600> 630600000 -KC
C       FROM CENTRAL MEMORY TO DISC AND VICE VERSA.  FOR THIS              ====630700> 630700000 -KC
C       PURPOSE, IT USES THE GFDL LOCAL UTILITIES, QDAM (QUEUED            ====630800> 630800000 -KC
C       DIRECT ACCESS MANAGER), WHERE:                                     ====630900> 630900000 -KC
C         LU   =I/O UNIT NUMBER                                            ====631000> 631000000 -KC
C         NTOT =LENGTH OF UNIT, IN WORDS                                   ====631100> 631100000 -KC
C         NBLK =LENGTH OF EACH BLOCK ON THE UNIT                           ====631200> 631200000 -KC
C         NBUF =NUMBER OF BUFFERS SUPPLIED TO THE UNIT                     ====631300> 631300000 -KC
C         NWDS =NUMBER OF WORDS TO TRANSFER                                ====631400> 631400000 -KC
C         NFRST=UNIT ADDRESS OF THE FIRST WORD TO BE TRANSFERRED           ====631500> 631500000 -KC
C         A    =ORIGINATION/DESTINATION ARRAY IN MEMORY                    ====631600> 631600000 -KC
C                                                                          ====631700> 631700000 -KC
C========================================================================631800> 631800000 -KC
C                                                                              631900> 631900000 -KC
C-----------------------------------------------------------------------       632000> 632000000 -KC
C  DEMENSION ARRAY PASSED IN ARGUMENT  (..NOTE.. 8 IS ARBITRARY)               632100> 632100000 -KC
C-----------------------------------------------------------------------       632200> 632200000 -KC
C                                                                              632300> 632300000 -KC
      DIMENSION A(8)                                                           632400> 632400000 -K
C                                                                             *<400010 632400010 -KC
C     TO SUPPRESS DISK I/O INITIALISE ISTAT TO ZERO                           *<400020 632400020 -KC
C                                                                             *<400030 632400030 -KC
      DATA ISTAT/9/                                                           *<400040 632400040 -K
C                                                                              632500> 632500000 -KC
C-----------------------------------------------------------------------       632600> 632600000 -KC
C  BEGIN EXECUTABLE CODE                                                       632700> 632700000 -KC
C-----------------------------------------------------------------------       632800> 632800000 -KC
C                                                                              632900> 632900000 -KC
C-----------------------------------------------------------------------       633000> 633000000 -KC
C  INITIALIZE ONE DISC UNIT                                                    633100> 633100000 -KC
C-----------------------------------------------------------------------       633200> 633200000 -KC
C                                                                              633300> 633300000 -KC
      ENTRY OSTART(LU,NTOT,NBLK,NBUF)                                          633400> 633400000 -K
      NLB=NBLK/512+1                                                          *633500> 633500000 -K
      CALL WOPEN(LU,NLB,ISTAT)                                                *633600> 633600000 -K
      RETURN                                                                   633700> 633700000 -K
C                                                                              633800> 633800000 -KC
C-----------------------------------------------------------------------       633900> 633900000 -KC
C  INITIATE A DISC-TO-MEMORY TRANSFER (BEGIN DATA FEED TO BUFFER)              634000> 634000000 -KC
C-----------------------------------------------------------------------       634100> 634100000 -KC
C                                                                              634200> 634200000 -KC
      ENTRY OFIND(LU,NWDS,NFRST)                                               634300> 634300000 -K
      CALL SEEK(LU,NFRST,NWDS)                                                *634400> 634400000 -K
      RETURN                                                                   634600> 634600000 -K
C                                                                              634700> 634700000 -KC
C-----------------------------------------------------------------------       634800> 634800000 -KC
C  COMPLETE A DISC-TO-MEMORY TRANSFER (FINISH DATA FEED TO BUFFER AND          634900> 634900000 -KC
C       TRANSFER INTO MEMORY)                                                  635000> 635000000 -KC
C-----------------------------------------------------------------------       635100> 635100000 -KC
C                                                                              635200> 635200000 -KC
      ENTRY OGET(LU,NWDS,NFRST,A)                                              635300> 635300000 -K
      CALL GETWA(LU,A,NFRST,NWDS)                                             *635400> 635400000 -K
      RETURN                                                                   635600> 635600000 -K
C                                                                              635700> 635700000 -KC
C-----------------------------------------------------------------------       635800> 635800000 -KC
C  INITIATE A MEMORY-TO-DISC TRANSFER (TRANSFER DATA TO BUFFER AND             635900> 635900000 -KC
```

```
C       BEGIN A FEED TO THE DISC)
C-----------------------------------------------------------------------
C
      ENTRY OPUT(LU,NWDS,NFRST,A)
      CALL APUTWA(LU,A,NFRST,NWDS)
      RETURN
C
C-----------------------------------------------------------------------
C  TERMINATE ALL I/O ON THE SPECIFIED UNIT
C      (GENERATE A SUMMARY REPORT FOR I/O ACTIVITY WHEN CLOSING UNIT 15)
C-----------------------------------------------------------------------
C
      ENTRY OCLOSE(LU)
      CALL WCLOSE(LU)
      IF(ISTAT.EQ.0)RETURN
      IF(LU.EQ.15)CALL WSUMMARY(ISTAT)
      RETURN
C
      END
      SUBROUTINE ODAM
      SUBROUTINE WSUMMARY(IST)
      REWIND('$STATS')
      CALL COPYD('$STATS','$OUT')
      RETURN
      END
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C=======================================================================
C
C  ODAM (OCEAN DIRECT ACCESS MANAGER) IS A SET OF ROUTINES WHICH
C       SIMULATES MEMORY/DISC DATA TRANSFER MANAGEMENT, USING THE
C       CENTRAL MEMORY ARRAY, "BIG" AS A VIRTUAL DISC. THIS RESULTS
C       IN A TOTALLY CORE CONTAINED SYSTEM. THE ARGUMENTS, IN TERMS
C       OF NORMAL DISC I/O, ARE:
C         LU   =I/O UNIT NUMBER
C         NTOT =LENGTH OF UNIT, IN WORDS
C         NBLK =LENGTH OF EACH BLOCK ON THE UNIT
C         NBUF =NUMBER OF BUFFERS SUPPLIED TO THE UNIT
C         NWRS =NUMBER OF WORDS TO TRANSFER
C         NFRST=UNIT ADDRESS OF THE FIRST WORD TO BE TRANSFERRED
C         A    =ORIGINATION/DESTINATION ARRAY IN MEMORY
C
C=======================================================================
C
C-----------------------------------------------------------------------
C  DEFINE GLOBAL DATA
C-----------------------------------------------------------------------
C
*CALL PARAM
C
C-----------------------------------------------------------------------
C  DEFINE AND EQUIVALENCE LOCAL DATA
C-----------------------------------------------------------------------
C
      PARAMETER(N11=20,N12=IMT*JMT*NKFLDS,
     * NSL=((NT+2)*IMTKM+LBC*IMT)*JMT,NTB=N11+N12+3*NSL)
      COMMON /BG/ BIG(NTB)
      DIMENSION A(8),C(8)
      EQUIVALENCE (BIG,C)
C
C-----------------------------------------------------------------------
C  BEGIN EXECUTABLE CODE
C-----------------------------------------------------------------------
C
```

```
      ENTRY OSTART(LU,NTOT,NBLK,NBUF)
C
C-----------------------------------------------------------------------
C  FORCE THE ARRAY "BIG" TO BE MAPPED ON LARGE PAGES
C-----------------------------------------------------------------------
C
      IF(LU.NE.11) RETURN
      LEN=NTB/512+1
      LEN=NTB/1024+1
      CALL Q5MAPIN('DROPF','VBA=',BIG,'LEN=',LEN,'LPAGE')
      RETURN
      ENTRY OFIND(LU,NWRS,NFRST)
      RETURN
C
C-----------------------------------------------------------------------
C  VIRTUAL DISC-TO-MEMORY TRANSFER
C-----------------------------------------------------------------------
C
      ENTRY OGET(LU,NWRS,NFRST,A)
      NSKP=0
      IF(LU.GT.11) NSKP=NSKP+N11
      IF(LU.GT.12) NSKP=NSKP+N12
      NS=NFRST+NSKP
      IF(LU.GE.13) NS=NSKP+(3*((NFRST-1)/NWRS)+(LU-13))*NWRS+1
      NE=NS+NWRS-1
      DO 100 N=NS,NE
        A(N-NS+1)=C(N)
 100  CONTINUE
      RETURN
C
C-----------------------------------------------------------------------
C  MEMORY-TO-VIRTUAL DISC TRANSFER
C-----------------------------------------------------------------------
C
      ENTRY OPUT(LU,NWRS,NFRST,A)
      NSKP=0
      IF(LU.GT.11) NSKP=NSKP+N11
      IF(LU.GT.12) NSKP=NSKP+N12
      NS=NFRST+NSKP
      IF(LU.GE.13) NS=NSKP+(3*((NFRST-1)/NWRS)+(LU-13))*NWRS+1
      NE=NS+NWRS-1
      DO 120 N=NS,NE
        C(N)=A(N-NS+1)
 120  CONTINUE
      RETURN
      ENTRY OCLOSE(LU)
      RETURN
C
C-----------------------------------------------------------------------
C  INITIALIZE VIRTUAL DISC FROM TAPE
C-----------------------------------------------------------------------
C
      ENTRY ORD(LO)
      READ(LO)BIG
      RETURN
C
C-----------------------------------------------------------------------
C  SAVE VIRTUAL DISC TO TAPE
C-----------------------------------------------------------------------
C
      ENTRY OWRT(LO)
      REWIND 22
      WRITE(LO)BIG
      RETURN
```

END

648100> 648100000 K

```
*DECK FILTER
      SUBROUTINE FILTER(S,IM,MM,N,ISS)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C=====================================================================
C                                                                ====
C  FILTER FOURIER ANALYSES THE ARRAYS OF VARIOUS                 ====
C         PHYSICAL QUANTITIES, THEN TRUNCATES THE SERIES AND     ====
C         RESYNTHESIZES THE FILTERED QUANTITIES WHERE:           ====
C              S  =THE STRING TO BE FILTERED                     ====
C              IM =THE LENGTH OF S                               ====
C              MM =1 (COSINE SERIES, BNDRY PTS=0)                ====
C                 =2 (SINE SERIES, DERIV AT BNDRY=0)             ====
C                 =3 (FULL SERIES, CYCLIC)                       ====
C              N  =NUMBER OF WAVES TO KEEP                       ====
C              ISS=0 (CANT USE FOURIER COEFS FROM PREVIOUS CALL) ====
C              ISS>0 (CAN  USE FOURIER COEFS FROM PREVIOUS CALL) ====
C                                                                ====
C=====================================================================
C
C-------------------------------------------------------------------
C  DEFINE GLOBAL DATA
C-------------------------------------------------------------------
C
*CALL PARAM
C
C-------------------------------------------------------------------
C  DEFINE LOCAL DATA AND DIMENSION ARGUMENT ARRAYS
C-------------------------------------------------------------------
C
      PARAMETER (IMTX2=IMT*2,NI=IMT)
      PARAMETER (IMTD2=IMT/2,LQMSUM=IMTD2*(IMT-IMTD2),LHSUM=IMT*IMTP1/2)
      PARAMETER (IMTX4=IMT*4,IMTX8=IMT*8,IMTIMT=IMT*IMT)
      PARAMETER (IMP1X2=IMTP1*2)
C
C     COSSAV MUST REMAIN FULL PRECISION IF MOST OF FILTER IS MADE HALF-P
      REAL COSSAV
C
      DIMENSION ICBASE(IMTP1),IDBASE(IMTP1),IND(IMTP1)
      DIMENSION COSSAV(LQMSUM),DENMSV(LHSUM),COSNPI(IMT)
      DIMENSION CIRCLE(4)
      DIMENSION INDX(IMTX8),COF(IMTX8)
      DIMENSION COSINE(IMTX8),FTARR(IMTIMT)
      DIMENSION DENOM(IMTX4)
      DIMENSION S(IMT),SPRIME(IMT)
      DIMENSION DIFF(IMP1X2)
C
      REAL T0,T1,T2,TIME
      COMMON /TIME/ TIME(10)
C
      LOGICAL INITDN
      DATA INITDN/.FALSE./
C
      DATA PI/3.141592653589793/, CIRCLE/0.,-1.,0.,1./
C
C-------------------------------------------------------------------
C  BEGIN EXECUTABLE CODE
C-------------------------------------------------------------------
C
      CALL GETIME(T0,T2)
      IF (IM.LT.1 .OR. MM.LT.1 .OR. MM.GT.3 .OR. N.LT.0 .OR. ISS.LT.0)
     * GOTO 4000
C
```

```
      IF (INITDN) GOTO 90
C
C  THIS SECTION SETS UP TABLES FOR FILTER; IT MUST BE CALLED ONCE PER
C    EXECUTION OF OCEAN
C
C  NOTE: LQMSUM IS THE SUM OF (IM-1)/2 FOR IM=1,IMTP1
C          LHSUM IS THE SUM OF IM-1 FOR IM=1,IMTP1
C
      IMSAVE=IM
C
C  ASSEMBLE INDEX ARRAY
C
      DO 10 I=1,IMTP1
      IND(I)=I
 10   CONTINUE
C
C  CALCULATE AND SAVE ALL COSINES WHICH WILL BE NEEDED
C
      IBASE=0
      JBASE=0
      DO 50 IM=1,IMTP1
      FIMR=1.0/FLOAT(IM)
C
      IMM1=IM-1
      IF (IMM1.EQ.0) GOTO 25
      DO 20 I=1,IMM1
      DENMSV(IBASE+I)=1.0/(1.0-COS(PI*FLOAT(I)*FIMR))
 20   CONTINUE
 25   IDBASE(IM)=IBASE
      IBASE=IBASE+IMM1
C
      IMQC=(IM-1)/2
      IF (IMQC.EQ.0) GOTO 35
      DO 30 I=1,IMQC
      COSSAV(JBASE+I)=COS(PI*FLOAT(I)*FIMR)
 30   CONTINUE
 35   ICBASE(IM)=JBASE
      JBASE=JBASE+IMQC
C
 50   CONTINUE
C
C  CALCULATE ADJUSTMENTS FOR GENERAL FOURIER CASE IF IM=2*N
C
      DO 60 IM=1,IMT
      COSNPI(IM)=CIRCLE(MOD(IM-1,4)+1)
 60   CONTINUE
      INITDN=.TRUE.
      IM=IMSAVE
C
C
C  CALCULATE SOME USEFUL CONSTANTS
C
 90   IF(MM.EQ.2 .AND. N.EQ.0) THEN
        DO 92 I=1,IM
          S(I)=0.0
 92     CONTINUE
        GO TO 3950
      ENDIF
      IF (MM.EQ.1) THEN
        NMAX=N-1
      ELSE
        NMAX=N
      ENDIF
      NMAXP1=NMAX+1
```

```
C
      IF (MM.EQ.2) THEN
        LCY=2*(IM+1)
        FNORM=2.0/FLOAT(IM+1)
      ELSE
        LCY=2*IM
        FNORM=2.0/FLOAT(IM)
      ENDIF
      LH=LCY/2
      LHM1=LH-1
      LQM=(LH-1)/2
      L2CY=2*LCY
      LCYM1=LCY-1
      LCYP1=LCY+1
C
      IMX2=IM*2
      IMX4=IM*4
      IMX8=IM*8
C
C  AVERAGE INCOMING ARRAY
C
      SSUM=0.0
      DO 100 I=1,IM
 100  SSUM=SSUM + S(I)
C
C     MM = 1  DERIVATIVE MUST BE ZERO AT BOUNDARIES (COSINE)
C     MM = 2  VALUE MUST BE ZERO AT BOUNDARIES (SINE)
C     MM = 3  CYCLIC BOUNDARY CONDITIONS (GENERAL FOURIER SERIES)
C
      FIM=FLOAT(IM)
      FIMR=1.0/FIM
      STEMP=SSUM*FIMR
      IF (N.GT.1 .OR. MM.NE.1) GO TO 400
      DO 300 I=1,IM
 300  S(I)=STEMP
      GO TO 3950
 400  CONTINUE
      IF(MM.NE.2) THEN
        DO 450 I=1,IM
        S(I)=S(I)-STEMP
 450    CONTINUE
      ENDIF
      IF (ISS.GT.0) GO TO 3000
C
C
C  ASSEMBLE APPROPRIATE 1-CYCLE (2*PI) COSINE ARRAY
C
C  USE STORED 1/4 CYCLE TO CALCULATE FIRST 1/2 CYCLE
      JBASE=ICBASE(LH)
      DO 700 I=1,LQM
      COSINE(I)=COSSAV(JBASE+I)
      COSINE(LH-I)=-COSSAV(JBASE+I)
 700  CONTINUE
C  FILL IN COS(PI/2) IF LH IS EVEN
      IF (2*(LQM+1).EQ.LH) COSINE(LQM+1)=0.0
C  FILL IN COS(PI) IN ANY CASE
      COSINE(LH)=-1.0
C  FILL IN REST OF CYCLE
      DO 710 I=1,LH
      COSINE(LH+I)=-COSINE(I)
 710  CONTINUE
C
C  ASSEMBLE DENOMINATOR ARRAY
C
```

```
      IBASE=IDBASE(LH)
      FXA=0.25
      DO 720 I=1,LHM1
      DENOM(I)=FXA*DENMSV(IBASE+I)
  720 CONTINUE
      DENOM(LH)=0.125
      DO 721 I=1,LHM1
      DENOM(LH+I)=DENOM(LH-I)
  721 CONTINUE
      NPRINT=0
      DENOM(LCY)=0.0
      DO 730 I=LCYP1,IMX4
      DENOM(I)=DENOM(I-LCY)
  730 CONTINUE
C
C  ASSEMBLE APPROPRIATE SUBSCRIPT ARRAYS
C
C  CALCULATE NEEDED INDICES
C
      IF (MM.EQ.3) THEN
        FACT1=2*NMAX
        FACT2=2*NMAXP1
      ELSE
        FACT1=NMAX
        FACT2=NMAXP1
      ENDIF
      DO 740 I=1,IMX4
      INDX(I)=I*FACT1
      INDX(IMX4+I)=I*FACT2
  740 CONTINUE
C  CALCULATE PARAMETERS FOR REDUCING INDICES
      MAXIND=IMX4*FACT2
      NCYC=(MAXIND-1)/LCY + 1
      MAXNDX=LCY
      IF (MAXNDX.GE.MAXIND) GOTO 790
      DO 750 NPWR=1,NCYC+2
      MAXNDX=2*MAXNDX
      IF (MAXNDX.GE.MAXIND) GOTO 760
  750 CONTINUE
      STOP 'ERROR: FELL THROUGH DO-LOOP TERMINATION AT 750'
  760 DO 770 NP=1,NPWR
      MAXNDX=MAXNDX/2
      DO 765 I=1,IMX8
        IF(INDX(I).GT.MAXNDX) INDX(I)=INDX(I)-MAXNDX
  765 CONTINUE
      WHERE (INDX(1;IMX8).GT.MAXNDX)
        INDX(1;IMX8)=INDX(1;IMX8)-MAXNDX
      ENDWHERE
  770 CONTINUE
  790 CONTINUE
C
C  GATHER COEFFICIENTS
C
  800 DO 810 J=1,IMX8
      COF(J)=COSINE(INDX(J))
  810 CONTINUE
C
      GO TO (1000,1300,1600),MM
C
C  ASSEMBLE TRANSFORMATION ARRAY WHICH WILL FILTER S
C
C
C  COSINE TRANSFORM
 1000 IOFF1=LCY
```

```
      IOFF2=LCY+IMX4
      FXA=0.5
      DO 1200 J=1,IM
      JOFF=(J-1)*IMT
      DO 1100 I=1,IM
      FTARR(JOFF+I)=(COF(I-J+IOFF1)-COF(I-J+IOFF2))*DENOM(I-J+IOFF1)
     *             +(COF(I+J-1)-COF(IMX4+I+J-1))*DENOM(I+J-1) - FXA
 1100 CONTINUE
      FTARR(JOFF+J)=FTARR(JOFF+J)+0.5*FLOAT(NMAX)+0.25
 1200 CONTINUE
      GOTO 3000
C
C  SINE TRANSFORM
 1300 IOFF1=LCY
      IOFF2=LCY+IMX4
      DO 1500 J=1,IM
      JOFF=(J-1)*IMT
      DO 1400 I=1,IM
      FTARR(JOFF+I)=(COF(I-J+IOFF1)-COF(I-J+IOFF2))*DENOM(I-J+IOFF1)
     *             -(COF(I+J)-COF(IMX4+I+J))*DENOM(I+J)
 1400 CONTINUE
      FTARR(JOFF+J)=FTARR(JOFF+J)+0.5*FLOAT(NMAX)+0.25
 1500 CONTINUE
      GOTO 3000
C
C  GENERAL FOURIER TRANSFORM
 1600 IF (2*N.EQ.IM) THEN
        GENADJ=0.5
      ELSE
        GENADJ=0.0
      ENDIF
      IOFF1=LCY
      IOFF2=LCY+IMX4
      FXA=2.0
      FXB=0.5
      DO 1800 J=1,IM
      JOFF=(J-1)*IMT
      DO 1700 I=1,IM
      FTARR(JOFF+I)=
     * (FXA*(COF(I-J+IOFF1)-COF(I-J+IOFF2)))*DENOM(2*I-2*J+IOFF1)
     * -FXB - GENADJ*COSNPI(I)*COSNPI(J)
 1700 CONTINUE
      FTARR(JOFF+J)=FTARR(JOFF+J)+FLOAT(NMAX)+0.5
 1800 CONTINUE
C     GOTO 3000
C
C  FILTER S
C
 3000 DO 3010 I=1,IM
      SPRIME(I)=0.
 3010 CONTINUE
      DO 3100 I=1,IM
      IOFF=(I-1)*IMT
      DO 3100 J=1,IM
C  NOTE THAT FTARR(J,I)=FTARR(I,J), SO FOLLOWING IS LEGAL
      SPRIME(J)=SPRIME(J)+S(I)*FTARR(IOFF+J)
 3100 CONTINUE
      DO 3110 I=1,IM
      SPRIME(I)=FNORM*SPRIME(I)
 3110 CONTINUE
      IF(MM.EQ.2) THEN
        DO 3150 I=1,IM
          S(I)=SPRIME(I)
 3150   CONTINUE
```

```
          GO TO 3950
        ENDIF
C
 3700 SSM=0.0
      DO 3800 I=1,IM
      SSM=SSM+SPRIME(I)
 3800 CONTINUE
      SSM=(SSUM-SSM)*FIMR
      DO 3900 I=1,IM
      S(I)=SSM+SPRIME(I)
 3900 CONTINUE
 3950 CONTINUE
      CALL GETIME(T1,T2)
      TIME(9)=TIME(9)+T1-T0
      RETURN
 4000 PRINT 4001, IM,MM,N,ISS
 4001 FORMAT (' BAD ARGUMENT(S) IN CALL TO FILTER'/' IM,MM,N,ISS = ',
     *  4I10)
      STOP 'BAD ARGUMENT(S) IN CALL TO FILTER'
      END
```

```
*DECK FINDEX
      SUBROUTINE FINDEX(FKXX,JJMAX,KMAX,JF1,JF2,IMAX,ISF,IEF)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
C=======================================================================
C                                                                   ====
C  FINDEX FINDS AND PRINTS STARTING AND ENDING INDICES              ====
C         FOR FILTERING, WHERE:                                     ====
C              FKXX =FIELD OF MAXIMUM LEVELS FOR THE QUANTITY       ====
C                     BEING FILTERED                                ====
C              JJMAX=NUMBER OF ROWS TO BE FILTERED                  ====
C              KMAX =MAXIMUM NUMBER OF LEVELS TO BE FILTERED        ====
C              JF1  =LAST ROW IN THE SOUTH TO BE FILTERED           ====
C              JF2  =FIRST ROW IN THE NORTH TO BE FILTERED          ====
C              IMAX =MAXIMUM I INDEX TO BE FILTERED                 ====
C              ISF  =RETURNED VALUES OF STARTING INDICES            ====
C              IEF  =RETURNED VALUES OF ENDING INDICES              ====
C                                                                   ====
C=======================================================================
C
C-----------------------------------------------------------------------
C  DEFINE GLOBAL DATA
C-----------------------------------------------------------------------
C
*CALL PARAM
C
C-----------------------------------------------------------------------
C  DEFINE LOCAL DATA AND DIMENSION ARGUMENT ARRAYS
C-----------------------------------------------------------------------
C
      DIMENSION FKXX(IMT,JMT)
     * ,ISF(JJMAX,LSEGF,KMAX),IEF(JJMAX,LSEGF,KMAX)
     * ,IIS(LSEGF+1),IIE(LSEGF+1)
C
C-----------------------------------------------------------------------
C  BEGIN EXECUTABLE CODE
C-----------------------------------------------------------------------
C
C FIND START AND END INDICES
C
      JJ = 0
      DO 100 J = JFRST,JMTM1
        IF (J.GT.JF1 .AND. J.LT.JF2) GOTO 100
        JJ = JJ+1
        DO 80 K = 1,KMAX
          DO 30 L = 1,LSEGF+1
            IIS(L) = 0
            IIE(L) = 0
 30       CONTINUE
          L = 1
          IF (FKXX(2,J).GE.K) THEN
            IIS(1) = 2
          ENDIF
          DO 50 I = 2,IMAX-1
            IF (FKXX(I-1,J).LT.K .AND. FKXX(I,J).GE.K) THEN
              IIS(L) = I
            ENDIF
            IF (FKXX(I,J).GE.K .AND. FKXX(I+1,J).LT.K) THEN
              IF (I.NE.IIS(L) .OR. (I.EQ.2 .AND. FKXX(1,J).GE.K))THEN
                IIE(L) = I
                L = L+1
              ELSE
                IIS(L) = 0
```

```
                  ENDIF
               ENDIF
 50         CONTINUE
            IF (FKXX(IMAX-1,J).GE.K .AND. FKXX(IMAX,J).GE.K) THEN
               IIE(L) = IMAX-1
               L = L+1
            ENDIF
            LM = L-1
            IF (LM.GT.1) THEN
               IF (IIS(1).EQ.2 .AND. IIE(LM).EQ.IMAX-1
     *                       .AND. FKXX(1,J).GE.K) THEN
                  IIS(1) = IIS(LM)
                  IIE(1) = IIE(1) + IMAX-2
                  IIS(LM) = 0
                  IIE(LM) = 0
                  LM = LM-1
               ENDIF
            ENDIF
            IF (LM .GT. LSEGF) THEN
               PRINT 1000, J, K
               STOP 34
            ENDIF
 1000       FORMAT (1X,'LSEGF NOT LARGE ENOUGH.  J=',I4,'    K=',I3)
            DO 70 L = 1,LSEGF
               ISF(JJ,L,K) = IIS(L)
               IEF(JJ,L,K) = IIE(L)
 70         CONTINUE
 80      CONTINUE
 100  CONTINUE
C
C PRINT THEM
C
      LLAST=LSEGF
      IF (LLAST .GT. 11) LLAST=11
      JJ=JJ+1
      DO 200 J=JMTM1,JFRST,-1
         IF (J.GT.JF1 .AND. J.LT.JF2) GO TO 200
         JJ = JJ-1
         IF (KMAX .GT. 1) THEN
            PRINT 1010,J
            DO 150 K=1,KMAX
               PRINT 1011,K,(ISF(JJ,L,K),IEF(JJ,L,K),L=1,LLAST)
 150        CONTINUE
         ELSE
            PRINT 1011,J,(ISF(JJ,L,1),IEF(JJ,L,1),L=1,LLAST)
         ENDIF
 200  CONTINUE
 1010 FORMAT (/1X,'INDICES FOR ROW ',I3,':')
 1011 FORMAT (1X,I9,3X,11(I5,I4))
      RETURN
      END
```

```
*DECK TROPIC                                                            *800000> 800000000
      SUBROUTINE TROPIC                                                 *800100> 800100000
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                <150000 800150000
C                                                                       *800200> 800200000 C
C=======================================================================*800300> 800300000 C
C                                                                       *800400> 800400000 C
C        S. M. PATERSON (19/1/1988) D.STAINFORTH (6/4/1989)             *800500> 800500000 C
C                                                                       *800600> 800600000 C
C  TROPIC TAKES AS INPUT THE EXTERNAL MODE FORCING CALCULATED IN        *800700> 800700000 C
C        "CLINIC" (XF,YF) AND BY TIME STEPPING THE BAROTROPIC EQUATIONS*800800> 800800000 C
C        CALCULATES THE SURFACE DISPLACEMENT (ETA) AND THE BAROTROPIC  *800900> 800900000 C
C        TRANSPORTS (UBT,VBT).  THE INITIAL VALUES OF THE TRANSPORTS   *801000> 801000000 C
C        ARE WRITTEN TO DISK FOR USE IN THE NEXT BAROCLINIC TIME STEP. *801100> 801100000 C
C                                                                       *801200> 801200000 C
C        THIS VERSION DOES NOT INCLUDE ANY CODE FOR THE COX UPDATE      *801300> 801300000 C
C        OPTIONS 'F','H' OR 'T'.                                        *801400> 801400000 C
C                                                                       *801500> 801500000 C
C=======================================================================*801600> 801600000 C
C                                                                       *801700> 801700000 C
C---------------------------------------------------------------------- *801800> 801800000 C
C  DEFINE GLOBAL DATA                                                   *801900> 801900000 C
C---------------------------------------------------------------------- *802000> 802000000 C
C                                                                       *802100> 802100000 C
*CALL PARAM                                                             *802200> 802200000
*CALL FULLWD                                                            *802300> 802300000
*CALL SCALAR                                                            *802400> 802400000
*CALL ONEDIM                                                            *802500> 802500000
*CALL FIELDS                                                            *802600> 802600000
*CALL WRKSPB                                                            *802700> 802700000
C                                                                       *802800> 802800000 C
C---------------------------------------------------------------------- *802900> 802900000 C
C  LOCAL DATA VARIABLE ETAD IS SET UP IN THE WRKSPB                     *803000> 803000000 C
C---------------------------------------------------------------------- *803100> 803100000 C
C                                                                       *803200> 803200000 C
C-----------------------------------------------------------------------*803300> 803300000 C
C  QUEUE UP DISK READS FOR TROPIC.                                      *803400> 803400000 C
C-----------------------------------------------------------------------*803500> 803500000 C
C                                                                       *803600> 803600000 C
      CALL OFIND(KFLDS,NWDS,6*NWDS+1)                                   *803700> 803700000
      CALL OFIND(KFLDS,NWDS,  NWDS+1)                                   *803800> 803800000
      CALL OFIND(KFLDS,NWDS,2*NWDS+1)                                   *803900> 803900000
C                                                                       *804000> 804000000 C
C-----------------------------------------------------------------------*804100> 804100000 C
C  SET ETAGRD TO ZERO TO AVOID UNSET VARIABLE ERROR                     *804200> 804200000 C
C-----------------------------------------------------------------------*804300> 804300000 C
C                                                                       *804400> 804400000 C
      DO 20 J=1,JMT                                                     *804500> 804500000
      DO 20 I=1,IMT                                                     *804600> 804600000
        ETAGRD(I,J) = 0.                                                *804700> 804700000
 20   CONTINUE                                                          *804800> 804800000
C                                                                       *804900> 804900000 C
C-----------------------------------------------------------------------*805000> 805000000 C
C  CALCULATE ARRAYS USED IN THE HORIZONTAL DIFFUSION TERM AND ALSO THE  *805100> 805100000 C
C  CORIOLIS PARAMETER.                                                  *805200> 805200000 C
C-----------------------------------------------------------------------*805300> 805300000 C
C                                                                       *805400> 805400000 C
      DO 30 J=2,JMTM2                                                   *805500> 805500000 -S
      DO 30 J=2,JMTM1                                                   *805600> 805600000 S
        BHD(J) = CSR(J)**2                                              *805700> 805700000
        CHD(J) = DYUR(J)*CSR(J)*CST(J+1)                                *805800> 805800000
        DHD(J) = DYUR(J)*CSR(J)*CST(J)                                  *805900> 805900000
        GHD(J) = (1.-TNG(J)**2)/(RADIUS**2)                             *806000> 806000000
        HHD(J) = SINE(J)/(RADIUS*CS(J)**2)                              *806100> 806100000
```

```
        FUV(J) = 2.*OMEGA*SINE(J)                                            *806200> 806200000
 30   CONTINUE                                                               *806300> 806300000
C                                                                            *806400> 806400000 C
C----------------------------------------------------------------------------*806500> 806500000 C
C   READ THE MASK EM.                                                        *806600> 806600000 C
C   READ FORCING TERMS XF,YF AND WRITE OUT CURRENT TRANSPORTS UBT,VBT.       *806700> 806700000 C
C----------------------------------------------------------------------------*806800> 806800000 C
C                                                                            *806900> 806900000 C
      CALL OGET(KFLDS,NWDS,6*NWDS+1,EM)                                      *807000> 807000000
C                                                                            *807100> 807100000 C
      CALL OGET(KFLDS,NWDS,  NWDS+1,XF)                                      *807200> 807200000
      CALL OGET(KFLDS,NWDS,2*NWDS+1,YF)                                      *807300> 807300000
C                                                                            *807400> 807400000 CS
C   SET Y FORCING ON SYMMETRY ROW TO ZERO                                    *807500> 807500000 CS
C                                                                            *807600> 807600000 CS
      DO 35 I=1,IMT                                                          *807700> 807700000 S
        YF(I,JMTM1)=0.0                                                      *807800> 807800000 S
 35   CONTINUE                                                               *807900> 807900000 S
C                                                                            *808000> 808000000 C
      CALL OPUT(KFLDS,NWDS,  NWDS+1,UBT)                                     *808100> 808100000
      CALL OPUT(KFLDS,NWDS,2*NWDS+1,VBT)                                     *808200> 808200000
C                                                                            *808300> 808300000 CD
C----------------------------------------------------------------------------*808400> 808400000 CD
C  CALCULATE THE ARRAY SPLR USED TO SET PSUEDO VALUES OF ETA ON LAND.        *808500> 808500000 CD
C----------------------------------------------------------------------------*808600> 808600000 CD
C                                                                            *808700> 808700000 CD
      DO 40 L=1,NSPL                                                         *808800> 808800000 D
        I = ISPL(L)                                                          *808900> 808900000 D
        J = JSPL(L)                                                          *809000> 809000000 D
        SPLR(L) = 1./( EM(I+1,J) + EM(I-1,J) + EM(I,J+1) + EM(I,J-1) )       *809100> 809100000 D
 40   CONTINUE                                                               *809200> 809200000 D
C                                                                            *809300> 809300000 C
C----------------------------------------------------------------------------*809400> 809400000 C
C  CALCULATE BAROTROPIC VELOCITIES UT AND VT                                 *809500> 809500000 C
C----------------------------------------------------------------------------*809600> 809600000 C
C                                                                            *809700> 809700000 C
      DO 43  J=1,JMT                                                         *809800> 809800000
      DO 43  I=1,IMU                                                         *809900> 809900000
        UT(I,J) = UBT(I,J)*HR(I,J)                                           *810000> 810000000
        VT(I,J) = VBT(I,J)*HR(I,J)                                           *810100> 810100000
 43   CONTINUE                                                               *810200> 810200000
C                                                                            *810300> 810300000 C
C----------------------------------------------------------------------------*810400> 810400000 C
C  START LOOP TO SUBTRACT FRICTION TERM FROM THE FORCING TERMS XF/YF         *810500> 810500000 C
C----------------------------------------------------------------------------*810600> 810600000 C
C                                                                            *810700> 810700000 C
      DO 45  J=2,JMTM2                                                       *810800> 810800000 -S
      DO 45  J=2,JMTM1                                                       *810900> 810900000 S
      DO 45  L=1,LSU(J)                                                      *811000> 811000000
      DO 45  I=ISU(J,L),IEU(J,L)                                             *811100> 811100000
C                                                                            *811200> 811200000 C
C----------------------------------------------------------------------------*811300> 811300000 C
C  CALCULATE FRICTION TERM FOR UBT/VBT.  THEN SUBTRACT THIS FROM THE         *811400> 811400000 C
C  APPROPRIATE FORCING TERM XF/YF                                            *811500> 811500000 C
C----------------------------------------------------------------------------*811600> 811600000 C
C                                                                            *811700> 811700000 C
        FRIC = BHD(J)*DXUR(I)*( DXTR(I+1)*( UT(I+1,J) - UT(I   ,J) )         *811800> 811800000
     >                        - DXTR(I   )*( UT(I   ,J) - UT(I-1,J) ) )      *811900> 811900000
     >       + CHD(J)*DYTR(J+1)*( UT(I,J+1) - UT(I,J   ) )                   *812000> 812000000
     >       - DHD(J)*DYTR(J   )*( UT(I,J   ) - UT(I,J-1) )                  *812100> 812100000
     >       + GHD(J)*UT(I,J)                                                *812200> 812200000
     >       - HHD(J)*DXUR(I)*( VT(I+1,J) - VT(I-1,J) )                      *812300> 812300000
C                                                                            *812400> 812400000 C
        BTFRIC  = AM*FRIC/HR(I,J)                                            *812500> 812500000
```

