AD-A058 244 NAVAL RESEARCH LAB WASHINGTON D C F/G 4/2

THE MODIFIED CANADIAN SPECTRAL FORECAST MODEL: DISCUSSION AND D--ETC(U)

MAR 78 W W JONES, R V MADALA, M R SCHOEBERL

UNCLASSIFIED NRL-MR-3698 SBIE-AD-E000 193 NL

1 OF 3

AD
A058244

NRL Memorandum Report 3698

The Modified Canadian Spectral Forecast Model: Discussion and Documentation

Walter W. Jones, Rangarao V. Madala,
and Mark R. Schoeberl

Plasma Physics Division

March 1978

NAVAL RESEARCH LABORATORY
Washington, D. C.

(18) SBIE (19) AD-E~~000~~ 193

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NRL Memorandum Report 3698
2. GOVT ACCESSION NO.:
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): (6) THE MODIFIED CANADIAN SPECTRAL FORECAST MODEL: DISCUSSION AND DOCUMENTATION.
5. TYPE OF REPORT & PERIOD COVERED: Interim report: July, 1976 to December, 1977.
6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): (10) Walter W. Jones, Rangarao V. Madala [redacted] Mark R. Schoeberl
8. CONTRACT OR GRANT NUMBER(s):
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Research Laboratory, Washington, D.C. 20375
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: NRL Problem A03-23, Project WF52-551-713
11. CONTROLLING OFFICE NAME AND ADDRESS: Director, Naval Environmental Prediction Research Facility, Monterey, Calif.
12. REPORT DATE: (11) Mar[redacted]78
13. NUMBER OF PAGES: 193
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): (9) Interim rept. Jul 76-Dec 77,
15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:
16. DISTRIBUTION STATEMENT (of this Report): Approved for public release; distribution unlimited.

(14) NRL-MR-3698

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report): (16) F52551 (17) WF52551713
18. SUPPLEMENTARY NOTES:

This research was sponsored by Naval Environmental Prediction Research Facility Block Fund under Project WF52-551-713

19. KEY WORDS (Continue on reverse side if necessary and identify by block number):
General Circulation Model
Spectral Forecast Model
Atmospheric Modeling

(12) 194 p.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Canadian spectral forecast model developed by Daley, et al., (1976) has been modified for use in medium range forecasts. These modifications involve primarily the vertical differencing scheme (which is changed to conserve energy), the dry convective adjustment, and some of the program structure which allows the model to run efficiently, ~~on the Texas Instrument Computer (ASC)~~. Results of a medium range integration are compared using both the modified and original model and actual weather observation.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

78 07 27 001

251 950

CONTENTS

THE MODIFIED CANADIAN SPECTRAL FORECAST MODEL: DISCUSSION AND DOCUMENTATION

I. INTRODUCTION

The Canadian spectral forecast model (CSFM) is a multi-dimensional global model of the atmosphere in sigma coordinates using spectral decomposition of dependent variables in the horizontal (Daley, et al., 1976). This model is currently being used at NRL for various atmospheric research programs. It was chosen because of its computational efficiency and extensive documentation.

However, one of the disadvantages of the CSFM is that it does not conserve energy with its vertical differencing scheme. We have, therefore, replaced the vertical differencing scheme with one which conserves energy. In connection with this modification, we have also altered the boundary layer approximations and the dry convective adjustment procedure. Other minor changes have been made to the original CSFM to allow the model to run on the NRL's Texas Instrument computer (ASC). The entire modified code is documented in Appendix I.

In the section which follows, we describe briefly the procedure used by the CSFM to generate forecasts and introduce the hydrodynamic equations solved by the model. Section III gives a derivation of the modified equations used for the energy conserving grid, and the final section compares the results obtained from these two models.

*Manuscript submitted January 16, 1978.

II. MODEL DESCRIPTION AND EQUATIONS

The CSFM is primarily a spectral model which implies that the dependent variables have been expanded (in the horizontal) in terms of spherical harmonics. The use of fast numerical transforms between the grid network and the spectral system gives this method computational efficiency for relatively small grids. Further, since the horizontal differentiation is done "exactly", a spectral model conserves energy for the horizontal grid up to the truncation of the spectral series and nonlinear aliasing generated on the Gaussian grid.

For the vertical grid, the CSFM divides the atmosphere into a number of layers. In the CSFM, the variables ϕ, ξ and D are defined at the levels while T, γ and W are defined between the levels, Fig. (1a). The finite difference form of the equations in the vertical direction using this scheme does not conserve total energy. For short range forecasts (1-2 days) energy conservation is not critical but for medium and long range forecasts (3-10 days) the forecast model must conserve energy in order to retain any skill. In order to conserve the total energy, we define the variables ϕ, ξ, D and T at the levels and γ and W between the levels Fig. (1b).

Figure 1a shows the vertical differencing scheme used in the original CSFM. The modified scheme is shown in Figure 1b. Although the differences in the geometry of the two schemes appear slight, the modifications to the model are extensive since the vertical differencing scheme plays an important role in the semi-implicit algorithm used by the CSFM, as well as in calculation of the non-linear terms.

The equations solved by the CSFM are Equations (37) - (42) in Daley, et al., (1976). These equations are

$$\frac{\partial \zeta_l^m}{\partial t} = -\{\alpha\ (A, B)\}_l^m, \tag{1}$$

$$\frac{\partial D_l^m}{\partial t} - a^{-2} l(l+1) P_l^m = \{\alpha(B, -A) - a^2 \nabla^2 E\}_l^m, \tag{2}$$

$$\sigma \frac{\partial^2 P_l^m}{\partial \sigma \partial t} + R\gamma^* W_l^m = R\{\alpha(UT', VT') - B_T\}_l^m, \tag{3}$$

$$\frac{\partial W_l^m}{\partial \sigma} + D_l^m = \{-G\}_l^m, \tag{4}$$

$$\frac{\partial q_l^m}{\partial t} - W_{sl}^m = 0, \tag{5}$$

$$T_l^m = -\frac{\sigma}{R} \frac{\partial \Phi_l^m}{\partial \sigma}, \tag{6}$$

where

$$\{F\}_l^m = \frac{1}{2\pi} \int_{-\pi/2}^{\pi/2} \int_0^{2\pi} F(\lambda, \theta, \sigma, t)\, Y_l^m(\lambda, \theta) \cos\theta\, d\lambda\, d\theta. \tag{7}$$

and

$$\hat{F}^\sigma = \int_\sigma^1 F d\sigma,\ \hat{F} = \int_0^1 F d\sigma.$$

In the equations above we use the following definitions for the non-linear terms which appear on the right hand sides of Eq. (1 - 6)

$$\alpha(X, Y) = \frac{1}{\cos^2\theta}\left[\frac{\partial X}{\partial \phi} + \cos\theta \frac{\partial Y}{\partial \theta}\right], \text{ where } \theta = \text{latitude}, \phi = \text{longitude}$$

$$A = (\zeta + f)\, U + \dot{\sigma} \frac{\partial V}{\partial \sigma} + \frac{RT'}{a^2} \cos\theta \frac{\partial q}{\partial \theta} - \cos\theta \frac{F_\theta}{a},$$

$$B = (\zeta + f)\, V - \dot{\sigma} \frac{\partial U}{\partial \sigma} - \frac{RT'}{a^2} \frac{\partial q}{\partial \lambda} + \cos\theta \left[\frac{F_\theta}{a}\right],$$

$$G = \frac{1}{\cos^2\theta}\left[U \frac{\partial q}{\partial \lambda} + V \cos\theta \frac{\partial q}{\partial \theta}\right],$$

$$E = \frac{(U^2 + V^2)}{2 \cos^2 \theta},$$

$$\dot{\sigma} = (\sigma - 1)(\hat{G} + \hat{D}) + \hat{G}^{\sigma} + \hat{D}^{\sigma},$$

$$B_T = T'D + \gamma'\dot{\sigma} - \frac{RT'}{C_p}(\hat{G} + \hat{D}) + \frac{RT}{C_p} G + H_T.$$

and

$$B_s = SD - \dot{\sigma}\frac{\partial S}{\partial \sigma} + \left[\frac{RT}{C_p} - \frac{RT_d^2}{GL(T_d)}\right] \times \left[\frac{\dot{\sigma}}{\sigma} + G - \hat{G} - \hat{D}\right] + H_T - H_m$$

Equations 1 and 2 are the vorticity and divergence equations respectively. Equation (3) is a modified form of the thermodynamic equation, using the variable $P = \phi + RT^*q$. Equations (4,5) are forms of the continuity and surface pressure tendency equations respectively. The hydrostatic equation is given by equation (6). The definitions of the symbols used are

f = Coriolis parameter,

V = horizontal vector wind [with components (U, V)]

ζ = vertical component of vorticity $= \hat{k} \cdot \nabla \times \mathbf{V}$,

D = horizontal divergence $= \nabla \cdot V$,

T = the absolute temperature (°K)

q = $ln(p_s)$,

γ = static stability $= \left(\frac{RT}{C_p\sigma} - \frac{\partial T}{\partial \sigma}\right)$,

$\dot{\sigma}$ = vertical motion in sigma coordinates

$= (\sigma - 1)(\hat{D} + \hat{V} \cdot \nabla q) + \hat{D}^{\sigma} + \hat{V}^{\sigma} \cdot \nabla q$

Φ = geopotential height,

F = the horizontal frictional force per unit mass,

H_T = the diabatic heating,

R is the gas constant for dry air,

C_p is the specific heat of dry air at constant pressure.

Special variables are defined by

$$P = \Phi + RT^*q$$

$$W = \dot{\sigma} - \sigma\ (\hat{G} + \hat{D})$$

and

$$W_s \equiv W\ (\sigma = 1) = -\ (\hat{G} + \hat{D}).$$

The $(\ )^*$ notation indicates a horizontal mean and $(\ )'$ is the deviation from that mean. Thus $T = T^* + T'$. The $(\ )_l^m$ indicates the coefficient of a spherical harmonic Y_l^m defined

$$Y_l^m\ (\lambda,\theta) = x_l^m\ (\sin\ \theta) e^{im\lambda}$$

where x_l^m are the associated Legendre functions of the first kind of order (l,m). m is the east-west (zonal) wave number, l is the degree of the Legendre function, λ is longitude and θ is latitude. The expansion is defined by

$$(F) = \sum_{m=-J}^{J} \sum_{l=|m|}^{|m|+J} (F)_l^m\ Y_l^m.$$

This assumes rhomboidal truncation. As long as a "sufficient" number of harmonics are retained, there does not appear to be any advantage of using the triangular truncation scheme $\left[\sum_{l=0}^{J} \sum_{M=-l}^{+l}\right]$ over the computationally more efficient rhomboidal scheme. The vertical coordinate is $\sigma = p/p_s$ where p_s is surface pressure and p is pressure.

III. DIFFERENCING FOR ENERGY CONSERVING VERTICAL GRID

We will use the following definitions for weighted variables and difference operators

$$[A_n] = (A_{n+1} + A_n)/2$$

$$\delta\,(A_n) = (A_{n+1} - A_n)$$

$$\Delta_n = ln\,(\tilde{\sigma}_{n+1}/\tilde{\sigma}_n)$$

$$d_n = ln\,(\sigma_{n+1}/\sigma_n)$$

and N is the total number of *levels* (see Figure 1) in the model. The $(\sim)$ denotes the dashed lines in Fig. (1), which represent layers. The spacing between the levels is on a logorithmic scale so that $\delta(\sigma)$ need not be a constant.

The hydrostatic Equation (6) becomes

$$\delta\,(\Phi_n) = -Rd_n\,[T_n] \tag{8}$$

where we have dropped $(\ )_l^m$ notation in this and all of what follows. Thus these equations apply to each spectral coefficient. When the non-linear terms are to be formulated, an inverse transform to a real Gaussian grid must be performed. At the surface we impose an artificial boundary layer defining a lapse rate at the lowest level:

$$\frac{\partial T}{\partial h} = -\frac{\Gamma}{g} \tag{9}$$

where Γ is the lapse rate. In the version of the model discussed here, Γ is held at 6°/km; later versions of the model will contain explicit boundary layer formulations which will generate Γ. Using Eq. (9) we define a surface temperature T_s

$$T_s = T_N\left(\frac{1}{\sigma_N}\right)^{R\Gamma/g} = C_T T_N.$$

Equations (8) and (9) then define a matrix equation of the form

$$\bar{\bar{M}}_I\,\Phi + \Phi_s/d_N\,\delta^N = -\frac{R}{2}\,\bar{\bar{M}}_{II}\,T \tag{10}$$

where

$$\left(\bar{\bar{M}}_I\right)_{i,j} = \begin{Bmatrix} -1/d_i, j = i \\ +1/d_i, j = i+1 \\ 0, \text{otherwise} \end{Bmatrix},$$

$$\left(\bar{\bar{M}}_{II}\right)_{i,j} = \begin{Bmatrix} 1, j = i \\ 1, j = i+1 \\ 0, \text{otherwise} \end{Bmatrix}$$

with

$$\left(\bar{\bar{M}}_{II}\right)_{N,N} = 1 + C_T$$

In the above equation both T and ϕ are vectors of length N, and T_s and Φ_s, the surface values, are carried separately. The delta notation, δ^N, is used to indicate that the term Φ_s/d_N will be included only with the equation for T_N. In order to compute T from Φ we use Eq. (10) to obtain

$$T = -\frac{2}{R}\,[\bar{\bar{M}}_{II}^{-1}\,\bar{\bar{M}}_I\,\Phi + M_{II}^{-1}\,\delta^N\Phi_s/d_N]$$
$$= -\,[M_p\,\Phi + \frac{2}{R}\,M_{II}^{-1}\,\delta^N\Phi_s/d_N] \qquad (11)$$

The finite difference form for γ, the static stability, becomes

$$\gamma_n = \frac{1}{\bar{\sigma}_n}\left(\frac{R}{C_p}\,[T_n] - \frac{\delta(T_n)}{d_n}\right). \qquad (12)$$

At the surface we define γ_s by using Eq. (9)

$$\gamma_s = C_T\,T_N\left(\frac{R}{C_p} - \frac{R\Gamma}{g}\right) \qquad (13)$$

We can rewrite Eqs. (12, 13) in the following form

$$\gamma = \bar{\bar{M}}_\gamma\,T$$

where the vectors are

$$\gamma_n^T = (\gamma_1, \gamma_2 \ldots \gamma_{N-1}, \gamma_N, 0)$$

$$T_n^T = (T_1, T_2, \ldots T_{N-1}, T_N, T_N)$$

and the matrix $\bar{\bar{M}}_\gamma$ is

$$\left(\bar{\bar{M}}_\gamma\right)_{i,j} = \begin{cases} \dfrac{1}{\bar{\sigma}_i}\left(\dfrac{R}{2c_p} + \dfrac{1}{d_i}\right) & \text{for } j = i \\ \dfrac{1}{\bar{\sigma}_i}\left(\dfrac{R}{2C_p} - \dfrac{1}{d_i}\right) & \text{for } j = i + 1 \\ C_T\left(\dfrac{R}{C_p} - \dfrac{R\Gamma}{g}\right) & \text{for } i = N \text{ and } j = N + 1 \\ 0, & \text{otherwise} \end{cases}$$

There is no relation between T_0 and T_1 analogous to Eq. (9). In addition, since $\sigma_0 = 0$, we can not differentiate with respect to σ at the top boundary in our log (σ) coordinate system. Thus, for the top boundary, we will use a simple deviative with respect to σ (see below), rather than $\dfrac{1}{\sigma}\dfrac{\partial}{\partial\sigma} \rightarrow 1/\ln(\sigma_{n+1}/\sigma_n)$.

At this point we describe the procedure used for dry convective adjustment. If the lapse rate exceeds the dry adiabatic lapse rate ($\gamma < 0$), the lapse rate is adjusted to be adiabatic in such a way that the total potential energy of the column is conserved. We define the quantity (Manabe et al.)

$$PE = \int_1^0 T\, C_p\, d\sigma \qquad (14)$$

which is proportional to the potential energy of a column, and let the last element of the γ vector contain PE. We then modify $\bar{\bar{M}}_\gamma$ by adding elements

$$\left(\bar{\bar{M}}_\gamma\right)_{i,N+1} = \Delta\sigma_i.$$

to the last column.

In the case of convective adjustment we can then determine a new temperature profile consistant with Equation (14) by setting γ_j elements which are less than zero to zero and multiplying by $\bar{M}_\gamma^{-1}$ to obtain T. This procedure has the affect of adjusting the temperature profile to obtain stability while conserving potential energy.

We now derive an equation for the vertical velocity. Starting with Eq. (4) which is given in finite difference as

$$\frac{\delta(W_n)}{\Delta_n} = -\sigma_n (D_n + G_n).$$

Using the boundary conditions on W:

$$W_0 = 0$$

$$W_s = -(\hat{G} + \hat{D})$$

we can write the matrix equation

$$W = \bar{M}_w^{-1} (D + G)\bar{\sigma} \tag{15}$$

where

$$\left(\bar{M}_w\right)_{i,j} = \begin{cases} -1/\Delta_{i-1}, & i = j \text{ and } i > 1 \\ +1/\Delta_{i-1}, & i = j+1 \text{ and } i > 1 \\ -1/\bar{\sigma}_i, & i = 1, j = 1 \\ 0, & \text{otherwise} \end{cases}$$

$$W^T = (W_1, W_2, \ldots W_{N-1}, W_s)$$

and

$$(D + G)^T = (D_1 + G_1, D_2 + G_2, \cdots, D_N + G_N).$$

To develop the correct form for the geopotential we begin with the thermodynamic equation

$$\frac{\partial T}{\partial t} - \gamma^* \dot{\sigma} - \frac{RT^*}{C_p}\frac{\partial q}{\partial t} = -\alpha (UT', VT') + B_T. \tag{16}$$

Since $\dot{\sigma} = W + \bar{\sigma}(\hat{G} + \hat{D})$ we define the quantities $\gamma^* \dot{\sigma}$ at the intermediate levels. Thus the finite difference form for Eq. (16) can be written as

$$\frac{\partial T}{\partial t} - \bar{M}_2 W = -\alpha (UT', VT') + B_T \tag{17}$$

where we define the matrix operator $\bar{M}_2$ as

$$\bar{M}_2 = \bar{M}_a + \bar{M}_b$$

$$\left(\bar{M}_a\right)_{i,j} = \begin{cases} \gamma_i^*/2 \text{ for } j = i,\ i \leqslant N,\ (\gamma_N \equiv \gamma_s) \\ \gamma_i^*/2 \text{ for } j = i-1,\ i < N \\ 0 \text{ otherwise} \end{cases}$$

$$(\bar{M}_b)_{i,j} = \delta_{j,N}\left[\frac{RT_i^*}{C_p} - \frac{1}{2}\left(\gamma_{i-1}\,\bar{\sigma}_{i-1} + \gamma_i\,\bar{\sigma}_i\right)\right]$$

Now we use the definition of $P = \Phi + RT^*q$, together with Eq. (10)

to yield

$$T = -\left[\bar{M}_p \Phi + \frac{2}{R}\bar{M}_{II}^{-1}\frac{\Phi_s \delta^N}{d_N}\right]$$

or

$$T = -\bar{M}_p\left[P - RqT^*\right] - \frac{2}{R}\bar{M}_{II}^{-1}\frac{\Phi_s \delta^N}{d_N} \tag{18}$$

where

$$\bar{M}_p = \frac{2}{R}\bar{M}_{II}^{-1}\bar{M}_I.$$

Substituting this equation into Equation (17) and noting that $\frac{\partial \Phi_s}{\partial t} = 0$ and $\frac{\partial q}{\partial t} = W_s$ we obtain

$$\bar{\bar{M}}_p \frac{\partial P}{\partial t} + \bar{\bar{M}}_2 W - R \bar{\bar{M}}_p T^* W_s = \alpha (UT', VT') - B_T$$

Collecting terms by noting that

$$W = \bar{\bar{M}}_w (D + G)$$

and defining

$$\bar{\bar{M}}_1 = (\bar{\bar{M}}_2 - R \bar{\bar{M}}_p T^*) \bar{\bar{M}}_W$$

we obtain

$$\bar{\bar{M}}_p \frac{\partial P}{\partial t} + \bar{\bar{M}}_1 D = \alpha (UT', VT') - B_T - \bar{\bar{M}}_1 G \tag{19}$$

where

$$P^T = (P_1, P_2, \cdots P_N).$$

The equations appear the same in spectral form except that the non-linear terms on the right hand side will appear as integrals and be written in the bracket notation { } as per Eq. (7). Thus in spectral form we have

$$\bar{\bar{M}}_p \frac{\partial P_m^l}{\partial t} + \bar{\bar{M}}_1 D_M^l = \{\alpha (UT', VT') - B_T - \bar{\bar{M}}_1 G\}_M^l \tag{20}$$

for the thermodynamic equation.

Within the numerical model, time histories of ζ, D, P, and q are carried. The other variables, T, U, and V are then calculated for diagnostic purposes.

The use of an semi-implicit time integration scheme allows us to use time steps larger than that given by the CFL condition for the Lamb wave. We have also used a frequency filter similar to the one described by Asselin (1972), to control the time splitting. The finite difference form of Eq. (2) for a semi-implicit integration becomes

$$\frac{D^{t+\Delta t} + D^{t-\Delta t}}{2\Delta t} + a^{-2}(l)(l+1)\frac{P^{t+\Delta t} + P^{t-\Delta t}}{2} = \frac{D^{t-\Delta t}}{\Delta t} +$$

$$\{\alpha(B, -A) - a^2\nabla^2 E\},$$

and Eq. (20) for the temperature becomes

$$\bar{M}_p\left[\frac{P^{t+\Delta t} + P^{t-\Delta t}}{2\Delta t}\right] + \bar{M}_1\left[\frac{D^{t+\Delta t} + D^{t-\Delta t}}{2}\right]$$

$$= \{\alpha(nT', VT') - B_T - \bar{M}_1 G\} + \bar{M}_P\frac{P^{t-\Delta t}}{\Delta t}.$$

The use of the notation $Q^t = \frac{Q^{t+\Delta t} + Q^{t-\Delta t}}{2}$ simplifies the notation and allows us to rewrite these two equations as

$$D^t + \Delta t\, a^{-2}\, l(l+1)\, P^t = D^{t-\Delta t} + \Delta t\{\alpha(B, -A) - a^2\nabla^2 E\} \tag{21}$$

and

$$P^t + \Delta t\bar{M}_p^{-1}\bar{M}_1 D^t = P^{t-\Delta t} + \Delta t\,\bar{M}_p^{-1}\{\alpha(UT', VT') - B_T\} - \Delta t\,\bar{M}_p^{-1}\bar{M}_1 G. \tag{22}$$

By substituting (21) into (22) and defining $\bar{M} = \bar{M}_P^{-1}\bar{M}_1$ we obtain

$$[1 + (\Delta t)^2 a^{-2}\, l\,(l+1)\bar{M}]\, P^t = P^{t-\Delta t}$$

$$+ \Delta t\,\bar{M}_p^{-1}\{\alpha(UT', VT') - B_T\}$$

$$+ \Delta t\,\bar{M}\{-G - D^{t-\Delta t} - \Delta t\{\alpha(B, -A) - a^2\nabla^2 E\}\} \tag{23}$$

Equations (1, 4, 5, 21 and 23) form a closed set.

IV. COMPARISON OF MODELS

A five level version of the model with two horizontal resolutions, namely 10 and 20 waves, has been integrated for three days. Figure (2) shows the 500 mb temperature forecast for the Northeast United States using both models as well as observed data for verification. This location was picked at random; other locations yield similar results. The new model clearly shows an increase in skill for one and two day temperature forcasts over the previous model. Beyond two days both models deviate considerably from observations, the primary problem being lack of moisture and treatment of the upper boundary condition.

In order to illustrate further some of the dynamics of the model, we show the zonal harmonic structure at 45°N lattitude for each level in Fig. (3). Examination of the figures indicates that the forecast of the planetary scale waves ($M = 1 - 3$) is improved as are the synoptic disturbances ($M = 5 - 8$) near the surface. The improvement in the surface forecast is probably a direct result of adjustments in the lower boundary made in the modified model. We expect that even better results could be obtained with the incorporation of a complete boundary layer model.

For tests on extended range forecasts, we find that energy tends to accumulate in the upper levels of the model for large zonal wavenumber for forcasts longer than three days. This severly degrades medium range forecasts. We attribute this effect to the use of a simple pressure difference at the upper boundary region rather than the logrithmic derivatives used elsewhere.

ACKNOWLEDGEMENTS

This work is supported by Naval Environmental Prediction Research Facility Block Fund task number WF52-551-713.

REFERENCES

Asselin, R. (1972), Monthly Weather Review **100,** 487.

Daley, R., C. Girard, J. Henderson, and I. Simmonds (1976), Short-term forecasting with a multi-level special primitive equation model part I - model formulation. Atmosphere, **14**, 98.

Haltiner, G. J. (1971) *Numerical Weather Prediction*, John Wiley and Sons, New York, 317pp.

Manabe, S., Smagorinsky, J., and Strickler, R. F., 1965, Mon. Wea. Rev. **93**, 769

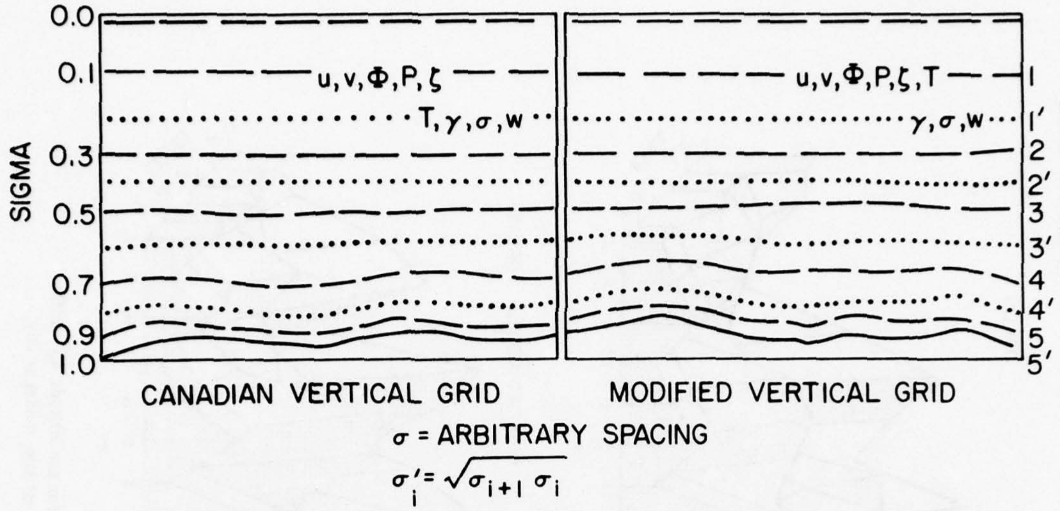

Fig. 1 – (a) The grid system used in the original CSFM and (b) the grid system used in the modified version.

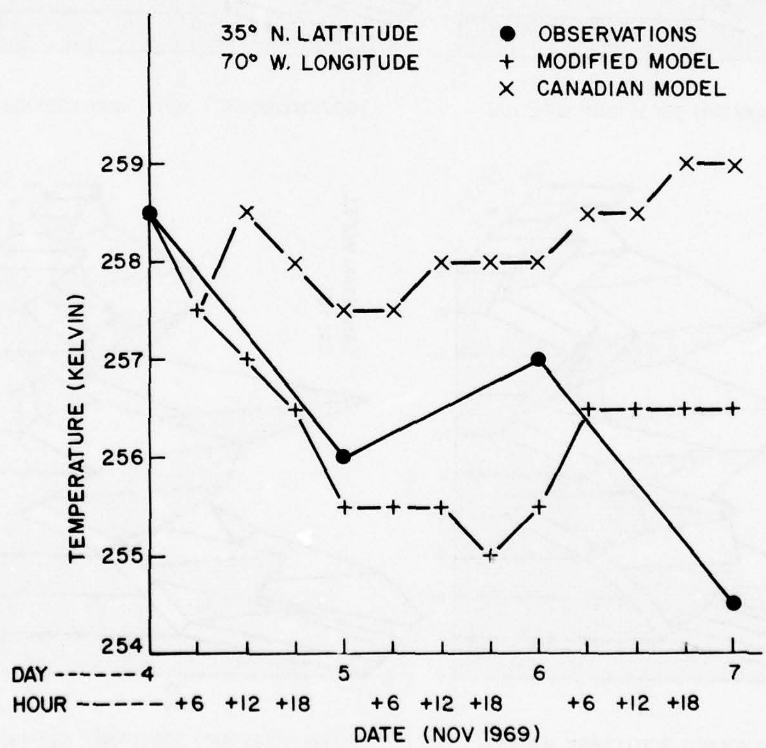

Fig. 2 – Temperature history for North East United States, showing GARP data [November 4-9, 1969], the predictions using the new model (–) and predictions using the old model (---).

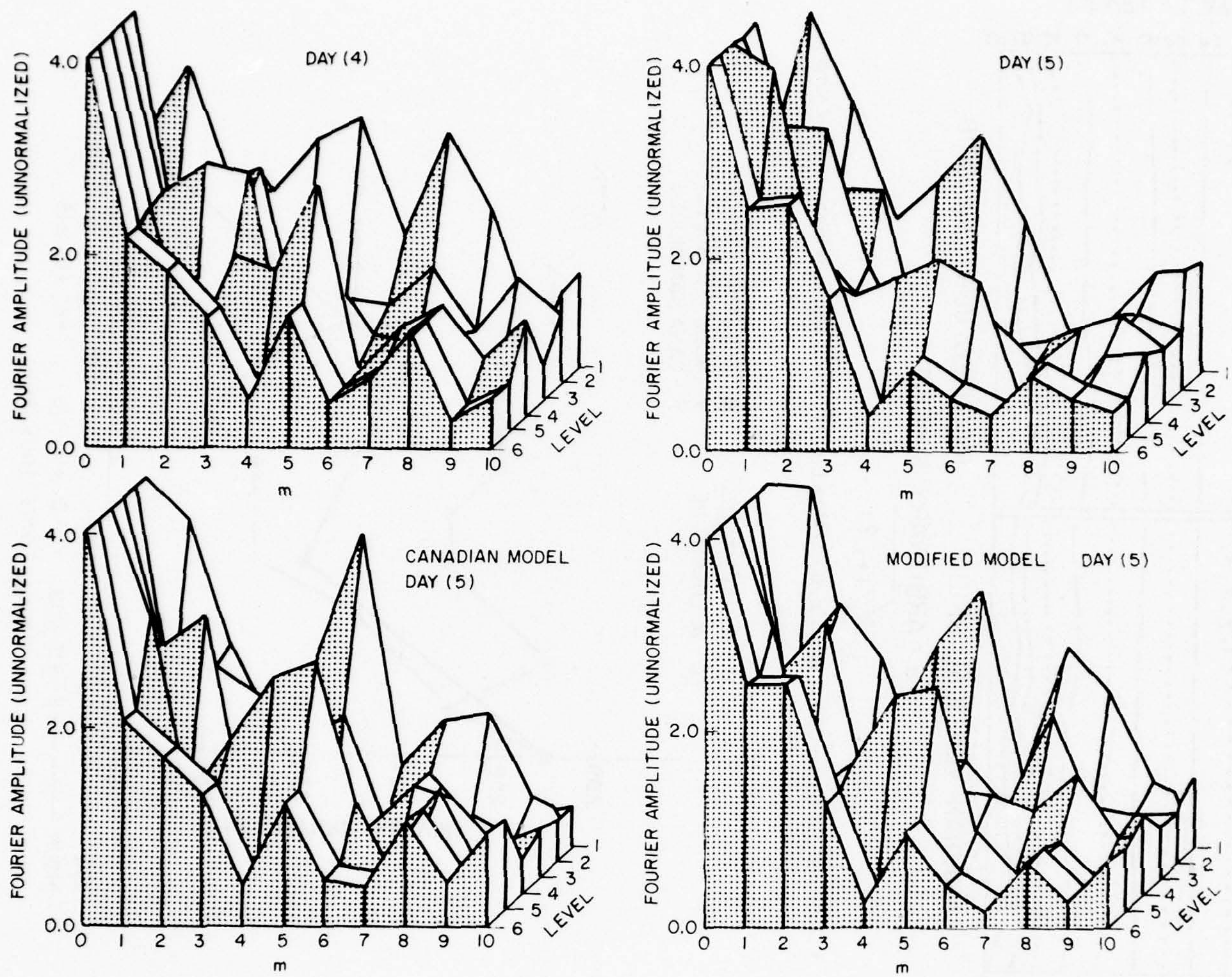

Fig. 3 – A comparison of the amplitudes of the Fourier components of temperature for the latitude 45° in the northern hemisphere: a) data at day 4; b) data at day 5; c) old model at day 5; d) new model at day 5.

Appendix I

Listing of the Spectral Forecast Model

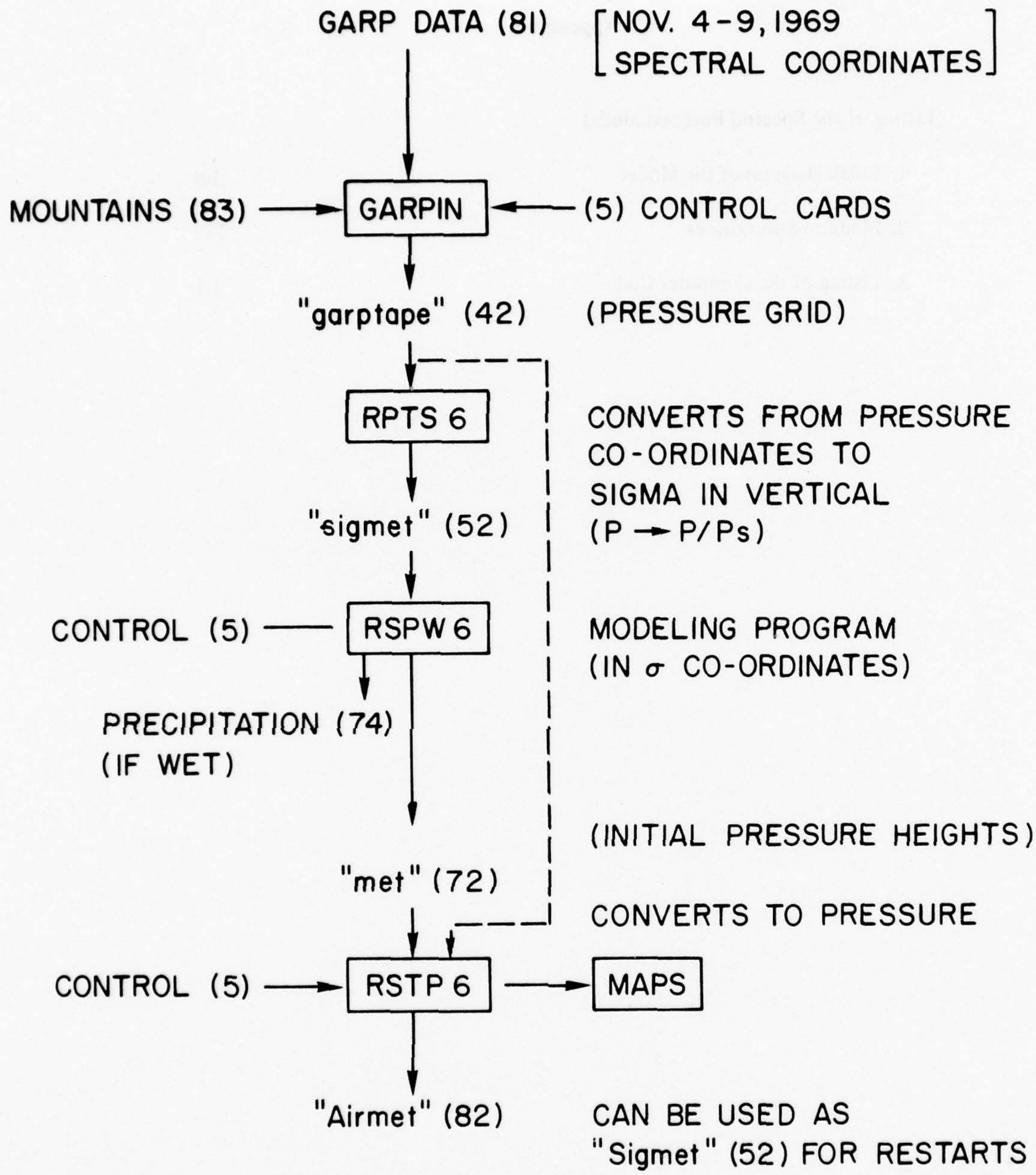

Fig. 4 – Block diagram of the interaction of the form modules of the model and the history tapes which they generate.

