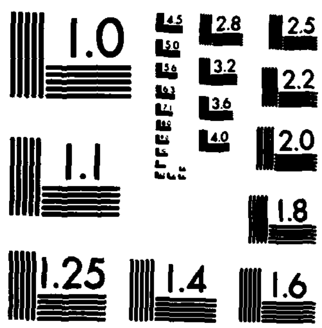
AD-A146 548 SIGMA CODE TESTING(U) SCIENCE APPLICATIONS INC MCLEAN VA J L SEFTOR ET AL. 28 MAR 84 SAI-84/1073 N00014-83-C-0289 1/2

UNCLASSIFIED F/G 8/10 NL



OPY RESOLUTION TEST CHART

Sigma Code Testing

SAI Final Report No. SAI-84/1073

March 28, 1984

J. Laurence Seftor
and
Glyn O. Roberts

SCIENCE APPLICATIONS, INC.

Sigma Code Testing

SAI Final Report No. SAI-84/1073

March 28, 1984

J. Laurence Seftor
and
Glyn O. Roberts

ATLANTA • ANN ARBOR • BOSTON • CHICAGO • CLEVELAND • DENVER • HUNTSVILLE • LA JOLLA
LITTLE ROCK • LOS ANGELES • SAN FRANCISCO • SANTA BARBARA • TUCSON • WASHINGTON

Sigma Code Testing

SAI Final Report No. SAI-84/1073

March 28, 1984

Submitted to:

Dr. S.A. Piacsek
Naval Ocean Research
and
Development Activity
Code 322, NSTL Station, MS 39529

Prepared by:

J. Laurence Seftor

and

Glyn O. Roberts

Prepared Under:

Contract No. N00014-83-C-0289

Science Applications, Inc.

1710 Goodridge Drive
P.O. Box 1303
McLean, Virginia 22102
(703) 734-5840

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
SAI-84/1073

2. GOVT ACCESSION NO.
AD-A146548

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Sigma Code Testing

5. TYPE OF REPORT & PERIOD COVERED
Final 3/83-3/84

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
J. Laurence Seftor and Glyn O. Roberts

8. CONTRACT OR GRANT NUMBER(s)
N00014-83-C-0289

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Science Applications, Inc.
1710 Goodridge Drive, P.O. Box 1303
McLean, Virginia 22102

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Office of Naval Research
800 N. Quincy Street
Arlington, VA 22217

12. REPORT DATE
28 March 1984

13. NUMBER OF PAGES
163

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)
Naval Ocean Research and Development Activity
Code 322, NSTL Station, MS 39529
Attn: Dr. S.A. Piacsek

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Distribution Unlimited

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Numerical Ocean Forcasting
Ocean Circulation Modeling
Open Boundary Conditions

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

The SAI/NORDA Sigma Coordinate Ocean Forcasting computer code is a complicated simulation tool for modelling the behavior of the world's oceans. In this report, we present the result of a set of tasks which increase the code's accuracy and usefulness as a simulation tool. These tasks include the implementation of open boundary conditions, the creation of initialization data from mixed data sets, the improvement of output options, and

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

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) the installation of the code on the VAX supermini computer. We also present the results of preliminary set up tasks for a simulation of the semiclosed basin of the eastern Mediterranean.

Accession For	
NTIS GRA&I	☑
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

1.0 Introduction

The SAI/NORDA sigma coordinate ocean forecasting computer code (sigma code) is a complex tool for predicting ocean temperature, salinity, density, sound speed, and the three components of current velocity as functions of location and depth. The non-uniform sigma vertical representation is used to simplify the treatment of the severe topography encountered in the ocean and to concentrate computation levels near the surface. A stable, temporally implicit finite-difference representation of the model differential equations allows the use of large time steps.

The basic code has been developed under past contracts (see references). The testing of the model, however, has been limited by the simple no-flux lateral boundary conditions installed in the model during its development.

This report will discuss the development and partial implementation of open boundary conditions into the model.

Also presented are the results of a series of smaller tasks which have been undertaken to increase the code's usefulness. They include new mechanisms for data initialization, new output options, and the conversion of the code to the VAX super minicomputer.

Finally, initial results will be presented for a study of ocean behavior in the semi-closed basin of the Eastern Mediterranean.

2.0 Open Boundary Conditions

The original form of the code used a very simple lateral boundary condition; no flux was allowed. This rigid wall approximation, while useful for testing, is not valid for any ocean region of interest. All ocean areas have significant inflow and outflow.

Two approaches were taken for providing open boundaries. The first is a simple condition, which can be of use in regions where the flow in and out is of limited spatial extent. Two examples are the Gulf of Mexico and the Mediterranean. This scheme will be discussed in section 2.1.