```
C                                                                    *812600> 812600000 C
C                                                                    *812700> 812700000 C
          XF(I,J) = XF(I,J) - BTFRIC                                 *812800> 812800000
C                                                                    *812900> 812900000 C
C                                                                    *813000> 813000000 C
        FRIC = BHD(J)*DXUR(I)*( DXTR(I+1)*( VT(I+1,J) - VT(I   ,J) )    *813100> 813100000
     >                         - DXTR(I   )*( VT(I   ,J) - VT(I-1,J) ) )  *813200> 813200000
     >        + CHD(J)*DYTR(J+1)*( VT(I,J+1) - VT(I,J   ) )          *813300> 813300000
     >        - DHD(J)*DYTR(J   )*( VT(I,J   ) - VT(I,J-1) )          *813400> 813400000
     >        + GHD(J)*VT(I,J)                                       *813500> 813500000
     >        + HHD(J)*DXUR(I)*( UT(I+1,J) - UT(I-1,J) )             *813600> 813600000
C                                                                    *813700> 813700000 C
        BTFRIC = AM*FRIC/HR(I,J)                                     *813800> 813800000
C                                                                    *813900> 813900000 C
          YF(I,J) = YF(I,J) - BTFRIC                                 *814000> 814000000
 45   CONTINUE                                                       *814100> 814100000
C                                                                    *814200> 814200000 C-D
C-------------------------------------------------------------------*814300> 814300000 C-D
C  SET ETAD EQUIVALENT TO ETA.                                       *814400> 814400000 C-D
C-------------------------------------------------------------------*814500> 814500000 C-D
      DO 80 J=1,JMT                                                  *814600> 814600000 -D
      DO 80 I=1,IMT                                                  *814700> 814700000 -D
        ETAD(I,J)=ETA(I,J)                                           *814800> 814800000 -D
 80   CONTINUE                                                       *814900> 814900000 -D
C                                                                    *815000> 815000000 C
C===================================================================*815100> 815100000 C
C  TIME STEP BAROTROPIC EQUATIONS BETWEEN (LARGER) BAROCLINIC TIME STEPS*815200> 815200000 C
C===================================================================*815300> 815300000 C
C                                                                    *815400> 815400000 C
      DO 500 ITBT = 1,NB                                             *815500> 815500000
C                                                                    *815600> 815600000 C
C                                                                    *815700> 815700000 CD
C-------------------------------------------------------------------*815800> 815800000 CD
C  FIRST SET PSEUDO ETA VALUES ON LAND FOR USE BY DLPL.              *815900> 815900000 CD
C-------------------------------------------------------------------*816000> 816000000 CD
C                                                                    *816100> 816100000 CD
      DO 50 L=1,NSPL                                                 *816200> 816200000 D
        I = ISPL(L)                                                  *816300> 816300000 D
        J = JSPL(L)                                                  *816400> 816400000 D
        ETA(I,J) = ( EM(I+1,J)*ETA(I+1,J)                            *816500> 816500000 D
     >              + EM(I-1,J)*ETA(I-1,J)                           *816600> 816600000 D
     >              + EM(I,J+1)*ETA(I,J+1)                           *816700> 816700000 D
     >              + EM(I,J-1)*ETA(I,J-1) )*SPLR(L)                 *816800> 816800000 D
 50   CONTINUE                                                       *816900> 816900000 D
C                                                                    *817000> 817000000 CD
      DO 60 I=2,IMTM1                                                *817100> 817100000 D
        ETA(I,  1) = ETA(I,2)                                        *817200> 817200000 D
        ETA(I,JMT) = ETA(I,JMTM1)                                    *817300> 817300000 D
 60   CONTINUE                                                       *817400> 817400000 D
C                                                                    *817500> 817500000 CD
      DO 70 J=2,JMT                                                  *817600> 817600000 DS
      DO 70 J=2,JMTM1                                                *817700> 817700000 D-S
        ETA(  1,J) = ETA(2,J)                                        *817800> 817800000 D-O
        ETA(IMT,J) = ETA(IMTM1,J)                                    *817900> 817900000 D-O
        ETA(  1,J) = ETA(IMTM1,J)                                    *818000> 818000000 OD
        ETA(IMT,J) = ETA(2,J)                                        *818100> 818100000 OD
 70   CONTINUE                                                       *818200> 818200000 D
C                                                                    *818300> 818300000 CD
C-------------------------------------------------------------------*818400> 818400000 CD
C  SET ETAD EQUIVALENT TO ETA.                                       *818500> 818500000 CD
C-------------------------------------------------------------------*818600> 818600000 CD
      DO 80 J=1,JMT                                                  *818700> 818700000 D
      DO 80 I=1,IMT                                                  *818800> 818800000 D
        ETAD(I,J)=ETA(I,J)                                           *818900> 818900000 D
```

```
 80   CONTINUE                                                          *819000> 819000000 D
C                                                                       *819100> 819100000 C
C-----------------------------------------------------------------------*819200> 819200000 C
C  CALCULATE BAROTROPIC VELOCITIES UT AND VT                            *819300> 819300000 C
C-----------------------------------------------------------------------*819400> 819400000 C
C                                                                       *819500> 819500000 C
      DO 140 J=1,JMT                                                    *819600> 819600000 S
      DO 140 J=1,JMTM2                                                  *819700> 819700000 -S
      DO 140 I=1,IMU                                                    *819800> 819800000
        UT(I,J) = UBT(I,J)*HR(I,J)                                      *819900> 819900000
        VT(I,J) = VBT(I,J)*HR(I,J)                                      *820000> 820000000
 140  CONTINUE                                                          *820100> 820100000
C                                                                       *820200> 820200000 C-G
C-----------------------------------------------------------------------*820300> 820300000 C-G
C  SET UP 'DO LOOP' TO PERFORM  EULER BACKWARD TIMESTEP.                *820400> 820400000 C-G
C-----------------------------------------------------------------------*820500> 820500000 C-G
C                                                                       *820600> 820600000 C-G
      DO 90 ID=1,2                                                      *820700> 820700000 -G
C                                                                       *820800> 820800000 C
C-----------------------------------------------------------------------*820900> 820900000 C
C  COMPUTE ETAGRD FROM ETA,UBT,VBT                                      *821000> 821000000 C
C-----------------------------------------------------------------------*821100> 821100000 C
C                                                                       *821200> 821200000 C
      FX = DTBT*WGHT*GRAV*ZDZ(KM)                                       *821300> 821300000 D
C                                                                       *821400> 821400000 C
      DO 100 J=2,JMTM1                                                  *821500> 821500000
      DO 100 L=1,LSE(J)                                                 *821600> 821600000
      DO 100 I=ISE(J,L),IEE(J,L)                                        *821700> 821700000
C                                                                       *821800> 821800000 CD
        DLPL = ETA(I,J+1) + ETA(I+1,J)                                  *821900> 821900000 D
     >       + ETA(I-1,J) + ETA(I,J-1) - 4.*ETA(I,J)                    *822000> 822000000 D
C                                                                       *822100> 822100000 CD
        DLCR = 0.5*( EM(I+1,J+1)*ETA(I+1,J+1)                           *822200> 822200000 D
     >             + EM(I-1,J+1)*ETA(I-1,J+1)                           *822300> 822300000 D
     >             + EM(I+1,J-1)*ETA(I+1,J-1)                           *822400> 822400000 D
     >             + EM(I-1,J-1)*ETA(I-1,J-1)                           *822500> 822500000 D
     >             - ( EM(I+1,J+1) + EM(I-1,J+1)                        *822600> 822600000 D
     >               + EM(I+1,J-1) + EM(I-1,J-1) )*ETA(I,J) )           *822700> 822700000 D
C                                                                       *822800> 822800000 D
C                                                                       *822900> 822900000 C
        ETAGRD(I,J) = DXT2R(I)*DYTR(J)                                  *823000> 823000000
     >                *( UBT(I  ,J)*DYU(J) + UBT(I  ,J-1)*DYU(J-1)      *823100> 823100000
     >                 -( UBT(I-1,J)*DYU(J) + UBT(I-1,J-1)*DYU(J-1))    *823200> 823200000
     >           + ( VBT(I,J)*DXU(I) + VBT(I-1,J)*DXU(I-1))*CS(J)       *823300> 823300000
     >           -(VBT(I,J-1)*DXU(I) + VBT(I-1,J-1)*DXU(I-1))*CS(J-1))  *823400> 823400000
     >               - FX*DXTR(I)*DYTR(J)*( DLPL - DLCR )               *823500> 823500000 D
C                                                                       *823600> 823600000 C
 100  CONTINUE                                                          *823700> 823700000
C                                                                       *823800> 823800000 C
C-----------------------------------------------------------------------*823900> 823900000 CG
C  IF TIDAL THEN COMPUTE ETA AND ETAD FROM ETAGRD AND ETA (OLD)         *824000> 824000000 CG
C-----------------------------------------------------------------------*824100> 824100000 CG
C                                                                       *824200> 824200000 CG
      DO 110 J=1,JMT                                                    *824300> 824300000 G
      DO 110 I=1,IMT                                                    *824400> 824400000 G
        ETA(I,J) = EM(I,J)*( ETA(I,J) - DTBT*CSTR(J)*ETAGRD(I,J) )      *824500> 824500000 G
        ETAD(I,J) = ETA(I,J)                                            *824600> 824600000 G
 110  CONTINUE                                                          *824700> 824700000 G
C                                                                       *824800> 824800000 CG
      DO 115 I=1,IMT                                                    *824900> 824900000 GS
        ETAD(I,JMT)=ETAD(I,JMTM1)                                       *825000> 825000000 GS
        ETA(I,JMT)=ETA(I,JMTM1)                                         *825100> 825100000 GS
 115  CONTINUE                                                          *825200> 825200000 GS
C                                                                       *825300> 825300000 CG
```

```
      DO 120 J=2,JMT                                                     *825400> 825400000 OG
        ETA(  1,J) = ETA(IMTM1,J)                                        *825500> 825500000 OG
        ETAD( 1,J) =ETA( 1,J)                                            *825600> 825600000 OG
        ETA(IMT,J) = ETA(2,J)                                            *825700> 825700000 OG
        ETAD(IMT,J)= ETA(IMT,J)                                          *825800> 825800000 OG
 120  CONTINUE                                                           *825900> 825900000 OG
C-----------------------------------------------------------------------*826000> 826000000 C
C  START LOOP TO STEP UBT AND VBT FROM ETAD,UT AND VT.                   *826100> 826100000 C
C-----------------------------------------------------------------------*826200> 826200000 C
C                                                                        *826300> 826300000 C
     DO 160 J=2,JMTM2                                                    *826400> 826400000 -S
     DO 160 J=2,JMTM1                                                    *826500> 826500000 S
       FACTOR=FUV(J)*DTBT*0.5                                            *826600> 826600000
       FAC2=1.0+FACTOR**2                                                *826700> 826700000
     DO 160 L=1,LSU(J)                                                   *826800> 826800000
     DO 160 I=ISU(J,L),IEU(J,L)                                          *826900> 826900000
C                                                                        *827000> 827000000 C
C  CALCULATE H FROM HR(I,J) FOR USE BELOW.                               *827100> 827100000 C
C                                                                        *827200> 827200000 C
     H = 1./HR(I,J)                                                      *827300> 827300000
C                                                                        *827400> 827400000 C
C-----------------------------------------------------------------------*827500> 827500000 C
C  CALCULATE FRICTION TERM FOR UBT AND THEN CALCULATE UBTGRD.            *827600> 827600000 C
C-----------------------------------------------------------------------*827700> 827700000 C
C                                                                        *827800> 827800000 C
       FRIC = BHD(J)*DXUR(I)*( DXTR(I+1)*( UT(I+1,J) - UT(I  ,J) )       *827900> 827900000
    >                        - DXTR(I  )*( UT(I  ,J) - UT(I-1,J) ) )     *828000> 828000000
    >        + CHD(J)*DYTR(J+1)*( UT(I,J+1) - UT(I,J  ) )                *828100> 828100000
    >        - DHD(J)*DYTR(J  )*( UT(I,J  ) - UT(I,J-1) )                *828200> 828200000
    >        + GHD(J)*UT(I,J)                                            *828300> 828300000
    >        - HHD(J)*DXUR(I)*( VT(I+1,J) - VT(I-1,J) )                  *828400> 828400000
C                                                                        *828500> 828500000 C
       BTFRIC  = AM*H*FRIC                                               *828600> 828600000
C                                                                        *828700> 828700000 C
C                                                                        *828800> 828800000 C
       UBTGRD = XF(I,J)                                                  *828900> 828900000
    >         + BTFRIC                                                   *829000> 829000000
    >         + H * ( FUV(J)*VT(I,J)*0.5                                 *829100> 829100000
    >                 - GRAV*CSR(J)*DXU2R(I)                             *829200> 829200000
    >                   *( ETAD(I+1,J+1) + ETAD(I+1,J)                   *829300> 829300000
    >                    - ETAD(I  ,J+1) - ETAD(I  ,J) ) )               *829400> 829400000
C                                                                        *829500> 829500000 C
       UBTDUM= ( UT(I,J) * H )  + DTBT*UBTGRD                            *829600> 829600000
C                                                                        *829700> 829700000 C
C-----------------------------------------------------------------------*829800> 829800000 C
C  CALCULATE FRICTION TERM FOR VBT AND THEN CALC VBTGRD.                 *829900> 829900000 C
C-----------------------------------------------------------------------*830000> 830000000 C
C                                                                        *830100> 830100000 C
       FRIC = BHD(J)*DXUR(I)*( DXTR(I+1)*( VT(I+1,J) - VT(I  ,J) )       *830200> 830200000
    >                        - DXTR(I  )*( VT(I  ,J) - VT(I-1,J) ) )     *830300> 830300000
    >        + CHD(J)*DYTR(J+1)*( VT(I,J+1) - VT(I,J  ) )                *830400> 830400000
    >        - DHD(J)*DYTR(J  )*( VT(I,J  ) - VT(I,J-1) )                *830500> 830500000
    >        + GHD(J)*VT(I,J)                                            *830600> 830600000
    >        + HHD(J)*DXUR(I)*( UT(I+1,J) - UT(I-1,J) )                  *830700> 830700000
C                                                                        *830800> 830800000 C
       BTFRIC = AM*H*FRIC                                                *830900> 830900000
C                                                                        *831000> 831000000 C
C                                                                        *831100> 831100000 C
       VBTGRD = YF(I,J)                                                  *831200> 831200000
    >         + BTFRIC                                                   *831300> 831300000
    >         - H * ( FUV(J)*UT(I,J)*0.5                                 *831400> 831400000
    >                 + GRAV*DYU2R(J)                                    *831500> 831500000
    >                   *( ETAD(I,J+1) + ETAD(I+1,J+1)                   *831600> 831600000
    >                    - ETAD(I,J  ) - ETAD(I+1,J  ) ) )               *831700> 831700000
```

```
C                                                                          *831800> 831800000 C
      VBTDUM =( VT(I,J) * H ) + DTBT*VBTGRD                                *831900> 831900000
C                                                                          *832000> 832000000 C
C--------------------------------------------------------------------------*832100> 832100000 C
C  CALCULATE UBT AND VBT FROM UBTDUM AND VBTDUM.                           *832200> 832200000 C
C--------------------------------------------------------------------------*832300> 832300000 C
C                                                                          *832400> 832400000 C
      UBT(I,J)=(UBTDUM+FACTOR*VBTDUM)/FAC2                                 *832500> 832500000
      VBT(I,J)=(VBTDUM-FACTOR*UBTDUM)/FAC2                                 *832600> 832600000
 160  CONTINUE                                                             *832700> 832700000
C                                                                          *832800> 832800000 CS
C--------------------------------------------------------------------------*832900> 832900000 CS
C  SET SYMMETRIC CONDITIONS ON UBT AND VBT.                                *833000> 833000000 CS
C--------------------------------------------------------------------------*833100> 833100000 CS
      DO 165 I=1,IMT                                                       *833200> 833200000 S
        UBT(I,JMT)=UBT(I,JMTM2)                                            *833300> 833300000 S
        VBT(I,JMTM1)=0.0                                                   *833400> 833400000 S
 165  CONTINUE                                                             *833500> 833500000 S
C                                                                          *833600> 833600000 OC
C--------------------------------------------------------------------------*833700> 833700000 OC
C  ALLOW FOR CYCLIC CONDITIONS                                             *833800> 833800000 OC
C--------------------------------------------------------------------------*833900> 833900000 OC
C                                                                          *834000> 834000000 OC
      DO 240 J=2,JMT                                                       *834100> 834100000 OS
      DO 240 J=2,JMTM2                                                     *834200> 834200000 O-S
        UBT(  1,J) = UBT(IMUM1,J)                                          *834300> 834300000 O
        VBT(  1,J) = VBT(IMUM1,J)                                          *834400> 834400000 O
        UBT(IMU,J) = UBT(2,J)                                              *834500> 834500000 O
        VBT(IMU,J) = VBT(2,J)                                              *834600> 834600000 O
 240  CONTINUE                                                             *834700> 834700000 O
C                                                                          *834800> 834800000 C-G
C--------------------------------------------------------------------------*834900> 834900000 C-G
C  IF NOT TIDAL THEN COMPUTE ETAD FROM ETAGRD AND ETA.                     *835000> 835000000 C-G
C--------------------------------------------------------------------------*835100> 835100000 C-G
C                                                                          *835200> 835200000 C-G
        DO 170 J=1,JMT                                                     *835300> 835300000 -G
        DO 170 I=1,IMT                                                     *835400> 835400000 -G
          ETAD(I,J) = EM(I,J)*( ETA(I,J) - DTBT*CSTR(J)*ETAGRD(I,J) )      *835500> 835500000 -G
 170    CONTINUE                                                           *835600> 835600000 -G
C                                                                          *835700> 835700000 S-G
        DO 175 I=1,IMT                                                     *835800> 835800000 S-G
          ETAD(I,JMT)=ETAD(I,JMTM1)                                        *835900> 835900000 S-G
 175    CONTINUE                                                           *836000> 836000000 S-G
C                                                                          *836100> 836100000 C-G
        DO 180 J=2,JMT                                                     *836200> 836200000 O-G
          ETAD(  1,J) = ETAD(IMTM1,J)                                      *836300> 836300000 O-G
          ETAD(IMT,J) = ETAD(2,J)                                          *836400> 836400000 O-G
 180    CONTINUE                                                           *836500> 836500000 O-G
 90   CONTINUE                                                             *836600> 836600000 -G
C                                                                          *836700> 836700000 C-G
C--------------------------------------------------------------------------*836800> 836800000 C-G
C IF NOT TIDAL THEN SET ETA EQUAL TO ETAD.                                 *836900> 836900000 C-G
C--------------------------------------------------------------------------*837000> 837000000 C-G
C                                                                          *837100> 837100000 C-G
      DO 200 J=1,JMT                                                       *837200> 837200000 -G
      DO 200 I=1,IMT                                                       *837300> 837300000 -G
        ETA(I,J)=ETAD(I,J)                                                 *837400> 837400000 -G
 200  CONTINUE                                                             *837500> 837500000 -G
C                                                                          *837600> 837600000 C
      ITTBT = (ITT-1)*NB + ITBT                                            *837700> 837700000
C                                                                          *837800> 837800000
C--------------------------------------------------------------------------*837900> 837900000
C  COMPUTE TOTAL RATE OF CHANGE OF KE FOR EXTERNAL MODE ON ENERGY T'STEP*838000> 838000000
C--------------------------------------------------------------------------*838100> 838100000
```

```
      IF(NERGY.EQ.1 .AND. MXP.NE.1) THEN
      DO 330 J=2,JMTM2
      DO 330 J=2,JMTM1
       FX=CS(J)*DYU(J)/(NB*DTBT)
       IF(J.EQ.JMTM1) FX=FX*0.5
         DO 330 I=2,IMUM1
           ENGEXT(1)=ENGEXT(1)+( UBT(I,J)*(UBT(I,J)*HR(I,J)-UT(I,J))
     *                +VBT(I,J)*(VBT(I,J)*HR(I,J)-VT(I,J)) )*FX*DXU(I)
 330  CONTINUE
      ENDIF
C
C-----------------------------------------------------------------------
C  PRINT OUT THE ENERGIES AND MASS ERROR AT GIVEN TIME STEPS.
C-----------------------------------------------------------------------
C
      IF ( MOD(ITTBT,NDW) .EQ. 0 ) THEN
C
      ETANRG = 0.
      DO 300 J=2,JMTM1
      DO 300 L=1,LSE(J)
      DO 300 I=ISE(J,L),IEE(J,L)
        ETANRG = ETANRG + DXT(I)*CST(J)*DYT(J)*ETA(I,J)**2
 300  CONTINUE
      ETANRG = 0.5*GRAV*ETANRG
C
      FX=1.0
      VELNRG = 0.
      DO 310 J=2,JMTM2
      DO 310 J=2,JMTM1
      IF (J.EQ.JMTM1) FX=0.5
      DO 310 L=1,LSU(J)
      DO 310 I=ISU(J,L),IEU(J,L)
        VELNRG = VELNRG
     >          + FX*DXU(I)*CS(J)*DYU(J)*HR(I,J)
     >            *( UBT(I,J)**2 + VBT(I,J)**2 )
 310  CONTINUE
      VELNRG = 0.5*VELNRG
C
      TOTNRG = ETANRG + VELNRG
C
      ERRMSS = 0.
      DO 320 J=2,JMTM1
      DO 320 L=1,LSE(J)
      DO 320 I=ISE(J,L),IEE(J,L)
        ERRMSS = ERRMSS + DXT(I)*CST(J)*DYT(J)*ETA(I,J)
 320  CONTINUE
C
      WRITE (8,900) ITTBT,VELNRG,ETANRG,TOTNRG,ERRMSS
C
      ENDIF
C
C-----------------------------------------------------------------------
C  PLOT FIELDS AT GIVEN TIME STEPS
C-----------------------------------------------------------------------
C
      IF ( MOD(ITTBT,NC) .EQ. 0 ) THEN
C
        ISTRT=1
        ISTOP=IMT
C
        SCL=0.
        WRITE (6,910) 'ETA SURFACE DISPLACEMENT  ',ITTBT
        CALL MATRIX(ETA,IMT,ISTRT,ISTOP,JMT,0,SCL)
C
```

```
         WRITE (6,910) 'UBT TRANSPORT  ',ITTBT
         CALL MATRIX(UBT,IMT,ISTRT,ISTOP,JMT,0,SCL)
C
         WRITE (6,910) 'VBT TRANSPORT  ',ITTBT
         CALL MATRIX(VBT,IMT,ISTRT,ISTOP,JMT,0,SCL)
C
C
      ENDIF
C
  500 CONTINUE
C
  900 FORMAT (1X,'AT TIME STEP ',I6,5X,1P,4E12.4)
  910 FORMAT (1X,A,'AT (BAROTROPIC) TIMESTEP',I7)
C
      RETURN
      END
```

L. Running models

The procedure for creating and running a model remains almost identical to the standard procedure, and is detailed in Cox. The occurrences of "??" in the base code remain unaltered. Precisely one extra parameter must be set in the new base code: the integer MSPL, which replaces NISLE. Arrays of this dimension are used to hold the I, J indices of land points needed for the free surface smoothing (and are not declared if the option is not used, when MSPL may take the value unity). The program will fault if this value is too small, and suggest a larger value.

The KNUDSEN procedure remains identical to the Cox version, and is not discussed here.

The NAMELIST input data file specifies the following:

CONTRL

NFIRST	= 0	the run uses restart data
	x	start from timestep x
NLAST	= y	stop after timestep y
NNERGY	= x	print fields every x timesteps
NTSI	= x	print single line of information every x timesteps
NMIX	= x	do a mixing timestep every x timesteps
NDW	= x	print single line of barotropic mode information every x barotropic timesteps
NWRITE	= x	save restart data every x timesteps
NA	= 0 / 1	don't / do save restart data at the end of the run
NB	= x	there are x barotropic timesteps for each baroclinic one
NC	= x	print barotropic fields every x timesteps

EDDY

AMF	=	horizontal mixing of momentum
AHF	=	horizontal mixing of heat, salinity, tracers
FKPMF	=	vertical mixing of momentum
FKPHF	=	vertical mixing of heat, salinity, tracers

TSTEPS

DTTSF	=	length of timestep on temp., sal., tracers
DTUVF	=	length of timestep on internal mode velocities
DTBTF	=	length of timestep on barotropic mode

PARMS

ACORF = x > 0 treat Coriolis term implicitly with forward timestep weighted by x and past timestep by (1.0 - x)

0 treat Coriolis term explicitly

WGHTF = weighting factor for the smoothing term on the eta field in TROPIC

NB. WGHTF DOES NOT APPEAR IN THE NAMELIST UNLESS THE 'D'OPTION IS SET

TSPROF

TINITF = initial values of temp., sal., tracers

Sample models

Two sample models have been generated. These are (a) a free surface version of Cox's model 1 (itself derived from Semtner's model), provided so the user may test his implementation; and (b) free surface and Cox versions of Cox model 2, provided for comparison between the models.

(a) Cox model 1

Model 1 is a version of Cox's model 1, consisting of a closed, rectangular basin with a western shelf and a small island. The options chosen are K, C (core contained, retain comments). Because the free surface model does not permit very long tracer timesteps, both tracer and baroclinic velocity steps have been made 2 hours, although the model will run with 6 hour tracer steps. The behaviour is extremely close to Cox's original, and no comparisons are made in the text. The tidal version (option G) gives almost identical results, but needs much shorter baroclinic timesteps.

The NAMELIST file specifies the following:

NFIRST=1	start from scratch
NLAST=42	stop after timestep 42
NNERGY=20	print fields every 20 timesteps
NTSI=1	print single line of info every timestep
NMIX=10	do a mixing timestep every 10 timesteps
NDW=40	print barotropic information every 40 barotropic timesteps
NB=40	take 40 barotropic steps per baroclinic step
NC=800	print out barotropic fields every 800 barotropic steps
AMF=1.E9	horizontal mixing of momentum
AHF=2.E7	horizontal mixing of heat, etc.
FKPMF=1.0	vertical mixing of momentum
FKPHF=1.0	vertical mixing of heat, etc.
DTTSF=7200.	length of timestep for heat, etc.
DTUVF=7200.	length of timestep for internal mode velocities
DTBTF=180.	length of timestep for barotropic fields
ACORF=0.6	weighting for semi-implicit Coriolis term in internal mode velocity calculation
TINITF= ...	initial values of temperature and salinity

UPDATES FOR MODEL 1:

```
OPT=K,C
      PARAMETER (IMT=25,JMT=20,KM=6,NT=2,LSEG=2,MSPL=15,LBC=2                   200000
      EB=.FALSE.                                                             107400000
      SWLDEG=16.0                                                            111100000
        DYT(J)=2.0                                                           111800000
        DXT(I)=2.0                                                           112900000
      DZ(1)=50.00E2                                                          114400000
      DZ(2)=200.0E2                                                          114400010
      DZ(3)=500.0E2                                                          114400020
      DZ(4)=800.0E2                                                          114400030
      DZ(5)=1000.0E2                                                         114400040
      DZ(6)=1200.0E2                                                         114400050
        FKMP(I,J)=KM                                                         133400000
      DO 702 J=2,JMTM1                                                       133500010
        FKMP(2,J)=2                                                          133500020
      DO 702 I=3,6                                                           133500030
        FKMP(I,J)=I-1                                                        133500040
 702  CONTINUE                                                               133500050
      DO 704 J=7,14                                                          133500060
        FKMP(14,J)=5                                                         133500070
        FKMP(15,J)=5                                                         133500080
 704  CONTINUE                                                               133500090
      DO 706 J=8,13                                                          133500100
        FKMP(14,J)=4                                                         133500110
        FKMP(14,J)=4                                                         133500120
 706  CONTINUE                                                               133500130
      DO 708 J=9,12                                                          133500140
      DO 708 I=13,16                                                         133500150
        FKMP(I,J)=4                                                          133500160
 708  CONTINUE                                                               133500170
      FKMP(14,10)=0                                                          133500180
      FKMP(15,10)=0                                                          133500190
      FKMP(14,11)=0                                                          133500200
      FKMP(15,11)=0                                                          133500210
      DO 710 J=1,JMT                                                         133500220
      DO 710 I=1,IMT                                                         133500230
        IF (I-J.GE.18) FKMP(I,J)=0                                           133500240
 710  CONTINUE                                                               133500250
      DO 712 J=1,JMT                                                         133500260
      DO 712 I=1,IMT                                                         133500270
        IF (I+J.GE.39) FKMP(I,J)=0                                           133500280
 712  CONTINUE                                                               133500290
        FX=-COS(PI*FLOAT(J-2)/FLOAT(JMT-4))                                  151900010
          WSX(I)=FX                                                          152400000
          WSY(I)=0.0                                                         152500000
          CALL STINIT ()                                                     210800000
      MTEST=MOD(J,7)                                                         249100000
      IPRT=IMT                                                               249400000
       W(I,KMP1)=FX                                                          320980000
       DO 250 K=1,KMM1                                                       321400000
       DO 255 K=1,KMM1                                                       322200000
C****************  SPECIAL PATCH FOR PRESCRIBED SURFACE VALUES ******        423500010
      FXA=SIN(2.*PI*FLOAT(J-2)/FLOAT(JMT-3))                                 423500020
      IF(NERGY.EQ.1 .AND. MXP.EQ.0) THEN                                     423500030
        FXB=CST(J)*DYT(J)*DZ(1)/C2DTTS                                       423500040
        DO 876 I=2,IMTM1                                                     423500050
          TTDTOT(5,1)=TTDTOT(5,1)                                            423500060
     *   +(27.-25.*FLOAT(J-2)/FLOAT(JMT-3)-TA(I,1,1))*FXB*DXT(I)*FM(I,1)     423500070
          TTDTOT(5,2)=TTDTOT(5,2)                                            423500080
     *   +(.0007*FXA-TA(I,1,2))*FXB*DXT(I)*FM(I,1)                           423500090
 876    CONTINUE                                                             423500100
      ENDIF                                                                  423500110
```

```
      DO 878 I=2,IMTM1                                                  423500120
        TA(I,1,1)=(27.-25.*FLOAT(J-2)/FLOAT(JMT-3))*FM(I,1)             423500130
        TA(I,1,2)=.0007*FXA*FM(I,1)                                     423500140
 878  CONTINUE                                                          423500150
C****************  END SPECIAL PATCH  ********************************   423500160
-600300000,601500000                                                            0
*CALL ONEDIM                                                            602200000
-603100000,604000000                                                            0
-604700000,606300000                                                            0
      DO 100 K=1,KM                                                     606300010
        FACTOR=5891.+ZDZZ(K)/1013.                                      606300020
      DO 100 I=1,IMT                                                    606300030
        RHO(I,K)=FACTOR+3000.*(SX(I,K)+.035)+(38.-0.375*TX(I,K))*TX(I,K) 606300040
 100  CONTINUE                                                          606300050
      DO 200 K=1,KM                                                     606300060
      DO 200 I=1,IMT                                                    606300070
        RHO(I,K)=(1779.5+(11.25-0.0745*TX(I,K))*TX(I,K)                 606300080
     *           -(3800.+10.*TX(I,K))*(SX(I,K)+.035))/RHO(I,K)          606300090
 200  CONTINUE                                                          606300110
      DO 300 K=1,KM                                                     606300120
      DO 300 I=1,IMT                                                    606300130
        RHO(I,K)=1.0/(0.698+RHO(I,K))-1.02                              606300140
 300  CONTINUE                                                          606300150
-607000000,608900000                                                            0
-609100000,609900000                                                            0
      KD=IND-1                                                          609900010
      DO 400 K=1,KM                                                     609900020
        L=K+MOD(K+KD,2)                                                 609900030
        FACTOR=5891.+ZDZZ(L)/1013.                                      609900040
      DO 400 I=1,IMT                                                    609900050
        RHO(I,K)=FACTOR+3000.*(SX(I,K)+.035)+(38.-0.375*TX(I,K))*TX(I,K) 609900060
 400  CONTINUE                                                          609900070
      DO 500 K=1,KM                                                     609900080
      DO 500 I=1,IMT                                                    609900090
        RHO(I,K)=(1779.5+(11.25-0.0745*TX(I,K))*TX(I,K)                 609900110
     *           -(3800.+10.*TX(I,K))*(SX(I,K)+.035))/RHO(I,K)          609900120
 500  CONTINUE                                                          609900130
      DO 600 K=1,KM                                                     609900140
      DO 600 I=1,IMT                                                    609900150
        RHO(I,K)=1.0/(0.698+RHO(I,K))-1.02                              609900160
 600  CONTINUE                                                          609900170
              ENTRY STINIT()                                            610400000
-610500000,616600000                                                            0
```

the free surface model updates were included here

DATA FOR MODEL 1 :

```
&CONTRL NFIRST=1,NLAST=42,NNERGY=20,NTSI=1,NMIX=10,NDW=40,NB=40,NC=800,&END
&EDDY AMF=1.E9,AHF=2.E7,FKPMF=1.0,FKPHF=1.0,&END
&TSTEPS DTTSF=7200.,DTUVF=7200.,DTBTF=180.,&END
&PARMS ACORF=0.6,&END
&TSPROF TINITF=4.,4.,4.,4.,4.,4.,.0349,.0349,.0349,.0349,.0349,.0349,&END
```

```
&CONTRL  NFIRST = 1,  NLAST = 42,  NNERGY = 20,  NMIX = 10,  NWRITE = 1000000,  NDW = 40,  NTSI = 1,  NA = 0,  NB = 40,
 NC = 800,  &END
&EDDY  AMF = 1000000000.,  AHF = 20000000.,  FKPMF = 1.,  FKPHF = 1.,  &END
&TSTEPS  DTTSF = 7200.,  DTUVF = 7200.,  DTBTF = 180.,  &END
&PARMS  ACORF = 0.6,  &END

 GRID BOX THICKNESS  'DZ'
  0.50000E+04  0.20000E+05  0.50000E+05  0.80000E+05  0.10000E+06  0.12000E+06

 GRID POINT SEPARATION  'DZZ'
  0.25000E+04  0.12500E+05  0.35000E+05  0.65000E+05  0.90000E+05  0.11000E+06  0.60000E+05

 DEPTH OF BOX BOTTOM  'ZDZ'
  0.50000E+04  0.25000E+05  0.75000E+05  0.15500E+06  0.25500E+06  0.37500E+06

 DEPTH OF GRID POINT  'ZDZZ'
  0.25000E+04  0.15000E+05  0.50000E+05  0.11500E+06  0.20500E+06  0.31500E+06  0.37500E+06

 LATITUDE OF T,S POINTS (RADIANS)  'PHIT'
  0.26180E+00  0.29671E+00  0.33161E+00  0.36652E+00  0.40143E+00  0.43633E+00  0.47124E+00  0.50615E+00  0.54105E+00  0.57596E+00
  0.61087E+00  0.64577E+00  0.68068E+00  0.71558E+00  0.75049E+00  0.78540E+00  0.82030E+00  0.85521E+00  0.89012E+00  0.92502E+00

 LATITUDE OF U,V POINTS (RADIANS)  'PHI'
  0.27925E+00  0.31416E+00  0.34907E+00  0.38397E+00  0.41888E+00  0.45379E+00  0.48869E+00  0.52360E+00  0.55851E+00  0.59341E+00
  0.62832E+00  0.66323E+00  0.69813E+00  0.73304E+00  0.76794E+00  0.80285E+00  0.83776E+00  0.87266E+00  0.90757E+00  0.94248E+00

 COSINE OF T,S LATITUDE  'CST'
  0.96593E+00  0.95630E+00  0.94552E+00  0.93358E+00  0.92050E+00  0.90631E+00  0.89101E+00  0.87462E+00  0.85717E+00  0.83867E+00
  0.81915E+00  0.79864E+00  0.77715E+00  0.75471E+00  0.73135E+00  0.70711E+00  0.68200E+00  0.65606E+00  0.62932E+00  0.60182E+00

 COSINE OF U,V LATITUDE  'CS'
  0.96126E+00  0.95106E+00  0.93969E+00  0.92718E+00  0.91355E+00  0.89879E+00  0.88295E+00  0.86603E+00  0.84805E+00  0.82904E+00
  0.80902E+00  0.78801E+00  0.76604E+00  0.74314E+00  0.71934E+00  0.69466E+00  0.66913E+00  0.64279E+00  0.61566E+00  0.58779E+00

 SINE OF U,V LATITUDE  'SINE'
  0.27564E+00  0.30902E+00  0.34202E+00  0.37461E+00  0.40674E+00  0.43837E+00  0.46947E+00  0.50000E+00  0.52992E+00  0.55919E+00
  0.58779E+00  0.61566E+00  0.64279E+00  0.66913E+00  0.69466E+00  0.71934E+00  0.74314E+00  0.76604E+00  0.78801E+00  0.80902E+00

DO LOOP INDICES FOR SURFACE DISPLACEMENT  (ETA)

      20 ...
      19 ...     2   19
      18 ...     2   20
      17 ...     2   21
      16 ...     2   22
      15 ...     2   23
      14 ...     2   24
      13 ...     2   24
      12 ...     2   24
      11 ...     2   13   16   24
      10 ...     2   13   16   24
       9 ...     2   24
       8 ...     2   24
       7 ...     2   24
       6 ...     2   23
       5 ...     2   22
       4 ...     2   21
       3 ...     2   20
       2 ...     2   19
       1 ...

DO LOOP INDICES FOR TRANSPORTS  (UBT,VBT)
```

20 ...				
19 ...				
18 ...	2	18		
17 ...	2	19		
16 ...	2	20		
15 ...	2	21		
14 ...	2	22		
13 ...	2	23		
12 ...	2	23		
11 ...	2	12	16	23
10 ...	2	12	16	23
9 ...	2	12	16	23
8 ...	2	23		
7 ...	2	23		
6 ...	2	22		
5 ...	2	21		
4 ...	2	20		
3 ...	2	19		
2 ...	2	18		
1 ...				