MEMBER

***Indicates a control program (see block diagram)**

```
      SUBROUTINE ABCII (DT, A, ILEV, IR)

      PARAMETER IL=5, I1=IL+1, IR$=25, IRM=2*IR$+1

      THIS SUBROUTINE CALCULATES THE VERTICAL DIFFERENCE MATRIX
      FOR THE IMPLICIT TIME STEP CALCULATION OF PEE.  SINCE THE MATRIX
      DEPENDS ON "DT", IT HAS TO BE CALLED EACH TIME DT IS CHANGED,
      NAMELY AT THE BEGINNING (DT=DEET/2) AND AT THE SECOND TIME
      STEP WHEN (DT=DEET)

      PEE(T) X (1+(DT/A)**2*L(L+1)*M) = RIGHT HAND SIDE
      M1 = THE TERM IN BRACKETS FOR EACH VALUE OF L (THE WAVE NUMBER)
      MPI = INVERSE OF EACH M1 SO IS DIMENSIONED MPI(ILEV,ILEV,IRM)
      M1 IS USED SOLEY AS A SCRATCH MATIRX

      REAL MI,MII,MIIM1,M1,M2,MP,MPM1,M,MW,MPI,M2M1
      COMMON / NEWMAT / MI(IL,IL),MII(IL,IL),MIIM1(IL,IL),M1(IL,IL),
     *                  M2(IL,IL),MP(IL,IL),MPM1(IL,IL),M(IL,IL),
     *                  MW(IL,IL),MPI(IL,IL,IRM),M2M1(IL,IL)

      IR2P1=2*IR+1
      DTS = (DT/A)**2
      DO 11 L = 1, IR2P1
      FNS=FLOAT((L-1)*L) * DTS

      DO 10 J = 1, ILEV
      DO 10 I = 1, ILEV
   10 M1(I,J) = FNS * M(I,J)
      DO 12 I = 1, ILEV
   12 M1(I,I) = M1(I,I) + 1.
   11 CALL MTXINV (MPI(1,1,L), M1, ILEV)

      RETURN
      END
```

```
      SUBROUTINE ALPAS2(ALP,LALP,LM,WRKS)

      * ALP(LALP,LM) CAN CONTAIN THE LEGENDRE POLYNOMIALS CALCULATED
      *  BY SUBROUTINE ALPMN2, OR THEIR N-S DERIVATIVES CALCULATED
      *  BY SUBROUTINE ALPDR2.
      * THE SYMMETRIC AND ANTISYMMETRIC VALUES IN EACH ROW OF ALP
      * ARE ORIGINALLY INTERLEAVED. E.G. ROW 1 = (0,1,2,3,...LALP).
      * THIS SUBROUTINE SEPARATES EACH ROW INTO TWO PARTS.
      *  E.G. ROW 1 = (0,2,4...LALP/2, 1,3,5...LALP ).
      * WRKS IS AN SCM WORK FIELD OF LALP WORDS.

      * WARNING - LALP MUST BE EVEN.

      DIMENSION ALP(LALP,1)
      DIMENSION WRKS(1)

      LALPH=LALP/2
      LALPH1=LALPH+1

      DO 30 M=1,LM

      * TRANSFER ONE ROW OF ALP TO WRKS.

      DO 20 N=1,LALP
   20 WRKS(N)=ALP(N,M)

      * PUT THE SYMMETRIC VALUES IN WORDS 1 TO LALP/2 OF ALP.

      NSYM=-1
      DO 22 N=1,LALPH
      NSYM=NSYM+2
   22 ALP(N,M)=WRKS(NSYM)

      * PUT THE ANTISYMMETRIC VALUES IN WORDS LALP/2+1 TO LALP.

      NASM=0
      DO 24 N=LALPH1,LALP
      NASM=NASM+2
   24 ALP(N,M)=WRKS(NASM)

   30 CONTINUE

      RETURN
      END
```

```
      SUBROUTINE ALPDR2(DALP,ALP,LALP,LM,EPSI)

     * CALCULATES N-S DERIVATIVES OF EACH ASSOCIATED LEGENDRE POLYNOMIAL
     * DALP(LALP,LM)    WILL CONTAIN N-S DERIVATIVE OF ALP.
     * ALP(LALP,LM)   CONTAINS LEGENDRE POLYNOMIALS FOR ONE LATITUDE.
     * EPSI(LALP,LM)    CONTAINS PREVIOUSLY CALCULATED CONSTANTS.

     * WARNING - LALP MUST BE EVEN.
     *         - LAST ELEMENT OF EACH ROW IS SET TO ZERO.

      DIMENSION DALP(LALP,1),ALP(LALP,1),EPSI(LALP,1)

      LALPM=LALP-1

      DO 30 M=1,LM

      DO 20 N=1,LALPM
      FNS=FLOAT(M+N-2)
      ALPILM=0.
      IF(N.GT.1) ALPILM=ALP(N-1,M)
      DALP(N,M)=(FNS+1.)*EPSI(N,M)*ALPILM - FNS*EPSI(N+1,M)*ALP(N+1,M)
   20 CONTINUE

   30 DALP(LALP,M)=0.

      RETURN
      END
```

```
      SUBROUTINE ALPNM2(ALP,LALP,LM,SINLAT,EPSI)

      * PUTS LEGENDRE POLYNOMIALS IN ALP(LALP,LM) FOR ONE LATITUDE.
      * SINLAT IS THE SINE OF THE REQUIRED LATITUDE.
      * EPSI IS A FIELD OF CONSTANTS THE SAME SIZE AS ALP.

      * THE SYMMETRIC AND ANTISYMMETRIC VALUES IN ALP ARE INTERLEAVED
      *  IN EACH ROW.    E.G. ROW 1 = ( 0,1,2,3....LALP ).

      * WARNING - LALP MUST BE EVEN.   ITS MINIMUM VALUE IS 4.

      DIMENSION ALP(LALP,1),EPSI(LALP,1)

      COS2=1.-SINLAT**2
      PROD=1.
      A=1.
      B=0.

      * LOOP 30 COVERS LONGITUDINAL WAVE NUMBERS 0 TO LM-1.

      DO 30 M=1,LM
      FM=FLOAT(M-1)
      IF(M.EQ.1) GO TO 12
      A=A+2.
      B=B+2.
      PROD=PROD*COS2*A/B

      * COMPUTE THE FIRST TWO ELEMENTS OF THE ROW.

   12 ALP(1,M)=SQRT(.5*PROD)
      ALP(2,M)=SQRT(2.*FM+3.)*SINLAT*ALP(1,M)

      * NOW COMPUTE ELEMENTS 3 TO LR IN THE ROW IN PAIRS.

      DO 20 N=3,LALP,2
      ALP(N,M)=(SINLAT*ALP(N-1,M) - EPSI(N-1,M)*ALP(N-2,M)) / EPSI(N,M)
      ALP(N+1,M)=(SINLAT*ALP(N,M) - EPSI(N,M)*ALP(N-1,M)) / EPSI(N+1,M)
   20 CONTINUE

   30 CONTINUE

      RETURN
      END
```

```
      SUBROUTINE BPFT(PEE,T,PS,PHIS,LA,LRS,LM,ILEV,TMEAN,RGAS,SF)

      * CALCULATES PEE FROM T,PS,PHIS,TMEAN BY INTEGRATING UP FROM THE G

      * IF MODEL IS HEMISPHERIC PEE,T ARE SYMMETRIC.
      * EACH LEVEL IS DIMENSIONED (LRS,LM).
      * EACH LEVEL IS SEPARATED BY LA COMPLEX WORDS.
      * PEE AND T MAY BE EQUIVALENCED.

      COMPLEX T(LA,1),PEE(LA,1),PHIS(1),PS(1)
      DIMENSION TMEAN(1),SF(1)
      COMPLEX PSURF

      ILEVP = ILEV + 1
      ILEVM = ILEV - 1

      DO 30 M=1,LM
      MR=(M-1)*LRS
      DO 30 N=1,LRS
      IL=MR+N

      PSURF = RGAS * TMEAN(ILEVP) * PS(IL)
      IF(IL.GT.1) PSURF = PSURF + PHIS(IL)

      PEE(IL,ILEV) = PSURF
     1             + SF(ILEV) * 0.5 * RGAS * (T(IL,ILEVP)+T(IL,ILEV))
     2             - RGAS * (TMEAN(ILEVP)-TMEAN(ILEV)) * PS(IL)
      DO 10 IHI = 1, ILEVM
      IH = ILEV - IHI
      PEE(IL,IH) = PEE(IL,IH+1)
      PEE(IL,IH) = PEE(IL,IH) + SF(IH)*0.5*RGAS*(T(IL,IH+1)+T(IL,IH))
   10 PEE(IL,IH) = PEE(IL,IH) - RGAS*PS(IL)*(TMEAN(IH+1)-TMEAN(IH))

   30 CONTINUE

      RETURN
      END
```

```
      SUBROUTINE CFVAL2(VAL,CFC,NW,RLON)

      * EVALUATES COMPLEX FOURIER SERIES IN CFC TO WAVE NUMBER NW
      *   AT POINT RLON (RADIANS).
      * CFC ACTUALLY CONTAINS THE POSITIVE HALF OF THE COMPLETE
      *   COMPLEX SERIES. THE MEAN IS IN CFC(1).

      DIMENSION CFC(1)


      NWP1=NW+1
      VAL=0.
      DO 20 MP=2,NWP1
      MM=MP+MP
      FMX=RLON*FLOAT(MP-1)
      SINMX=SIN(FMX)
      COSMX=COS(FMX)
      VAL=VAL+COSMX*CFC(MM-1)-SINMX*CFC(MM)
   20 CONTINUE
      VAL=2.*VAL+CFC(1)

      RETURN
      END
```

```
      FUNCTION CHIC(H,HC)

            FOR  H  BETWEEN  HC  AND  1.0
            *CHIC*  IS A LINEAR FUNCTION OF  H

      IF(H.GE.1.)  GO TO 1

      CHIC = (H-HC)/(1.-HC)
      RETURN

    1 CHIC = 1.0
      RETURN
      END
```

```
      SUBROUTINE CONADJ(T,Q,PCP,PRESSG,W,P,ST,NUPS,NSUPS,ILEVM,DEL)

      *     PERFORMS CONVECTIVE ADJUSTMENT
      *     COMPUTES BOTH LARGE AND SMALL SCALE PRECIPITATION
      *     COMPUTES EVAPORATION OF FALLING PRECIPITATION

      COMMON/ADJPCP/HC,HF,HM,AA,DEPTH,LHEAT,MOIADJ,MOIFLX
      DIMENSION T(1),Q(1),P(1),ST(ILEVM,1)
      DIMENSION DEL(1)
      LOGICAL ADJ,WET

      * GAM,GAC  STABILITY FACTOR AND ITS CRITICAL VALUE
      * HS       CRITICAL SATURATION RELATIVE HUMIDITY
      * DTF      TEMPERATURE CHANGE CAUSED BY A CONVECTIVE ADJUSTMENT
      * DTH      TEMPERATURE CHANGE CAUSED BY A RELEASE OF LATENT HEAT
      * DQF      MOISTURE CHANGE CAUSED BY A CONVECTIVE ADJUSTMENT
      * DQR      MOISTURE EQUIVALENT TO PRECIPITATION AMOUNT
      * DQI,DQJ  SATURATION DEFICITS

      PCP=0.0
      ITER=0
      PCPCH=PRESSG*DEPTH
      WET=.FALSE.

    5 ADJ=.FALSE.
      ITER=ITER+1
      DQR=0.0

      DO 50 I=1,ILEVM

      J=I+1
      GAC=0.0
      DTF=0.0
      DTH=0.0
      DQF=0.0
      DQH=0.0
              **************************
              * CONVECTIVE HEAT FLUX *
              **************************
      *     COMPUTE  GAM
      TT=ST(I,1)*T(I)+ST(I,2)*T(J)
      GAM=TT+ST(I,3)*(T(I)-T(J))
      *     COMPUTE  H
      H=Q(J)/SPCHUM(T(J),P(J))
      *     COMPUTE  HS
      HS=AMIN1(H,1.)
      IF(W.GE.0.0)                   GO TO 15
      IF(H.LE.HM.OR.ITER.NE.1)       GO TO 10
      HS=CRIRLH(HM,H,AA)
   10 IF(H*MOIADJ.LT.HC)             GO TO 15
```

```
      *    COMPUTE  GAC
      QST=SPCHUM(TT,P(J)*ST(I,6))
      GAC=CHIC(H,HC)*GAMSAT(TT,QST)
   15 IF(GAM.GE.GAC)                         GO TO 20

      IF(GAC.NE.0.0)  WET=.TRUE.
      DTF=ST(I,4)*(GAC-GAM)
      T(I)=T(I)+DTF*ST(I,5)
      T(J)=T(J)+DTF
                ******************
                * MOISTURE FLUX *
                ******************
   20 QSJ=HS*SPCHUM(T(J),P(J))
      DQJ=QSJ-Q(J)
      IF(DQJ.GE.0.0)                         GO TO 45
      IF(DTF*MOIFLX.EQ.0.0)                  GO TO 30
      QSI=AMAX1(HS,HF)*SPCHUM(T(I),P(I))
      DQI=QSI-Q(I)
      IF(DQI.LE.0.0)                         GO TO 30

      DQF=AMAX1(DQI/ST(I,5),DQJ)
      Q(I)=Q(I)+DQF*ST(I,5)
      Q(J)=Q(J)+DQF
      DQJ=QSJ-Q(J)
   30 IF(HS.LT.AMIN1(HC,HM))                 GO TO 45
      ******************************************************
      * CONVECTIVE OR STABLE HEATING BY CONDENSATION *
      ******************************************************
      DQH=DELTAQ(T(J),QSJ,DQJ,HS)
      DQH=LHEAT*(DQH-DQJ)+DQJ
      DTH=-HTVOCP(T(J))*DQH*LHEAT
      T(J)=T(J)+DTH
      Q(J)=Q(J)+DQH
       ******************************************
       * CONVECTIVE OR STABLE PRECIPITATION *
       ******************************************
      DQR = DQR + DEL(I+1)*DQH

   45 IF(DTF*(DTF-DTH).GT.0.01)  ADJ=.TRUE.

   50 CONTINUE

      PCP = PCP - DQR*PCPDH

      IF(ADJ)                                GO TO  5

      IF(ITER.EQ.1)                          GO TO 60
      NUPS=NUPS+1
      IF(WET)  NSUPS=NSUPS+1

   60 RETURN
```

END

```
      SUBROUTINE CONTOR(O,XV,YQ,NX,NY,NLIN,MCONT,NX1,NX2,NY1,NY2,TITLE,
     *SUBTIT,VARI,RINC,ROR)

      THIS CONTOUR ROUTINE IS GENERATED TO REPLACE ROBERT'S ROUTINE
      THE CONTROL VARIABLES ARE THE SAME AS IS THE CALL. THE ADDITIONAL
      VARIABLES FINC AND FOR ARE USED IF THE USER WISHES TO SPECIFY
      THE ORIGIN AND INCREMENT HIMSELF. ORIGIN AND INCREMENT ARE PICKED
      IF FINC IS SET TO ZERO . PROGRESSIVE CHARACTER SETS FOR PLOTTING
      ARE USED: NUMBERS  FOR CONTOURS GREATER THAN ORIGIN,LETTERS
      FOR PLOTS LESS THAN ORIGIN. CHANGE THE EXTERNAL CALL FOR VARIOUS
      SITUATIONS:
               RPLOT - LARGE PLOT, DENSE POINT FIELD
               IPLOT - LARGE PLOT, SPARSE POINT FIELD

      EXTERNAL RPLOT
      EXTERNAL IPLOT
      DIMENSION O(NX,NY)
      DIMENSION XV(NX),YQ(NY),D(100),D1(100),D2(100)
      INTEGER TITLE(5),SUBTIT(2),CHAR1(10),CHAR2(10),ZERO,TIT(126),DOLLAR
     1R,BLANK
      DATA ZERO/'0***'/,BLANK/'    '/,DOLLAR/'$$$$'/
      DATA CHAR1/'1   2   3   4   5   6   7   8   9   *   '/
       DATA CHAR2/'A   B   C   D   E   F   G   H   I   J   '/
      FINC=RINC
      FOR=ROR
      NCONT=MCONT
      DO 88 J=2,6
88    TIT(J)=TITLE(J-1)
      TIT(7)=DOLLAR

      DETERMING MAXIMUM AND MINIMUM

      NS=NX
1118  I1=MAXVAL(O)+1
      I2=MINVAL(O)+1
      IK1=I1/NS
      IK2=I2/NS
      IJ2=I2-(IK2*NS)
      IJ1=I1-(IK1*NS)
      IK1=IK1+1
      IK2=IK2+1
      FMX=O(IJ1,IK1)
      FMN=O(IJ2,IK2)
      IF(ABS(FMX-FMN).GT.1.0E-9) GO TO 520
      WRITE(6,98456)
      RETURN
520   CONTINUE
98456 FORMAT(' VALUE IS TOO SMALL FOR THIS PLOTTER')
      IF(FINC.NE.0) GO TO 60

      SELECT CONTOURS
```

```
98      IF(NCONT.GT.MCONT) NCONT=MCONT
        A=ALOG10((FMX-FMN)/NCONT)+.1505
        I=A
        IF (A.LT.0)I=I-1
         B=I
        C=A-I
        FINC=10.**B
        IF(C.LT..1505) GO TO 9
        FINC=10.**B*1.5
        IF(C.LT..3010) GO TO 9
        FINC=10.**B*2
        IF(C.LT..4516) GO TO 9
        FINC=10.**B*3
        IF(C.LT..6505) GO TO 9
        FINC=10.**B*5
        IF(C.LT..8490) GO TO 9
        FINC=10.**B*7.5
9       CONTINUE
        I=FMN/FINC
        FOR=I*FINC
        IF(FMX*FMN.LE.0)FOR=0
60      I=1
        RST=-1
        IF(FMN.GT.FOR) RST=1

        DETERMINE ORIGIN

        D(1)=FOR
20      IF(D(1).LE.FMN.AND.D(1)+FINC.GE.FMN) GO TO 10
        D(1)=D(1)+FINC*RST
        GO TO 20
10      CONTINUE
        I=I+1
        IF(I.GT.MCONT+20) WRITE(6,455)I
        IF(I.GT.MCONT+20) GO TO 98
455      FORMAT(' NUMBER OF CONTOURS EXCEEDS MCONT (',I6,')AUTOMATIC
       1SCALING BEGUN')
446     FORMAT(' ONLY ONE CONTOUR  PLOTTED,RESCALING')
        D(I)=D(I-1)+FINC
        IF(D(I).LT.FMX) GO TO 10
        IF(I.LE.2) WRITE(6,446)
        IF(I.LE.2) GO TO 98
        NCON=I
        TIT(1)=BLANK
        CALL NOGRID
        REMOVES GRID
        CALL USXUSY(XV(1),XV(NX),.FALSE.,YQ(1),YQ(NY),.FALSE.,TIT)

        PLOT ORIGIN
```

```
201     I1=0
        I2=0
        DO 11 J=1,NCON
        IF(ABS(D(J)-FOR).LT.1.0E-4*FINC) GO TO 11
        IF(D(J).LT.FOR) GO TO 70
        I1=I1+1
        D1(I1)=D(J)
70      IF(D(J).GT.FOR) GO TO 11
        I2=I2+1
        D2(I2)=D(J)
11      CONTINUE

        PLOT CONTOURS GREATER THAN ORIGIN

        IF(I1.EQ.0) GO TO 200
        DO 100 J=1,I1
        M=(J-1)*0.1 +1.0E-4
        TIT(1)=CHAR1(J-M*10)
100     CALL CNTOUR(D,NS,XV,1,NX,YQ,1,NY,D1(J),IPLOT,TIT)
200     I3=I2
        IF(I3.LE.0) GO TO 300

        PLOT CONTOURS LESS THAN ORIGIN

        DO 101 J=1,I3
        I=I2-J+1
        M=(J-1)*0.1 +1.0E-4
        TIT(1)=CHAR2(J-M*10)
101     CALL CNTOUR(D,NS,XV,1,NX,YQ,1,NY,D2(I),IPLOT,TIT)
300     TIT(1)=ZERO
        CALL CNTOUR(D,NS,XV,1,NX,YQ,1,NY,FOR,IPLOT,TIT)
        WRITE(6,51) FMX,FMN,FOR,FINC
        WRITE(6,52) SUBTIT(1),SUBTIT(2),VARI
51      FORMAT(' MAXIMUM=',G12.4,'  MINIMUM=',G12.4,'  ORIGIN (0) =',G12.4
       1,'  INCREMENT=',G12.4)
52      FORMAT(' ',2A4,' = ',G12.3)
        RETURN
        END
```

```
FUNCTION CRIRLH(HM,H,AA)

* COMPUTES  *CRIRLH*  ,THE CRITICAL SATURATION RELATIVE HUMIDITY

IF(H.LE.1.)  DH = AA*(H-HM)**3
IF(H.GT.1.)  DH=AA*(2.-HM-H)**3 + H - 1.
CRIRLH= H - DH
RETURN
END
```

```
      SUBROUTINE DELEK(CT,LA,LRS,LM,EF,ILH,ILEV,ALP,DELALP,
     1                 LALP,KHEM,IR,ILG,WOCSL,WRKS)

  * THE LAPLACIAN OF KINETIC ENERGY MUST BE ADDED TO THE
  * DIVERGENCE TENDENCY CT. THIS IS ACTUALLY A LINEAR TERM AND COULD
  * BE DONE SPECTRALLY, BUT IT IS PERFORMED ON THE GRID TO SAVE
  * THE SPACE IN CORE OF A COMPLEX SPECTRAL E FIELD.

  * THE TERM IS COMPUTED AND ADDED TO CT THE NORMAL WAY SIMPLY
  * BY CONVERTING ALP TO N*(N+1)*ALP AND USING FASP2.

  * EF(ILG,ILAT) = GRID VALUES OF (U**2+V**2).   (ILH=ILG/2)
  * ALP = LEGENDRE POLYNOMIALS.
  * DELALP = WORK FIELD FOR N(N+1)ALP.

  * NOTE - IN THE PROGRAM DALP IS USED FOR DELALP SO THIS ROUTINE
  *        MUST BE CALLED LAST IN THE LATITUDE LOOP.


      COMPLEX CT(LA,1),EF(ILH,1)
      DIMENSION ALP(LALP,1),DELALP(LALP,1)


  * COMPUTE N*(N+1)*ALP IN DELALP.
  * IF MODEL IS HEMISPHERIC SYMMETRIC VALUES OF ALP ARE IN
  * THE FIRST HALF OF EACH ROW IN ALP.

      LR=LALP
      IF(KHEM.NE.0) LR=LALP/2

      DO 210 M=1,LM
      DO 210 N=1,LR
      NS=M+N-2
      IF(KHEM.NE.0) NS=NS+(N-1)
      FNS=FLOAT(NS*(NS+1))
      DELALP(N,M)=FNS*ALP(N,M)
  210 CONTINUE

  * CONVERT GRID VALUES TO FOURIER COEFFICIENTS.
  * INCLUDE IN WOCSI THE FACTOR .5 IN THE DEFINITION OF KE.
  * ADD DELSQ(KE) TO CT.

      CALL FFWFG2(EF,ILH,EF,ILG,IR,ILG,WRKS,ILEV)
      WOCSI=WOCSL*.5
      IF(KHEM.NE.0) WOCSI=WOCSL

      DO 310 L=1,ILEV
  310 CALL FASP2(CT(1,L),LRS,LM,EF(1,L),DELALP,LALP,WOCSI)

      RETURN
      END
```