A more realistic approach requires full information about temperature, salinity, and velocities at each point of the simulation region boundary. To provide such information, a nested run approach was undertaken. In this scheme, a coarse run is performed over a large region. This run stores initial and boundary data, which is used by a second run whose simulation region is nested within the region of the large run. Because of the very complicated spatial representation used within the Sigma Code, this second approach requires very complex coding. The partial implementation of this second approach will be described in section 2.

2.1 Flux Specified Boundaries

In this approach, a single flux is specified for each of the four edges, of the system. A function of lateral area is used to determine how much of the total flux will be deposited at each spatial location. For example, if the west edge of a simulation region is all land, except for a narrow strait, virtually all the flux will occur at the strait. This follows from the fact that the strait contains all the lateral area for that edge of the system (see figure 1). It should be remembered that in the sigma code there are a full set of computational levels, even over land areas. This scheme ensures that over such land areas, no flux will be allocated.

The flux at each lateral point is given by

$$u_n = A*f(t)*U(\bar{z}^k, z_b, DINFL, DINFLS)$$

where A is determined below, f(t)=1 -exp(-ISTP/30) with ISTP the time step number, and U is the functional dependence of the vertical profile. We define

$$U = \frac{V*W}{|V|+W} \quad , \qquad \text{where}$$

$$W = \frac{z_b - Z}{DBOTBL} \qquad \text{and}$$

$$V = \frac{DINFLS-Z}{(DINFLS+Z)\ (1+ Z/DINFL)} \quad .$$

DINFL, DBOTBL, and DINFLS are given as input parameters, z is the depth, and z_b is the bottom depth. DBOTBL, through W, controls a function which linearly drops from z_b/DBOTBL at the surface, to 0 at the bottom.

The function V determines the actual shape of the vertical profile. DINFL sets the approximate depth of the inflow. Since it is sometimes desirable to have both inflow and outflow occuring at the same location, V changes sign at depth DINFLS. By setting DINFLS very large (1.E10), no sign change occurs, and V is always positive. When DINFLS is 0., the sign change occurs at the surface, and V is always negative.

The constant A is adjusted according to

$$\mathrm{FLUXIN} = -A\Sigma U\alpha_n \quad ,$$

so that the total flux at each edge totals up to the input parameter FLUXIN. U is as defined above, α_n is a geometry factor which depends on the edge being specified, and the sum runs over all points on the edge.

There are two modes of operation for this boundary condition. The first is unidirectional flow. The user sets DINFLS large (>1.E10), and sets DBOTBL, DINFL, and FLUXIN for the desired flow.

The second mode is bidirectional flow. In this case, the user can set the ratio of inflow to outflow, DINFLR. The code, by an iterative scheme, adjusts one of the other variables to obtain the correct ratio. If DINFLR>Ø, DINFLS is adjusted. DINFLR<Ø is a way to adjust DINFL instead. For best results, a good guess for DINFL and DINFLS should be made. By running the code with NSTEP=Ø, the final value of DINFLR can be compared to that requested.

The input variables are:

FLUXIN(i)	total flux (outward) for this edge,
DBOTBL(i)	scaling factor for linear profile,
DINFL(i)	depth of inflow,
DINFLS(i)	depth of sign change, and
DINFLR(i)	ratio of inflow to outflow.

Here i determines which edge of the system according to

i=1	west edge
2	east edge
3	south edge
4	north edge

The adjustment of DINFL or DINFLS, and the determination of A is performed in new subroutine FLUXST. The flux calculated by this scheme is added to the velocity arrays in UVBND.

2.2 Open Boundary Conditions Based on a Nested Run

A nested run consists of a fine resolution run whose simulation region is located within the simulation region of a coarse, larger area run. (See Figure 2). There are four aspects of such a simulation. One, a coarse simulation is performed. This run writes out both data for initialization of the fine run, plus boundary data on the embedded boundaries. Second the fine run performs a complex interpolation from this data to obtain an initial condition. Third, the fine run performs a complex interpolation, at each time step, to find appropriate values along its boundary. Fourth, these boundary values are used in a suitable boundary condition. Each of these steps will be described below.

2.2.1 Coarse Run

The initialization data written out by the coarse run consists of temperature, salinity, pressure and velocities over a region which encompasses the fine region. All the coarse data is not needed, and it is not written out. There may, however, be more than one fine region nested within a particular coarse region. The data for each fine region is written to a different file.