```
NUMBER OF LEVELS AT T,S POINTS AND U,V POINTS

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 2 2 3 4 5 6 6 6 6 6 6 6 6 6 6 6 6 6 0 0 0 0 0 0
 0 2 2 3 4 5 6 6 6 6 6 6 6 6 6 6 6 6 0 0 0 0 0 0
0 2 2 3 4 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 0 0 0 0 0
 0 2 2 3 4 5 6 6 6 6 6 6 6 6 6 6 6 6 6 0 0 0 0 0
0 2 2 3 4 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 0 0 0 0
 0 2 2 3 4 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 0 0 0 0
0 2 2 3 4 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 0 0 0
 0 2 2 3 4 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 0 0 0
0 2 2 3 4 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 0 0
 0 2 2 3 4 5 6 6 6 6 6 6 5 5 5 6 6 6 6 6 6 6 0 0
0 2 2 3 4 5 6 6 6 6 6 6 6 5 5 6 6 6 6 6 6 6 6 6 0
 0 2 2 3 4 5 6 6 6 6 6 6 4 4 5 6 6 6 6 6 6 6 6 0
0 2 2 3 4 5 6 6 6 6 6 6 6 4 5 6 6 6 6 6 6 6 6 6 0
 0 2 2 3 4 5 6 6 6 6 6 4 4 4 4 4 6 6 6 6 6 6 6 0
0 2 2 3 4 5 6 6 6 6 6 6 4 4 4 4 6 6 6 6 6 6 6 6 0
 0 2 2 3 4 5 6 6 6 6 6 4 0 0 0 4 6 6 6 6 6 6 6 0
0 2 2 3 4 5 6 6 6 6 6 6 4 0 0 4 6 6 6 6 6 6 6 6 0
 0 2 2 3 4 5 6 6 6 6 6 4 0 0 0 4 6 6 6 6 6 6 6 0
0 2 2 3 4 5 6 6 6 6 6 6 4 0 0 4 6 6 6 6 6 6 6 6 0
 0 2 2 3 4 5 6 6 6 6 6 4 0 0 0 4 6 6 6 6 6 6 6 0
0 2 2 3 4 5 6 6 6 6 6 6 4 4 4 4 6 6 6 6 6 6 6 6 0
 0 2 2 3 4 5 6 6 6 6 6 4 4 4 4 4 6 6 6 6 6 6 6 0
0 2 2 3 4 5 6 6 6 6 6 6 6 4 5 6 6 6 6 6 6 6 6 6 0
 0 2 2 3 4 5 6 6 6 6 6 6 4 4 5 6 6 6 6 6 6 6 6 0
0 2 2 3 4 5 6 6 6 6 6 6 6 5 5 6 6 6 6 6 6 6 6 6 0
 0 2 2 3 4 5 6 6 6 6 6 6 5 5 5 6 6 6 6 6 6 6 0 0
0 2 2 3 4 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 0 0
 0 2 2 3 4 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 0 0 0
0 2 2 3 4 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 0 0 0
 0 2 2 3 4 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 0 0 0 0
0 2 2 3 4 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 0 0 0 0
 0 2 2 3 4 5 6 6 6 6 6 6 6 6 6 6 6 6 6 0 0 0 0 0
0 2 2 3 4 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 0 0 0 0 0
 0 2 2 3 4 5 6 6 6 6 6 6 6 6 6 6 6 6 0 0 0 0 0 0
0 2 2 3 4 5 6 6 6 6 6 6 6 6 6 6 6 6 6 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

SURFACE AREA = 1.533836E+17      VOLUME = 4.623960E+22
&TSPROF  TINITF = 6*4.,  6*3.49E-2,  &END
TS=    1  YEAR=   0.00  DAY=  0.1  ENERGY= 0.000000E+00  DTEMP= 2.663940E-05  DSALT= 4.903325E-07
TS=    2  YEAR=   0.00  DAY=  0.2  ENERGY= 6.834550E-03  DTEMP= 1.332024E-05  DSALT= 2.451662E-07
TS=    3  YEAR=   0.00  DAY=  0.3  ENERGY= 1.367781E-01  DTEMP= 9.080669E-08  DSALT= 4.209106E-12
TS=    4  YEAR=   0.00  DAY=  0.3  ENERGY= 1.929144E-01  DTEMP= 8.421006E-08  DSALT= 4.121883E-12
TS=    5  YEAR=   0.00  DAY=  0.4  ENERGY= 1.645820E-01  DTEMP= 8.524273E-08  DSALT= 3.926785E-12
TS=    6  YEAR=   0.00  DAY=  0.5  ENERGY= 1.855238E-01  DTEMP= 8.451471E-08  DSALT= 3.947360E-12
TS=    7  YEAR=   0.00  DAY=  0.6  ENERGY= 2.235545E-01  DTEMP= 8.301165E-08  DSALT= 3.772135E-12
TS=    8  YEAR=   0.00  DAY=  0.7  ENERGY= 2.799450E-01  DTEMP= 7.868379E-08  DSALT= 3.625195E-12
TS=    9  YEAR=   0.00  DAY=  0.8  ENERGY= 2.176946E-01  DTEMP= 7.471603E-08  DSALT= 3.373430E-12
TS=   10  YEAR=   0.00  DAY=  0.8  ENERGY= 2.192573E-01  DTEMP= 7.480867E-08  DSALT= 3.456262E-12
TS=   11  YEAR=   0.00  DAY=  0.9  ENERGY= 2.302653E-01  DTEMP= 1.366557E-07  DSALT= 6.260671E-12
TS=   12  YEAR=   0.00  DAY=  1.0  ENERGY= 1.768676E-01  DTEMP= 7.093603E-08  DSALT= 3.270822E-12
TS=   13  YEAR=   0.00  DAY=  1.1  ENERGY= 1.788965E-01  DTEMP= 6.712104E-08  DSALT= 3.139483E-12
TS=   14  YEAR=   0.00  DAY=  1.2  ENERGY= 2.053338E-01  DTEMP= 6.985543E-08  DSALT= 3.238684E-12
TS=   15  YEAR=   0.00  DAY=  1.2  ENERGY= 1.484293E-01  DTEMP= 6.478933E-08  DSALT= 3.053215E-12
TS=   16  YEAR=   0.00  DAY=  1.3  ENERGY= 1.161641E-01  DTEMP= 6.688701E-08  DSALT= 3.026367E-12
TS=   17  YEAR=   0.00  DAY=  1.4  ENERGY= 1.552906E-01  DTEMP= 6.861778E-08  DSALT= 3.095475E-12
TS=   18  YEAR=   0.00  DAY=  1.5  ENERGY= 1.578846E-01  DTEMP= 6.404506E-08  DSALT= 3.034104E-12
TS=   19  YEAR=   0.00  DAY=  1.6  ENERGY= 1.264329E-01  DTEMP= 5.947462E-08  DSALT= 2.803348E-12
TEMPERATURE FOR J =    7 AT TIMESTEP     20
```

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65
2	4.01	4.02	4.02	4.02	4.02	4.02	4.02	4.02	4.02	4.02	4.01	4.01	4.01	4.00	4.00	4.00	4.00	4.01	4.01	4.01
3	0.00	0.00	3.99	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
4	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
5	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
6	0.00	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00

	22	23	24	25
1	19.65	19.65	19.65	0.00
2	4.02	4.01	4.02	0.00
3	4.00	4.00	4.00	0.00
4	4.00	4.00	4.00	0.00
5	4.00	4.00	4.00	0.00
6	4.00	4.00	4.00	0.00

SALINITY FOR J = 7 AT TIMESTEP 20

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67
2	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
3	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
4	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
5	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
6	10.00	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10

	22	23	24	25
1	0.67	0.67	0.67	10.00
2	-0.10	-0.10	-0.10	10.00
3	-0.10	-0.10	-0.10	10.00
4	-0.10	-0.10	-0.10	10.00
5	-0.10	-0.10	-0.10	10.00
6	-0.10	-0.10	-0.10	10.00

W VELOCITY FOR J = 7 AT TIMESTEP 20

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	2.64	0.92	0.61	-0.37	0.15	-0.12	0.10	0.06	0.14	0.19	0.27	0.23	0.40	0.17	0.31	0.12	0.17	0.14	0.11	0.23
2	2.21	0.58	0.80	-0.51	0.00	-0.29	-0.08	-0.12	-0.04	0.01	0.09	0.03	0.22	-0.01	0.14	-0.06	-0.01	-0.04	-0.06	0.07
3	0.00	0.00	2.32	-0.33	0.11	-0.25	-0.08	-0.12	-0.04	0.00	0.10	-0.03	0.20	0.00	0.21	-0.07	0.00	-0.04	-0.06	0.06
4	0.00	0.00	0.00	0.68	0.40	-0.15	-0.08	-0.10	-0.03	0.00	0.12	-0.16	0.15	0.02	0.38	-0.07	0.00	-0.03	-0.05	0.05
5	0.00	0.00	0.00	0.00	1.18	0.02	-0.08	-0.07	-0.03	-0.02	0.15	-0.38	0.08	0.07	0.66	-0.07	0.01	-0.02	-0.04	0.04
6	0.00	0.00	0.00	0.00	0.00	0.41	-0.07	-0.03	-0.02	-0.01	0.10	-0.89	0.00	0.00	1.04	-0.09	0.00	-0.01	-0.02	0.02
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

	22	23	24	25
1	0.12	0.31	0.12	0.00
2	-0.01	0.37	0.25	0.00
3	-0.01	0.36	0.23	0.00
4	0.00	0.32	0.20	0.00
5	0.00	0.25	0.15	0.00
6	0.00	0.14	0.08	0.00
7	0.00	0.00	0.00	0.00

U VELOCITY FOR J = 7 AT TIMESTEP 20

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0.00	-1.38	-0.68	0.15	0.68	0.62	0.79	0.67	0.77	0.69	0.79	0.73	0.80	0.76	0.78	0.69	0.63	0.69	0.71	0.73
2	0.00	-2.25	-1.54	-0.70	-0.17	-0.23	-0.06	-0.18	-0.07	-0.16	-0.06	-0.11	-0.04	-0.09	-0.07	-0.16	-0.22	-0.15	-0.13	-0.11
3	0.00	0.00	0.00	-0.90	-0.18	-0.23	-0.06	-0.18	-0.07	-0.16	-0.06	-0.11	-0.05	-0.10	-0.08	-0.17	-0.22	-0.15	-0.13	-0.11
4	0.00	0.00	0.00	0.00	-0.27	-0.24	-0.06	-0.18	-0.07	-0.15	-0.06	-0.11	-0.05	-0.11	-0.09	-0.17	-0.22	-0.15	-0.13	-0.11
5	0.00	0.00	0.00	0.00	0.00	-0.27	-0.06	-0.18	-0.07	-0.15	-0.05	-0.13	0.00	0.00	-0.13	-0.19	-0.22	-0.15	-0.13	-0.11
6	0.00	0.00	0.00	0.00	0.00	0.00	-0.08	-0.18	-0.07	-0.16	-0.05	-0.13	0.00	0.00	0.00	-0.20	-0.22	-0.16	-0.13	-0.11

	21	22	23	24	25
1	0.77	0.84	0.83	0.00	0.00
2	-0.10	-0.09	-0.07	0.00	0.00
3	-0.10	-0.09	-0.07	0.00	0.00
4	-0.10	-0.09	-0.07	0.00	0.00
5	-0.10	-0.08	-0.07	0.00	0.00
6	-0.10	-0.08	-0.07	0.00	0.00

V VELOCITY FOR J = 7 AT TIMESTEP 20

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0.00	2.21	1.68	2.27	2.87	2.70	2.71	2.59	2.59	2.54	2.53	2.59	2.61	2.38	2.32	2.29	2.19	2.21	2.27	2.28
2	0.00	-0.15	-0.82	-0.25	0.34	0.17	0.19	0.06	0.06	0.01	0.01	0.07	0.10	-0.13	-0.19	-0.23	-0.33	-0.31	-0.25	-0.24
3	0.00	0.00	0.00	-0.23	0.32	0.16	0.18	0.06	0.07	0.02	0.02	0.08	0.13	-0.11	-0.18	-0.22	-0.33	-0.31	-0.25	-0.25
4	0.00	0.00	0.00	0.00	0.31	0.16	0.18	0.06	0.07	0.02	0.02	0.09	0.14	-0.11	-0.18	-0.22	-0.33	-0.31	-0.25	-0.25
5	0.00	0.00	0.00	0.00	0.00	0.15	0.19	0.06	0.07	0.02	0.03	0.09	0.00	0.00	-0.17	-0.24	-0.34	-0.31	-0.25	-0.25
6	0.00	0.00	0.00	0.00	0.00	0.00	0.18	0.06	0.07	0.02	0.03	0.09	0.00	0.00	0.00	-0.24	-0.34	-0.31	-0.25	-0.25

	21	22	23	24	25
1	2.33	2.27	2.12	0.00	0.00
2	-0.19	-0.20	-0.11	0.00	0.00
3	-0.20	-0.21	-0.11	0.00	0.00
4	-0.20	-0.21	-0.11	0.00	0.00
5	-0.20	-0.21	-0.11	0.00	0.00
6	-0.20	-0.21	-0.11	0.00	0.00

TEMPERATURE FOR J = 14 AT TIMESTEP 20

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	9.35	9.35	9.35	9.35	9.35	9.35	9.35	9.35	9.35	9.35	9.35	9.35	9.35	9.35	9.35	9.35	9.35	9.35	9.35	9.35
2	4.00	4.01	4.00	4.00	4.00	4.00	4.01	4.01	4.01	4.01	4.01	4.01	4.01	4.01	4.01	4.00	4.00	4.00	4.00	4.01
3	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
4	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
5	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
6	0.00	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00

	22	23	24	25
1	9.35	9.35	9.35	0.00
2	4.01	4.01	4.01	0.00
3	4.00	4.00	4.00	0.00
4	4.00	4.00	4.00	0.00
5	4.00	4.00	4.00	0.00
6	4.00	4.00	4.00	0.00

SALINITY FOR J = 14 AT TIMESTEP 20

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	-0.67	-0.67	-0.67	-0.67	-0.67	-0.67	-0.67	-0.67	-0.67	-0.67	-0.67	-0.67	-0.67	-0.67	-0.67	-0.67	-0.67	-0.67	-0.67	-0.67
2	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
3	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
4	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
5	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
6	10.00	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10

	22	23	24	25
1	-0.67	-0.67	-0.67	10.00
2	-0.10	-0.10	-0.10	10.00
3	-0.10	-0.10	-0.10	10.00
4	-0.10	-0.10	-0.10	10.00
5	-0.10	-0.10	-0.10	10.00
6	-0.10	-0.10	-0.10	10.00

W VELOCITY FOR J = 14 AT TIMESTEP 20

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	-0.21	-0.54	0.64	0.37	0.53	0.21	0.30	0.00	0.03	-0.27	-0.20	-0.55	-0.52	-0.74	-0.74	-0.71	-0.80	-0.74	-0.77	-0.76
2	-0.11	-0.49	0.74	0.39	0.55	0.21	0.28	-0.02	0.01	-0.28	-0.21	-0.59	-0.50	-0.73	-0.74	-0.72	-0.80	-0.75	-0.77	-0.76
3	0.00	0.00	1.22	0.56	0.71	0.27	0.25	-0.02	0.01	-0.27	-0.19	-0.68	-0.37	-0.61	-0.65	-0.68	-0.76	-0.71	-0.73	-0.72
4	0.00	0.00	0.00	0.91	1.08	0.44	0.18	-0.02	0.01	-0.23	-0.12	-0.88	-0.03	-0.32	-0.43	-0.58	-0.65	-0.61	-0.63	-0.62
5	0.00	0.00	0.00	0.00	1.79	0.69	0.08	-0.02	0.01	-0.16	-0.01	-1.22	0.50	0.15	-0.08	-0.43	-0.47	-0.45	-0.46	-0.45
6	0.00	0.00	0.00	0.00	0.00	1.19	-0.05	-0.01	0.01	-0.08	0.01	-1.22	0.00	0.00	0.39	-0.23	-0.26	-0.24	-0.25	-0.25
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

	22	23	24	25
1	-0.70	-0.76	-0.58	0.00
2	-0.67	-0.66	-0.49	0.00
3	-0.63	-0.63	-0.47	0.00
4	-0.54	-0.54	-0.40	0.00
5	-0.40	-0.40	-0.30	0.00
6	-0.22	-0.22	-0.16	0.00
7	0.00	0.00	0.00	0.00

U VELOCITY FOR J = 14 AT TIMESTEP 20

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0.00	-0.51	0.28	-0.39	-0.19	-0.08	-0.08	-0.01	-0.07	-0.01	-0.05	0.04	-0.04	-0.02	-0.05	0.09	0.11	0.16	0.17	0.17
2	0.00	-0.85	0.12	-0.54	-0.34	-0.23	-0.23	-0.15	-0.21	-0.16	-0.20	-0.11	-0.19	-0.17	-0.20	-0.05	-0.03	0.01	0.02	0.01
3	0.00	0.00	0.00	-0.50	-0.33	-0.23	-0.23	-0.16	-0.22	-0.16	-0.20	-0.11	-0.19	-0.17	-0.20	-0.05	-0.03	0.01	0.02	0.01
4	0.00	0.00	0.00	0.00	-0.37	-0.23	-0.23	-0.16	-0.22	-0.16	-0.20	-0.11	-0.19	-0.17	-0.20	-0.05	-0.03	0.01	0.02	0.01
5	0.00	0.00	0.00	0.00	0.00	-0.26	-0.23	-0.16	-0.22	-0.16	-0.20	-0.11	-0.19	-0.17	-0.20	-0.05	-0.03	0.01	0.02	0.01
6	0.00	0.00	0.00	0.00	0.00	0.00	-0.26	-0.16	-0.22	-0.16	-0.20	-0.14	0.00	0.00	0.00	-0.05	-0.03	0.01	0.02	0.01

	21	22	23	24	25
1	0.27	0.40	0.00	0.00	0.00
2	0.06	-0.02	0.00	0.00	0.00
3	0.06	-0.02	0.00	0.00	0.00
4	0.06	-0.02	0.00	0.00	0.00
5	0.06	-0.02	0.00	0.00	0.00
6	0.06	-0.02	0.00	0.00	0.00

V VELOCITY FOR J = 14 AT TIMESTEP 20

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0.00	1.46	0.21	-0.70	-0.73	-1.09	-1.11	-1.29	-1.29	-1.41	-1.37	-1.44	-1.50	-1.47	-1.35	-1.39	-1.39	-1.46	-1.46	-1.47
2	0.00	2.78	1.55	0.65	0.62	0.26	0.24	0.06	0.06	-0.06	-0.02	-0.09	-0.15	-0.12	0.00	-0.04	-0.04	-0.11	-0.11	-0.12
3	0.00	0.00	0.00	0.65	0.64	0.26	0.23	0.06	0.06	-0.06	-0.02	-0.09	-0.15	-0.12	0.01	-0.04	-0.03	-0.11	-0.10	-0.12
4	0.00	0.00	0.00	0.00	0.63	0.26	0.24	0.06	0.06	-0.06	-0.02	-0.09	-0.15	-0.12	0.01	-0.04	-0.03	-0.11	-0.10	-0.12
5	0.00	0.00	0.00	0.00	0.00	0.25	0.23	0.06	0.06	-0.06	-0.02	-0.09	-0.14	-0.11	0.00	-0.04	-0.03	-0.11	-0.10	-0.12
6	0.00	0.00	0.00	0.00	0.00	0.00	0.21	0.05	0.06	-0.06	-0.02	-0.07	0.00	0.00	0.00	-0.03	-0.03	-0.11	-0.10	-0.12

	21	22	23	24	25
1	-1.45	-1.36	0.00	0.00	0.00
2	-0.11	-0.06	0.00	0.00	0.00
3	-0.11	-0.06	0.00	0.00	0.00
4	-0.11	-0.06	0.00	0.00	0.00
5	-0.11	-0.06	0.00	0.00	0.00
6	-0.11	-0.06	0.00	0.00	0.00

ETA SURFACE DISPLACEMENT AT (BAROTROPIC) TIMESTEP 800

	1	2	3	4	5	6	7	8	9	10
20	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
19	0.00000E+00	5.74829E+00	-1.23941E+00	-1.74145E+01	-1.50060E+01	-1.86130E+01	-1.74219E+01	-2.02926E+01	-1.98024E+01	-2.17873E+01
18	0.00000E+00	-3.48853E-02	-4.68688E+00	-3.36368E+00	-8.10428E+00	-5.74136E+00	-8.28044E+00	-8.94208E+00	-1.17280E+01	-1.15271E+01
17	0.00000E+00	-2.85887E+00	-2.57308E+00	-6.94706E+00	-3.31747E+00	-6.06837E+00	-5.10779E+00	-7.66466E+00	-7.43113E+00	-9.61363E+00
16	0.00000E+00	-1.25992E+01	-1.01805E+01	-4.46130E+00	-5.82210E+00	-3.22036E+00	-5.47829E+00	-4.61315E+00	-6.59605E+00	-6.07052E+00
15	0.00000E+00	-1.94445E+01	-9.86740E+00	-8.43797E+00	-3.50932E+00	-4.80129E+00	-2.91022E+00	-4.60364E+00	-3.72610E+00	-5.36132E+00
14	0.00000E+00	-1.70857E+01	-1.54664E+01	-5.44297E+00	-5.95802E+00	-2.64260E+00	-3.73885E+00	-2.39180E+00	-3.66144E+00	-2.89669E+00
13	0.00000E+00	-1.95752E+01	-8.19138E+00	-8.96506E+00	-3.26000E+00	-3.95580E+00	-2.03297E+00	-2.90397E+00	-1.98486E+00	-3.01843E+00
12	0.00000E+00	-6.89467E+00	-1.48202E+01	-3.46676E+00	-4.93050E+00	-2.29488E+00	-2.72492E+00	-1.75908E+00	-2.29194E+00	-1.73904E+00
11	0.00000E+00	-1.93457E+01	-1.27805E+00	-6.59105E+00	-2.68473E+00	-2.96320E+00	-1.90528E+00	-1.95318E+00	-1.52204E+00	-1.74503E+00
10	0.00000E+00	-3.48591E+00	-1.09404E+01	-2.98385E+00	-3.52970E+00	-2.38659E+00	-1.74390E+00	-1.56697E+00	-1.14333E+00	-1.07180E+00
9	0.00000E+00	-1.28053E+01	-4.35077E+00	-4.70152E+00	-3.31249E+00	-1.64250E+00	-1.69250E+00	-5.94941E-01	-8.11111E-01	4.18273E-02
8	0.00000E+00	-1.83332E+00	-6.67447E+00	-5.24170E+00	-1.57191E+00	-2.03966E+00	1.83430E-01	-4.68293E-01	1.06110E+00	6.07081E-01
7	0.00000E+00	-7.11278E-02	-6.79964E+00	-9.86602E-01	-2.84883E+00	1.12801E+00	1.48408E-01	2.42547E+00	1.62923E+00	3.49203E+00
6	0.00000E+00	-5.28942E+00	1.14177E+00	-4.31850E+00	2.24231E+00	7.69189E-01	4.31571E+00	3.10417E+00	5.53150E+00	4.66552E+00
5	0.00000E+00	-6.31735E+00	-6.99538E+00	2.55653E+00	1.26138E+00	6.51590E+00	5.19448E+00	8.20317E+00	7.02246E+00	9.50810E+00
4	0.00000E+00	-1.28914E+01	-7.20311E+00	1.16915E+00	8.62627E+00	7.66137E+00	1.15871E+01	1.00144E+01	1.29067E+01	1.18383E+01
3	0.00000E+00	-1.88351E+01	-3.10059E+00	8.35400E+00	1.01888E+01	1.51059E+01	1.36051E+01	1.68805E+01	1.55646E+01	1.83280E+01
2	0.00000E+00	-1.43166E+01	-9.72379E+00	9.31936E+00	1.69833E+01	1.65885E+01	2.10047E+01	1.94701E+01	2.25643E+01	2.14364E+01
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

	11	12	13	14	15	16	17	18	19	20
20	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
19	-2.16438E+01	-2.31090E+01	-2.32465E+01	-2.42207E+01	-2.45487E+01	-2.50657E+01	-2.55069E+01	-2.56772E+01	-2.59156E+01	0.00000E+00
18	-1.34194E+01	-1.32828E+01	-1.47075E+01	-1.47084E+01	-1.56924E+01	-1.58619E+01	-1.64252E+01	-1.66794E+01	-1.70212E+01	-1.70742E+01
17	-9.30479E+00	-1.10502E+01	-1.07729E+01	-1.21752E+01	-1.20030E+01	-1.29753E+01	-1.30033E+01	-1.36402E+01	-1.36972E+01	-1.42897E+01
16	-7.90549E+00	-7.43594E+00	-9.11729E+00	-8.62093E+00	-1.00069E+01	-9.68466E+00	-1.05920E+01	-1.04424E+01	-1.11310E+01	-1.07646E+01
15	-4.72321E+00	-6.41504E+00	-5.73470E+00	-7.51703E+00	-6.79081E+00	-8.06327E+00	-7.62379E+00	-8.39171E+00	-7.98605E+00	-8.83379E+00
14	-4.37739E+00	-3.59940E+00	-5.23917E+00	-4.44838E+00	-6.31034E+00	-5.49669E+00	-6.33200E+00	-5.79665E+00	-6.51644E+00	-5.83778E+00
13	-2.37310E+00	-3.93493E+00	-2.92259E+00	-4.94891E+00	-3.92761E+00	-5.20601E+00	-4.26851E+00	-4.79316E+00	-4.25187E+00	-5.08340E+00
12	-2.63043E+00	-2.07518E+00	-4.12312E+00	-2.86562E+00	-4.87597E+00	-3.23927E+00	-3.50347E+00	-3.11490E+00	-3.61676E+00	-2.91000E+00
11	-1.46395E+00	-2.21224E+00	-1.24118E+00	0.00000E+00	0.00000E+00	-1.85849E+00	-2.14630E+00	-2.20394E+00	-1.92351E+00	-2.67014E+00
10	-9.34325E-01	-8.89717E-01	-5.31327E-02	0.00000E+00	0.00000E+00	-1.22554E+00	-3.72952E-01	-8.18521E-01	-1.19118E+00	-6.63723E-01
9	-1.47686E-01	6.30374E-01	9.45281E-01	3.48109E+00	1.13516E+00	2.48115E+00	7.09629E-01	7.66860E-01	6.50337E-01	-1.06472E-01
8	1.94020E+00	1.53450E+00	3.39446E+00	2.41968E+00	4.38549E+00	2.67948E+00	3.39293E+00	2.50078E+00	2.09685E+00	2.53431E+00

7	2.86640E+00	4.68793E+00	3.82901E+00	6.22622E+00	4.96871E+00	6.02460E+00	4.94071E+00	4.95389E+00	4.90940E+00	4.06276E+00
6	6.76222E+00	5.97072E+00	7.95120E+00	7.19557E+00	8.71981E+00	7.97717E+00	8.27830E+00	7.97795E+00	7.49369E+00	8.03532E+00
5	8.62841E+00	1.07704E+01	1.00185E+01	1.17187E+01	1.11135E+01	1.19792E+01	1.16899E+01	1.15795E+01	1.18544E+01	1.09270E+01
4	1.42735E+01	1.34697E+01	1.54172E+01	1.48414E+01	1.60771E+01	1.58366E+01	1.61600E+01	1.63346E+01	1.58129E+01	1.63389E+01
3	1.73936E+01	1.96392E+01	1.90064E+01	2.05942E+01	2.03335E+01	2.10494E+01	2.12387E+01	2.11113E+01	2.14868E+01	2.06552E+01
2	2.39722E+01	2.32103E+01	2.51518E+01	2.47816E+01	2.59062E+01	2.60325E+01	2.62191E+01	2.67890E+01	2.65329E+01	0.00000E+00
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

	21	22	23	24	25
20	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
19	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
18	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
17	-1.38368E+01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
16	-1.16714E+01	-1.06065E+01	0.00000E+00	0.00000E+00	0.00000E+00
15	-8.00977E+00	-9.29237E+00	-7.58633E+00	0.00000E+00	0.00000E+00
14	-6.93102E+00	-5.64017E+00	-7.38870E+00	-5.04198E+00	0.00000E+00
13	-4.08090E+00	-5.51976E+00	-3.75946E+00	-5.96303E+00	0.00000E+00
12	-4.00922E+00	-2.64470E+00	-4.50821E+00	-2.51471E+00	0.00000E+00
11	-1.67492E+00	-3.14716E+00	-1.48770E+00	-3.62273E+00	0.00000E+00
10	-1.76708E+00	-4.50666E-01	-2.25391E+00	-3.77308E-01	0.00000E+00
9	8.08409E-01	-6.54091E-01	9.38584E-01	-1.06265E+00	0.00000E+00
8	1.37845E+00	2.64518E+00	9.04814E-01	2.70550E+00	0.00000E+00
7	4.96797E+00	3.45985E+00	4.93787E+00	3.03736E+00	0.00000E+00
6	6.77580E+00	7.95788E+00	6.21112E+00	0.00000E+00	0.00000E+00
5	1.17769E+01	1.01831E+01	0.00000E+00	0.00000E+00	0.00000E+00
4	1.50353E+01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
3	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
2	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

UBT TRANSPORT AT (BAROTROPIC) TIMESTEP 800

	1	2	3	4	5	6	7	8	9	10
20	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
19	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
18	0.00000E+00	-2.47153E+03	5.19926E+04	1.48852E+05	2.62048E+05	3.55416E+05	2.78408E+05	1.35970E+05	8.17548E+04	5.59852E+04
17	0.00000E+00	-1.18182E+04	-4.12229E+04	-4.75563E+04	-5.87110E+04	-9.61748E+04	-3.64342E+04	4.65237E+04	4.12955E+04	4.22345E+04
16	0.00000E+00	-5.33432E+04	-6.23238E+04	-9.86183E+04	-1.12275E+05	-1.10123E+05	-9.00424E+04	-4.95610E+04	-6.10154E+03	-7.62505E+03
15	0.00000E+00	-3.85098E+04	-8.10443E+04	-9.03191E+04	-1.18753E+05	-1.11401E+05	-8.61005E+04	-5.91417E+04	-5.27252E+04	-2.51347E+04
14	0.00000E+00	-3.98486E+04	-6.15034E+04	-9.22483E+04	-9.12321E+04	-1.00115E+05	-8.46396E+04	-7.28142E+04	-4.97134E+04	-6.36170E+04
13	0.00000E+00	-3.04637E+04	-6.56059E+04	-6.11486E+04	-8.48522E+04	-7.77950E+04	-7.90504E+04	-6.36522E+04	-7.06068E+04	-4.54192E+04
12	0.00000E+00	-3.90612E+04	-4.19742E+04	-6.97448E+04	-6.34154E+04	-7.85490E+04	-6.94062E+04	-7.43058E+04	-5.40049E+04	-7.00366E+04
11	0.00000E+00	-3.73723E+04	-6.58267E+04	-5.65206E+04	-7.82968E+04	-7.18203E+04	-8.07719E+04	-6.55881E+04	-7.30831E+04	-4.53236E+04
10	0.00000E+00	-3.14509E+04	-5.35063E+04	-8.25235E+04	-7.94273E+04	-9.17954E+04	-7.98671E+04	-8.69554E+04	-6.38163E+04	-6.54269E+04
9	0.00000E+00	-3.06654E+04	-7.62738E+04	-8.68823E+04	-1.02400E+05	-9.54390E+04	-1.05299E+05	-8.63100E+04	-8.90667E+04	-5.87860E+04
8	0.00000E+00	-4.06846E+04	-8.09467E+04	-1.01352E+05	-1.07429E+05	-1.20130E+05	-1.04889E+05	-1.13570E+05	-9.15276E+04	-8.69398E+04
7	0.00000E+00	-6.36739E+04	-8.36117E+04	-1.07774E+05	-1.22752E+05	-1.16966E+05	-1.31200E+05	-1.12389E+05	-1.17958E+05	-1.01103E+05
6	0.00000E+00	-5.42617E+04	-9.41567E+04	-1.05358E+05	-1.14643E+05	-1.30819E+05	-1.15899E+05	-1.31637E+05	-1.20216E+05	-1.26394E+05
5	0.00000E+00	-5.87100E+04	-6.30333E+04	-9.73323E+04	-1.06005E+05	-9.75847E+04	-1.22182E+05	-1.18132E+05	-1.33671E+05	-1.29821E+05
4	0.00000E+00	-9.03002E+03	-7.14958E+04	-5.76364E+04	-5.14416E+04	-7.74307E+04	-9.31895E+04	-1.25294E+05	-1.22397E+05	-1.31062E+05
3	0.00000E+00	-6.92543E+04	-2.37979E+04	-1.76294E+04	-4.40029E+04	-7.71631E+04	-1.11165E+05	-1.07831E+05	-1.18091E+05	-1.17115E+05
2	0.00000E+00	-4.05108E+04	-1.44142E+05	-1.60770E+05	-1.57744E+05	-1.45148E+05	-1.12659E+05	-1.06744E+05	-1.02166E+05	-1.09777E+05
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

	11	12	13	14	15	16	17	18	19	20
20	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
19	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
18	4.55882E+04	3.75497E+04	3.55992E+04	3.18714E+04	2.99993E+04	2.28716E+04	1.60367E+04	1.33287E+03	0.00000E+00	0.00000E+00
17	3.04658E+04	3.33484E+04	2.62133E+04	2.75671E+04	2.52206E+04	2.44220E+04	2.08284E+04	2.11879E+04	5.18029E+03	0.00000E+00

16	6.04582E+03	-5.75775E+03	4.39370E+03	3.40477E+03	6.98434E+03	1.26705E+04	1.82959E+04	1.79384E+04	2.77747E+04	9.45422E+03
15	-3.69759E+04	-1.89356E+04	-3.50122E+04	-2.48333E+04	-8.40767E+03	-2.27612E+03	7.59964E+03	2.07596E+04	1.73179E+04	3.15225E+04
14	-3.65730E+04	-5.12137E+04	-2.82169E+04	-3.13335E+04	-3.38448E+04	-1.04909E+04	4.25266E+03	7.90456E+03	2.37695E+04	1.53375E+04
13	-8.24962E+04	-5.97316E+04	-4.16221E+04	-1.77691E+04	-1.18502E+04	-1.05692E+04	1.06796E+03	1.13234E+04	4.30810E+03	2.17400E+04
12	-3.47632E+04	-5.25356E+04	-4.24586E+04	-2.52642E+04	1.28855E+03	5.92502E+03	-8.82879E+02	-3.62641E+03	1.14082E+04	-3.73011E+03
11	-6.00604E+04	-1.91200E+04	0.00000E+00	0.00000E+00	0.00000E+00	7.49307E+03	2.06869E+02	7.28281E+03	-1.42493E+04	5.21804E+03
10	-2.75431E+04	-7.46888E+02	0.00000E+00	0.00000E+00	0.00000E+00	3.43063E+03	1.68969E+04	-2.29953E+04	-2.86869E+03	-2.60784E+04
9	-3.51549E+04	-2.28374E+04	0.00000E+00	0.00000E+00	0.00000E+00	6.36186E+03	-3.49858E+04	-8.48031E+03	-4.29241E+04	-1.37602E+04
8	-5.93808E+04	-4.40664E+04	-6.96472E+04	-4.14826E+04	-4.44717E+04	-5.30305E+04	-1.89971E+04	-6.82587E+04	-2.99143E+04	-5.19452E+04
7	-1.05848E+05	-1.09606E+05	-5.16277E+04	-4.62721E+04	-5.08732E+04	-7.13778E+04	-1.12988E+05	-5.25791E+04	-7.34408E+04	-3.63683E+04
6	-1.24184E+05	-1.16395E+05	-1.41318E+05	-1.29371E+05	-1.08915E+05	-1.15700E+05	-8.69273E+04	-9.59630E+04	-5.41161E+04	-6.77204E+04
5	-1.32277E+05	-1.47132E+05	-1.60532E+05	-1.74370E+05	-1.76925E+05	-1.33508E+05	-1.07714E+05	-7.26973E+04	-7.65812E+04	-4.32886E+04
4	-1.34125E+05	-1.39356E+05	-1.45213E+05	-1.48113E+05	-1.34070E+05	-1.21903E+05	-8.94891E+04	-7.75780E+04	-4.99904E+04	-5.35892E+04
3	-1.23500E+05	-1.24080E+05	-1.25607E+05	-1.17860E+05	-1.11092E+05	-9.02271E+04	-7.46113E+04	-5.24502E+04	-4.51226E+04	0.00000E+00
2	-1.08606E+05	-1.11476E+05	-1.04695E+05	-1.00441E+05	-8.56014E+04	-7.28706E+04	-5.28576E+04	-2.91428E+04	0.00000E+00	0.00000E+00
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

	21	22	23	24	25
20	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
19	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
18	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
17	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
16	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
15	1.22843E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
14	3.05896E+04	1.20989E+04	0.00000E+00	0.00000E+00	0.00000E+00
13	8.97650E+03	2.37096E+04	7.52077E+03	0.00000E+00	0.00000E+00
12	1.43189E+04	-3.29872E+03	7.52808E+03	0.00000E+00	0.00000E+00
11	-1.34269E+04	7.28175E+03	-1.13251E+04	0.00000E+00	0.00000E+00
10	-7.51332E+02	-1.92947E+04	8.32772E+03	0.00000E+00	0.00000E+00
9	-3.29410E+04	-8.85600E+02	-1.98597E+04	0.00000E+00	0.00000E+00
8	-1.83140E+04	-3.42246E+04	5.91487E+03	0.00000E+00	0.00000E+00
7	-5.33258E+04	-1.81912E+04	-3.14169E+04	0.00000E+00	0.00000E+00
6	-3.28428E+04	-4.64163E+04	0.00000E+00	0.00000E+00	0.00000E+00
5	-5.39033E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
4	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
3	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
2	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

VBT TRANSPORT AT (BAROTROPIC) TIMESTEP 800

	1	2	3	4	5	6	7	8	9	10
20	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
19	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
18	0.00000E+00	5.85420E+03	2.39231E+04	8.13767E+04	6.98788E+04	3.23199E+04	-1.81572E+05	-8.21111E+04	-3.20871E+04	-4.81437E+04
17	0.00000E+00	6.08279E+03	9.90151E+04	1.32667E+05	1.42350E+05	2.86122E+04	-2.28606E+05	-1.35039E+05	-9.84755E+04	-4.18537E+04
16	0.00000E+00	5.63923E+04	4.69315E+04	1.34721E+05	9.83014E+04	-7.79424E+02	-8.29520E+04	-1.12684E+05	-7.21260E+04	-8.97433E+04
15	0.00000E+00	5.86680E+04	6.60442E+04	8.44152E+04	8.77568E+04	2.13538E+04	-5.57572E+04	-4.68058E+04	-8.29124E+04	-5.05208E+04
14	0.00000E+00	8.35311E+04	4.33703E+04	8.38126E+04	6.48988E+04	3.42155E+04	-1.39998E+04	-5.04255E+04	-4.51024E+04	-8.66006E+04
13	0.00000E+00	7.43692E+04	5.89465E+04	5.39375E+04	6.71345E+04	3.53258E+04	-4.97273E+03	-2.17788E+04	-6.24265E+04	-5.41618E+04
12	0.00000E+00	6.43456E+04	3.10248E+04	5.72253E+04	4.50199E+04	4.09307E+04	6.21857E+03	-2.44931E+04	-3.54153E+04	-7.27211E+04
11	0.00000E+00	5.10560E+04	3.78613E+04	2.80502E+04	5.09056E+04	3.08149E+04	1.21662E+04	-8.52387E+03	-3.73791E+04	-4.42419E+04
10	0.00000E+00	4.48361E+04	1.51090E+04	3.72727E+04	2.97615E+04	4.38726E+04	1.29403E+04	-1.66744E+03	-1.40847E+04	-3.68798E+04
9	0.00000E+00	3.40936E+04	2.43188E+04	1.42016E+04	5.04871E+04	3.12721E+04	2.85482E+04	8.84071E+03	-2.14087E+03	-2.61106E+03
8	0.00000E+00	2.82497E+04	7.17342E+03	3.79528E+04	2.72605E+04	5.55056E+04	2.51067E+04	2.38694E+04	1.70290E+04	1.51680E+04
7	0.00000E+00	3.01287E+04	2.61086E+04	6.12799E+03	5.70768E+04	3.82151E+04	4.61357E+04	2.71487E+04	2.88609E+04	3.65091E+04
6	0.00000E+00	4.67111E+04	-2.94557E+03	3.80739E+04	2.77479E+04	6.42579E+04	3.62690E+04	3.77327E+04	3.36056E+04	3.52473E+04
5	0.00000E+00	4.27275E+04	3.28787E+04	2.00583E+01	6.17519E+04	4.21487E+04	4.54072E+04	2.57367E+04	3.29393E+04	3.75216E+04
4	0.00000E+00	8.35309E+04	-1.23116E+04	5.50495E+04	3.41708E+04	4.66034E+04	8.02167E+03	2.29737E+04	2.62528E+04	2.56240E+04
3	0.00000E+00	5.58298E+04	8.74183E+04	2.02284E+04	3.93362E+04	-1.05418E+04	-2.06621E+03	6.48913E+03	1.91538E+04	2.28479E+04
2	0.00000E+00	5.55024E+04	2.18406E+04	5.74227E+04	-1.94297E+04	4.27236E+03	-1.46260E+04	9.67317E+03	7.88257E+03	6.83329E+03
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

	11	12	13	14	15	16	17	18	19	20
20	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
19	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
18	-8.29050E+02	-4.85684E+04	1.03357E+04	-4.97412E+04	9.16723E+03	-4.58675E+04	-2.83966E+03	-3.73542E+04	0.00000E+00	0.00000E+00
17	-7.17378E+04	-1.28667E+04	-6.82676E+04	-4.84651E+03	-6.58394E+04	-1.60201E+04	-6.50346E+04	-2.94271E+04	-4.45598E+04	0.00000E+00
16	-3.69596E+04	-8.26006E+04	-1.44462E+04	-7.69111E+04	-1.55387E+04	-7.10931E+04	-3.18596E+04	-6.84399E+04	-4.08849E+04	-3.93685E+04
15	-9.14719E+04	-3.29223E+04	-9.01322E+04	-6.65401E+03	-7.02300E+04	-2.12521E+04	-6.57521E+04	-3.84224E+04	-6.19390E+04	-4.50850E+04
14	-4.38553E+04	-9.31943E+04	-3.40612E+04	-6.92882E+04	-9.80693E+03	-5.08189E+04	-2.40834E+04	-5.98003E+04	-4.30359E+04	-5.41560E+04
13	-9.95470E+04	-2.80814E+04	-5.38597E+04	-2.75975E+04	-4.89902E+04	6.31832E+03	-4.74387E+04	-3.40119E+04	-5.45710E+04	-4.33470E+04
12	-7.35810E+04	-4.99157E+04	1.01997E+04	-3.62754E+04	-2.15619E+03	-2.87899E+04	-1.83595E+04	-5.02330E+04	-3.98021E+04	-5.11118E+04
11	-7.61703E+04	-3.88775E+03	0.00000E+00	0.00000E+00	0.00000E+00	-2.88484E+04	-3.49390E+04	-4.08785E+04	-5.52584E+04	-4.21455E+04
10	-4.09121E+04	4.14292E+04	0.00000E+00	0.00000E+00	0.00000E+00	-1.86979E+04	-5.37797E+04	-6.41596E+04	-4.75339E+04	-5.54515E+04
9	-4.45350E+03	6.49676E+04	0.00000E+00	0.00000E+00	0.00000E+00	-2.86944E+04	-8.37657E+04	-6.52702E+04	-6.79630E+04	-4.33916E+04
8	4.77402E+04	3.71071E+04	3.14467E+04	-1.80757E+04	1.47829E+03	-4.72363E+04	-9.01243E+04	-9.26108E+04	-5.17102E+04	-5.82313E+04
7	4.37280E+04	5.91280E+04	3.16192E+04	1.68055E+04	-5.79209E+04	-3.37910E+04	-1.23034E+05	-6.12604E+04	-7.20302E+04	-3.87004E+04
6	4.62145E+04	4.73527E+04	7.96822E+04	-2.78880E+04	-3.34521E+03	-1.31742E+05	-5.09951E+04	-8.56833E+04	-4.03599E+04	-5.48806E+04
5	3.07057E+04	5.45310E+04	1.34974E+04	3.07548E+04	-5.72620E+04	-4.61281E+04	-8.72013E+04	-3.40010E+04	-6.04859E+04	-3.12069E+04
4	3.58843E+04	1.07440E+04	3.19739E+04	-2.30263E+04	1.87618E+03	-5.79451E+04	-2.21375E+04	-5.78096E+04	-2.57685E+04	-4.46214E+04
3	1.04642E+04	2.51949E+04	-9.20760E+03	1.72677E+04	-3.20230E+04	3.35959E+03	-4.26228E+04	-1.24046E+04	-4.17666E+04	0.00000E+00
2	1.49473E+04	-3.93154E+03	1.90241E+04	-1.71309E+04	1.88416E+04	-2.54457E+04	1.46881E+04	-2.83769E+04	0.00000E+00	0.00000E+00
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

	21	22	23	24	25
20	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
19	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
18	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
17	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
16	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
15	-3.35424E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
14	-4.38334E+04	-2.69270E+04	0.00000E+00	0.00000E+00	0.00000E+00
13	-4.65020E+04	-3.89362E+04	-1.83170E+04	0.00000E+00	0.00000E+00
12	-4.22065E+04	-4.13359E+04	-3.21473E+04	0.00000E+00	0.00000E+00
11	-4.94599E+04	-4.16737E+04	-3.79848E+04	0.00000E+00	0.00000E+00
10	-4.17183E+04	-4.67790E+04	-3.48416E+04	0.00000E+00	0.00000E+00
9	-5.02672E+04	-3.79488E+04	-3.67978E+04	0.00000E+00	0.00000E+00
8	-3.63972E+04	-4.01071E+04	-2.82324E+04	0.00000E+00	0.00000E+00
7	-4.66741E+04	-2.80693E+04	-2.04618E+04	0.00000E+00	0.00000E+00
6	-3.06100E+04	-3.32938E+04	0.00000E+00	0.00000E+00	0.00000E+00
5	-4.08971E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
4	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
3	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
2	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

TS= 20 YEAR= 0.00 DAY= 1.7 ENERGY= 1.174412E-01 DTEMP= 5.795937E-08 DSALT= 2.842708E-12

WORK BY:	INTERNAL MODE	EXTERNAL MODE
TIME RATE OF CHANGE	1.344834E-06	5.158160E-06
HORIZONTAL ADVECTION	1.275295E-10	-2.331084E-10
VERTICAL ADVECTION	-3.190303E-10	1.209213E-10
HORIZONTAL FRICTION	-3.731929E-07	-3.197660E-07
VERTICAL FRICTION	-9.670263E-10	1.433362E-26
PRESSURE FORCES	1.993436E-06	2.125182E-06
IMPLICIT EFFECTS	-2.704566E-08	3.482673E-06
WORK BY WIND	-2.470726E-07	-1.298173E-07
BOTTOM DRAG	-1.319502E-10	0.000000E+00

	TEMPERATURE	SALINITY
TIME RATE OF CHANGE	-4.852940E-08	2.459556E-12
HORIZONTAL ADVECTION	-7.163774E-22	7.342366E-27
VERTICAL ADVECTION	-1.148949E-09	-2.512023E-13
HORIZONTAL DIFFUSION	3.393922E-23	-9.190848E-28
SURFACE FLUX	-4.738045E-08	2.710758E-12
TRUNCATION ERROR	-8.298805E-19	2.462155E-23
CHANGE OF VARIANCE	-3.606173E-07	-4.228310E-16

WORK BY BUOYANCY FORCES 4.118617E-06 ENERGY CONVERSION ERROR -3.373224E-17 NONLINEAR EXCHANGE ERROR -3.036880E-10

NORTHWARD TRANSPORT OF HEAT (X10**15 WATTS)

NORTHWARD TRANSPORT OF SALT (X10**10 CM**3/SEC)

	X MEAN	X EDDY	Z MEAN	Z EDDY	EKMAN	TOT ADV	DIFFUS	TOTAL	X MEAN	X EDDY	Z MEAN	Z EDDY	EKMAN	TOT ADV	DIFFUS	TOTAL
18	0.008	-0.004	0.003	-0.003	0.004	0.004	0.066	0.070	0.005	0.005	-0.001	-0.004	0.015	0.010	-0.099	-0.090
17	0.026	0.000	0.028	0.001	-0.002	0.026	0.008	0.034	0.017	0.000	-0.028	-0.020	0.065	0.017	-0.028	-0.011
16	0.029	0.000	0.042	0.005	-0.018	0.029	0.009	0.037	0.041	0.000	-0.044	-0.033	0.118	0.041	-0.022	0.020
15	0.004	0.000	0.032	0.008	-0.035	0.004	0.009	0.014	0.078	0.000	-0.038	-0.033	0.149	0.078	-0.011	0.068
14	-0.046	0.000	-0.004	0.007	-0.049	-0.046	0.010	-0.036	0.117	0.000	-0.011	-0.020	0.149	0.117	0.004	0.121
13	-0.109	0.000	-0.053	0.001	-0.057	-0.109	0.011	-0.098	0.139	0.000	0.026	-0.003	0.116	0.139	0.020	0.160
12	-0.150	0.000	-0.094	-0.004	-0.052	-0.150	0.011	-0.139	0.119	0.000	0.054	0.005	0.060	0.119	0.035	0.154
11	-0.165	0.000	-0.128	-0.007	-0.031	-0.165	0.010	-0.155	0.092	0.000	0.077	0.003	0.012	0.092	0.041	0.132
10	-0.139	0.000	-0.137	-0.002	0.000	-0.139	0.011	-0.129	0.083	0.000	0.083	-0.001	0.000	0.083	0.045	0.128
9	-0.082	0.000	-0.140	0.010	0.048	-0.082	0.011	-0.071	0.124	0.000	0.083	0.007	0.034	0.124	0.043	0.167
8	0.055	0.000	-0.124	0.054	0.126	0.055	0.012	0.068	0.249	0.000	0.069	0.054	0.126	0.249	0.038	0.287
7	0.234	0.000	-0.090	0.104	0.221	0.234	0.013	0.247	0.422	0.000	0.050	0.119	0.253	0.422	0.023	0.445
6	0.422	0.000	-0.027	0.127	0.322	0.422	0.012	0.434	0.546	0.000	0.032	0.146	0.369	0.546	0.005	0.551
5	0.555	0.000	0.035	0.089	0.431	0.555	0.012	0.567	0.551	0.000	0.021	0.090	0.440	0.551	-0.013	0.537
4	0.558	0.000	0.079	-0.064	0.544	0.558	0.012	0.570	0.397	0.000	0.012	-0.052	0.436	0.397	-0.029	0.368
3	0.428	0.000	0.076	-0.303	0.654	0.428	0.011	0.439	0.195	0.000	0.008	-0.161	0.347	0.195	-0.039	0.155
2	0.385	0.000	0.059	-0.431	0.757	0.385	0.008	0.393	0.077	0.000	-0.003	-0.105	0.184	0.077	-0.044	0.033

MERIDIONAL MASS TRANSPORT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0.00	3.92	4.58	6.45	7.30	6.68	5.16	3.14	1.16	-0.01	-0.94	-1.69	-2.05	-1.91	-1.67	-1.43	-1.27	-0.98	0.00	0.00
2	0.00	5.50	6.63	7.81	7.84	6.16	4.21	2.40	0.94	0.34	-0.56	-0.88	-1.10	-0.20	0.56	1.06	1.08	1.95	0.00	0.00
3	0.00	4.82	6.16	7.50	6.79	5.22	2.87	1.09	0.00	-0.50	-1.06	-1.31	-0.78	0.54	2.04	2.89	2.79	4.63	0.00	0.00
4	0.00	3.34	5.06	6.20	5.57	3.35	1.11	-1.05	-1.93	-2.01	-2.80	-2.19	-1.34	0.86	2.69	3.75	3.23	5.92	0.00	0.00
5	0.00	2.18	3.40	4.52	3.37	0.99	-1.80	-3.57	-4.58	-4.61	-4.74	-3.59	-1.99	0.21	2.52	3.27	2.65	4.70	0.00	0.00
6	0.00	1.19	1.73	2.17	0.57	-2.17	-5.25	-7.23	-8.34	-8.29	-7.76	-5.77	-3.37	-0.57	1.56	2.32	1.62	0.59	0.00	0.00