THIS PAGE IS BLANK

```
FUNCTION DELTAQ(T,QS,DQ,HS)

* COMPUTES THE MOISTURE RELEASE  *DELTAQ*  AT TEMPERATURE  T
* WHEN THE SPECIFIC HUMIDITY IS GREATER THAN ITS CRITICAL VALLE QS
* BY AN AMOUNT  DQ  .
* HS  IS THE CRITICAL RELATIVE HUMIDITY TO WHICH  QS  CORRESPONDS
* REF. LANGLOIS,TELLUS 25,1973,86-87

COMMON/EPS/A,B,EPS1,EPS2

EPH=EPS2/(HS*EPS1)
 H=HTVOCP(T)
 Y=B/(T*T)*QS*(1.+QS*EPH)
HY=H*Y
HZ=HY/(T*(1.+HY))

DELTAQ = DQ/(1.+HY) * (1.0+DQ/Y*HZ*HZ*(B*(QS*EPH+0.5)-T) )

RETURN
END
```

```
FUNCTION DEWPNT(Q,P)

* COMPUTES DEW POINT TEMPERATURE *DEWPNT* USING
* SPECIFIC HUMIDITY  Q  AND PRESSURE  P(MB)
*  E  IS THE VAPOR PRESSURE RELATED TO  *DEWPNT* BY
*                             E=EXP(A-B/DEWPNT)

COMMON/EPS/A,B,EPS1,EPS2

E=Q*P/(EPS1+EPS2*Q)

DEWPNT=B/(A-ALOG(E))

RETURN
END
```

```
      SUBROUTINE DFDS (G,F,S,NN,CON1,CON2)
      DIMENSION G(NN),F(NN),S(NN)

      * GIVEN A FUNCTION F AT NN UNEVENLY SPACED POINTS, THIS ROUTINE
      *    CALCULATES ITS FIRST DIFFERENCE G AT THESE POINTS.

      * S MUST CONTAIN THE INVERSE OF THE INTERVAL LENGTHS.
      * BOUNDARY CONDITIONS SPECIFIED BY CON1,CON2, (SEE BELOW).

      NNM=NN-1
      DO 10 N=1,NNM
   10 G(N+1)=S(N)*(F(N+1)-F(N))

      A=G(2)

      DO 20 N=2,NNM
   20 G(N)=(S(N)*G(N+1)+S(N-1)*G(N))/(S(N)+S(N-1))

      * BOUNDARIES

      G(1)=CON1*A+(1.-CON1)*G(2)
      G(NN)=CON2*G(NN) +(1.-CON2)*G(NNM)

      RETURN
      END
```

```
      SUBROUTINE DFDSM(G,F,S,NN)

*  GIVEN A FUNCTION F AT NN UNEVENLY SPACED POINTS, THIS ROUTINE
*     CALCULATES ITS FIRST DIFFERENCE G AT THESE INTERNAL POINTS.
*  S MUST CONTAIN THE INVERSE OF THE INTERVAL LENGTHS.

      DIMENSION G(1),F(1),S(1)


      NNM=NN-1
      DO 10 N=1,NNM
   10 G(N+1)=S(N)*(F(N+1)-F(N))

      DO 20 N=2,NNM
   20 G(N)=(S(N)*G(N+1)+S(N-1)*G(N))/(S(N)+S(N-1))

      RETURN
      END
```

```
SUBROUTINE DFDSQD(G,F,PR,S,NN)

THIS ROUTINE IS THE SAME AS DFDS EXCEPT THAT THE END DERIVATIVES A
FOUND FROM DIFFERENTIATING THE QUADRATIC THROUGH THE THE LAST THRE
POINTS AT EACH END.
CLEARLY , NN MUST BE GREATER THAN 2.
PR ARE THE COORDINATES OF THE DATA , AND S ARE THE INVERSE OF THE
SEPERATION - IDENTICAL TO S IN DFDS.

DIMENSION G(1),F(1),PR(1),S(1)


CALL DFDSM(G,F,S,NN)

G(1)  = F(1)*(2*PR(1)-PR(2)-PR(3))/((PR(1)-PR(2))*(PR(1)-PR(3))) +
1        F(2)*(PR(1)-PR(3))/((PR(2)-PR(1))*(PR(2)-PR(3)))         +
2        F(3)*(PR(1)-PR(2))/((PR(3)-PR(1))*(PR(3)-PR(2)))

G(NN) = F(NN-2)*
1        (PR(NN)-PR(NN-1))/((PR(NN-2)-PR(NN-1))*(PR(NN-2)-PR(NN)))+
2        F(NN-1)*
3        (PR(NN)-PR(NN-2))/((PR(NN-1)-PR(NN-2))*(PR(NN-1)-PR(NN)))+
4        F(NN)*(2*PR(NN)-PR(NN-1)-PR(NN-2))   /
5        ((PR(NN)-PR(NN-2))*(PR(NN)-PR(NN-1)))

RETURN
END
```

```
SUBROUTINE DIMCAL(LRS,LRA,LRU,LRV,LALP,LM,LA,LAW,IR,KHEM)

* COMPUTES ROW LENGTHS AND DIMENSIONS FOR SPECTRAL ARRAYS
* GIVEN THE RESOLUTION (IR) AND THE REGION OF INTEGRATION (KHEM).

* IF THE MODEL IS GLOBAL LRS=LRA=IR+1, LRU=LRV=LALP=IR+2.

* IF THE MODEL IS HEMISPHERIC ALL FIELDS ARE EITHER SYMMETRIC
*  OR ANTISYMMETRIC, AND THE ZEROES ARE SQUEEZED OUT.

* LRS = ROW LENGTH OF SYMMETRIC FIELDS
* LRA = ROW LENGTH OF ANTISYMMETRIC FIELDS
* LRU = ROW LENGTH OF E-W WIND COMPONENT FIELD
* LRV = ROW LENGTH OF N-S WIND COMPONENT FIELD
* LALP = ROW LENGTH OF LEGENDRE POLYNOMIALS (ALP) AND
*         ASSOCIATED ARRAYS (DALP,EPSI).
* LM   = NUMBER OF ROWS IN EACH FIELD.
* LAW  = SEPARATION BETWEEN SUCCESSIVE LEVELS OF WIND FIELDS.
* LA   = SEPARATION BETWEEN SUCCESSIVE LEVELS OF ALL OTHER FIELDS.

* IR = RESOLUTION OF THE MODEL.
* KHEM=0 FOR GLOBAL, 1 FOR N HEM, 2 FOR S HEM.


LRS=IR+1
IF(KHEM.NE.0) LRS=IR/2+1

LRA=IR+1
IF(KHEM.NE.0) LRA=(IR+1)/2

LRU=LRS+1
IF(KHEM.NE.0.AND.MOD(IR,2).EQ.0) LRU=LRS

LRV=LRA+1
IF(KHEM.NE.0.AND.MOD(IR,2).NE.0) LRV=LRA

LALP=IR+2
IF(MOD(IR,2).NE.0) LALP=LALP+1

LM=IR+1
LA=LRS*LM
LAW=LRU*LM

RETURN
END
```

```
      SUBROUTINE EGRAFS(ETOTS,LTOT,NSPLAT,IGE,IGM,IGQ,IGD)

      * GRAPH ENERGIES,MOMENTUM,VORTICITY,DIVERGENCE IF REQUESTED.
      * THE SIX LEVELS OF ETOT CONTAIN NSPLAT VALUES OF...
      *      1) KINETIC ENERGY
      *      2) POTENTIAL ENERGY
      *      3) TOTAL ENERGY
      *      4) E-W MOMENTUM
      *      5) MEAN SQUARE VORTICITY
      *      6) MEAN SQUARE DIVERGENCE
      * EACH IS A MEAN VALUE OF ALL THE LEVELS IN THE MODEL
      *  AT ONE TIMESTEP.

      DIMENSION ETOTS(LTOT,6)

      IF(NSPLAT.LE.0) RETURN

      * SUBTRACT INITIAL VALUES OF KE,PE AND TE TO GET CHANGES ONLY.

      DO 20 J=1,3
      EZERO=ETOTS(1,J)
      DO 20 I=1,NSPLAT
      ETOTS(I,J)=ETOTS(I,J)-EZERO
   20 CONTINUE

      * NOW DRAW THE GRAPHS. EACH GRAPH PICKS ITS OWN SCALE.
      * GRAPHS TURNED OFF IF IGE,IGM,IGQ,IGD SET TO ZERO RESPECTIVELY.
      *  OTHERWISE THESE NUMBERS ACT AS THE GRAPH INTERVALS.

      CALL SPLAT2(ETOTS(1,1),LTOT,3,NSPLAT,IGE,0.,0.)
      CALL SPLAT2(ETOTS(1,4),LTOT,1,NSPLAT,IGM,0.,0.)
      CALL SPLAT2(ETOTS(1,5),LTOT,1,NSPLAT,IGQ,0.,0.)
      CALL SPLAT2(ETOTS(1,6),LTOT,1,NSPLAT,IGD,0.,0.)

      RETURN
      END
```

```
      SUBROUTINE EKLAT(TOTK,EG,PRESSG,ILG,ILEV,WOCSI,KHEM)

*     THIS SUBROUTINE IS PURELY DIAGNOSTIC.

*     THIS SUBROUTINE CALCULATES KINETIC ENERGY AT ILEV LEVELS
*      IN TOTK FOR LATER USE IN SUBROUTINE ENERD.
*     IF ENERD IS REMOVED FROM THE PROGRAM, EKLAT CAN ALSO BE.

*     EG = U**2+V**2
*     PRESSG = SURFACE PRESSURE

      DIMENSION EG(ILG,1),PRESSG(1),TOTK(1)


      WXX=.5*WOCSI/FLOAT(ILG)
      IF(KHEM.NE.0) WXX=WXX+WXX

      DO 20 IK=1,ILG
      DO 20 L=1,ILEV
   20 TOTK(L)=TOTK(L)+EG(IK,L)*PRESSG(IK)*WXX

      RETURN
      END
```

```
      SUBROUTINE ENERD (P,C,T,U,PS,PHIS,PRESS,LA,LAW,
     1           LRS,LRA,LRU,LM,ILEV,KHEM,TMEAN,TMEANH,SF,DS,
     2               TOTK,TOTP,TOTE,TOTM,PSITOT,CHITOT,ETOT)

      * THIS SUBROUTINE IS PURELY DIAGNOSTIC.
      * CALCULATES ENERGETICS FOR SPECTRAL MULTILEVEL MODEL.
      * NOTE - TOTK HAS BEEN CALCULATED PREVIOUSLY.

      * IF MODEL IS HEMISPHERIC, P IS ANTISYMMETRIC.
      *                           ALL OTHER FIELDS ARE SYMMETRIC.

      COMPLEX P(LA,1),C(LA,1),T(LA,1),U(LAW,1)
      COMPLEX PS(1),PHIS(1),PRESS(1)

      DIMENSION TMEAN(1),TMEANH(1),SF(1),DS(1),ETOT(6)
      DIMENSION TOTK(1),TOTP(1),TOTE(1),TOTM(1),PSITOT(1),CHITOT(1)

      COMMON/PARAMS/ WW,TW,A,ASQ,GRAV,RGAS,RGOCP,RGOASQ,CPRES

      COMPLEX TBAR(15),TTOP,PRXK,PRXP,PRXM

      ILEVM=ILEV-1
      SQRT2=SQRT(2.0)
      CPOASQ=3.5*RGAS/ASQ
      PBAR=PRESS(1)/SQRT2
      CONS=ASQ/PBAR*0.5
      CONSM=SQRT(8.0/5.0)/3.0*WW

      DO 1 IH=1,ILEV
      TOTE(IH)=0.0
      TOTM(IH)=0.0
      TOTP(IH)=0.0
      PSITOT(IH)=0.0
      CHITOT(IH)=0.0
    1 CONTINUE

      * POTENTIAL ENERGY.

      DO 40 M=1,LM
      DO 40 N=1,LRS
      IL=(M-1)*LRS+N
      TTOP = (SF(2)*T(IL,1) + SF(1)*T(IL,2))/(SF(1)+SF(2))
      TTOP = 1.5*T(IL,1)-0.5*TTOP
      DO 36 IH=1,ILEVM
   36 TBAR(IH) = T(IL,IH)*CPOASQ
      TBAR(1) = TBAR(1)+0.5*TTOP*CPOASQ
      TBAR(ILEV) = 0.5*T(IL,ILEV)*CPOASQ
      IF(IL.GT.1)GO TO 37
      DO 38 IH = 1,ILEVM
   38 TBAR(IH) = TBAR(IH)+TMEANH(IH)*SQRT2*CPOASQ
      TBAR(1) = TBAR(1)+0.5*TMEAN(1)*SQRT2*CPOASQ
```

```
      TBAR(ILEV) = TBAR(ILEV)+0.5*TMEANH(ILEV)*SQRT2*CP0ASQ
   37 CONTINUE
      DO 39 IH = 1,ILEV
   39 TBAR(IH) = TBAR(IH) + PHIS(IL)*FLOAT(ILEV)/ASQ
      DO 40 IH=1,ILEV
      PRXP=TBAR(IH)*CONJG(PRESS(IL))
      PRXP=PRXP+CONJG(PRXP)
      IF(M.EQ.1) PRXP=PRXP*0.5
      TOTP(IH)=TOTP(IH)+PRXP
   40 CONTINUE

      * MOMENTUM.

      DO 60 L=1,ILEV
      DO 60 M=1,LM
      DO 60 N=1,LRS
      MN=(M-1)*LRU+N
      MX=(M-1)*LRS+N
      PRXM=U(MN,L)*CONJG(PRESS(MX))
      PRXM=PRXM+CONJG(PRXM)
      IF(M.EQ.1) PRXM=PRXM*0.5
      TOTM(L)=TOTM(L)+PRXM
   60 CONTINUE

      * SQUARE DIVERGENCE.

      DO 65 L=1,ILEV
      DO 65 M=1,LM
      DO 65 N=1,LRA
      MN=(M-1)*LRA+N
      PSISQ=P(MN,L)*CONJG(P(MN,L))
      IF(M.GT.1) PSISQ=PSISQ+PSISQ
      PSITOT(L)=PSITOT(L)+PSISQ
   65 CONTINUE

      * SQUARE VORTICITY.

      DO 70 L=1,ILEV
      DO 70 M=1,LM
      DO 70 N=1,LRS
      MN=(M-1)*LRS+N
      CHISQ=C(MN,L)*CONJG(C(MN,L))
      IF(M.GT.1) CHISQ=CHISQ+CHISQ
      CHITOT(L)=CHITOT(L)+CHISQ
   70 CONTINUE

      * COMPUTE VERTICAL SUMS IN ETOT.

      DO 83 N=1,6
   83 ETOT(N)=0.
      K=3
```

```
      IF(KHEM.NE.0) K=2
      DO 85 IH=1,ILEV
      TOTM(IH)=(TOTM(IH)-CONSM*PRESS(K))*CONS
      TOTK(IH)= TOTK(IH)*CONS
      TOTP(IH)= TOTP(IH)*CONS
      CHITOT(IH)=SQRT(CHITOT(IH)*0.5)
      PSITOT(IH)=SQRT(PSITOT(IH)*0.5)
      TOTE(IH)= TOTK(IH)+TOTP(IH)
      ETOT(1)=ETOT(1)+DS(IH)*TOTK(IH)
      ETOT(2)=ETOT(2)+TOTP(IH)/FLOAT(ILEV)
      ETOT(4)=ETOT(4)+DS(IH)*TOTM(IH)
      ETOT(5)=ETOT(5)+DS(IH)*PSITOT(IH)
      ETOT(6)=ETOT(6)+DS(IH)*CHITOT(IH)
   85 CONTINUE
      ETOT(3)=ETOT(1)+ETOT(2)

      RETURN
      END
```

```
      SUBROUTINE ENOUT(P,C,T,U,PS,PHIS,PRESS,TMEANH,TMEAN,SF,DS,
     1                     TOTK,IR,ILEV,KHEM,ETOTS,LTOT,NOGR,IEPR,DIVCH)

      * THIS SUBROUTINE IS PURELY DIAGNOSTIC.
      * CALCULATES ENERGETICS FOR SPECTRAL MULTILEVEL MODEL.
      * PRINTS DIAGNOSTICS AND FILLS GRAPH ARRAY IF REQUESTED.
      * NOTE - TOTK HAS BEEN CALCULATED PREVIOUSLY.


      PARAMETER $ILV=15,$LEV=0,$LV=10,$ILT=52,$ILG=64,$ILTH=26,$IR=20
      COMPLEX P(1),C(1),T(1),U(1)
      COMPLEX PS(1),PHIS(1),PRESS(1)
      DIMENSION ETOTS(LTOT,6)

      DIMENSION TMEAN(1),TMEANH(1),SF(1),DS(1),TOTK(1)
      COMMON/TIMES/ DEET,KOUNT,KSTART,KTOTAL,IFDIFF

      DIMENSION TOTP($ILV),TOTE($ILV),TOTM($ILV),PSITOT($ILV),
     *CHITOT($ILV),ETOT(6)


      NSPLAT=KOUNT-KSTART+1
      CALL DIMCAL(LRS,LRA,LRU,LRV,LALP,LM,LA,LAW,IR,KHEM)
      CALL ENERD        (P,C,T,U,PS,PHIS,PRESS,LA,LAW,
     1           LRS,LRA,LRU,LM,ILEV,KHEM,TMEAN,TMEANH,SF,DS,
     2              TOTK,TOTP,TOTE,TOTM,PSITOT,CHITOT,ETOT)

      * ETOTS HOLDS THE MEAN VALUES TO BE GRAPHED AT THE END OF THE RUN.
      * IF NOGR=0 NO GRAPHS ARE DRAWN AND ETOTS IS NOT FILLED.

      IF(NOGR.EQ.0) GO TO 84
      DO 82 J=1,6
   82 ETOTS(NSPLAT,J)=ETOT(J)

      * IEPR IS THE INTERVAL AT WHICH THE ENERGIES ARE TO BE PRINTED.
      * IF IEPR=0 NO PRINTING IS DONE.

   84 IF(IEPR.LE.0) GO TO 86
      IF(MOD(NSPLAT-1,IEPR).NE.0) GO TO 86
      PS00=PS(1)/SQRT(2.)
      PBAR=PRESS(1)/SQRT(2.0)
      WRITE(6,6080) KOUNT,PS00,PBAR,ETOT
      WRITE(6,6082)(L,TOTK(L),TOTP(L),TOTE(L),TOTM(L),PSITOT(L),
     1           CHITOT(L),L=1,ILEV)

      * IF THE MEAN DIVERGENCE EXCEEDS DIVCH THE MODEL IS FORCED
      * TO TERMINATE BY SETTING KOUNT = KTOTAL + 1.
      * IF DIVCH=0. THIS TEST IS NOT PERFORMED.

   86 IF(DIVCH.GT.0. .AND. ETOT(6).GT.DIVCH) KOUNT=KTOTAL+1
      RETURN
```

```
6080 FORMAT(7HOKOUNT=,I4,10X,11HLN SF PRES=,E14.7,18X,8HSF PRES=,E14.7
    1     //12H    TOTAL  K=,E14.7,6H     P=,E14.7,6H  K+P=,E14.7,
    2          6H  MOM=,E14.7,6H VORT=,E14.7,6H   DIV=,E14.7)
6082 FORMAT(1H ,I5,6E20.7)
     END
```

```
      SUBROUTINE EPSIL2(EPSI,LALP,LM)

      * CALCULATES EPSILON(N,M) = SQRT((N**2-M**2)/(4*N**2 - 1))
      * FOR N FROM 0 TO LALP-1, AND M FROM 0 TO LM-1 IN EPSI(LALP,LM).
      * EPSILON IS A FIELD OF CONSTANTS USED IN THE SPECTRAL MODELS.

      DIMENSION EPSI(LALP,1)


      DO 20 M=1,LM
      MS=M-1
      N1=1
      IF(MS.EQ.0) N1=2
      DO 20 N=N1,LALP
      NS=MS+N-1
      FNUM=FLOAT(NS**2 - MS**2)
      FDEN=FLOAT(4*NS**2 - 1)
   20 EPSI(N,M)=SQRT(FNUM/FDEN)
      EPSI(1,1)=0.

      RETURN
      END
```

```
      SUBROUTINE FASP2(SC,LR,LM,CFC,ALP,LALP,WEIGHT)

      * CONTRIBUTION OF COMPLEX FOURIER WAVES IN CFC AT ONE LATITUDE
      * ADDED TO SPECTRAL COEFF IN SC(LR,LM).
      * THIS SUBROUTINE IS CALLED ONCE FOR EACH GAUSSIAN LATITUDE.
      * BEFORE THE FIRST CALL SET ALL OF SC TO (0.,0.).

      * IF SC IS GLOBAL, ROW 1 CONTAINS WAVES 0,1,2...(LR-1).
      * IF SC IS SYMMETRIC, ROW 1 CONTAINS WAVES 0,2,4...2*(LR-1).
      * IF SC IS ANTISYMMETRIC, ROW 1 CONTAINS WAVES 1,3,5...2*(LR-1)+1.
      * ALP(LALP,LM) CONTAINS LEGENDRE POLYNOMIALS FOR ONE LATITUDE.
      *   ITS ROWS MUST HAVE THE SAME STRUCTURE AS THOSE OF SC.
      * WEIGHT IS THE GAUSSIAN WEIGHT FOR THAT LATITUDE.

      COMPLEX SC(LR,1),CFC(1)
      DIMENSION ALP(LALP,1)

      * CALCULATE (WEIGHT*CFC(M)*ALP(N,M)) FOR EACH M AND N.

      DO 20 M=1,LM
      CFCR= REAL(CFC(M))*WEIGHT
      CFCI=AIMAG(CFC(M))*WEIGHT
      DO 20 N=1,LR
      SC(N,M)=SC(N,M)+CMPLX(CFCR*ALP(N,M),CFCI*ALP(N,M))
   20 CONTINUE

      RETURN
      END
```

```
SUBROUTINE FCONW2(Z,CINT,SCALE,LI,LJ,IW,JW,LL,MM,MTYPE)

* MAPS LL BY MM WINDOW FROM POINT (IW,JW) IN GRID Z(LI,LJ)
* THE CONTINENTAL OUTLINE IS INTERPOLATED FROM A 1/20M SCALE GRID
*  OF SIZE (51,55) WITH NORTH POLE AT (26,28).
* RIGHT AND TOP GRID POINTS OMITTED FOR STAR GRIDS IF LJ NEGATIVE.
* CONTOURS FROM 0. IF CINT.GT.0, FROM -.5*CINT IF CINT.LT.0
* MTYPE = POS, 6 LINES/INCH; NEG, 8 LINES/INCH
* MTYPE = 0 EXIT WITH NO MESSAGE
* MTYPE = 1 TO 3 MTYPE = GRID DISTANCE IN INCHES
* MTYPE = 21 TO 40 ...
* MTYPE=MAPSCL+MESH. MESH CAN BE ANY VALUE FROM 1 TO 10 EXCEPT 7.
* MAPSCL=20 FOR 1/20M, 30 FOR 1/30M (ASSUMING GRID DIST OF 381KM).
* PRINTS FOUR LOWEST ORDER DIGITS OF SCALED FIELD
*       EVERY 2*MESH GRID POINTS FROM (IW,JW).
* 1/20M - PRINTED POINTS EVERY 1.5 INCHES, 16*MESH GRID SPACES PER
* 1/30M - PRINTED POINTS EVERY 1.0 INCHES, 25*MESH GRID SPACES PER
* ILLEGAL CALL GENERATES MESSAGE AND RETURN.

* MAP UNITS REFER TO A COORDINATE SYSTEM CORNERED ON (IW,JW)
* IN WHICH ONE INCH EGUALS 1440 UNITS.

 DIMENSION Z(LI,LJ)
 DIMENSION X(100),Y(100)
 INTEGER NPRLIN(130),NABCD(8),NUMBER(10)
 DIMENSION LWB(8,242),LW1(256),LW2(256),LW3(256),LW4(256),
*LW5(256),LW6(256),LW7(256),LW8(144)
 EQUIVALENCE(LWB(1,1),LW1(1)),(LWB(1,33),LW2(1)),(LWB(1,65),LW3(1))
 EQUIVALENCE (LWB(1,97),LW4(1)),(LWB(1,129),LW5(1)),(LWB(1,161),
* LW6(1)),(LWB(1,193),LW7(1)),(LWB(1,225),LW8(1))

 DATA NABCD/1H ,1HA,1H ,1HB,1H ,1HC,1H ,1HD/
 DATA NUMBER/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
 DATA NBLANK,NPLUS,NMINUS,NSTAR,NDOT/1H ,1H+,1H-,1H*,1H./
 DATA NDPC,NDPL,NPCL,INCH,INCHAH/144,240,125,1440,2160/

* ARRAY LWB CONTAINS THE CONTINENTAL OUTLINE COORDINATES FOR
* 242 PRINT LINES OF A 1/20M SCALE MAP. EACH PRINTED LINE CAN HAVE
* A MAXIMUM OF 8 POINTS, PACKED FOUR TO A WORD IN TWO SUCCESSIVE
* WORDS OF LWB. MAP SCALES OTHER THAN 1/20M ARE INTERPOLATED.

 DATA LW1/
*178,303,  0,  0,  0,  0,  0,  0,177,185,189,198,201,303,  0,  0,
*174,182,192,195,203,302,  0,  0,172,180,204,299,  0,  0,  0,  0,
*171,180,204,297,  0,  0,  0,  0,171,179,206,264,272,294,  0,  0,
*170,177,207,235,  0,266,281,290,170,177,209,230,239,257,285,  0,
*178,222,225,251,254,  0,  0,  0,167,179,246,  0,  0,  0,  0,  0,
*166,179,  0,  0,  0,  0,  0,  0,159,162,179,  0,  0,  0,  0,  0,
*157,179,  0,  0,  0,  0,  0,  0,155,179,  0,  0,  0,  0,  0,  0,
*153,174,177,  0,  0,  0,  0,  0,152,  0,168,171,  0,  0,  0,  0,
*152,165,  0,  0,  0,  0,  0,  0,149,164,  0,  0,  0,  0,  0,  0,
```

```
*  0,141,144,164,  0,  0,  0,  0,138,  0,  0,165,  0,  0,  0,  0,
*135,167,  0,  0,  0,  0,  0,  0,132,  0,152,155,  0,168,  0,  0,
*131,149,158,168,  0,  0,  0,  0,129,146,159,169,  0,  0,  0,  0,
*128,144,161,170,  0,  0,  0,  0,127,144,162,170,  0,  0,  0,  0,
*144,164,165,  0,  0,  0,  0,  0,124,143,  0,  0,  0,  0,  0,  0,
*123,143,  0,  0,  0,  0,  0,  0,122,143,  0,  0,  0,  0,  0,  0,
*120,143,  0,  0,  0,  0,  0,  0,120,143,  0,  0,  0,  0,  0,  0/
 DATA LW2/
*120,143,188,  0,  0,  0,  0,  0,122,144,185,189,  0,  0,  0,  0,
*122,144,184,189,  0,  0,  0,  0,144,189,  0,  0,  0,  0,  0,  0,
*122,145,183,188,  0,  0,  0,  0,122,146,183,188,  0,  0,  0,  0,
*122,147,182,187,  0,  0,  0,  0,121,149,179,186,  0,  0,  0,  0,
*120,150,164,176,  0,186,  0,  0,113,119,153,159,167,173,186,  0,
*111,117,187,  0,  0,  0,  0,  0,111,  0,188,  0,  0,  0,  0,  0,
*111,191,  0,  0,  0,  0,  0,  0,111,192,  0,  0,  0,  0,  0,  0,
*111,193,  0,  0,  0,  0,  0,  0,111,197,  0,  0,  0,  0,  0,  0,
*111,198,  0,  0,  0,  0,  0,  0,111,198,  0,  0,  0,  0,  0,  0,
*110,198, 0,  0,  0,  0,  0,  0,111,197,  0,  0,  0,  0,  0,  0,
*111,197,  0,  0,  0,  0,  0,  0,111,198,  0,000,  0,  0,  0,  0,
*111,199,  0,  0,  0,  0,  0,  0,111,200,  0,  0,  0,  0,  0,  0,
*111,201,  0,  0,  0,  0,  0,  0,112,203,  0,  0,  0,  0,  0,  0,
*113,204,  0,  0,  0,  0,  0,  0,113,206,  0,  0,  0,  0,  0,  0,
*113,207,  0,  0,  0,  0,  0,  0,113,207,  0,  0,  0,  0,  0,  0,
*112,209,  0,  0,  0,  0,  0,  0,111,212,215,  0,  0,  0,  0,  0/
 DATA LW3/
*111,216,  0,  0,  0,  0,  0,  0,111,219,  0,  0,  0,  0,  0,  0,
*111,219,  0,  0,  0,  0,  0,  0,111,215,218,  0,  0,  0,  0,  0,
*112,212,  0,  0,  0,  0,  0,  0,113,210,  0,  0,  0,  0,  0,  0,
*113,210,  0,  0,  0,  0,  0,  0,189,191,210,356,359,  0,  0,  0,
*113,188,192,213,  0,349,362,368,113,187,192,  0,216,348,362,368,
*114,186,192,219,347,371,377,  0,115,186,191,344,  0,  0,  0,  0,
*117,186,191,341,  0,  0,  0,  0,119,183,192,218,332,335,338,  0,
*121,180,192,215,330,  0,  0,  0,122,179,192,214,329,  0,  0,  0,
*177,192,326,  0,  0,  0,  0,  0,126,174,191,  0,210,326,  0,  0,
*126,174,191,203,210,326,  0,  0,127,173,191,201,207,324,  0,  0,
*128,174,192,200,323,  0,  0,  0,128,174,192,200,322,  0,  0,  0,
*129,176,197,321,  0,  0,  0,  0,130,178,320,  0,  0,  0,  0,  0,
*131,179,320,  0,  0,  0,  0,  0,131,180,204,  0,  0,  0,  0,  0,
*132,183,220,224,319,  0,  0,  0,132,184,218,  0,225,318,  0,  0,
*132,186,216,225,317,  0,  0,  0, 29, 31,133,  0,184,215,224,316,
* 29, 30,134,177,183,215,224,315, 29, 32,135,173,180,213,224,311/
 DATA LW4/
* 33,135,  0,170,211,223,309,  0,  0,135,166,  0,210,222,309,  0,
*164,209,223,294,297,  0,309,  0,135,164,207,224,  0,291,300,308,
*135,164,206,224,288,  0,300,308,135,162,204,224,285,302,305,  0,
*134,161,203,224,285,303,  0,  0,134,161,203,224,285,303,  0,  0,
*134,159,197,201.223,286,303,  0,134,156,194,222,287,303,  0,  0,
*134,155,194,221,287,303,  0,  0,133,193,221,  0,  0,  0,  0,  0,
*132,153,194,219,  0,285,299,306,130,153,194,218,279,  0,298,306,
*123,128,152,195,216,276,296,305,120,152,197,213,275,293,305,  0,
*120,152,197,212,275,  0,305,  0,125,198,209,273,291,306,  0,  0,
```

```
*117,126,152,201,207,272,290,306,117,126,152,203,207,272,290,306,
*129,  0,150,206,269,  0,  0,  0,132,147,  0,  0,  0,251,290,306,
*  0,138,143,  0,248,254,290,306,140,245,255,261,  0,267,288,305,
*  0,242,254,260,288,305,  0,  0,140,242,254,260,288,305,312,  0,
*138,143,234,240,257,291,308,  0,139,146,231,294,315,  0,  0,  0,
*140,147,230,294,315,  0,  0,  0,149,296,  0,  0,  0,  0,  0,  0,
*  0,138,150,228,297,317,  0,  0,135,  0,150,228,297,317,  0,  0/
 DATA LW5/
*134,150,227,299,318,  0,  0,  0,134,150,228,320,  0,  0,  0,  0,
*134,150,228,296,320,  0,  0,  0,132,152,228,294,320,  0,  0,  0,
*131,154,228,  0,321,  0,  0,  0,131,155,228,297,321,  0,  0,  0,
*132,159,300,  0,  0,  0,  0,  0,134,161,  0,230,  0,303,314,  0,
*134,163,188,230,296,303,314,  0,133,166,186,191,227,293,300,314,
*131,167,183,194,223,293,314,  0,131,167,182,195,221,293,314,  0,
*128,170,176,198,210,215,219,296,125,136,  0,180,206,212,  0,314,
*124,134,138,  0,203,  0,299,314,132,300,314,  0,  0,  0,  0,  0,
*123,129,139,300,314,  0,  0,  0,123,127,140,300,314,  0,  0,  0,
*122,125,141,314,  0,  0,  0,  0,122,142,299,312,  0,  0,  0,  0,
*  0,143,299,312,  0,  0,  0,  0,144,299,311,  0,  0,  0,  0,  0,
*145,296,312,  0,  0,  0,  0,  0,146,296,312,  0,  0,  0,  0,  0,
*146,299,311,  0,  0,  0,  0,  0,146,302,308,  0,  0,  0,  0,  0,
*146,  0,  0,  0,  0,  0,  0,  0,141,146,  0,  0,  0,  0,  0,  0,
*138,144,317,320,  0,  0,  0,  0,137,  0,317,321,  0,  0,  0,  0,
*135,317,323,  0,  0,  0,  0,  0,134,325,  0,  0,  0,  0,  0,  0/
 DATA LW6/
*134,318,326,  0,  0,  0,  0,  0,134,328,  0,  0,  0,  0,  0,  0,
*133,322,  0,  0,  0,  0,  0,  0,132,323,329,  0,  0,  0,  0,  0,
*  0,  0,132,323,330,  0,  0,  0,  0,  0,131,324,  0,  0,  0,  0,
*  0,  0,131,325,333,  0,  0,  0,  0,  0,131,326,336,  0,  0,  0,
*131,327,  0,  0,  0,  0,  0,  0,119,131,328,338,  0,  0,  0,  0,
*116,120,132,329,  0,  0,  0,  0,114,119,134,  0,  0,339,  0,  0,
*114,118,135,294,330,340,  0,  0,113,117,136,293,297,332,  0,  0,
*111,117,137,293,299,  0,341,  0,111,117,137,293,299,333,342,  0,
*113,119,137,293,299,334,342,  0,114,120,138,293,300,  0,343,  0,
*114,120,138,293,300,335,  0,  0,116,120,138,299,344,  0,  0,  0,
*119,135,  0,  0,291,300,339,344,  0,135,144,150,291,300,340,344,
*122,132,141,147,153,290,299,  0,131,138,155,287,294,  0,  0,  0,
*131,137,155,287,294,342,345,  0,134,135,153,287,291,339,347,  0,
*151,287,291,336,348,  0,  0,  0,150,287,291,335,348,  0,  0,  0,
*149,285,  0,  0,  0,  0,  0,  0,149,282,289,  0,329,345,  0,  0,
*148,281,290,326,344,  0,  0,  0,146,279,291,320,323,342,  0,  0/
 DATA LW7/
*144,276,293,  0,  0,317,339,  0,143,275,  0,296,314,  0,339,  0,
*142,273,299,302,305,308,311,336,141,271,333,  0,  0,  0,  0,  0,
*140,270,332,  0,  0,  0,  0,  0,138,270,  0,  0,  0,  0,  0,  0,
*138,269,336,  0,374,377,  0,  0,138,269,  0,342,374,377,  0,  0,
*139,269,350,356,362,368,  0,  0,140,269,  0,  0,  0,  0,  0,  0,
*140,269,  0,  0,  0,  0,  0,  0,140,269,  0,  0,  0,  0,  0,  0,
*141,267,  0,  0,  0,  0,  0,  0,141,265,  0,  0,  0,  0,  0,  0,
*143,264,  0,  0,  0,  0,  0,  0,144,263,  0,  0,  0,  0,  0,  0,
*144,263,  0,  0,  0,  0,  0,  0,146,264,  0,  0,  0,  0,  0,  0,
```

```
     *149,  0,266,  0,  0,  0,  0,  0,  0,152,266,  0,  0,  0,  0,  0,
     *153,213,267,  0,  0,  0,  0,  0,155,  0,212,216,219,  0,267,  0,
     *156,165,212,  0,  0,222,267,  0,159,162,210,225,267,  0,  0,  0,
     *161,170,209,230,  0,267,  0,  0,  0,170,209,  0,233,267,  0,  0,
     *171,209,236,239,267,  0,  0,  0,173,207,  0,245,267,  0,  0,  0,
     *173,207,242,  0,267,  0,  0,  0,171,206,248,267,  0,  0,  0,  0,
     *170,206,249,267,  0,  0,  0,  0,169,206,249,267,  0,  0,  0,  0/
      DATA LW8/
     *165,206,251,267,  0,  0,  0,  0,163,197,201,204,252,267,  0,  0,
     *162,197,  0,  0,252,267,  0,  0,161,197,254,267,  0,  0,  0,  0,
     *161,197,255,267,  0,  0,  0,  0,161,197,256,267,  0,  0,  0,  0,
     *160,195,258,267,  0,  0,  0,  0,159,  0,  0,194,261,  0,267,  0,
     *159,186,188,194,  0,263,267,  0,160,180,189,194,266,  0,  0,  0,
     *161,179,191,194,  0,  0,  0,  0,162,178,191,194,  0,  0,  0,  0,
     *164,177,191,194,  0,  0,  0,  0,166,176,192,195,  0,  0,  0,  0,
     *168,175,192,195,  0,  0,  0,  0,170,173,192,195,  0,  0,  0,  0,
     *171,191,195,  0,  0,  0,  0,  0,  0,191,195,  0,  0,  0,  0,  0/

      * DETERMINE THE MAP SCALE AND THE GRID POINTS TO BE PRINTED.
      * NDGP = DISTANCE BETWEEN GRID POINTS IN MAP UNITS.
      * NWST = MAXIMUM WIDTH OF A MAP STRIP IN MAP UNITS.
      * NPRINT = DISTANCE BETWEEN PRINTED GRID POINTS IN MAP UNITS.

      MT=IABS(MTYPE)
      IF(MT.EQ.0) RETURN
      IF(MT.GT.40) GO TO 98
      IF(MTYPE.LT.0) NDPL=180
      IF(MT.LE.30) GO TO 11
      MESH=MT-30
      NWST=125*NDPC
      NDGP=720/MESH
      NPRINT=INCH
      GO TO 13
   11 IF(MT.LT.20) GO TO 12
      MESH=MT-20
      NWST=120*NDPC
      NDGP=1080/MESH
      NPRINT=INCHAH
      GO TO 13
   12 MESH=-1
      NDGP=MT*INCH
      NWST=((125*NDPC)/NDGP)*NDGP
      NPRINT=NDGP
   13 NDGPH=NDGP/2
      GDIST=NDGP/FLOAT(INCH)
      IPDIST=NPRINT/FLOAT(NDGP)
      FNDGPH=FLOAT(NDGPH)
      FNDGP=FLOAT(NDGP)
      FNDPC=FLOAT(NDPC)
      FNDPL=FLOAT(NDPL)
```

```
MAPSCL=(MT-MESH)*MESH
SCL20=20./FLOAT(MAPSCL)

* DEFINE THE WINDOW TO BE MAPPED.
* ICMIN,ICMAX = LEFT AND RIGHT SIDES OF WINDOW IN MAP UNITS.
* JCMIN,JCMAX = BOTTOM AND TOP EDGES OF WINDOW IN MAP UNITS.
* JCNJ = TOP OF THE GRID IN MAP UNITS.

NI=LI
IF(LJ.LT.0) NI=LI-1
NJ=IABS(LJ)
IF(LJ.LT.0) NJ=NJ-1
L=LL
IF(IW+L.GT.NI) L=NI-IW
M=MM
IF(JW+M.GT.NJ) M=NJ-JW
IF(NI.LT.4.OR.NJ.LT.4) GO TO 98
IF( L.LT.3.OR. M.LT.3) GO TO 98
IF(IW.LT.1.OR.JW.LT.1) GO TO 98
ICMIN=0
ICMAX=NDGP*L
ICMAX=ICMAX-MOD(ICMAX,NDPC)
ICIW=NDGP*(IW-1)
JCMIN=0
JCMAX=NDGP*M
JCMAX=JCMAX-MOD(JCMAX,NDPL)
JCNJ=NDGP*(NJ-JW)
JCJW=NDGP*(JW-1)

* CORRECTIONS FOR STAR GRID AND FOR NON-STANDARD POLE.

STCORX=0.
IF(LJ.LT.0) STCORX=FNDGPH/FNDPC
STCORY=0.
IF(LJ.LT.0) STCORY=FNDGPH/FNDPL
FIP=51./FLOAT(LI-1)*FLOAT(IW-1)
FJP=55./FLOAT(LJ-1)*FLOAT(JW-1)
PCORX=FIP*FNDGP/FNDPC
PCORY=FJP*FNDGP/FNDPL

* CALCULATE INTERPOLATION CONSTANTS.

IF(CINT.EQ.0.) GO TO 98
CSHIFT=0.0
IF(CINT.LT.0.) CSHIFT=0.5
CSC=ABS(SCALE/CINT)
CSC2=CSC/2.
CSC6=CSC/6.
SIXTH=1./6.
```

```
* MAP STRIP LOOP. RETURN IF LAST STRIP IS FINISHED.
* IPCL,IPCR = LEFT AND RIGHT SIDES OF MAP STRIP IN MAP UNITS.
* IPNTL = POSITION OF FIRST PRINTED GRID POINT IN MAP UNITS.

   IPCR=ICMIN
15 IPCL=IPCR
   IF(IPCL.GE.ICMAX) GO TO 16
   IPCR=IPCL+NWST
   IF(IPCR.GT.ICMAX) IPCR=ICMAX
   NCHMAX=(IPCR-IPCL)/NDPC
   FPCL=FLOAT(IPCL+ICIW)/FNDPC
   IPNTL=IPCL
   IPNTC=MOD(IPCL,NPRINT)
   IF(IPNTC.NE.0) IPNTL=IPNTL+NPRINT-IPNTC
   WRITE(6,610) CINT,SCALE,GDIST,IPDIST,LI,LJ,IW,JW,LL,MM,MT

* START OF PRINT LINE LOOP FOR EACH MAP STRIP.
* J IDENTIFIES THE LOWER SIDE OF THE INNER INTERPOLATION SQUARE.
* Q = DISTANCE OF THE CURRENT ROW FROM ROW J IN GRID UNITS.
* JPL = POSITION OF CURRENT PRINT LINE IN MAP UNITS.

   JPL=JCMAX
21 J=JPL/NDGP+JW
   IF(J.GT.NJ-2) J=NJ-2
   IF(J.LT.2)    J=2
   JDIF=JPL-(J-JW)*NDGP
   Q=FLOAT(JDIF)/FNDGP
   GA=CSC6*(-Q*(Q-1.)*(Q-2.))
   GB=CSC2*(   (Q-1.)*(Q+1.)*(Q-2.))
   GC=CSC2*(-Q*(Q+1.)*(Q-2.))
   GD=CSC6*( Q*(Q+1.)*(Q-1.))

* START OF PRINT CHARACTER LOOP FOR EACH PRINT LINE.
* I IDENTIFIES THE LEFT SIDE OF THE INNER INTERPOLATION SQUARE.
* P = DISTANCE OF THE CURRENT PRINT CHARACTER FROM I IN MAP UNITS.

   ILAST=-1
   NCH=0
   DO 40 IPC=IPCL,IPCR,NDPC
   NCH=NCH+1
   I=IPC/NDGP+IW
   IF(I.GT.NI-2) I=NI-2
   IF(I.LT.2)    I=2
   IDIF=IPC-(I-IW)*NDGP
   P=FLOAT(IDIF)/FNDGP
   IF( I.EQ.ILAST) GO TO 39
   IF(ILAST.GT.0 ) GO TO 38
   ZB=GA*Z(I-1,J-1)+GB*Z(I-1,J)+GC*Z(I-1,J+1)+GD*Z(I-1,J+2)
   ZC=GA*Z(I  ,J-1)+GB*Z(I  ,J)+GC*Z(I  ,J+1)+GD*Z(I  ,J+2)
   ZD=GA*Z(I+1,J-1)+GB*Z(I+1,J)+GC*Z(I+1,J+1)+GD*Z(I+1,J+2)
```

```
38 ZA=ZB
   ZB=ZC
   ZC=ZD
   ZD=QA*Z(I+2,J-1)+QB*Z(I+2,J)+QC*Z(I+2,J+1)+QD*Z(I+2,J+2)
   ILAST=I
39 CONTINUE
   PA=SIXTH*(-P*(P-1.)*(P-2.))
   PB=  0.5*(    (P-1.)*(P+1.)*(P-2.))
   PC=  0.5*(-P*(P+1.)*(P-2.))
   PD=SIXTH*( P*(P+1.)*(P-1.))
   CONT=PA*ZA+PB*ZB+PC*ZC+PD*ZD + CSHIFT
   NCONT=INT(CONT)
   NCONT=MOD(NCONT,8)+1
   IF(CONT.LT.0.) NCONT=7+NCONT
   NPRLIN(NCH)=NABCD(NCONT)
40 CONTINUE

   * DRAW LAND-WATER BOUNDARIES.
   * NJROW IS THE CURRENT ROW IMAGE IN THE 1/20M SCALE MAP.

   NJROW=(FLOAT(JPL+JCJW)/FNDPL + STCORY + PCORY)/SCL20 + 0.5
   IF(NJROW.LT.1.OR.NJROW.GT.242) GO TO 49
   DO 44 K=1,8
   LWU=LWB(K,NJROW)
   IF(LWU.EQ.0) GO TO 44
   NCB=FLOAT(LWU)*SCL20-FPCL-STCORX-PCORX+0.5
   IF(NCB.GT.0.AND.NCB.LE.NCHMAX) NPRLIN(NCB)=NDOT
44 CONTINUE

   * INSERT GRID POINT VALUES IF THIS IS A GRID ROW TO BE PRINTED.
   * FLOATING GRID VALUES ARE SCALED, ROUNDED AND CONVERTED TO INTEGE
   * SIGN AND LOWEST FOUR DIGITS ARE INSERTED INTO NPRLIN.

49 NCHP=NCH+1
   DO 51 NN=NCHP,130
51 NPRLIN(NN)=NBLANK
   IF(JPL/NPRINT*NPRINT.NE.JPL) GO TO 57
   K=J
   IF(JPL.EQ.JCNJ ) K=NJ
   IF(JPL.EQ.JCNJ-NDGP) K=NJ-1
   IF(JPL.EQ.0)      K=JW
   NPRLIN(1)=NSTAR
   NPRLIN(NCH)=NSTAR
   DO 55 IPNT=IPNTL,IPCR,NPRINT
   I=IPNT/NDGP+IW
   NCH=(IPNT-IPCL)/NDPC+6
   P=Z(I,K)*SCALE
   IG=INT(ABS(P)+0.5)
   NPRLIN(NCH-5)=NPLUS
   IF(P.LT.0.) NPRLIN(NCH-5)=NMINUS
   DO 54 NN=1,4
```

```
      IQ10=IQ/10
      NPRLIN(NCH-NN)=NUMBER(IQ-IQ10*10+1)
   54 IQ=IQ10
   55 CONTINUE

      * PRINT ONE LINE. IF NOT END OF STRIP GO BACK TO PRINT LINE LOOP
      *  AT STATEMENT 21. OTHERWISE START NEW STRIP AT STATEMENT 15.

   57 WRITE(6,620) NPRLIN
      JPL=JPL-NDPL
      IF(JPL.GE.JCMIN) GO TO 21
      GO TO 15
   16 DO 17 IX=1,LI
   17 X(IX)=FLOAT(IX)
      DO 18 IY=1,LJ
   18 Y(IY)=FLOAT(IY)
      CALL CONTOR(Z,X,Y,LI,LJ,0,10,IW,LL,JW,MM,' $',' $',FLOAT(MTYPE),
     *0.,0.,0.)
      RETURN

   98 WRITE(6,698) CINT,SCALE,LI,LJ,IW,JW,LL,MM,MTYPE
      RETURN

  610 FORMAT(1H1,5X,8HCONTOLR=,1PE11.4,5X,6HSCALE=,E11.4,5X,5HGRID=,
     *0PF7.4,21H INCHES   PRINT EVERY,I3,5X,6I4,I6//)
  620 FORMAT(1H ,130A1)
  698 FORMAT(22H ILLEGAL CALL TO PCONW,1P2E14.4,9I6)
      END
```

```
      SUBROUTINE FFGFW2(GR,LGR,FW,LFW,IR,ILONG,WRKS,NL)

      * CONVERTS IR COMPLEX FOURIER COEFFICIENTS IN FW(LFW,NL)
      *  TO ILONG EQUALLY SPACED GRID POINT VALUES IN GR(LGR,NL)
      *  BY THE FAST FOURIER SUBROUTINE FOR12S.

      COMPLEX FW(LFW,NL),WRKS(1)
      DIMENSION GR(LGR,NL)
      COMPLEX AA,BB,CC
      DIMENSION NY(3)

      IRP1=IR+1
      IF(MOD(ILONG,3).EQ.0) GO TO 100

      * CASE 1 - ILONG IS A POWER OF 2.
      * MAX IR IS ILONG/2.  WRKS=(ILONG+1) COMPLEX WORDS.

      IF(IR.GT.ILONG/2) WRITE(6,6010) IR,ILONG
      IF(IR.GT.ILONG/2) RETURN
      ILG2=ILONG+2
      NWMAX=ILONG/2  + 1
      WRKS(NWMAX+1)=(0.,0.)

      DO 50 L=1,NL

      * COPY WAVES 0 TO IR TO WRKS. FILL TO WAVE ILONG/2 WITH ZEROS.

      DO 20 J=IRP1,NWMAX
   20 WRKS(J)=(0.,0.)
      DO 30 J=1,IRP1
   30 WRKS(J)=FW(J,L)

      * PERFORM THE FAST FOURIER TRANSFORM.

      N=ILONG
      ISIGN=1
      NTYP=-1
      CALL FOUR2(WRKS,N,1,ISIGN,NTYP)

      * COPY ILONG GRID POINTS FROM WRKS TO GR.

      NX=0
      DO 40 I=1,ILONG,2
      NX=NX+1
      GR(I  ,L)= REAL(WRKS(NX))
      GR(I+1,L)=AIMAG(WRKS(NX))
   40 CONTINUE
   50 CONTINUE
      RETURN

      * CASE 2 - ILONG IS 3 TIMES A POWER OF 2.
```

```
      * MAX VALUE OF IR IS ILONG/3.  WRKS=(ILONG/6+1)*8 COMPLEX WORDS.

  100 K3 = ILONG/3
      IF(IR.GT.K3) WRITE(6,6010) IR,ILONG
      IF(IR.GT.K3) RETURN
      K6 = K3/2
      KP6 = K6 + 1
      NWMAX = 2*KP6
      WRKS(7*KP6+1) = (0.,0.)

      CALL PERM(WRKS(2*KP6+1),WRKS(3*KP6+1),AA,BB,KP6,K3)

      DO 150 L=1,NL

      DO 120 J=IRP1,NWMAX
  120 WRKS(J) = (0.,0.)
      DO 130 J=1,IRP1
  130 WRKS(J) = FW(J,L)

      DO 160 K=1,KP6
      CC = CONJG(WRKS(K3+2-K))
      WRKS(4*KP6+K) = WRKS(K) + CC
      WRKS(5*KP6+K) = (WRKS(K)+BB*CC)*WRKS(2*KP6+K)
  160 WRKS(6*KP6+K) = (WRKS(K) + AA*CC)*WRKS(3*KP6+K)

      N=K3
      ISIGN = 1
      NTYP = -1
      NY(1)=K3
      NY(2)=K3
      NY(3)=K3
      CALL FOUR2(WRKS(4*KP6+1),NY,3,ISIGN,NTYP)
      WRKS(7*KP6) = WRKS(6*KP6+1)

      DO 170 K=1,K6
      KP = (K-1)*6
      GR(KP+1,L)= REAL (WRKS(4*KP6+K))
      GR(KP+2,L) =REAL (WRKS(5*KP6+K))
      GR(KP+3,L)= AIMAG(WRKS(6*KP6+K))
      GR(KP+4,L)= AIMAG(WRKS(4*KP6+K))
      GR(KP+5,L)= AIMAG(WRKS(5*KP6+K))
  170 GR(KP+6,L)= REAL(WRKS(6*KP6+K+1))

  150 CONTINUE
      RETURN

 6010 FORMAT(34H ILLEGAL CALL TO FFGFW2..IR,ILONG=,2I8)
      END
```

```
      SUBROUTINE FFWFG2(FW,LFW,GR,LGR,IR,ILONG,WRKS,NL)

      * CONVERTS ILONG EQUALLY SPACED GRID POINT VALUES IN EACH LEVEL OF
      *  GR(LGR,NL) TO COMPLEX FOURIER COEFFICIENTS IN FW(LFW,NL)
      *  TO WAVE NUMBER IR, USING THE FAST FOURIER ROUTINE FOR12S.

      COMPLEX FW(LFW,NL)
      DIMENSION GR(LGR,NL),WRKS(1)
      COMPLEX AA,BB
      DIMENSION NY(3)

      IRP1=IR+1
      IF(MOD(ILONG,3).EQ.0) GO TO 100

      * CASE 1 - ILONG IS A POWER OF 2.

      IF(IR.GT.ILONG/2) WRITE(6,6010) IR,ILONG
      IF(IR.GT.ILONG/2) RETURN
      ILG2=ILONG+2
      FLINV=1./FLOAT(ILONG)
      WRKS(ILG2+1)=0.

      DO 50 L=1,NL

      * TRANSFER GR TO WRKS AND DIVIDE BY (2*ILONG).

      DO 20 I=1,ILONG
   20 WRKS(I)=GR(I,L)*FLINV

      * PERFORM THE FAST FOURIER TRANSFORM.

      N=ILONG
      ISIGN=-1
      NTYP=0
      CALL FOUR2(WRKS,N,1,ISIGN,NTYP)

      * TRANSFER COMPLEX WAVES 0 TO IR FROM WRKS TO FW.

      NX=-1
      DO 40 J=1,IRP1
      NX=NX+2
   40 FW(J,L)=CMPLX(WRKS(NX  ),WRKS(NX+1  ))
   50 CONTINUE
      RETURN

      * CASE 2 - ILONG IS 3 TIMES A POWER OF 2.
      * MAX VALUE OF IR IS ILONG/3.  WRKS=(ILONG/6+1)*8 COMPLEX WORDS.

  100 FLINV = .5/FLOAT(ILONG)
      K3 = ILONG/3
      IF(IR.GT.K3) WRITE(6,6010) IR,ILONG
```

```
      IF(IR.GT.K3) RETURN
      K6 = K3/2
      KP6 = K6 + 1
      WRKS(14*KP6+1) = 0.

      CALL PERM(WRKS(4*KP6+1),WRKS(6*KP6+1),AA,BB,KP6,K3)

      DO 150 L=1,NL

      DO 130 K=1,K3
      KP = (K-1)*3
      WRKS(8*KP6+K) = GR(KP+1,L)*FLINV
      WRKS(10*KP6+K) = GR(KP+2,L)*FLINV
  130 WRKS(12*KP6+K+1) = GR(KP+3,L)*FLINV
      WRKS(12*KP6+1) = WRKS(12*KP6+K3+1)

      N=K3
      ISIGN = -1
      NTYP = 0
      NY(1)=K3
      NY(2)=K3
      NY(3)=K3
      CALL FOUR2(WRKS(8*KP6+1),NY,3,ISIGN,NTYP)

      CALL RCOM(WRKS(4*KP6+1),WRKS(6*KP6+1),AA,BB,WRKS(8*KP6+1),
     1   WRKS(10*KP6+1),WRKS(12*KP6+1),WRKS(1),KP6,K3)

      NX = -1
      DO 160 J=1,IRP1
      NX = NX+2
  160 FW(J,L) = CMPLX(WRKS(NX),WRKS(NX+1))

  150 CONTINUE
      RETURN

 6010 FORMAT(34H ILLEGAL CALL TO FFWFG2..IR,ILONG=,2I8)
      END
```