The number of such subregions is specified by input parameter NREGON. For each of these NREGON subregions, the location of the boundaries is specified by:

SAVLLO(i) west edge of region i in degrees
SAVLHI(i) east edge of region i in degrees
SAVPLO(i) south edge of region i in degrees
SAVPHI(i) north edge of region i in degrees.

The coarse run uses these input variables in subroutine REGNST to establish the areas on its grid that will have to be written out for each of the subregions. It also determines the locations of its grid lines which lie on either side of the embedded boundaries, for both T-S and U-V data, again for each subregion.

At each time step, subroutine SAVREG is called to write out the appropriate data. The writing of data is controlled by:

SAVSTR(i) time step of initial write for region i.
SAVINC(i) modulus on time step to perform write for region i.

On only the initial write, data for initialization is written out. On this, and all subsequent writes, boundary data is written. Each subregion has its own file. Therefore, the location and frequency of the data may be completely different for each subregion. The file unit number is 70 + i, where i is the region number.

Subroutine RWBND is used to read or write the boundary data.

2.2.2 Initialization of Final Run

The data written out by the coarse run must be interpolated onto the fine mesh of the nested run. This is complicated by the complex spatial representation of the sigma code. As can be seen in Figure 2, the lateral mesh spacing may be nonuniform and arbitrary. As shown in Figure 3, the vertical spacing is nonuniform and depends both on the topography, and the number of vertical levels. In a typical simulation, no two grid nodes are located at the same depth.

The initialization in the sigma code is performed by one of the "CASE" routines. For the nested run the input parameter CASE is set to 8. This causes subroutine CASE8 to be called to perform an initialization from coarse run data. Storage must be carefully handled so that no excess array space is allocated. To do this, CASE8 was designed as a dummy routine to handle storage for subroutine CASE8A where the actual interpolation is performed.

In CASE8A an interpolation is performed using the 8 surrounding data points from the coarse data. Weight factors are determined, and then the new value is calculated from,

$$X_f = \frac{\sum_{i=1}^{8} \omega_{(i)} X_c(i)}{\sum_{i=1}^{8} \omega_{(i)}} .$$

Because of the representation, the eight weights have different values for each of the fine mesh grid points.

Additional problems arise because of the use of real topography. In Figure 3, point A is a possible location of a fine mesh point. It can use the coarse value to the left, but the coarse value to the right does not lie deep enough for point A, and it cannot be used. The interpolation scheme is therefore modified to exclude the point to the right, and the weights for the points actually used are corrected.

Point B is the location of a second type of problem. At this location no coarse data is available at the depth of point B. For this problem a different procedure is used. Subroutine CASE8B is called to perform a spline extrapolation of the surrounding data to obtain values at the correct depth. Then a four point lateral interpolation of these extrapolated values is used to obtain a reasonable result. This involved procedure is only used for temperature and salinity values. For a velocity point at location B, when no coarse data are available, the velocity is set to zero. This ensures that no artifical flows will result.

It should be noted that different weights are required for temperature-salinity (T-S) and velocity data, since T-S data locations are spatial shifted from velocity locations.

2.2.3 Determination of Boundary Value Data for Fine Run

At each step during the fine run, the values of temperature, salinity, the velocity, and the pressure must be determined at the location of the physical boundaries of the smaller system. These values are calculated, and put into a set of two-dimensional arrays. These arrays are used by the boundary conditions in determining the values of the relevant quantities at the guard cells. It should be noted that these boundary values are not used directly, but are used in the application of the boundary conditions.

Data is written out on both sides of the boundary by the coarse run. Since T-S and U-V data are spatial displaced, these surrounding boundary planes are located differently for the two types of data.

Subroutine EXBND is used to determine the values of data on the boundaries. As in the use of CASE8, EXBND acts as a storage manager for EXBNDA, where the actual interpolation is performed. The interpolation scheme is the same as that used in CASE8A, with subroutine CASE8B used here alsc for points out of range.

2.2.4 Implementation of Nested Boundary Conditions

Our planned implementation of nested boundary conditions is based on a major extension of the analysis in § 4.4 of the prior study (Roberts, 1982). In that section we studied the reflection and transmission properties of different sets of open boundary conditions applied to the one-dimensional wave equation, and demonstrated that the spurious waves could be reduced to a minimum by imposing

$$cu_n + p$$

on the boundary, where p is the scaled pressure, u_n is the outward flow, and c is the local phase speed of the wave. The success of this method follows from the fact that the outward and inward propagating wave modes have

$$cu_n = \pm p,$$

respectively.

In the prior report we reached the conclusion that a generalization of this boundary condition to the three-dimensional sigma code, with rotation, diffusion, stratification, and bottom topography effects, in addition to the surface gravity waves, was impractical. In § 5 we outlined a simpler set of boundary conditions based on the normal flow and on conservation requirements.