```
TS=   21  YEAR=   0.00  DAY=  1.7  ENERGY= 1.667378E-01  DTEMP= 1.097333E-07  DSALT= 5.370094E-12
TS=   22  YEAR=   0.01  DAY=  1.8  ENERGY= 2.137340E-01  DTEMP= 6.222088E-08  DSALT= 2.890371E-12
TS=   23  YEAR=   0.01  DAY=  1.9  ENERGY= 2.309532E-01  DTEMP= 5.913992E-08  DSALT= 2.771998E-12
TS=   24  YEAR=   0.01  DAY=  2.0  ENERGY= 2.277940E-01  DTEMP= 6.002398E-08  DSALT= 2.780265E-12
TS=   25  YEAR=   0.01  DAY=  2.1  ENERGY= 2.361318E-01  DTEMP= 5.950344E-08  DSALT= 2.786627E-12
TS=   26  YEAR=   0.01  DAY=  2.2  ENERGY= 2.271235E-01  DTEMP= 5.844495E-08  DSALT= 2.725285E-12
TS=   27  YEAR=   0.01  DAY=  2.2  ENERGY= 2.702772E-01  DTEMP= 6.081823E-08  DSALT= 2.811629E-12
TS=   28  YEAR=   0.01  DAY=  2.3  ENERGY= 2.808576E-01  DTEMP= 6.247479E-08  DSALT= 2.894565E-12
TS=   29  YEAR=   0.01  DAY=  2.4  ENERGY= 2.449890E-01  DTEMP= 5.782685E-08  DSALT= 2.697363E-12
TS=   30  YEAR=   0.01  DAY=  2.5  ENERGY= 2.522542E-01  DTEMP= 5.591028E-08  DSALT= 2.616980E-12
TS=   31  YEAR=   0.01  DAY=  2.6  ENERGY= 3.007651E-01  DTEMP= 1.043446E-07  DSALT= 4.967292E-12
TS=   32  YEAR=   0.01  DAY=  2.7  ENERGY= 3.260234E-01  DTEMP= 5.801645E-08  DSALT= 2.667549E-12
TS=   33  YEAR=   0.01  DAY=  2.7  ENERGY= 3.346493E-01  DTEMP= 5.538947E-08  DSALT= 2.570159E-12
TS=   34  YEAR=   0.01  DAY=  2.8  ENERGY= 3.154805E-01  DTEMP= 5.232523E-08  DSALT= 2.489015E-12
TS=   35  YEAR=   0.01  DAY=  2.9  ENERGY= 3.257777E-01  DTEMP= 5.510040E-08  DSALT= 2.594175E-12
TS=   36  YEAR=   0.01  DAY=  3.0  ENERGY= 3.080222E-01  DTEMP= 5.380062E-08  DSALT= 2.539254E-12
TS=   37  YEAR=   0.01  DAY=  3.1  ENERGY= 3.220920E-01  DTEMP= 5.414483E-08  DSALT= 2.510466E-12
TS=   38  YEAR=   0.01  DAY=  3.2  ENERGY= 3.539797E-01  DTEMP= 5.723722E-08  DSALT= 2.641021E-12
TS=   39  YEAR=   0.01  DAY=  3.2  ENERGY= 3.702835E-01  DTEMP= 5.333864E-08  DSALT= 2.518305E-12
```

TEMPERATURE FOR J = 7 AT TIMESTEP 40

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65	19.65
2	4.00	4.04	4.03	4.03	4.03	4.03	4.03	4.03	4.03	4.03	4.03	4.03	4.03	4.03	4.03	4.03	4.03	4.03	4.03	4.03
3	0.00	0.00	3.99	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
4	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
5	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
6	0.00	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00

	22	23	24	25
1	19.65	19.65	19.65	0.00
2	4.04	4.02	4.03	0.00
3	4.00	4.00	4.00	0.00

OUTPUT FROM MODEL 1 ON CHANNEL 8 , GIVING BAROTROPIC DATA:

AT TIME STEP	40	8.9780E+18	2.4539E+18	1.1432E+19	-2.4000E+01
AT TIME STEP	80	4.8995E+21	4.4421E+21	9.3417E+21	4.9920E+03
AT TIME STEP	120	6.3778E+21	1.1897E+22	1.8275E+22	6.1440E+03
AT TIME STEP	160	3.3449E+21	2.2555E+22	2.5900E+22	-3.1744E+04
AT TIME STEP	200	3.0433E+21	2.2279E+22	2.5322E+22	-3.0208E+04
AT TIME STEP	240	3.9193E+21	1.6528E+22	2.0447E+22	2.5600E+04
AT TIME STEP	280	5.4387E+21	1.2529E+22	1.7968E+22	-3.6608E+04
AT TIME STEP	320	3.0976E+21	1.3649E+22	1.6746E+22	-5.1200E+04
AT TIME STEP	360	2.3696E+21	1.3393E+22	1.5762E+22	-4.1984E+04
AT TIME STEP	400	4.3685E+21	6.4286E+21	1.0797E+22	-2.4576E+04
AT TIME STEP	440	2.5178E+21	4.7149E+21	7.2327E+21	-3.6096E+04
AT TIME STEP	480	3.2482E+21	5.3736E+21	8.6218E+21	-2.6624E+04
AT TIME STEP	520	5.3251E+21	1.0298E+22	1.5623E+22	-1.6384E+04
AT TIME STEP	560	3.1513E+21	2.0874E+22	2.4026E+22	7.1680E+03
AT TIME STEP	600	2.3904E+21	2.4522E+22	2.6912E+22	2.0992E+04
AT TIME STEP	640	4.3463E+21	1.9044E+22	2.3390E+22	4.6592E+04
AT TIME STEP	680	4.7244E+21	1.2208E+22	1.6932E+22	1.0496E+04
AT TIME STEP	720	3.0451E+21	9.4208E+21	1.2466E+22	7.6800E+03
AT TIME STEP	760	2.3674E+21	8.9926E+21	1.1360E+22	-4.2752E+04
AT TIME STEP	800	4.0250E+21	8.0473E+21	1.2072E+22	-8.9600E+03
AT TIME STEP	840	5.5809E+21	7.0590E+21	1.2640E+22	2.3296E+04
AT TIME STEP	880	5.6049E+21	8.7189E+21	1.4324E+22	1.7408E+04
AT TIME STEP	920	5.0628E+21	1.3531E+22	1.8594E+22	2.4832E+04
AT TIME STEP	960	4.6263E+21	1.9455E+22	2.4082E+22	4.1984E+04
AT TIME STEP	1000	4.1138E+21	2.2910E+22	2.7024E+22	4.5056E+04
AT TIME STEP	1040	5.5836E+21	1.9027E+22	2.4611E+22	7.6800E+03
AT TIME STEP	1080	6.1139E+21	1.2141E+22	1.8255E+22	3.5328E+04
AT TIME STEP	1120	4.2864E+21	8.6268E+21	1.2913E+22	4.1728E+04
AT TIME STEP	1160	4.7859E+21	7.2450E+21	1.2031E+22	4.8640E+04
AT TIME STEP	1200	6.9742E+21	7.7851E+21	1.4759E+22	4.0192E+04
AT TIME STEP	1240	8.3120E+21	1.0021E+22	1.8333E+22	1.9456E+04
AT TIME STEP	1280	8.5996E+21	1.2823E+22	2.1423E+22	3.6352E+04
AT TIME STEP	1320	8.0793E+21	1.5818E+22	2.3897E+22	1.0752E+04
AT TIME STEP	1360	8.0353E+21	1.8692E+22	2.6728E+22	1.8432E+04
AT TIME STEP	1400	7.4598E+21	2.0785E+22	2.8245E+22	2.4064E+04
AT TIME STEP	1440	7.4766E+21	1.9023E+22	2.6499E+22	1.5872E+04
AT TIME STEP	1480	8.8569E+21	1.2646E+22	2.1503E+22	-3.0720E+03
AT TIME STEP	1520	9.0486E+21	7.7646E+21	1.6813E+22	4.8640E+03
AT TIME STEP	1560	9.5227E+21	6.2951E+21	1.5818E+22	1.0752E+04
AT TIME STEP	1600	1.1052E+22	8.3346E+21	1.9387E+22	1.7152E+04
AT TIME STEP	1640	1.1910E+22	1.3055E+22	2.4965E+22	-4.6080E+03
AT TIME STEP	1680	1.2475E+22	1.7045E+22	2.9520E+22	1.7408E+04

(b) Cox model 2

Model 2 is a version of Cox's second model. This is a 15 level, rectangular basin with a western shelf, extending from 54°S to the equator, where the symmetry condition is applied, using the S option. A zonally telescoping grid is included. Note that the energetic contributions from the surface Newtonian conditions (Cox's updates 4272000010 to 427200070) lack a factor of BOXVOL; for consistency this factor is omitted in the free surface version given here.

The model is integrated in both free surface and Cox formulations for 10000 steps, again with tracer and internal mode velocity timesteps of 2 hours. Output from both runs is given below, for comparison. Again, the tidal version yields very similar output, and is not shown. The results of the free surface calculation show a sufficiently steady barotropic mode after 10000 steps that a streamfunction may be defined (since the time derivative in (21) becomes negligible). The streamfunction was computed by taking the divergence of the barotropic mode and solving a Poisson equation; although very steady, the small vertical velocity at the surface implied sufficient divergence that simple attempts to integrate V w.r.t. x yielded errors of up to 10 Sv. Figs. 2 and 3 show the free surface and Cox stremafunctions respectively.

The two figures are extremely similar, and almost identical over the flat bottom. The only area of difference is in the near-equatorial gyre over the topography, where the different way of handling the topographic terms gives the answers differing by a few sverdrups.

```
UPDATES FOR MODEL 2:

OPT=C,K,S
      PARAMETER (IMT=25,JMT=20,KM=15,NT=2,LSEG=1,MSPL=20,LBC=2          200000
      SWLDEG=-54.0                                                   111100000
        DYT(J)=3.0                                                   111800000
        DXT(I)=2.0                                                   112900000
        IF(I.GT.8)DXT(I)=2.5-0.5*COS(PI*(I-8)/16.)                   112900010
      DZ(1)=30.00E2                                                  114400000
      DZ(2)=46.30E2                                                  114400010
      DZ(3)=67.450E2                                                 114400020
      DZ(4)=94.460E2                                                 114400030
      DZ(5)=128.46E2                                                 114400040
      DZ(6)=170.570E2                                                114400050
      DZ(7)=221.91E2                                                 114400060
      DZ(8)=283.51E2                                                 114400070
      DZ(9)=356.21E2                                                 114400080
      DZ(10)=440.57E2                                                114400090
      DZ(11)=536.78E2                                                114400100
      DZ(12)=644.54E2                                                114400110
      DZ(13)=763.00E2                                                114400120
      DZ(14)=890.71E2                                                114400130
      DZ(15)=1025.53E2                                               114400140
      DO 700 J=2,JMT                                                 133200000
        FKMP(I,J)=KM                                                 133400000
      DO 702 J=2,JMT                                                 133500010
      DO 702 I=2,7                                                   133500020
        FKMP(I,J)=2*I                                                133500030
 702  CONTINUE                                                       133500040
        FX=(-SIN(PI*FLOAT(J-2)/FLOAT(JMT-4)))*2.                     151900010
          WSX(I)=FX                                                  152400000
          WSY(I)=0.0                                                 152500000
          CALL STINIT()                                              210800000
      MTEST=MOD(J,7)                                                 249100000
      IPRT=20                                                        249400000
       W(I,KMP1) = FX                                                320980000
       DO 250 K=1,KMM1                                               321400000
       DO 255 K=1,KMM1                                               322200000
C  ASSUME 10 DEGREE TURNING ANGLE AT BOTTOM BOUNDARY                 328900010
C                                                                    328900020
      CD=1.3E-3                                                      328900030
      FXA=CD/FKPM                                                    328900040
        FXB=FXA*SQRT(UB(I,KZ)**2+VB(I,KZ)**2)*DZZ(KZ+1)              329100010
        UDIF(I,KZ+1)=UB(I,KZ)-FXB*(UB(I,KZ)*.98481-VB(I,KZ)*.17365)  329200000
        VDIF(I,KZ+1)=VB(I,KZ)-FXB*(UB(I,KZ)*.17365+VB(I,KZ)*.98481)  329300000
C                                                                    420100010
C-----------------------------------------------------------------  420100020
C  SET NEWTONIAN SURFACE BOUNDARY CONDITION                          420100030
C-----------------------------------------------------------------  420100040
C                                                                    420100050
      FXA=25.0                                                       420100060
      FXB=20.0                                                       420100070
      IF(M.EQ.2) THEN                                                420100080
        FXA=0.0                                                      420100090
        FXB=-.001                                                    420100100
      ENDIF                                                          420100110
      DO 846 I=2,IMTM1                                               420100120
        TA(I,1,M)=TA(I,1,M)+2.315E-7*(FXA+FXB*PHIT(J)-TB(I,1,M))     420100130
 846  CONTINUE                                                       420100140
C                                                                    427200010
C  ADD IN CONTRIBUTION FROM NEWTONIAN BOUNDARY CONDITION             427200020
C                                                                    427200030
```

```
      IF(M.EQ.1) TTDTOT(5,1)=TTDTOT(5,1)                                427200040
     *                    +2.315E-7*(25.0+20.0*PHIT(J)-TB(I,1,1))       427200050
      IF(M.EQ.2) TTDTOT(5,2)=TTDTOT(5,2)                                427200060
     *                    +2.315E-7*(     -.001*PHIT(J)-TB(I,1,2))      427200070
      DATA TO/                                                          603800000
     *      13.4979166,13.4926065,13.4846595,13.4732852, 8.4679890,     603800010
     *       8.4517059, 5.9421330, 4.4281250, 3.9040599, 2.8802061,     603800020
     *       2.8360754, 2.7780657, 2.7025290, 2.6052822, 2.9625234/     603800030
      DATA SO/                                                          603900000
     *      -0.0022500,-0.0022500,-0.0022500,-0.0022500, 0.0001500,     603900010
     *       0.0001500,-0.0001000,-0.0001000,-0.0002500,-0.0002000,     603900020
     *      -0.0002000,-0.0002000,-0.0002000,-0.0002000,-0.0002000/     603900030
      DATA (C( 1,N),N=1,9)/                                             604000000
     *       -.2019152E-03,0.7709387E+00,-.4915766E-05,-.2007672E-02,   604000010
     *       0.4496149E+00,0.3652747E-07,0.4725372E-02,0.3768380E-04,   604000020
     *       0.6548196E+01/                                             604000030
      DATA (C( 2,N),N=1,9)/                                             604000040
     *       -.2026279E-03,0.7706838E+00,-.4903913E-05,-.2004045E-02,   604000050
     *       0.4497566E+00,0.3639690E-07,0.4712010E-02,0.3761764E-04,   604000060
     *       0.6546772E+01/                                             604000070
      DATA (C( 3,N),N=1,9)/                                             604000080
     *       -.2036885E-03,0.7703042E+00,-.4886267E-05,-.1998629E-02,   604000090
     *       0.4499598E+00,0.3620095E-07,0.4692196E-02,0.3751913E-04,   604000100
     *       0.6544648E+01/                                             604000110
      DATA (C( 4,N),N=1,9)/                                             604000120
     *       -.2051946E-03,0.7697646E+00,-.4861205E-05,-.1990901E-02,   604000130
     *       0.4502325E+00,0.3591934E-07,0.4664216E-02,0.3737920E-04,   604000140
     *       0.6541624E+01/                                             604000150
      DATA (C( 5,N),N=1,9)/                                             604000160
     *       -.1632162E-03,0.7810057E+00,-.5277472E-05,-.2322824E-02,   604000170
     *       0.7159905E+00,0.4943163E-07,0.5148399E-02,0.3859627E-04,   604000180
     *       0.6612560E+01/                                             604000190
      DATA (C( 6,N),N=1,9)/                                             604000200
     *       -.1664980E-03,0.7799287E+00,-.5223968E-05,-.2305658E-02,   604000210
     *       0.6964293E+00,0.4903556E-07,0.5108021E-02,0.3838043E-04,   604000220
     *       0.6607319E+01/                                             604000230
      DATA (C( 7,N),N=1,9)/                                             604000240
     *       -.1440378E-03,0.7842682E+00,-.5519369E-05,-.2473774E-02,   604000250
     *       0.5910575E+00,0.5793752E-07,0.6154590E-02,0.3926047E-04,   604000260
     *       0.6638472E+01/                                             604000270
      DATA (C( 8,N),N=1,9)/                                             604000280
     *       -.1336783E-03,0.7860867E+00,-.5672235E-05,-.2557069E-02,   604000290
     *       0.3528398E+00,0.6399606E-07,0.7004312E-02,0.3962390E-04,   604000300
     *       0.6655343E+01/                                             604000310
      DATA (C( 9,N),N=1,9)/                                             604000320
     *       -.1357195E-03,0.7848000E+00,-.5628947E-05,-.2554850E-02,   604000330
     *       0.1057585E+00,0.6555468E-07,0.7215849E-02,0.3964243E-04,   604000340
     *       0.6652038E+01/                                             604000350
      DATA (C(10,N),N=1,9)/                                             604000360
     *       -.1348395E-03,0.7842621E+00,-.5657959E-05,-.2578544E-02,   604000370
     *       -.6064741E+00,0.7002210E-07,0.7931478E-02,0.3974567E-04,   604000380
     *       0.6657214E+01/                                             604000390
      DATA (C(11,N),N=1,9)/                                             604000400
     *       -.1472340E-03,0.7804559E+00,-.5461105E-05,-.2512906E-02,   604000410
     *       -.2829495E+01,0.6901463E-07,0.7878453E-02,0.3931736E-04,   604000420
     *       0.6640513E+01/                                             604000430
      DATA (C(12,N),N=1,9)/                                             604000440
     *       -.1619146E-03,0.7759107E+00,-.5224071E-05,-.2433677E-02,   604000450
     *       -.4372853E+01,0.6757406E-07,0.7827411E-02,0.3881627E-04,   604000460
     *       0.6620337E+01/                                             604000470
      DATA (C(13,N),N=1,9)/                                             604000480
     *       -.1789724E-03,0.7705813E+00,-.4943833E-05,-.2339711E-02,   604000490
     *       -.6454825E+01,0.6554043E-07,0.7786706E-02,0.3824815E-04,   604000500
     *       0.6596465E+01/                                             604000510
```

```
      DATA (C(14,N),N=1,9)/                                              604000520
     *          -.1984058E-03,0.7644511E+00,-.4619129E-05,-.2230396E-02, 604000530
     *          -.9215148E+01,0.6271601E-07,0.7769342E-02,0.3762998E-04, 604000540
     *          0.6568817E+01/                                           604000550
      DATA (C(15,N),N=1,9)/                                              604000560
     *          -.2240840E-03,0.7565406E+00,-.4162414E-05,-.2069851E-02, 604000570
     *          -.9645579E+01,0.5543124E-07,0.7290867E-02,0.3652558E-04, 604000580
     *          0.6526393E+01/                                           604000590
                ENTRY STINIT()                                           610400000
```

the free surface model updates were included here

DATA FOR MODEL 2:

```
&CONTRL NFIRST=1,NLAST=10000,NNERGY=10000,NTSI=100,NMIX=20,NB=40,NDW=4000,NC=400000,NWRITE=30,&END
&EDDY AMF=1.E9,AHF=2.E7,FKPMF=1.0,FKPHF=1.0,&END
&TSTEPS DTTSF=7200.,DTUVF=7200.,DTBTF=180.,&END
&PARMS ACORF=0.6,&END
&TSPROF TINITF=4.,4.,4.,4.,4.,4.,4.,4.,4.,4.,4.,4.,4.,4.,4.,
.0349,.0349,.0349,.0349,.0349,.0349,.0349,.0349,.0349,.0349,.0349,.0349,.0349,.0349,.0349,&END
```

```
FREE SURFACE MODEL RESULTS FOR MODEL 2:

&CONTRL  NFIRST = 1,  NLAST = 10000,  NNERGY = 10000,  NMIX = 20,  NWRITE = 1000000,  NDW = 4000,  NTSI = 100,  NA = 0,  NB = 40,
 NC = 400000,  &END
&EDDY  AMF = 1000000000.,  AHF = 20000000.,  FKPMF = 1.,  FKPHF = 1.,  &END
&TSTEPS  DTTSF = 7200.,  DTUVF = 7200.,  DTBTF = 180.,  &END
&PARMS  ACORF = 0.6,  &END

 GRID BOX THICKNESS  'DZ'
  0.30000E+04  0.46300E+04  0.67450E+04  0.94460E+04  0.12846E+05  0.17057E+05  0.22191E+05  0.28351E+05  0.35621E+05  0.44057E+05
  0.53678E+05  0.64454E+05  0.76300E+05  0.89071E+05  0.10255E+06

 GRID POINT SEPARATION  'DZZ'
  0.15000E+04  0.38150E+04  0.56875E+04  0.80955E+04  0.11146E+05  0.14951E+05  0.19624E+05  0.25271E+05  0.31986E+05  0.39839E+05
  0.48867E+05  0.59066E+05  0.70377E+05  0.82685E+05  0.95812E+05  0.51276E+05

 DEPTH OF BOX BOTTOM  'ZDZ'
  0.30000E+04  0.76300E+04  0.14375E+05  0.23821E+05  0.36667E+05  0.53724E+05  0.75915E+05  0.10427E+06  0.13989E+06  0.18394E+06
  0.23762E+06  0.30208E+06  0.37838E+06  0.46745E+06  0.57000E+06

 DEPTH OF GRID POINT  'ZDZZ'
  0.15000E+04  0.53150E+04  0.11002E+05  0.19098E+05  0.30244E+05  0.45195E+05  0.64820E+05  0.90090E+05  0.12208E+06  0.16192E+06
  0.21078E+06  0.26985E+06  0.34023E+06  0.42291E+06  0.51872E+06  0.57000E+06

 LATITUDE OF T,S POINTS (RADIANS)  'PHIT'
 -0.96866E+00 -0.91630E+00 -0.86394E+00 -0.81158E+00 -0.75922E+00 -0.70686E+00 -0.65450E+00 -0.60214E+00 -0.54978E+00 -0.49742E+00
 -0.44506E+00 -0.39270E+00 -0.34034E+00 -0.28798E+00 -0.23562E+00 -0.18326E+00 -0.13090E+00 -0.78540E-01 -0.26180E-01  0.26180E-01

 LATITUDE OF U,V POINTS (RADIANS)  'PHI'
 -0.94248E+00 -0.89012E+00 -0.83776E+00 -0.78540E+00 -0.73304E+00 -0.68068E+00 -0.62832E+00 -0.57596E+00 -0.52360E+00 -0.47124E+00
 -0.41888E+00 -0.36652E+00 -0.31416E+00 -0.26180E+00 -0.20944E+00 -0.15708E+00 -0.10472E+00 -0.52360E-01 -0.29088E-13  0.52360E-01

 COSINE OF T,S LATITUDE  'CST'
  0.56641E+00  0.60876E+00  0.64945E+00  0.68835E+00  0.72537E+00  0.76041E+00  0.79335E+00  0.82413E+00  0.85264E+00  0.87882E+00
  0.90259E+00  0.92388E+00  0.94264E+00  0.95882E+00  0.97237E+00  0.98325E+00  0.99144E+00  0.99692E+00  0.99966E+00  0.99966E+00

 COSINE OF U,V LATITUDE  'CS'
  0.58779E+00  0.62932E+00  0.66913E+00  0.70711E+00  0.74314E+00  0.77715E+00  0.80902E+00  0.83867E+00  0.86603E+00  0.89101E+00
  0.91355E+00  0.93358E+00  0.95106E+00  0.96593E+00  0.97815E+00  0.98769E+00  0.99452E+00  0.99863E+00  0.10000E+01  0.99863E+00

 SINE OF U,V LATITUDE  'SINE'
 -0.80902E+00 -0.77715E+00 -0.74314E+00 -0.70711E+00 -0.66913E+00 -0.62932E+00 -0.58779E+00 -0.54464E+00 -0.50000E+00 -0.45399E+00
 -0.40674E+00 -0.35837E+00 -0.30902E+00 -0.25882E+00 -0.20791E+00 -0.15643E+00 -0.10453E+00 -0.52336E-01 -0.29088E-13  0.52336E-01

DO LOOP INDICES FOR SURFACE DISPLACEMENT  (ETA)

      20 ...
      19 ...     2   24
      18 ...     2   24
      17 ...     2   24
      16 ...     2   24
      15 ...     2   24
      14 ...     2   24
      13 ...     2   24
      12 ...     2   24
      11 ...     2   24
      10 ...     2   24
       9 ...     2   24
       8 ...     2   24
       7 ...     2   24
       6 ...     2   24
       5 ...     2   24
       4 ...     2   24
       3 ...     2   24
```

2	...	2	24
1	...		

DO LOOP INDICES FOR TRANSPORTS (UBT,VBT)

20	...		
19	...	2	23
18	...	2	23
17	...	2	23
16	...	2	23
15	...	2	23
14	...	2	23
13	...	2	23
12	...	2	23
11	...	2	23
10	...	2	23
9	...	2	23
8	...	2	23
7	...	2	23
6	...	2	23
5	...	2	23
4	...	2	23
3	...	2	23
2	...	2	23
1	...		

NUMBER OF LEVELS AT T,S POINTS AND U,V POINTS

```
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
SURFACE AREA = 3.122914E+17      VOLUME = 1.518226E+23
&TSPROF  TINITF = 15*4.,  15*3.49E-2,  &END
TS=   100  YEAR=   0.02  DAY=  8.3  ENERGY= 5.460233E-01  DTEMP= 1.540568E-08  DSALT= 6.841918E-13
TS=   200  YEAR=   0.05  DAY= 16.7  ENERGY= 6.796510E-01  DTEMP= 1.362368E-08  DSALT= 6.222023E-13
TS=   300  YEAR=   0.07  DAY= 25.0  ENERGY= 7.271109E-01  DTEMP= 1.207734E-08  DSALT= 5.737112E-13
TS=   400  YEAR=   0.09  DAY= 33.3  ENERGY= 7.634732E-01  DTEMP= 1.079706E-08  DSALT= 5.349307E-13
TS=   500  YEAR=   0.11  DAY= 41.7  ENERGY= 8.036156E-01  DTEMP= 9.714697E-09  DSALT= 4.973754E-13
TS=   600  YEAR=   0.14  DAY= 50.0  ENERGY= 8.437468E-01  DTEMP= 8.796088E-09  DSALT= 4.642919E-13
TS=   700  YEAR=   0.16  DAY= 58.3  ENERGY= 8.796686E-01  DTEMP= 8.051221E-09  DSALT= 4.371128E-13
TS=   800  YEAR=   0.18  DAY= 66.7  ENERGY= 9.084941E-01  DTEMP= 7.447661E-09  DSALT= 4.154260E-13
TS=   900  YEAR=   0.21  DAY= 75.0  ENERGY= 9.292678E-01  DTEMP= 6.958000E-09  DSALT= 3.996712E-13
TS=  1000  YEAR=   0.23  DAY= 83.3  ENERGY= 9.425608E-01  DTEMP= 6.549083E-09  DSALT= 3.858755E-13
TS=  1100  YEAR=   0.25  DAY= 91.7  ENERGY= 9.491005E-01  DTEMP= 6.209508E-09  DSALT= 3.743385E-13
TS=  1200  YEAR=   0.27  DAY=100.0  ENERGY= 9.499292E-01  DTEMP= 5.930132E-09  DSALT= 3.645769E-13
TS=  1300  YEAR=   0.30  DAY=108.3  ENERGY= 9.467246E-01  DTEMP= 5.713605E-09  DSALT= 3.569918E-13
TS=  1400  YEAR=   0.32  DAY=116.7  ENERGY= 9.407818E-01  DTEMP= 5.527468E-09  DSALT= 3.498689E-13
TS=  1500  YEAR=   0.34  DAY=125.0  ENERGY= 9.332621E-01  DTEMP= 5.371494E-09  DSALT= 3.468742E-13
TS=  1600  YEAR=   0.37  DAY=133.3  ENERGY= 9.246637E-01  DTEMP= 5.217004E-09  DSALT= 3.424110E-13
TS=  1700  YEAR=   0.39  DAY=141.7  ENERGY= 9.154165E-01  DTEMP= 5.081245E-09  DSALT= 3.391217E-13
TS=  1800  YEAR=   0.41  DAY=150.0  ENERGY= 9.060297E-01  DTEMP= 4.962264E-09  DSALT= 3.329970E-13
TS=  1900  YEAR=   0.43  DAY=158.3  ENERGY= 8.965763E-01  DTEMP= 4.917994E-09  DSALT= 3.377885E-13
TS=  2000  YEAR=   0.46  DAY=166.7  ENERGY= 8.874843E-01  DTEMP= 4.782509E-09  DSALT= 3.263422E-13
```