FAST FOURIER TRANSFORM TO REPLACE "FOUR2" AND "FOUR12S" IN THE GLOBEX MODEL. SHOULD BE TRANSPARENT TO THE USER.

```
      SUBROUTINE FOUR2(F,N,T,IS,IT)

      COMMON/FFTDAT/T1(400),T2(400)
      REAL D(2,200),E(1),F(2,1)
      EQUIVALENCE (D(1,1),E(1))
      DATA IC,IH/0,0/

      CHECK INPUT DATA - VERY CURSORY

      IF((N/2)*2.NE.N) STOP 69
      IF(IT.LT.-1.OR.IT.GT.0) STOP 70

      IF(IH.NE.IS.OR.IC.EQ.0) CALL FOURIN(IS,N)
      IH=IS
      IC=1
      ISW=IT+2
      IN2=N/2
      IN2M1=IN2-1
      IN2P1=IN2+1
      GO TO (1,2),ISW

      CHOOSE THE TYPE OF TRANSFORM

         1 => 1/2 COMPLEX ARRAY TO A REAL ARRAY
         2 => A REAL ARRAY TO 1/2 COMPLEX ARRAY

    1 DO 3 I=1,IN2P1
      D(1,I)=F(1,I)
    3 D(2,I)=F(2,I)
      DO 11 I=1,IN2M1
      D(1,N-I+1)=F(1,I+1)
   11 D(2,N-I+1)=-F(2,I+1)
      GO TO 4
    2 DO 5 I=1,IN2
      DO 5 J=1,2
    5 D(J,I)=F(J,I)
      DO 6 I=1,N
      IM=N-I+1
      D(1,IM)=E(IM)
    6 D(2,IM)=0.0
    4 CALL FOURTR(D,D,T2,N,IFIX(ALOG(FLOAT(N))/ALOG(2.)+0.5))
      GO TO (7,10),ISW

      PREPARE THE OUTPUT ARRAY

    7 DO 8 I=1,N
    8 E(I)=D(1,I)
      DO 12 I=1,IN2
      DO 12 J=1,2
   12 F(J,I)=D(J,I)
      RETURN
```

```
10 DO 9 I=1,IN2P1
   DO 9 J=1,2
 9 F(J,I)=D(J,I)
   RETURN
   END
```

```
      SUBROUTINE FPAK
C
C     DUMMY OUT THE PACKING ROUTINES AND TRAP SPURIOUS CALLS
C
      ENTRY MOVLEV
      WRITE(6,1)
    1 FORMAT('0 PACKING NOT ALLOWED ON ASC VERSION OF GLOBAL MODEL')
      RETURN
      END
```

```
      FUNCTION GAMSAT(T,QS)

* COMPUTES  *GAMSAT*  WHICH WHEN MULTIPLIED BY  CAPA/SIGMA
* GIVES THE MOIST ADIABATIC LAPSE RATE OF POTENTIAL TEMPERATURE

      COMMON/GAMS/EPSS,CAPA

      HT=HTVOCP(T)
       A=HT*QS/(CAPA*T)
      AB=A*EPSS*HT/T

      GAMSAT=T*(AB-A)/(1.+AB)


      RETURN
      END
```

```
      SUBROUTINE GARPIN(NF1,NFMTN,NF2,NDAY,LV,PR,ILG,ILAT,
     1                  MTN,WET,KDR,KWT,KCV, IW,JW,LL,MM,
     2                  MPGZ,MPT,MPW,MPWSP,MPGZS,MPSP,MPCR,MPWT,MPCV)

 * THIS PROGRAM GETS SPECTRAL DATA FOR ONE DAY (NDAY) FROM THE
 *   GARP TAPE (NF1) AND CONVERTS TO GAUSSIAN GRIDS FOR OUTPUT(NF2).

 * ALL SPECTRAL FIELDS ON THE TAPE ARE DIMENSIONED (27,26),
 * BUT THEY ARE TRUNCATED AT WAVE NUMBER 25 SO THE LAST COMPLEX
 * NUMBER IN EACH ROW IS UNUSED. SUBROUTINE SMOV2 IS USED TO
 * CONVERT THESE FIELDS TO THE STANDARD (26,26) SIZE.
 * NOTE - WINDS U,V HAVE SIZE (27,26) IN THE PROGRAM.

 * GEOPOTENTIALS (GZ) ARE COMPUTED FROM TEMPERATURES (TGG)
 *   AND SURFACE PRESSURE (PSGG).  GZ AND TGG ARE STORED ON NF2.
 * WIND COMPONENTS (U,V) ARE COMPUTED FROM VORTICITY (Q) AND
 *   DIVERGENCE (D) AND SAVED ON NF2).
 * NOTE - ACTUAL WINDS COMPUTED ARE (U,V)*COS(LAT)/(EARTH RADIUS)

 * THE DAYS ON THE TAPE ARE NOV 4,5,6,7,8,9 (1969).
 * T,Q,D ARE AVAILABLE ON 10 PRESSURE LEVELS.
 * THE ORDER OF THE PRESSURE LEVEL DATA ON THE TAPE IS
 *       50,150,250,350,450,550,650,750,850,950 MB.

      PARAMETER $ILV=15,$LEV=0,$LV=10,$ILT=52,$ILG=64,$ILTH=26,$IR=20
      LOGICAL MTN
      LOGICAL WET,KDR,KWT,KCV
      DIMENSION PR(LV)
      DIMENSION MPT(1),MPGZ(1),MPW(1),MPWSP(1)

      COMPLEX Q,D,U,V
      COMMON/LCM1/ Q(26,26),D(26,26),U(27,26),V(27,26)

      COMMON/LCM2/ GG(3380),GGP(2000),WRKL(130)

      COMMON/LCM3/PSGG(3380),GZS(3380),TGG(3380,11),DUMMY(3380)
      DIMENSION GZ(3380,11)
      EQUIVALENCE(GZ(1,1),TGG(1,2))

      COMMON/ALPCOM/ ALP(28,26),EPSI(28,26)
      COMMON/GAUSS/W($ILTH),WOCS($ILTH),COA($ILTH),SIA($ILTH),RAD($ILTH)
     *,ANG($ILTH)
      DIMENSION WRKS(300,2),PRL(10)

 * OUTPUT PACKING DENSITIES. NO PACKING IF SET TO 1.

      DATA      NPGZS,NPGZ,NPT,NPW,NPES,NPWT,NPCV,NPCR
     1          /   1,   1,  1,  1,   1,   1,   1,   1 /

 * INITIAL FIELDS ARE GLOBAL AND CONTAIN TO WAVE 25.
 * ILALP = ROW LENGTH OF ALP,EPSI (MUST BE EVEN AND GT IIRP2).
```

```
      ILALP=28
      IKHEM=0
      IKSYM=0
      IIR=25
      IIRP1=IIR+1
      IIRP2=IIR+2

*     SET CONSTANTS.

      A = 6.37122E6
      GRAV=9.80616
      RGAS=287.
      RGOCP=2./7.
      ILATH=ILAT/2
      ILG1=ILG+1
      LGG=ILG1*ILAT
      CALL EPSIL2(EPSI,ILALP,IIRP1)
      CALL GAUSSG(ILATH,COA,W,SIA,RAD,WOCS)

*     PRL CONTAINS THE LOG OF SUCCESSIVE PRESSURE RATIOS
*      FOR LATER USE IN THE HYDROSTATIC EQUATION INTEGRAL.

      DO 15 L=1,9
   15 PRL(L)=ALOG(PR(L+1)/PR(L))
      PRL(10)=ALOG(1000./PR(10))

      ID=0
      REWIND NF1

*     IF MTNS=.F. JUST SET MOUNTAIN FIELD TO ZERO. OTHERWISE READ
*     SPECTRAL MOUNTAINS FROM NFMTN ONE ROW AT A TIME AND CONVERT
*     TO GAUSSIAN GRID. CONVERT FROM METERS TO GZ AND SAVE ON NF2.

*     NOTE - MOUNTAINS HAVE SIZE (26,26) ON THE FILE.
*     NOTE - MOUNTAINS ARE NOT ON THE GARP TAPE. USER MUST SUPPLY.

      DO 20 I=1,LGG
   20 GZS(I)=0.
      IF(.NOT.MTN) GO TO 23
      DO 21 M=1,IIRP1
      READ(5,5020) (Q(N,M),N=1,IIRP1)
   21 CONTINUE
      CALL SPAGG2(GZS ,ILG1,ILATH,COA,IKHEM,
     1            Q,IIRP1,IIRP1,IKSYM,ALP,EPSI,ILALP,WRKS,WRKL)
      DO 22 I=1,LGG
   22 GZS(I)=GZS(I)*GRAV
      CALL FCONW2(GZS,-500.,1./GRAV,ILG1,ILAT,IW,JW,LL,MM,MPGZS)
   23 CALL WSGGP(GZS,ILG1,ILAT,NF2,ID,4H GZS,1,NPGZS,GGP,WRKS)

*     READ SPECTRAL TEMPERATURES (DEG C). CONVERT TO GAUSSIAN GRIDS.
```

```
      DO 24 ND=4,9
      DO 24  L=1,10
      READ(NF1,5101) U
      IF(ND.EQ.NDAY) WRITE(6,6030) L,ND
      IF(ND.EQ.NDAY) CALL SMOV2(Q,IIRP1,U,IIRP2,IIRP1)
      IF(ND.EQ.NDAY) CALL SPAGG2(TGG(1,L),ILG1,ILATH,COA,IKHEM,
     1              Q,IIRP1,IIRP1,IKSYM,ALP,EPSI,ILALP,WRKS,WRKL)
   24 CONTINUE

    * CONVERT TEMPERATURES TO DEG K AND SAVE ON NF2.

      DO 32 L=1,10
      DO 31 I=1,LGG
   31 TGG(I,L)=TGG(I,L) + 273.16
      CALL FCONW2(TGG(1,L), 5.,1.,ILG1,ILAT,IW,JW,LL,MM,MPT(L))
      CALL WSGGP(TGG(1,L),ILG1,ILAT,NF2,ID,4H   T,L, NPT,GGP,WRKS)
   32 CONTINUE

    * GET SPECTRAL MSL PRESSURE (MB). CONVERT TO GAUSSIAN GRID.

      DO 36 ND=4,9
      READ(NF1,5101) U
      IF(ND.EQ.NDAY) WRITE(6,6040) ND
      IF(ND.EQ.NDAY) CALL SMOV2(Q,IIRP1,U,IIRP2,IIRP1)
      IF(ND.EQ.NDAY) CALL SPAGG2(  PSGG  ,ILG1,ILATH,COA,IKHEM,
     1              Q,IIRP1,IIRP1,IKSYM,ALP,EPSI,ILALP,WRKS,WRKL)
   36 CONTINUE

    * COMPUTE T1000 FROM T950 ASSUMING 0.6*(DRY LAPSE RATE).

      CON=(1000./950.)**(RGAS/GRAV*5.9E-3)
      DO 42 I=1,LGG
   42 TGG(I,11)=TGG(I,10)*CON
      CALL FCONW2(TGG(1,11),5.,1.,ILG1,ILAT,IW,JW,LL,MM,MPT(11))
      CALL FCONW2(PSGG,5.,1.,ILG1,ILAT,IW,JW,LL,MM,MPSP)

    * COMPUTE GZ1000 FROM P(MSL)AND T1000, ASSUMING T(MSL)=T1000.

      DO 44 I=1,LGG
   44 GZ(I,11)=RGAS*TGG(I,11)*ALOG(PSGG(I)/1000.)

    * COMPUTE THE GEOPOTENTIALS BY INTEGRATING THE HYDROSTATIC
    *  EQUATION UP FROM THE 1000 MB LEVEL.

      DO 46 KK=1,10
      L=11-KK
      DO 46 I=1,LGG
      GZ(I,L)=GZ(I,L+1) + RGAS*.5*(TGG(I,L)+TGG(I,L+1))*PRL(L)
   46 CONTINUE
```

```
      * SAVE GEOPOTENTIALS ON FILE NF2.

      SCAL=1./(10.*GRAV)
      DO 48 L=1,10
      CALL FCONW2(GZ(1,L), 6.0,SCAL,ILG1,ILAT,IW,JW,LL,MM,MPGZ(L))
      CALL WSGGP( GZ(1,L),ILG1,ILAT,NF2,ID,4H  GZ,L,NPGZ,GGP,WRKS)
   48 CONTINUE
      CALL FCONW2(GZ(1,11),6.0,SCAL,ILG1,ILAT,IW,JW,LL,MM,MPGZ(11))

      * READ SPECTRAL VORTICITY (Q) AND DIVERGENCE (D) FROM TAPE.
      * CONVERT TO WIND COMPONENTS ON GAUSSIAN GRIDS. SAVE ON FILE NF2.
      * WIND SPEED IS COMPUTED FOR DIAGNOSTIC PURPOSES ONLY.

      DO 50 ND=4,NDAY
      DO 50  L=1,10
      READ(NF1,5101) U
      READ(NF1,5101) V
      IF(ND.LT.NDAY) GO TO 50
      WRITE(6,6050) L,ND

      * COMPUTE WIND COMPONENTS FROM VORTICITY AND DIVERGENCE.

      CALL SMOV2(Q,IIRP1,U,IIRP2,IIRP1)
      CALL SMOV2(D,IIRP1,V,IIRP2,IIRP1)
      CALL QDAW2(U,V,Q,D,EPSI,WRKS(1,1),WRKS(1,2),
     1           IIRP2,IIRP2,IIRP1,IIRP1,ILALP,IIRP1,IKHEM,IIR)

      * CONVERT U TO GAUSSIAN GRID AND SAVE ON FILE NF2.

      CALL SPAGG2(GG,ILG1,ILATH,COA,IKHEM,
     1           U,IIRP2,IIRP1,+IKSYM,ALP,EPSI,ILALP,WRKS,WRKL)
      CALL FCONW2(GG,10.,A,ILG1,ILAT,IW,JW,LL,MM,MPW(L))
      CALL WSGGP(GG,ILG1,ILAT,NF2,ID,4H   U,L,NPW,GGP,WRKS)
      DO 491 I=1,LGG
  491 PSGG(I)=GG(I)

      * CONVERT V TO GAUSSIAN GRID AND SAVE ON FILE NF2.

      CALL SPAGG2(GG,ILG1,ILATH,COA,IKHEM,
     1           V,IIRP2,IIRP1,-IKSYM,ALP,EPSI,ILALP,WRKS,WRKL)
      CALL FCONW2(GG,10.,A,ILG1,ILAT,IW,JW,LL,MM,MPW(L))
      CALL WSGGP(GG,ILG1,ILAT,NF2,ID,4H   V,L,NPW,GGP,WRKS)

      * COMPUTE WIND SPEED AND MAP IF REQUESTED.

      DO 492 I=1,LGG
  492 PSGG(I)=SQRT(GG(I)**2 + PSGG(I)**2)
      CALL FCONW2(PSGG,10.,A,ILG1,ILAT,IW,JW,LL,MM,MPWSP(L))
      IF(L.EQ.10) GO TO 52
   50 CONTINUE
   52 REWIND NF1
```

```
* THE GRID PHYSICS FIELDS ARE SET TO ARBITRARY BUT REALISTIC
*  VALUES FOR THE PURPOSE OF TESTING THE SPECTRAL MODEL.

* SURFACE DRAG IS SIMPLY A FUNCTION OF THE CRESSMAN DRAG
*  COEFFICIENT AND THE MOUNTAIN HEIGHTS.

      IF(.NOT.KDR) GO TO 55
      CRESSD=1.2E-3
      DO 54 I=1,LGG
   54 GG(I)=CRESSD*(1.0 + GZS(I)/1.E+4)
      CALL FCONW2(GG,200.,1.E5,ILG1,ILAT,IW,JW,LL,MM,MPDR)
      CALL WSGGP(GG,ILG1,ILAT,NF2,ID,4H  DR,1,NPDR,GGP,WRKS)

* WATER TEMPERATURE IS A FUNCTION OF LATITUDE.

   55 IF(.NOT.KWT) GO TO 57
      DO 56 J=1,ILAT
      IF(J.LE.ILATH) WTEMP=273.+.5*FLOAT(J)
      IF(J.GT.ILATH) WTEMP=273.+.5*FLOAT(ILAT+1-J)
      DO 56 K=1,ILG1
      I=(J-1)*ILG1 + K
      GG(I)=WTEMP
   56 CONTINUE
      CALL FCONW2(GG,  3.,1.00,ILG1,ILAT,IW,JW,LL,MM,MPWT)
      CALL WSGGP(GG,ILG1,ILAT,NF2,ID,4H  WT,1,NPWT,GGP,WRKS)

* SEA COVER IS DETERMINED FROM THE MOUNTAINS AND AN ASSUMED POLAR
*  SNOW COVER OF 6 LATITUDE CIRCLES.

   57 IF(.NOT.KCV) GO TO 99
      DO 58 J=1,ILAT
      SNOW=0.
      IF(J.LE.6.OR.J.GE.(ILAT-5)) SNOW=1.
      DO 58 K=1,ILG1
      I=(J-1)*ILG1 + K
      GG(I)=1.
      IF(GZS(I).GT.999. .OR. SNOW.GT.0.) GG(I)=0.
   58 CONTINUE
      CALL FCONW2(GG,-10.,10.0,ILG1,ILAT,IW,JW,LL,MM,MPCV)
      CALL WSGGP(GG,ILG1,ILAT,NF2,ID,4H  CV,1,NPCV,GGP,WRKS)
   99 RETURN

 5020 FORMAT(10F8.3)
 5101 FORMAT(6E12.5)
 6030 FORMAT(11H TEMP LEVEL,I3,5H  NOV,I2,15H READ FROM TAPE)
 6040 FORMAT(22H SURFACE PRESSURE  NOV,I2,15H READ FROM TAPE)
 6050 FORMAT(15H VORT,DIV LEVEL,I3,5H  NOV,I2,15H READ FROM TAPE)
      END
```

```
PROGRAM GARP6

* CONTROL PROGRAM FOR GARPIN.
* READS 10 LEVEL DATA FROM THE GARP TAPE FOR DATE NDAY
* FOR INPUT TO THE MULTILEVEL SPECTRAL MODEL.

THE PARAMETER STATEMENT IS USED TO DEFINE THE MAXIMUM NUMBER OF
LEVELS OR POINTS ALLOWED FOR EACH TYPE OF MATRIX

$ILV = NUMBER OF SIGMA LEVELS (NOT COUNTING 0,1)
$LEV = MOISTURE LEVELS
$LV = PRESSURE LEVELS
$ILT = LATTITUDE POINTS IN GRID
$ILG MAXIMIM NUMBER OF GRID POINTS IN LONGITUDE CIRCLE
$ILTH = $ILT/2
$IR = MAXIMUM NUMBER OF WAVES PERMITTED
$ILP1 = $ILG + 1
$LAW = ($ILG + 1)*$ILT OR ($IR + 2)*($IR + 1)*$ILV

PARAMETER $ILV=15,$LEV=0,$LV=10,$ILT=52,$ILG=64,$ILTH=26,$IR=20

LOGICAL MTN,WET,KDR,KWT,KCV,KDIV
DIMENSION PR($LV),SIG($ILV),IDATIM(14)
DIMENSION MPGZ(11),MPT(11),MPW(10),MPWSP(10)
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

DATA LV,PR/10, 50.,150.,250.,350.,450.,550.,650.,750.,850.,950. /
DATA IDATIM/6*0,7*'   X',0/

* MAP CONTROLS (0 MEANS NO MAP).

DATA IW,JW,LL,MM/ 1,1,32,52/
DATA MPGZ/ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 /
DATA MPT / 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 /
DATA MPW / 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 /
DATA MPWSP/0, 0, 0, 0, 0, 0, 0, 0, 0, 0 /
DATA MPGZS,MPSP,MPDR,MPWT,MPCV/ 0, 0, 0, 0, 0 /

* INPUT TAPE (NF1), MOUNTAIN FILE (NFMTN), OUTPUT FILE (NF2).

DATA NF1,NFMTN,NF2/ 81,83,42 /

* SPECIFY THE DATE NDAY (NOV 4 TO 9).

CALL INDUMP
READ(5,5010) NDAY
MPT(5)=22
MPWSP(5)=22
MPSP=22
MPDR=22
```

```
      * READ MODEL PARAMETERS FROM CARDS.

      READ(5,5010) ILG,ILAT,KHEM,IR,LEVS,DEET
      READ(5,5010) IW,JW,LL,MM
      READ(5,5012) NK,(SIG(N),N=1,NK)
      READ(5,5015) MTN,WET,KDR,KWT,KCV,KDIV,SCDRL,SCDRW
      READ(5,5010) KSTART,KTOTAL,IPRG,IPCP

      * WRITE CONTROL LABEL ON FILE NF2.

      CALL SPLAB(NF2,2,LV,PR,NK,SIG,LEVS,ILG,ILAT,KHEM,IR,DEET,
     1      MTN,WET,KDR,KWT,KCV,KDIV,KSTART,KTOTAL,IPRG,IPCP,IDATIM)

      * READ THE GARP TAPE AND PUT OUTPUT ON FILE NF2.

      CALL GARPIN(NF1,NFMTN,NF2,NDAY,LV,PR,ILG,ILAT,
     1              MTN,WET,KDR,KWT,KCV, IW,JW,LL,MM,
     2             MPGZ,MPT,MPW,MPWSP,MPGZS,MPSP,MPCR,MPWT,MPCV)

 5010 FORMAT(5I4,F8.0)
 5012 FORMAT(I5,15F5.3)
 5015 FORMAT(6L4,2F8.2)
      STOP
      END
```

```
      SUBROUTINE GAUSSG(NZERO,F,WT,SIA,RAD,WOCS)

      * THIS ROUTINE CALCULATES THE ROOTS (F) OF THE ORDINARY LEGENDRE
      * POLYNOMIALS OF ORDER NZERO.  THE FIRST STEP IS TO MAKE AN
      * INITIAL GUESS FOR EACH ROOT AND THEN TO USE THE ORDINARY
      * LEGENDRE ALGORITHM (ORDLEG) AND NEWTONS METHOD TO REFINE
      * THE SOLUTION UNTIL THE CRITERION XLIM IS SATISFIED.
      *     F = COSINE OF COLATITUDE
      *    WT = CORRESPONDING GAUSSIAN WEIGHT
      *   SIA = SINE OF COLATITUDE
      *   RAD = COLATITUDE IN RADIANS
      *  WOCS = GAUSSIAN WEIGHT / COS(COLAT)**2

      DIMENSION F(1),WT(1),SIA(1),RAD(1),WOCS(1)

      XLIM=1.0E-06
      PI = 3.1415926535898
      IR = 2*NZERO
      FI=FLOAT(IR)
      FI1=FI+1.
      FN=FLOAT(NZERO)

      DO 20 I=1,NZERO
      DOT=FLOAT(I-1)
      F(I)=-PI*.5*(DOT+.5)/FN + PI*.5
      F(I) =  SIN(F(I))
   20 CONTINUE

      DN = FI/SQRT(4.*FI*FI-1.)
      DN1=FI1/SQRT(4.*FI1*FI1-1.)
      A = DN1*FI
      B = DN*FI1
      IRP = IR + 1
      IRM = IR -1

      DO 50 I=1,NZERO
   42 CALL ORDLEG(G,F(I),IR)
      CALL ORDLEG(GM,F(I),IRM)
      CALL ORDLEG(GP,F(I),IRP)
      GT = (A*GP-B*GM)/(F(I)*F(I)-1.)
      FTEMP = F(I) - G/GT
      GTEMP = F(I) - FTEMP
      F(I) = FTEMP
      IF( ABS(GTEMP).GT.XLIM) GO TO 42
   50 CONTINUE

      DO 60 I=1,NZERO
      A=2.*(1.-F(I)**2)
      CALL ORDLEG(B,F(I),IRM)
      B = B*B*FI*FI
      WT(I)=A*(FI-.5)/B
```

```
      RAD(I) =    ACOS(F(I))
      SIA(I) =  SIN(RAD(I))
      WOCS(I) =WT(I)/SIA(I)**2
   60 CONTINUE

      RETURN
      END
```

```
      SUBROUTINE GGASP2(P,LR,LM,KSM,GG,ILG1,ILATH,COA,W,KHEM,
     1                  ALP,EPSI,LALP,WRKS,WRKL)

*     SPECTRAL ANALYSIS OF GAUSSIAN GRID GG.
*     PUTS SPECTRAL COEFF INTO P(LR,LM).
*     IF GLOBAL GG=(ILG1,ILATH*2), OTHERWISE GG=(ILG1,ILATH).
*     LATITUDE IS ZERO AT LEFT OF GRID AND POSITIVE EASTWARD.

*     KHEM  0 = GLOBAL,  1 = N HEM ONLY,  2 = S HEM ONLY.
*     KSM   +1=SYMMETRIC,  0=GLOBAL,  -1=ANTISYMMETRIC
*     COA(ILATH) CONTAINS THE COSINES OF THE COLATITUDE (N TO S).
*     W(ILATH) CONTAINS THE GAUSSIAN WEIGHTS (N HEM, N TO S).
*     ALP(LALP,LM) IS A WORK FIELD FOR LEGENDRE POLYNOMIALS.
*     EPSI IS A FIELD OF CONSTANTS THE SAME SIZE AS ALP.
*     WRKS,WRKL ARE SCM,LCM WORK ARRAYS OF (ILG+2) WORDS.
*     FAST FOURIER TRANSFORM REQUIRES THAT ILONG BE A POWER OF 2.

*     WARNING - KHEM AND KSM MUST BE BOTH ZERO OR BOTH NON-ZERO.
*             - ANTISYMMETRIC FIELDS ANALYSED FROM A HEMISPHERE
*               WILL ALWAYS HAVE A MEAN OF ZERO.

      COMPLEX P(1)
      DIMENSION GG(ILG1,1),ALP(1),EPSI(1),WRKL(1)
      DIMENSION COA(1),W(1),WRKS(1)

*     CALCULATE CONSTANTS AND SET P TO ZERO.

      ILG=ILG1-1
      ILGH=ILG/2
      MAXF=LM-1
      ILP=1
      IF(KSM.LT.0) ILP=LALP/2+1
      NP=LR*LM
      DO 10 MN=1,NP
   10 P(MN)=(0.,0.)

*     DO NORTHERN HEMISPHERE IF REGUESTED (KHEM=0 OR 1).

      IF(KHEM.EQ.2) GO TO 38
      DO 30 IH=1,ILATH
      JRX=ILATH+1-IH
      IF(KHEM.EQ.0) JRX=JRX+ILATH
      SINLAT= COA(IH)
      CALL ALPNM2(ALP,LALP,LM,SINLAT,EPSI)
      IF(KSM.NE.0) CALL ALPAS2(ALP,LALP,LM,WRKS)
      CALL FFWFG2(WRKL,ILGH,GG(1,JRX),ILG1,MAXF,ILG,WRKS,1)
      CALL FASP2(P,LR,LM,WRKL,ALP(ILP),LALP,W(IH))
   30 CONTINUE

*     DO SOUTHERN HEMISPHERE IF REGUESTED (KHEM=0 OR 2).
```

```
38 IF(KHEM.EQ.1) GO TO 50
   DO 40 IH=1,ILATH
   JRX=IH
   SINLAT=-COA(IH)
   CALL ALPNM2(ALP,LALP,LM,SINLAT,EPSI)
   IF(KSM.NE.0) CALL ALPAS2(ALP,LALP,LM,WRKS)
   CALL FFWFG2(WRKL,ILGH,GG(1,JRX),ILG1,MAXF,ILG,WRKS,1)
   CALL FASP2(P,LR,LM,WRKL,ALP(ILP),LALP,W(IH))
40 CONTINUE

   * IF ONLY ONE HEMISPHERE WAS ANALYSED, DOUBLE ALL COEFF IN P.

50 IF(KHEM.NE.0) CALL SCOF2(P,LR,LM,2)

   RETURN
   END
```

```
      SUBROUTINE GWAQD2(Q,D,LRQ,LRD,LM,UGG,VGG,ILG1,ILATH,COA,WOCS,KHEM,
     1                  ALP,DALP,EPSI,LALP,WRKS,WRKL)

C     * CONVERTS GAUSSIAN GRIDS OF U,V IN UGG,VGG(ILG1,NLAT).
C     * TO SPECTRAL COEFFICIENTS OF VORTICITY Q(LRQ,LM)
C     *  AND DIVERGENCE D(LRD,LM).
C     * IF GLOBAL, NLAT=ILAT.  IF HEMISPHERIC, NLAT=ILATH.
C     * NOTE V IS ACTUALLY V*COS(LAT)/(EARTH RADIUS). SAME FOR U.

C     * KHEM  0 = GLOBAL,  1 = N HEM ONLY,  2 = S HEM ONLY.
C     * COA(ILATH) CONTAINS THE COSINES OF THE COLATITUDE (N TO S).
C     * WOCS(ILATH) CONTAINS (GAUSSIAN WEIGHTS)/COS(COLAT) (N TO S).
C     * ALP(LALP,LM) IS A WORK FIELD FOR LEGENDRE POLYNOMIALS.
C     * DALP(LALP,LM) IS FOR THE N-S DERIVATIVES OF ALP.
C     * EPSI IS A FIELD OF CONSTANTS THE SAME SIZE AS ALP.
C     * WRKS IS AN SCM WORK ARRAY OF LENGTH LR COMPLEX WORDS.
C     * WRKL IS AN LCM WORK ARRAY OF LENGTH LR COMPLEX WORDS.
C     * FAST FOURIER TRANSFORM REQUIRES THAT ILONG BE A POWER OF 2.

      COMPLEX Q(LRQ,1),D(LRD,1),WRKL(1)
      DIMENSION UGG(ILG1,1),VGG(ILG1,1)
      DIMENSION ALP(1),DALP(1),EPSI(1)

      DIMENSION COA(1),WOCS(1),WRKS(1)

C     * INITIAL CONSTANTS. SET Q,D TO ZERO.

      ILG=ILG1-1
      ILGH=ILG/2
      ILP=1
      IF(KHEM.NE.0) ILP=LALP/2+1
      ILAT=ILATH*2
      NLAT=ILATH
      IF(KHEM.EQ.0) NLAT=ILATH*2
      CALL SCOF2(Q,LRQ,LM,0)
      CALL SCOF2(D,LRD,LM,0)

C     * PERFORM THE CONVERSION ONE ROW AT A TIME.

      DO 40 JR=1,NLAT

      IH=JR
      IF(JR.GT.ILATH) IH=ILAT+1-JR
      IF(KHEM.EQ.1) IH=ILATH+1-JR
      SINLAT=-COA(IH)
      IF(JR.GT.ILATH) SINLAT=COA(IH)
      IF(KHEM.EQ.1) SINLAT=COA(IH)
      WEIGHT=WOCS(IH)

C     * COMPUTE ALP,DALP AND REORDER ROWS IF GRIDS ARE HEMISPHERIC.
```

```
      CALL ALPNM2(ALP,LALP,LM,SINLAT,EPSI)
      CALL ALPDR2(DALP,ALP,LALP,LM,EPSI)
      IF(KHEM.NE.0) CALL ALPAS2( ALP,LALP,LM,WRKS)
      IF(KHEM.NE.0) CALL ALPAS2(DALP,LALP,LM,WRKS)

      * TERMS DEPENDING ON L.

      CALL FFWFG2(WRKL,ILGH,UGG(1,JR),ILG1,LM-1,ILG,WRKS,1)
      CALL FASP2(Q,LRQ,LM,WRKL,DALP(ILP),LALP, WEIGHT)
      DO 22 M=1,LM
      FMS=FLOAT(M-1)
      WRKL(M)=CMPLX(-FMS*AIMAG(WRKL(M)),FMS*REAL(WRKL(M)))
   22 CONTINUE
      CALL FASP2(D,LRD,LM,WRKL,ALP,LALP, WEIGHT)

      * TERMS DEPENDING ON V.

      CALL FFWFG2(WRKL,ILGH,VGG(1,JR),ILG1,LM-1,ILG,WRKS,1)
      CALL FASP2(D,LRD,LM,WRKL,DALP,LALP,-WEIGHT)
      DO 24 M=1,LM
      FMS=FLOAT(M-1)
      WRKL(M)=CMPLX(-FMS*AIMAG(WRKL(M)),FMS*REAL(WRKL(M)))
   24 CONTINUE
      CALL FASP2(Q,LRQ,LM,WRKL,ALP(ILP),LALP, WEIGHT)
   40 CONTINUE

      * IF GRIDS ARE HEMISPHERIC, DOUBLE ALL COEFF IN Q AND D.

      IF(KHEM.NE.0) CALL SCOF2(Q,LRQ,LM,2)
      IF(KHEM.NE.0) CALL SCOF2(D,LRD,LM,2)

      RETURN
      END
```

```
      SUBROUTINE GZFBP (PHI,PEE,PS,TMEAN,PEEMN,PSMN,LA,
     1                  LRS,LM,ILEV,RGAS)

*  CALCULATES GZ IN PHI FROM BIG P IN PEE FOR ILEV LEVELS.

*  IF MODEL IS HEMISPHERIC PEE,PHIS,PS ARE SYMMETRIC.
*  EACH LEVEL IS DIMENSIONED (LRS,LM).
*  EACH LEVEL IS SEPARATED BY LA COMPLEX WORDS.
*  PEE AND PHI CAN BE EQUIVALENCED.

      COMPLEX PHI(LA,1),PEE(LA,1),PS(1)
      REAL TMEAN(1),PEEMN(1)


      DO 30 L=1,ILEV

      DO 20 M=1,LM
      MR=(M-1)*LRS
      DO 20 N=1,LRS
      MN=MR+N
   20 PHI(MN,L)=PEE(MN,L) - RGAS*TMEAN(L)*PS(MN)

   30 PHI(1,L) =PHI(1,L)+PEEMN(L)-RGAS*TMEAN(L)*PSMN

      RETURN
      END
```

```
      FUNCTION HTVOCP(T)

*     COMPUTES  *HTVOCP*  ,THE RATIO  OF
*     LATENT HEAT OF VAPORIZATION OF WATER OR ICE  TO
*     SPECIFIC HEAT OF AIR AT CONSTANT PRESSURE  CP
*            REF  SMITHSONIAN TABLES,1958

       T.GE.T1S  HTVOCP(WATER) = (3.15213E+6-2.380E+3*T)/CP
       T.LE.T2S  HTVOCP( ICE ) = (2.88053E+6-0.167E+3*T)/CP
              WITH LINEAR INTERPOLATION IN BETWEEN

       PARAMETERS COMPUTED IN ADJPAR
      COMMON/HTCP/T1S,T2S,AI,BI,AW,BW,SLP

      IF(T.GE.T1S)  GO TO 1

      IF(T.LE.T2S)  GO TO 2

      HTVOCP = SLP*( (T-T2S)*(AW-BW*T) + (T1S-T)*(AI-BI*T) )
      RETURN

    1 HTVOCP = AW-BW*T
      RETURN

    2 HTVOCP = AI-BI*T
      RETURN
      END
```

```
      SUBROUTINE INPOC(LC,ILEV,NVAR)

      * PRINTER OUTPUT CONTROL FOR SPECTRAL MULTILEVEL MODEL.
      * LC = OUTPUT SWITCHES FOR EACH TIME(NT), LEVEL(ILEV), AND VARIABL

      * LAB = ALPHANUMERIC LABEL FOR EACH VARIABLE (4 CHARACTERS).
      * CS = CONTOUR INTERVAL AND SCALE FACTOR FOR EACH VARIABLE.
      * KGGM = GG WINDOW AND MAP SCALE FOR EACH TIME (IW,JW,LL,MM,MS).
      * KPSM = PS GRID SIZE,POLE AND MAP SCALE FOR EACH TIME
      *          (NI,NJ,IP,JP,MS).
      * MAXIMUM POSSIBLE VALUES... ILEV=15,  NVAR=10.

      DIMENSION LC(3,ILEV,1),LCR(3,15)
      COMMON/PCOM1/ LRPR,LMPR,KGGM(5,3),KPSM(5,3),D60,DGRW,NHEM
      COMMON/PCOM2/ LAB(10),CS(2,10)

      DIMENSION LABX(10),CSX(2,10)
      DATA CSX/-400.,1.E+7,-400.,1.E+8,-20.,1.,120.,1.,5.,.01,-500.,1.,
     1         500.,100.,  6*0. /
      DATA LABX  /4HVORT,4H DIV,4HTEMP,4H PHI,4HSFPR,4HPHIS,4H  ES,3*0 /
      DATA LAST/4HXXXX/

      * FIRST COPY CSX TO CS AND LABX TO LAB.

      DO 15 I=1,NVAR
   15 LAB(I)=LABX(I)
      DO 17 I=1,NVAR
      DO 17 K=1,2
   17 CS(K,I)=CSX(K,I)
      CS(2,4)=1./9.80616
      CS(2,6)=CS(2,4)

      * SET DEFAULT VALUE OF ALL SWITCHES TO ZERO (I.E. NO OUTPUT).

      DO 20 NV=1,NVAR
      DO 20  L=1,ILEV
      DO 20 NT=1,3
   20 LC(NT,L,NV)=0

      * READ KGGM AND KPSM FROM CARDS.

      READ(5,5020) KGGM
      WRITE(6,6020) KGGM
      READ(5,5020) KPSM,D60,DGRW,NHEM
      WRITE(6,6030) KPSM,D60,DGRW,NHEM

      * READ CONTROL CARD, CHECK LABEL, INSERT IN CORRECT PLACE IN LC.
      * REPEAT UNTIL ALL CARDS HAVE BEEN READ (LAST CARD CONTAINS XXXX )

   26 READ(5,5030) LABR,((LCR(NT,L),NT=1,3),L=1,ILEV)
      IF(LABR.EQ.LAST) GO TO 40
```

```
      DO 35 NV=1,NVAR
      IF(LABR.NE.LAB(NV)) GO TO 35
      DO 32 L=1,ILEV
      DO 32 NT=1,3
   32 LC(NT,L,NV)=LCR(NT,L)
   35 CONTINUE
      GO TO 26

*     WRITE LC ON THE PRINTER.

   40 DO 45 NV=1,NVAR
   45 WRITE(6,6040) LAB(NV),((LC(N,L,NV),N=1,3),L=1,ILEV)

      RETURN

 5020 FORMAT(15I3,E10.3,F10.2,I3)
 5030 FORMAT(A4,15(1X,3I1))
 6020 FORMAT(6H0KGGM=,3(2X,5I3))
 6030 FORMAT(6H KPSM=,3(2X,5I3),5X,14HD60,DGRW,NHEM=,E10.3,F10.2,I5)
 6040 FORMAT(1H0,A4,15H OUTPUT CONTROL,10(2X,3I2))
      END
```

```
      SUBROUTINE INPTGG(NF1,NF2,DRPAK,WTPAK,CVPAK,ILG1,ILATH,KHEM,
     1           KDR,KWT,KCV,NPDR,NPWT,NPCV,GG,GGP,WRKS)

      * GETS GAUSSIAN GRIDS OF DRAG,WATER-TEMP, AND SEA-COVER.
      *  FOR THE SPECTRAL MULTILEVEL MODEL.
      * THEY ARE ALSO WRITTEN TO FINE NF2 FOR POSSIBLE RESTART.

      *  KDR,ETC = LOGICAL SWITCHES FOR EACH VARIABLE.
      * NPDR,ETC = PACKING DENSITIES FOR EACH VARIABLE IN THE MODEL.
      *            (ALSO USED FOR OUTPUT PACKING ON NF2).
      * GG,GGP,WRKS ARE WORK FIELDS USED IN PACKING.

      * INPUT FIELDS ON FILE NF1 CAN BE IN ANY ORDER, BUT
      * THEY SHOULD OCCUR AT THE BEGINNING OF NF1 TO MINIMIZE THE
      *  EFFECT OF THE REWINDS.


      DIMENSION DRPAK(1),WTPAK(1),CVPAK(1)
      DIMENSION GG(ILG1,1),GGP(1)
      DIMENSION WRKS(1)

      LOGICAL OK,KDR,KWT,KCV

      * /MAPGG/ CONTAINS MAP CONTROLS (SET IN THE MAIN PGM).

      COMMON/MAPGG/ MGGDR,MGGWT,MGGCV

      ID=0
      NLAT=ILATH
      IF(KHEM.EQ.0) NLAT=ILATH*2

      * DRAG FIELD (CRESSMAN DRAG COEFFICIENT - NONDIMENSIONAL).

      IF(.NOT.KDR) GO TO 30
      REWIND NF1
      CALL RSGGP(DRPAK,ILG1,NLAT,NF1,ID,4H  DR,1,OK,GGP,WRKS)
      CALL WSGGP(DRPAK,ILG1,NLAT,NF2,ID,4H  DR,1,NPDR,GGP,WRKS)
      CALL FCONW2(DRPAK,100.,1.E5,ILG1,NLAT,1,1,ILG1,NLAT,MGGDR)

      * WATER TEMPERATURE (DEG K).

   30 IF(.NOT.KWT) GO TO 40
      REWIND NF1
      CALL RSGGP(WTPAK,ILG1,NLAT,NF1,ID,4H  WT,1,OK,GGP,WRKS)
      CALL WSGGP(WTPAK,ILG1,NLAT,NF2,ID,4H  WT,1,NPWT,GGP,WRKS)
      CALL FCONW2(WTPAK,3.,1.,ILG1,NLAT,1,1,ILG1,NLAT,MGGWT)

      * LAND - SEA COVER (FRACTION OF WATER 0. TO 1.).

   40 IF(.NOT.KCV) GO TO 99
      REWIND NF1
```

```
      CALL RSGGP(CVPAK,ILG1,NLAT,NF1,ID,4H  CV,1,OK,GGP,WRKS)
      CALL WSGGP(CVPAK,ILG1,NLAT,NF2,ID,4H  CV,1,NPCV,GGP,WRKS)
      CALL FCONW2(CVPAK,-100.,100.,ILG1,NLAT,1,1,ILG1,NLAT,MGGCV)

   99 RETURN
      END
```

```
      SUBROUTINE INPTSP(NF1,ID,NF2,PHIS,PS,PHI,P,C,ES,LA,LRS,LRA,
     1                  LM,ILEV,LEVS,KHEM,ILG1,ILATH,COA,W,WOCS,GG,GGX,GGP,
     2                  ALP,DALP,EPSI,LALP,WRKS,WRKL)

C     * PERFORMS SPECTRAL ANALYSIS OF GAUSSIAN GRIDS (ILG1,ILAT)
C     *  FOR INPUT TO THE MULTILEVEL SPECTRAL MODEL.
C     * LRS,LRA ARE ROW LENGTHS FOR SYMMETRIC AND ANTISYMMETRIC
C     *  SPECTRAL ARRAYS. THEY DIFFER ONLY IF THE MODEL IS HEMISPHERIC.

C     * KHEM  0 = GLOBAL,  1 = N HEM ONLY,  2 = S HEM ONLY.     (GG)

C     *  INPUT IS ON SEQUENTIAL FILE NF1 IN THE FOLLOWING ORDER....
C     *    PHIS,PS,PHI(ILEV),(U,V(ILEV)),ES(ILEV).

      COMPLEX PHIS(1),PS(1),PHI(LA,1),P(LA,1),C(LA,1),ES(LA,1)
      DIMENSION ALP(1),DALP(1),EPSI(1)
      DIMENSION COA(1),W(1),WOCS(1),WRKS(1)

C     * GG,GGX ARE WORK FIELDS FOR GAUSSIAN GRIDS (ILG1,NLAT).
C     * GGP IS A WORK FIELD USED ONLY IF NF1 CONTAINS PACKED FIELDS.
C     * WRKS,WRKL ARE SCM,LCM WORK ARRAYS OF (ILG+2) WORDS.

      DIMENSION GG(1),GGX(1),GGP(1),WRKL(1)

      LOGICAL OK

C     * OUTPUT PACKING DENSITIES (SET IN MAIN PROGRAM).

      COMMON/PKOUT/ NPGZ,NPSP,NPW,NPES

      REWIND NF1
      KSYM=1
      IF(KHEM.EQ.0) KSYM=0
      NLAT=ILATH
      IF(KHEM.EQ.0) NLAT=ILATH*2

C     * MOUNTAIN FIELD (M/SEC)**2  READ FROM FILE NF1.
C     * NOTE - MOUNTAINS ARE WRITTEN TO FILE NF2.

      CALL RSGGP(GG,ILG1,NLAT,NF1, 0,4H GZS,1,OK,GGP,WRKS)
      CALL GGASP2(PHIS,LRS,LM,KHEM,GG,ILG1,ILATH,COA,W,KSYM,
     1                        ALP,EPSI,LALP,WRKS,WRKL)
      CALL SPAGG2(GG,ILG1,ILATH,COA,KHEM,PHIS,LRS,LM,KSYM,
     1                        ALP,EPSI,LALP,WRKS,WRKL)
      CALL WSGGP(GG,ILG1,NLAT,NF2, 0,4H GZS,1,NPGZ,GGP,WRKS)

C     * LOG OF SURFACE PRESSURE (MILLIBARS) READ FROM FILE NF1.
C     * CONVERT FROM MILLIBARS TO (NEWTONS/M**2).

      CALL RSGGP(GG,ILG1,NLAT,NF1,ID,4HLNSP,1,OK,GGP,WRKS)
      CALL GGASP2( PS ,LRS,LM,KHEM,GG,ILG1,ILATH,COA,W,KSYM,
```

```
     1                       ALP,EPSI,LALP,WRKS,WRKL)
      PS(1) = PS(1)+ALOG(100.)*SQRT(2.)

      * GEOPOTENTIALS (M/SEC)**2 READ FROM FILE NF1.

      DO 250 L=1,ILEV
      CALL RSGGP(GG,ILG1,NLAT,NF1,ID,4H  GZ,L,OK,GGP,WRKS)
      CALL GGASP2(PHI(1,L),LRS,LM,KHEM,GG,ILG1,ILATH,COA,W,KSYM,
     1                       ALP,EPSI,LALP,WRKS,WRKL)
  250 CONTINUE

      * WIND COMPONENTS (U,V)*COS(PHI)/(EARTH RADIUS) READ FROM FILE NF1
      * CONVERT TO VORTICITY (1./SEC) AND DIVERGENCE (1./SEC).

      DO 350 L=1,ILEV
      CALL RSGGP( GG,ILG1,NLAT,NF1,ID,4H   U,L,OK,GGP,WRKS)
      CALL RSGGP(GGX,ILG1,NLAT,NF1,ID,4H   V,L,OK,GGP,WRKS)
      CALL GWAQD2(P(1,L),C(1,L),LRA,LRS,LM,GG,GGX,ILG1,ILATH,COA,WOCS,
     1                  KHEM,ALP,DALP,EPSI,LALP,WRKS,WRKL)
  350 CONTINUE

      * DEW POINT DEPRESSIONS ES (DEG C) READ FROM FILE NF1.

      IF(LEVS.EQ.0) GO TO 999
      DO 450 N=1,LEVS
      L=(ILEV-LEVS) + N
      CALL RSGGP(GG,ILG1,NLAT,NF1,ID,4H  ES,L,OK,GGP,WRKS)
      CALL GGASP2( ES(1,N),LRS,LM,KHEM,GG,ILG1,ILATH,COA,W,KHEM,
     1                       ALP,EPSI,LALP,WRKS,WRKL)
  450 CONTINUE

  999 RETURN
      END
```

```
      SUBROUTINE INVRSI(PEEBAR,AI,BI,CI,DI,LA,LRS,LM,ILEV,KHEM)

    * THIS ROUTINE IS REQUIRED TO IMPLEMENT THE SEMI-IMPLICIT
    * ALGORITHM.  THE TRI-DIAGONAL MATRIX DEFINED BY AI,BI, AND CI
    * IS SOLVED TO OBTAIN THE PEEBAR FROM THE RHS (DI)  FOR EACH
    * HORIZONTAL MODE.  THE GENERAL TRI-DIAGONAL MATRIX SOLVER
    * ROSR12 IS USED FOR SOLUTION.

      * IF MODEL IS HEMISPHERIC, PEEBAR,DI ARE SYMMETRIC.
      * EACH LEVEL IS DIMENSIONED (LRS,LM).
      * EACH LEVEL IS SEPARATED BY LA COMPLEX WORDS.
      * MAXIMUM VALUE OF ILEV IS 15.

      COMPLEX PEEBAR(LA,1),DI(LA,1)
      DIMENSION AI(ILEV,1),BI(ILEV,1),CI(ILEV,1)
      DIMENSION PEBR(15),PEBI(15),DIR(15),DII(15),WORK(15)


      DO 40 M=1,LM
      DO 40 N=1,LRS
      IL=(M-1)*LRS+N
      NS=M+N-2
      IF(KHEM.NE.0) NS=NS+(N-1)
      NSP=NS+1

      DO 30 IH=1,ILEV
      DIR(IH) = REAL(DI(IL,IH))
      DII(IH) = AIMAG(DI(IL,IH))
   30 CONTINUE

      CALL ROSR12(PEBR,AI(1,NSP),BI(1,NSP),CI(1,NSP),DIR,WORK,ILEV)
      CALL ROSR12(PEBI,AI(1,NSP),BI(1,NSP),CI(1,NSP),DII,WORK,ILEV)

      DO 31 IH=1,ILEV
      PEEBAR(IL,IH) = CMPLX(PEBR(IH),PEBI(IH))
   31 CONTINUE

   40 CONTINUE

      RETURN
      END
```

```
      SUBROUTINE LLFXY(DLAT,DLON,X,Y,D60,DGRW,NHEM)

      * CALCULATE LATITUDE AND LONGITUDE IN DEGREES OF POINT (X,Y)
      *  MEASURED FROM THE POLE. (LONGITUDE IS POSITIVE EASTWARD).
      * GRID IS POLAR STEREOGRAPHIC WITH STANDARD LATITUDE AT 60 DEG.
      *  AND GRID SIZE D60 METERS.
      * ZERO DEGREES LONGITUDE IN THE GRID IS (DGRW) DEGREES
      *  IN MAP COORDINATES.
      * NHEM 1 = NORTHERN HEMISPHERE.   NHEM 2 = SOUTHERN HEMISPHERE.
      * 1.866025=(1+SIN60),     6.371E+6=EARTH RADIUS IN METERS.

      RE=1.866025*6.371E+6/D60
      RE2=RE**2
      C1=180./3.14159

      * IF POINT IS AT POLE SET COORD TO (0.,90.).

      DLAT=90.
      DLON=0.
      IF(X.EQ.0. .AND. Y.EQ.0.) GO TO 39

      * CALCULATE LONGITUDE IN MAP COORDINATES.

      IF(X.EQ.0.) DLON=SIGN(90.,Y)
      IF(X.NE.0.) DLON=ATAN(Y/X)*C1
      IF(X.LT.0.) DLON=DLON+SIGN(180.,Y)

      * ADJUST LONGITUDE FOR GRID ORIENTATION.

      DLON=DLON-DGRW
      IF(DLON.GT.+180.) DLON=DLON-360.
      IF(DLON.LT.-180.) DLON=DLON+360.

      * CALCULATE LATITUDE.

      R2=X**2+Y**2
      DLAT=(RE2-R2)/(RE2+R2)
      DLAT= ASIN(DLAT)*C1

      * CHANGE SIGNS IF IN SOUTHERN HEMISPHERE.

   39 IF(NHEM.EQ.2) DLAT=-DLAT
      IF(NHEM.EQ.2) DLON=-DLON
      RETURN
      END
```

```
      SUBROUTINE LNER(CT,PEET,SDS,CM,PEEM,LA,LRS,LM,ILEV,DT,RGAS,SF)
      * ADD LINEAR TERMS OF RHS TO PEET,SDS.
      PARAMETER $IMV = 15, IM = 5, $IR = 25, IRM = 2*$IR+1
      * IF MODEL IS HEMISPHERIC, ALL FIELDS ARE SYMMETRIC.
      * EACH LEVEL IS DIMENSIONED (LRS,LM).
      * EACH LEVEL IS SEPARATED BY LA COMPLEX WORDS.

      COMPLEX CT(LA,1),PEET(LA,1),SDS(LA,1),CM(LA,1),PEEM(LA,1)
      COMPLEX TEMP($IMV)
      REAL MI,MII,MIIM1,M1,M2,MP,MPM1,MM,MW,MPI,M2M1
      COMMON / NEWMAT / MI(IM,IM),MII(IM,IM),MIIM1(IM,IM),M1(IM,IM),
     *                  M2(IM,IM),MP(IM,IM),MPM1(IM,IM),MM(IM,IM),
     *                  MW(IM,IM),MPI(IM,IM,IRM),M2M1(IM,IM)
      DIMENSION SF(1)

      ILEVM=ILEV-1

      DO 30 M=1,LM
      DO 30 N=1,LRS
      IL=(M-1)*LRS+N

      DO 20 IH=1,ILEV
      SDS(IL,IH) = -SDS(IL,IH) - DT*CT(IL,IH) - CM(IL,IH)
   20 CONTINUE

      DO 24 IH=1,ILEV
      TEMP(IH)=0.0
      DO 24 IK=1,ILEV
   24 TEMP(IH)=TEMP(IH) + MPM1(IK,IH)*PEET(IL,IK)
      DO  25 IH=1,ILEV
   25 TEMP(IH) = DT * TEMP(IH)
      DO 26 IH=1,ILEV
   26 PEET(IL,IH) = TEMP(IH) + PEEM(IL,IH)

   30 CONTINUE
      RETURN
      END
            SUBROUTINE MATMLT(X1,X2,X3,N)
            DIMENSION X1(N,N),X2(N,N),X3(N,N)
            DO 1 J=1,N
            DO 1 I=1,N
            X1(I,J)=0.0
            DO 1 K=1,N
    1       X1(I,J)=X1(I,J)+X2(I,K)*X3(K,J)
            RETURN
            END
```

AD-A058 244 NAVAL RESEARCH LAB WASHINGTON D C F/G 4/2
THE MODIFIED CANADIAN SPECTRAL FORECAST MODEL: DISCUSSION AND D--ETC(U)
MAR 78 W W JONES, R V MADALA, M R SCHOEBERL