However, we now believe that for most situations of interest, spurious surface and internal gravity waves generated at the open boundaries by poorly chosen boundary conditions will pose the greatest problem in nested computations. We therefore plan to implement a generalization of the above boundary condition to the full three-dimensional sigma-coordinate system of equations.

Our boundary conditions are

$$\bar{u}_n^n = u_{nd} + F(\bar{\pi}^{\parallel} - \pi_d) + \beta g(z,z_b)$$

$$\delta_n u_{\parallel} + \alpha(\bar{u}_{\parallel}^n - u_{\parallel d}) = 0$$

$$\delta_n \bar{T}^n + \alpha(\bar{T}^{nn} - T_d) = 0$$

$$\delta_n \bar{S}^n + (\bar{S}^{nn} - S_d) = 0$$

Here

u_n and $u_{\parallel}$ denote the velocity components outward and parallel to the boundary,

$\bar{f}^n$ and $\bar{f}^{\parallel}$ denote averages of adjacent mesh values in the same directions, for any variable f,

f_d denotes data values obtained by interpolation from the previous coarse run, at the appropriate positions and times,

α is a positive function of $u_n \delta x/K_H$, and is zero for large values (passive outflow condition) and very large for negative values (value imposed on inflow),

$\delta_n \bar{T}^n$ is the difference over two mesh intervals (outside minus inside) divided by two,

$\bar{T}^{nn}$ is the mean of the outside and inside values, ignoring the boundary value (a non-standard notation),

g is $(z-z_b)/(1+z/z_i)$

β is zero or is chosen so that the total normal is an imposed value obtained from data on the movement of the free surface.

F is an operator designed to separate the internal wave modes, and divide each by an appropriate wave speed c; the simplest operator option is to divide by a single imposed c value.

These boundary conditions replace our previous conditions

$$\bar{u}_n^n = U$$

$$\delta_n u_\| = 0$$

$$\delta_n \bar{T}^n = 0$$

$$\delta_n \bar{S}^n = 0 \quad .$$

For our Mediterranean simulation we imposed a nonzero U distribution; U has otherwise been zero.

2.2.5 Status of Implementations

The first three steps have been implemented, and preliminary tests performed. Coarse runs have been used to write out initial and boundary data. Fine runs have read in initial data, and have interpolated this data to get an initial condition for the system. In addition, the boundary data has been interpolated to obtain values of the relevant quantities on the boundaries.

There has been insufficient time, however, to implement the algorithms of Section 2.2.4. These algorithms will go into subroutines TSBND and UVBND, and will use the boundary value data which now exists in the code.

Subroutines which have been changed and new subroutines are listed in Appendix A.

3.8 External Data for Initialization

The quality of a simulation is very dependent on the quality of the data used for the initialization of the run. It is, therefore, very important to use the best quality data available for initialization. OTS and EOTS data is of better quality than climatology, but such data does not go deep enough to initialize the entire simulation region. The best solution, therefore, is to smoothly merge two data sets to provide OTS or EOTS where available, and climatology where not. The transition between the two data sets must be smooth, or numerical noise will result.

If T_E is OTS or EOTS data, and T_C is climatology data, the procedure is as follows. For depths above the lowest T_E point (z=d)

$$T = T_E(z), \qquad (z \le d)$$

where T_E at arbitray values of z are determined from a spline fit. For z>d,

$$T = T_C(z) + \Delta T f(z) + \Delta T' g(z) .$$

Here $T_C(z)$ is again obtained from a spline fit, and

$$\Delta T = T_E - T_C \qquad \text{at} \qquad z = d$$

$$\Delta T' = T'_E - T'_C \qquad \text{at} \qquad z = d \quad .$$

The functional forms of f and g are given as

$$f(z) = \frac{z}{d} \exp\left(1-\frac{z}{d}\right),$$

$$g(z) = (z-d) \exp\left(1-\frac{z}{d}\right).$$

At z = d these functions result in

$$f(d)=1 \quad , \qquad f'(d)=0,$$

$$g(d)=0 \quad , \qquad g'(d)=1.$$

The implementation of this capability involved two sets of changes. First the data tape handling program TAPER2 had to be slightly modified to create a data file with both OTS (or EOTS) and climatology. Secondly, two new CASE routines, CASE6 and CASE7 were written to read in the combined data, and to perform the merging. CASE6 merges OTS and climatology, while CASE7 handles EOTS and climatology.