```
TS=  2100  YEAR=  0.48  DAY=175.0  ENERGY= 8.787525E-01  DTEMP= 4.675970E-09  DSALT= 3.217106E-13
TS=  2200  YEAR=  0.50  DAY=183.3  ENERGY= 8.706654E-01  DTEMP= 4.580536E-09  DSALT= 3.197874E-13
TS=  2300  YEAR=  0.52  DAY=191.7  ENERGY= 8.631569E-01  DTEMP= 4.492615E-09  DSALT= 3.171304E-13
TS=  2400  YEAR=  0.55  DAY=200.0  ENERGY= 8.564457E-01  DTEMP= 4.427453E-09  DSALT= 3.151363E-13
TS=  2500  YEAR=  0.57  DAY=208.3  ENERGY= 8.504133E-01  DTEMP= 4.369690E-09  DSALT= 3.140366E-13
TS=  2600  YEAR=  0.59  DAY=216.7  ENERGY= 8.452549E-01  DTEMP= 4.303695E-09  DSALT= 3.116093E-13
TS=  2700  YEAR=  0.62  DAY=225.0  ENERGY= 8.410124E-01  DTEMP= 4.236202E-09  DSALT= 3.088114E-13
TS=  2800  YEAR=  0.64  DAY=233.3  ENERGY= 8.377390E-01  DTEMP= 4.169011E-09  DSALT= 3.061915E-13
TS=  2900  YEAR=  0.66  DAY=241.7  ENERGY= 8.353845E-01  DTEMP= 4.113572E-09  DSALT= 3.046672E-13
TS=  3000  YEAR=  0.68  DAY=250.0  ENERGY= 8.340891E-01  DTEMP= 4.073308E-09  DSALT= 3.041561E-13
TS=  3100  YEAR=  0.71  DAY=258.3  ENERGY= 8.339189E-01  DTEMP= 4.030938E-09  DSALT= 3.033499E-13
TS=  3200  YEAR=  0.73  DAY=266.7  ENERGY= 8.349349E-01  DTEMP= 4.052863E-09  DSALT= 3.140909E-13
TS=  3300  YEAR=  0.75  DAY=275.0  ENERGY= 8.371324E-01  DTEMP= 3.928609E-09  DSALT= 2.988385E-13
TS=  3400  YEAR=  0.78  DAY=283.3  ENERGY= 8.405686E-01  DTEMP= 3.898257E-09  DSALT= 2.998921E-13
TS=  3500  YEAR=  0.80  DAY=291.7  ENERGY= 8.451545E-01  DTEMP= 3.853062E-09  DSALT= 2.980450E-13
TS=  3600  YEAR=  0.82  DAY=300.0  ENERGY= 8.509867E-01  DTEMP= 3.851377E-09  DSALT= 3.030611E-13
TS=  3700  YEAR=  0.84  DAY=308.3  ENERGY= 8.580925E-01  DTEMP= 3.840505E-09  DSALT= 3.053100E-13
TS=  3800  YEAR=  0.87  DAY=316.7  ENERGY= 8.664488E-01  DTEMP= 3.805552E-09  DSALT= 3.034212E-13
TS=  3900  YEAR=  0.89  DAY=325.0  ENERGY= 8.762385E-01  DTEMP= 3.768168E-09  DSALT= 3.015203E-13
TS=  4000  YEAR=  0.91  DAY=333.3  ENERGY= 8.875809E-01  DTEMP= 3.737796E-09  DSALT= 3.007891E-13
TS=  4100  YEAR=  0.94  DAY=341.7  ENERGY= 9.005579E-01  DTEMP= 3.701411E-09  DSALT= 2.991331E-13
TS=  4200  YEAR=  0.96  DAY=350.0  ENERGY= 9.151978E-01  DTEMP= 3.669555E-09  DSALT= 2.981841E-13
TS=  4300  YEAR=  0.98  DAY=358.3  ENERGY= 9.315860E-01  DTEMP= 3.647116E-09  DSALT= 2.985015E-13
TS=  4400  YEAR=  1.00  DAY=  1.4  ENERGY= 9.498099E-01  DTEMP= 3.710553E-09  DSALT= 3.133677E-13
TS=  4500  YEAR=  1.03  DAY=  9.7  ENERGY= 9.700364E-01  DTEMP= 3.625472E-09  DSALT= 3.016374E-13
TS=  4600  YEAR=  1.05  DAY= 18.1  ENERGY= 9.923584E-01  DTEMP= 3.596709E-09  DSALT= 2.992929E-13
TS=  4700  YEAR=  1.07  DAY= 26.4  ENERGY= 1.016772E+00  DTEMP= 3.570989E-09  DSALT= 2.970335E-13
TS=  4800  YEAR=  1.10  DAY= 34.7  ENERGY= 1.043006E+00  DTEMP= 3.553301E-09  DSALT= 2.967108E-13
TS=  4900  YEAR=  1.12  DAY= 43.1  ENERGY= 1.070952E+00  DTEMP= 3.528236E-09  DSALT= 2.955616E-13
TS=  5000  YEAR=  1.14  DAY= 51.4  ENERGY= 1.100362E+00  DTEMP= 3.495391E-09  DSALT= 2.932495E-13
TS=  5100  YEAR=  1.16  DAY= 59.7  ENERGY= 1.130840E+00  DTEMP= 3.464250E-09  DSALT= 2.908418E-13
TS=  5200  YEAR=  1.19  DAY= 68.1  ENERGY= 1.162059E+00  DTEMP= 3.439534E-09  DSALT= 2.893349E-13
TS=  5300  YEAR=  1.21  DAY= 76.4  ENERGY= 1.192940E+00  DTEMP= 3.462952E-09  DSALT= 2.952992E-13
TS=  5400  YEAR=  1.23  DAY= 84.7  ENERGY= 1.222123E+00  DTEMP= 3.455478E-09  DSALT= 2.969122E-13
TS=  5500  YEAR=  1.25  DAY= 93.1  ENERGY= 1.249015E+00  DTEMP= 3.434014E-09  DSALT= 2.964259E-13
TS=  5600  YEAR=  1.28  DAY=101.4  ENERGY= 1.273746E+00  DTEMP= 3.392617E-09  DSALT= 2.927570E-13
TS=  5700  YEAR=  1.30  DAY=109.7  ENERGY= 1.295605E+00  DTEMP= 3.352608E-09  DSALT= 2.899370E-13
TS=  5800  YEAR=  1.32  DAY=118.1  ENERGY= 1.314462E+00  DTEMP= 3.313647E-09  DSALT= 2.874613E-13
TS=  5900  YEAR=  1.35  DAY=126.4  ENERGY= 1.330537E+00  DTEMP= 5.018709E-09  DSALT= 5.700282E-13
TS=  6000  YEAR=  1.37  DAY=134.7  ENERGY= 1.344194E+00  DTEMP= 3.240086E-09  DSALT= 2.833123E-13
TS=  6100  YEAR=  1.39  DAY=143.1  ENERGY= 1.355652E+00  DTEMP= 3.239227E-09  DSALT= 2.891134E-13
TS=  6200  YEAR=  1.41  DAY=151.4  ENERGY= 1.365225E+00  DTEMP= 3.202297E-09  DSALT= 2.875937E-13
TS=  6300  YEAR=  1.44  DAY=159.7  ENERGY= 1.373141E+00  DTEMP= 3.171989E-09  DSALT= 2.856364E-13
TS=  6400  YEAR=  1.46  DAY=168.1  ENERGY= 1.379786E+00  DTEMP= 3.154626E-09  DSALT= 2.853743E-13
TS=  6500  YEAR=  1.48  DAY=176.4  ENERGY= 1.385494E+00  DTEMP= 3.190703E-09  DSALT= 2.942945E-13
TS=  6600  YEAR=  1.51  DAY=184.7  ENERGY= 1.390404E+00  DTEMP= 3.153180E-09  DSALT= 2.904130E-13
TS=  6700  YEAR=  1.53  DAY=193.1  ENERGY= 1.394699E+00  DTEMP= 3.120942E-09  DSALT= 2.874190E-13
TS=  6800  YEAR=  1.55  DAY=201.4  ENERGY= 1.398540E+00  DTEMP= 3.094082E-09  DSALT= 2.853438E-13
TS=  6900  YEAR=  1.57  DAY=209.7  ENERGY= 1.402046E+00  DTEMP= 3.062307E-09  DSALT= 2.823008E-13
TS=  7000  YEAR=  1.60  DAY=218.1  ENERGY= 1.405306E+00  DTEMP= 3.038083E-09  DSALT= 2.802350E-13
TS=  7100  YEAR=  1.62  DAY=226.4  ENERGY= 1.408417E+00  DTEMP= 3.018712E-09  DSALT= 2.789442E-13
TS=  7200  YEAR=  1.64  DAY=234.7  ENERGY= 1.411444E+00  DTEMP= 3.047340E-09  DSALT= 2.860119E-13
TS=  7300  YEAR=  1.67  DAY=243.1  ENERGY= 1.414451E+00  DTEMP= 3.015764E-09  DSALT= 2.824942E-13
TS=  7400  YEAR=  1.69  DAY=251.4  ENERGY= 1.417416E+00  DTEMP= 3.868927E-09  DSALT= 4.289562E-13
TS=  7500  YEAR=  1.71  DAY=259.7  ENERGY= 1.420478E+00  DTEMP= 3.003323E-09  DSALT= 2.847270E-13
TS=  7600  YEAR=  1.73  DAY=268.1  ENERGY= 1.423577E+00  DTEMP= 2.979360E-09  DSALT= 2.824663E-13
TS=  7700  YEAR=  1.76  DAY=276.4  ENERGY= 1.426630E+00  DTEMP= 2.966597E-09  DSALT= 2.820672E-13
TS=  7800  YEAR=  1.78  DAY=284.8  ENERGY= 1.429631E+00  DTEMP= 3.187744E-09  DSALT= 3.207965E-13
TS=  7900  YEAR=  1.80  DAY=293.1  ENERGY= 1.432559E+00  DTEMP= 2.929770E-09  DSALT= 2.795668E-13
TS=  8000  YEAR=  1.83  DAY=301.4  ENERGY= 1.435417E+00  DTEMP= 2.904986E-09  DSALT= 2.774271E-13
TS=  8100  YEAR=  1.85  DAY=309.8  ENERGY= 1.438193E+00  DTEMP= 3.067536E-09  DSALT= 3.128530E-13
TS=  8200  YEAR=  1.87  DAY=318.1  ENERGY= 1.440872E+00  DTEMP= 2.868229E-09  DSALT= 2.747189E-13
TS=  8300  YEAR=  1.89  DAY=326.4  ENERGY= 1.443440E+00  DTEMP= 2.862428E-09  DSALT= 2.754070E-13
TS=  8400  YEAR=  1.92  DAY=334.8  ENERGY= 1.445883E+00  DTEMP= 2.854847E-09  DSALT= 2.755993E-13
TS=  8500  YEAR=  1.94  DAY=343.1  ENERGY= 1.448189E+00  DTEMP= 2.839340E-09  DSALT= 2.745680E-13
TS=  8600  YEAR=  1.96  DAY=351.4  ENERGY= 1.450344E+00  DTEMP= 2.817763E-09  DSALT= 2.721360E-13
```