UNCLASSIFIED NRL-MR-3698 SBIE-AD-E000 193 NL

2 OF 3

AD
A058244

```
      SUBROUTINE MHANLW(PT,CT,PEET,SDS,TDUM,EST,ESDUM,PRESS,LA,
     1                 PUTF,PVTF,TUTF,TVTF,PEETF,SDSF,EF,TF,SUSF,SVSF,
     2             ESTF,ESF,PRESSF,ILH,ILM,LRS,LRA,LM,
     3             ALP,DALP,LALP,WRKS,WL,WOCSL,ILONG,KHEM,ILEV,LEVS)

*  CONVERTS ILONG GRID VALUES TO FOURIER COEFF THEN CALCULATES
*  CONTRIBUTIONS TO SPECTRAL FIELDS FOR ONE GAUSSIAN LATITUDE.
*  MOISTURE VARIABLES OMITTED IF LEVS = 0.

*  IF THE MODEL IS HEMISPHERIC PT IS ANTISYMMETRIC.
*   ALL OTHER SPECTRAL FIELDS ARE SYMMETRIC.

*  ALP(LALP,LM) CONTAINS LEGENDRE POLYNOMIALS FOR ONE LATITUDE.
*  DALP(LALP,LM) CONTAINS N-S DERIVATIVE OF ALP.
*  WRKS IS AN SCM WORK ARRAY USED BY THE FAST FOURIER TRANSFORM.

      COMPLEX PT(LA,1),CT(LA,1),PEET(LA,1),SDS(LA,1)
      COMPLEX EST(LA,1),TDUM(LA,1),ESDUM(LA,1),PRESS(1)

      COMPLEX PUTF(ILH,1),PVTF(ILH,1),TUTF(ILH,1),TVTF(ILH,1)
      COMPLEX PEETF(ILH,1),SDSF(ILH,1),EF(ILH,1),TF(ILH,1)
      COMPLEX SUSF(ILH,1),SVSF(ILH,1),ESTF(ILH,1),ESF(ILH,1)
      COMPLEX PRESSF(1)

      DIMENSION DALP(1),ALP(1),WRKS(1)

*  IF MODEL IS HEMISPHERIC, THE SYMMETRIC AND ANTISYMMETRIC
*    COEFF IN THE ROWS OF ALP,DALP HAVE BEEN SEPARATED.

      ILEVP=ILEV+1
      ILP=1
      IF(KHEM.NE.0) ILP=LALP/2+1
      IR=LM-1

*  ALSO, WI AND WOCSI ARE DOUBLED TO ACCOUNT FOR THE MISSING HEM.

      WI=WL
      IF(KHEM.NE.0) WI=WL+WL
      WOCSI=WOCSL
      IF(KHEM.NE.0) WOCSI=WOCSL+WOCSL

*  CONVERT GRID POINT VALUES TO  FOURIER COEFF.
*  ILH,ILM=REAL,COMPLEX DIMENSIONS OF PEETF,ETC (ILM=2*ILH).

      CALL FFWFG2(PEETF,ILH,PEETF,ILM,IR,ILONG,WRKS,ILEV)
      CALL FFWFG2( TUTF,ILH, TUTF,ILM,IR,ILONG,WRKS,ILEV)
      CALL FFWFG2( TVTF,ILH, TVTF,ILM,IR,ILONG,WRKS,ILEV)
      CALL FFWFG2( PUTF,ILH, PUTF,ILM,IR,ILONG,WRKS,ILEV)
      CALL FFWFG2( PVTF,ILH, PVTF,ILM,IR,ILONG,WRKS,ILEV)
      CALL FFWFG2( SDSF,ILH, SDSF,ILM,IR,ILONG,WRKS,ILEV)
      CALL FFWFG2(   TF,ILH,   TF,ILM,IR,ILONG,WRKS,ILEVP)
```

```
      CALL FFWFG2(PRESSF,ILM,PRESSF,ILM,IR,ILONG,WRKS,1)

      * MOISTURE VARIABLES (OMITTED IF LEVS=0).

      IF(LEVS.EQ.0) GO TO 13
      CALL FFWFG2( SUSF,ILM, SUSF,ILM,IR,ILONG,WRKS,LEVS)
      CALL FFWFG2( SVSF,ILM, SVSF,ILM,IR,ILONG,WRKS,LEVS)
      CALL FFWFG2( ESTF,ILM, ESTF,ILM,IR,ILONG,WRKS,LEVS)
      CALL FFWFG2(  ESF,ILM,  ESF,ILM,IR,ILONG,WRKS,LEVS)


      *  FOURIER CONTRIBUTIONS AT ONE LAT CONVERTED TO SPECTRAL COEFF.
      * FOR VORTICITY AND DIVERGENCE-TYPE TERMS OF PT,CT,PEET,EST.

      * CALCULATION OF NORTH-SOUTH DERIVATIVE TERMS.

   13 DO 20 L=1,ILEV
      CALL FASP2(PEET(1,L),LRS,LM,TVTF(1,L),DALP      ,LALP,-WOCSI)
      CALL FASP2(  PT(1,L),LRA,LM,PVTF(1,L),DALP(ILP),LALP, WOCSI)
      CALL FASP2(  CT(1,L),LRS,LM,PUTF(1,L),DALP      ,LALP, WOCSI)
   20 CONTINUE

     * MOISTURE VARIABLES (OMITTED IF LEVS=0)

      IF(LEVS.EQ.0) GO TO 30
      DO 21 L=1,LEVS
      CALL FASP2( EST(1,L),LRS,LM,SVSF(1,L),DALP,LALP, WOCSI)
   21 CONTINUE

      * EAST - WEST DERIVATIVE TERMS CALCULATED

   30 DO 36 L=1,ILEV
      FSQ = WOCSI/WI

      DO 32 M=1,LM
      BI = FLOAT(M-1)*FSQ
      SCR = - BI*AIMAG(TUTF(M,L))
      SCI = BI*REAL(TUTF(M,L))
      PEETF(M,L) = PEETF(M,L) + CMPLX(SCR,SCI)
   32 CONTINUE

      DO 33 M=1,LM
      BI = FLOAT(M-1)*FSQ
      SCR = BI*AIMAG(PUTF(M,L))
      SCI = -BI*REAL(PUTF(M,L))
      PUTF(M,L) = CMPLX(SCR,SCI)
   33 CONTINUE

      DO 34 M=1,LM
      BI = FLOAT(M-1)*FSQ
      SCR = -BI*AIMAG(PVTF(M,L))
```

```
      SCI = BI*REAL(PVTF(M,L))
      PVTF(M,L) = CMPLX(SCR,SCI)
   34 CONTINUE

      CALL FASP2(  PT(1,L),LRA,LM,PUTF(1,L), ALP(ILP),LALP, WI)
      CALL FASP2(  CT(1,L),LRS,LM,PVTF(1,L), ALP,LALP, WI)
   36 CONTINUE

    * MOISTURE VARIABLES (OMITTED IF LEVS=0)

      IF(LEVS.EQ.0) GO TO 49
      DO 46 L=1,LEVS
      FSQ = WOCSI/WI

      DO 41 M=1,LM
      BI = FLOAT(M-1)*FSQ
      SCR = BI*AIMAG(SUSF(M,L))
      SCI = -BI*REAL(SUSF(M,L))
      ESTF(M,L) = ESTF(M,L) + CMPLX(SCR,SCI)
   41 CONTINUE

   46 CONTINUE

      * ADD THE REMAINING TERMS.

   49 DO 50 L=1,ILEV
      CALL FASP2(PEET(1,L),LRS,LM,PEETF(1,L),ALP,LALP,WI)
      CALL FASP2( SDS(1,L),LRS,LM, SDSF(1,L),ALP,LALP,WI)
      CALL FASP2(TDUM(1,L),LRS,LM,    TF(1,L),ALP,LALP,WI)
   50 CONTINUE
      CALL FASP2(TDUM(1,ILEVP),LRS,LM,TF(1,ILEVP),ALP,LALP,WI)
      CALL FASP2(PRESS,LRS,LM,PRESSF,ALP,LALP,WI)

      * MOISTURE VARIABLES (OMITTED IF LEVS=0).

      IF(LEVS.EQ.0) GO TO 99
      DO 52 L=1,LEVS
      CALL FASP2( EST(1,L),LRS,LM, ESTF(1,L),ALP,LALP,WI)
      CALL FASP2(ESDUM(1,L),LRS,LM,ESF(1,L),ALP,LALP,WI)
   52 CONTINUE

   99 RETURN
      END
```

```
      SUBROUTINE MHEXPW(PF,CF,TF,ESF,UF,VF,PSDLF,PSDPF,PRESSF,ILH,ILM,
     1           P,C,T,ES,U,V,PS,LA,LAW,LRS,LRA,LRU,LRV,LM,ILEV,LEVS,
     2                     ILONG,KHEM,ALP,DALP,LALP,WRKS)

*  CONVERTS ARRAYS OF SPECTRAL COEFF TO FOURIER COEFF
*   AND THEN TO ILONG GRID VALUES FOR ONE GAUSSIAN LATITUDE.

*  IF MODEL IS HEMISPHERIC C,T,ES,U,PS ARE SYMMETRIC,
*                          P,V ARE ANTISYMMETRIC.
*  LEVELS OF U,V ARE SEPARATED BY LAW COMPLEX WORDS.
*   ALL OTHER VARIABLES USE LA COMPLEX WORDS.

*  ALP(LALP,LM) CONTAINS LEGENDRE POLYNOMIALS FOR ONE LATITUDE.
*  DALP(LALP,LM) CONTAINS N-S DERIVATIVE OF ALP.
*  WRKS IS AN SCM WORK ARRAY USED BY THE FAST FOURIER TRANSFORM.

      COMPLEX P(LA,1),C(LA,1),T(LA,1),ES(LA,1)
      COMPLEX U(LAW,1),V(LAW,1),PS(1)

      COMPLEX PF(ILH,1),CF(ILH,1),TF(ILH,1),ESF(ILH,1)
      COMPLEX UF(ILH,1),VF(ILH,1)
      COMPLEX PSDLF(1),PSDPF(1),PRESSF(1)

      DIMENSION ALP(1),DALP(1)
      DIMENSION WRKS(1)

*  IF MODEL IS HEMISPHERIC, THE SYMMETRIC AND ANTISYMMETRIC
*   COEFF IN THE ROWS OF ALP,DALP HAVE BEEN SEPARATED.

      ILEVP=ILEV+1
      ILP=1
      IF(KHEM.NE.0) ILP=LALP/2+1
      IR=LM-1

*  FOURIER COEFF OF P,C,T,U,V.

      DO 70 L=1,ILEV
      CALL SPAF2(PF(1,L),P(1,L),LRA,LM,ALP(ILP),LALP)
      CALL SPAF2(CF(1,L),C(1,L),LRS,LM,ALP,LALP)
      CALL SPAF2(TF(1,L),T(1,L),LRS,LM,ALP,LALP)
      CALL SPAF2(UF(1,L),U(1,L),LRU,LM,ALP,LALP)
      CALL SPAF2(VF(1,L),V(1,L),LRV,LM,ALP(ILP),LALP)
   70 CONTINUE
      CALL SPAF2(TF(1,ILEVP),T(1,ILEVP),LRS,LM,ALP,LALP)

*  FOURIER COEFF OF PS AND ITS N-S AND E-W DERIVATIVES.

      CALL SPAF2(PRESSF,PS,LRS,LM,ALP,LALP)
      CALL SPAF2( PSDPF,PS,LRS,LM,DALP,LALP)

      PSDLF(1)=(0.,0.)
```

```
      DO 72 M=2,LM
      FMS=FLOAT(M-1)
      RPS= REAL(PRESSF(M))
      CPS=AIMAG(PRESSF(M))
      PSDLF(M)=CMPLX(-FMS*CPS, FMS*RPS)
   72 CONTINUE

      * CONVERT  FOURIER COEFF TO GRID POINT VALUES.

      CALL FFGFW2(PF,ILM,PF,ILH,IR,ILONG,WRKS,ILEV)
      CALL FFGFW2(CF,ILM,CF,ILH,IR,ILONG,WRKS,ILEV)
      CALL FFGFW2(TF,ILM,TF,ILH,IR,ILONG,WRKS,ILEVP)
      CALL FFGFW2(UF,ILM,UF,ILH,IR,ILONG,WRKS,ILEV)
      CALL FFGFW2(VF,ILM,VF,ILH,IR,ILONG,WRKS,ILEV)
      CALL FFGFW2( PSDLF,ILM, PSDLF,ILH,IR,ILONG,WRKS,1)
      CALL FFGFW2( PSDPF,ILM, PSDPF,ILH,IR,ILONG,WRKS,1)
      CALL FFGFW2(PRESSF,ILM,PRESSF,ILH,IR,ILONG,WRKS,1)

      * MOISTURE VARIABLES (OMITTED IF LEVS =0).

      IF(LEVS.EQ.0) GO TO 99
      DO 90 L=1,LEVS
      CALL SPAF2(ESF(1,L),ES(1,L),LRS,LM,ALP,LALP)
   90 CONTINUE
      CALL FFGFW2(ESF,ILM,ESF,ILH,IR,ILONG,WRKS,LEVS)

   99 RETURN
      END
```

```
      SUBROUTINE MTXINV(A,Z,N)

      * FINDS THE INVERSE OF MATRIX Z(N,N) BY THE EXCHANGE METHOD.
      * MAX VALUE OF N IS 15.   RESULT IS PUT IN A(N,N).

      DIMENSION A(N,N),Z(N,N)
      DIMENSION IX(15),IC(15)


      DO 10 J=1,N
      IX(J)=0
      DO 10 I=1,N
   10 A(I,J)=Z(I,J)

      DO 42 K=1,N
      B=0.0
      DO 2 J=1,N
      IF(IX(J).GT.0) GO TO 2
      S = ABS(A(K,J))
      IF(S.LE.B) GO TO 2
      B=S
      L=J
    2 CONTINUE
      IX(L)=K
      DO 4 J=1,N
      D=A(K,J)/A(K,L)
      IF(J.EQ.L) GO TO 4
      DO 3 I=1,N
    3 A(I,J)=A(I,J)-D*A(I,L)
      A(K,J)=-D
    4 CONTINUE
      D = 1.0/A(K,L)
      DO 41 I=1,N
   41 A(I,L) = D*A(I,L)
      A(K,L) = D
   42 CONTINUE

      * REORDER ROWS AND COLUMNS.

      DO 45 I=1,N
   45 IC(IX(I))=I
      DO 70 K=1,N
      IF(IX(K).EQ.K) GO TO 60
      DO 51 L=1,N
      IF(IX(L).EQ.K) GO TO 53
   51 CONTINUE
   53 DO 55 I=1,N
      D=A(I,K)
      A(I,K)=A(I,L)
   55 A(I,L)=D
      IX(L)=IX(K)
```

```
      IX(K)=K
   60 IF(IC(K).EQ.K) GO TO 70
      DO 61 L=1,N
      IF(IC(L).EQ.K) GO TO 63
   61 CONTINUE
   63 DO 65 J=1,N
      D=A(K,J)
      A(K,J)=A(L,J)
   65 A(L,J)=D
      IC(L)=IC(K)
      IC(K)=K
   70 CONTINUE

      RETURN
      END
```

```
      SUBROUTINE NEWBP(PEE,PEEM,PEEBAR,LA,LRS,LM,ILEV,FPEE,IFDIFF)

      * PERFORMS ONE TIMESTEP FOR PEE FROM PEEM,PEEBAR.

      * IF MODEL IS HEMISPHERIC PEE,PEEM,PEEBAR ARE SYMMETRIC.
      * EACH LEVEL IS DIMENSIONED (LRS,LM).
      * EACH LEVEL IS SEPARATED BY LA COMPLEX WORDS.

      * IFDIFF...1=FOREWARD TIMESTEP,  0=CENTERED TIMESTEP.
      * FPEE IS A TIME FILTER FOR PEE.

      COMPLEX PEE(LA,1),PEEM(LA,1),PEEBAR(LA,1)
      COMPLEX PEEP

      DO 30 L=1,ILEV

      DO 20 M=1,LM
      MR=(M-1)*LRS
      DO 20 N=1,LRS
      MN=MR+N
      PEEP=2.*PEEBAR(MN,L)-PEEM(MN,L)
      PEE(MN,L)=PEE(MN,L) + FPEE*(PEEP+PEEM(MN,L)-2.*PEE(MN,L))
      IF(IFDIFF.EQ.0) PEEM(MN,L)=PEE(MN,L)
   20 PEE(MN,L)=PEEP

   30 CONTINUE

      RETURN
      END
```

```
      SUBROUTINE NEWC(C,CM,CT,PEEBAR,LA,LRS,LM,ILEV,ASQ,
     1                 DIFUSD,DT,FC,IFDIFF,KHEM)

      * PERFORMS ONE TIMESTEP FOR DIVERGENCE (C) FROM CM,CT,PEEBAR.

      * IF MODEL IS HEMISPHERIC C,CM,CT,PEEBAR ARE SYMMETRIC.
      * EACH LEVEL IS DIMENSIONED (LRS,LM).
      * EACH LEVEL IS SEPARATED BY LA COMPLEX WORDS.

      * DIFUSD = HORIZONTAL DIFFUSION OF DIVERGENCE.
      * DT IS THE LENGTH OF THE JUMP (DEET IF CENTERED, DEET/2 IF FWD).
      * FC IS A TIME FILTER FOR C.
      * IFDIFF...1=FOREWARD TIMESTEP,  0=CENTERED TIMESTEP.
      * KHEM=0 FOR GLOBAL, 1 OR 2 FOR HEMISPHERIC.

      COMPLEX C(LA,1),CM(LA,1),CT(LA,1),PEEBAR(LA,1)

      COMPLEX CP

      FACTD=1./(1.+2.*DT*DIFUSD)

      DO 30 L=1,ILEV

      DO 20 M=1,LM
      MR=(M-1)*LRS
      DO 20 N=1,LRS
      MN=MR+N
      NS=(M-1)+(N-1)
      IF(KHEM.NE.0) NS=NS+(N-1)
      FNS1=FLOAT(NS*(NS+1))

      CP=2.*DT*(FNS1*PEEBAR(MN,L)/ASQ + CT(MN,L)) + CM(MN,L)
      CP=CP*FACTD
      C(MN,L)=C(MN,L) + FC*(CP+CM(MN,L)-2.*C(MN,L))
      IF(IFDIFF.EQ.0) CM(MN,L)=C(MN,L)
      C(MN,L)=CP

   20 CONTINUE

   30 CONTINUE

      RETURN
      END
```

```
      SUBROUTINE NEWES(ES,ESM,EST,LA,LRS,LM,LEVS,ASQ,
     1                 DIFUSS,DT,FS,IFDIFF,KHEM)

* MULTILEVEL SPECTRAL MODEL TIMESTEP SUBROUTINE.
* OBTAINS VALUES AT THE NEXT TIMESTEP OF DEW-POINT DEPRESSION (ES)
*  FROM THE TENDENCIES (EST) AND THE PREVIOUS VALUES (ESM).

* IF MODEL IS HEMISPHERIC ES,ESM,EST ARE SYMMETRIC.
* EACH LEVEL IS DIMENSIONED (LRS,LM).
* EACH LEVEL IS SEPARATED BY LA COMPLEX WORDS.

* LEVS = NUMBER OF MOISTURE LEVELS IN THE MODEL.
* TO OMIT MOISTURE CALCULATIONS SET LEVS = 0.
* DIFUSS = HORIZONTAL DIFFUSION OF ES.
* DT IS THE LENGTH OF THE JUMP (DEET IF CENTERED, DEET/2 IF FWD).
* FS = TIME FILTER FOR ES.
* IFDIFF...1=FOREWARD TIMESTEP,  0=CENTERED TIMESTEP.
* KHEM=0 FOR GLOBAL, 1 OR 2 FOR HEMISPHERIC.

      COMPLEX ES(LA,1),EST(LA,1),ESM(LA,1)

      COMPLEX ESP

      IF(LEVS.EQ.0) RETURN
      FACTS=DIFUSS/ASQ

      DO 40 M=1,LM
      DO 40 N=1,LRS
      IL=(M-1)*LRS+N
      NS=M+N-2
      IF(KHEM.NE.0) NS=NS+(N-1)
      FNS1=FLOAT(NS*(NS+1))
   28 DIFLPS=FACTS*FNS1

      DO 30 IH=1,LEVS
      ESP=ESM(IL,IH)+2.*DT*(EST(IL,IH)-DIFLPS*ESM(IL,IH))
      ES(IL,IH)=ES(IL,IH) + FS*(ESP+ESM(IL,IH)-2.*ES(IL,IH))
      IF(IFDIFF.EQ.0) ESM(IL,IH)=ES(IL,IH)
      ES(IL,IH)=ESP
   30 CONTINUE

   40 CONTINUE

      RETURN
      END
```

```
      SUBROUTINE NEWP (P,PM,PT,LA,LRA,LM,ILEV,DT,FP,IFDIFF)

      * PERFORMS ONE TIMESTEP FOR VORTICITY (P) FROM PM AND PT.

      * IF MODEL IS HEMISPHERIC P,PM,PT ARE ANTISYMMETRIC
      * EACH LEVEL IS DIMENSIONED (LRA,LM).
      * EACH LEVEL IS SEPARATED BY LA COMPLEX WORDS.

      * DT IS THE LENGTH OF THE JUMP (DEET IF CENTERED, DEET/2 IF FWD).
      * FP IS A TIME FILTER FOR P.
      * IFDIFF...1=FOREWARD TIMESTEP,  0=CENTERED TIMESTEP.

      COMPLEX P(LA,1),PM(LA,1),PT(LA,1)

      COMPLEX PP

      DO 30 L=1,ILEV

      DO 20 M=1,LM
      MR=(M-1)*LRA
      DO 20 N=1,LRA
      MN=MR+N
      PP=PM(MN,L) + 2.*DT*PT(MN,L)
      P(MN,L)=P(MN,L) + FP*(PP+PM(MN,L)-2.*P(MN,L))
      IF(IFDIFF.EQ.0) PM(MN,L)=P(MN,L)
      P(MN,L)=PP

   20 CONTINUE

   30 CONTINUE

      RETURN
      END
```

```
      SUBROUTINE NEWPS(PS,PSM,PEET,PEEBAR,LA,LRS,LM,ILEV,AMBDA,SF,AVERT,
     1                     RGAS,DT,FPS,IFDIFF)

      PARAMETER IM=5, IRM=2*25+1

*     PERFORMS ONE TIMESTEP FOR LN(SURFACE PRESSURE) IN PS
*      FROM PSM,PEET AND PEEBAR.

*     IF MODEL IS HEMISPHERIC PS,PSM,PEET,PEEBAR ARE SYMMETRIC.
*     EACH LEVEL IS DIMENSIONED (LRS,LM).
*     EACH LEVEL IS SEPARATED BY LA COMPLEX WORDS.

*     DT IS THE LENGTH OF THE JUMP (DEET IF CENTERED, DEET/2 IF FWD).
*     FPS IS A TIME FILTER FOR PS.
*     IFDIFF...1=FOREWARD TIMESTEP,  0=CENTERED TIMESTEP.

      COMPLEX PS(1),PSM(1),PEET(LA,1),PEEBAR(LA,1)

      DIMENSION AMBDA(1),SF(1)

      COMPLEX PSP,TEMP(15)
      REAL MI,MII,MIIM1,M1,M2,MP,MPM1,MM,MW,M2M1
      COMMON / NEWMAT / MI(IM,IM),MII(IM,IM),MIIM1(IM,IM),
     *                  M1(IM,IM),M2(IM,IM),MP(IM,IM),MPM1(IM,IM),
     *                  MM(IM,IM),MW(IM,IM),MPI(IM,IM,IRM),M2M1(IM,IM)

      ILEVM=ILEV-1

      DO 20 M=1,LM
      MR=(M-1)*LRS
      DO 20 N=1,LRS
      IL=MR+N

      PSP=0.0
      DO 10 IH=1,ILEV
   10 PSP=PSP+(PEET(IL,IH)-PEEBAR(IL,IH))*M2M1(IH,ILEV)
      PSP = PSM(IL) +2.*PSP
      PS(IL)=PS(IL)+FPS*(PSP+PSM(IL)-2.*PS(IL))
      IF(IFDIFF.EQ.0) PSM(IL)=PS(IL)
      PS(IL)=PSP

   20 CONTINUE

      RETURN
      END
```

```
      SUBROUTINE ORDLEG(SX,COA,IR)

      * THIS ROUTINE IS A SUBSET OF BELOUSOVS ALGORITHM
      * USED TO CALCULATE ORDINARY LEGENDRE POLYNOMIALS.

      *  SX = LEGENDRE POLYNOMIAL EVALUATED AT COA
      * COA = COSINE OF COLATITUDE
      *  IR = WAVE NUMBER


      PI = 3.1415926535898
      SQR2=SQRT(2.)
      IRPP = IR + 1
      IRPPM = IRPP - 1
      DELTA =   ACOS(COA)
      SIA =  SIN(DELTA)

      THETA=DELTA
      C1=SQR2

      DO 20 N=1,IRPPM
      FN=FLOAT(N)
      FN2=2.*FN
      FN2SQ=FN2*FN2
      C1=C1* SQRT(1.0-1.0/FN2SQ)
   20 CONTINUE

      N=IRPPM
      ANG=FN*THETA
      S1=0.0
      C4=1.0
      A=-1.0
      B=0.0
      N1=N+1

      DO 27 KK=1,N1,2
      K=KK-1
      IF (K.EQ.N) C4=0.5*C4
      S1=S1+C4* COS(ANG)
      A=A+2.0
      B=B+1.0
      FK=FLOAT(K)
      ANG=THETA*(FN-FK-2.0)
      C4=(A*(FN-B+1.0)/(B*(FN2-A)))*C4
   27 CONTINUE

      SX=S1*C1

      RETURN
      END
```

```
      SUBROUTINE PBLMAT(RGAS,RGOCP,S,SH,SF,SHF,AMBDA,TMEAN,GRAV,
     *                   GAMMA,ILEV)
      PARAMETER IL=5, I1=IL+1, IRS=25, IRM=2*IRS+1

      THIS ROUTINE SETS UP THE MATRICIES TO BE USED IN THE
      VERTICAL DIFFERENCING SCHEME

      IL IS THE NUMBER OF LEVELS - SAME AS $ILV IN NRL(MODIFIED)
      CANADIAN MODEL

      REAL S(IL),SH(IL),SF(IL),SHF(IL),AMBDA(IL),TMEAN(I1)
      REAL MI,MII,MIIM1,M1,M2,MP,MPM1,M,MW,MPI,M2M1
      COMMON / NEWMAT / MI(IL,IL),MII(IL,IL),MIIM1(IL,IL),
     *                  M1(IL,IL),M2(IL,IL),MP(IL,IL),MPM1(IL,IL),
     *                  M(IL,IL),MW(IL,IL),MPI(IL,IL,IRM),
     *                  M2M1(IL,IL)

      ILM=IL-1
      CN=S(IL)**(RGAS*GAMMA/GRAV)
      CN = 1. / CN
      DO 12 K = 1, IRM
      DO 3 J = 1, IL
      DO 3 I = 1, IL
      MPI(I,J,K) = 0.0
      MW(I,J)=0.0
      M(I,J)=0.0
      MPM1(I,J)=0.0
      MP(I,J)=0.0
      M2(I,J)=0.0
      M1(I,J)=0.0
      MIIM1(I,J) = 0.0
      MII(I,J) = 0.0
    3 MI(I,J) = 0.0
   12 CONTINUE

      SET UP THE MATRICIES

      DO 2 I = 1, ILM
      MII(I+1,I) = 1.0
    2 MII(I,I) = 1.0
      MII(IL,IL) = CN + 1.
      CALL MTXINV(MIIM1,MII,IL)
      DO 5 I = 1, ILM
      MI(I+1,I)=1./SF(I)
    5 MI(I,I) = (-1.) / SF(I)
      MI(IL,IL) = (-1.) / SF(IL)
      CALL MATMLT(MP,MI,MIIM1,IL)
      CONST=2./RGAS
      DO 4 J=1,IL
      DO 4 I=1,IL
    4 MP(I,J)=CONST*MP(I,J)
```

```
      CALL MTXINV(MPM1,MP,IL)
      DO 7 I=2,IL
      M1(I,I) = (-1.) / SHF(I-1)
 7    M1(I-1,I) = 1. / SHF(I-1)
      M1(1,1) = (-1.) / SH(1)
      CALL MTXINV(MW,M1,IL)
      DO 6 J = 1, IL
      DO 6 I = 1, IL
      IF(I.EQ.1.AND.J.EQ.1) GO TO 6
      MW(I,J) = S(J) * MW(I,J)
 6    CONTINUE
      DO 13 J=1,IL
      DO 13 I=1,IL
      IF(ABS(MW(I,J)).LT.1.E-07) MW(I,J)=.0
13    CONTINUE
      DO 8 I = 1, IL
 8    M2(I,I) = 0.5 * AMBDA(I)
      DO 18 I = 2, IL
18    M2(I-1,I) = 0.5 * AMBDA(I-1)
      DO 9 I=2,ILM
 9    M2(IL,I) = M2(IL,I)
     *          - 0.5 * (SH(I)*AMBDA(I)+SH(I-1)*AMBDA(I-1))
     *          + RGOCP * TMEAN(I)
      M2(IL,1) = M2(IL,1) - 0.5*SH(1)*AMBDA(1) + RGOCP*TMEAN(1)
      M2(IL,IL) = M2(IL,IL)
     * -0.5*(SH(ILM)*AMBDA(ILM)+AMBDA(IL))
     *          + RGOCP * TMEAN(IL)
      CALL MATMLT(M1,M2,MPM1,IL)
      DO 14 I=1,IL
14    M1(I,I)=M1(I,I)-RGAS*TMEAN(I)
      CALL MTXINV(M2M1,M1,IL)
      DO 10 I = 1, IL
      SUM = 0.0
      DO 11 J = 1, IL
11    SUM = SUM + TMEAN(J) * MP(J,I)
      SUM = SUM * RGAS
10    M2(IL,I) = M2(IL,I) - SUM
      CALL MATMLT(M1,MW,M2,IL)
      CALL MATMLT(M,M1,MPM1,IL)
      END
```

```
      SUBROUTINE PCOF2(SC,LR,LM,NPR,MPR)

      * SC(LR,LM) CONTAINS COMPLEX SPECTRAL COEFFICIENTS.
      * THIS SUBROUTINE PRINTS NPR COEFF FROM EACH OF THE FIRST
      *  MPR ROWS OF SC.

      COMPLEX SC(LR,LM)

      NLIM=NPR
      IF(NLIM.GT.LR) NLIM=LR
      MLIM=MPR
      IF(MLIM.GT.LM) MLIM=LM
      WRITE(6,6010) LR,LM,NLIM,MLIM

      DO 30 M=1,MLIM
      MS=M-1
      WRITE(6,6020) MS
      WRITE(6,6030) ( SC(N,M),N=1,NLIM)
   30 CONTINUE

      RETURN

 6010 FORMAT(1H1/ 14H COMPLEX ARRAY,2I4,4X,5HPRINT,2I4/)
 6020 FORMAT(3H M=,I2)
 6030 FORMAT((1H ,4(4X,1PE13.6,1H,,1PE13.6)))
      END
```

```
      SUBROUTINE PCPADJ(PRECIP,STBROW,ILG1 ,ESG,TG,OMEGAG,PRESSG,ILM,
     1             THEANH,SH,STAWS,ILEV,LEVS,LTBS,NUPS,NSUPS,KOUNT,DEL)

       * CALCULATION OF BOTH SMALL AND LARGE SCALE PRECIPITATION,
       * CALCULATION OF CONVECTIVE HEAT AND MOISTURE FLUXES
       *          THROUGH A CONVECTIVE ADJUSTMENT PROCEDURE,
      * CALCULATION, EVENTUALLY, OF PRECIPITATION EVAPORATION
      * FOR ILONG POINTS ABOUT ONE GAUSSIAN LATITUDE CIRCLE.

      PARAMETER $ILV=15,$LEV=0,$LV=10,$ILT=52,$ILG=64,$ILTH=26,$IR=20

      DIMENSION PRECIP(1),STBROW(1)
      DIMENSION ESG(ILM,1),TG(ILM,1),OMEGAG(ILM,1),PRESSG(1)
      DIMENSION THEANH(1),SH(1),STAWS(1),DEL(1)

      DIMENSION VERTT($ILV),VERTQ($ILV),PMB($ILV)

      ILONG=ILG1-1
      ILEVM=ILEV-1
      ILEVP=ILEV+1

      DO 600 IK=1,ILONG

      * VERTT = VERTICAL COLUMN OF TEMP AT EVEN LEVELS.
      * VERTQ = VERTICAL COLUMN OF SPHM AT EVEN LEVELS
      *   PMB = VERTICAL COLUMN OF PRESSURE(MB) AT EVEN LEVELS

      DO 12 IH=1,LTBS
   12 VERTQ(IH)=1.E-8
      DO 13 IH=1,ILEV
   13 VERTT(IH)=TG(IK,IH)+THEANH(IH)
      DO 14 IH=1,ILEV
   14 PMB(IH)=SH(IH)*PRESSG(IK)/100.
      DO 17 IH=1,LEVS
      TD=VERTT(IH+LTBS)-ESG(IK,IH)
   17 VERTQ(IH+LTBS)=SPCHUM(TD,PMB(IH+LTBS))
      LREP=ILEV/2+1
      W=OMEGAG(IK,LREP)
      PRESG=PRESSG(IK)
      NSUPX=NSUPS
      CALL CONADJ(VERTT,VERTQ,PCP,PRESG,W,PMB,STAWS,NUPS,NSUPS,
     1            ILEVP,DEL)
      IF(KOUNT.EQ.0) PRECIP(IK)=0.
      IF(KOUNT.EQ.1) PRECIP(IK)=PCP
      IF(KOUNT.GT.1) PRECIP(IK)=PRECIP(IK)+PCP*.5
      IF(KOUNT.EQ.0) STBROW(IK)=FLOAT(NSUPS-NSUPX)
      IF(KOUNT.GT.0) STBROW(IK)=STBROW(IK)+FLOAT(NSUPS-NSUPX)

      * RESTORE CORRECTED VERTICAL COLUMNS OF TEMP AND ES.

      DO 40 IH=1,ILEV
```

```
40 TG(IK,IH) = VERTT(IH)-THEANH(IH)
   DO 42 IH=1,LEVS
42 ESG(IK,IH)=VERTT(IH+LTBS)-DEWPNT(VERTQ(IH+LTBS),PMB(IH+LTBS))

600 CONTINUE
    RETURN
    END
```

```
      SUBROUTINE PERM(WORKA,WORKB,AA,BB,KP6,K3)
*     CALCULATES SETUP FIELD FOR 3 TIMES POWER OF 2 TRANSFORM
      COMPLEX WORKA(1),WORKB(1),AA,BB

      PI = 3.14159265358979
      FACT = 2.*PI/3.
      AA = CMPLX(COS(FACT),SIN(FACT))
      BB = CONJG(AA)
      FT = FACT/FLOAT(K3)
      DO 125 K=1,KP6
      FK = FT*FLOAT(K-1)
      WORKA(K) = CMPLX(COS(FK),SIN(FK))
  125 WORKB(K) = CONJG(WORKA(K))
      RETURN
      END
```

```
      SUBROUTINE PHSCON(SEACON,RK,CEEACH,RKL,BETA,S,DEL,ILEV)

* CALCULATES CONSTANTS FOR SPECTRAL MODEL PHYSICAL EFFECTS.

      DIMENSION RK(1),S(1),DEL(1)

      COMMON/PARAMS/WW,TW,A,ASQ,GRAV,RGAS,RGOCP,RGOASQ,CPRES

* CONSTANTS FOR SEA SURFACE FLUX

      SEACON = GRAV*CEEACH*BETA*A/(RGAS*DEL(ILEV))

* CONSTANTS FOR EDDY VERTICAL MOMENTUM FLUX CALCULATIONS

       ILEVM = ILEV-1
      TZIP = 273.
      DO 20 IH=1,ILEVM
      ST = (S(IH)+S(IH+1))*.5
      RK(IH) = (RKL/DEL(IH))*(GRAV*ST/(RGAS*TZIP))**2
   20 CONTINUE

      RETURN
      END
```

```
      SUBROUTINE POUT(P,C,T,PHI,PRESS,PHIS,ES,LA,LRS,LRA,LM,ILEV,LEVS,
     1               PEE,PS,PEEMN,PSMN,TMEAN,TMEANH,RGAS,IPR,LC,KSYM,
     2               GLL,ILG1,ILATH,COA,KHEM,ALP,EPSI,LALP,WRKS,WRKL)

      * CONTROLS PRINTED OUTPUT FOR SPECTRAL MULTILEVEL MODEL.
      * EACH VARIABLE IS PASSED TO SUBR POUTF FOR OUTPUT.
      * LC CONTAINS THE CONTROL PARAMETERS FOR OUTPUT.

      COMPLEX P(LA,1),C(LA,1),T(LA,1),PHI(LA,1),PRESS(1),PHIS(1)
      COMPLEX ES(LA,1),PEE(LA,1),PS(1)
      DIMENSION GLL(1),ALP(1),EPSI(1),WRKL(1)

      DIMENSION PEEMN(1),TMEAN(1),TMEANH(1)
      DIMENSION WRKS(1),COA(1)
      DIMENSION LC(3,ILEV,1)
      COMPLEX TDZZ

      COMMON/TIMES/ DEET,KOUNT,KSTART,KTOTAL,IFDIFF

      * ALWAYS OUTPUT INITIAL AND FINAL TIMES. OTHERWISE EVERY IPR TIMES

      IF(KOUNT.EQ.KSTART.OR.KOUNT.EQ.KTOTAL) GO TO 82
      IF(IPR.EQ.0) RETURN
      IF(MOD((KOUNT-KSTART),IPR).NE.0) RETURN
   82 KT=1
      IF(KOUNT.GT.KSTART) KT=2+(KOUNT-KSTART)/(KTOTAL-KSTART)

      * VORTICITY, LEVELS 1 TO ILEV.

      DO 100 L=1,ILEV
      KO=LC(KT,L,1)
      IF(KO.GT.0) CALL POLTF(1,KO,KT,KOUNT,  P(1,L),LRA,LM,L,-KSYM,
     1               GLL,ILG1,ILATH,COA,KHEM,ALP,EPSI,LALP,WRKS,WRKL)
  100 CONTINUE

      * DIVERGENCE, LEVELS 1 TO ILEV.

      DO 200 L=1,ILEV
      KO=LC(KT,L,2)
      IF(KO.GT.0) CALL POLTF(2,KO,KT,KOUNT,  C(1,L),LRS,LM,L, KSYM,
     1               GLL,ILG1,ILATH,COA,KHEM,ALP,EPSI,LALP,WRKS,WRKL)
  200 CONTINUE

      * TEMPERATURE, LEVELS 1 TO ILEV.
      * NOTE - T CONTAINS THE DEVIATION ONLY, SO MUST ADD TMEANH(L).

      DO 300 L=1,ILEV
      KO=LC(KT,L,3)
      TDZZ=T(1,L)
      T(1,L)=T(1,L)+TMEANH(L)*SQRT(2.0)
```

```
      IF(K0.GT.0) CALL POLTF(3,K0,KT,KOUNT,  T(1,L),LRS,LM,L, KSYM,
     1             GLL,ILG1,ILATH,COA,KHEM,ALP,EPSI,LALP,WRKS,WRKL)
      T(1,L)=TDZZ
  300 CONTINUE

      * GEOPOTENTIALS, LEVELS 1 TO ILEV.
      * FIRST CONVERT PEE TO GZ AT ALL LEVELS.

      CALL GZFBP (PHI,PEE,PS,TMEAN,PEEMN,PSMN,LA,LRS,LM,ILEV,RGAS)
      DO 400 L=1,ILEV
      K0=LC(KT,L,4)
      IF(K0.GT.0) CALL POLTF(4,K0,KT,KOUNT,PHI(1,L),LRS,LM,L, KSYM,
     1             GLL,ILG1,ILATH,COA,KHEM,ALP,EPSI,LALP,WRKS,WRKL)
  400 CONTINUE

      * SURFACE PRESSURE.

      K0=LC(KT,1,5)
      IF(K0.GT.0) CALL POLTF(5,K0,KT,KOUNT,PRESS,LRS,LM,1, KSYM,
     1             GLL,ILG1,ILATH,COA,KHEM,ALP,EPSI,LALP,WRKS,WRKL)

      * MOUNTAIN FIELD.

      K0=LC(KT,1,6)
      IF(K0.GT.0) CALL POLTF(6,K0,KT,KOUNT, PHIS,LRS,LM,1, KSYM,
     1             GLL,ILG1,ILATH,COA,KHEM,ALP,EPSI,LALP,WRKS,WRKL)

      * DEW POINT DEPRESSION, LEVS LEVELS ENDING AT ILEV.

      IF(LEVS.EQ.0) RETURN
      DO 700 N=1,LEVS
      L=ILEV-LEVS+N
      K0=LC(KT,N,7)
      IF(K0.GT.0) CALL POLTF(7,K0,KT,KOUNT, ES(1,N),LRS,LM,L, KSYM,
     1             GLL,ILG1,ILATH,COA,KHEM,ALP,EPSI,LALP,WRKS,WRKL)
  700 CONTINUE

      RETURN
      END
```

```
      SUBROUTINE POUTF(NV,K0,KT,KOUNT,SC,LR,LM,L,KSM,
     1                 GLL,ILG1,ILATH,COA,KHEM,ALP,EPSI,LALP,WRKS,WRKL)

C     * PRINTS SPECTRAL COEFF, MAPS GAUSSIAN GRID GLL(ILG1,NLAT),
C     *  OR MAPS POLAR STEREOGRAPHIC GRID GLL(NI,NJ)
C     * FOR COMPLEX COEFFICIENTS IN SC(LR,LM).
C     * GLL IS AN LCM WORK FIELD FOR MAPS (NI*NJ OR ILG1*NLAT).
C     * COA(ILATH) CONTAINS THE COSINES OF THE COLATITUDE (N TO S).
C     * ALP(LALP,LM) IS A WORK FIELD FOR LEGENDRE POLYNOMIALS.
C     * EPSI IS A FIELD OF CONSTANTS THE SAME SIZE AS ALP.

      COMPLEX SC(1)
      DIMENSION GLL(1),ALP(1),EPSI(1),WRKL(1)
      DIMENSION WRKS(1),COA(1)

      COMMON/PCOM1/ LRPR,LMPR,KGGM(5,3),KPSM(5,3),D60,DGRW,NHEM
      COMMON/PCOM2/ LAB(10),CS(2,10)

      CINT=CS(1,NV)
      SCAL=CS(2,NV)

C     * IF THE 4-BIT OF K0 IS 1, PRINT THE SPECTRAL COEFFICIENTS.
C     * LRPR COEFF ARE PRINTED FROM THE FIRST LMPR ROWS OF SC.

      IF(K0.LT.4) GO TO 24
      CALL PCOF2(SC,LR,LM,LRPR,LMPR)
      WRITE(6,6010) LAB(NV),L,KOUNT

C     * IF THE 2-BIT OF K0 IS 1, PREPARE THE GAUSSIAN GRID.
C     * IF KHEM=0 THE MODEL IS GLOBAL AND GG=(ILG1,ILATH*2).
C     *  OTHERWISE THE MODEL IS HEMISPHERIC AND GG=(ILG1,ILATH).
C     * THE WINDOW MAPPED IS LL BY MM WITH LOWER LEFT CORNER (IW,JW).

   24 IF(MOD(K0,4).LE.1) GO TO 28
      NLAT=ILATH
      IF(KHEM.EQ.0) NLAT=ILATH*2
      CALL SPAGG2(GLL,ILG1,ILATH,COA,KHEM,SC,LR,LM,KSM,
     1                        ALP,EPSI,LALP,WRKS,WRKL)
      IW=KGGM(1,KT)
      JW=KGGM(2,KT)
      LL=KGGM(3,KT)
      MM=KGGM(4,KT)
      CALL FCONW2(GLL,CINT,SCAL,ILG1,NLAT,IW,JW,LL,MM,KGGM(5,KT))
      WRITE(6,6010) LAB(NV),L,KOUNT

C     * IF THE 1-BIT OF K0 IS 1, PREPARE THE POLAR STEREOGRAPHIC GRID.
C     * THE GRID IS (NI,NJ) WITH POLE AT (IP,JP), GRID SIZE D60 M.
C     *  AT 60 DEG. AND GREENWICH IS ORIENTED AT DGRW DEGREES.
C     * NHEM=1 COMPUTES THE N HEM, NHEM=2 THE S HEM.

   28 IF(MOD(K0,2).NE.1) RETURN
```

```
      NI=KPSM(1,KT)
      NJ=KPSM(2,KT)
      IP=KPSM(3,KT)
      JP=KPSM(4,KT)
      CALL SPAPS2(GLL,NI,NJ,IP,JP,D60,DGRW,NHEM,SC,LR,LM,KSM,
     1               ALP,EPSI,LALP,WRKS,WRKL)
      CALL FCONW2(GLL,CINT,SCAL,NI,NJ,1,1,NI,NJ,KPSM(5,KT))
      WRITE(6,6010) LAB(NV),L,KOUNT

      RETURN
 6010 FORMAT(1H0,A4,7H LEVEL ,I2,10H  AT STEP ,I4)
      END
```

```
      SUBROUTINE PTS6(NF1,NF2,GZ,T,U,V,ES,GZS,SP,GLL,LG,
     1                LV,PR,NK,SIG,ILG,ILAT,KHEM,WET,KDR,KWT,KCV)

C     * INTERPOLATES NK SIGMA LEVELS FROM LV PRESSURE LEVELS
C     *  OF GAUSSIAN GRIDS (ILG1,NLAT).

C     * IF KHEM=0 THE MODEL IS GLOBAL AND NLAT=ILAT.
C     * IF KHEM=1,2 THE MODEL IS HEMISPHERIC AND NLAT=ILAT/2

C     * LEVELS ARE NUMBERED FROM THE TOP DOWN (AS IN ARRAY PR).
C     * MAXIMUM NUMBER OF LEVELS - PRESSURE=15, SIGMA=15.

C     * INPUT  IS ON SEQUENTIAL FILE NF1 IN THE FOLLOWING ORDER....
C     *      GZS,GZ(LV),T(LV),((U,V)(LV)),ES(LV),(DR,WT,CV)
C     * OUTPUT IS ON SEQUENTIAL FILE NF2 IN THE FOLLOWING ORDER....
C     *      GZS,LNSP,GZ(NK),((U,V)(NK)),ES(NK),(DR,WT,CV)

C     * CORE MAY BE SHARED BY (GZ,U,ES),(T,V).
C     * GLL IS A WORK FIELD OF LA WORDS IN LCM.

      PARAMETER $ILV=15,$LEV=0,$LV=10,$ILT=52,$ILG=64,$ILTH=26,$IR=20
      PARAMETER $ILP1=$ILV+1

      DIMENSION GZ(LG,1),T(LG,1),U(LG,1),V(LG,1),ES(LG,1)
      DIMENSION GZS(1),SP(1),GLL(1)

      LOGICAL WET,KDR,KWT,KCV
      DIMENSION PR(1),SIG(1)

C     * SCM WORK ARRAYS.

      DIMENSION ZL($ILV),TL($ILV),UL($ILV),VL($ILV),SL($ILV),TUL($ILV),
     *TVL($ILV),TSL($ILV),SG($ILV),SUL($ILV),SVL($ILV),SSL($ILV),
     *SGE($ILV),SGZL($ILV),S($ILP1),WRKS(130)

C     * OUTPUT PACKING CONTROLS.

      COMMON/PKCOM/ NPGZS,NPSP,NPGZ,NPW,NPES,NPWT,NPCV,NPDR

      LOGICAL OK
      DATA GRAV,RGAS/ 9.80616, 287. /

C     * SET CONSTANTS.

      ID=0
      ACC=.1*GRAV
      RGASIN=1./RGAS
      NLAT=ILAT/2
      IF(KHEM.EQ.0) NLAT=ILAT
      ILG1=ILG+1
      LA=ILG1*NLAT
```

```
      LV1=LV-1

      * CONVERT PR TO LOG(PRESSURE).

      DO  2 L=1,LV
    2 PR(L) = ALOG(PR(L))

      DLNP = PR(LV)-PR(LV-1)

      * SET SG TO LOG(SIGMA).

       DO 4 K=1,NK
    4 SG(K) = ALOG(SIG(K))

      * AVERAGE SG TO EVEN LEVELS IN SGE.

      NKM1 = NK-1
      DO 5 K=1,NKM1
    5 SGE(K)= .5 * (SG(K)+SG(K+1))
      SGE(NK)=.5*SG(NK)

      DO 6 L=1,LV1
    6 S(L)=1./(PR(L+1)-PR(L))

      * READ MOUNTAIN FIELD (M/SEC)**2.
      * READ GEOPOTENTIALS (M/SEC)**2 AND TEMPERATURES (DEG K).
      * GZ AND T MAY BE IN ANY ORDER IN NF1.

      CALL RSGGP(GZS,ILG1,NLAT,NF1,ID,4H GZS,1,OK,GLL,WRKS)
      REWIND NF1
      DO 210 L=1,LV
  210 CALL RSGGP(GZ(1,L),ILG1,NLAT,NF1,ID,4H  GZ,L,OK,GLL,WRKS)
      REWIND NF1
      DO 220 L=1,LV
  220 CALL RSGGP( T(1,L),ILG1,NLAT,NF1,ID,4H   T,L,OK,GLL,WRKS)

      * CALCULATE SURFACE PRESSURE AND INTERPOLATE HEIGHTS TO SIGMA SFC.

      DO 270  I=1,LA

      DO 245 L=1,LV
      ZL(L)=GZ(I,L)
      TL(L)=-RGAS*T(I,L)
  245 CONTINUE

      GZSI=GZS(I)
      GUESS= PR(LV) - GZSI/(RGAS*250.)
      CALL TERP1(GUESS,GZSI,ZL,TL,PR,ACC,LV)
      SP(I)=GUESS

      DO 250 K=1,NK
```

```
      XLNP=SP(I)+SG(K)
      RLAPSE=(TL(LV)-TL(LV-1))/ DLNP
      CALL TERP2(SGZL(K),DUMMY,XLNP,ZL,TL,PR,LV,RLAPSE)
  250 CONTINUE

      DO 255 K=1,NK
  255 GZ(I,K)=SGZL(K)

  270 CONTINUE

      * OUTPUT SURFACE GEOPOTENTIAL (M/SEC)**2.
      * OUTPUT LOG SURFACE PRESSURE (MILLIBARS).
      * OUTPUT GEOPOTENTIALS (M/SEC)**2.

      CALL WSGGP(GZS,ILG1,NLAT,NF2,ID,4H GZS,1,NPGZS,GLL,WRKS)
      CALL WSGGP( SP,ILG1,NLAT,NF2,ID,4HLNSP,1,NPSP,GLL,WRKS)
      DO 280 K=1,NK
  280 CALL WSGGP(GZ(1,K),ILG1,NLAT,NF2,ID,4H  GZ,K,NPGZ,GLL,WRKS)

      * READ WINDS (U,V)=(U,V)*COS(LAT)/(EARTH RADIUS)

      DO 310 L=1,LV
      CALL RSGGP( U(1,L),ILG1,NLAT,NF1,ID,4H   U,L,OK,GLL,WRKS)
      CALL RSGGP( V(1,L),ILG1,NLAT,NF1,ID,4H   V,L,OK,GLL,WRKS)
  310 CONTINUE

      * COMPUTE VERTICAL DERIVATIVE AND INTERPOLATE ONE COL AT A TIME.

      DO 371 I=1,LA

      DO 346 L=1,LV
      UL(L)=U(I,L)
      VL(L)=V(I,L)
  346 CONTINUE

      CALL DFDS(TUL,UL,S,LV,1.,1.)
      TUL(1)=0.
      CALL DFDS(TVL,VL,S,LV,1.,1.)
      TVL(1)=0.

      DO 351 K=1,NK
      XLNP=SP(I)+SG(K)
      RLAPSE=(TUL(LV)-TUL(LV-1))/ DLNP
      CALL TERP2(SUL(K),DUMMY,XLNP,UL,TUL,PR,LV,RLAPSE)
      RLAPSE=(TVL(LV)-TVL(LV-1)) / DLNP
      CALL TERP2(SVL(K),DUMMY,XLNP,VL,TVL,PR,LV,RLAPSE)
  351 CONTINUE

      DO 356 K=1,NK
      U(I,K)=SUL(K)
      V(I,K)=SVL(K)
```

```
356 CONTINUE

371 CONTINUE

    * OUTPUT WINDS ON FILE NF2.

    DO 381 K=1,NK
    CALL WSGGP( U(1,K),ILG1,NLAT,NF2,ID,4H   U,K,NPW ,GLL,WRKS)
    CALL WSGGP( V(1,K),ILG1,NLAT,NF2,ID,4H   V,K,NPW ,GLL,WRKS)
381 CONTINUE

    * READ DEW POINT DEPRESSION (DEG C) FROM FILE NF1.

    IF(.NOT.WET) GO TO 510
    DO 410 L=1,LV
    CALL RSGGP(ES(1,L),ILG1,NLAT,NF1,ID,4H  ES,L,0K,GLL,WRKS)
410 CONTINUE

    * COMPUTE VERTICAL DERIVATIVE AND INTERPOLATE ONE COL AT A TIME.

    DO 472 I=1,LA

    DO 447 L=1,LV
447 SL(L)=ES(I,L)*RGAS

    CALL DFDS(TSL,SL,S,LV,1.,1.)
    TSL(1)=0.

    DO 452 K=1,NK
    XLNP=SP(I)+SGE(K)
    RLAPSE=(TSL(LV)-TSL(LV-1)) / DLNP
    CALL TERP2(SSL(K),DUMMY,XLNP,SL,TSL,PR,LV,RLAPSE)
452 CONTINUE

    DO 457 K=1,NK
457 ES(I,K)=SSL(K)*RGASIN

472 CONTINUE

    * OUTPUT DEW POINT DEPRESSIONS ON FILE NF2.

    DO 482 K=1,NK
482 CALL WSGGP(ES(1,K),ILG1,NLAT,NF2,ID,4H  ES,K,NPES,GLL,WRKS)

    * TRANSFER GAUSSIAN GRID PHYSICS FROM NF1 TO NF2.

510 IF(.NOT.KDR) GO TO 520
    CALL RSGGP(SP,ILG1,NLAT,NF1,ID,4H  DR,1, 0K ,GLL,WRKS)
    CALL WSGGP(SP,ILG1,NLAT,NF2,ID,4H  DR,1,NPDR,GLL,WRKS)

520 IF(.NOT.KWT) GO TO 530
```

```
      CALL RSGGP(SP,ILG1,NLAT,NF1,ID,4H  WT,1, OK ,GLL,WRKS)
      CALL WSGGP(SP,ILG1,NLAT,NF2,ID,4H  WT,1,NPWT,GLL,WRKS)

  530 IF(.NOT.KCV) GO TO 999
      CALL RSGGP(SP,ILG1,NLAT,NF1,ID,4H  CV,1, OK ,GLL,WRKS)
      CALL WSGGP(SP,ILG1,NLAT,NF2,ID,4H  CV,1,NPCV,GLL,WRKS)

  999 RETURN
      END
```

```
      SUBROUTINE QDAW2(U,V,P,C,EPSI,PROW,CROW,
     1                 LRU,LRV,LRA,LRS,LALP,LM,KHEM,IR)

      * SPECTRAL CALCULATION OF WIND COMPONENTS U(LRU,LM), V(LRV,LM)
      *  FROM RELATIVE VORTICITY P(LRA,LM) AND DIVERGENCE C(LRS,LM).

      * IF MODEL IS HEMISPHERIC, U,C ARE SYMMETRIC,
      *                          V,P ARE ANTISYMMETRIC.

      * PROW,CROW ARE WORK ARRAYS FOR ONE UNSQUEEZED ROW OF P,C.
      *  THEY MUST HAVE A LENGTH OF IR+2 COMPLEX WORDS.

      COMPLEX U(LRU,1),V(LRV,1),P(LRA,1),C(LRS,1)
      DIMENSION EPSI(LALP,1)
      COMPLEX PROW(1),CROW(1)

      LOGICAL HEMI
      COMPLEX PR,CR,PL,CL, IZ

      HEMI=.FALSE.
      IF(KHEM.NE.0) HEMI=.TRUE.
      IRP2=IR+2
      IZ=(0.,1.)
      DO 12 K=1,IRP2
      PROW(K)=(0.,0.)
   12 CROW(K)=(0.,0.)

      DO 40 M=1,LM
      MS=M-1
      FMS=FLOAT(MS)

      * PUT UNSQUEEZED ROW OF P,C IN PROW,CROW.

      DO 14 N=1,LRA
      K=N
      IF(HEMI) K=N+N
      PROW(K)=P(N,M)
   14 CONTINUE
      DO 16 N=1,LRS
      K=N
      IF(HEMI) K=N+N-1
      CROW(K)=C(N,M)
   16 CONTINUE

      * COMPUTE ONE ROW OF U.
      * N INDEXES THE (POSSIBLY) SQUEEZED ROW OF U.
      * K INDEXES THE UNSQUEEZED ROWS OF P,C,EPSI.

      DO 20 N=1,LRU
      NS=M+N-2
      IF(HEMI) NS=NS+(N-1)
```

```
      IF(NS.EQ.0) GO TO 20
      FNS=FLOAT(NS)
      CON=FMS/(FNS*(FNS+1.))
      K=N
      IF(HEMI) K=N+N-1
      IF(K.LT.IRP2) EPSR=EPSI(K+1,M)/(FNS+1.)
      EPS =EPSI(K,M)/FNS
      PR=(0.,0.)
      IF(K.LT.IRP2) PR=PROW(K+1)
      PL=(0.,0.)
      IF(K.GT.1) PL=PROW(K-1)
      U(N,M)= - EPS*PL + EPSR*PR - IZ*CROW(K)*CON
   20 CONTINUE

      * COMPUTE ONE ROW OF V.
      * N INDEXES THE (POSSIBLY) SQUEEZED ROW OF V.
      * K INDEXES THE UNSQUEEZED ROWS OF P,C,EPSI.

      DO 30 N=1,LRV

      NS=M+N-2
      IF(HEMI) NS=NS+N
      IF(NS.EQ.0) GO TO 30
      FNS=FLOAT(NS)
      CON=FMS/(FNS*(FNS+1.))

      K=N
      IF(HEMI) K=N+N
      IF(K.LT.IRP2) EPSR=EPSI(K+1,M)/(FNS+1.)
      EPS =EPSI(K,M)/FNS
      CR=(0.,0.)
      IF(K.LT.IRP2) CR=CROW(K+1)
      CL=(0.,0.)
      IF(K.GT.1) CL=CROW(K-1)
      V(N,M)= + EPS*CL - EPSR*CR - IZ*PROW(K)*CON
   30 CONTINUE
   40 CONTINUE

      * SET THE MEAN VALUES OF U AND V.

      IF(HEMI) U(1,1)= P(1,1)*EPSI(2,1)
      IF(.NOT.HEMI) U(1,1)= P(2,1)*EPSI(2,1)
      IF(.NOT.HEMI) V(1,1)=-C(2,1)*EPSI(2,1)

      RETURN
      END
```

```
      SUBROUTINE RCOM(WORKA,WORKB,AA,BB,DATA,DATB,DATC,FIN,KP6,K3)
*     COMPLETES GRID TO FOURIER TRANSFORM FOR 3 TIMES A POWER OF 2.
      COMPLEX WORKA(1),WORKB(1),AA,BB,CC,DD
      COMPLEX DATA(1),DATB(1),DATC(1),FIN(1)

      DO 100 K=1,KP6
      CC = WORKB(K)*DATB(K)
      DD = WORKA(K)*DATC(K)
      KP = K3+2-K
      FIN(K) = DATA(K) + CC + DD
      FIN(KP) = CONJG(DATA(K) + AA*CC + BB*DD)
  100 CONTINUE

      RETURN
      END
```

```
      SUBROUTINE RHSSI(PEEBAR,PEET,SDS,LA,LRS,LM,ILEV,SF,DT)

      PARAMETER IM = 5, $IR = 25, IRM = 2*$IR+1

*     THIS ROUTINE IS REGUIRED TO IMPLEMENT THE SEMI-IMPLICIT TIME-STEP
*     METHOD.REFERENCE SHOULD BE MADE TO THE NOTES.

      * IF MODEL IS HEMISPHERIC, DI,PEET,SDS ARE SYMMETRIC.
      * EACH LEVEL IS DIMENSIONED (LRS,LM).
      * EACH LEVEL IS SEPARATED BY LA COMPLEX WORDS.
      * MAXIMUM VALUE OF ILEV IS 15.

      COMPLEX PEEBAR(LA,1),PEET(LA,1),SDS(LA,1),TEMP(IM)
      COMPLEX SUM
      DIMENSION SF(1)
      REAL MI,MII,MIIM1,M1,M2,MP,MPM1,M,MW,MPI,M2M1
      COMMON / NEWMAT / MI(IM,IM),MII(IM,IM),MIIM1(IM,IM),M1(IM,IM),
     *                  M2(IM,IM),MP(IM,IM),MPM1(IM,IM),M(IM,IM),
     *                  MW(IM,IM),MPI(IM,IM,IRM),M2M1(IM,IM)

      ILEVM=ILEV-1
      LMSG=LRS*LM

      DO 1 IL1 = 1, LRS
      DO 1 IL2 = 1, LM
      IL = (IL2-1)*LRS + IL1
      IP = IL1 + IL2 - 1
      DO 2 IH = 1, ILEV
      TEMP(IH) = 0.0
      DO 2 JH = 1, ILEV
    2 TEMP(IH) = TEMP(IH) + SDS(IL,JH)*M(JH,IH)
      DO 3 IH = 1, ILEV
    3 TEMP(IH) = PEET(IL,IH) + DT*TEMP(IH)
      DO 5 IH = 1, ILEV
      SUM = 0.0
      DO 4 JH = 1, ILEV
    4 SUM = SUM + TEMP(JH)*MPI(JH,IH,IP)
    5 PEEBAR(IL,IH) = SUM
    1 CONTINUE

      RETURN
      END
```

```
      PROGRAM RPTS6
      * CONTROL PROGRAM FOR PTS6.
      * INTERPOLATES NK SIGMA LEVELS FROM LV PRESSURE LEVELS.

      PARAMETER $ILV=15,$LEV=0,$LV=10,$ILT=52,$ILG=64,$ILTH=26,$IR=20
      PARAMETER $LAW = ($ILG+1)*$ILT

      COMMON//GZ($LAW,$ILV),T($LAW,$ILV),GZS($LAW),SP($LAW),GLL($LAW)
      DIMENSION U(1),V(1),ES(1)
      EQUIVALENCE (U,ES,GZ),(V,T)

      * SPECTRAL MODEL CONTROL PARAMETERS.

      LOGICAL MTN,WET,KDR,KWT,KCV,KDIV
      COMMON/PSCOM/LV,PR($LV),NK,SIG($ILV)
      COMMON/DTCOM/ IDATIM(14)

      * OUTPUT PACKING CONTROLS.

      COMMON/PKCOM/ NPGZS,NPSP,NPGZ,NPW,NPES,NPWT,NPCV,NPDR

      DATA NF1,NF2,NPGZS,NPSP,NPGZ,NPW,NPES,NPWT,NPCV,NPDR
     1 /   42, 52,     1,   1,   1,  1,   1,   1,   1,   1 /

      * READ CONTROL LABEL FROM FILE NF1 AND COPY TO FILE NF2.
      CALL INDUMP

      CALL SPLAB(NF1,1,LV,PR,NK,SIG,LEVS,ILG,ILAT,KHEM,IR,DEET,
     1      MTN,WET,KDR,KWT,KCV,KDIV,KSTART,KTOTAL,IPRG,IPCP,IDATIM)
      IF(LV.EQ.0) STOP
      CALL SPLAB(NF2,2,LV,PR,NK,SIG,LEVS,ILG,ILAT,KHEM,IR,DEET,
     1      MTN,WET,KDR,KWT,KCV,KDIV,KSTART,KTOTAL,IPRG,IPCP,IDATIM)

      * PERFORM THE VERTICAL INTERPOLATION.
      * LG INDEXES THE MULTILEVEL ARRAYS INSIDE PTS5.

      NLAT=ILAT/2
      IF(KHEM.EQ.0) NLAT=ILAT
      LG=NLAT*(ILG+1)
      CALL PTS6(NF1,NF2,GZ,T,U,V,ES,GZS,SP,GLL,LG,
     1              LV,PR,NK,SIG,ILG,ILAT,KHEM,WET,KDR,KWT,KCV)

      WRITE(6,6090) (IDATIM(I),I=7,13)
      STOP

 6090 FORMAT(1H0,7A4,6X,19H END SPECTRAL RPTS6)
      END
```

```
      SUBROUTINE RSGGP(GG,NLG,NLAT,NF,ID,NAME,N,OK,GGP,WRKS)

      * READS ARRAY GG(NLG,NLAT) FROM SEQUENTIAL FILE NF
      *  PRECEDED BY AN IDENTIFYING LABEL OF 7 WORDS.
      *  WRITTEN BY SUBROUTINE WSGGP.
      * ID = IDENTIFICATION NUMBER FOR THE RECORD.
      * NAME = ALPHANUMERIC LABEL FOR THE FIELD.
      *  N   = NUMERIC LABEL FOR THE FIELD.
      * OK = .T. OR .F. ACCORDING TO WHETHER FIELD IS FOUND OR NOT.
      * GGP,WRKS ARE WORK FIELDS USED ONLY IF FIELD IS PACKED.
      * GGP = NLAT*((NLG-1)/NPACK+2) WORDS IN LCM.
      * WRKS = NLG WORDS IN SCM.

      DIMENSION GG(1),GGP(1),WRKS(1)
      LOGICAL OK
      DIMENSION LAB(7)
      DATA KFLD/ 4HGRID /

      OK=.TRUE.
      IF(NF.LE.0) RETURN

      * READ LABEL OF THE NEXT RECORD ON FILE NF.

   15 READ(NF,END=16) LAB
      GO TO 17

      * IF LABEL IS NOT FOUND BEFORE END-OF-FILE IS ENCOUNTERED
      *  SET OK=.FALSE., REWIND THE FILE, AND RETURN.

   16 OK=.FALSE.
      REWIND NF
      WRITE(6,610) NF,ID,NAME,N,NLG,NLAT
      RETURN

      * IF LABEL NOT CORRECT GO BACK AND READ THE NEXT RECORD.

   17 IF(KFLD.NE.LAB(1)) GO TO 15
      IF(  ID.NE.LAB(2)) GO TO 15
      IF(NAME.NE.LAB(3)) GO TO 15
      IF(   N.NE.LAB(4)) GO TO 15
      IF( NLG.NE.LAB(5)) GO TO 15
      IF(NLAT.NE.LAB(6)) GO TO 15

      * IF LABEL IS CORRECT, BACKSPACE TO PREPARE FOR READING THE FIELD.
      * GET THE PACKING DENSITY FROM THE LABEL.

      BACKSPACE NF
      NPACK=LAB(7)

      * IF THERE IS NO PACKING READ GG DIRECTLY AND RETURN.
```

```
      IF(NPACK.GT.1) GO TO 25
      LA=NLG*NLAT
      READ(NF) LAB,(GG(I),I=1,LA)
      WRITE(6,620) ID,NAME,N,NLG,NLAT,NPACK,NF
      RETURN

      AN ERROR HAS BEEN ENCOUNTERED IN THE LABEL FIELD

   25 CALL FPAK
   99 RETURN


  610 FORMAT(1H ,13H..EOF ON FILE,I3,18H  LOOKING FOR GRID,I6,A4,3I5)
  620 FORMAT(1H ,I5,2X,A4,I3,6H  GRID,I5,I4,8H  NPACK=,I2,
     1        16H  READ FROM FILE,I3)
      END
```

```
      PROGRAM RSPW6

*     CONTROL PROGRAM FOR MULTILEVEL SPECTRAL MODEL - VERSION 6.

      PARAMETER $ILV=15,$LEV=0,$LV=10,$ILT=52,$ILG=64,$ILTH=26,$IR=20

      LOGICAL MTN,WET,KDR,KWT,KCV,KDIV
      COMMON/SWCOM/ MTN,WET,KDR,KWT,KCV,KDIV
      DIMENSION PR($LV),SIG($ILV),IDATIM(14)

      COMMON/TIMES/ DEET,KOUNT,KSTART,KTOTAL,IFDIFF
      COMMON/PARAM2/ BETA,AVERT,DIVCH,FP,FC,FPEE,FPS,FS
      COMMON/PARAM3/ DIFUSD,DIFUSS,VIFUSS
      COMMON/PARAM4/ CEEACH,RKL
      COMMON/ADJPCP/HC,HF,HM,AA,DEPTH,LHEAT,MOIADJ,MOIFLX

      DIMENSION LC(3,$ILV,10)
      COMMON/PCOM1/ LRPR,LMPR,KGGM(5,3),KPSM(5,3),D60,DGRW,NHEM
      COMMON/PCOM2/ LAB(10),CS(2,10)
      COMMON/MAPGG/ MGGDR,MGGWT,MGGCV
      COMMON/PKOUT/ NPGZ,NPSP,NPW,NPES
      COMMON/EGRAPH/ ETOTS(181,6)

*     NF1 = INPUT FILE,  NF2 = OUTPUT FILE,  NFPCP = PRECIP OUTPUT.

      DATA NF1,NF2,NFPCP/ 52, 72, 74 /

*     MODEL PARAMETERS.

      DATA IFDIFF/ 1 /
      DATA   BETA, DIVCH, AVERT,  FP,  FC,  FPEE,  FPS,  FS
     1  /    1.,  2.0E-5, .25, .05, .05, .05, .05, .05 /
      DATA CEEACH,RKL/ 1.2E-3, 20. /
      DATA DIFUSD,DIFUSS,VIFUSS/1.E-5,1.E+5, 0.0 /
      DATA HC,HF,HM/ 1.0,0.8,0.8 /
      DATA LHEAT,MOIADJ,MOIFLX/ 1 , 1 , 1 /

*     PRINTER OUTPUT CONTROLS.

      DATA MGGDR,MGGWT,MGGCV/ 0, 0, 0 /
      DATA NVAR,LTOT/ 7,181 /
      DATA LRPR,LMPR / 12,13 /
      DATA IGE,IGM,IGG,IGD/ 0,0,0,1 /

      DATA NPGZ,NPSP,NPW,NPES,NPPCP/ 1,1,1,1,1 /

*     READ CONTROL LABEL FROM FILE NF1 AND COPY TO FILES NF2,NFPCP.

      CALL INDUMP
      CALL SPLAB(NF1,1,LV,PF,NK,SIG,LEVS,ILG,ILAT,KHEM,IR,DEET,
     1     MTN,WET,KDR,KWT,KCV,KDIV,KSTART,KTOTAL,IPRG,IPCP,IDATIM)
```

```
      IF(LV.EQ.0) STOP
      IF(.NOT.WET) LEVS=0
      IF(WET.AND.LEVS.EQ.0) WET=.FALSE.
      CALL SPLAB(NF2,2,LV,PR,NK,SIG,LEVS,ILG,ILAT,KHEM,IR,DEET,
     1     MTN,WET,KDR,KWT,KCV,KDIV,KSTART,KTOTAL,IPRG,IPCP,IDATIM)
      CALL SPLAB(NFPCP,2,LV,PR,NK,SIG,LEVS,ILG,ILAT,KHEM,IR,DEET,
     1     MTN,WET,KDR,KWT,KCV,KDIV,KSTART,KTOTAL,IPRG,IPCP,IDATIM)
      ILEV=NK

      * READ MODEL I/O CONTROL PARAMETERS.

      READ(5,5010) IPR,IEPR,NOGR,NTEST
      CALL INPOC(LC,ILEV,NVAR)

      * PERFORM THE FORECAST.

      CALL SPW6(ILG,ILAT,KHEM,IR,ILEV,LEVS,SIG,
     1                  NF1,NF2,NFPCP,LC,ETOTS,LTOT,IEPR,NOGR,IPR,
     2                        NTEST,IPRG,IPCP,NPFCP,IDATIM)

      * DRAW ENERGY GRAPHS IF REQUESTED.

      IF(NOGR.EQ.0.OR.KOUNT.LT.NOGR) GO TO 99
      CALL EGRAFS(ETOTS,LTOT,KOUNT+1,IGE,IGM,IGQ,IGD)

   99 WRITE(6,6090) (IDATIM(I),I=7,13)
      STOP

 5010 FORMAT(6I3)
 6005 FORMAT(1H1//32H P.E. SPECTRAL MODEL - VERSION 6)
 6090 FORMAT(1H0,7A4,6X,19H END SPECTRAL RSPW6)
      END
```

```
      PROGRAM RSTP6

      * CONTROL PROGRAM FOR STP6.
      * CONVERTS NK SIGMA LEVELS OF GAUSSIAN GRIDS TO LV PRESSURE LEVELS

      PARAMETER $ILV=15,$LEV=0,$LV=10,$ILT=52,$ILG=64,$ILTH=26,$IR=20
      PARAMETER $LAW = ($ILG+1)*$ILT

      COMMON//GZ($LAW,$ILV),T($LAW,$ILV)
      DIMENSION U(1),V(1),ES(1)
      EQUIVALENCE (U,ES,GZ),(V,T)

      LOGICAL MTN,WET,KDR,KWT,KCV,KDIV
      DIMENSION PR($LV),SIG($ILV),IDATIM(14)
      DIMENSION MPGZ($LV),MPT($LV),MPW($LV),MPWSP($LV),MPES($LV)

      DATA NF1,NF12,NF2/ 72,42,82 /
      DATA NPSP,NPGZ,NPT,NPW,NPES/ 1,1,1,1,1 /

      * READ CONTROL LABEL FROM FILE NF1 AND COPY TO FILE NF2.

      CALL INDUMP
      CALL SPLAB(NF1,1,LV,PR,NK,SIG,LEVS,ILG,ILAT,KHEM,IR,DEET,
     1       MTN,WET,KDR,KWT,KCV,KDIV,KSTART,KTOTAL,IPRG,IPCP,IDATIM)
      IF(LV.EQ.0) STOP
      CALL SPLAB(NF2,2,LV,PR,NK,SIG,LEVS,ILG,ILAT,KHEM,IR,DEET,
     1       MTN,WET,KDR,KWT,KCV,KDIV,KSTART,KTOTAL,IPRG,IPCP,IDATIM)

      * IF THE MODEL IS HEMISPHERIC (KHEM=1,2) CALCULATIONS ARE DONE
      *  FOR THE SOUTHERN HEMISPHERE ONLY.

      NLAT=ILAT
      IF(KHEM.NE.0) NLAT=ILAT/2
      LGG=NLAT*(ILG+1)

      * READ OUTPUT CONTROLS FROM CARDS AND PERFORM INTERPOLATION.
      * REPEAT UNTIL NSTEPS IS NEGATIVE, THEN STOP.

  110 READ(5,5010) NSTEPS, MAPS,MPSP,IW,JW,LL,MM
      IF(NSTEPS.LT.0) WRITE(6,6090) (IDATIM(I),I=7,13)
      IF(NSTEPS.LT.0) STOP

      * IF MAPS=0, MAP CONTROLS ARE NOT READ.

      IF(MAPS.EQ.0) GO TO 210
      READ(5,5010) (MPGZ(L),L=1,LV)
      READ(5,5010) ( MPT(L),L=1,LV)
      READ(5,5010) ( MPW(L),L=1,LV)
      READ(5,5010)(MPWSP(L),L=1,LV)
      READ(5,5010) (MPES(L),L=1,LV)
```

```
  210 CALL STP6(NF1,NF12,NF2,PR,LV,SIG,NK,GZ,T,U,V,ES,LEVS,LGG,
     1           ILG,NLAT,NSTEPS, MAPS,MPSP,MPGZ,MPT,MPW,MPWSP,MPES,
     2              IW,JW,LL,MM,NPSP,NPGZ,NPT,NPW,NPES)

      GO TO 110

 5010 FORMAT(15I3)
 6090 FORMAT(1H0,7A4,6X,19H END SPECTRAL RSTP6)
      END
```

```
      SUBROUTINE SAVPCP(PCPPAK,STBPAK,ILG1,ILATH,KHEM,NPPCM,
     1                  KOUNT,KSTART,IPCP,NFPCP,NPPCP,GLL,WRKS)

C     * SPECTRAL MODEL PRECIP AND STABILITY COUNT SAVING ROUTINE.
C     * SAVES GAUSSIAN GRIDS OF PRECIPITATION AND STABILITY COUNT
C     * EVERY IPCP TIMESTEPS IF REQUESTED.

C     * KSYM = 0 MODEL IS GLOBAL AND COMPLETE GRID IS SAVED.
C     * KSYM = 1 MODEL IS HEMISPHERIC AND ONE HEM. ONLY IS SAVED.

      DIMENSION PCPPAK(1),STBPAK(1),GLL(ILG1,1)
      DIMENSION WRKS(1)

C     * SAVE EVERY IPCP TIMESTEPS PROVIDING IPCP.NE.0

      IF(IPCP.EQ.0) RETURN
      IF(MOD((KOUNT-KSTART),IPCP).NE.0) RETURN

C     * SET CONSTANTS

      NLAT=ILATH
      IF(KHEM.EQ.0) NLAT=ILATH*2
      ID=KOUNT

C     * WRITE THE GRID ON FILE NF WITH PACKING NPPCP.

      CALL WSGGP(PCPPAK,ILG1,NLAT,NFPCP,ID,4H PCP,1,NPPCP,GLL,WRKS)
      CALL WSGGP(STBPAK,ILG1,NLAT,NFPCP,ID,4H STB,1,NPPCP,GLL,WRKS)

      RETURN
      END
```

```
      SUBROUTINE SAVPRG(NF2,PHI,U,V,ES,PHIS,PS,PEE,TMEAN,PEEMN,PSMN,
     1            RGAS,LA,LAW,LRS,LRA,LRU,LRV,LM,ILEV,LEVS,KSYM,
     2            KSTART,KTOTAL,KOUNT,IPRG,GG,GGP,
     3            ILG1,ILATH,COA,KHEM,ALP,EPSI,LALP,WRKS,WRKL)

      * SAVES SIGMA LEVEL GAUSSIAN GRID FORECAST ON FILE NF2.

      COMPLEX PHI(LA,1),U(LAW,1),V(LAW,1),ES(LA,1)
      COMPLEX PHIS(1),PS(1),PEE(LA,1)
      DIMENSION TMEAN(1),PEEMN(1)

      * GGP IS A WORK FIELD USED ONLY IF FCST IS TO BE PACKED.
      *  (IT MAY BE EGUIVALENCED TO GG).

      DIMENSION GG(1),GGP(1),ALP(1),EPSI(1),WRKL(1)
      DIMENSION COA(1),WRKS(1)
      COMPLEX PS1

      * OUTPUT PACKING DENSITIES (SET IN MAIN PROGRAM).

      COMMON/PKOUT/ NPGZ,NPSP,NPW,NPES

      NSTEPS=KOUNT-KSTART
      ID=NSTEPS
      NLAT=ILATH
      IF(KHEM.EQ.0) NLAT=ILATH*2

      * NO PROG IS SAVED IF NVERIF=0. OTHERWISE PROG IS SAVED AT
      * INITIAL AND FINAL TIMES AND AT INTERVALS OF IPRG TIMESTEPS.

      IF(IPRG.EQ.0.OR.NF2.EG.0) RETURN
      IF(KOUNT.EQ.KTOTAL) GO TO 150
      IF(KOUNT.EQ.KSTART) GO TO 150
      IF(MOD(NSTEPS,IPRG).NE.0) RETURN

      * CONVERT LOG OF SURFACE PRESSURE FROM (N/M**2) TO (MB).

  150 PS1=PS(1)
      PS(1) = PS(1)+PSMN-ALOG(100.)*SQRT(2.)
      CALL SPAGG2(GG,ILG1,ILATH,COA,KHEM, PS ,LRS,LM,KSYM,
     1                        ALP,EPSI,LALP,WRKS,WRKL)
      CALL WSGGP(GG,ILG1,NLAT,NF2,ID,4HLNSP,1,NPSP,GGP,WRKS)
      PS(1)=PS1

      * CALCULATE PHI FROM PEE FOR ILEV LEVELS.
      * CONVERT TO GAUSSIAN GRIDS AND WRITE ON FILE NF2.

      CALL GZFBP (PHI,PEE,PS,TMEAN,PEEMN,PSMN,LA,LRS,LM,ILEV,RGAS)
      DO 310 L=1,ILEV
      CALL SPAGG2(GG,ILG1,ILATH,COA,KHEM,PHI(1,L),LRS,LM,KSYM,
     1                        ALP,EPSI,LALP,WRKS,WRKL)
```

```
      CALL WSGGP(GG,ILG1,NLAT,NF2,ID,4H  GZ,L,NPGZ,GGP,WRKS)
  310 CONTINUE

      * CONVERT U,V TO GAUSSIAN GRIDS AND SAVE ON NF2.

      DO 410 L=1,ILEV
      CALL SPAGG2(GG,ILG1,ILATH,COA,KHEM,  U(1,L),LRU,LM, KSYM,
     1                            ALP,EPSI,LALP,WRKS,WRKL)
      CALL WSGGP(GG,ILG1,NLAT,NF2,ID,4H   U,L,NPW,GGP,WRKS)
      CALL SPAGG2(GG,ILG1,ILATH,COA,KHEM,  V(1,L),LRV,LM,-KSYM,
     1                            ALP,EPSI,LALP,WRKS,WRKL)
      CALL WSGGP(GG,ILG1,NLAT,NF2,ID,4H   V,L,NPW,GGP,WRKS)
  410 CONTINUE

      * CONVERT ES TO GAUSSIAN GRIDS AND SAVE ON NF2.
      * MOISTURE OMITTED IF LEVS=0.

      IF(LEVS.EQ.0) GO TO 99
      DO 510 N=1,LEVS
      L=(ILEV-LEVS) + N
      CALL SPAGG2(GG,ILG1,ILATH,COA,KHEM, ES(1,N),LRS,LM,KSYM,
     1                            ALP,EPSI,LALP,WRKS,WRKL)
      CALL WSGGP(GG,ILG1,NLAT,NF2,ID,4H  ES,L,NPES,GGP,WRKS)
  510 CONTINUE

   99 RETURN
      END
```

```
      SUBROUTINE SCOF2(SC,LR,LM,KIND)

      * ZERO OR DOUBLE THE COMPLEX SPECTRAL COEFFICIENTS IN SC(LR,LM)
      * DEPENDING ON THE VALUE OF KIND.

      COMPLEX SC(LR,1)

      * IF KIND=0 SET ALL OF SC TO (0.,0.).

      IF(KIND.NE.0) GO TO 30
      DO 20 M=1,LM
      DO 20 N=1,LR
   20 SC(N,M)=(0.,0.)

      * IF KIND=2 DOUBLE ALL OF SC.

   30 IF(KIND.NE.2) GO TO 99
      DO 40 M=1,LM
      DO 40 N=1,LR
   40 SC(N,M)=SC(N,M)+SC(N,M)

   99 RETURN
      END
```

```
      SUBROUTINE SEAFLX(TG,ESG,UG,VG,TMEAN,ILM,ILONG,ILEV,LEVS,DT,
     1           SIAI,SEACON, RGOCP,SH,TSEA,COV)

*  FOR ILONG POINTS ABOUT ONE GAUSSIAN LATITUDE CIRCLE
*    CALCULATES CORRECTIONS TO TG,ESG IN LOWEST LAYER CAUSED BY
*    SENSIBLE HEAT AND MOISTURE FLUXES FROM UNDERLYING OCEAN.
*    SUCH A PROCEDURE IS EQUIVALENT TO A LAGGING
*    OF THE FLUX TERMS IN TIME.

*  TSEA IS SEA SURFACE TEMP IN DEGREES K
*  COV IS FRACTION OF LAND IN GRID SQUARE

      DIMENSION TG(ILM,1),ESG(ILM,1),UG(ILM,1),VG(ILM,1)
      DIMENSION THEAN(1),SH(1),TSEA(1),COV(1)

      BOT2 = 5418./(273.*273.)
      SFCEXT=1./SH(ILEV)**RGOCP
      ILEVP=ILEV+1

      DO 100 IK=1,ILONG

      IF(COV(IK).LE.0.)  GO TO 100
      TLEV = TG(IK,ILEVP) + TMEAN(ILEVP)
      DTSEA = TSEA(IK) - TLEV*SFCEXT
      WIND=SQRT( UG(IK,ILEV)**2 + VG(IK,ILEV)**2 )
      FACT=COV(IK)*SEACON*WIND*2.*DT/(TLEV*SIAI)
      IF(DTSEA.LT.0.)  DTSEA=0.0
      TG(IK,ILEVP) = TG(IK,ILEVP) + FACT*DTSEA

      IF(LEVS.EQ.0)  GO TO 100
      QSOQ = SH(ILEV)*EXP( BOT2 * ( TSEA(IK) - TLEV + ESG(IK,LEVS) ) )
      DTDSEA = ( QSOQ - 1. ) / BOT2
      IF(DTDSEA.LT.0.0)  DTDSEA=0.0
      ESG(IK,LEVS) = ESG(IK,LEVS) + FACT*( DTSEA - DTDSEA )

  100 CONTINUE

      RETURN
      END
```

```
SUBROUTINE SECOND(X)
X=0.0
RETURN
END
```

```
      SUBROUTINE SETL(SCT,SCF,LA,ILEV)

      * COPIES COMPLEX SPECTRAL COEFF IN SCF INTO SCT.
      * EACH ARRAY HAS ILEV LEVELS OF LA WORDS.
      * BOTH ARRAYS ARE IN LCM.

      COMPLEX SCT(LA,1),SCF(LA,1)


      DO 20 L=1,ILEV
      DO 20 MN=1,LA
   20 SCT(MN,L)=SCF(MN,L)

      RETURN
      END
```

```
SUBROUTINE SETOLD(PM,CM,PEEM,ESM,PSM,P,C,PEE,ES,PS,LA,ILEV,LEVS)

* COPIES CURRENT VALUES OF P,C,PEE,ES,PS INTO THE FIELDS HOLDING
*  THE PREVIOUS TIME VALUES PM,CM,ETC.

COMPLEX PM(LA,1),CM(LA,1),PEEM(LA,1),ESM(LA,1),PSM(1)
COMPLEX  P(LA,1), C(LA,1), PEE(LA,1), ES(LA,1), PS(1)

CALL SETL( PM , P ,LA,ILEV)
CALL SETL( CM , C ,LA,ILEV)
CALL SETL(PEEM,PEE,LA,ILEV)

CALL SETL( PSM, PS,LA,  1 )

IF(LEVS.GT.0) CALL SETL(ESM,ES,LA,LEVS)

RETURN
END
```

```
      SUBROUTINE SETZT (PT,CT,PEET,SDS,TDUM,EST,ESDUM,PRESS,
     1         LA,LRS,LRA,LM,ILEV,LEVS)
      * INITIALIZES SPECTRAL ARRAYS TO ZERO BEFORE STARTING THE
      * LATITUDE LOOP IN THE SPECTRAL MULTILEVEL MODEL.
      * MOISTURE VARIABLES OMITTED IF LEVS = 0.

      * IF MODEL IS HEMISPHERIC PT IS ANTISYMMETRIC,
      *                        ALL OTHER FIELDS ARE SYMMETRIC.
      * EACH LEVEL IS SEPARATED BY LA COMPLEX WORDS.

      COMPLEX PT(LA,1),CT(LA,1),PEET(LA,1)
      COMPLEX SDS(LA,1),TDUM(LA,1)
      COMPLEX EST(LA,1),ESDUM(LA,1)
      COMPLEX PRESS(1)


      DO 20 L=1,ILEV
      CALL SCOF2(  PT(1,L),LRA,LM,0)
      CALL SCOF2(  CT(1,L),LRS,LM,0)
      CALL SCOF2(PEET(1,L),LRS,LM,0)
      CALL SCOF2( SDS(1,L),LRS,LM,0)
      CALL SCOF2(TDUM(1,L),LRS,LM,0)
   20 CONTINUE
      CALL SCOF2(TDUM(1,ILEV+1),LRS,LM,0)

      CALL SCOF2(PRESS,LRS,LM,0)

      IF(LEVS.EQ.0) GO TO 99
      DO 73 L=1,LEVS
      CALL SCOF2(  EST(1,L),LRS,LM,0)
      CALL SCOF2(ESDUM(1,L),LRS,LM,0)
   73 CONTINUE

   99 RETURN
      END
```

```
      SUBROUTINE SFDRAG(PUTG,PVTG,UG,VG,TG,TMEAN,CDRAGG,ILM,ILEV,ILONG,
     1                  GRAV,A,RGAS,DS,SIAI)

      * ADDS CONTRIBUTION OF SURFACE DRAG TO PUTG,PVTG
      * FOR ILONG POINTS ABOUT ONE GAUSSIAN LATITUDE CIRCLE.

      DIMENSION PUTG(ILM,1),PVTG(ILM,1),UG(ILM,1),VG(ILM,1)
      DIMENSION TG(ILM,1),TMEAN(1),CDRAGG(1),DS(1)


      ILEVP=ILEV+1
      DO 20 IK=1,ILONG
      FACT = CDRAGG(IK)*GRAV*(A/SIAI)/(RGAS*DS(ILEV))
      VMAG=SQRT( UG(IK,ILEV)**2 + VG(IK,ILEV)**2 )
      FACT = FACT * VMAG / (TG(IK,ILEVP)+TMEAN(ILEVP))
      PUTG(IK,ILEV)=PUTG(IK,ILEV)+FACT*VG(IK,ILEV)
      PVTG(IK,ILEV)=PVTG(IK,ILEV)-FACT*UG(IK,ILEV)
   20 CONTINUE

      RETURN
       END
```

```
      SUBROUTINE SGTPRE(FX,FDER,SIGMA,SP,LA,LB,PRL,LV,SG,F,G,NN,GLAPSE,
     1             EXTRAP,KIND,GZPI,SFCPI)

      * EXTRAP=1. FOR HEIGHTS AND TEMP, 0. FOR WINDS.
      * GLAPSE IS LAPSE RATE BELOW SIGMA=1.  SELECT SOME POSITIVE VALUE
      *  FOR GEOPOTENTIALS. SET TO ZERO FOR ALL OTHER VARIABLES.
      * GZPI= GZ INITIAL ON PR SFC.  SFCPI= INITIAL LN(SF PRES).

      * KIND=1 RETURNS VERTICAL DERIVATIVE IN FDER. (OFF IF KIND=0)
      * NOTE - IF LA=LB, FX OR FDER MAY BE EQUIVALENCED TO SIGMA.

SGTPRT DOES VERTICAL INTERPOLATION ON A FIELD SIGMA, ASSUMED KNOWN ON
NN SIGMA LEVELS AT LA POINTS PER LEVEL, STORED CONSECUTIVELY,
PRODUCING AN OUTPUT FX(LA,LV), LOCATED AT THE SAME HORIZONTAL POINTS,
BUT AT THE PRESSURE LEVELS LOCATED BY PRL(LV). INTERPOLATION IS DONE
USING THE LOGARITHM OF PRESSURE/SIGMA AS THE VERTICAL COORDINATE.
FDER(LA,LV) CONTAINS THE VERTICAL DERIVATIVE OF FX IF KIND=1.

SP(LA) IS A FIELD OF LOG OF SURFACE PRESSURE, FOR SIGMA/PRESSURE
CONVERSION. PRLOG IS THE LOG OF THE DESIRED PRESSURE LEVEL, (IN THE
SAME UNITS AS SP) WHILE SG(NN) CONTAINS THE LOGS OF THE SIGMA LEVELS.
NOTE THAT DIFFERENT SIGMA LEVELS ARE STORED LB POINTS APART IN SIGMA.

NOTE ALSO THAT IF THE FIELD SP IS 0.0, THE ROUTINE CAN BE THOUGHT OF
AS A RELATIVELY GENERAL PURPOSE INTERPOLATOR, GIVING FIELDS AT THE
COORDINATE PRLOG, IN TERMS OF FIELDS AT THE COORDINATES SG(NN).

       DIMENSION FX(LA,LV),FDER(LA,LV),GZPI(LA,LV)
      DIMENSION SIGMA(LB,NN),SP(LA),SFCPI(LA)
      DIMENSION SG(NN),F(NN),G(NN),PRL(LV)

      TOLAPS=0.
      SENTR=0.5

      DO 20 I=1,LA

      DO 10 N=1,NN
   10 F(N) = SIGMA(I,N)
      CALL TSIG(G,F,SG,NN,TOLAPS,SENTR,2.0)

      DO 20 L=1,LV
      X = PRL(L) - SP(I)
      IF(EXTRAP.GT.0.) XI=PRL(L)-SFCPI(I)
      IF(EXTRAP.GT.0.) FI=GZPI(I,L)
      CALL TERP2E(FXIL,DERIV,X,F,G,SG,NN,EXTRAP,GLAPSE,FI,XI)
      FX(I,L)=FXIL
      IF(KIND.EQ.1) FDER(I,L)=DERIV
   20 CONTINUE

      RETURN
      END
```