In practice, two TAPER runs are used to produce the combined input data set. The listing in Appendix B is an example which shows the procedure. A TAPER2 run is used to read in the appropriate OTS (or EOTS) data. Then a TAPER3 run is used for the climatology data. The output of TAPER3 is appended onto the output file from TAPER2 to form the required input data file.

In the sigma code run, one simply assigns the data file produced by the TAPER runs. Then the CASE input parameter is set to 6 for OTS, or 7 for EOTS, and the proper initialization is performed.

4.6 Enhanced Output Options

The sigma code has an extensive set of output options. Graphics output can be produced of all relevant data fields, on either plotting devices, or as printer plots. Several desired enhancements in the existing sigma code output were identified, however, and they are the subject of the current section.

The geostrophic velocity is the velocity that results from the balance between the horizontal pressure gradient and the horizontal component of the coriolis acceleration. It provides useful information about the state of the physical system being simulated. Therefore, plots of the geostrophic velocity were added as an output option.

The geostrophic velocity is calculated during the vertical sweep through the mesh by UVDOTB. Rather than allocate an additional three dimension array to hold this information, each two dimensional slice is written to a disk file by UVDOTB as calculated. At the end of the sweep, new subroutine UVGEO reads this information into a three dimensional scratch array, from which it is written onto the plot file.

Subroutine OCPLOT, in the plotting package, was modified to create plots from this newly available data. A typical output plot is shown in Figure 4.

In addition to this new capability, three enhancements were made to the existing code. First, on restarts, because of the way time step numbering was handled, plot specification did not work properly. This was corrected by a complete revision of time step handling on restarts.

Second, the orientation of the plots of vertical profiles was awkward, and did not follow the conventional orientation for such plots. This has now been corrected.

Third, up to now, certain quantities were only available as printer plots. Now such information can be displayed on plotting devices. Sample plots of the barotropic stream function are shown in Figure 5 and 6.

The required changes to the sigma code are included in Appendix A. Those subroutines in the plotting program which required change are listed in Appendix C.

The capability to plot velocities at a constant depth is available by modification of the TSPOP post processor. In the main program, the lateral averaging of the bottom topography is removed. In addition the calls to MESHST are changed so that the U-V positions, rather than the T-S positions are produced. The resulting code is given in Appendix D.

5.0 VAX Version of Code

The usefulness of the sigma code increases when it becomes available on additional computer systems. Because of the wide spread availability of the VAX 11/780 super minicomputer, it was decided to install the sigma code on this machine.

The code has been installed on the VAX and a sample output plot from a simple test run in shown in Figure 7. The code runs much in the same way as it does on the TI-ASC computer. The VAX files used by sigma code are listed below:

Unit Number	Name	Use	Preexisting
1	SIGRUNLOG	Log File	Yes
2	SIGBTOPOG	Bottom Topography	Yes
3	SIGTSPLOT	T-S Plot Data	
4	SIGBUOYIN	Buoyancy Coefficients	Yes
5	SIGINPUTD	Input Data	Yes
6	SIGPRINTF	Print File	
10	SIGUVPLOT	U-V Plot Data	
14	SIGBUOYOU	Buoyancy Coefficients(output)	
18	SIGINITIA	Initialization Data	Yes
19	SIGFORCEG	Forcing Data	Yes
26	SIGQCHEK1	Qcheck1 Print File	
27	SIGQCHEK2	Qcheck2 Print File	
40	SIGRESTRT	Restart Data	Yes
42	SIGDUMPFL	Dump File	
80	SIGWORKSP	Work Space	

These files noted as preexisting, must be available before the run, as needed.

Because it was not known what graphics software would be available on a particular VAX, no graphics device output is currently installed. However, printer graphics are installed, and working as shown in Figure 7.

An estimate of relative timing is difficult, as VAX timing depends on many factors, including system loading. A crude estimate shows that the VAX version of the code runs about a factor of 120 slower than the ASC version. This is about the ratio that was expected.

6.0 Mediterranean Tests

The sigma code was setup to perform studies of the eastern basin of the Mediterranean Sea. The process involved routine setup tasks as well as two code modifications. The setup procedure, as well as the code modifications, will be discussed below.

6.1 Setup

The region was established as 10°-37° E, and 32°-38° N. This includes the eastern boundary, but cuts off several features at the north and south. Although the bottom topography was available at 10' intervals, a 20' spacing was chosen to reduce memory requirements for this problem. The numbers of lateral grid points are:

```
(37-10)*3+2=83       in the east-west direction, and
(38-32)*3+2=20       in the north-south direction.
```

Five vertical levels were used. The three code parameters IF, JF, and KF were reset, and a recompilation of relevant routines was performed.