```
TS=  8700   YEAR=    1.98  DAY=359.8  ENERGY= 1.452334E+00  DTEMP= 2.790353E-09  DSALT= 2.685194E-13
TS=  8800   YEAR=    2.01  DAY=   2.8  ENERGY= 1.454154E+00  DTEMP= 2.766412E-09  DSALT= 2.654298E-13
TS=  8900   YEAR=    2.03  DAY=  11.2  ENERGY= 1.455795E+00  DTEMP= 2.760599E-09  DSALT= 2.660857E-13
TS=  9000 . YEAR=    2.05  DAY=  19.5  ENERGY= 1.457252E+00  DTEMP= 2.755342E-09  DSALT= 2.673455E-13
TS=  9100   YEAR=    2.08  DAY=  27.8  ENERGY= 1.458513E+00  DTEMP= 2.737278E-09  DSALT= 2.656286E-13
TS=  9200   YEAR=    2.10  DAY=  36.2  ENERGY= 1.459567E+00  DTEMP= 2.717752E-09  DSALT= 2.635247E-13
TS=  9300   YEAR=    2.12  DAY=  44.5  ENERGY= 1.460409E+00  DTEMP= 2.702920E-09  DSALT= 2.621799E-13
TS=  9400   YEAR=    2.14  DAY=  52.8  ENERGY= 1.461037E+00  DTEMP= 2.709993E-09  DSALT= 2.652825E-13
TS=  9500   YEAR=    2.17  DAY=  61.2  ENERGY= 1.461449E+00  DTEMP= 2.679892E-09  DSALT= 2.610100E-13
TS=  9600   YEAR=    2.19  DAY=  69.5  ENERGY= 1.461637E+00  DTEMP= 2.654191E-09  DSALT= 2.577921E-13
TS=  9700   YEAR=    2.21  DAY=  77.8  ENERGY= 1.461601E+00  DTEMP= 2.640832E-09  DSALT= 2.565530E-13
TS=  9800   YEAR=    2.24  DAY=  86.2  ENERGY= 1.461341E+00  DTEMP= 2.633867E-09  DSALT= 2.567101E-13
TS=  9900   YEAR=    2.26  DAY=  94.5  ENERGY= 1.460856E+00  DTEMP= 2.606645E-09  DSALT= 2.528486E-13
```

TEMPERATURE FOR J = 7 AT TIMESTEP 10000

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	12.46	13.02	12.75	12.58	12.65	12.52	12.44	12.42	12.43	12.45	12.47	12.47	12.47	12.47	12.47	12.47	12.47	12.47	12.47	12.49
2	11.42	12.19	12.75	12.58	12.09	11.78	11.55	11.44	11.40	11.41	11.43	11.44	11.44	11.44	11.44	11.44	11.44	11.44	11.45	11.47
3	8.88	9.70	10.40	10.49	10.25	9.92	9.64	9.48	9.40	9.37	9.38	9.38	9.38	9.38	9.38	9.38	9.38	9.38	9.38	9.42
4	6.12	6.50	7.11	7.28	7.16	6.94	6.74	6.63	6.58	6.55	6.55	6.55	6.55	6.55	6.55	6.55	6.54	6.54	6.55	6.58
5	0.00	4.64	4.88	4.96	4.90	4.81	4.74	4.71	4.70	4.70	4.70	4.70	4.70	4.70	4.70	4.70	4.70	4.70	4.70	4.71
6	0.00	4.09	4.13	4.16	4.15	4.13	4.12	4.13	4.13	4.14	4.14	4.14	4.14	4.14	4.14	4.14	4.14	4.14	4.14	4.14
7	0.00	0.00	4.01	4.02	4.02	4.02	4.03	4.04	4.05	4.05	4.06	4.06	4.06	4.06	4.06	4.06	4.06	4.06	4.06	4.06
8	0.00	0.00	4.00	4.00	4.00	4.01	4.01	4.02	4.03	4.04	4.04	4.05	4.05	4.05	4.05	4.05	4.05	4.05	4.05	4.05
9	0.00	0.00	0.00	4.00	4.00	4.00	4.01	4.01	4.02	4.02	4.02	4.02	4.02	4.02	4.02	4.02	4.02	4.02	4.02	4.02
10	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.01	4.01	4.01	4.01	4.01	4.01	4.01	4.01	4.01	4.01	4.01	4.01
11	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
12	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
13	0.00	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
14	0.00	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
15	0.00	0.00	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00

SALINITY FOR J = 7 AT TIMESTEP 10000

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0.55	0.54	0.48	0.50	0.58	0.58	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59
2	0.34	0.35	0.48	0.50	0.48	0.50	0.51	0.52	0.52	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51
3	0.14	0.16	0.23	0.28	0.31	0.33	0.34	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35
4	0.00	0.01	0.05	0.08	0.10	0.11	0.11	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
5	10.00	-0.07	-0.06	-0.05	-0.04	-0.04	-0.04	-0.04	-0.04	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03
6	10.00	-0.10	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08
7	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09
8	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09
9	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
10	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
11	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
12	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
13	10.00	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
14	10.00	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
15	10.00	10.00	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10

W VELOCITY FOR J = 7 AT TIMESTEP 10000

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	-0.01	-0.09	-0.15	-0.15	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14
3	0.06	-0.07	-0.17	-0.16	-0.15	-0.14	-0.13	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14
4	0.08	-0.04	-0.20	-0.19	-0.16	-0.14	-0.12	-0.13	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14
5	0.00	0.01	-0.25	-0.22	-0.18	-0.14	-0.12	-0.13	-0.13	-0.13	-0.13	-0.13	-0.14	-0.14	-0.14	-0.13	-0.13	-0.13	-0.13	-0.14
6	0.00	0.02	-0.21	-0.27	-0.21	-0.15	-0.11	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13
7	0.00	0.00	-0.15	-0.34	-0.25	-0.17	-0.10	-0.13	-0.12	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13
8	0.00	0.00	-0.08	-0.30	-0.30	-0.19	-0.09	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12

9	0.00	0.00	0.00	-0.23	-0.37	-0.22	-0.08	-0.12	-0.11	-0.11	-0.11	-0.11	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12
10	0.00	0.00	0.00	-0.13	-0.30	-0.25	-0.06	-0.11	-0.10	-0.10	-0.11	-0.11	-0.11	-0.11	-0.11	-0.11	-0.11	-0.11	-0.11	-0.11
11	0.00	0.00	0.00	0.00	-0.21	-0.29	-0.04	-0.11	-0.09	-0.09	-0.09	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
12	0.00	0.00	0.00	0.00	-0.11	-0.18	-0.01	-0.10	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08
13	0.00	0.00	0.00	0.00	0.00	-0.05	0.02	-0.09	-0.06	-0.07	-0.07	-0.07	-0.07	-0.07	-0.07	-0.07	-0.07	-0.07	-0.07	-0.07
14	0.00	0.00	0.00	0.00	0.00	-0.03	0.19	-0.08	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05
15	0.00	0.00	0.00	0.00	0.00	0.00	0.38	-0.07	-0.03	-0.02	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

U VELOCITY FOR J = 7 AT TIMESTEP 10000

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0.00	0.49	1.22	1.34	1.21	1.09	0.96	0.90	0.89	0.94	1.01	1.05	1.07	1.07	1.08	1.09	1.11	1.12	1.15	1.18
2	0.00	0.45	0.89	1.01	0.90	0.78	0.66	0.59	0.57	0.61	0.67	0.72	0.74	0.74	0.75	0.76	0.78	0.79	0.81	0.84
3	0.00	0.18	0.55	0.65	0.55	0.45	0.33	0.24	0.21	0.23	0.28	0.33	0.35	0.35	0.36	0.38	0.39	0.41	0.42	0.45
4	0.00	-0.03	0.33	0.38	0.30	0.21	0.10	0.00	-0.06	-0.05	-0.01	0.03	0.05	0.06	0.07	0.08	0.09	0.11	0.13	0.15
5	0.00	0.00	0.07	0.26	0.17	0.11	0.00	-0.11	-0.18	-0.18	-0.15	-0.11	-0.09	-0.08	-0.07	-0.06	-0.05	-0.03	-0.02	0.00
6	0.00	0.00	0.04	0.22	0.14	0.08	-0.03	-0.13	-0.21	-0.22	-0.19	-0.15	-0.13	-0.12	-0.11	-0.10	-0.09	-0.07	-0.06	-0.04
7	0.00	0.00	0.00	0.10	0.13	0.07	-0.03	-0.14	-0.22	-0.23	-0.20	-0.17	-0.15	-0.14	-0.13	-0.11	-0.10	-0.08	-0.07	-0.06
8	0.00	0.00	0.00	0.10	0.13	0.07	-0.03	-0.14	-0.23	-0.24	-0.21	-0.17	-0.15	-0.14	-0.13	-0.12	-0.10	-0.09	-0.08	-0.06
9	0.00	0.00	0.00	0.00	0.05	0.07	-0.03	-0.14	-0.23	-0.24	-0.21	-0.18	-0.16	-0.15	-0.14	-0.12	-0.11	-0.09	-0.08	-0.07
10	0.00	0.00	0.00	0.00	0.05	0.07	-0.03	-0.15	-0.23	-0.24	-0.21	-0.18	-0.16	-0.15	-0.14	-0.13	-0.11	-0.10	-0.08	-0.07
11	0.00	0.00	0.00	0.00	0.00	0.02	-0.03	-0.15	-0.23	-0.24	-0.21	-0.18	-0.16	-0.15	-0.14	-0.13	-0.11	-0.10	-0.08	-0.07
12	0.00	0.00	0.00	0.00	0.00	0.02	-0.03	-0.15	-0.23	-0.25	-0.21	-0.18	-0.16	-0.15	-0.14	-0.13	-0.11	-0.10	-0.09	-0.07
13	0.00	0.00	0.00	0.00	0.00	0.00	-0.06	-0.15	-0.23	-0.25	-0.22	-0.18	-0.16	-0.15	-0.14	-0.13	-0.12	-0.10	-0.09	-0.07
14	0.00	0.00	0.00	0.00	0.00	0.00	-0.06	-0.15	-0.23	-0.25	-0.22	-0.18	-0.16	-0.15	-0.14	-0.13	-0.12	-0.10	-0.09	-0.07
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.15	-0.23	-0.25	-0.22	-0.18	-0.16	-0.15	-0.14	-0.13	-0.12	-0.10	-0.09	-0.07

V VELOCITY FOR J = 7 AT TIMESTEP 10000

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0.00	-12.51	-10.54	-8.31	-6.96	-6.24	-5.87	-5.80	-6.01	-6.27	-6.39	-6.40	-6.37	-6.35	-6.34	-6.35	-6.35	-6.36	-6.37	-6.42
2	0.00	-5.71	-4.04	-1.98	-0.65	0.09	0.51	0.64	0.46	0.21	0.09	0.07	0.10	0.11	0.12	0.12	0.12	0.11	0.10	0.06
3	0.00	-5.21	-3.85	-2.13	-0.93	-0.18	0.32	0.55	0.44	0.22	0.10	0.07	0.10	0.11	0.12	0.11	0.12	0.11	0.11	0.07
4	0.00	-4.77	-3.48	-2.18	-1.20	-0.49	0.09	0.44	0.40	0.22	0.11	0.08	0.10	0.11	0.12	0.11	0.12	0.11	0.11	0.09
5	0.00	0.00	-3.19	-2.17	-1.36	-0.69	-0.06	0.36	0.37	0.21	0.11	0.08	0.10	0.11	0.12	0.11	0.12	0.11	0.12	0.11
6	0.00	0.00	-3.07	-2.15	-1.42	-0.76	-0.12	0.33	0.36	0.21	0.11	0.08	0.10	0.11	0.12	0.11	0.12	0.11	0.12	0.11
7	0.00	0.00	0.00	-2.14	-1.43	-0.79	-0.13	0.33	0.35	0.21	0.11	0.08	0.10	0.11	0.12	0.11	0.12	0.11	0.12	0.11
8	0.00	0.00	0.00	-2.13	-1.44	-0.79	-0.14	0.32	0.35	0.21	0.11	0.08	0.10	0.11	0.12	0.11	0.12	0.11	0.12	0.11
9	0.00	0.00	0.00	0.00	-1.44	-0.79	-0.14	0.32	0.35	0.21	0.11	0.08	0.10	0.11	0.12	0.11	0.12	0.11	0.12	0.11
10	0.00	0.00	0.00	0.00	-1.44	-0.79	-0.14	0.32	0.35	0.20	0.11	0.08	0.10	0.11	0.12	0.11	0.12	0.11	0.12	0.11
11	0.00	0.00	0.00	0.00	0.00	-0.79	-0.14	0.32	0.35	0.20	0.11	0.08	0.10	0.11	0.12	0.11	0.12	0.11	0.12	0.11
12	0.00	0.00	0.00	0.00	0.00	-0.79	-0.14	0.32	0.35	0.20	0.11	0.08	0.10	0.11	0.12	0.11	0.12	0.11	0.12	0.11
13	0.00	0.00	0.00	0.00	0.00	0.00	-0.14	0.32	0.35	0.20	0.11	0.08	0.10	0.11	0.12	0.11	0.12	0.11	0.12	0.11
14	0.00	0.00	0.00	0.00	0.00	0.00	-0.14	0.32	0.35	0.20	0.11	0.08	0.10	0.11	0.12	0.11	0.12	0.11	0.12	0.11
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.32	0.35	0.20	0.11	0.08	0.10	0.11	0.12	0.11	0.12	0.11	0.12	0.11

TEMPERATURE FOR J = 14 AT TIMESTEP 10000

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	18.33	17.75	17.81	18.02	18.07	18.04	17.98	17.91	17.95	18.00	18.04	18.07	18.12	18.18	18.26	18.35	18.45	18.54	18.62	18.66
2	14.64	13.86	14.20	14.56	14.68	14.72	14.71	14.58	14.59	14.67	14.73	14.75	14.78	14.84	14.94	15.08	15.24	15.40	15.52	15.51
3	9.50	9.08	9.53	9.90	10.05	10.13	10.17	10.07	10.06	10.13	10.18	10.20	10.21	10.26	10.36	10.50	10.68	10.88	11.03	11.00
4	5.66	5.59	5.87	5.94	5.93	5.96	6.04	6.06	6.07	6.10	6.12	6.13	6.13	6.16	6.21	6.30	6.41	6.55	6.64	6.62
5	0.00	4.40	4.42	4.35	4.28	4.26	4.30	4.33	4.35	4.36	4.36	4.37	4.37	4.37	4.39	4.41	4.45	4.49	4.52	4.51
6	0.00	4.04	4.05	4.03	4.02	4.01	4.02	4.02	4.03	4.03	4.03	4.03	4.03	4.03	4.03	4.04	4.04	4.05	4.05	4.05
7	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
8	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
9	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
10	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00

11	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
12	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
13	0.00	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
14	0.00	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
15	0.00	0.00	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00

SALINITY FOR J = 14 AT TIMESTEP 10000

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0.20	0.16	0.17	0.18	0.19	0.19	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18
2	0.09	0.07	0.09	0.12	0.14	0.14	0.13	0.12	0.11	0.11	0.12	0.12	0.12	0.11	0.11	0.11	0.11	0.11	0.12	0.12
3	-0.01	-0.02	0.00	0.02	0.04	0.05	0.04	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.04
4	-0.08	-0.08	-0.07	-0.06	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05
5	10.00	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09
6	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
7	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
8	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
9	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
10	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
11	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
12	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
13	10.00	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
14	10.00	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
15	10.00	10.00	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10

W VELOCITY FOR J = 14 AT TIMESTEP 10000

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	-0.09	-0.02	-0.07	-0.04	-0.03	-0.02	-0.03	-0.03	-0.03	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.05	-0.04	-0.04	-0.03
3	-0.09	-0.12	-0.13	-0.05	-0.01	0.00	-0.02	-0.02	-0.03	-0.04	-0.04	-0.04	-0.04	-0.04	-0.05	-0.06	-0.06	-0.05	-0.04	-0.02
4	-0.08	-0.23	-0.22	-0.06	0.02	0.04	0.01	-0.02	-0.04	-0.04	-0.04	-0.04	-0.04	-0.05	-0.06	-0.07	-0.07	-0.06	-0.04	-0.01
5	0.00	-0.28	-0.35	-0.08	0.06	0.10	0.04	-0.01	-0.04	-0.04	-0.04	-0.04	-0.04	-0.05	-0.06	-0.07	-0.08	-0.07	-0.04	0.00
6	0.00	-0.19	-0.36	-0.12	0.12	0.20	0.09	0.00	-0.04	-0.04	-0.04	-0.04	-0.04	-0.05	-0.06	-0.07	-0.08	-0.07	-0.04	0.00
7	0.00	0.00	-0.37	-0.17	0.20	0.32	0.15	0.01	-0.04	-0.04	-0.04	-0.04	-0.04	-0.05	-0.06	-0.07	-0.07	-0.07	-0.04	0.00
8	0.00	0.00	-0.21	-0.03	0.30	0.49	0.23	0.02	-0.04	-0.04	-0.04	-0.04	-0.04	-0.05	-0.06	-0.07	-0.07	-0.06	-0.04	0.00
9	0.00	0.00	0.00	0.15	0.43	0.71	0.33	0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.05	-0.06	-0.07	-0.06	-0.04	0.00
10	0.00	0.00	0.00	0.08	0.76	0.98	0.45	0.07	-0.04	-0.04	-0.04	-0.03	-0.04	-0.04	-0.05	-0.06	-0.06	-0.06	-0.03	0.00
11	0.00	0.00	0.00	0.00	1.17	1.31	0.61	0.10	-0.04	-0.04	-0.03	-0.03	-0.03	-0.04	-0.05	-0.05	-0.06	-0.05	-0.03	0.00
12	0.00	0.00	0.00	0.00	0.63	1.71	0.80	0.13	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.04	-0.05	-0.05	-0.05	-0.03	0.00
13	0.00	0.00	0.00	0.00	0.00	2.19	1.04	0.18	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.04	-0.04	-0.04	-0.02	0.00
14	0.00	0.00	0.00	0.00	0.00	1.17	1.07	0.23	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.02	-0.03	-0.03	-0.03	-0.02	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	1.11	0.30	-0.02	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01	-0.02	-0.02	-0.01	-0.01	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

U VELOCITY FOR J = 14 AT TIMESTEP 10000

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0.00	-1.24	-2.15	-2.64	-2.49	-2.59	-2.84	-3.11	-3.03	-2.82	-2.75	-2.71	-2.61	-2.43	-2.21	-1.97	-1.74	-1.59	-1.55	-1.57
2	0.00	-0.43	-1.64	-2.21	-2.22	-2.33	-2.51	-2.67	-2.59	-2.44	-2.38	-2.36	-2.28	-2.14	-1.97	-1.77	-1.58	-1.46	-1.45	-1.51
3	0.00	0.04	-0.66	-1.30	-1.52	-1.70	-1.73	-1.70	-1.63	-1.56	-1.53	-1.52	-1.47	-1.38	-1.27	-1.15	-1.03	-0.96	-0.97	-1.03
4	0.00	0.48	0.26	-0.43	-0.75	-0.95	-0.92	-0.81	-0.71	-0.66	-0.66	-0.66	-0.63	-0.59	-0.54	-0.49	-0.44	-0.40	-0.40	-0.43
5	0.00	0.00	0.36	0.01	-0.35	-0.57	-0.52	-0.40	-0.29	-0.25	-0.26	-0.26	-0.24	-0.21	-0.19	-0.17	-0.15	-0.13	-0.12	-0.12
6	0.00	0.00	0.48	0.15	-0.24	-0.47	-0.43	-0.30	-0.19	-0.15	-0.15	-0.16	-0.14	-0.12	-0.10	-0.09	-0.08	-0.06	-0.05	-0.03
7	0.00	0.00	0.00	-0.05	-0.23	-0.46	-0.42	-0.29	-0.18	-0.13	-0.14	-0.14	-0.13	-0.11	-0.09	-0.08	-0.07	-0.05	-0.04	-0.02
8	0.00	0.00	0.00	-0.05	-0.23	-0.46	-0.42	-0.29	-0.17	-0.13	-0.14	-0.14	-0.12	-0.10	-0.09	-0.08	-0.07	-0.05	-0.04	-0.01
9	0.00	0.00	0.00	0.00	-0.32	-0.46	-0.41	-0.29	-0.17	-0.13	-0.13	-0.13	-0.12	-0.10	-0.09	-0.07	-0.06	-0.05	-0.03	-0.01
10	0.00	0.00	0.00	0.00	-0.32	-0.46	-0.41	-0.28	-0.16	-0.12	-0.13	-0.13	-0.12	-0.10	-0.08	-0.07	-0.06	-0.05	-0.03	-0.01
11	0.00	0.00	0.00	0.00	0.00	-0.44	-0.41	-0.28	-0.15	-0.11	-0.12	-0.13	-0.11	-0.09	-0.08	-0.07	-0.06	-0.05	-0.03	-0.01
12	0.00	0.00	0.00	0.00	0.00	-0.44	-0.41	-0.27	-0.14	-0.10	-0.11	-0.12	-0.11	-0.09	-0.08	-0.07	-0.06	-0.04	-0.03	-0.01

13	0.00	0.00	0.00	0.00	0.00	0.00	-0.32	-0.26	-0.13	-0.09	-0.10	-0.11	-0.10	-0.08	-0.07	-0.06	-0.05	-0.04	-0.02	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	-0.31	-0.25	-0.12	-0.07	-0.09	-0.10	-0.09	-0.07	-0.06	-0.05	-0.04	-0.03	-0.02	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.14	-0.09	-0.05	-0.07	-0.08	-0.08	-0.06	-0.05	-0.04	-0.04	-0.02	-0.01	0.01

V VELOCITY FOR J = 14 AT TIMESTEP 10000

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0.00	-15.79	-17.31	-15.15	-12.43	-11.09	-10.96	-11.57	-12.78	-13.07	-12.79	-12.61	-12.69	-12.85	-13.01	-13.13	-13.23	-13.18	-12.91	-12.40
2	0.00	-4.28	-4.68	-2.29	0.19	1.41	1.46	0.84	-0.21	-0.47	-0.22	-0.05	-0.12	-0.25	-0.38	-0.50	-0.59	-0.55	-0.32	0.14
3	0.00	-5.36	-4.40	-1.80	0.38	1.45	1.38	0.67	-0.13	-0.33	-0.15	-0.01	-0.04	-0.13	-0.22	-0.29	-0.36	-0.34	-0.19	0.11
4	0.00	-5.99	-4.09	-1.50	0.49	1.50	1.39	0.60	-0.08	-0.23	-0.09	0.01	0.00	-0.04	-0.08	-0.10	-0.14	-0.13	-0.07	0.06
5	0.00	0.00	-3.96	-1.53	0.43	1.50	1.46	0.64	-0.05	-0.19	-0.07	0.02	0.01	0.00	-0.02	-0.01	-0.02	-0.02	-0.01	0.02
6	0.00	0.00	-3.97	-1.59	0.35	1.48	1.50	0.67	-0.03	-0.18	-0.07	0.02	0.02	0.01	0.00	0.01	0.01	0.02	0.01	0.00
7	0.00	0.00	0.00	-1.61	0.33	1.48	1.50	0.68	-0.03	-0.17	-0.07	0.02	0.02	0.01	0.00	0.02	0.02	0.02	0.01	0.00
8	0.00	0.00	0.00	-1.61	0.32	1.48	1.50	0.68	-0.03	-0.17	-0.07	0.02	0.02	0.01	0.00	0.02	0.02	0.02	0.01	0.00
9	0.00	0.00	0.00	0.00	0.34	1.47	1.50	0.67	-0.03	-0.17	-0.07	0.02	0.02	0.01	0.00	0.02	0.02	0.02	0.01	0.00
10	0.00	0.00	0.00	0.00	0.34	1.47	1.50	0.67	-0.03	-0.17	-0.07	0.02	0.02	0.01	0.00	0.02	0.02	0.02	0.01	0.00
11	0.00	0.00	0.00	0.00	0.00	1.49	1.50	0.67	-0.03	-0.17	-0.06	0.02	0.02	0.01	0.01	0.02	0.02	0.02	0.01	0.00
12	0.00	0.00	0.00	0.00	0.00	1.49	1.50	0.66	-0.03	-0.17	-0.06	0.03	0.02	0.01	0.01	0.02	0.02	0.02	0.01	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	1.51	0.66	-0.03	-0.17	-0.06	0.03	0.02	0.01	0.01	0.02	0.02	0.02	0.01	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	1.51	0.65	-0.04	-0.16	-0.06	0.03	0.02	0.01	0.01	0.02	0.02	0.02	0.01	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.66	-0.03	-0.16	-0.05	0.03	0.03	0.02	0.01	0.02	0.02	0.03	0.01	0.01

ETA SURFACE DISPLACEMENT AT (BAROTROPIC) TIMESTEP 400000

	1	2	3	4	5	6	7	8	9	10
20	0.00000E+00	-1.10914E+01	-1.08050E+01	-1.11517E+01	-9.29359E+00	-5.66259E+00	-3.52026E+00	-3.23222E+00	-3.85471E+00	-4.61223E+00
19	0.00000E+00	-1.10914E+01	-1.08050E+01	-1.11517E+01	-9.29359E+00	-5.66259E+00	-3.52026E+00	-3.23222E+00	-3.85471E+00	-4.61223E+00
18	0.00000E+00	-2.07607E+00	-1.06523E+01	-1.10622E+01	-1.04503E+01	-9.61481E+00	-8.95587E+00	-9.15180E+00	-9.75863E+00	-1.01266E+01
17	0.00000E+00	5.50979E-02	-4.08872E+00	-4.99396E+00	-3.97150E+00	-4.74273E+00	-5.86824E+00	-6.80647E+00	-7.07432E+00	-6.80626E+00
16	0.00000E+00	2.20224E+00	-4.48965E-01	1.48962E+00	3.13088E+00	2.68197E+00	9.49686E-01	-5.32852E-01	-1.05965E+00	-4.74362E-01
15	0.00000E+00	1.67942E+00	3.11807E+00	6.03790E+00	8.50417E+00	8.29501E+00	7.03317E+00	5.48406E+00	4.68086E+00	5.02362E+00
14	0.00000E+00	1.03300E+00	4.48740E+00	9.50537E+00	1.16244E+01	1.16714E+01	1.04886E+01	9.36976E+00	8.56863E+00	8.70723E+00
13	0.00000E+00	-6.89321E-01	5.69547E+00	1.09264E+01	1.35051E+01	1.31648E+01	1.21171E+01	1.09960E+01	1.04103E+01	1.04218E+01
12	0.00000E+00	-2.65118E+00	5.04257E+00	1.16912E+01	1.37920E+01	1.36397E+01	1.23130E+01	1.12666E+01	1.05688E+01	1.06163E+01
11	0.00000E+00	-5.21906E+00	4.32414E+00	1.06769E+01	1.33173E+01	1.29006E+01	1.17220E+01	1.04057E+01	9.71430E+00	9.56573E+00
10	0.00000E+00	-7.86916E+00	1.93206E+00	9.16505E+00	1.14241E+01	1.14726E+01	1.01799E+01	8.92989E+00	7.94957E+00	7.76598E+00
9	0.00000E+00	-1.09598E+01	-4.27579E-01	6.05962E+00	8.92950E+00	8.94835E+00	8.10326E+00	6.72764E+00	5.75086E+00	5.23545E+00
8	0.00000E+00	-1.39433E+01	-4.16868E+00	2.73360E+00	5.24499E+00	5.92820E+00	5.21197E+00	4.19290E+00	3.01548E+00	2.45852E+00
7	0.00000E+00	-1.72303E+01	-7.60340E+00	-1.77791E+00	1.33929E+00	2.06393E+00	1.97431E+00	1.07568E+00	1.59352E-01	-7.03032E-01
6	0.00000E+00	-2.02627E+01	-1.20357E+01	-6.08262E+00	-3.35034E+00	-1.93720E+00	-1.90767E+00	-2.28750E+00	-3.19280E+00	-3.89151E+00
5	0.00000E+00	-2.35391E+01	-1.58581E+01	-1.10886E+01	-7.81165E+00	-6.50605E+00	-6.03040E+00	-6.42894E+00	-6.90855E+00	-7.76149E+00
4	0.00000E+00	-2.65962E+01	-2.08046E+01	-1.61118E+01	-1.33250E+01	-1.15920E+01	-1.14279E+01	-1.16215E+01	-1.21718E+01	-1.25824E+01
3	0.00000E+00	-2.97077E+01	-2.66774E+01	-2.44680E+01	-2.19361E+01	-2.03157E+01	-1.91168E+01	-1.88759E+01	-1.83381E+01	-1.83944E+01
2	0.00000E+00	-3.10909E+01	-3.05543E+01	-2.91774E+01	-2.78935E+01	-2.59493E+01	-2.47852E+01	-2.36393E+01	-2.30110E+01	-2.22627E+01
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

	11	12	13	14	15	16	17	18	19	20
20	-4.54105E+00	-3.93518E+00	-3.18649E+00	-2.41424E+00	-1.62312E+00	-7.80557E-01	1.32500E-01	1.17405E+00	2.29526E+00	3.47741E+00
19	-4.54105E+00	-3.93518E+00	-3.18649E+00	-2.41424E+00	-1.62312E+00	-7.80557E-01	1.32500E-01	1.17405E+00	2.29526E+00	3.47741E+00
18	-1.00680E+01	-9.55847E+00	-8.82298E+00	-8.01441E+00	-7.12612E+00	-6.10718E+00	-4.94195E+00	-3.67748E+00	-2.33362E+00	-8.45346E-01
17	-6.40541E+00	-5.98365E+00	-5.44983E+00	-4.75929E+00	-3.88472E+00	-2.82766E+00	-1.61073E+00	-3.11018E-01	9.90920E-01	2.26631E+00
16	3.59374E-02	3.57884E-01	6.88560E-01	1.21553E+00	1.94932E+00	2.89379E+00	3.99773E+00	5.17077E+00	6.22277E+00	6.98893E+00
15	5.59114E+00	5.83400E+00	5.99871E+00	6.27850E+00	6.77672E+00	7.44662E+00	8.30266E+00	9.23918E+00	1.00940E+01	1.05693E+01
14	9.10577E+00	9.35509E+00	9.38112E+00	9.47529E+00	9.67883E+00	1.00559E+01	1.05542E+01	1.12212E+01	1.19057E+01	1.24005E+01
13	1.07751E+01	1.09324E+01	1.09528E+01	1.08878E+01	1.09468E+01	1.10546E+01	1.13011E+01	1.16384E+01	1.21480E+01	1.26228E+01
12	1.08576E+01	1.10603E+01	1.09937E+01	1.09270E+01	1.08373E+01	1.08674E+01	1.08917E+01	1.10530E+01	1.12912E+01	1.17074E+01
11	9.85001E+00	9.95795E+00	9.96006E+00	9.78304E+00	9.70667E+00	9.59347E+00	9.58541E+00	9.54208E+00	9.65606E+00	9.84566E+00
10	7.85257E+00	8.03962E+00	7.93906E+00	7.83586E+00	7.62466E+00	7.54319E+00	7.38404E+00	7.33148E+00	7.23169E+00	7.33283E+00

9	5.30424E+00	5.30774E+00	5.31875E+00	5.09146E+00	4.95943E+00	4.72241E+00	4.61389E+00	4.39527E+00	4.30136E+00	4.16783E+00
8	2.17596E+00	2.21244E+00	2.05390E+00	1.95826E+00	1.67300E+00	1.52415E+00	1.24259E+00	1.09378E+00	8.16677E-01	7.00043E-01
7	-1.03617E+00	-1.33017E+00	-1.39490E+00	-1.66355E+00	-1.80923E+00	-2.13979E+00	-2.31974E+00	-2.65753E+00	-2.84603E+00	-3.16975E+00
6	-4.60251E+00	-4.89486E+00	-5.24969E+00	-5.37665E+00	-5.71045E+00	-5.89239E+00	-6.27437E+00	-6.49207E+00	-6.88541E+00	-7.10488E+00
5	-8.34375E+00	-8.98317E+00	-9.23793E+00	-9.61019E+00	-9.75704E+00	-1.01301E+01	-1.03368E+01	-1.07576E+01	-1.10016E+01	-1.14348E+01
4	-1.32610E+01	-1.36218E+01	-1.41084E+01	-1.42648E+01	-1.46059E+01	-1.47307E+01	-1.50986E+01	-1.52798E+01	-1.56900E+01	-1.59078E+01
3	-1.82367E+01	-1.82584E+01	-1.82676E+01	-1.85056E+01	-1.84891E+01	-1.87210E+01	-1.87402E+01	-1.90111E+01	-1.90747E+01	-1.93769E+01
2	-2.18377E+01	-2.15199E+01	-2.15520E+01	-2.13624E+01	-2.14613E+01	-2.13245E+01	-2.14496E+01	-2.13354E+01	-2.14658E+01	-2.13600E+01
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

	21	22	23	24	25
20	4.69653E+00	6.27034E+00	8.24375E+00	1.25785E+01	0.00000E+00
19	4.69653E+00	6.27034E+00	8.24375E+00	1.25785E+01	0.00000E+00
18	9.78207E-01	3.30587E+00	6.48844E+00	9.74413E+00	0.00000E+00
17	3.56013E+00	5.00042E+00	6.63892E+00	8.07267E+00	0.00000E+00
16	7.38290E+00	7.46708E+00	7.41032E+00	6.75905E+00	0.00000E+00
15	1.04602E+01	9.66919E+00	8.30331E+00	5.75187E+00	0.00000E+00
14	1.22852E+01	1.12688E+01	9.09710E+00	5.30825E+00	0.00000E+00
13	1.27718E+01	1.19384E+01	9.63691E+00	4.95190E+00	0.00000E+00
12	1.19832E+01	1.16290E+01	9.61690E+00	5.03925E+00	0.00000E+00
11	1.02188E+01	1.02299E+01	8.99536E+00	4.96203E+00	0.00000E+00
10	7.55379E+00	7.98409E+00	7.56200E+00	4.60689E+00	0.00000E+00
9	4.34407E+00	4.81567E+00	5.53259E+00	4.07751E+00	0.00000E+00
8	6.01437E-01	1.09286E+00	2.36692E+00	3.65803E+00	0.00000E+00
7	-3.26421E+00	-3.16397E+00	-1.84226E+00	8.44116E-01	0.00000E+00
6	-7.44934E+00	-7.42848E+00	-6.83317E+00	-4.45220E+00	0.00000E+00
5	-1.16653E+01	-1.19897E+01	-1.17585E+01	-1.08390E+01	0.00000E+00
4	-1.63353E+01	-1.65409E+01	-1.68716E+01	-1.68993E+01	0.00000E+00
3	-1.94758E+01	-1.98087E+01	-1.99343E+01	-2.04441E+01	0.00000E+00
2	-2.14925E+01	-2.13983E+01	-2.15443E+01	-2.14730E+01	0.00000E+00
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

UBT TRANSPORT AT (BAROTROPIC) TIMESTEP 400000

	1	2	3	4	5	6	7	8	9	10
20	0.00000E+00	-5.23565E+04	-8.61204E+04	-7.15685E+04	1.07774E+05	3.87475E+05	5.64789E+05	4.79427E+05	2.99796E+05	2.49780E+05
19	0.00000E+00	1.78892E+05	5.12365E+05	1.00741E+06	1.55822E+06	1.87878E+06	1.58384E+06	8.67359E+05	4.29584E+05	3.58317E+05
18	0.00000E+00	-5.23565E+04	-8.61204E+04	-7.15685E+04	1.07774E+05	3.87475E+05	5.64789E+05	4.79427E+05	2.99796E+05	2.49780E+05
17	0.00000E+00	-7.13054E+04	-3.06140E+05	-5.95834E+05	-8.30672E+05	-7.97791E+05	-4.31496E+05	-1.66754E+04	1.34507E+05	1.11833E+05
16	0.00000E+00	-3.24220E+02	-5.75130E+03	-9.94932E+04	-2.80494E+05	-4.69730E+05	-4.43813E+05	-1.96975E+05	-5.04706E+03	3.99918E+03
15	0.00000E+00	-1.64942E+04	-4.33782E+04	-5.99000E+04	-1.33429E+05	-2.34352E+05	-2.90221E+05	-2.00218E+05	-8.53487E+04	-6.29187E+04
14	0.00000E+00	-9.24769E+02	-3.26187E+03	-3.07492E+04	-8.06458E+04	-1.66199E+05	-2.08471E+05	-1.77873E+05	-1.15050E+05	-9.06491E+04
13	0.00000E+00	-5.75701E+03	-8.88148E+03	-1.39382E+03	-3.53000E+04	-1.01983E+05	-1.74276E+05	-1.72913E+05	-1.29881E+05	-1.02450E+05
12	0.00000E+00	2.47256E+03	7.07879E+03	2.74318E+03	-9.88945E+03	-6.85479E+04	-1.37717E+05	-1.69308E+05	-1.43465E+05	-1.13660E+05
11	0.00000E+00	-2.05967E+02	5.39918E+03	1.79976E+04	4.97275E+03	-3.45386E+04	-1.12078E+05	-1.60459E+05	-1.54204E+05	-1.25321E+05
10	0.00000E+00	4.95815E+03	1.37786E+04	1.74706E+04	1.93128E+04	-1.54188E+04	-8.06387E+04	-1.45471E+05	-1.57024E+05	-1.33331E+05
9	0.00000E+00	3.08428E+03	1.32343E+04	2.65778E+04	2.34781E+04	5.60902E+03	-5.63683E+04	-1.24987E+05	-1.52079E+05	-1.36032E+05
8	0.00000E+00	6.29704E+03	1.74180E+04	2.41101E+04	3.11044E+04	1.46210E+04	-3.01816E+04	-9.96265E+04	-1.39798E+05	-1.33618E+05
7	0.00000E+00	4.48422E+03	1.62108E+04	2.85069E+04	2.91663E+04	2.60413E+04	-1.03252E+04	-7.24569E+04	-1.19245E+05	-1.25655E+05
6	0.00000E+00	6.36387E+03	1.76492E+04	2.46329E+04	3.18568E+04	2.76956E+04	1.13613E+04	-3.57785E+04	-8.53196E+04	-1.02027E+05
5	0.00000E+00	4.56190E+03	1.58676E+04	2.67490E+04	2.88731E+04	3.62480E+04	3.50687E+04	1.23215E+04	-2.14495E+04	-4.36382E+04
4	0.00000E+00	6.72871E+03	1.98513E+04	3.16845E+04	4.71027E+04	6.15783E+04	8.62119E+04	1.01368E+05	9.10999E+04	7.27965E+04
3	0.00000E+00	1.05493E+04	3.81164E+04	7.73330E+04	1.19984E+05	1.66227E+05	2.09022E+05	2.38015E+05	2.45370E+05	2.33638E+05
2	0.00000E+00	8.44267E+03	3.27498E+04	7.74111E+04	1.47677E+05	2.23657E+05	2.77185E+05	3.07882E+05	3.22282E+05	3.18018E+05
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

	11	12	13	14	15	16	17	18	19	20
20	2.55911E+05	2.59606E+05	2.44762E+05	2.26589E+05	2.07257E+05	1.87608E+05	1.59186E+05	1.25722E+05	9.29465E+04	6.90045E+04
19	3.89347E+05	3.94358E+05	3.78459E+05	3.57867E+05	3.41269E+05	3.25281E+05	3.17135E+05	3.08203E+05	2.82846E+05	2.30636E+05

18	2.55911E+05	2.59606E+05	2.44762E+05	2.26589E+05	2.07257E+05	1.87608E+05	1.59186E+05	1.25722E+05	9.29465E+04	6.90045E+04
17	8.48943E+04	7.22151E+04	6.44606E+04	5.11410E+04	3.76352E+04	2.25867E+04	9.73130E+03	-1.66618E+03	-7.24948E+03	-1.01760E+04
16	-3.68256E+04	-5.13375E+04	-5.08443E+04	-4.81966E+04	-4.91454E+04	-4.79048E+04	-4.54400E+04	-3.97026E+04	-3.35151E+04	-2.50402E+04
15	-8.49021E+04	-9.57794E+04	-8.76461E+04	-7.77012E+04	-6.91258E+04	-6.20911E+04	-5.29793E+04	-4.41330E+04	-3.42767E+04	-2.59105E+04
14	-9.60432E+04	-9.94035E+04	-9.20794E+04	-7.98627E+04	-7.04191E+04	-6.15921E+04	-5.34168E+04	-4.42143E+04	-3.60604E+04	-2.71230E+04
13	-9.79340E+04	-9.86258E+04	-9.27082E+04	-8.29936E+04	-7.30587E+04	-6.51383E+04	-5.67548E+04	-4.86167E+04	-3.95188E+04	-3.12307E+04
12	-1.02273E+05	-1.00832E+05	-9.73480E+04	-8.88039E+04	-7.94589E+04	-7.04657E+04	-6.22652E+04	-5.30875E+04	-4.40349E+04	-3.38535E+04
11	-1.07751E+05	-1.03440E+05	-1.00603E+05	-9.37651E+04	-8.39745E+04	-7.45941E+04	-6.52677E+04	-5.60491E+04	-4.57643E+04	-3.58243E+04
10	-1.11198E+05	-1.02405E+05	-9.94314E+04	-9.36661E+04	-8.47537E+04	-7.47226E+04	-6.53520E+04	-5.56463E+04	-4.59189E+04	-3.51891E+04
9	-1.11447E+05	-9.80849E+04	-9.31604E+04	-8.86711E+04	-8.08191E+04	-7.14609E+04	-6.18846E+04	-5.27497E+04	-4.30319E+04	-3.34763E+04
8	-1.10392E+05	-9.26492E+04	-8.48884E+04	-8.02307E+04	-7.39728E+04	-6.54130E+04	-5.64537E+04	-4.74820E+04	-3.88324E+04	-2.95674E+04
7	-1.07368E+05	-8.82869E+04	-7.70129E+04	-7.13625E+04	-6.58346E+04	-5.86552E+04	-5.02223E+04	-4.18362E+04	-3.33969E+04	-2.54086E+04
6	-9.41925E+04	-7.84159E+04	-6.66763E+04	-6.01053E+04	-5.56340E+04	-5.02159E+04	-4.35515E+04	-3.61012E+04	-2.87211E+04	-2.11292E+04
5	-4.79066E+04	-4.26630E+04	-3.61177E+04	-3.26130E+04	-3.13229E+04	-3.02954E+04	-2.81097E+04	-2.48270E+04	-2.05716E+04	-1.59348E+04
4	5.74037E+04	4.86157E+04	4.33044E+04	3.82208E+04	3.17294E+04	2.43999E+04	1.71820E+04	1.08488E+04	5.52694E+03	1.65905E+03
3	2.13887E+05	1.93688E+05	1.76133E+05	1.60297E+05	1.44448E+05	1.27411E+05	1.09111E+05	8.98922E+04	7.02673E+04	5.05484E+04
2	3.01390E+05	2.80564E+05	2.60600E+05	2.42788E+05	2.25825E+05	2.07905E+05	1.87898E+05	1.65506E+05	1.40682E+05	1.13280E+05
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

	21	22	23	24	25
20	4.42097E+04	1.61437E+04	-1.30751E+04	0.00000E+00	0.00000E+00
19	1.72764E+05	1.20574E+05	6.18312E+04	0.00000E+00	0.00000E+00
18	4.42097E+04	1.61437E+04	-1.30751E+04	0.00000E+00	0.00000E+00
17	-8.74122E+03	-7.55716E+03	9.82945E+03	0.00000E+00	0.00000E+00
16	-1.82145E+04	-8.22196E+03	-1.38365E+04	0.00000E+00	0.00000E+00
15	-1.59992E+04	-1.05603E+04	7.27230E+03	0.00000E+00	0.00000E+00
14	-2.04421E+04	-1.03929E+04	-1.31101E+04	0.00000E+00	0.00000E+00
13	-2.08170E+04	-1.35104E+04	4.54129E+03	0.00000E+00	0.00000E+00
12	-2.51610E+04	-1.33420E+04	-1.21626E+04	0.00000E+00	0.00000E+00
11	-2.41294E+04	-1.50636E+04	2.13887E+03	0.00000E+00	0.00000E+00
10	-2.56288E+04	-1.37053E+04	-1.02726E+04	0.00000E+00	0.00000E+00
9	-2.25360E+04	-1.37318E+04	6.64569E+02	0.00000E+00	0.00000E+00
8	-2.11819E+04	-1.11698E+04	-7.51742E+03	0.00000E+00	0.00000E+00
7	-1.66624E+04	-9.74265E+03	3.05014E+02	0.00000E+00	0.00000E+00
6	-1.44413E+04	-7.08050E+03	-4.48955E+03	0.00000E+00	0.00000E+00
5	-1.03537E+04	-5.83838E+03	1.02775E+02	0.00000E+00	0.00000E+00
4	-9.52852E+02	-9.75965E+02	-1.54564E+03	0.00000E+00	0.00000E+00
3	3.17745E+04	1.47897E+04	4.27715E+03	0.00000E+00	0.00000E+00
2	8.28434E+04	4.96369E+04	1.59432E+04	0.00000E+00	0.00000E+00
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

VBT TRANSPORT AT (BAROTROPIC) TIMESTEP 400000

	1	2	3	4	5	6	7	8	9	10
20	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
19	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
18	0.00000E+00	1.90060E+05	2.60106E+05	5.05349E+05	5.91394E+05	3.10221E+05	-4.86918E+05	-7.15779E+05	-2.07418E+05	2.68869E+04
17	0.00000E+00	4.32333E+03	4.25793E+04	3.11035E+05	7.06524E+05	6.70268E+05	-2.77968E+04	-6.83047E+05	-2.86034E+05	-3.04066E+03
16	0.00000E+00	-1.04434E+05	-2.13222E+05	-1.30546E+04	4.06108E+05	7.42744E+05	4.99827E+05	-2.11871E+05	-2.44496E+05	-6.58723E+04
15	0.00000E+00	-1.31244E+05	-2.39064E+05	-1.58516E+05	1.65060E+05	5.50040E+05	6.58720E+05	1.47729E+05	-1.40206E+05	-1.24418E+05
14	0.00000E+00	-1.59951E+05	-2.60427E+05	-2.10525E+05	2.54854E+04	4.09079E+05	6.62576E+05	3.40907E+05	-5.87953E+04	-1.37093E+05
13	0.00000E+00	-1.72983E+05	-2.62592E+05	-2.47281E+05	-7.28185E+04	2.74078E+05	6.33644E+05	4.35727E+05	1.66335E+04	-1.34870E+05
12	0.00000E+00	-1.81490E+05	-2.59865E+05	-2.54480E+05	-1.46371E+05	1.50017E+05	5.47412E+05	4.93299E+05	7.73616E+04	-1.07472E+05
11	0.00000E+00	-1.81745E+05	-2.52528E+05	-2.59517E+05	-1.92242E+05	3.47871E+04	4.37061E+05	4.95194E+05	1.39993E+05	-7.61490E+04
10	0.00000E+00	-1.78340E+05	-2.42620E+05	-2.54945E+05	-2.27063E+05	-5.93505E+04	3.02811E+05	4.62553E+05	1.79467E+05	-2.74674E+04
9	0.00000E+00	-1.69557E+05	-2.30671E+05	-2.51724E+05	-2.46364E+05	-1.39417E+05	1.69598E+05	3.86919E+05	2.06643E+05	1.69273E+04
8	0.00000E+00	-1.58312E+05	-2.16911E+05	-2.45370E+05	-2.61998E+05	-1.97807E+05	3.79909E+04	2.90016E+05	2.02990E+05	6.62623E+04
7	0.00000E+00	-1.44132E+05	-2.02462E+05	-2.41537E+05	-2.70233E+05	-2.42785E+05	-7.33761E+04	1.69769E+05	1.80752E+05	9.74286E+04
6	0.00000E+00	-1.29110E+05	-1.87261E+05	-2.37720E+05	-2.79825E+05	-2.68290E+05	-1.62552E+05	5.22982E+04	1.27696E+05	1.17514E+05
5	0.00000E+00	-1.12969E+05	-1.72280E+05	-2.36081E+05	-2.86271E+05	-2.80435E+05	-2.05465E+05	-5.06388E+04	7.17067E+04	1.07436E+05
4	0.00000E+00	-9.47779E+04	-1.53191E+05	-2.27799E+05	-2.83959E+05	-2.65286E+05	-1.95627E+05	-8.95390E+04	1.90289E+04	8.48849E+04
3	0.00000E+00	-6.14435E+04	-1.09381E+05	-1.78796E+05	-2.31838E+05	-2.12462E+05	-1.23451E+05	-7.89646E+04	-1.85285E+03	4.59278E+04

2	0.00000E+00	-2.00987E+04	-3.78457E+04	-6.85839E+04	-9.88686E+04	-8.22226E+04	-4.53571E+04	-2.77337E+04	-6.42533E+03	1.62000E+04
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

	11	12	13	14	15	16	17	18	19	20
20	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
19	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
18	2.65667E+04	-1.37700E+04	-2.75080E+04	-2.30023E+04	-2.19504E+04	-2.09105E+04	-2.14053E+04	-2.58965E+04	-3.68988E+04	-4.33189E+04
17	2.62688E+04	-2.58502E+04	-4.61316E+04	-4.56369E+04	-4.08042E+04	-4.40272E+04	-4.63915E+04	-5.14225E+04	-5.34716E+04	-5.55795E+04
16	-9.64982E+03	-2.88790E+04	-5.34399E+04	-5.29919E+04	-5.22314E+04	-5.00670E+04	-5.31150E+04	-5.18706E+04	-5.30938E+04	-5.08467E+04
15	-4.43810E+04	-3.12594E+04	-4.02480E+04	-5.02895E+04	-4.62803E+04	-4.67972E+04	-4.38183E+04	-4.54972E+04	-4.28037E+04	-4.39632E+04
14	-7.43140E+04	-2.34401E+04	-2.74718E+04	-3.44677E+04	-3.98361E+04	-3.48124E+04	-3.61799E+04	-3.34873E+04	-3.56842E+04	-3.33433E+04
13	-8.10269E+04	-2.42693E+04	-9.00242E+03	-2.38000E+04	-2.62820E+04	-2.82556E+04	-2.38341E+04	-2.64841E+04	-2.42113E+04	-2.67506E+04
12	-8.75334E+04	-1.87776E+04	-1.55863E+03	-6.64997E+03	-1.83974E+04	-1.55099E+04	-1.69135E+04	-1.37488E+04	-1.67920E+04	-1.43516E+04
11	-7.73105E+04	-2.22990E+04	1.05682E+04	2.95643E+03	-2.53071E+03	-7.81084E+03	-3.25623E+03	-5.51832E+03	-2.89571E+03	-5.70105E+03
10	-6.53856E+04	-1.61163E+04	1.29558E+04	1.89972E+04	7.75631E+03	7.68973E+03	5.34842E+03	9.24365E+03	6.46830E+03	9.03426E+03
9	-3.48036E+04	-1.29033E+04	2.18387E+04	2.64279E+04	2.50489E+04	1.78519E+04	2.07487E+04	1.86263E+04	2.17003E+04	1.89271E+04
8	-2.62668E+03	5.34160E+03	2.45720E+04	3.92194E+04	3.49834E+04	3.48112E+04	3.07146E+04	3.39340E+04	3.14462E+04	3.40370E+04
7	4.20514E+04	2.46947E+04	3.81524E+04	4.46855E+04	5.02171E+04	4.55669E+04	4.71006E+04	4.39648E+04	4.65739E+04	4.38182E+04
6	7.57789E+04	5.63220E+04	4.94193E+04	5.72139E+04	5.77432E+04	6.12608E+04	5.77544E+04	5.97135E+04	5.66516E+04	5.85136E+04
5	1.01119E+05	7.65746E+04	6.72229E+04	6.21559E+04	6.76430E+04	6.72946E+04	7.08076E+04	6.82447E+04	7.03328E+04	6.75165E+04
4	9.41537E+04	8.53264E+04	6.82851E+04	6.47512E+04	6.26375E+04	6.83676E+04	6.86212E+04	7.26210E+04	7.14727E+04	7.45141E+04
3	6.76470E+04	6.14965E+04	5.45227E+04	4.55577E+04	4.72997E+04	4.75510E+04	5.30576E+04	5.36559E+04	5.82341E+04	5.90017E+04
2	2.17141E+04	2.44393E+04	1.84699E+04	1.83090E+04	1.54168E+04	1.87111E+04	1.80437E+04	2.16037E+04	2.10594E+04	2.47388E+04
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

	21	22	23	24	25
20	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
19	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
18	-4.20173E+04	-3.95628E+04	-4.89602E+04	0.00000E+00	0.00000E+00
17	-5.43289E+04	-5.50076E+04	-4.59239E+04	0.00000E+00	0.00000E+00
16	-5.11803E+04	-4.74584E+04	-4.21753E+04	0.00000E+00	0.00000E+00
15	-4.14306E+04	-4.21077E+04	-3.58245E+04	0.00000E+00	0.00000E+00
14	-3.53551E+04	-3.29576E+04	-3.01859E+04	0.00000E+00	0.00000E+00
13	-2.45155E+04	-2.62999E+04	-2.16068E+04	0.00000E+00	0.00000E+00
12	-1.67428E+04	-1.42653E+04	-1.38250E+04	0.00000E+00	0.00000E+00
11	-3.14824E+03	-5.35140E+03	-3.16415E+03	0.00000E+00	0.00000E+00
10	6.47085E+03	8.64480E+03	5.86229E+03	0.00000E+00	0.00000E+00
9	2.13287E+04	1.84967E+04	1.70935E+04	0.00000E+00	0.00000E+00
8	3.13086E+04	3.25438E+04	2.58515E+04	0.00000E+00	0.00000E+00
7	4.57018E+04	4.17765E+04	3.57721E+04	0.00000E+00	0.00000E+00
6	5.52131E+04	5.44929E+04	4.26178E+04	0.00000E+00	0.00000E+00
5	6.84051E+04	6.26005E+04	5.05126E+04	0.00000E+00	0.00000E+00
4	7.26918E+04	7.14007E+04	5.51573E+04	0.00000E+00	0.00000E+00
3	6.36010E+04	6.28781E+04	5.42047E+04	0.00000E+00	0.00000E+00
2	2.50984E+04	2.84390E+04	2.54337E+04	0.00000E+00	0.00000E+00
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

TS= 10000 YEAR= 2.28 DAY=102.8 ENERGY= 1.460152E+00 DTEMP= 2.589966E-09 DSALT= 2.511579E-13

WORK BY:	INTERNAL MODE	EXTERNAL MODE
TIME RATE OF CHANGE	-1.758143E-09	6.892828E-10
HORIZONTAL ADVECTION	-8.140633E-09	1.626149E-09
VERTICAL ADVECTION	8.634952E-08	3.923400E-08
HORIZONTAL FRICTION	-2.098318E-06	-1.287546E-06
VERTICAL FRICTION	-1.168175E-07	5.454812E-24
PRESSURE FORCES	2.286218E-07	1.064531E-06
IMPLICIT EFFECTS	4.967964E-09	-4.932260E-08
WORK BY WIND	1.962789E-06	2.829183E-07
BOTTOM DRAG	-6.121034E-08	-5.075196E-08

	TEMPERATURE	SALINITY
TIME RATE OF CHANGE	2.324253E-09	2.187360E-13
HORIZONTAL ADVECTION	4.485497E-21	-1.377910E-25
VERTICAL ADVECTION	-1.648896E-13	8.853388E-18
HORIZONTAL DIFFUSION	7.465883E-23	-2.166069E-26
SURFACE FLUX	2.421749E-24	1.325496E-28
TRUNCATION ERROR	2.324418E-09	2.187272E-13
CHANGE OF VARIANCE	3.382552E-08	3.099950E-17

WORK BY BUOYANCY FORCES 1.293153E-06 ENERGY CONVERSION ERROR -6.295149E-18 NONLINEAR EXCHANGE ERROR 1.190690E-07

NORTHWARD TRANSPORT OF HEAT (X10**15 WATTS)

	X MEAN	X EDDY	Z MEAN	Z EDDY	EKMAN	TOT ADV	DIFFUS	TOTAL
18	-0.179	-0.143	0.022	-0.344	0.000	-0.322	-0.010	-0.332
17	-0.767	-0.035	0.014	-0.026	-0.791	-0.803	0.035	-0.767
16	-1.133	0.013	-0.034	0.012	-1.098	-1.121	0.054	-1.066
15	-1.288	0.017	-0.070	0.032	-1.234	-1.271	0.042	-1.229
14	-1.328	0.015	-0.102	0.041	-1.252	-1.313	0.028	-1.285
13	-1.285	0.009	-0.125	0.040	-1.191	-1.276	0.016	-1.260
12	-1.192	0.006	-0.139	0.031	-1.079	-1.186	0.008	-1.178
11	-1.064	0.003	-0.143	0.017	-0.936	-1.061	0.000	-1.061
10	-0.916	0.001	-0.139	0.003	-0.779	-0.915	-0.006	-0.921
9	-0.760	-0.001	-0.128	-0.012	-0.621	-0.761	-0.010	-0.772
8	-0.609	-0.003	-0.114	-0.026	-0.472	-0.612	-0.013	-0.625
7	-0.463	-0.005	-0.096	-0.034	-0.338	-0.468	-0.014	-0.482
6	-0.325	-0.005	-0.078	-0.028	-0.223	-0.330	-0.009	-0.339
5	-0.208	0.000	-0.059	-0.015	-0.133	-0.208	0.007	-0.201
4	-0.116	0.004	-0.045	-0.006	-0.061	-0.112	0.028	-0.084
3	-0.042	-0.014	-0.036	-0.008	-0.012	-0.056	0.009	-0.047
2	-0.008	-0.012	-0.014	-0.006	0.000	-0.019	-0.021	-0.041

NORTHWARD TRANSPORT OF SALT (X10**10 CM**3/SEC)

	X MEAN	X EDDY	Z MEAN	Z EDDY	EKMAN	TOT ADV	DIFFUS	TOTAL
18	-0.027	-0.016	0.007	-0.050	0.000	-0.043	0.001	-0.042
17	-0.169	0.001	0.005	0.001	-0.174	-0.168	0.014	-0.154
16	-0.317	0.009	-0.009	0.007	-0.306	-0.308	0.026	-0.282
15	-0.450	0.013	-0.023	0.015	-0.430	-0.437	0.030	-0.408
14	-0.572	0.019	-0.039	0.024	-0.539	-0.553	0.032	-0.522
13	-0.676	0.026	-0.054	0.031	-0.628	-0.650	0.033	-0.617
12	-0.761	0.035	-0.068	0.033	-0.692	-0.726	0.034	-0.692
11	-0.822	0.043	-0.080	0.029	-0.728	-0.779	0.035	-0.744
10	-0.857	0.050	-0.091	0.020	-0.735	-0.807	0.036	-0.771
9	-0.862	0.054	-0.100	0.005	-0.713	-0.808	0.037	-0.771
8	-0.843	0.056	-0.106	-0.019	-0.662	-0.787	0.041	-0.746
7	-0.789	0.056	-0.109	-0.041	-0.583	-0.733	0.051	-0.682
6	-0.690	0.059	-0.108	-0.044	-0.479	-0.631	0.084	-0.547
5	-0.565	0.078	-0.097	-0.027	-0.362	-0.487	0.164	-0.322
4	-0.401	0.080	-0.101	-0.011	-0.209	-0.321	0.288	-0.033
3	-0.180	-0.038	-0.140	-0.030	-0.048	-0.218	0.270	0.052
2	-0.043	-0.061	-0.075	-0.029	0.000	-0.104	0.094	-0.010

MERIDIONAL MASS TRANSPORT

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0.00	-0.24	-1.66	-3.36	-5.33	-7.47	-9.69	-11.89	-14.03	-16.08	-17.93	-19.48	-20.65	-21.30	-21.16	-19.67	-14.34	-0.03	0.00	0.00
2	0.00	-0.60	-2.08	-3.89	-6.02	-8.33	-10.66	-12.85	-14.92	-16.91	-18.67	-20.14	-21.24	-21.85	-21.68	-20.17	-14.99	-7.45	0.00	0.00
3	0.00	-1.09	-2.70	-4.69	-7.05	-9.58	-12.03	-14.22	-16.22	-18.14	-19.84	-21.22	-22.22	-22.75	-22.41	-20.76	-15.31	-6.80	0.00	0.00
4	0.00	-1.70	-3.58	-5.86	-8.49	-11.25	-13.82	-16.01	-17.98	-19.87	-21.52	-22.85	-23.76	-24.17	-23.59	-21.67	-14.76	-5.96	0.00	0.00
5	0.00	-2.31	-4.36	-6.81	-9.59	-12.43	-15.01	-17.14	-19.03	-20.85	-22.41	-23.63	-24.40	-24.59	-23.68	-21.18	-12.78	-5.08	0.00	0.00
6	0.00	-2.96	-5.38	-8.09	-11.01	-13.91	-16.49	-18.58	-20.44	-22.19	-23.68	-24.77	-25.36	-25.25	-23.87	-20.51	-9.78	-2.52	0.00	0.00
7	0.00	-3.52	-6.10	-8.90	-11.83	-14.66	-17.15	-19.14	-20.87	-22.46	-23.75	-24.57	-24.83	-24.24	-22.31	-17.94	-6.37	-1.83	0.00	0.00
8	0.00	-3.99	-6.99	-9.99	-12.90	-15.63	-17.99	-19.84	-21.42	-22.80	-23.84	-24.32	-24.16	-22.96	-20.30	-14.66	-1.64	1.16	0.00	0.00
9	0.00	-4.23	-7.25	-10.23	-13.00	-15.54	-17.68	-19.32	-20.63	-21.71	-22.37	-22.44	-21.79	-20.11	-16.96	-10.78	1.10	-1.88	0.00	0.00
10	0.00	-4.26	-7.57	-10.56	-13.15	-15.44	-17.31	-18.66	-19.65	-20.35	-20.55	-20.11	-18.88	-16.59	-12.84	-6.03	4.51	-3.33	0.00	0.00
11	0.00	-3.97	-7.00	-9.70	-11.98	-13.94	-15.47	-16.48	-17.12	-17.44	-17.28	-16.50	-15.03	-12.67	-9.10	-3.09	3.88	-10.03	0.00	0.00
12	0.00	-3.34	-6.29	-8.69	-10.60	-12.14	-13.26	-13.87	-14.09	-13.95	-13.34	-12.17	-10.40	-7.95	-4.63	0.36	3.38	-13.13	0.00	0.00
13	0.00	-2.41	-4.63	-6.41	-7.79	-8.85	-9.55	-9.86	-9.85	-9.57	-8.95	-7.95	-6.56	-4.77	-2.49	0.41	-0.41	-17.54	0.00	0.00
14	0.00	-1.26	-2.66	-3.76	-4.52	-5.02	-5.23	-5.18	-4.90	-4.46	-3.83	-3.02	-2.07	-1.04	0.06	0.77	-2.69	-13.51	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.00	0.00	-0.01	-0.01	-0.01	-0.01	0.00	0.00	0.00	0.00	0.00	0.00

Output from Cox model 2 for comparison:

```
COX RESULTS FOR MODEL 2:

&CONTRL  NFIRST = 1,  NLAST = 10000,  NNERGY = 10000,  NMIX = 20,  NWRITE = 1000000,  NDW = 1000000,  NTSI = 100,  NA = 0,
 NB = 0,  NC = 0,  &END
&EDDY  AMF = 1000000000.,  AHF = 20000000.,  FKPMF = 1.,  FKPHF = 1.,  &END
&TSTEPS  DTTSF = 7200.,  DTUVF = 7200.,  DTSFF = 7200.,  &END
&PARMS  ACORF = 0.6,  MXSCAN = 100,  SORF = 1.6,  CRITF = 100000000.,  &END

 GRID BOX THICKNESS  'DZ'
  0.30000E+04  0.46300E+04  0.67450E+04  0.94460E+04  0.12846E+05  0.17057E+05  0.22191E+05  0.28351E+05  0.35621E+05  0.44057E+05
  0.53678E+05  0.64454E+05  0.76300E+05  0.89071E+05  0.10255E+06

 GRID POINT SEPARATION  'DZZ'
  0.15000E+04  0.38150E+04  0.56875E+04  0.80955E+04  0.11146E+05  0.14951E+05  0.19624E+05  0.25271E+05  0.31986E+05  0.39839E+05
  0.48867E+05  0.59066E+05  0.70377E+05  0.82685E+05  0.95812E+05  0.51276E+05

 DEPTH OF BOX BOTTOM  'ZDZ'
  0.30000E+04  0.76300E+04  0.14375E+05  0.23821E+05  0.36667E+05  0.53724E+05  0.75915E+05  0.10427E+06  0.13989E+06  0.18394E+06
  0.23762E+06  0.30208E+06  0.37838E+06  0.46745E+06  0.57000E+06

 DEPTH OF GRID POINT  'ZDZZ'
  0.15000E+04  0.53150E+04  0.11002E+05  0.19098E+05  0.30244E+05  0.45195E+05  0.64820E+05  0.90090E+05  0.12208E+06  0.16192E+06
  0.21078E+06  0.26985E+06  0.34023E+06  0.42291E+06  0.51872E+06  0.57000E+06

 LATITUDE OF T,S POINTS (RADIANS)  'PHIT'
 -0.96866E+00 -0.91630E+00 -0.86394E+00 -0.81158E+00 -0.75922E+00 -0.70686E+00 -0.65450E+00 -0.60214E+00 -0.54978E+00 -0.49742E+00
 -0.44506E+00 -0.39270E+00 -0.34034E+00 -0.28798E+00 -0.23562E+00 -0.18326E+00 -0.13090E+00 -0.78540E-01 -0.26180E-01  0.26180E-01