THIS PAGE IS BLANK

```
      SUBROUTINE SMOV2(QC,LRC,QI,LRI,LM)

      * FILLS COMPLEX QC(LRC,LM) FROM QI(LRI,LM) BY EXTRACTING
      * THE FIRST LRC COMPLEX WORDS FROM EACH ROW OF QI.
      * NOTE - LRC SHOULD NOT EXCEED LRI.

      COMPLEX QC(LRC,1),QI(LRI,1)


      DO 210 M=1,LM
      DO 210 N=1,LRC
  210 QC(N,M)=QI(N,M)

      RETURN
      END
```

```
      SUBROUTINE SPAF2(CFC,SC,LR,LM,ALP,LALP)

*     CALCULATES COMPLEX FOURIER COEFFICIENTS IN CFC(LM)
*      FROM SPECTRAL COEFFICIENTS IN SC(LR,LM).

*     CFC WILL CONTAIN WAVES 0 TO (LM-1).
*     IF SC IS GLOBAL, ROW 1 CONTAINS WAVES 0,1,2...(LR-1).
*     IF SC IS SYMMETRIC, ROW 1 CONTAINS WAVES 0,2,4...2*(LR-1).
*     IF SC IS ANTISYMMETRIC, ROW 1 CONTAINS WAVES 1,3,5...2*(LR-1)+1.
*     ALP(LALP,LM) CONTAINS LEGENDRE POLYNOMIALS FOR ONE LATITUDE.
*      ITS ROWS MUST HAVE THE SAME STRUCTURE AS THOSE OF SC.

      COMPLEX CFC(1),SC(LR,1)
      DIMENSION ALP(LALP,1)


      DO 30 M=1,LM
      FCR=0.
      FCI=0.
      DO 20 N=1,LR
      FCR=FCR+ALP(N,M)* REAL(SC(N,M))
      FCI=FCI+ALP(N,M)*AIMAG(SC(N,M))
   20 CONTINUE
      CFC(M)=CMPLX(FCR,FCI)
   30 CONTINUE

      RETURN
      END
```

```
      SUBROUTINE SPAGG2(GG,ILG1,ILATH,COA,KHEM,P,LR,LM,KSM,
     1                  ALP,EPSI,LALP,WRKS,WRKL)

C     * PRODUCES GAUSSIAN GRID FROM SPECTRAL COEFF IN P(LR,LM).
C     * IF GLOBAL GG=(ILG1,ILATH*2), OTHERWISE GG=(ILG1,ILATH).
C     * LATITUDE IS ZERO AT LEFT OF GRID AND POSITIVE EASTWARD.
C     * LEFT COLUMN IS COPIED INTO THE RIGHT COLUMN.

C     * KHEM   0 = GLOBAL,  1 = N HEM ONLY,  2 = S HEM ONLY.
C     * KSM   +1=SYMMETRIC,  0=GLOBAL,  -1=ANTISYMMETRIC
C     * COA(ILATH) CONTAINS THE COSINES OF THE COLATITUDE (N TO S).
C     * ALP(LALP,LM) IS A WORK FIELD FOR LEGENDRE POLYNOMIALS.
C     * EPSI IS A FIELD OF CONSTANTS THE SAME SIZE AS ALP.
C     * WRKS,WRKL ARE SCM,LCM WORK ARRAYS OF (ILG+2) WORDS.
C     * FAST FOURIER TRANSFORM REQUIRES THAT ILG BE A POWER OF 2.

      COMPLEX P(1)
      DIMENSION GG(ILG1,1),COA(1)
      DIMENSION ALP(1),EPSI(1),WRKS(1),WRKL(1)

      ILG=ILG1-1
      ILGH=ILG/2
      MAXF=LM-1
      ILP=1
      IF(KSM.LT.0) ILP=LALP/2+1

C     * DO NORTHERN HEMISPHERE IF REQUESTED (KHEM=0 OR 1).

      IF(KHEM.EQ.2) GO TO 38
      DO 30 IH=1,ILATH
      SINLAT=COA(IH)
      JRX=ILATH+1-IH
      IF(KHEM.EQ.0) JRX=JRX+ILATH
      CALL ALPNM2(ALP,LALP,LM,SINLAT,EPSI)
      IF(KSM.NE.0) CALL ALPAS2(ALP,LALP,LM,WRKS)
      CALL SPAF2(WRKL,P,LR,LM,ALP(ILP),LALP)
      CALL FFGFW2(GG(1,JRX),ILG1,WRKL,ILGH,MAXF,ILG,WRKS,1)
      GG(ILG1,JRX)=GG(1,JRX)
   30 CONTINUE

C     * DO SOUTHERN HEMISPHERE IF REQUESTED (KHEM=0 OR 2).

   38 IF(KHEM.EQ.1) RETURN
      DO 40 IH=1,ILATH
      SINLAT=-COA(IH)
      JRX=IH
      CALL ALPNM2(ALP,LALP,LM,SINLAT,EPSI)
      IF(KSM.NE.0) CALL ALPAS2(ALP,LALP,LM,WRKS)
      CALL SPAF2(WRKL,P,LR,LM,ALP(ILP),LALP)
      CALL FFGFW2(GG(1,JRX),ILG1,WRKL,ILGH,MAXF,ILG,WRKS,1)
      GG(ILG1,JRX)=GG(1,JRX)
```

```
40 CONTINUE
   RETURN
   END
```

```
      SUBROUTINE SPAPS2(G,NI,NJ,IP,JP,D60,DGRW,NHEM,P,LR,LM,KSM,
     1                        ALP,EPSI,LALP,WRKS,WRKL)

*     CALCULATES POLAR STEREOGRAPHIC GRID G(NI,NJ) WITH POLE (IP,JP)
*      FROM COMPLEX SPECTRAL COEFF IN P(LR,LM).
*     PS GRID SIZE IS ASSUMED TO BE D60 METERS.
*     PS GRID ORIENTATION HAS ZERO DEGREES LONG. AT (DGRW) DEGREES.

*     NHEM 1 = NORTHERN HEMISPHERE.   NHEM 2 = SOUTHERN HEMISPHERE.
*     ALP(LALP,LM) IS A WORK FIELD FOR LEGENDRE POLYNOMIALS.
*     EPSI IS A FIELD OF CONSTANTS THE SAME SIZE AS ALP.
*     WRKS IS AN SCM WORK FIELD OF LALP WORDS.
*     WRKL IS AN LCM WORK FIELD OF LM COMPLEX WORDS.

      COMPLEX P(1)
      DIMENSION G(NI,NJ),EPSI(1),ALP(1)
      DIMENSION WRKS(1),WRKL(1)

      DRCON=3.14159/180.
      MAXF=LM-1
      ILP=1
      IF(KSM.LT.0) ILP=LALP/2+1

*     SET GRID TO LARGE INITIAL VALUE.

      GFILL=1.11E+75
      DO 10 J=1,NJ
      DO 10 I=1,NI
   10 G(I,J)=GFILL

*     LOOP OVER THE ENTIRE GRID.

      DO 20 J=1,NJ
      JY=J-JP
      YP=JY
      DO 20 I=1,NI
      IX=I-IP
      XP=IX
      IF(G(I,J).NE.GFILL) GO TO 20

*     OBTAIN THE LATITUDE OF POINT (I,J) AND GET THE FOURIER COEFF
*      FOR THAT LAT FROM THE SPECTRAL COEFF IN P.

      CALL LLFXY(DLAT,DLON,XP,YP,D60,DGRW,NHEM)
      SINLAT=SIN(DLAT*DRCON)
      CALL ALPNM2(ALP,LALP,LM,SINLAT,EPSI)
      IF(KSM.NE.0) CALL ALPAS2(ALP,LALP,LM,WRKS)
      CALL SPAF2(WRKL,P,LR,LM,ALP(ILP),LALP)

*     GET THE COORDINATES OF EACH OF THE POSSIBLE EIGHT POINTS
*      IN THE GRID THAT LIE ON THE GIVEN LAT CIRCLE.
```

```
      II=IP+IX
      JJ=JP+JY
      X=XP
      Y=YP
      ASSIGN 12 TO NPT
      GO TO 19
   12 II=IP-IX
      X=-XP
      ASSIGN 13 TO NPT
      GO TO 19
   13 JJ=JP-JY
      Y=-YP
      ASSIGN 14 TO NPT
      GO TO 19
   14 II=IP+IX
      X=XP
      ASSIGN 15 TO NPT
      GO TO 19

   15 II=IP+JY
      JJ=JP+IX
      X=YP
      Y=XP
      ASSIGN 16 TO NPT
      GO TO 19
   16 II=IP-JY
      X=-YP
      ASSIGN 17 TO NPT
      GO TO 19
   17 JJ=JP-IX
      Y=-XP
      ASSIGN 18 TO NPT
      GO TO 19
   18 II=IP+JY
      X=YP
      ASSIGN 20 TO NPT

      * MAKE SURE POINT IS BOTH INSIDE GRID AND NOT ALREADY DONE.
      * GET LONGITUDE AND EVALUATE FOURIER SERIES AT THAT POINT.

   19 IF(II.LT.1.OR.II.GT.NI) GO TO 195
      IF(JJ.LT.1.OR.JJ.GT.NJ) GO TO 195
      IF(G(II,JJ).NE.GFILL) GO TO 195
      CALL LLFXY(DLAT,DLON,X,Y,D60,DGRW,NHEM)
      RLON=DLON*DRCON
      CALL CFVAL2( GIJ  ,WRKL,MAXF,RLON)
      G(II,JJ)=GIJ
  195 GO TO NPT,(12,13,14,15,16,17,18,20)
   20 CONTINUE
```

```
RETURN
END
```

```
      FUNCTION SPCHUM(TD,P)
*  COMPUTES SPECIFIC HUMIDITY *SPCHUM* USING
*  DEW POINT TEMPERATURE   TD  AND PRESSURE  P(MB)
*    E   IS THE VAPOR PRESSURE RELATED TO  TD  BY
*                                   E=EXP(A-B/TD)
*  FOR SATURATION VALUES TRUE TEMPERATURE REPLACES  TD  IN MAIN

      COMMON/EPS/A,B,EPS1,EPS2

      E=EXP(A-B/TD)

      SPCHUM=EPS1*E/(P-EPS2*E)

      RETURN
      END
```

```
      SUBROUTINE SPLAB(NF,KIND,LV,PR,NK,SIG,LEVS,ILG,ILAT,KHEM,IR,DEET,
     1         MTN,WET,KDR,KWT,KCV,KDIV,KSTART,KTOTAL,IPRG,IPCP,IDATIM)

*     THIS SUBROUTINE READS/WRITES THE CONTROL LABEL FOR THE
*      SPECTRAL MODEL PROGRAM SEQUENCE ON FILE NF.
*      KIND=1 READS,  KIND=2 WRITES.

*     PR(LV) CONTAINS THE LV INITIAL PRESSURE LEVELS.
*     SIG(NK) CONTAINS THE NK MODEL SIGMA LEVELS.
*     LEVS IS THE NUMBER OF MOISTURE LEVELS USED IN THE MODEL.
*     (ILG+1,ILAT) = GAUSSIAN GRID DIMENSIONS (GLOBAL).
*     KHEM 0=GLOBAL, 1=N HEM, 2=S HEM.
*     IR,DEET = MODEL RESOLUTION AND TIMESTEP.
*     MTN,WET,KDR,KWT,KCV,KDIV ARE LOGICAL SWITCHES FOR MOUNTAINS,
*       MOISTURE,DRAG,WATER-TEMP,SEA-COVER,INITIAL DIVERGENCE.
*     KSTART,KTOTAL = MODEL STARTING AND ENDING TIMESTEP NUMBERS.
*     IPRG,IPCP = FORECAST AND PRECIP SAVE INTERVALS.
*     INITIAL DATE TIME GROUP IS IN IDATIM

      DIMENSION PR(LV),SIG(NK),IDATIM(14)
      LOGICAL MTN,WET,KDR,KWT,KCV,KDIV
      DATA HEADER,IDENT,NAME/ 4HLABL, 0, 4HINFO /

      REWIND NF

*     READ THE LABEL IF KIND=1.

      IF(KIND.EQ.1) READ(NF) HEAD,ID,INFO,
     1               LV,(PR(L),L=1,LV),NK,(SIG(N),N=1,NK),LEVS,DEET,
     2               ILG,ILAT,KHEM,IR,MTN,WET,KDR,KWT,KCV,KDIV,
     3               KSTART,KTOTAL,IPRG,IPCP,IDATIM
      IF(KIND.EQ.1) WRITE(6,6010) NF

*     WRITE THE LABEL IF KIND=2.

      IF(KIND.EQ.2) WRITE(NF) HEADER,IDENT,NAME,
     1               LV,(PR(L),L=1,LV),NK,(SIG(N),N=1,NK),LEVS,DEET,
     2               ILG,ILAT,KHEM,IR,MTN,WET,KDR,KWT,KCV,KDIV,
     3               KSTART,KTOTAL,IPRG,IPCP,IDATIM
      IF(KIND.EQ.2) WRITE(6,6020) NF

*     PRINT OUT ALL LABEL INFORMATION.

   30 WRITE(6,6030) LV,PR
      WRITE(6,6032) NK,SIG
      WRITE(6,6040) LEVS,ILG,ILAT,KHEM,IR,DEET
      WRITE(6,6045) MTN,WET,KDR,KWT,KCV,KDIV
      WRITE(6,6050) KSTART,KTOTAL,IPRG,IPCP
      WRITE(6,6060) IDATIM
      RETURN
```

```
6010 FORMAT(31H0SPECTRAL LABEL  READ FROM FILE,I4)
6020 FORMAT(31H0SPECTRAL LABEL WRITTEN TO FILE,I4)
6030 FORMAT(22H  INITIAL PRESSURES   ,I5,15F6.0)
6032 FORMAT(22H  MODEL SIGMA LEVELS  ,I5,15F6.3)
6040 FORMAT(29H  LEVS,ILG,ILAT,KHEM,IR,DEET=,5I5,F8.2)
6045 FORMAT(29H  MTN,WET,KDR,KNT,KCV,KDIV   =,6L5)
6050 FORMAT(29H  KSTART,KTOTAL,IPRG,IPCP    =,4I5)
6060 FORMAT(22H  DATE-TIME GROUP      ,2X,6I3,7A4,I12)
     END
```

```
      SUBROUTINE SPLAT2(A,LI,NC,II,INCR,GMIN,GMAX)

      * PLOTS UP TO 6 CURVES ON THE SAME GRAPH.
      * PLOTS EVERY INCR POINTS FROM 1 TO II IN A(LI,NC)
      * CURVES 1 TO 6 USE SYMBOLS A,B,C,D,E,F.
      * VALUE OF LAST CURVE WRITTEN ALONG X-AXIS UNLESS INCR IS NEGATIVE
      * NO GRAPH DRAWN IF INCR=0 OR NC=0.
      * GRAPH LIMITS GMIN,GMAX AND MIDDLE VALUE PRINTED AT START AND END
      * IF GMIN=GMAX GRAPH LIMITS CALCULATED FIRST.
      * IF THE RANGE IS STILL ZERO, IT IS RESET TO 1.

      DIMENSION A(LI,NC)
      INTEGER NG(101),S(4),IGR(6)
      DATA S/1H ,1HI,1H-,1H*/
      DATA IGR/1HA,1HB,1HC,1HD,1HE,1HF/


      IF(INCR.EQ.0.OR.NC.EQ.0) GO TO 99
      INC=IABS(INCR)
      AMIN=GMIN
      AMAX=GMAX
      IF(AMIN.LT.AMAX) GO TO 14
      AMIN=A(1,1)
      AMAX=AMIN
      DO 12 J=1,NC
      DO 12 I=1,II,INC
      T=A(I,J)
      IF(T.LT.AMIN) AMIN=T
      IF(T.GT.AMAX) AMAX=T
   12 CONTINUE
      IF(AMIN.EQ.AMAX) AMAX=AMIN+1.

   14 RANGE=AMAX-AMIN
      DA100=100./RANGE
      AX=(AMAX+AMIN)*.5
      WRITE(6,605) IGR(NC),RANGE
      WRITE(6,610) AMIN,AX,AMAX
      WRITE(6,615)

      DO 24 I=1,II,INC
      DO 15 K=1,101
   15 NG(K)=S(1)
      NG(51)=S(2)
      IF(I/10*10.EQ.I) NG(51)=S(3)
      DO 20 J=1,NC
      Y=(A(I,J)-AMIN)*DA100+1.5
      K=INT(Y)
      IF(K.GE.1.AND.K.LE.101) NG(K)=IGR(J)
   20 CONTINUE
      IF(INCR.GT.0) WRITE(6,620) I,A(I,NC),NG
      IF(INCR.LT.0) WRITE(6,625) NG
```

```
24 CONTINUE
   WRITE(6,615)
   WRITE(6,610) AMIN,AX,AMAX
99 RETURN
605 FORMAT(8H1 CURVE ,A1,T87,7HRANGE =,1PE12.5)
610 FORMAT(1H ,16X,1PE12.5,T68,E12.5,T118,E12.5)
615 FORMAT(1H ,21X,10(10HI---------),1HI)
620 FORMAT(1H ,I4,1PE15.6,2X,101A1)
625 FORMAT(1H ,21X,101A1)
    END
```

```
      SUBROUTINE SPMCON(S,SH,SF,SHF,DS,DEL,FVORT,PI,GAMMA,ILEV)

*     SETS CONSTANTS FOR THE MULTILEVEL SPECTRAL MODEL.

      DIMENSION S(1),SH(1),SF(1),SHF(1),DS(1),DEL(1)
      COMMON/PARAMS/ WW,TW,A,ASQ,GRAV,RGAS,RGOCP,RGOASQ,CPRES
      ILEVM = ILEV-1

*        WW = EARTH ROTATION RATE (1/SEC).
*         A = EARTH RADIUS (M).
*      GRAV = GRAVITY ACCELERATION (M/SEC**2).
*      RGAS = DRY AIR GAS CONSTANT (JOULE/(KG*DEG)).
*     RGOCP = RGAS/(DRY AIR SPECIFIC HEAT)
*     GAMMA = LAPSE RATE FROM THE LOWEST LAYER TO THE SURFACE

      WW=7.292E-5
      TW=WW+WW
      GAMMA=5.90E-03
      A=6.37122E06
      ASQ=A*A
      GRAV=9.80616
      RGAS=287.04
      RGOCP=2./7.
      RGOASQ=RGAS/ASQ
      CPRES=RGAS/RGOCP

*     FVORT = VORTICITY OF EARTH ROTATION.

      PI=3.14159265358097
      FVORT=TW*SQRT(2./3.)

*      S = SIGMA AT THE ODD LEVELS.
*     SH = SIGMA AT THE EVEN LEVELS
*     DS = DELTASIGMA BETWEEN EVEN LEVELS.

      DO 30 IH=1,ILEVM
   30 DEL(IH) = S(IH+1) - S(IH)
      DEL(ILEV) = 1. - S(ILEV)

      DO 40 IH = 1,ILEVM
   40 SH(IH) = SQRT(S(IH)*S(IH+1))
      SH(ILEV) = SQRT(S(ILEV))

      DS(1) = SH(1)
      DO 50 IH=2,ILEVM
   50 DS(IH) = SH(IH)-SH(IH-1)
      DS(ILEV) = 1.-SH(ILEVM)

      DO 60 IH=1,ILEVM
   60 SF(IH) = ALOG(S(IH+1)/S(IH))
      SF(ILEV) = ALOG(1./S(ILEV))
```

```
      DO 70 IH=1,ILEVM
   70 SHF(IH)=ALOG(SH(IH+1)/SH(IH))
      SHF(ILEV)=ALOG(1./SH(ILEV))
      RETURN
      END
```

```
      SUBROUTINE SPW6(ILG,ILAT,KHEM,IR,ILEV,LEVS,S,
     1                NF1,NF2,NFPCP,LC,ETOTS,LTOT,IEPR,NOGR,IPR,
     2                    NTEST,IPRG,IPCP,NPPCP,IDATIM)

*     SPECTRAL IMPLICIT MULTILEVEL WET MODEL - VERSION 6.
*     FOR A COMPLETE DESCRIPTION OF THIS PROGRAM, SEE....
*      (THE D.P.R. MULTILEVEL SPECTRAL MODEL PROGRAM - VERSION 6)
*        AND OTHER DOCUMENTS  (AVAILABLE AT D.P.R. MONTREAL, CANADA).

*     NF1 = INPUT FILE FOR ALL VARIABLES AND PHYSICS GRIDS.
*     NF2 = OUTPUT FILE FOR PHI,PHIS,LNSP,U,V,ES.
*     NFPCP = OUTPUT FILE FOR PRECIPITATION.

*     I/O ARRAYS AND PARAMETERS.

      DIMENSION LC(1),ETOTS(LTOT,6),IDATIM(14)
      LOGICAL MTN,WET,KDR,KWT,KCV,KDIV
      COMMON/SWCOM/ MTN,WET,KDR,KWT,KCV,KDIV

*     LCM COMPLEX MULTILEVEL ARRAYS

      PARAMETER $ILV=15,$LEV=9,$LV=10,$ILT=52,$ILG=64,$ILTH=26,$IR=20
      PARAMETER $LA = ($IR+2)*($IR+1)
      PARAMETER $LAW = $LA*$ILV
      PARAMETER $LAX = $LA*($LEV+1)
      PARAMETER $IL1 = $ILV + 1
      PARAMETER $LW1 = $LA*$IL1
      PARAMETER $LGG = $ILG*$ILT
      PARAMETER $LGP = ($ILG+2)*$ILT
      PARAMETER $IRL = (2*$IR+1)*$ILV
      PARAMETER $CR  = $ILG*$ILV
      PARAMETER $CS = $ILG*($LEV+1)
      PARAMETER $CR1 = $ILG*$IL1
      PARAMETER $ILG3 = $ILG + 2

      COMPLEX P,C,PEE,PS, PM,CM,PEEM,PSM, PT,CT,PEET
      COMPLEX U,V,PHI,T,SDS,ES,ESM,EST
      COMMON//P($LAW),C($LAW),PEE($LAW),PS($LA),PM($LAW),CM($LAW),
     * PEEM($LAW),PSM($LA),PT($LAW),CT($LAW),PEET($LAW),U($LAW),V($LAW),
     *PHI($LW1),T($LW1),SDS($LAW),ES($LAX),ESM($LAX),EST($LAX)
      COMPLEX TDUM(1),DI(1),PEEBAR(1),ESDUM(1)
      EQUIVALENCE (TDUM,PHI),(DI,V),(PEEBAR,SDS),(ESDUM,PEE)

*     LCM PHYSICS - 1 LEVEL COMPLEX ARRAYS

      COMPLEX PHIS,PRESS
      COMMON//PHIS($LA),PRESS($LA)

*     LCM PHYSICS - GAUSSIAN GRIDS AND WORK ROWS.

      COMMON/GGPAK/DRPAK($LGP),WTPAK($LGP),CVPAK($LGP),PCPPAK($LGP),
```

```
*STBPAK($LGP)

 * LCM WORK ARRAYS

 COMMON/ALPCOM/ALP($LA),DALP($LA),EPSI($LA)
 COMMON//AI($IRL),BI($IRL),CI($IRL),WRKL(300)

 COMMON//PRESSG($ILG),PSDLG($ILG),PSDPG($ILG),CUMDUM($ILG),
*PG($CR),CG($CR),
*TG($CR1),UG($CR),VG($CR),EG($CR),PUTG($CR),PVTG($CR),TUTG($CR),
*TVTG($CR),ESG($CS),FEETG($CR),SDSG($CR),AMBDPG($CR),OMEGAG($CR)
 DIMENSION SUSG(1),SVSG(1),ESTG(1)
 EQUIVALENCE (SUSG(1),PG(1)),(SVSG(1),CG(1)),(ESTG(1),AMBDPG(1))
 DIMENSION GLL(1)
 EQUIVALENCE (GLL(1),PRESSG(1))

 * SCM WORK AND I/O FIELDS.

 COMMON/GAUSS/W($ILTH),WOCS($ILTH),COA($ILTH),SIA($ILTH),RAD($ILTH)
 COMMON/SCM1/WRKS($ILG3,8)

 * THE FOLLOWING ARRAYS ARE FOR FUNCTIONS OF ILEV.
 * CURRENTLY SET FOR A MAXIMUM OF 10 LEVELS.

 COMPLEX P2($ILV)
 DIMENSION STAMS($IL1,$IL1),STAIS($IL1,$IL1),STAWS($IL1,6),
*RCPSHD($ILV),AMBDA($IL1),PEEMN($ILV),RK($ILV),THEAN($IL1),
*S($ILV),SH($ILV),SF($ILV),SHF($ILV),DS($ILV),DEL($ILV),TOTK($ILV)

 * SCM COMMON BLOCKS FOR MODEL PARAMETERS

 COMMON/TIMES/ DEET,KOUNT,KSTART,KTOTAL,IFDIFF
 COMMON/PARAMS/ WW,TW,A,ASQ,GRAV,RGAS,RGOCP,RGOASQ,CPRES
 COMMON /PARAM2/ BETA,AVERT,DIVCH,FP,FC,FPEE,FPS,FS
 COMMON/PARAM3/ DIFUSD,DIFUSS,VIFUSS
 COMMON/PARAM4/ CEEACH,RKL
.....................................................................
 DATA NPOR,NPWT,NPCV,NPPCM/1,1,1,1/

 * SECTION 1 - CONSTANTS AND DATA PREPARATION.

 IF(NTEST.NE.0) CALL SECOND(SEC2)
 WRITE(6,6005)
 WRITE(6,6010) KSTART,KTOTAL,IFDIFF
 WRITE(6,6012) DEET,BETA,DIVCH,FP,FC,FPEE,FPS,FS
 WRITE(6,6012) DIFUSD,DIFUSS,CEEACH,RKL

 * CALCULATE CONSTANTS.

 KOUNT=KSTART
 ILEVP=ILEV+1
```

```
      ILATH=ILAT/2
      ILG1=ILG+1
      LTBS=ILEV-LEVS
      ILM=ILG
      ILH=ILM/2
      KSYM=0
      IF(KHEM.NE.0) KSYM=1
      DT=DEET
      IF(IFDIFF.EQ.1) DT=DEET/2.
      CALL DIMCAL(LRS,LRA,LRU,LRV,LALP,LM,LA,LAW,IR,KHEM)
      WRITE(6,6015) LRS,LRA,LRU,LRV,LALP,LM,LA,LAW

      CALL SPMCON(S,SH,SF,SHF,DS,DEL,FVORT,PI,GAMMA,ILEV)
      CALL EPSIL2(EPSI,LALP,LM)
      CALL GAUSSG(ILATH,COA,W,SIA,RAD,WOCS)
      CALL PHSCON(SEACON,RK,CEEACH,RKL,BETA,S,DEL,ILEV)

      IF(WET)CALL WETCON(STAWS,RCPSHD,S,SH,SF,DEL,RGOCP,ILEV-1,LEVS)
      CALL STMCAL (STAMS,STAIS,S,SH,SF,SHF,RGOCP,
     1            RGAS,GRAV,GAMMA,ILEV,ILEVP,EXTRAP)

      * READ INPUT FIELDS FROM SEQUENTIAL FILE NF1.

      CALL INPTGG(NF1,NF2,DRPAK,WTPAK,CVPAK,ILG1,ILATH,KHEM,
     1            KDR,KWT,KCV,NPDR,NPWT,NPCV,GLL,PM,WRKS)
      ID=KSTART
      CALL INPTSP(NF1,ID,NF2,PHIS,PS,PHI,P,C,ES,LA,LRS,LRA,
     1            LM,ILEV,LEVS,KHEM,ILG1,ILATH,COA,W,WOCS,GLL,PM,PT,
     2            ALP,DALP,EPSI,LALP,WRKS,WRKL)

      * SET INITIAL DIVERGENCE TO ZERO IF REQUESTED.

      IF(.NOT.KDIV) CALL SCOF2(C,LA,ILEV,0)

      * COMPUTE T FROM GZ, SET TO T-PRIME, SAVE MEANS IN TMEANH.

      CALL TFGZ(T,PHI,PHIS,LA,LRS,LM,ILEV,SF,RGAS,S,GRAV,GAMMA)
      DO 150 L=1,ILEVP
      N=(L-1)*LA+1
      TMEAN(L)=REAL(T(N))/SQRT(2.)
  150 T(N)=(0.,0.)
      CALL TMCAL(TMEAN,AMBDA,S,SH,SF,RGOCP,STAMS,ILEV,ILEVP)
      CALL PBLMAT(RGAS,RGOCP,S,SH,SF,SHF,AMBDA,TMEAN,GRAV,GAMMA,ILEV)

      * SET PS TO PS-PRIME, SAVE MEAN IN PSMN.

      PSMN=PS(1)
      PS(1)=(0.,0.)
      PSMNR2=PSMN/SQRT(2.)

      * COMPUTE MEANS OF PEE FROM GZ. STORE IN PEEMN.
```

```
      DO 180 L=1,ILEV
      N=(L-1)*LA+1
  180 PEEMN(L)=REAL(PHI(N)) + RGAS*TMEAN(L)*PSMN

      * SECTION 2 - BEGIN TIME STEP.  CALCULATE (U,V) FROM (P,C).

  200 IF(NTEST.EQ.0) GO TO 210
      SEC1=SEC2
      CALL SECOND(SEC2)
      TIME=SEC2-SEC1
      HR=KOUNT*DEET/3600.
      WRITE(6,6020) KOUNT,HR,TIME

  210 DO 220 L=1,ILEV
      NUV=(L-1)*LAW+1
      NPC=(L-1)*LA +1
      CALL QDAW2(U(NUV),V(NUV),P(NPC),C(NPC),EPSI,WRKS(1,1),WRKS(1,2),
     1                        LPU,LRV,LRA,LRS,LALP,LM,KHEM,IR)
  220 CONTINUE

      * ADD EARTH ROTATION TO VORTICITY. SET TENDENCIES TO ZERO.

      DO 230 L=1,ILEV
      N=(L-1)*LA+2-KSYM
      F2(L)=P(N)
  230 P(N)=P(N)+CMPLX(FVORT,0.)

      DO 240 L=1,ILEV
  240 TOTK(L)=0.
      CALL SETZT (PT,CT,PEET,SDS,TDUM,EST,ESDUM,PRESS,
     1            LA,LRS,LRA,LM,ILEV,LEVS)

      * SECTION 3 - LATITUDE LOOP.
      * THE NORTHERN HEMISPHERE IS DONE FIRST, FOLLOWED BY THE
      * SOUTHERN HEMISPHERE. EITHER CAN BE OMITTED IF NOT NEEDED.
      * NH=1 FOR NORTHERN HEMISPHERE, NH=2 FOR SOUTHERN HEMISPHERE.

      NUPS=0
      NSUPS=0

      DO 720 NH=1,2
      IF(NH.EQ.1.AND.KHEM.EQ.2) GO TO 720
      IF(NH.EQ.2.AND.KHEM.EQ.1) GO TO 720

      * LATITUDE LOOP IN ONE HEMISPHERE.
      * IHEM COUNTS FROM THE POLE TO THE EQUATOR.
      * IHGG INDEXES THE ROWS IN THE GAUSSIAN GRIDS STORED IN CORE.

      DO 700 IHEM=1,ILATH
      IHGG=IHEM
```

```
      IF(NH.EQ.1.AND.KHEM.EQ.1) IHGG=(ILATH+1-IHEM)
      IF(NH.EQ.1.AND.KHEM.EQ.0) IHGG=(ILATH+1-IHEM)+ILATH
      SIAI= SIA(IHEM)
      COAI=-COA(IHEM)
      IF(NH.EQ.1) COAI=COA(IHEM)

      * COMPUTE ALP,DALP AND REORDER ROWS IF GRIDS ARE HEMISPHERIC.

      CALL ALPNM2(ALP,LALP,LM,COAI,EPSI)
      CALL ALPDR2(DALP,ALP,LALP,LM,EPSI)
      IF(KHEM.NE.0) CALL ALPAS2( ALP,LALP,LM,WRKS)
      IF(KHEM.NE.0) CALL ALPAS2(DALP,LALP,LM,WRKS)

      * COMPUTE GRID POINT VALUES FROM SPECTRAL COEFF.

      CALL MHEXPW(PG,CG,TG,ESG,UG,VG,PSDLG,PSDPG,PRESSG,ILH,ILM,
     1             P,C,T,ES,U,V,PS,LA,LAW,LRS,LPA,LRU,LRV,LM,ILEV,LEVS,
     2                        ILG,KHEM,ALP,DALP,LALP,WRKS)
      DO 420 IK=1,ILG
  420 PRESSG(IK)=EXP(PRESSG(IK)+PSMNR2)

      * CALCULATE DYNAMIC PARTS OF RHS.

      CALL STBADJ(AMBDPG,TG,AMBDA,STAMS,STAIS,
     1       ILM,ILEV,ILEVP,ILG,NUPS,WET,EXTRAP)

      CALL VRTIGW(PUTG,PVTG,TUTG,TVTG,EG,SDSG,PEETG,OMEGAG,
     1             PG,CG,TG,UG,VG,PSDLG,PSDPG,ILM,ILG,ILEV,LEVS,
     2             AMBDPG,AMBDA,TMEAN,S,SH,SF,SHF,DS,DEL,SIAI,BETA,
     3              SUSG,SVSG,ESG,ESTG,LTBS,RCPSHD,AVERT)
      CALL EKLAT(TOTK,EG,PRESSG,ILG,ILEV,WOCS(IHEM),KHEM)

      * WET CONVECTIVE ADJUSTMENT AND CALCULATION OF PRECIP.

      IJ=(IHGG-1)*ILG1+1
      IF(.NOT.WET) GO TO 300
      CALL PCPADJ(PCPPAK(IJ),STBPAK(IJ),ILG1,ESG,TG,OMEGAG,PRESSG,ILM,
     *             TMEAN,SH,STAMS,ILEV,LEVS,LTBS,NUPS,NSUPS,KOUNT,DEL)
      PCPPAK(IJ+ILG1)=PCPPAK(IJ)
      STBPAK(IJ+ILG1)=STBPAK(IJ)

      * PHYSICAL EFFECTS IF REQUESTED.

  300 IF(KDR) CALL SFDRAG(PUTG,PVTG,UG,VG,TG,TMEAN,DRPAK(IJ),ILM,ILEV,
     *ILG,GRAV,A,PGAS,DS,SIAI)
      IF(KWT.AND.KCV) CALL SEAFLX(TG,ESG,UG,VG,TMEAN,ILM,ILG,ILEV,
     *LEVS,DT,SIAI,SEACON,RGOCP,SH,WTPAK(IJ),CVPAK(IJ))

      CALL VEMFLX(PUTG,PVTG,UG,VG,RK,DS,ILM,ILEV,ILG)

      * CONVERT GRID POINT VALUES BACK TO SPECTRAL COEFF.
```

```
      CALL MHANLW(PT,CT,PEET,SDS,TDUM,EST,ESDUM,PRESS,LA,
     1                 PLTG,PVTG,TUTG,TVTG,PEETG,SDSG,EG,TG,SUSG,SVSG,
     2            ESTG,ESG,PRESSG,ILH,ILM,LRS,LRA,LM,
     3       ALP,DALP,LALP,WRKS,W(IHEM),WOCS(IHEM),ILG,KHEM,ILEV,LEVS)

      * ADD DELSQ(KE) TO DIVERGENCE TENDENCY.

      CALL DELEK(CT,LA,LRS,LM,EG,ILH,ILEV,ALP,DALP,LALP,KHEM,
     1           IR,ILG, WOCS(IHEM),WRKS)
  700 CONTINUE
  720 CONTINUE

      * SECTION 4 - END OF LATITUDE LOOP.

      * RESTORE RELATIVE VORTICITY. WRITE OUT UNSTABLE POINTS.

      DO 740 L=1,ILEV
      N=(L-1)*LA+2-KSYM
  740 P(N)=P2(L)
      IF(NTEST.NE.0)  WRITE(6,6040) NUPS,NSUPS

      * RESTORE CONVECTIVELY CORRECTED T,ES FROM TDUM,ESDUM.

      CALL SETL(T,TDUM,LA,ILEVP)
      IF(WET) CALL SETL(ES,ESDUM,LA,LEVS)

      * CALCULATE NEW PEE FROM THE NEW TEMPERATURES.

      CALL BPFT(PEE,T,PS,PHIS,LA,LRS,LM,ILEV,TMEAN,RGAS,SF)

      * IF FWD STEP, SET PREVIOUS VALUES TO CURRENT VALUES.

      IF(IFDIFF.EQ.1) CALL SETOLD
     1           (PM,CM,PEEM,ESM,PSM,P,C,PEE,ES,PS,LA,ILEV,LEVS)

      * ADD LINEAR TERMS TO CT, PEET.

      CALL LNER(CT,PEET,SDS,CM,PEEM,LA,LPS,LM,ILEV,DT,RGAS,SF)

      * SECTION 5 - OUTPUT SECTION.

      * CALCULATE ENERGY DIAGNOSTICS IF REQUESTED.

  800 IF(NOGR.GT.0.OR.IEPR.GT.0.OR.DIVCH.GT.0.) CALL ENOUT
     1                 (P,C,T,U,PS,PHIS,PRESS,TMEAN,TMEAN,SF,DS,
     2                 TOTK,IR,ILEV,KHEM,ETOTS,LTOT,NOGR,IEPR,DIVCH)

      * PRINT COEFF AND/OR MAPS EVERY IPR STEPS IF REQUESTED.

      CALL POUT(P,C,T,PHI,PRESS,PHIS,ES,LA,LRS,LRA,LM,ILEV,LEVS,
```

```
 1          PEE,PS,PEEMN,PSMN,TMEAN,TMEAN,RGAS,IPR,LC,KSYM,
 2              GLL,ILG1,ILATH,COA,KHEM,ALP,EPSI,LALP,WRKS,WRKL)

  * SAVE FORECAST ON FILE NF2 IF REQUESTED.

  CALL SAVPRG(NF2,PHI,U,V,ES,PHIS,PS,PEE,TMEAN,PEEMN,PSMN,
 1              RGAS,LA,LAW,LRS,LRA,LRU,LRV,LM,ILEV,LEVS,KSYM,
 2              KSTART,KTOTAL,KOUNT,IPRG,GLL,GLL,
 3              ILG1,ILATH,COA,KHEM,ALP,EPSI,LALP,WRKS,WRKL)

  * SAVE PRECIP ON FILE NFPCP EVERY IPCP STEPS IF REQUESTED.

  CALL SAVPCP(PCPPAK,STBPAK,ILG1,ILATH,KHEM,NPPCM,
 1              KOUNT,KSTART,IPCP,NFPCP,NPPCP,GLL,WRKS)

  * SECTION 6 - PERFORM ONE IMPLICIT TIMESTEP.

  IF(KOUNT.GE.KTOTAL) RETURN
  IF(KOUNT.LE.KSTART+1) CALL ABCII(DT, A, ILEV, IR)
  CALL RHSSI(PEEBAR,PEET,SDS,LA,LRS,LM,ILEV,SF,DT)

  * GET NEW PS,P,C,PEE. COMPUTE NEW T.

  CALL NEWPS(PS,PSM,PEET,PEEBAR,LA,LRS,LM,ILEV,AMBDA,SF,AVERT,
 1                         RGAS,DT,FPS,IFDIFF)
  CALL NEWP (P,PM,PT,LA,LRA,LM,ILEV,DT,FP,IFDIFF)
  CALL NEWC(C,CM,CT,PEEBAR,LA,LRS,LM,ILEV,ASQ,
 1                      DIFUSD,DT,FC,IFDIFF,KHEM)
  CALL NEWBP(PEE,PEEM,PEEBAR,LA,LRS,LM,ILEV,FPEE,IFDIFF)
  IF(WET) CALL NEWES(ES,ESM,EST,LA,LRS,LM,LEVS,ASQ,
 1              DIFUSS,DT,FS,IFDIFF,KHEM)
  CALL TFBP(T,PEE,PS,PHIS,LA,LRS,LM,ILEV,TMEAN,RGAS,SF,EXTRAP)
  DT=DEET
  IFDIFF=0
  KOUNT=KOUNT+1
  GO TO 200

6005 FORMAT(1H1//32H P.E. SPECTRAL MODEL - VERSION 6)
6010 FORMAT(7H0 SPIMW,15I5)
6015 FORMAT(33H0LRS,LRA,LRU,LRV,LALP,LM,LA,LAW =,8I5)
6012 FORMAT(7H0 SPIMW,1P10E12.3)
6020 FORMAT(12H0END OF STEP,I5,6H   HR=,F8.2,6H  TIME,F8.2)
6040 FORMAT(' ',45X,'NUPS = ',I5,'   NSUPS = ',I5)
  END
```

```
      SUBROUTINE STBADJ(AMBDPG,TG,AMBDA,STAMS,STAIS,ILM,ILEV,ILEVP,
     1         ILONG,NUPS,WET,EXTRAP)

      * THIS SUBROUTINE IS TO BE CALLED BEFORE VERTIG IN THE
      *   DRY MULTILEVEL SPECTRAL MODEL.
      * COMPUTE PRIMED STATIC STABILITIES IN AMBDPG AND
      * CORRECT THE VERTICAL TEMPERATURE PROFILES IN TG
      * FOR ILONG POINTS ABOUT ONE GAUSSIAN LATITUDE CIRCLE.

      DIMENSION AMBDPG(ILM,1),TG(ILM,1)
      DIMENSION STAMS(ILEVP,1),STAIS(ILEVP,1)
      DIMENSION AMBDA(1)

      LOGICAL WET

      ILEVM=ILEV-1

      DO 300 JH = 1, ILEV
      IH1 = JH
      IH2 = JH + 1
      DO 300 IK=1,ILONG

      * CALCULATION OF PRIME STATIC STABILITY

      AMBDPG(IK,JH)=0.0
      DO 300 IH=IH1,IH2
      AMBDPG(IK,JH)=AMBDPG(IK,JH)+STAMS(IH,JH)*TG(IK,IH)

  300 CONTINUE

      IF(WET)  RETURN

      * DRY CONVECTIVE ADJUSTMENT


      DO 600 IK=1,ILONG

      DO 30 JH = 1, ILEV
      * CALCULATION OF TOTAL STATIC STABILITY  TAMBDA
      TAMBDA = AMBDPG(IK,JH) + AMBDA(JH)
      IF(TAMBDA.GE.0.)  GO TO 30

      * DAMBDA IS THE STABILITY CORRECTION ADDED TO AMBDPG
      DAMBDA=-TAMBDA*1.0001
      DAMBDA=DAMBDA+10.
      AMBDPG(IK,JH)=AMBDPG(IK,JH)+DAMBDA
      * CALCULATION OF THE ADJUSTED TEMPERATURES
      DO 20 IH=1,ILEV
   20 TG(IK,IH)=TG(IK,IH)+DAMBDA*STAIS(JH,IH)
      TG(IK,ILEVP)=TG(IK,ILEV)*EXTRAP
      NUPS=NUPS+1
```

```
   30 CONTINUE
  600 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE STMCAL (STAMAT,STAINV,S,SH,SF,SHF,RGOCP,RGAS,GRAV,
     1                   GAMMA,ILEV,ILEVP,EXTRAP)

*  CALCULATE THE DRY CONVECTIVE ADJUSTMENT MATRIX "STAMAT" AND
*  ITS INVERSE "STAINV".  TO INSURE CONSERVATIVE ENERGY
*  REDISTRIBUTION, THE RIGHT HAND COLUMN (STAMAT(I,ILEV+1)) IS
*  FILLED WITH AN INTEGRATING VECTOR WHICH DEFINES THE ILEV+1
*  ELEMENT OF AMBDA AS A CONSTANT WHICH DEPENDS ONLY UPON THE
*  TOTAL POTENTIAL ENERGY IN THE VERTICAL

      DIMENSION S(1),SH(1),SF(1),DEL(1)
      DIMENSION STAMAT(ILEVP,1),STAINV(ILEVP,1),SHF(1)
      ILEVM = ILEV-1

      EXTRAP=(1./S(ILEV))**(RGAS*GAMMA/GRAV)
      CP=RGAS/RGOCP
      DO 10 J = 1, ILEVP
      DO 10 I = 1, ILEVP
   10 STAMAT(I,J) = 0.

      DO 1 I=1,ILEVM
      STAMAT(I,I)=(RGOCP*0.5+1./SF(I))/SH(I)
    1 STAMAT(I+1,I)=(RGOCP*0.5-1./SF(I))/SH(I)
      STAMAT(ILEV,ILEV)=(RGOCP-RGAS*GAMMA/GRAV)
      DO 2 I = 2, ILEVM
    2 STAMAT(I,ILEVP)=S(I+1)-S(I-1)
      STAMAT(1,ILEVP)=S(2)-S(1)
      STAMAT(ILEV,ILEVP)=1.-S(ILEVM)
      STAMAT(ILEVP,ILEVP)=1.-S(ILEV)
      DO 3 I=1,ILEVP
    3 STAMAT(I,ILEVP)=0.5*STAMAT(I,ILEVP)*CP
      CALL MTXINV(STAINV,STAMAT,ILEVP)

      RETURN
      END
```

```
      SUBROUTINE STP6(NF1,NF12,NF2,PR,LV,SIG,NK,GZ,T,U,V,ES,LEVS,LGG,
     1               ILG,NLAT,NSTEPS, MAPS,MPSP,MPGZ,MPT,MPW,MPWSP,MPES,
     2                    IW,JW,LL,MM,NPSP,NPGZ,NPT,NPW,NPES)

 * CONVERTS NK LEVELS OF FORECAST GAUSSIAN GRIDS TO LV PRESSURE LEV

 * INPUT UNITS GZ,GZS=(M/SEC)**2, LNSP=(MB).
 *             T=(DEG K),  U,V=(M/SEC)/(EARTH RADIUS)

 * GZ IS A WORK FIELD FOR NK+1 GRID FIELDS OF FORECAST PHI(SIGMA)
 *                        AND LV GRID FIELDS OF FORECAST PHI(PRES).
 * T IS FOR LV LEVELS OF TEMPERATURE.
 * U,V ARE FOR WINDS. THEY CAN BE EQUIVALENCED TO GZ AND T.
 * PR CONTAINS THE PRESSURE LEVELS IN MILLIBARS.
 * SIG CONTAINS THE SIGMA LEVELS.

 * MPSP,MPGZ,MPT,MPW,MPWSP,MPES ARE MAP CONTROLS FOR
 *   MSL PR,GZ,TEMP,(U,V), WIND SPEED, DEW POINT DEPRESSION.
 * IF MAPS=0 CONTROL CARDS WERE NOT READ. NO MAPS ARE PRODUCED.

      DIMENSION GZ(LGG,1),T(LGG,1),U(LGG,1),V(LGG,1),ES(LGG,1)
      DIMENSION PR(LV),SIG(NK)
      DIMENSION MPGZ(1),MPT(1),MPW(1),MPWSP(1),MPES(1)

 * SCM WORK FIELDS.

      PARAMETER $ILV=15,$LEV=0,$LV=10,$ILT=52,$ILG=64,$ILTH=26,$IR=20
      PARAMETER $ILP1=$ILV+1
      PARAMETER $LAW=$ILT*($ILG+1)

      DIMENSION PRL($LV),F($ILP1),G($ILP1),SG($ILP1),SGE($ILP1)
      DIMENSION WRKS(130)

 * LCM WORK FIELDS.

      COMMON/LCM1/PS($LAW),PGG($LAW),GGP($LAW),WRKL(130)

      LOGICAL OK

 * SET CONSTANTS.

      RGOCP=2./7.
      RGAS=287.
      GRAV = 9.80616
      ERAD=6.371E+6
      ILG1=ILG+1
      NSGL=NK+1

 * SET SG TO LOG(SIGMA) AND PR TO LOG(PRESSURE).
 * SGE IS LOG(SIGMA) ON THE EVEN LEVELS.
```

```
      DO 15 N=1,NK
   15 SG(N)=ALOG(SIG(N))
      SG(NSGL)=0.
      DO 16 N=1,NK
   16 SGE(N)=.5*(SG(N)+SG(N+1))
      DO 17 L=1,LV
   17 PRL(L)=ALOG(PR(L))

      * READ INITIAL LN(SF.PRES.) FROM NF1.

      ID = 0
      REWIND NF1
      CALL RSGGP(PGG,ILG1,NLAT,NF1,ID,4HLNSP,1,OK,GGP,WRKS)
      IF(.NOT.OK) RETURN

      * READ INITIAL PRESSURE LEVELS OF GZ FROM NF12. PUT INTO V.

      REWIND NF12
      ID = 0
      DO 251 L=1,LV
      CALL RSGGP(  V(1,L),ILG1,NLAT,NF12,ID,4H  GZ,L,OK,GGP,WRKS)
      IF(.NOT.OK) RETURN
  251 CONTINUE

      * READ MOUNTAINS AND FORECAST OF LNSP,PHI FROM FILE NF1.

  305 ID=NSTEPS
      REWIND NF1
      CALL RSGGP(GZ(1,NSGL),ILG1,NLAT,NF1, 0,4H GZS,1,OK,GGP,WRKS)
      CALL RSGGP(PS,ILG1,NLAT,NF1,ID,4HLNSP,1,OK,GGP,WRKS)
      DO 310 N=1,NK
      CALL RSGGP(GZ(1,N),ILG1,NLAT,NF1,ID,4H  GZ,N,OK,GGP,WRKS)
      IF(.NOT.OK) GO TO 505
  310 CONTINUE

      * INTERPOLATE FROM SIGMA LEVELS IN SG TO PRESSURE LEVELS IN PR.
      * THE CALCULATIONS ARE DONE IN PLACE FOR GZ, TEMPERATURES ARE IN T
      * INITIAL PRESSURE LEVELS OF GZ ARE IN V.

      GLAPSE = 6.5E-3/GRAV
      EXTRAP=1.0
      KIND=1
      CALL SGTPRE(GZ,T,GZ,PS,LGG,LGG,PRL,LV,SG,F,G,NSGL,GLAPSE,EXTRAP,
     1            KIND,V,PGG)

      * MAP GZ IF REQUESTED AND SAVE ON NF2 WITH PACKING DENSITY NPGZ.

      DO 410 L=1,LV
      IF(MAPS.NE.0) CALL FCONW2
     1           (GZ(1,L),6.0,1./98.0616,ILG1,NLAT,IW,JW,LL,MM,NPGZ(L))
```

```
      CALL WSGGP(GZ(1,L),ILG1,NLAT,NF2,ID,4H  GZ,L,NPGZ,GGP,WRKS)
  410 CONTINUE

      * T ACTUALLY CONTAINS -RGAS*T. CONVERT TO DEG K.
      * MAP  T IF REQUESTED AND SAVE ON NF2 WITH PACKING DENSITY NPT.

      DO 420 L=1,LV
      DO 415 I=1,LGG
  415 T(I,L)=T(I,L)*(-1./RGAS)
      IF(MAPS.NE.0) CALL FCONW2
     1            (T(1,L),5.,1.,ILG1,NLAT,IW,JW,LL,MM,MPT(L))
      CALL WSGGP( T(1,L),ILG1,NLAT,NF2,ID,4H   T,L,NPT ,GGP,WRKS)
  420 CONTINUE

      * COMPUTE MSL PRESSURE (USING THE FOLLOWING ASSUMPTIONS)...
      *  A) .6*(DRY LAPSE RATE) FROM LOWEST TEMPERATURE TO 1000 MB,
      *  B) MSL TEMPERATURE EQUALS 1000MB TEMPERATURE.
      * MAP IF REQUESTED AND SAVE ON NF2 WITH PACKING DENSITY NPSP.

      DO 440 I=1,LGG
      T1000=T(I,LV)*(1000./PR(LV))**(.6*RGOCP)
      GZ1000=GZ(I,LV)-.5*RGAS*(T1000+T(I,LV))*ALOG(1000./PR(LV))
      GGP(I)=1000.*EXP(GZ1000/(RGAS*T1000))
  440 CONTINUE
      IF(MAPS.NE.0) CALL FCONW2
     1            (GGP,5.,1.,ILG1,NLAT,IW,JW,LL,MM,MPSP)
      CALL WSGGP( GGP   ,ILG1,NLAT,NF2,ID,4HSFPR,1,NPSP,GGP,WRKS)

      * READ FORECAST GRIDS OF U,V FROM FILE NF1.

  505 ID=NSTEPS
      DO 510 N=1,NK
      CALL RSGGP( U(1,N),ILG1,NLAT,NF1,ID,4H   U,N,OK,GGP,WRKS)
      IF(.NOT.OK) GO TO 605
      CALL RSGGP( V(1,N),ILG1,NLAT,NF1,ID,4H   V,N,OK,GGP,WRKS)
      IF(.NOT.OK) GO TO 605
  510 CONTINUE

      * INTERPOLATE FROM SIGMA LEVELS IN SG TO PRESSURE LEVELS IN PR.

      GLAPSE=0.
      EXTRAP=0.
      KIND=0
      CALL SGTPRE(U,DUMMY,U,PS,LGG,LGG,PRL,LV,SG,F,G,NK,GLAPSE,EXTRAP,
     1            KIND,DUMMY,DUMMY)
      CALL SGTPRE(V,DUMMY,V,PS,LGG,LGG,PRL,LV,SG,F,G,NK,GLAPSE,EXTRAP,
     1            KIND,DUMMY,DUMMY)

      * MAP U,V IF REQUESTED AND SAVE ON NF2 WITH PACKING DENSITY NPW.

      DO 520 L=1,LV
```

```
      IF(MAPS.NE.0) CALL FCONW2
     1             (U(1,L),10.,ERAD,ILG1,NLAT,IW,JW,LL,MM,MPW(L))
      CALL WSGGP( U(1,L),ILG1,NLAT,NF2,ID,4H   U,L,NPW ,GGP,WRKS)
      IF(MAPS.NE.0) CALL FCONW2
     1             (V(1,L),10.,ERAD,ILG1,NLAT,IW,JW,LL,MM,MPW(L))
      CALL WSGGP( V(1,L),ILG1,NLAT,NF2,ID,4H   V,L,NPW ,GGP,WRKS)
  520 CONTINUE

      * WIND SPEED IS MAPPED IF REQUESTED.

      DO 530 L=1,LV
      IF(MPWSP(L).EQ.0) GO TO 530
      DO 526 I=1,LGG
  526 GGP(I)=SQRT(U(I,L)**2+V(I,L)**2)
      IF(MAPS.NE.0) CALL FCONW2
     1             (GGP,10.,ERAD,ILG1,NLAT,IW,JW,LL,MM,MPWSP(L))
  530 CONTINUE


      * READ FORECAST GRIDS OF ES FROM FILE NF1.

  605 ID=NSTEPS
      IF(LEVS.EQ.0) RETURN
      DO 610 K=1,LEVS
      N=(NK-LEVS) + K
      CALL RSGGP( U(1,N),ILG1,NLAT,NF1,ID,4H  ES,N,OK,GGP,WRKS)
      IF(.NOT.OK) RETURN
  610 CONTINUE

      * FILL IN MISSING LEVELS OF ES.

      LDUM=NK-LEVS
      IF(LDUM.EQ.0) GO TO 618
      DO 615 K=1,LDUM
      DO 615 I=1,LGG
  615 ES(I,K)=ES(I,LDUM+1)

      * INTERPOLATE FROM SIGMA LEVELS IN SGE TO PRESSURE LEVELS IN PR.

  618 GLAPSE=0.
      EXTRAP=0.
      KIND=0
      CALL SGTPRE(ES,DUMMY,ES,PS,LGG,LGG,PRL,LV,SGE,F,G,NK,GLAPSE,
     1              EXTRAP,KIND,DUMMY,DUMMY)

      * MAP ES IF REQUESTED AND SAVE ON NF2 WITH PACKING DENSITY NPES.

      DO 620 L=1,LV
      IF(MAPS.NE.0) CALL FCONW2
     1             (ES(1,L),5.0, 1.0,ILG1,NLAT,IW,JW,LL,MM,MPES(L))
      CALL WSGGP(ES(1,L),ILG1,NLAT,NF2,ID,4H  ES,L,NPES,GGP,WRKS)
```

```
620 CONTINUE
    RETURN
    END
```

```
      SUBROUTINE TERP1 (X,FX,F,G,Y,ACC,NN)
      DIMENSION F(NN),G(NN),Y(NN)

      * GIVEN THE VALUES OF A MONOTONIC FUNCTION F AND THE VALUES OF ITS
      *   DERIVATIVE G AT NN POINTS Y(1) TO Y(NN), THIS ROUTINE FINDS X
      *   POINT AT WHICH F ASSUMES THE SPECIFIED VALUE FX.

      * NOTE  AT INPUT A FIRST GUESS SHOULD BE PROVIDED FOR X.
      * NOTE  WE ASSUME FX LE F(1)

      IF (FX.LE.F(NN)) GO TO 40
      IF (FX.GT.F(1) ) WRITE(6,600)

      * INTERPOLATION

      DO 10 N=1,NN
      IF (FX.GT.F(N)) GO TO 20
   10 CONTINUE

   20 F1=F(N-1)
      F0=F(N)
      G1=G(N-1)
      G0=G(N)
      Y1=Y(N-1)
      Y0=Y(N)
      DY=Y1-Y0
      A=+F1/DY
      B=-F0/DY
      C=(G0+G1)/DY**2 - 2.*(F1-F0)/DY**3
      CD=(Y1*G0+Y0*G1-(A+B)*(Y1+Y0))/DY**2

      * NEWTON FORMULA ITERATION LOOP

   30 P=X-Y0
      Q=X-Y1
      R=C*X-CD
      ER=A*P+B*Q+P*Q*R-FX
      IF (ABS(ER).LT.ACC) RETURN
      DER=A+B+P*R+Q*R+C*P*Q
      X = X -ER/DER
      GO TO 30

   40 IF (FX.EQ.F(NN)) GO TO 60

      * EXTRAPOLATION

      F1=F(NN-1)
      F0=F(NN)
      G1=G(NN-1)
      G0=G(NN)
      Y1=Y(NN-1)
```

```
      Y0=Y(NN)
      ROOT = G0**2 - 2.*(G0-G1)*(F0-FX)/(Y0-Y1)

*     IF  ROOT GE 0  USE QUADRATIC EXTRAPOLATION
*     IF  ROOT LT 0  USE LINEAR FORMULA

      IF (ROOT.GE.0.) GO TO 50
      X=Y0+(FX-F0)/G0
      RETURN

   50 ROOT=SQRT(ROOT)
      X=Y0-(Y0-Y1)*(G0+ROOT)/(G0-G1)
      RETURN

   60 X=Y(NN)
      RETURN

  600 FORMAT(12H1TERP1 ERROR)
      END
```

```
      SUBROUTINE TERP2 (FX,GX,X,F,G,Y,NN,RLAPSE)
      DIMENSION F(NN),G(NN),Y(NN)

      * GIVEN A FUNCTION F AND ITS FIRST DERIVATIVE G AT A SET OF NN
      *   UNEVENLY SPACED POINTS Y, THIS ROUTINE CALCULATES FX AND GX,
      *   THE VALUES OF F AND G AT THE SPECIFIED POINT X.

      * RLAPSE IS A LAPSE RATE USED FOR EXTRAPOLATING.

      * EXTRAPOLATION

      IF (X.GT.Y(1)) GO TO 10
      GX=G(1)
      FX=F(1)+(X-Y(1))*G(1)
      RETURN

   10 IF (X.LT.Y(NN)) GO TO 20
      GX=G(NN)+(X-Y(NN))*RLAPSE
      FX=F(NN)+.5*(X-Y(NN))*(GX+G(NN))
      RETURN

      * INTERPOLATION

   20 DO 30 N=2,NN
      IF (X.LT.Y(N)) GO TO 40
   30 CONTINUE

   40 FA=F(N-1)
      FB=F(N)
      GA=G(N-1)
      GB=G(N)
      A=Y(N-1)
      B=Y(N)

      * GIVEN FA,FB,GA AND GB THE VALUES OF F AND G AT POINTS A AND B
      *   RESPECTIVELY, THIS CUBIC INTERPOLATION ALGORITHM GIVES FX AND
      *   GX THE VALUES OF THE FUNCTIONS F AND G AT THE INTERMEDIATE POI

      D=B-A
      E=X-.5*(A+B)
      R=.125*D*D-.5*E*E
      FM0=.5*(FA+FB)
      FM1=(FB-FA)/D
      FM2=(GB-GA)/D
      FM3=(GB+GA-FM1-FM1)/(D*D)
      FL2=FM2+2.*E*FM3
      FL1=FM1+E*FL2
      FL0=FM0+E*FM1
      FX=FL0-R*FL2
      GX=FL1-2.*R*FM3
      RETURN
```