The topography was created from the 10' data file by use of the program tested below (Note: All programs and listings referred to in this section can be found in Appendix E). A rigid wall was imposed on the topography at all lateral boundaries, exept for the Strait of Sicily, where the inflow-outflow occurs.

Since the forcing and initialization data lies on a polar-stereographic (PS) grid, the PS limits of the simulation region were determined. PS data was found to be required in the grid whose indices ran from I=45-50, and J=31-40.

An initialization file was created from climatology using the TAPER3 program. This was adequate for testing purposes. For a better initialization, the procedure of Section 3 of this report, may be used.

The forcing data was extracted from the data tape "SAIATM2", by use of the TAPER program. As shown in the listing, the data starts on January 7, 1977. A six hour interval was chosen.

6.2 Code Modifications

The original formulation for flux specified boundaries allowed only inflow or outflow at each edge. The full formulation, as described in Section 2.1, was implemented to allow bidirectional flow, as required in the Mediterranean Tests.

The second addition to the code was required because of a fundamental difficulty in the code. The code solves for the time evolution of a general variable x by the process,

$$\dot{x} = A(x)$$
$$x^{n+1} = x^n + F(\dot{x}, dt).$$

A is a representation of the model differential equations, and F is a fixing (stabilization) operator. For a simple explicit scheme $F(\dot{x},dt) = \dot{x}dt$. However, such a scheme is unstable, and very complicated procedures have been used to develop appropriate formulations for F, for the equations to be solved here (see references).

Since the Sigma Code solves a three dimensional problem in a minimum of computer memory, careful attention was paid to the code architecture. The problem is solved in slices, with intermediate quantities overwritten as each level is solved. This dictates the implementation of the fixing operator. Specifically, lateral fixing is performed during the vertical sweep, before the vertical fixing, which requires a knowledge of all the vertical levels.

The problem arises because the lateral fixing propagates the unstable solution horizontally, before the vertical fixing has modified the result. The vertical fixing can subsequently stabilize the solution at each point, due to the physics at each point. It cannot, however, stabilize the part of the instability which has been horizontally propagated.

In the past, this problem has been circumvented by turning off the lateral fixing, thus avoiding the horizontal propopagation of the instability. Because of the smaller lateral mesh spacing required in the

Mediterranean tests, however, the lateral fixing must be on.

One solution is to add a preliminary vertical diffusion model, before the horizontal fixing. This can take a number of forms. The simple solution implemented here is to limit $\dot{x}$ (i.e. $\dot{u}$), before the horizontal fixing process. A new routine DULIM, modeled on routine ULIM has been written. It sets

$$\dot{u} = \frac{V \cdot \dot{u}}{V + |\dot{u}|} ,$$

where $V = DUMAXN \cdot DT \cdot z_b$. DUMAXN is under input control, DT is the time step, and z_b is the local depth.

6.3 Results

A preliminary test run was performed. This simulation is intended to demonstrate that the code is properly setup to perform studies of the eastern Mediterranean. No attempt was made to model actual physical processes. Rather this run establishes procedures for modeling the eastern Mediterranean. Additionally, in the process of performing this test, problems with the code, when applied to this region, were identified and corrected.

It should be noted that in the figures which follow, problems, such as inadequate labeling of contour lines, are problems in the Disspla plotting package. They result, in part, from the fact that the ASC computer center is using an out of date version of Disspla.

Figure 8 is a plot of depth values in the region. The values in the interior are the values obtained from the data file. The boundaries demonstrate the rigid wall that was articifically imposed. The one open spot is at the Strait of Sciliy. Because the labeling in the Disspla plot is sparse, a printer plot of the same data is shown in Figure 9. Since each contour is labeled by a letter, actual values can be read off this plot.

A two week interval was modeled using 56 time steps of 6 hours each. A great number of output plots can be produced from the data files which were generated during this run. Indeed, there is no limit to the data representations available. Most major quantities can be displayed on vertical slices of arbitrary great circle arcs. Additionally, horizontal slices can be displayed at arbitrary depths. The plotting programs perform the required interpolations, so that the spatial representation in the code does not impose limits on how the data may be displayed. A representative sample of possible output plots follows.

The general shape of the vertically integrated flux as shown in Figure 10, with actual values appearing in the printer plot of figure 11. The remainder of the plots show quantities on the great circle with endpoints at 10° E, 35° N and 37° E, 35° N. The temperature is displayed in Figure 12. This figure has the depth adjusted so that the entire vertical extent of the simulation is shown. Figure 13 is the same display with the maximum depth adjusted to 600 m. Such a plot allows closer examination of surface dynamics. The same set of plots for salinity are shown in Figures 14 and 15. Once again the entire vertical domain, and a closer look at the surface are shown.