 LATITUDE OF U,V POINTS (RADIANS)  'PHI'
 -0.94248E+00 -0.89012E+00 -0.83776E+00 -0.78540E+00 -0.73304E+00 -0.68068E+00 -0.62832E+00 -0.57596E+00 -0.52360E+00 -0.47124E+00
 -0.41888E+00 -0.36652E+00 -0.31416E+00 -0.26180E+00 -0.20944E+00 -0.15708E+00 -0.10472E+00 -0.52360E-01 -0.29088E-13  0.52360E-01

 COSINE OF T,S LATITUDE  'CST'
  0.56641E+00  0.60876E+00  0.64945E+00  0.68835E+00  0.72537E+00  0.76041E+00  0.79335E+00  0.82413E+00  0.85264E+00  0.87882E+00
  0.90259E+00  0.92388E+00  0.94264E+00  0.95882E+00  0.97237E+00  0.98325E+00  0.99144E+00  0.99692E+00  0.99966E+00  0.99966E+00

 COSINE OF U,V LATITUDE  'CS'
  0.58779E+00  0.62932E+00  0.66913E+00  0.70711E+00  0.74314E+00  0.77715E+00  0.80902E+00  0.83867E+00  0.86603E+00  0.89101E+00
  0.91355E+00  0.93358E+00  0.95106E+00  0.96593E+00  0.97815E+00  0.98769E+00  0.99452E+00  0.99863E+00  0.10000E+01  0.99863E+00

 SINE OF U,V LATITUDE  'SINE'
 -0.80902E+00 -0.77715E+00 -0.74314E+00 -0.70711E+00 -0.66913E+00 -0.62932E+00 -0.58779E+00 -0.54464E+00 -0.50000E+00 -0.45399E+00
 -0.40674E+00 -0.35837E+00 -0.30902E+00 -0.25882E+00 -0.20791E+00 -0.15643E+00 -0.10453E+00 -0.52336E-01 -0.29088E-13  0.52336E-01
```

NUMBER OF LEVELS AT T,S POINTS AND U,V POINTS

```
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 4 6 8 A C E F F F F F F F F F F F F F F F F 0
0 4 6 8 A C E F F F F F F F F F F F F F F F F F 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

SURFACE AREA = 3.122914E+17   VOLUME = 1.518226E+23

```
 START AND END INDICES FOR STREAM FUNCTION
J= 20        0   0
J= 19        3  23
J= 18        3  23
J= 17        3  23
J= 16        3  23
J= 15        3  23
J= 14        3  23
J= 13        3  23
J= 12        3  23
J= 11        3  23
J= 10        3  23
J=  9        3  23
J=  8        3  23
J=  7        3  23
J=  6        3  23
J=  5        3  23
J=  4        3  23
J=  3        3  23
J=  2        0   0
J=  1        0   0
&TSPROF  TINITF = 15*4.,  15*3.49E-2,  &END
TS=   100  YEAR=   0.02  DAY=  8.3  ENERGY= 5.470501E-01  DTEMP= 1.542271E-08  DSALT= 6.839375E-13  SCANS= 24
TS=   200  YEAR=   0.05  DAY= 16.7  ENERGY= 6.903798E-01  DTEMP= 1.361511E-08  DSALT= 6.223775E-13  SCANS= 17
TS=   300  YEAR=   0.07  DAY= 25.0  ENERGY= 7.445145E-01  DTEMP= 1.207407E-08  DSALT= 5.736518E-13  SCANS= 15
TS=   400  YEAR=   0.09  DAY= 33.3  ENERGY= 7.844634E-01  DTEMP= 1.080457E-08  DSALT= 5.349325E-13  SCANS= 15
TS=   500  YEAR=   0.11  DAY= 41.7  ENERGY= 8.265022E-01  DTEMP= 9.721377E-09  DSALT= 4.974838E-13  SCANS= 10
TS=   600  YEAR=   0.14  DAY= 50.0  ENERGY= 8.677011E-01  DTEMP= 8.812870E-09  DSALT= 4.643967E-13  SCANS= 10
TS=   700  YEAR=   0.16  DAY= 58.3  ENERGY= 9.042804E-01  DTEMP= 8.070132E-09  DSALT= 4.372552E-13  SCANS=  7
TS=   800  YEAR=   0.18  DAY= 66.7  ENERGY= 9.336020E-01  DTEMP= 7.471608E-09  DSALT= 4.156034E-13  SCANS=  6
TS=   900  YEAR=   0.21  DAY= 75.0  ENERGY= 9.550519E-01  DTEMP= 6.983172E-09  DSALT= 3.998542E-13  SCANS=  5
TS=  1000  YEAR=   0.23  DAY= 83.3  ENERGY= 9.691335E-01  DTEMP= 6.573797E-09  DSALT= 3.859865E-13  SCANS=  4
TS=  1100  YEAR=   0.25  DAY= 91.7  ENERGY= 9.764347E-01  DTEMP= 6.229803E-09  DSALT= 3.745452E-13  SCANS=  4
TS=  1200  YEAR=   0.27  DAY=100.0  ENERGY= 9.780403E-01  DTEMP= 5.951203E-09  DSALT= 3.647818E-13  SCANS=  3
TS=  1300  YEAR=   0.30  DAY=108.3  ENERGY= 9.754908E-01  DTEMP= 5.741614E-09  DSALT= 3.572659E-13  SCANS=  4
TS=  1400  YEAR=   0.32  DAY=116.7  ENERGY= 9.701260E-01  DTEMP= 5.560742E-09  DSALT= 3.513468E-13  SCANS=  3
TS=  1500  YEAR=   0.34  DAY=125.0  ENERGY= 9.632596E-01  DTEMP= 5.392267E-09  DSALT= 3.470733E-13  SCANS=  3
TS=  1600  YEAR=   0.37  DAY=133.3  ENERGY= 9.553733E-01  DTEMP= 5.232120E-09  DSALT= 3.425889E-13  SCANS=  3
TS=  1700  YEAR=   0.39  DAY=141.7  ENERGY= 9.470843E-01  DTEMP= 5.091471E-09  DSALT= 3.376177E-13  SCANS=  3
TS=  1800  YEAR=   0.41  DAY=150.0  ENERGY= 9.385288E-01  DTEMP= 4.993616E-09  DSALT= 3.332910E-13  SCANS=  3
TS=  1900  YEAR=   0.43  DAY=158.3  ENERGY= 9.301391E-01  DTEMP= 4.933548E-09  DSALT= 3.355347E-13  SCANS=  3
TS=  2000  YEAR=   0.46  DAY=166.7  ENERGY= 9.219918E-01  DTEMP= 4.798156E-09  DSALT= 3.254242E-13  SCANS=  3
TS=  2100  YEAR=   0.48  DAY=175.0  ENERGY= 9.144417E-01  DTEMP= 4.692057E-09  DSALT= 3.220809E-13  SCANS=  3
TS=  2200  YEAR=   0.50  DAY=183.3  ENERGY= 9.074322E-01  DTEMP= 4.594836E-09  DSALT= 3.200374E-13  SCANS=  4
TS=  2300  YEAR=   0.52  DAY=191.7  ENERGY= 9.011144E-01  DTEMP= 4.517383E-09  DSALT= 3.174522E-13  SCANS=  4
TS=  2400  YEAR=   0.55  DAY=200.0  ENERGY= 8.956002E-01  DTEMP= 4.451941E-09  DSALT= 3.154122E-13  SCANS=  4
TS=  2500  YEAR=   0.57  DAY=208.3  ENERGY= 8.907814E-01  DTEMP= 4.392683E-09  DSALT= 3.142719E-13  SCANS=  4
TS=  2600  YEAR=   0.59  DAY=216.7  ENERGY= 8.867854E-01  DTEMP= 4.325803E-09  DSALT= 3.118243E-13  SCANS=  4
TS=  2700  YEAR=   0.62  DAY=225.0  ENERGY= 8.836999E-01  DTEMP= 4.255433E-09  DSALT= 3.090558E-13  SCANS=  4
TS=  2800  YEAR=   0.64  DAY=233.3  ENERGY= 8.815718E-01  DTEMP= 4.187956E-09  DSALT= 3.064213E-13  SCANS=  4
TS=  2900  YEAR=   0.66  DAY=241.7  ENERGY= 8.804458E-01  DTEMP= 4.137479E-09  DSALT= 3.049890E-13  SCANS=  4
TS=  3000  YEAR=   0.68  DAY=250.0  ENERGY= 8.803429E-01  DTEMP= 4.094681E-09  DSALT= 3.044281E-13  SCANS=  4
TS=  3100  YEAR=   0.71  DAY=258.3  ENERGY= 8.812314E-01  DTEMP= 4.048837E-09  DSALT= 3.036107E-13  SCANS=  4
TS=  3200  YEAR=   0.73  DAY=266.7  ENERGY= 8.831196E-01  DTEMP= 3.996742E-09  DSALT= 3.015809E-13  SCANS=  4
TS=  3300  YEAR=   0.75  DAY=275.0  ENERGY= 8.860780E-01  DTEMP= 3.943468E-09  DSALT= 2.990750E-13  SCANS=  4
TS=  3400  YEAR=   0.78  DAY=283.3  ENERGY= 8.901730E-01  DTEMP= 3.910221E-09  DSALT= 3.001233E-13  SCANS=  4
TS=  3500  YEAR=   0.80  DAY=291.7  ENERGY= 8.954584E-01  DTEMP= 3.864470E-09  DSALT= 2.983021E-13  SCANS=  4
TS=  3600  YEAR=   0.82  DAY=300.0  ENERGY= 9.019876E-01  DTEMP= 3.866879E-09  DSALT= 3.033875E-13  SCANS=  4
TS=  3700  YEAR=   0.84  DAY=308.3  ENERGY= 9.097938E-01  DTEMP= 3.852663E-09  DSALT= 3.055742E-13  SCANS=  4
TS=  3800  YEAR=   0.87  DAY=316.7  ENERGY= 9.189396E-01  DTEMP= 3.813183E-09  DSALT= 3.036122E-13  SCANS=  4
TS=  3900  YEAR=   0.89  DAY=325.0  ENERGY= 9.295367E-01  DTEMP= 3.772334E-09  DSALT= 3.016911E-13  SCANS=  5
TS=  4000  YEAR=   0.91  DAY=333.3  ENERGY= 9.415550E-01  DTEMP= 5.120508E-09  DSALT= 5.276898E-13  SCANS=  4
TS=  4100  YEAR=   0.94  DAY=341.7  ENERGY= 9.550726E-01  DTEMP= 3.929566E-09  DSALT= 3.365863E-13  SCANS=  5
TS=  4200  YEAR=   0.96  DAY=350.0  ENERGY= 9.701834E-01  DTEMP= 3.665354E-09  DSALT= 2.983372E-13  SCANS=  5
TS=  4300  YEAR=   0.98  DAY=358.3  ENERGY= 9.869805E-01  DTEMP= 3.638639E-09  DSALT= 2.986854E-13  SCANS=  4
TS=  4400  YEAR=   1.00  DAY=  1.4  ENERGY= 1.005637E+00  DTEMP= 3.647582E-09  DSALT= 3.043312E-13  SCANS=  5
```

```
TS=  4500  YEAR=   1.03  DAY=   9.7  ENERGY= 1.026235E+00  DTEMP= 3.615559E-09  DSALT= 3.018346E-13  SCANS=  5
TS=  4600  YEAR=   1.05  DAY=  18.1  ENERGY= 1.048861E+00  DTEMP= 3.579486E-09  DSALT= 2.994179E-13  SCANS=  5
TS=  4700  YEAR=   1.07  DAY=  26.4  ENERGY= 1.073593E+00  DTEMP= 3.546539E-09  DSALT= 2.970962E-13  SCANS=  5
TS=  4800  YEAR=   1.10  DAY=  34.7  ENERGY= 1.100452E+00  DTEMP= 3.515020E-09  DSALT= 2.955992E-13  SCANS=  5
TS=  4900  YEAR=   1.12  DAY=  43.1  ENERGY= 1.129534E+00  DTEMP= 3.494384E-09  DSALT= 2.956330E-13  SCANS=  5
TS=  5000  YEAR=   1.14  DAY=  51.4  ENERGY= 1.160661E+00  DTEMP= 3.462242E-09  DSALT= 2.933426E-13  SCANS=  5
TS=  5100  YEAR=   1.16  DAY=  59.7  ENERGY= 1.193617E+00  DTEMP= 3.425414E-09  DSALT= 2.900455E-13  SCANS=  5
TS=  5200  YEAR=   1.19  DAY=  68.1  ENERGY= 1.228099E+00  DTEMP= 3.404349E-09  DSALT= 2.894064E-13  SCANS=  5
TS=  5300  YEAR=   1.21  DAY=  76.4  ENERGY= 1.263594E+00  DTEMP= 3.428555E-09  DSALT= 2.954241E-13  SCANS=  5
TS=  5400  YEAR=   1.23  DAY=  84.7  ENERGY= 1.299553E+00  DTEMP= 3.423397E-09  DSALT= 2.970459E-13  SCANS=  5
TS=  5500  YEAR=   1.25  DAY=  93.1  ENERGY= 1.334946E+00  DTEMP= 3.405797E-09  DSALT= 2.966071E-13  SCANS=  5
TS=  5600  YEAR=   1.28  DAY=101.4  ENERGY= 1.368570E+00  DTEMP= 3.368616E-09  DSALT= 2.929924E-13  SCANS=  5
TS=  5700  YEAR=   1.30  DAY=109.7  ENERGY= 1.399816E+00  DTEMP= 3.332775E-09  DSALT= 2.900374E-13  SCANS=  4
TS=  5800  YEAR=   1.32  DAY=118.1  ENERGY= 1.428154E+00  DTEMP= 3.300040E-09  DSALT= 2.876340E-13  SCANS=  5
TS=  5900  YEAR=   1.35  DAY=126.4  ENERGY= 1.452855E+00  DTEMP= 5.014573E-09  DSALT= 5.705171E-13  SCANS=  4
TS=  6000  YEAR=   1.37  DAY=134.7  ENERGY= 1.473785E+00  DTEMP= 3.245273E-09  DSALT= 2.836626E-13  SCANS=  5
TS=  6100  YEAR=   1.39  DAY=143.1  ENERGY= 1.491094E+00  DTEMP= 3.252372E-09  DSALT= 2.895206E-13  SCANS=  4
TS=  6200  YEAR=   1.41  DAY=151.4  ENERGY= 1.505121E+00  DTEMP= 3.224977E-09  DSALT= 2.879178E-13  SCANS=  4
TS=  6300  YEAR=   1.44  DAY=159.7  ENERGY= 1.516323E+00  DTEMP= 3.201822E-09  DSALT= 2.861406E-13  SCANS=  4
TS=  6400  YEAR=   1.46  DAY=168.1  ENERGY= 1.525174E+00  DTEMP= 3.188964E-09  DSALT= 2.858914E-13  SCANS=  3
TS=  6500  YEAR=   1.48  DAY=176.4  ENERGY= 1.532178E+00  DTEMP= 3.227765E-09  DSALT= 2.946317E-13  SCANS=  3
TS=  6600  YEAR=   1.51  DAY=184.7  ENERGY= 1.537876E+00  DTEMP= 3.193116E-09  DSALT= 2.909139E-13  SCANS=  3
TS=  6700  YEAR=   1.53  DAY=193.1  ENERGY= 1.542509E+00  DTEMP= 3.162680E-09  DSALT= 2.881036E-13  SCANS=  3
TS=  6800  YEAR=   1.55  DAY=201.4  ENERGY= 1.546205E+00  DTEMP= 3.136944E-09  DSALT= 2.858409E-13  SCANS=  3
TS=  6900  YEAR=   1.57  DAY=209.7  ENERGY= 1.549168E+00  DTEMP= 3.105212E-09  DSALT= 2.828078E-13  SCANS=  3
TS=  7000  YEAR=   1.60  DAY=218.1  ENERGY= 1.551575E+00  DTEMP= 3.082357E-09  DSALT= 2.808701E-13  SCANS=  2
TS=  7100  YEAR=   1.62  DAY=226.4  ENERGY= 1.553569E+00  DTEMP= 3.062660E-09  DSALT= 2.794906E-13  SCANS=  2
TS=  7200  YEAR=   1.64  DAY=234.7  ENERGY= 1.555293E+00  DTEMP= 3.091614E-09  DSALT= 2.866094E-13  SCANS=  4
TS=  7300  YEAR=   1.67  DAY=243.1  ENERGY= 1.556768E+00  DTEMP= 3.060205E-09  DSALT= 2.830617E-13  SCANS=  4
TS=  7400  YEAR=   1.69  DAY=251.4  ENERGY= 1.558013E+00  DTEMP= 5.240505E-09  DSALT= 6.560927E-13  SCANS=  5
TS=  7500  YEAR=   1.71  DAY=259.7  ENERGY= 1.559222E+00  DTEMP= 3.044761E-09  DSALT= 2.850865E-13  SCANS=  3
TS=  7600  YEAR=   1.73  DAY=268.1  ENERGY= 1.560493E+00  DTEMP= 3.021072E-09  DSALT= 2.830016E-13  SCANS=  3
TS=  7700  YEAR=   1.76  DAY=276.4  ENERGY= 1.561705E+00  DTEMP= 3.007542E-09  DSALT= 2.825999E-13  SCANS=  3
TS=  7800  YEAR=   1.78  DAY=284.8  ENERGY= 1.562788E+00  DTEMP= 2.987905E-09  DSALT= 2.810414E-13  SCANS=  2
TS=  7900  YEAR=   1.80  DAY=293.1  ENERGY= 1.563752E+00  DTEMP= 2.968735E-09  DSALT= 2.800111E-13  SCANS=  3
TS=  8000  YEAR=   1.83  DAY=301.4  ENERGY= 1.564617E+00  DTEMP= 2.943171E-09  DSALT= 2.778728E-13  SCANS=  3
TS=  8100  YEAR=   1.85  DAY=309.8  ENERGY= 1.565389E+00  DTEMP= 3.646563E-09  DSALT= 4.236907E-13  SCANS=  3
TS=  8200  YEAR=   1.87  DAY=318.1  ENERGY= 1.566077E+00  DTEMP= 2.906041E-09  DSALT= 2.751833E-13  SCANS=  3
TS=  8300  YEAR=   1.89  DAY=326.4  ENERGY= 1.566687E+00  DTEMP= 2.903768E-09  DSALT= 2.767301E-13  SCANS=  3
TS=  8400  YEAR=   1.92  DAY=334.8  ENERGY= 1.567225E+00  DTEMP= 2.890648E-09  DSALT= 2.761581E-13  SCANS=  3
TS=  8500  YEAR=   1.94  DAY=343.1  ENERGY= 1.567757E+00  DTEMP= 2.874199E-09  DSALT= 2.750062E-13  SCANS=  3
TS=  8600  YEAR=   1.96  DAY=351.4  ENERGY= 1.568248E+00  DTEMP= 2.853234E-09  DSALT= 2.726868E-13  SCANS=  2
TS=  8700  YEAR=   1.98  DAY=359.8  ENERGY= 1.568652E+00  DTEMP= 2.825135E-09  DSALT= 2.690651E-13  SCANS=  3
TS=  8800  YEAR=   2.01  DAY=   2.8  ENERGY= 1.568967E+00  DTEMP= 2.799366E-09  DSALT= 2.658121E-13  SCANS=  3
TS=  8900  YEAR=   2.03  DAY=  11.2  ENERGY= 1.569212E+00  DTEMP= 2.783789E-09  DSALT= 2.648092E-13  SCANS=  3
TS=  9000  YEAR=   2.05  DAY=  19.5  ENERGY= 1.569346E+00  DTEMP= 2.786773E-09  DSALT= 2.677426E-13  SCANS=  3
TS=  9100  YEAR=   2.08  DAY=  27.8  ENERGY= 1.569466E+00  DTEMP= 2.768639E-09  DSALT= 2.661737E-13  SCANS=  3
TS=  9200  YEAR=   2.10  DAY=  36.2  ENERGY= 1.569464E+00  DTEMP= 2.748378E-09  DSALT= 2.640238E-13  SCANS=  3
TS=  9300  YEAR=   2.12  DAY=  44.5  ENERGY= 1.569333E+00  DTEMP= 2.732527E-09  DSALT= 2.626566E-13  SCANS=  3
TS=  9400  YEAR=   2.14  DAY=  52.8  ENERGY= 1.569062E+00  DTEMP= 2.738651E-09  DSALT= 2.657662E-13  SCANS=  3
TS=  9500  YEAR=   2.17  DAY=  61.2  ENERGY= 1.568643E+00  DTEMP= 2.707877E-09  DSALT= 2.614909E-13  SCANS=  3
TS=  9600  YEAR=   2.19  DAY=  69.5  ENERGY= 1.568066E+00  DTEMP= 2.681621E-09  DSALT= 2.582510E-13  SCANS=  2
TS=  9700  YEAR=   2.21  DAY=  77.8  ENERGY= 1.567323E+00  DTEMP= 2.667787E-09  DSALT= 2.570054E-13  SCANS=  2
TS=  9800  YEAR=   2.24  DAY=  86.2  ENERGY= 1.566408E+00  DTEMP= 2.660615E-09  DSALT= 2.571906E-13  SCANS=  2
TS=  9900  YEAR=   2.26  DAY=  94.5  ENERGY= 1.565318E+00  DTEMP= 2.632614E-09  DSALT= 2.533119E-13  SCANS=  2
TEMPERATURE FOR J =    7 AT TIMESTEP   10000
       2      3      4      5      6      7      8      9     10     11     12     13     14     15     16     17     18     19     20     21

  1  12.46  13.02  12.76  12.58  12.65  12.52  12.44  12.42  12.43  12.46  12.47  12.47  12.47  12.47  12.47  12.47  12.47  12.47  12.47  12.49
  2  11.42  12.19  12.76  12.58  12.10  11.79  11.56  11.44  11.40  11.41  11.43  11.44  11.44  11.44  11.44  11.44  11.44  11.44  11.45  11.47
  3   8.88   9.70  10.40  10.49  10.26   9.93   9.64   9.47   9.40   9.38   9.38   9.38   9.38   9.38   9.38   9.38   9.38   9.38   9.38   9.42
  4   6.12   6.50   7.11   7.28   7.16   6.94   6.74   6.63   6.57   6.55   6.55   6.55   6.55   6.55   6.54   6.54   6.54   6.54   6.55   6.58
  5   0.00   4.64   4.88   4.96   4.91   4.81   4.74   4.71   4.70   4.70   4.70   4.70   4.70   4.70   4.70   4.70   4.70   4.70   4.70   4.71
  6   0.00   4.09   4.13   4.16   4.15   4.13   4.12   4.13   4.13   4.14   4.14   4.14   4.14   4.14   4.14   4.14   4.14   4.14   4.14   4.14
  7   0.00   0.00   4.01   4.02   4.02   4.02   4.03   4.04   4.05   4.05   4.06   4.06   4.06   4.06   4.06   4.06   4.06   4.06   4.06   4.06
  8   0.00   0.00   4.00   4.00   4.00   4.01   4.01   4.02   4.03   4.04   4.04   4.05   4.05   4.05   4.05   4.05   4.05   4.05   4.05   4.05
```

9	0.00	0.00	0.00	4.00	4.00	4.00	4.01	4.01	4.02	4.02	4.02	4.02	4.02	4.02	4.02	4.02	4.02	4.02	4.02	4.02
10	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.01	4.01	4.01	4.01	4.01	4.01	4.01	4.01	4.01	4.01	4.01	4.01
11	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
12	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
13	0.00	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
14	0.00	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
15	0.00	0.00	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00

SALINITY FOR J = 7 AT TIMESTEP 10000

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0.55	0.54	0.48	0.50	0.58	0.58	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59
2	0.34	0.35	0.48	0.50	0.48	0.50	0.51	0.52	0.52	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51
3	0.14	0.16	0.23	0.28	0.31	0.33	0.34	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35
4	0.00	0.01	0.05	0.08	0.10	0.11	0.11	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12	0.12
5	10.00	-0.07	-0.06	-0.05	-0.04	-0.04	-0.04	-0.04	-0.04	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03
6	10.00	-0.10	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08
7	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09
8	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09
9	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
10	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
11	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
12	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
13	10.00	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
14	10.00	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
15	10.00	10.00	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10

W VELOCITY FOR J = 7 AT TIMESTEP 10000

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	-0.01	-0.09	-0.15	-0.15	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14
3	0.06	-0.07	-0.17	-0.16	-0.15	-0.14	-0.13	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14
4	0.08	-0.04	-0.20	-0.19	-0.16	-0.14	-0.13	-0.13	-0.13	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14	-0.14
5	0.00	0.01	-0.25	-0.22	-0.18	-0.14	-0.12	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.14
6	0.00	0.02	-0.21	-0.27	-0.21	-0.15	-0.11	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13
7	0.00	0.00	-0.15	-0.34	-0.25	-0.17	-0.10	-0.12	-0.12	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13	-0.13
8	0.00	0.00	-0.08	-0.29	-0.30	-0.19	-0.09	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12
9	0.00	0.00	0.00	-0.23	-0.37	-0.22	-0.08	-0.12	-0.11	-0.11	-0.11	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12
10	0.00	0.00	0.00	-0.13	-0.30	-0.25	-0.06	-0.11	-0.10	-0.10	-0.11	-0.11	-0.11	-0.11	-0.11	-0.11	-0.11	-0.11	-0.11	-0.11
11	0.00	0.00	0.00	0.00	-0.21	-0.29	-0.04	-0.11	-0.09	-0.09	-0.09	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
12	0.00	0.00	0.00	0.00	-0.11	-0.18	-0.01	-0.10	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08
13	0.00	0.00	0.00	0.00	0.00	-0.05	0.02	-0.09	-0.06	-0.07	-0.07	-0.07	-0.07	-0.07	-0.07	-0.07	-0.07	-0.07	-0.07	-0.07
14	0.00	0.00	0.00	0.00	0.00	-0.03	0.18	-0.08	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05
15	0.00	0.00	0.00	0.00	0.00	0.00	0.38	-0.07	-0.03	-0.02	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

U VELOCITY FOR J = 7 AT TIMESTEP 10000

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0.00	0.51	1.23	1.34	1.22	1.09	0.96	0.89	0.89	0.94	1.01	1.05	1.07	1.07	1.08	1.09	1.11	1.13	1.15	1.19
2	0.00	0.47	0.90	1.01	0.91	0.79	0.66	0.58	0.57	0.61	0.68	0.72	0.74	0.74	0.75	0.76	0.78	0.79	0.82	0.85
3	0.00	0.20	0.56	0.65	0.56	0.46	0.33	0.24	0.20	0.23	0.29	0.33	0.35	0.36	0.36	0.38	0.39	0.41	0.43	0.45
4	0.00	-0.01	0.34	0.38	0.30	0.21	0.10	0.00	-0.06	-0.05	-0.01	0.03	0.05	0.06	0.07	0.08	0.10	0.11	0.13	0.15
5	0.00	0.00	0.08	0.25	0.17	0.11	0.00	-0.11	-0.18	-0.18	-0.15	-0.11	-0.09	-0.08	-0.07	-0.06	-0.05	-0.03	-0.02	0.00
6	0.00	0.00	0.05	0.21	0.14	0.08	-0.03	-0.13	-0.21	-0.22	-0.19	-0.15	-0.13	-0.12	-0.11	-0.10	-0.09	-0.07	-0.06	-0.04
7	0.00	0.00	0.00	0.10	0.13	0.07	-0.03	-0.14	-0.22	-0.23	-0.20	-0.17	-0.15	-0.14	-0.13	-0.11	-0.10	-0.08	-0.07	-0.06
8	0.00	0.00	0.00	0.09	0.13	0.07	-0.03	-0.14	-0.23	-0.24	-0.21	-0.17	-0.15	-0.14	-0.13	-0.12	-0.10	-0.09	-0.08	-0.06
9	0.00	0.00	0.00	0.00	0.05	0.07	-0.03	-0.14	-0.23	-0.24	-0.21	-0.18	-0.16	-0.15	-0.14	-0.12	-0.11	-0.09	-0.08	-0.07
10	0.00	0.00	0.00	0.00	0.05	0.07	-0.03	-0.14	-0.23	-0.24	-0.21	-0.18	-0.16	-0.15	-0.14	-0.13	-0.11	-0.10	-0.08	-0.07

11	0.00	0.00	0.00	0.00	0.00	0.02	-0.03	-0.15	-0.23	-0.24	-0.21	-0.18	-0.16	-0.15	-0.14	-0.13	-0.11	-0.10	-0.08	-0.07
12	0.00	0.00	0.00	0.00	0.00	0.02	-0.03	-0.15	-0.23	-0.25	-0.21	-0.18	-0.16	-0.15	-0.14	-0.13	-0.11	-0.10	-0.09	-0.07
13	0.00	0.00	0.00	0.00	0.00	0.00	-0.06	-0.15	-0.23	-0.25	-0.22	-0.18	-0.16	-0.15	-0.14	-0.13	-0.12	-0.10	-0.09	-0.07
14	0.00	0.00	0.00	0.00	0.00	0.00	-0.06	-0.15	-0.23	-0.25	-0.22	-0.18	-0.16	-0.15	-0.14	-0.13	-0.12	-0.10	-0.09	-0.07
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.15	-0.23	-0.25	-0.22	-0.18	-0.16	-0.15	-0.14	-0.13	-0.12	-0.10	-0.09	-0.07

V VELOCITY FOR J = 7 AT TIMESTEP 10000

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0.00	-12.51	-10.54	-8.32	-6.96	-6.25	-5.86	-5.79	-6.02	-6.28	-6.39	-6.40	-6.36	-6.35	-6.34	-6.35	-6.35	-6.36	-6.37	-6.43
2	0.00	-5.71	-4.04	-1.99	-0.65	0.09	0.52	0.65	0.45	0.20	0.09	0.07	0.10	0.12	0.12	0.12	0.12	0.11	0.10	0.05
3	0.00	-5.21	-3.85	-2.13	-0.94	-0.19	0.32	0.56	0.43	0.22	0.10	0.08	0.10	0.12	0.12	0.12	0.12	0.11	0.11	0.07
4	0.00	-4.77	-3.48	-2.18	-1.20	-0.49	0.09	0.44	0.40	0.22	0.11	0.08	0.10	0.11	0.12	0.12	0.12	0.11	0.11	0.09
5	0.00	0.00	-3.19	-2.17	-1.36	-0.69	-0.06	0.36	0.37	0.21	0.11	0.08	0.10	0.11	0.12	0.12	0.12	0.11	0.11	0.11
6	0.00	0.00	-3.07	-2.15	-1.42	-0.76	-0.12	0.34	0.35	0.21	0.11	0.08	0.10	0.11	0.12	0.12	0.12	0.11	0.12	0.11
7	0.00	0.00	0.00	-2.13	-1.44	-0.78	-0.14	0.33	0.35	0.21	0.11	0.08	0.10	0.11	0.12	0.12	0.12	0.11	0.12	0.11
8	0.00	0.00	0.00	-2.13	-1.44	-0.79	-0.14	0.32	0.35	0.21	0.11	0.08	0.10	0.11	0.12	0.12	0.12	0.11	0.12	0.11
9	0.00	0.00	0.00	0.00	-1.44	-0.79	-0.14	0.32	0.35	0.21	0.11	0.08	0.10	0.11	0.12	0.12	0.12	0.11	0.12	0.11
10	0.00	0.00	0.00	0.00	-1.44	-0.79	-0.14	0.32	0.35	0.21	0.11	0.08	0.10	0.11	0.12	0.12	0.12	0.11	0.12	0.11
11	0.00	0.00	0.00	0.00	0.00	-0.79	-0.14	0.32	0.35	0.21	0.11	0.08	0.10	0.11	0.12	0.12	0.12	0.11	0.12	0.11
12	0.00	0.00	0.00	0.00	0.00	-0.79	-0.14	0.32	0.35	0.21	0.11	0.08	0.10	0.11	0.12	0.12	0.12	0.11	0.12	0.11
13	0.00	0.00	0.00	0.00	0.00	0.00	-0.14	0.32	0.35	0.20	0.11	0.08	0.10	0.11	0.12	0.12	0.12	0.11	0.12	0.11
14	0.00	0.00	0.00	0.00	0.00	0.00	-0.14	0.32	0.35	0.20	0.11	0.08	0.10	0.11	0.12	0.12	0.12	0.11	0.12	0.11
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.32	0.34	0.20	0.11	0.08	0.10	0.11	0.12	0.12	0.12	0.11	0.12	0.11

TEMPERATURE FOR J = 14 AT TIMESTEP 10000

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	18.32	17.73	17.79	17.99	18.02	17.99	17.96	17.94	17.95	17.98	18.02	18.06	18.12	18.18	18.27	18.37	18.47	18.57	18.66	18.70
2	14.63	13.84	14.19	14.56	14.66	14.66	14.67	14.64	14.63	14.65	14.69	14.73	14.77	14.84	14.94	15.09	15.27	15.45	15.59	15.60
3	9.50	9.06	9.53	9.92	10.05	10.08	10.14	10.13	10.11	10.12	10.15	10.18	10.21	10.26	10.36	10.51	10.72	10.95	11.11	11.11
4	5.65	5.58	5.87	5.95	5.93	5.94	6.04	6.08	6.09	6.09	6.10	6.12	6.13	6.16	6.22	6.31	6.44	6.59	6.71	6.70
5	0.00	4.40	4.42	4.35	4.28	4.26	4.30	4.34	4.35	4.36	4.36	4.36	4.37	4.38	4.39	4.42	4.46	4.50	4.54	4.53
6	0.00	4.04	4.05	4.03	4.02	4.01	4.02	4.03	4.03	4.03	4.03	4.03	4.03	4.03	4.03	4.04	4.04	4.05	4.05	4.05
7	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
8	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
9	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
10	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
11	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
12	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
13	0.00	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
14	0.00	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
15	0.00	0.00	0.00	0.00	0.00	0.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00

SALINITY FOR J = 14 AT TIMESTEP 10000

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0.20	0.16	0.17	0.18	0.19	0.19	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.17	0.18	0.18	0.18	0.18
2	0.09	0.07	0.09	0.12	0.14	0.14	0.13	0.12	0.11	0.11	0.12	0.12	0.12	0.11	0.11	0.11	0.11	0.11	0.12	0.12
3	-0.01	-0.02	0.00	0.03	0.04	0.05	0.04	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.04
4	-0.08	-0.08	-0.07	-0.06	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05
5	10.00	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09	-0.09
6	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
7	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
8	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
9	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
10	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
11	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
12	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
13	10.00	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10

14	10.00	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10
15	10.00	10.00	10.00	10.00	10.00	10.00	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10	-0.10

W VELOCITY FOR J = 14 AT TIMESTEP 10000

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	-0.08	-0.02	-0.07	-0.04	-0.03	-0.02	-0.03	-0.03	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.05	-0.05	-0.04	-0.04	-0.03
3	-0.09	-0.12	-0.13	-0.05	-0.01	0.00	-0.01	-0.03	-0.04	-0.04	-0.04	-0.04	-0.04	-0.05	-0.05	-0.06	-0.06	-0.06	-0.04	-0.02
4	-0.08	-0.22	-0.21	-0.06	0.01	0.04	0.01	-0.02	-0.04	-0.04	-0.04	-0.04	-0.04	-0.05	-0.06	-0.07	-0.07	-0.07	-0.05	-0.01
5	0.00	-0.27	-0.34	-0.08	0.06	0.10	0.04	-0.02	-0.04	-0.04	-0.04	-0.04	-0.04	-0.05	-0.06	-0.07	-0.08	-0.07	-0.05	0.00
6	0.00	-0.18	-0.34	-0.11	0.12	0.20	0.09	-0.01	-0.04	-0.04	-0.04	-0.04	-0.04	-0.05	-0.06	-0.07	-0.08	-0.08	-0.05	0.00
7	0.00	0.00	-0.33	-0.16	0.20	0.32	0.15	0.00	-0.04	-0.04	-0.04	-0.04	-0.04	-0.05	-0.06	-0.07	-0.08	-0.07	-0.05	0.00
8	0.00	0.00	-0.18	-0.01	0.30	0.48	0.23	0.02	-0.04	-0.04	-0.04	-0.04	-0.04	-0.05	-0.06	-0.07	-0.08	-0.07	-0.04	0.00
9	0.00	0.00	0.00	0.18	0.43	0.69	0.33	0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.05	-0.06	-0.07	-0.07	-0.07	-0.04	0.00
10	0.00	0.00	0.00	0.10	0.76	0.96	0.45	0.06	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.05	-0.06	-0.07	-0.06	-0.04	0.00
11	0.00	0.00	0.00	0.00	1.16	1.28	0.61	0.09	-0.04	-0.04	-0.03	-0.03	-0.03	-0.04	-0.05	-0.06	-0.06	-0.06	-0.04	0.00
12	0.00	0.00	0.00	0.00	0.63	1.68	0.80	0.13	-0.04	-0.04	-0.03	-0.03	-0.03	-0.04	-0.04	-0.05	-0.05	-0.05	-0.03	0.00
13	0.00	0.00	0.00	0.00	0.00	2.15	1.03	0.17	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.04	-0.04	-0.04	-0.03	0.00
14	0.00	0.00	0.00	0.00	0.00	1.15	1.07	0.23	-0.03	-0.03	-0.02	-0.02	-0.02	-0.02	-0.03	-0.03	-0.03	-0.03	-0.02	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	1.11	0.30	-0.02	-0.02	-0.01	-0.01	-0.01	-0.01	-0.01	-0.02	-0.02	-0.02	-0.01	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