END

```
      SUBROUTINE TERP2E(FX,GX,X,F,G,Y,NN,EXTRAP,GLAPSE,FI,XI)

      * GIVEN A FUNCTION F AND ITS FIRST DERIVATIVE G AT A SET OF NN
      *   UNEVENLY SPACED POINTS Y, THIS ROUTINE CALCULATES FX AND GX,
      *   THE VALUES OF F AND G AT THE SPECIFIED POINT X.

      * EXTRAP=1. FOR HEIGHTS AND TEMP, 0. FOR WINDS.
      * RLAPSE IS A LAPSE RATE USED FOR EXTRAPOLATING.

      DIMENSION F(NN),G(NN),Y(NN)

      * EXTRAPOLATION ABOVE TOP LEVEL.

      IF (X.GT.Y(1)) GO TO 10
      GX=G(1)
      FX=F(1)+(X-Y(1))*G(1)
      RETURN

      * EXTRAPOLATION BELOW SIGMA=1.

   10 IF (X.LT.Y(NN)) GO TO 20
      TLAPSE = .142857*EXTRAP
      GX = G(NN)*(1. + TLAPSE*(X-Y(NN)))
      FX = F(NN) + .5*(X-Y(NN))*(GX+G(NN))*EXTRAP
      IF(EXTRAP.LE.0.) RETURN
      IF(XI-Y(NN).LT..001) RETURN
      TMI = (F(NN)-FI)/XI
      FX = F(NN) - TMI*X
      GX = G(NN) + GLAPSE*(F(NN)-FI)
      RETURN

      * INTERPOLATION

   20 DO 30 N=2,NN
      IF (X.LT.Y(N)) GO TO 40
   30 CONTINUE

   40 FA=F(N-1)
      FB=F(N)
      GA=G(N-1)
      GB=G(N)
      A=Y(N-1)
      B=Y(N)

      * GIVEN FA,FB,GA AND GB THE VALUES OF F AND G AT POINTS A AND B
      *   RESPECTIVELY, THIS CUBIC INTERPOLATION ALGORITHM GIVES FX AND
      *   GX THE VALUES OF THE FUNCTIONS F AND G AT THE INTERMEDIATE POI

      D=B-A
      E=X-.5*(A+B)
      R=.125*D*D-.5*E*E
```

```
FM0=.5*(FA+FB)
FM1=(FB-FA)/D
FM2=(GB-GA)/D
FM3=(GB+GA-FM1-FM1)/(D*D)
FL2=FM2+2.*E*FM3
FL1=FM1+E*FL2
FL0=FM0+E*FM1
FX=FL0-R*FL2
GX=FL1-2.*R*FM3
RETURN

END
```

```
      SUBROUTINE TFBP(T,PEE,PS,PHIS,LA,LRS,LM,ILEV,TMEAN,RGAS,SF,EXTRAP)

*     CALCULATES T FROM PEE,PS,PHIS.

*     IF MODEL IS HEMISPHERIC T,PEE,PS,PHIS ARE SYMMETRIC.
*     EACH LEVEL IS DIMENSIONED (LRS,LM).
*     EACH LEVEL IS SEPARATED BY LA COMPLEX WORDS.
*     T AND PEE MAY BE EQUIVALENCED.

      COMPLEX T(LA,1),PEE(LA,1),PS(1),PHIS(1)

      DIMENSION TMEAN(1),SF(1)

      COMPLEX PHIS1,PSURF

      PHIS1=PHIS(1)
      PHIS(1)=(0.0,0.0)
      ILEVM=ILEV-1
      ILEVP=ILEV+1
      CT=2./((1.+EXTRAP)*RGAS*SF(ILEV))

      DO 30 M=1,LM
      MR=(M-1)*LRS
      DO 30 N=1,LRS
      IL=MR+N

      PSURF=RGAS*TMEAN(ILEVP)*PS(IL)+PHIS(IL)
      T(IL,ILEV)=CT*(PEE(IL,ILEV)-PSURF+RGAS*PS(IL)*
     1(TMEAN(ILEVP)-TMEAN(ILEV)))
      T(IL,ILEVP)=EXTRAP*T(IL,ILEV)
      DO 50 IH=1,ILEVM
      IH1=ILEV-IH
   50 T(IL,IH1)=-T(IL,IH1+1)-2.*(PEE(IL,IH1+1)-PEE(IL,IH1)-RGAS*
     *(TMEAN(IH1+1)-TMEAN(IH1))*PS(IL))/(RGAS*SF(IH1))

   30 CONTINUE

      PHIS(1)=PHIS1
      RETURN
      END
```

```
      SUBROUTINE TFGZ(T,PHI,PHIS,LA,LRS,LM,ILEV,SF,RGAS,S,GRAV,GAMMA)

* CALCULATE TEMPERATURES (T) AT ODD LEVELS FROM GEOPOTENTIALS
* AT ODD LEVELS; ASSUME SURFACE TEMPERATURE FROM A LAPSE
* RATE OF "GAMMA"

* IF MODEL IS HEMISPHERIC T,PHI ARE SYMMETRIC.
* EACH LEVEL IS DIMENSIONED (LRS,LM).
* EACH LEVEL IS SEPARATED BY LA COMPLEX WORDS.
* T AND PHI CAN BE EQUIVALENCED.

      COMPLEX T(LA,1),PHI(LA,1),PHIS(1)
      DIMENSION SF(1),S(1)

      ILEVM=ILEV-1
      ILEVP = ILEV + 1
      EXTRAP=1./S(ILEV)**(RGAS*GAMMA/GRAV)
      CON=-2./(RGAS*SF(ILEV)*(1.+EXTRAP))

      DO 1 I = 1, LA
    1 PHI(I,ILEVP)=PHIS(I)

      DO 30 M=1,LM
      MR=(M-1)*LRS
      DO 30 N=1,LRS
      MN=MR+N

      T(MN,ILEV)=(PHI(MN,ILEVP)-PHI(MN,ILEV))*CON
      T(MN,ILEVP)=EXTRAP*T(MN,ILEV)
      DO 20 L=2, ILEV
      L1=ILEVP-L
   20 T(MN,L1)=-T(MN,L1+1)-2.*(PHI(MN,L1+1)-PHI(MN,L1))/(RGAS*SF(L1))
   30 CONTINUE

      RETURN
      END
```

```
      SUBROUTINE TMCAL(TMEAN,AMBDA,S,SH,SF,RGOCP,STAMS,ILEV,ILEVP)

     * SUBROUTINE CALCULATES HORIZONTALLY AVERAGED TEMPERATURES AT
     * THE FULL LEVELS AND ALSO THE HORIZONTALLY AVERAGED STATIC STABILI

      DIMENSION TMEAN(1),AMBDA(1),S(1),SF(1),SH(1),STAMS(ILEVP,1)

      WRITE(6,6012) (TMEAN(L),L=1,ILEVP)

      * SET AMBDA TO AVERAGE STATIC STABILITY AT HALF LEVELS.

      DO 30 JH=1,ILEVP
      AMBDA(JH) = 0.
      DO 30 IH=1,ILEVP
   30 AMBDA(JH) = AMBDA(JH) + STAMS(IH,JH)*TMEAN(IH)
      WRITE(6,6012) (AMBDA(L),L=1,ILEVP)

      RETURN
 6012 FORMAT(7H0 TMCAL,1P10E12.3)
      END
```

```
      SUBROUTINE TSIG (RT,GZ,SG,NN,TOLAPS,SENTR,CON)
      DIMENSION RT(NN),GZ(NN),SG(NN)

      * GIVEN GEOPOTENTIALS, GZ, AT A SET OF (UNEVENLY) SPACED POINTS,
         THIS ROUTINE COMPUTES RT=-RGAS*TEMPERATURE AT THOSE SAME POINT

      * SG MUST CONTAIN THE POINT CO-ORDINATES.
      * BOUNDARY CONDITIONS SPECIFIED BY TOLAPS,SENTR,CON (SEE BELOW).

      DO 10 N=2,NN
   10 RT(N)=(GZ(N)-GZ(N-1))/(SG(N)-SG(N-1))

      A=RT(2)

      NNM=NN-1
      DO 20 N=2,NNM
   20 RT(N)=((SG(N+1)-SG(N))*RT(N)+(SG(N)-SG(N-1))*RT(N+1)) /
     1        (SG(N+1)-SG(N-1))

      * BOUNDARIES

      RT(1)=SENTR*((3.+TOLAPS)*A-(1.+TOLAPS)*RT(2))
      RT(NN)=CON*RT(NN)+(1.-CON)*RT(NNM)

      RETURN
      END
```

```
      SUBROUTINE VEMFLX(PUTG,PVTG,UG,VG,RK,DS,ILM,ILEV,ILONG)

*     CALCULATES VERTICAL EDDY MOMENTUM FLUX EXCEPT FOR CONTRIBUTIONS
*     AT THE GROUND (THE SURFACE DRAG TERM)
*     FOR ILONG POINTS ABOUT ONE GAUSSIAN LATITUDE CIRCLE.
*     EACH ARRAY HAS ILEV LEVELS. MAX VALUE IS 15.

      DIMENSION UG(ILM,1),VG(ILM,1),PUTG(ILM,1),PVTG(ILM,1)
      DIMENSION RK(1),DS(1)
      DIMENSION TAUX(15),TAUY(15)


      ILEVM = ILEV-1
       ILEVP = ILEV+1
      TAUX(1) = 0.
      TAUY(1) = 0.
      TAUX(ILEVP) = 0.
      TAUY(ILEVP) = 0.

      DO 30 IK=1,ILONG

      DO 15 IH=1,ILEVM
      IHP = IH+1
      TAUX(IHP) = RK(IH)*(UG(IK,IHP) - UG(IK,IH))
      TAUY(IHP) = -RK(IH)*(VG(IK,IHP)-VG(IK,IH))
   15 CONTINUE

      DO 20 IH=1,ILEV
      IHP = IH+1
      PUTG(IK,IH) = PUTG(IK,IH)+ (TAUY(IHP)-TAUY(IH))/DS(IH)
      PVTG(IK,IH) = PVTG(IK,IH) + (TAUX(IHP) - TAUX(IH))/DS(IH)
   20 CONTINUE

   30 CONTINUE

      RETURN
      END
```

```
      SUBROUTINE VRTIGW(PLTG,PVTG,TUTG,TVTG,EG,SDSG,PEETG,OMEGAG,
     1          PG,CG,TG,UG,VG,PSDLG,PSDPG,ILM,ILONG,ILEV,LEVS,
     2          AMBDPG,AMBDA,TMEAN,S,SH,SF,SHF,DS,DEL,SIAI,BETA,
     3          SUSG,SVSG,ESG,ESTG,LTBS,RCPSHD,AVERT)

    * CALCULATES DYNAMIC PARTS OF RHS OF SPECTRAL MODEL EQUATIONS
    * FOR ILONG POINTS ABOUT ONE GAUSSIAN LATITUDE CIRCLE.
    * LEVELS IN THIS SUBR ARE NUMBERED DOWN FROM THE TOP
    * STARTING WITH SIGMA = 0.  EVERY SECOND LEVEL DOWN FROM HERE
    * TO THE SURFACE IS CALLED AN EVEN LEVEL. THE OTHERS ARE ODD.


      REAL PUTG(ILM,1),PVTG(ILM,1),TUTG(ILM,1),TVTG(ILM,1)
      REAL EG(ILM,1),SDSG(ILM,1),PEETG(ILM,1),OMEGAG(ILM,1)
      REAL PG(ILM,1),CG(ILM,1),TG(ILM,1),UG(ILM,1),VG(ILM,1)
      REAL   SUSG(ILM,1),SVSG(ILM,1),ESG(ILM,1),ESTG(ILM,1)
      REAL PSDLG(1),PSDPG(1),AMBDPG(ILM,1)

      DIMENSION AMBDA(1),TMEAN(1),S(1),SH(1),SF(1),DS(1)
      DIMENSION RCPSHD(1),DEL(1),SHF(1)

      COMMON/PARAMS/ WW,TW,A,ASQ,GRAV,RGAS,RGOCP,RGOASQ,CPRES

      WORK ARRAYS FOR THIS SUBROUTINE. MAX LEVEL IS $ILV

      PARAMETER $ILV=16,$LEV=0,$LV=10,$ILT=52,$ILG=64,$ILTH=26,$IF=20

      DIMENSION CIGH($ILV),UIGH($ILV),VIGH($ILV),SDH($ILV),DUSH($ILV),
     *DVSH($ILV),TGF($ILV),UGH($ILV),VGH($ILV),CGH($ILV),SDDUS($ILV),
     *SDDVS($ILV),VERTT($ILV),VERTS($ILV),DSSH($ILV),WA($ILV),WB($ILV)
     *,TGH($ILV),VERTH($ILV)

      DATA EPS/.62197/

      ILEVM=ILEV-1
      ILEVP=ILEV+1
      RECCSQ=1./(SIAI**2)
      DO 14 IH=1,ILEVM
      WA(IH) = (SH(IH)-S(IH))/(S(IH+1)-S(IH))
      WB(IH) = (S(IH+1)-SH(IH))/(S(IH+1)-S(IH))
   14 CONTINUE

      DO 900 IK=1,ILONG

    * VERTT = VERTICAL COLUMN OF TEMP AT ODD LEVELS.

      DO 16 IH=1,ILEVP
   16 VERTT(IH) = TG(IK,IH) +TMEAN(IH)

    * (DIV,U,V) BARSIGMA AT EVEN LEVELS.
```

```
      CIGH(ILEV) = -CG(IK,ILEV)*DS(ILEV)
      UIGH(ILEV) = -UG(IK,ILEV)*DS(ILEV)
      VIGH(ILEV) = -VG(IK,ILEV)*DS(ILEV)
      DO 10 IHI=1,ILEVM
      IH=ILEV-IHI
      CIGH(IH) = CIGH(IH+1) - CG(IK,IH)*DS(IH)
      UIGH(IH) = UIGH(IH+1) - UG(IK,IH)*DS(IH)
   10 VIGH(IH) = VIGH(IH+1) - VG(IK,IH)*DS(IH)

      * LAT AND LONG DERIVATIVES OF LNSP. ADVECTION OF LNSP.

      PSDPOC=PSDPG(IK)*RECCSQ
      PSDLOC=PSDLG(IK)*RECCSQ
      VMDPS=UIGH(1)*PSDLOC+VIGH(1)*PSDPOC
      DPSDT=CIGH(1)+VMDPS


      * SIGMADOT AT EVEN LEVELS.

      DO 20 IH=1,ILEVM
      SIG=1.-SH(IH)
      SDH(IH)=SIG*CIGH(1)-CIGH(IH+1)+
     1        (SIG*UIGH(1)-UIGH(IH+1))*PSDLOC+
     2        (SIG*VIGH(1)-VIGH(IH+1))*PSDPOC
   20 CONTINUE
      SDH(ILEV)=AVERT*SDH(ILEVM)

      * D(U,V)/D(SIGMA) AT EVEN LEVELS.

      DO 30 IH=1,ILEVM
      DUSH(IH) = (UG(IK,IH+1) - UG(IK,IH))/DEL(IH)
   30 DVSH(IH) = (VG(IK,IH+1) - VG(IK,IH))/DEL(IH)

      * (DIV,U,V) AVERAGED TO EVEN LEVELS.

      DO 35 IH = 2,ILEV
      UGH(IH-1) = UG(IK,IH-1)*WB(IH-1) + UG(IK,IH)*WA(IH-1)
      VGH(IH-1) = VG(IK,IH-1)*WB(IH-1) + VG(IK,IH)*WA(IH-1)
   35 CGH(IH-1) = CG(IK,IH-1)*WB(IH-1) + CG(IK,IH)*WA(IH-1)
      UGH(ILEV) = BETA*UG(IK,ILEV)
      VGH(ILEV) = BETA*VG(IK,ILEV)
      CGH(ILEV) = BETA*CG(IK,ILEV)

      * VERTICAL MOTION AT EVEN LEVELS.

      DO 36 IH=1,ILEV
      OMG = SH(IH)*(UGH(IH) *PSDLOC + VGH(IH)*PSDPOC)
      OMEGAG(IK,IH) = OMG + SH(IH)*DPSDT + SDH(IH)
   36 CONTINUE

      * TEMPERATURE LOG AVERAGED TO ODD LEVELS.
```

```
      DO 37 IH=1,ILEV
37    TGF(IH)=TG(IK,IH)

      * AND TO EVEN LEVELS

      DO 38 IH=2,ILEV
      TGH(IH)=(SHF(IH)*TG(IK,IH)+SHF(IH-1)*TG(IK,IH+1))/
     *(SHF(IH)+SH(IH-1))
38    VERTH(IH)=(VERTT(IH)*SHF(IH)+VERTT(IH+1)*SHF(IH-1))/
     *(SHF(IH)+SH(IH-1))
      TGH(1)=(TG(IK,1)*ALOG(S(2)/SH(1))+TG(IK,2)*ALOG(SH(1)/S(1)))/
     *SF(1)
      VERTH(1)=(VERTT(1)*ALOG(S(2)/SH(1))+VERTT(2)*ALOG(SH(1)/S(1)))/
     *SF(1)

      * SIGMADOT*D(U,V)/D(SIGMA) AT ODD LEVELS.

      DO 40 IH=2,ILEVM
      SDDUS(IH) = (DEL(IH-1)*SDH(IH)*DUSH(IH) + SDH(IH-1)*DUSH(IH-1)*
     1 DEL(IH))/(DEL(IH) + DEL(IH-1))
40    SDDVS(IH) = (DEL(IH-1)*SDH(IH)*DVSH(IH) + SDH(IH-1)*DVSH(IH-1)*
     1 DEL(IH))/(DEL(IH) + DEL(IH-1))
      SDDUS(1) = S(1)*SDH(1)*DUSH(1)/DS(1)
      SDDVS(1) = S(1)*SDH(1)*DVSH(1)/DS(1)
      SDDUS(ILEV) = DEL(ILEV)*SDH(ILEVM)*DUSH(ILEVM)/DS(ILEV)
      SDDVS(ILEV) = DEL(ILEV)*SDH(ILEVM)*DVSH(ILEVM)/DS(ILEV)

      * NOW BEGIN TO COMPUTE THE RIGHT HAND SIDES.

      DO 51 IH=1,ILEV
      PVTG(IK,IH)=PG(IK,IH)*VG(IK,IH)-SDDUS(IH)
     1              -TGF(IH)*RGOASQ*PSDLG(IK)
      PUTG(IK,IH)=PG(IK,IH)*UG(IK,IH)+SDDVS(IH)
     1              +TGF(IH)*RGOASQ*PSDPG(IK)
51    CONTINUE

      DO 53 IH=1,ILEV
      TUTG(IK,IH)=TGH(IH)*UGH(IH)
53    TVTG(IK,IH)=TGH(IH)*VGH(IH)

      DO 55 IH=1,ILEV
      SDSG(IK,IH)=UG(IK,IH)*PSDLOC+VG(IK,IH)*PSDPOC
55    EG(IK,IH)=UG(IK,IH)**2+VG(IK,IH)**2

      DO 60 IH=1,ILEVM
      RGOCPT=RGOCP*VERTH(IH)
      PEETG(IK,IH)=-TGH(IH)*CGH(IH)-RGOCPT*(UGH(IH)*PSDLOC+
     *VGH(IH)*PSDPOC)-AMBCPG(IK,IH)*SDH(IH)-RGOCP*TGH(IH)*DPSDT
60    CONTINUE
      RGOCPT=RGOCP*VERTH(ILEV)
```

```
      PEETG(IK,ILEV)=-TGH(ILEV)*CGH(ILEV)-RGOCPT*BETA*SDSG(IK,ILEV)
     *-RGOCP*TGH(ILEV)*DPSDT-AVERT*AMBDPG(IK,ILEVM)*SDH(ILEVM)

      * MOISTURE VARIABLES (OMITTED IF LEVS=0).

      IF(LEVS.EQ.0) GO TO 900
      DO 61 IH=1,LEVS
      SUSG(IK,IH) = ESG(IK,IH)*UGH(IH+LTBS)
   61 SVSG(IK,IH) = ESG(IK,IH)*VGH(IH+LTBS)

      DO 64 IH=1,LEVS
   64 VERTS(IH)=ESG(IK,IH)
      CALL DFDSQD(DSSH,VERTS,SH(LTBS+1),RCPSHD,LEVS)

      DO 70 IH=1,LEVS
      TD =VERTT(IH+LTBS)-ESG(IK,IH)
      ESTG(IK,IH) =        SDH(IH+LTBS)/SH(IH+LTBS)
     1                     +(UIGH(1)+UGH(IH+LTBS))*PSDLOC
     2                     +(VIGH(1)+VGH(IH+LTBS))*PSDPOC + CIGH(1)
      ESTG(IK,IH) = ESTG(IK,IH)*
     1                 RGOCP*(VERTT(IH+LTBS)-TD*TD/(EPS*HTVOCP(TD)))
      ESTG(IK,IH) = ESTG(IK,IH) + ESG(IK,IH)*CGH(IH+LTBS)
     1                          - SDH(IH+LTBS)*DSSH(IH)
   70 CONTINUE
  900 CONTINUE
      RETURN
      END
```