A similar set of plots were also generated from the velocity data. Figure 16 is a plot of u velocity. For this slice, a positive u velocity is a velocity approximately to the right. Since this is a display on a great circle arc, rather than an arc of constant latitude, the velocity is not exactly in the plane of the plot. Figure 17 is a plot of v velocity. Again, for this slice which runs approximately east-west, a positive v velocity is a velocity roughly into the page.

The last two plots, Figures 18 and 19, demonstrate the new ability of the code to plot the two components of the geostrophic velocity. The comments made in relation to the velocity also apply here.

6.4 Conclusion

The Sigma code is now working in the eastern basin of the Mediterranean. Externally supplied data has properly been integrated into the code for both bottom topography and initialization. In addition, a combined inflow-outflow condition at an open strait is working.

Actual simulation studies will involve longer runs. A detailed examination of the possible output displays, made in light of the topography of the region, will provide an insight into the processes at work. As such studies are performed, it may become evident that changes in the externally supplied data are necessary. These may include the addition of new data (e.g. wind forcing), or the use of better data than is currently used (e.g. EOTS rather than climatology).

REFERENCES

(1) Glyn O. Roberts, J. Laurence Seftor and Walter J. Gabowski, "A Sigma Coordinate Ocean Forecasting Computer Code, Part I. Model Differential Equations, Spatial Finite-Difference Representation, and Conservation Properties." September 1980. Report SAI-80-956-WA.

(2) Glyn O. Roberts and J. Laurence Seftor, "A Sigma Coordinate Ocean Forecasting Computer Code, Part II. Time Respresentation and Stability Properties." March 1980. Report SAI-80-957-WA.

(3) J. Laurence Seftor and Glyn O. Roberts, "A Sigma Coordinate Ocean Forecasting Computer Code, Part III. Code Description." November 1980. Report SAI-80-958-WA.

(4) J. Laurence Seftor, "A Sigma Coordinate Ocean Forecasting Computer Code, Part IV. Description of Graphics Capability." September 1980. Report SAI-81-262-WA.

(5) J. Laurence Seftor and Glyn O. Roberts, "A Sigma Coordinate Ocean Forecasting Computer Code, Part V. Results." December 1980. Report SAI-81-299-WA.

(6) J. Laurence Seftor, "Description of Revised Program TAPER for Data Tape Processing." September 1982. Report SAI-83-941-WA.

(7) Glyn O. Roberts, "Open Boundary Conditions in Ocean Forecasting." November 1982. Report SAI-83-996-WA.

(8) J. Laurence Seftor, "Description of the Use of the SAI/NORDA Sigma Co-ordinate Ocean Forecasting Code with Externally Supplied Data." September 1982. Report SAI-83-940-WA.