U VELOCITY FOR J = 14 AT TIMESTEP 10000

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0.00	-1.29	-2.17	-2.64	-2.56	-2.68	-2.84	-2.96	-2.97	-2.90	-2.81	-2.72	-2.60	-2.42	-2.18	-1.91	-1.65	-1.47	-1.43	-1.47
2	0.00	-0.48	-1.68	-2.21	-2.25	-2.37	-2.49	-2.57	-2.56	-2.50	-2.44	-2.37	-2.28	-2.13	-1.94	-1.72	-1.50	-1.36	-1.35	-1.43
3	0.00	-0.02	-0.73	-1.32	-1.52	-1.67	-1.71	-1.69	-1.63	-1.58	-1.55	-1.52	-1.47	-1.38	-1.26	-1.11	-0.98	-0.89	-0.90	-0.99
4	0.00	0.39	0.16	-0.46	-0.75	-0.92	-0.91	-0.82	-0.72	-0.67	-0.67	-0.66	-0.63	-0.59	-0.54	-0.47	-0.41	-0.37	-0.38	-0.42
5	0.00	0.00	0.24	-0.04	-0.36	-0.55	-0.52	-0.42	-0.31	-0.26	-0.26	-0.26	-0.24	-0.22	-0.19	-0.17	-0.15	-0.13	-0.12	-0.12
6	0.00	0.00	0.35	0.09	-0.26	-0.46	-0.43	-0.32	-0.20	-0.15	-0.16	-0.16	-0.14	-0.12	-0.11	-0.09	-0.08	-0.07	-0.05	-0.03
7	0.00	0.00	0.00	-0.11	-0.24	-0.44	-0.42	-0.31	-0.19	-0.14	-0.14	-0.14	-0.13	-0.11	-0.09	-0.08	-0.07	-0.06	-0.04	-0.02
8	0.00	0.00	0.00	-0.11	-0.24	-0.44	-0.42	-0.30	-0.18	-0.13	-0.14	-0.14	-0.13	-0.11	-0.09	-0.08	-0.07	-0.06	-0.04	-0.02
9	0.00	0.00	0.00	0.00	-0.33	-0.44	-0.42	-0.30	-0.18	-0.13	-0.13	-0.14	-0.12	-0.10	-0.09	-0.08	-0.07	-0.05	-0.04	-0.02
10	0.00	0.00	0.00	0.00	-0.33	-0.44	-0.42	-0.29	-0.17	-0.12	-0.13	-0.13	-0.12	-0.10	-0.09	-0.08	-0.07	-0.05	-0.04	-0.02
11	0.00	0.00	0.00	0.00	0.00	-0.42	-0.41	-0.29	-0.16	-0.11	-0.12	-0.13	-0.12	-0.10	-0.08	-0.07	-0.06	-0.05	-0.03	-0.01
12	0.00	0.00	0.00	0.00	0.00	-0.42	-0.41	-0.28	-0.15	-0.10	-0.11	-0.12	-0.11	-0.09	-0.08	-0.07	-0.06	-0.05	-0.03	-0.01
13	0.00	0.00	0.00	0.00	0.00	0.00	-0.32	-0.27	-0.13	-0.08	-0.10	-0.11	-0.10	-0.08	-0.07	-0.06	-0.05	-0.04	-0.03	-0.01
14	0.00	0.00	0.00	0.00	0.00	0.00	-0.30	-0.25	-0.11	-0.06	-0.08	-0.10	-0.09	-0.07	-0.06	-0.05	-0.05	-0.03	-0.02	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.14	-0.08	-0.04	-0.06	-0.08	-0.07	-0.06	-0.05	-0.04	-0.04	-0.03	-0.01	0.01

V VELOCITY FOR J = 14 AT TIMESTEP 10000

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0.00	-15.76	-17.33	-15.12	-12.30	-11.01	-11.15	-11.89	-12.59	-12.85	-12.78	-12.68	-12.73	-12.87	-13.04	-13.20	-13.31	-13.27	-12.98	-12.44
2	0.00	-4.26	-4.71	-2.29	0.29	1.48	1.33	0.60	-0.06	-0.29	-0.21	-0.11	-0.15	-0.26	-0.41	-0.56	-0.66	-0.62	-0.38	0.10
3	0.00	-5.37	-4.41	-1.80	0.43	1.48	1.32	0.57	-0.06	-0.23	-0.13	-0.04	-0.06	-0.14	-0.23	-0.33	-0.40	-0.39	-0.23	0.09
4	0.00	-5.99	-4.07	-1.49	0.50	1.49	1.38	0.58	-0.07	-0.20	-0.09	0.00	-0.01	-0.04	-0.08	-0.12	-0.16	-0.15	-0.09	0.05
5	0.00	0.00	-3.93	-1.52	0.42	1.49	1.47	0.64	-0.05	-0.19	-0.07	0.01	0.01	-0.01	-0.02	-0.02	-0.02	-0.02	-0.01	0.02
6	0.00	0.00	-3.93	-1.59	0.34	1.47	1.51	0.68	-0.03	-0.19	-0.07	0.02	0.02	0.00	0.00	0.01	0.01	0.02	0.01	0.00
7	0.00	0.00	0.00	-1.60	0.31	1.47	1.51	0.68	-0.03	-0.18	-0.07	0.02	0.02	0.01	0.01	0.02	0.02	0.02	0.01	0.00
8	0.00	0.00	0.00	-1.60	0.31	1.46	1.51	0.68	-0.03	-0.18	-0.06	0.02	0.02	0.01	0.01	0.02	0.02	0.02	0.01	0.00
9	0.00	0.00	0.00	0.00	0.33	1.46	1.51	0.68	-0.03	-0.18	-0.06	0.02	0.02	0.01	0.01	0.02	0.02	0.02	0.01	0.00
10	0.00	0.00	0.00	0.00	0.33	1.46	1.51	0.67	-0.03	-0.18	-0.06	0.02	0.02	0.01	0.01	0.02	0.02	0.02	0.01	0.00
11	0.00	0.00	0.00	0.00	0.00	1.48	1.51	0.67	-0.03	-0.18	-0.06	0.02	0.02	0.01	0.01	0.02	0.02	0.02	0.01	0.00
12	0.00	0.00	0.00	0.00	0.00	1.48	1.51	0.66	-0.04	-0.18	-0.06	0.02	0.02	0.01	0.01	0.02	0.02	0.02	0.02	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	1.52	0.66	-0.04	-0.17	-0.05	0.03	0.02	0.01	0.01	0.02	0.02	0.02	0.02	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	1.52	0.65	-0.04	-0.17	-0.05	0.03	0.03	0.01	0.01	0.02	0.02	0.03	0.02	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.65	-0.04	-0.16	-0.04	0.04	0.03	0.01	0.01	0.02	0.03	0.03	0.02	0.01

TS= 10000 YEAR= 2.28 DAY=102.8 ENERGY= 1.564047E+00 DTEMP= 2.615693E-09 DSALT= 2.516282E-13 SCANS= 3

WORK BY:	INTERNAL MODE	EXTERNAL MODE
TIME RATE OF CHANGE	-2.532551E-09	5.444185E-10
HORIZONTAL ADVECTION	-6.344492E-10	1.107209E-10
VERTICAL ADVECTION	1.121961E-08	-1.069588E-08
HORIZONTAL FRICTION	-2.042819E-06	-1.265078E-06
VERTICAL FRICTION	-1.240506E-07	1.911197E-24
PRESSURE FORCES	1.728648E-07	1.034483E-06
IMPLICIT EFFECTS	2.469470E-10	-1.835201E-11
WORK BY WIND	2.041012E-06	2.864382E-07
BOTTOM DRAG	-6.037195E-08	-4.469476E-08

	TEMPERATURE	SALINITY
TIME RATE OF CHANGE	2.326228E-09	2.189734E-13
HORIZONTAL ADVECTION	4.006320E-21	-1.409569E-25
VERTICAL ADVECTION	-3.293317E-23	3.015121E-28
HORIZONTAL DIFFUSION	8.159803E-23	-1.956632E-26
SURFACE FLUX	2.753010E-24	1.191033E-28
TRUNCATION ERROR	2.326228E-09	2.189734E-13
CHANGE OF VARIANCE	3.391344E-08	3.102923E-17

WORK BY BUOYANCY FORCES 1.207348E-06 ENERGY CONVERSION ERROR -7.284483E-18 NONLINEAR EXCHANGE ERROR -1.058791E-22

NORTHWARD TRANSPORT OF HEAT (X10**15 WATTS)

	X MEAN	X EDDY	Z MEAN	Z EDDY	EKMAN	TOT ADV	DIFFUS	TOTAL
18	-0.180	-0.136	0.033	-0.349	0.000	-0.316	-0.012	-0.328
17	-0.770	-0.043	0.014	-0.042	-0.785	-0.813	0.039	-0.774
16	-1.142	0.012	-0.033	0.000	-1.097	-1.130	0.055	-1.075
15	-1.298	0.019	-0.071	0.026	-1.234	-1.280	0.042	-1.237
14	-1.335	0.016	-0.102	0.036	-1.253	-1.319	0.028	-1.292
13	-1.291	0.010	-0.125	0.036	-1.192	-1.281	0.016	-1.265
12	-1.196	0.007	-0.139	0.028	-1.079	-1.190	0.007	-1.182
11	-1.069	0.004	-0.143	0.014	-0.936	-1.065	0.000	-1.065
10	-0.920	0.001	-0.139	0.000	-0.779	-0.919	-0.006	-0.925
9	-0.763	-0.001	-0.129	-0.014	-0.621	-0.764	-0.010	-0.775
8	-0.611	-0.003	-0.114	-0.028	-0.472	-0.614	-0.013	-0.628
7	-0.465	-0.005	-0.097	-0.036	-0.338	-0.471	-0.014	-0.485
6	-0.327	-0.005	-0.079	-0.030	-0.223	-0.332	-0.010	-0.341
5	-0.209	0.000	-0.059	-0.016	-0.133	-0.209	0.006	-0.202
4	-0.116	0.004	-0.045	-0.006	-0.061	-0.112	0.028	-0.084
3	-0.042	-0.014	-0.036	-0.008	-0.012	-0.056	0.009	-0.047
2	-0.008	-0.012	-0.014	-0.006	0.000	-0.019	-0.021	-0.041

NORTHWARD TRANSPORT OF SALT (X10**10 CM**3/SEC)

	X MEAN	X EDDY	Z MEAN	Z EDDY	EKMAN	TOT ADV	DIFFUS	TOTAL
18	-0.027	-0.014	0.009	-0.050	0.000	-0.041	0.000	-0.041
17	-0.170	0.001	0.005	-0.002	-0.173	-0.169	0.015	-0.155
16	-0.319	0.010	-0.009	0.005	-0.305	-0.309	0.026	-0.284
15	-0.453	0.013	-0.023	0.013	-0.429	-0.439	0.030	-0.410
14	-0.574	0.019	-0.039	0.022	-0.538	-0.555	0.032	-0.524
13	-0.679	0.026	-0.054	0.029	-0.627	-0.652	0.033	-0.620
12	-0.764	0.035	-0.068	0.031	-0.691	-0.729	0.034	-0.694
11	-0.825	0.043	-0.081	0.026	-0.728	-0.782	0.035	-0.747
10	-0.860	0.050	-0.091	0.017	-0.735	-0.810	0.036	-0.774
9	-0.866	0.055	-0.100	0.002	-0.713	-0.811	0.037	-0.774
8	-0.846	0.056	-0.106	-0.022	-0.662	-0.790	0.041	-0.749
7	-0.792	0.055	-0.109	-0.045	-0.583	-0.737	0.051	-0.686
6	-0.693	0.058	-0.108	-0.048	-0.479	-0.635	0.084	-0.551
5	-0.566	0.078	-0.097	-0.029	-0.362	-0.488	0.164	-0.324
4	-0.401	0.080	-0.100	-0.011	-0.209	-0.321	0.287	-0.033
3	-0.180	-0.038	-0.140	-0.029	-0.048	-0.218	0.270	0.052
2	-0.043	-0.061	-0.075	-0.029	0.000	-0.104	0.094	-0.010

MERIDIONAL MASS TRANSPORT

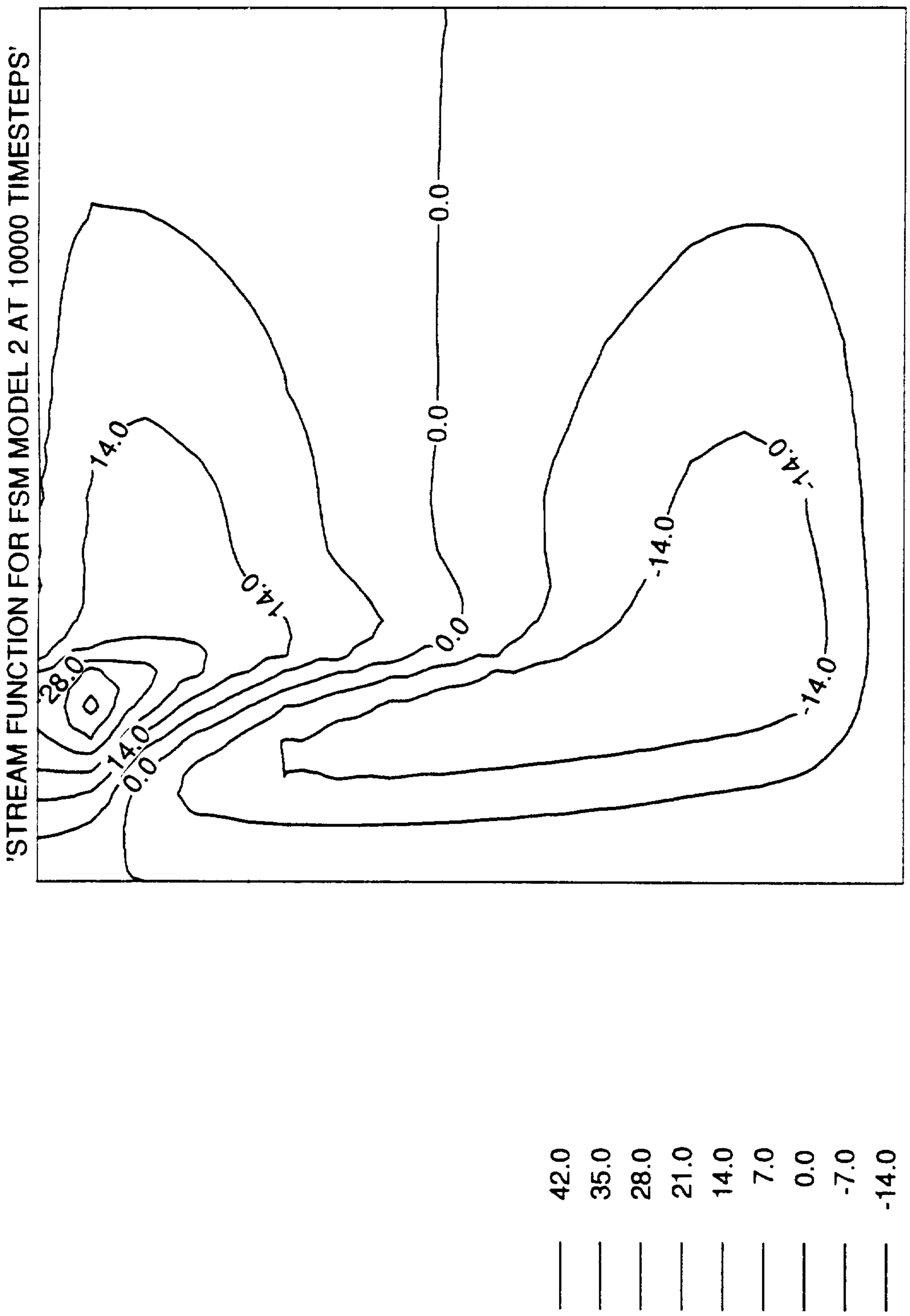
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0.00	-0.24	-1.66	-3.36	-5.33	-7.48	-9.71	-11.90	-14.05	-16.10	-17.95	-19.51	-20.68	-21.34	-21.23	-19.79	-14.43	0.14	0.00	0.00
2	0.00	-0.60	-2.08	-3.89	-6.03	-8.36	-10.69	-12.88	-14.96	-16.96	-18.73	-20.20	-21.31	-21.94	-21.83	-20.39	-15.25	-7.66	0.00	0.00
3	0.00	-1.09	-2.69	-4.69	-7.07	-9.63	-12.09	-14.27	-16.29	-18.22	-19.92	-21.31	-22.33	-22.88	-22.61	-21.00	-15.62	-7.29	0.00	0.00
4	0.00	-1.69	-3.58	-5.86	-8.52	-11.31	-13.90	-16.08	-18.06	-19.97	-21.63	-22.95	-23.89	-24.30	-23.81	-21.86	-14.94	-5.72	0.00	0.00
5	0.00	-2.30	-4.36	-6.81	-9.62	-12.51	-15.10	-17.22	-19.12	-20.95	-22.52	-23.73	-24.52	-24.72	-23.90	-21.27	-12.82	-4.34	0.00	0.00
6	0.00	-2.96	-5.38	-8.10	-11.05	-13.98	-16.58	-18.66	-20.53	-22.29	-23.78	-24.86	-25.48	-25.37	-24.10	-20.46	-9.66	-1.54	0.00	0.00
7	0.00	-3.52	-6.10	-8.91	-11.87	-14.74	-17.24	-19.21	-20.95	-22.55	-23.84	-24.66	-24.94	-24.36	-22.52	-17.75	-6.29	-1.03	0.00	0.00
8	0.00	-3.99	-6.99	-10.00	-12.94	-15.70	-18.08	-19.92	-21.49	-22.89	-23.93	-24.40	-24.26	-23.08	-20.51	-14.33	-1.70	1.19	0.00	0.00
9	0.00	-4.23	-7.25	-10.23	-13.03	-15.61	-17.76	-19.38	-20.70	-21.79	-22.45	-22.52	-21.89	-20.23	-17.13	-10.37	0.80	-1.75	0.00	0.00
10	0.00	-4.26	-7.58	-10.57	-13.18	-15.50	-17.38	-18.72	-19.72	-20.42	-20.62	-20.18	-18.96	-16.72	-12.97	-5.53	4.04	-3.50	0.00	0.00
11	0.00	-3.97	-7.00	-9.70	-12.01	-13.99	-15.53	-16.54	-17.18	-17.51	-17.34	-16.57	-15.10	-12.78	-9.18	-2.69	2.82	-10.51	0.00	0.00
12	0.00	-3.34	-6.29	-8.69	-10.62	-12.19	-13.31	-13.91	-14.13	-14.00	-13.39	-12.22	-10.46	-8.06	-4.65	0.64	1.68	-14.88	0.00	0.00
13	0.00	-2.41	-4.63	-6.41	-7.81	-8.89	-9.60	-9.89	-9.89	-9.61	-8.99	-7.98	-6.60	-4.85	-2.53	0.40	-2.42	-17.97	0.00	0.00
14	0.00	-1.26	-2.66	-3.76	-4.53	-5.04	-5.26	-5.20	-4.93	-4.49	-3.85	-3.04	-2.09	-1.08	0.03	0.56	-4.59	-13.42	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

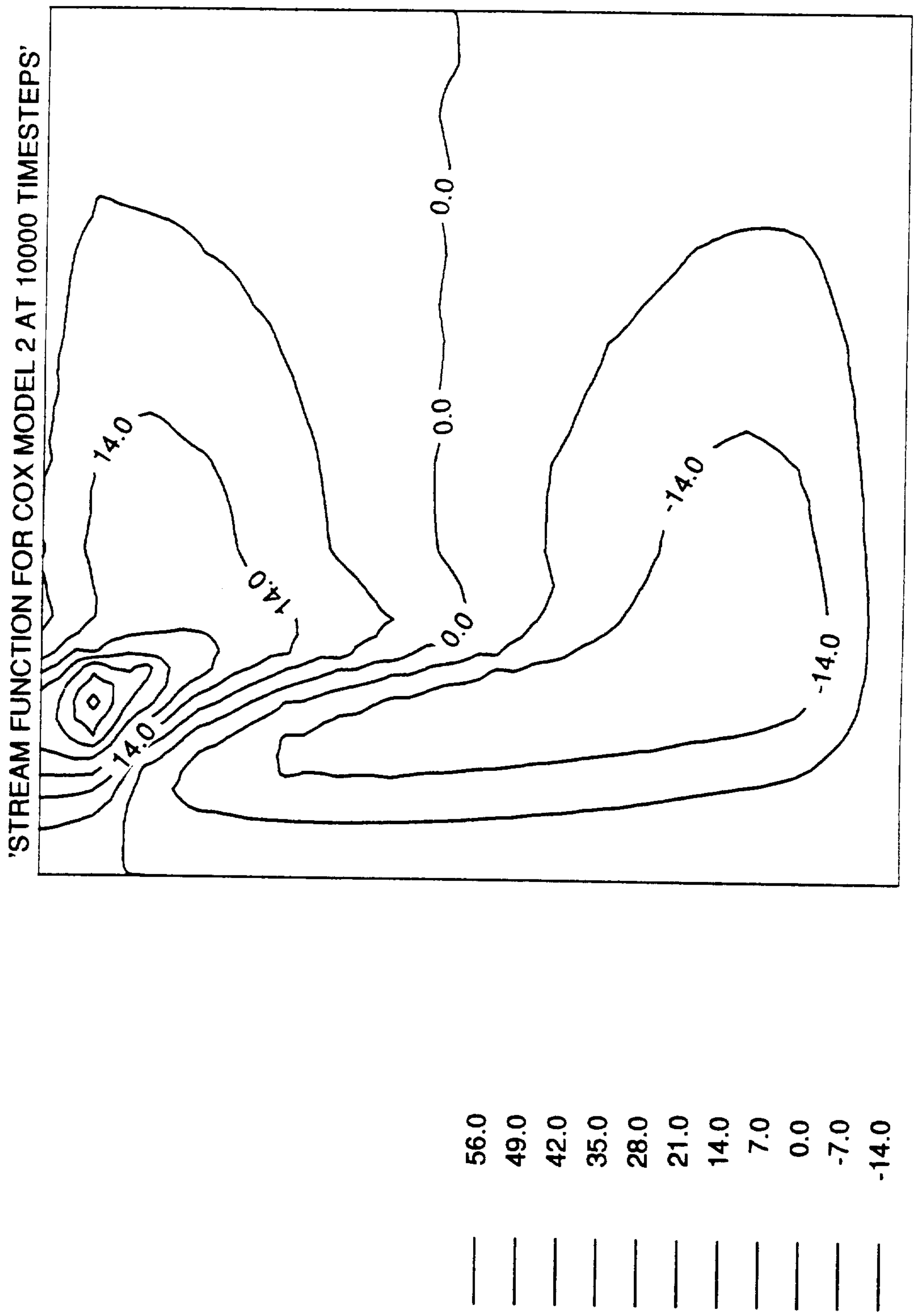
STREAM FUNCTION IN SVERDRUPS, TS= 10000

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
20	0.00	-5.24	-11.55	-22.50	-30.58	-31.27	-18.50	-8.41	-5.28	-6.68	-6.66	-6.95	-6.00	-6.19	-5.42	-5.63	-5.04	-5.36	-4.20	-3.45
19	0.00	5.24	11.55	22.50	30.58	31.27	18.50	8.41	5.28	6.68	6.66	6.95	6.00	6.19	5.42	5.63	5.04	5.36	4.20	3.45
18	0.00	2.36	9.32	22.37	41.86	52.21	40.66	19.55	13.90	14.36	15.94	14.88	14.33	12.87	12.47	11.02	10.31	8.45	7.24	5.06
17	0.00	-2.18	-6.34	-2.93	10.73	31.30	37.38	25.12	17.75	17.86	17.82	17.61	15.70	14.78	12.95	11.96	9.92	8.67	6.50	5.14
16	0.00	-2.42	-7.50	-10.80	-4.76	9.47	24.74	23.61	19.30	16.59	16.57	15.33	14.55	12.68	11.76	9.92	8.91	7.01	5.93	4.03
15	0.00	-3.35	-8.81	-12.51	-11.21	-0.62	13.39	19.98	17.27	14.69	12.83	12.66	11.36	10.70	9.12	8.42	6.87	6.10	4.53	3.70

14	0.00	-3.52	-9.20	-14.52	-14.74	-7.85	6.70	14.67	14.81	11.26	9.92	8.93	8.94	7.79	7.34	6.09	5.63	4.38	3.85	2.57
13	0.00	-3.82	-9.27	-14.39	-17.19	-12.43	-0.31	10.19	10.72	8.50	6.16	6.10	5.59	5.61	4.65	4.43	3.50	3.27	2.32	2.01
12	0.00	-3.72	-9.05	-14.50	-17.77	-16.30	-5.68	4.29	7.04	4.63	3.19	2.35	2.83	2.44	2.51	1.86	1.91	1.32	1.33	0.73
11	0.00	-3.68	-8.62	-13.70	-18.23	-18.26	-11.16	-0.95	2.01	1.28	-0.65	-0.74	-0.79	-0.20	-0.45	-0.16	-0.42	-0.09	-0.32	-0.01
10	0.00	-3.39	-8.07	-13.10	-17.56	-19.89	-14.96	-6.85	-2.58	-3.01	-3.78	-4.43	-3.75	-3.49	-2.82	-2.77	-2.18	-2.05	-1.42	-1.25
9	0.00	-3.15	-7.36	-12.03	-17.05	-20.08	-18.49	-11.66	-7.91	-6.77	-7.45	-7.31	-7.07	-6.09	-5.61	-4.75	-4.32	-3.43	-2.92	-1.99
8	0.00	-2.76	-6.63	-11.13	-15.88	-20.23	-20.38	-16.41	-12.48	-11.10	-10.49	-10.44	-9.61	-8.90	-7.74	-6.98	-5.86	-5.05	-3.88	-3.00
7	0.00	-2.43	-5.84	-10.03	-15.02	-19.39	-21.89	-19.73	-17.10	-14.85	-13.88	-12.93	-12.24	-11.02	-10.00	-8.63	-7.55	-6.15	-5.00	-3.57
6	0.00	-2.03	-5.09	-9.12	-13.81	-18.71	-21.82	-22.18	-20.33	-18.41	-16.59	-15.44	-14.17	-13.09	-11.64	-10.33	-8.75	-7.35	-5.72	-4.26
5	0.00	-1.70	-4.33	-8.11	-12.88	-17.24	-20.93	-22.23	-21.70	-19.94	-18.25	-16.62	-15.39	-14.04	-12.77	-11.22	-9.73	-8.02	-6.41	-4.62
4	0.00	-1.27	-3.46	-6.86	-11.01	-14.99	-17.44	-18.97	-18.87	-17.90	-16.44	-15.17	-13.94	-12.92	-11.76	-10.59	-9.21	-7.82	-6.24	-4.68
3	0.00	-0.55	-1.62	-3.55	-6.30	-8.63	-9.86	-10.65	-10.82	-10.37	-9.71	-8.98	-8.38	-7.80	-7.25	-6.60	-5.92	-5.11	-4.26	-3.28
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

	22	23	24
20	-2.21	-2.04	0.00
19	2.21	2.04	0.00
18	3.69	1.40	0.00
17	2.97	1.65	0.00
16	2.92	1.12	0.00
15	2.13	1.23	0.00
14	1.94	0.70	0.00
13	1.07	0.69	0.00
12	0.70	0.15	0.00
11	-0.20	0.06	0.00
10	-0.64	-0.43	0.00
9	-1.43	-0.54	0.00
8	-1.81	-0.92	0.00
7	-2.37	-1.00	0.00
6	-2.63	-1.22	0.00
5	-2.96	-1.27	0.00
4	-2.96	-1.34	0.00
3	-2.24	-1.07	0.00
2	0.00	0.00	0.00
1	0.00	0.00	0.00


'STREAM FUNCTION FOR FSM MODEL 2 AT 10000 TIMESTEPS'
42.0
35.0
28.0
21.0
14.0
7.0
0.0
-7.0
-14.0



'STREAM FUNCTION FOR COX MODEL 2 AT 10000 TIMESTEPS'
14.0
14.0
14.0
0.0
0.0
0.0
-14.0
-14.0
56.0
49.0
42.0
35.0
28.0
21.0
14.0
7.0
0.0
-7.0
-14.0

FIGURE 3

M. Restarting Procedure

This is exactly as in Cox, save that the restart data are dumped onto unit <u>23</u>, not unit 21, at the end of the run. They continue to be read in on unit 21 as before at the beginning of a restart. The CONTRL variable NA determines whether restart data are saved at the end of a run (see Section L).

Appendix. Barotropic grid-splitting.

This appensix discusses grid-splitting, which can occur in the tidal option of the code.

Solutions for this problem are standard in the atmospheric literature; cf. Mesinger, 1973, Jancic, 1974, for example. The methods all use the idea of evaluating the $\nabla.\mathbf{U}$ term with components from both the checkerboard grids, which re-links the two grids and acts to smooth out the +/- behaviour. The actual formula involves adding a term of form

$$\alpha g H \Delta t(\nabla_+^2 - \nabla_x^2)\eta \qquad \text{(A1)}$$

to the r.h.s. of (21), where $0 < \alpha < \alpha_{max}$, for some α_{max} of order unity (dependent on numerical scheme). Note that α is referred to as "WGHT" in the program listing. Here ∇^2 represents the traditional 5-point formulation of ∇^2, i.e. involving the arrangement

```
     o
  o  x  o
     o
```

while ∇_x^2 uses the wider grid

```
  o     o
     x
  o     o
```

and both formulae are implicitly on an equispaced Cartesian grid. For disturbances of large horizontal scale, the effect of (A1) is minimal: the two versions of the Laplacian are almost equal, so that their difference is tiny. The smoothing is only efficient for large wavenumber disturbances (i.e. waves near the +/- limit).

The atmospheric literature is concerned, naturally, with the *stability* and accurate *phase velocities* of the schemes considered. Conservation properties, etc., are hardly mentioned; we shall return to this point below.

Applying the atmospheric solution was nontrivial, for two reasons. First, the grid-linkage scheme implicitly relies on a uniform metric (i.e. $\Delta x = \Delta y$, no spherical 'coordinates). Indeed, even permitting the small modification that Δx and Δy take different values, the formula (A1) ceases to be a smoothing operator for all wavenumbers, and causes growth for some disturbances. (Seeking a Fourier component in λ and ϕ for the disturbance shows that the natural definitions for the Laplacian operators permits growth for waves varying only in ϕ if the local Δx is larger than the local Δy, and similarly growth for waves varying only in λ if the reverse is true.) Second, atmospheric models lack boundaries. The schemes all involve various estimates of $\nabla^2\eta$, so that values for η are required outside oceanic boundaries.

It proved impossible to modify the uniform metric scheme for a spherical grid, since the arguments used depend implicitly on 45^o rotations of grids. It is also necessary to maintain mass conservation, so that the integrated effect of the smoother must be zero apart from boundary effects, dealt with later. After experimentation, a term of the form

$$(\alpha g H_{max} \Delta t / a^2 \cos\phi \ \Delta\lambda_T \Delta\phi_T)[A(\eta) - B(\eta)] \quad \text{(A2)}$$

is found to have the right properties. Here $\Delta\lambda_T$, etc. refer, as in the Cox code, to the values of the grid spacing centred on tracer and η points. This conserves mass, because mass conservation involves a local metric of precisely the denominator of (A2). The functions $A(\eta)$, $B(\eta)$ are given by

$$A(\eta) = [\eta_{i-1,j} - \eta_{i,j}] + [\eta_{i+1,j} - \eta_{1,j}]$$

$$+ [\eta_{i,j-1} - \eta_{i,j}] + [\eta_{i,j+1} - \eta_{i,j}] \quad \text{(A3)}$$

$$B(\eta) = (1/2)\{[\eta_{i-1,j-1} - \eta_{i,j}] + [\eta_{i-1,j+1} - \eta_{i,j}]$$

$$+ [\eta_{i+j,j-1} - \eta_{i,j}] + [\eta_{i+1,j+1} - \eta_{i,j}]\} \quad \text{(A4)}$$

which are 'local' estimates of ∇^2 on an equispaced grid. It is straightforward to show that disturbances of any Fourier component in λ and ϕ are damped by this term.

The second problem, to define values of η outside ocean boundaries, needed much numerical investigation. Were any normal Laplacian diffusion operator being used, the natural definition of η values outside a boundary would be such that η_n vanished, where n is again a normal coordinate to the boundary. However, η values outside a boundary can be involved in several evaluations of (A3, A4). Consider the situation in Fig. A. The point 'Y' will be referred to by evaluations of (A3) or (A4) from four different internal points - exactly what should its value be? Recall that we require simultaneously that mass be conserved and that the scheme should continue to smooth even at boundaries. More formally, the sum of all the external point contributions to (A3, A4) must vanish; and all Fourier components of disturbances must continue to be damped even near boundaries.

The obvious definition for an external η value was such that its integrated contribution to the mass budget was zero. Unfortunately, this is impossible. Consider the case of a straight boundary, as in Fig. B. The external value A is used in damping internal points a, b, and c. To ensure zero contribution to the mass budget, we need

$$(A - b) - 1/2(A - a) - 1/2(A - c) = 0 \quad \text{(A5)}$$

assuming for the moment an equispaced grid. (The first term comes from the ∇_+^2 term at b, the other two from the ∇_x^2 terms at a and c.) But the value A cancels out, leaving an impossibility.

A second option is to ignore any pairwise contributions in (A3, A4) which involve a boundary, which automatically conserves mass. This indeed acts a a smoother along a straight boundary, but fails for other boundary conditions, e.g. a northeast coast as in Fig. C. We assume for simplicity that a potential instability is concentrated near the coast, and takes a +/- form as shown, with zero values elsewhere. The effect of (A3, A4) on the +1 value is

[(ignored) + (ignored) + (0-1) + (0-1)] -

1/2[(-1-1) + (-1-1) + (0-1) + (ignored)]

$$= 1/2 > 0$$

so that the +1 grows, and the disturbance is amplified.

The solution implemented in the code is as follows:

The value of an external point is defined as the average of the internal values which access that point when ∇_+^2 is evaluated; whenever ∇_x^2 needs to use an external point, the contribution of that pair is neglected

As an example, Fig. D shows a coastline indentation, with capital letters referring to internal values, and lower case to external values. The values of the external points are:

$$a = 1/2(A + D);$$

f, h are irrelevant (only involved in ∇_x^2);

$$g = D;$$

$$e = 1/2(C + D).$$

It is simple to show that these definitions conserve mass and provide a stable scheme for all coastline configurations. There are some coastline shapes for which this formula does not yield a smoothing, but any small horizontal viscosity handles this. The scheme is vectorisable because land masks can be set up at the beginning of the calculation, and used both for computing external values and as multipliers in ∇_x^2.

Jancic, Z. J., 1974: A stable centred difference scheme free of two-grid-interval noise. Monthly Weather Review,102, 319-323.

Mesinger, F., 1973: A method for construction of second-order accuracy difference schemes permitting no false two-grid-interval waves in the height field.Tellus, 25, 444-457.

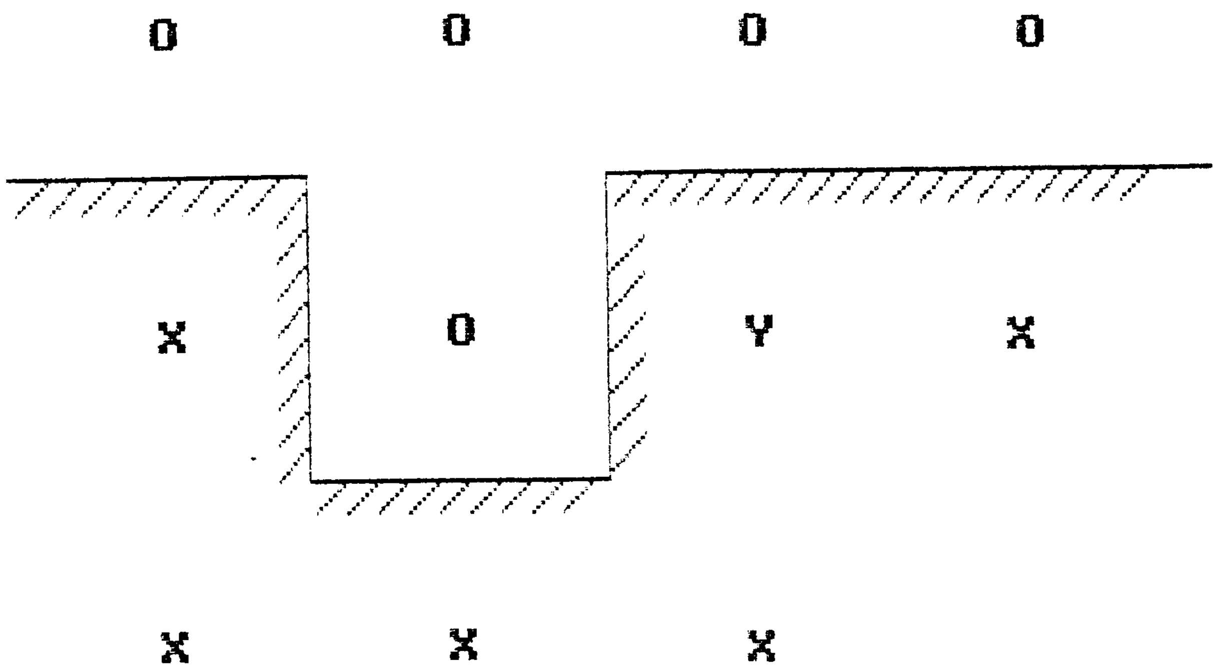


Fig. A

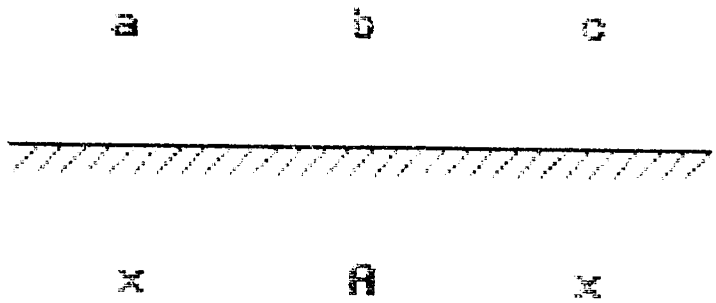


Fig. B

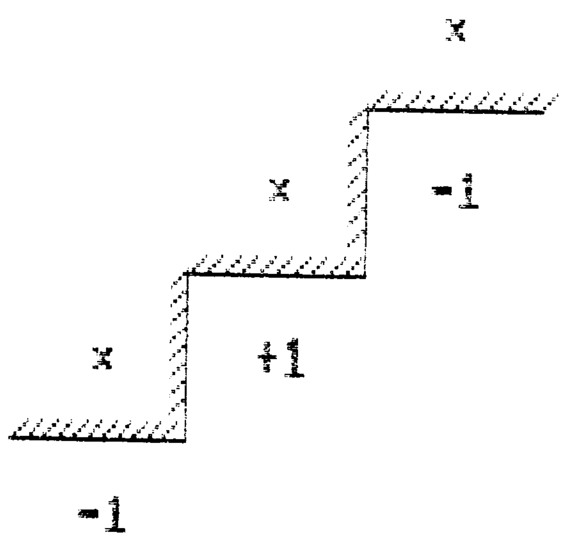


Fig. C

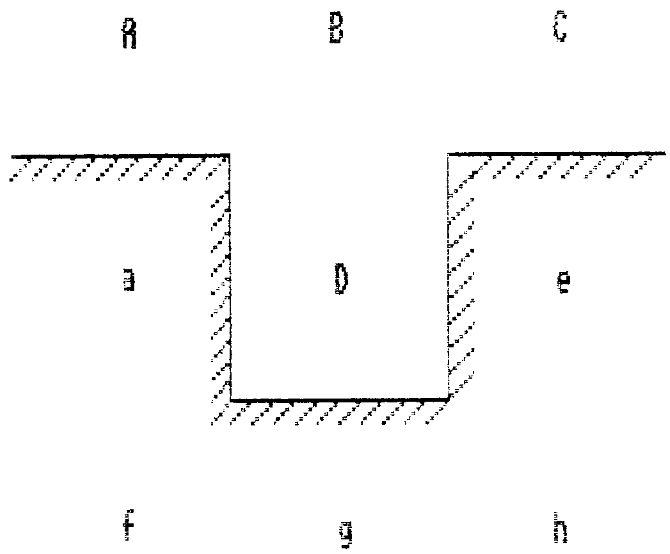


Fig. D