AD-A058 244 NAVAL RESEARCH LAB WASHINGTON D C F/G 4/2
THE MODIFIED CANADIAN SPECTRAL FORECAST MODEL: DISCUSSION AND D--ETC(U)
MAR 78 W W JONES, R V MADALA, M R SCHOEBERL
UNCLASSIFIED NRL-MR-3698 SBIE-AD-E000 193 NL

3 OF 3

AD A058244

END
DATE FILMED
10-78
DDC

```
      SUBROUTINE WETCON(ST,RCPSHD,S,SH,SF,DEL,RGOCP,ILEVM,LEVS)
*     COMPUTES ST, THE STABILIZATION MATRIX SUCH THAT
*                         TT = ST1*T(LOWER) + ST2*T(UPPER)
*                        GAM = TT +ST3*(T(LOWER)-T(UPPER))
*                   DT(UPPER) =ST4*(GAC-GAM)
*                   DT(LOWER) =ST5*DT(UPPER)
*     COMPUTES RCPSHD, THE RECIPROCAL OF D(SH)

      DIMENSION ST(ILEVM,6),RCPSHD(1),S(1),SH(1),SF(1)

*     PARAMETERS USED BY  FUNCTION  HTVOCP

      COMMON/HTCP/T1S,T2S,AI,BI,AW,BW,SLP

*     PARAMETERS USED BY  FUNCTION  DEWPNT,  SPCHUM,  DELT

      COMMON/EPS/A,B,EPS1,EPS2

*     PARAMETERS USED BY  FUNCTION  GAMSAT

      COMMON/GAMS/EPSS,CAPA

*  PARAMETERS USED IN  CONADJ

      COMMON/ADJPCP/HC,HF,HM,AA,DEPTH,LHEAT,MOIADJ,MOIFLX
      DIMENSION DEL(1)

*  ILEVM IS JUST ILEV-1.  LTBS IS THE NUMBER OF MISSING WET LEVELS.

      LTBS=(ILEVM+1)-LEVS

      THIRD=1./3.
      DO 10 I=1,ILEVM
      X = 1./(SF(I) + SF(I+1))
      ST(I,1) = X*(SF(I+1) + 2.*SF(I))/3.
      ST(I,2) = X*(SF(I) + 2.*SF(I+1))/3.
      ST(I,3) = 2.*X/RGOCP
      ST(I,5) = -DEL(I+1)/DEL(I)
   10 ST(I,6)=(S(I)/SH(I+1))**THIRD

      DO 20 I=1,ILEVM
   20 ST(I,4)=1./(ST(I,2)-ST(I,3)+ST(I,5)*(ST(I,1)+ST(I,3)))

      LEVSM=LEVS-1
      DO 30 I=1,LEVSM
   30 RCPSHD(I)= 1.0 / (SH(I+LTBS+1)-SH(I+LTBS))

      RAUW=1.E+3
      GRAV=9.80616
      DEPTH = 1./(RAUW*GRAV)
```

```
CP=1004.5
CAPA=RGOCP
T1S=273.16
T2S=258.16
AW=3.15213E+6/CP
BW=2.38E+3/CP
AI=2.88053E+6/CP
BI=0.167E+3/CP
SLP=1./(T1S-T2S)
A=21.656
B=5418.
EPS1=0.622
EPS2=0.378
EPSS=EPS1
AA=0.0
IF(HM.LT.1.)  AA=1./(6.*(1.-HM)**2)

RETURN
END
```

```
      SUBROUTINE WSGGP(GG,NLG,NLAT,NF,ID,NAME,N,NPACK,GGP,WRKS)

*     WRITES ARRAY GG(NLG,NLAT) ONTO SEQUENTIAL FILE NF
*      PRECEDED BY AN IDENTIFYING LABEL OF 7 WORDS.
*     ID = IDENTIFICATION NUMBER FOR THE RECORD.
*     NAME = ALPHANUMERIC LABEL FOR THE FIELD.
*      N   = NUMERIC LABEL FOR THE FIELD.
*     GGP,WRKS ARE WORK FIELDS USED ONLY IF FIELD IS PACKED.
*     GGP = NLAT*((NLG-1)/NPACK+2) WORDS IN LCM.
*     WRKS = NLG WORDS IN SCM.

      DIMENSION GG(1),GGP(1),WRKS(1)
      DATA KFLD/ 4HGRID /


*     IF THERE IS NO PACKING, WRITE THE LABEL AND FIELD
*      DIRECTLY ONTO FILE NF AND RETURN.

      IF(NPACK.GT.1) GO TO 25
      LA=NLG*NLAT
      WRITE(NF) KFLD,ID,NAME,N,NLG,NLAT,NPACK,(GG(I),I=1,LA)
      WRITE(6,620) ID,NAME,N,NLG,NLAT,NPACK,NF
      RETURN

      AN ERROR HAS BEEN ENCOUNTERED IN THE LABEL FIELD

   25 CALL FPAK
      RETURN


  620 FORMAT(1H ,60X,I5,2X,A4,I3,6H  GRID,I5,I4,8H  NPACK=,I2,
     1        8H TO FILE,I3)
      END
```