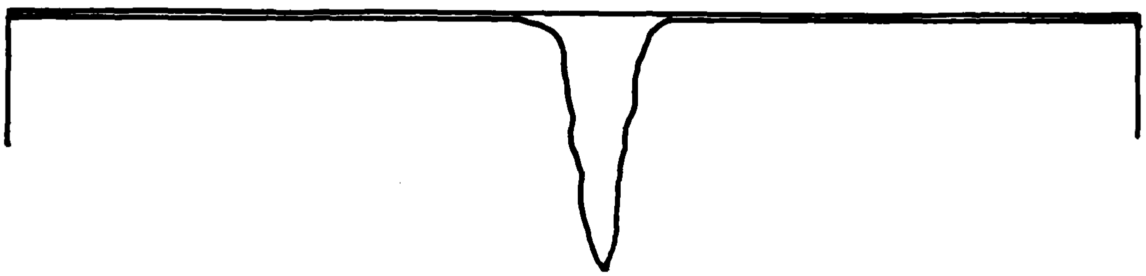


Figure 1

Depth profile at side of system, showing strait

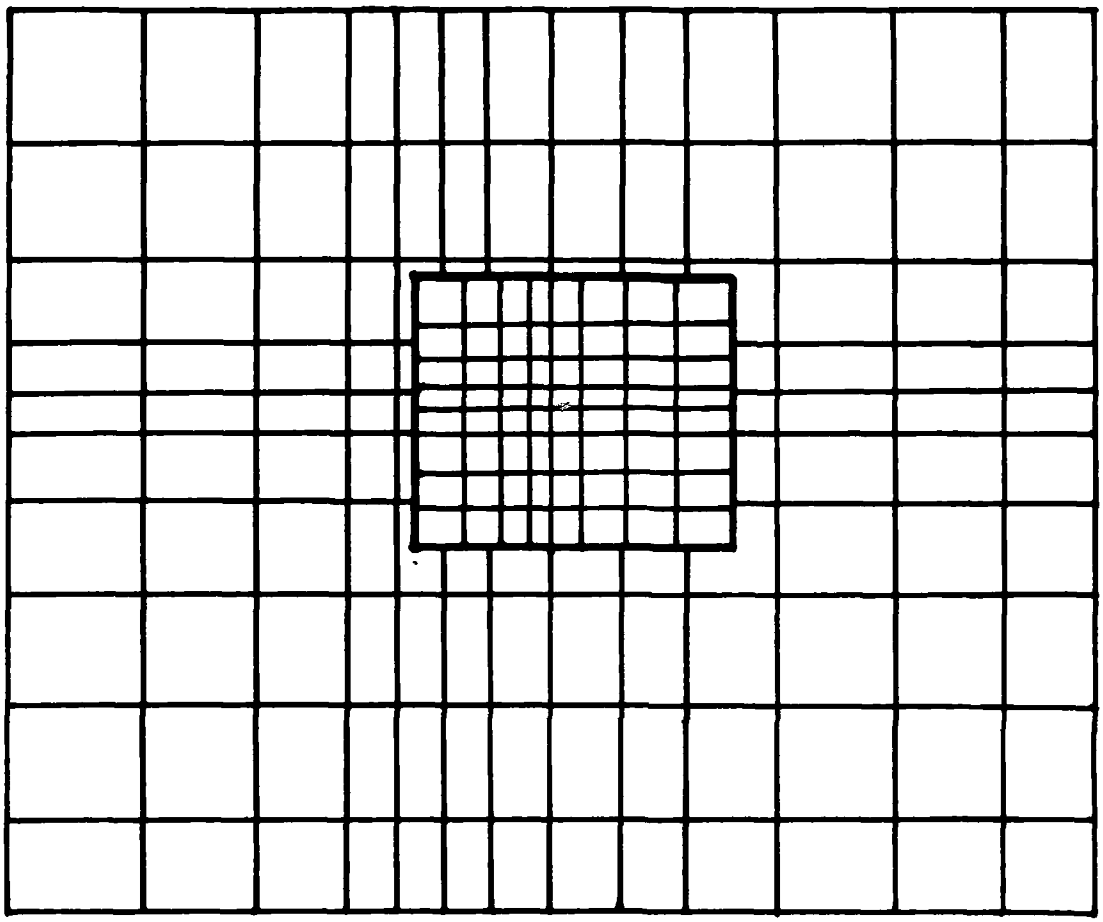


Figure 2
Lateral mesh spacing, with imbedded grid

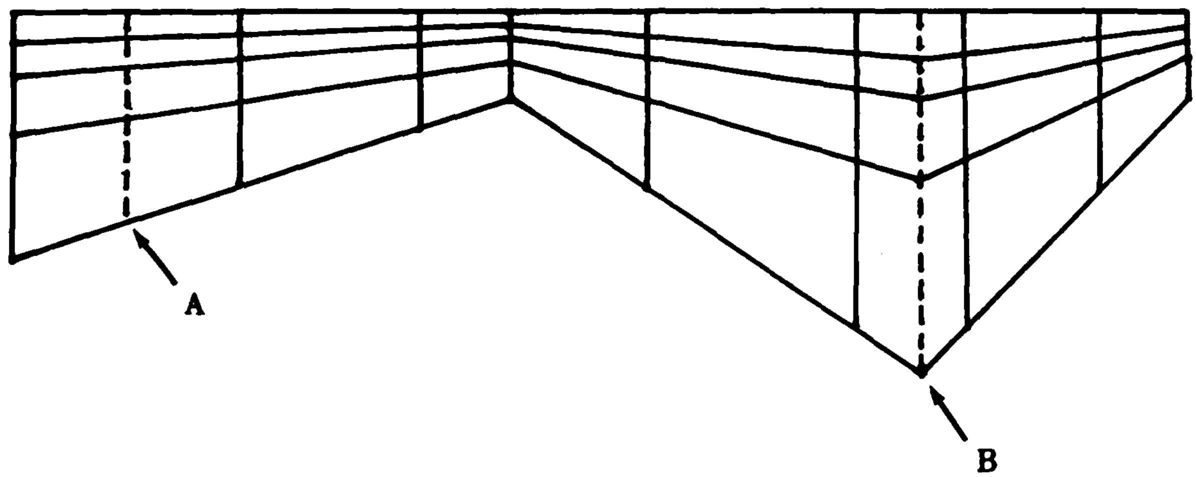


Figure 3
Typical depth profile

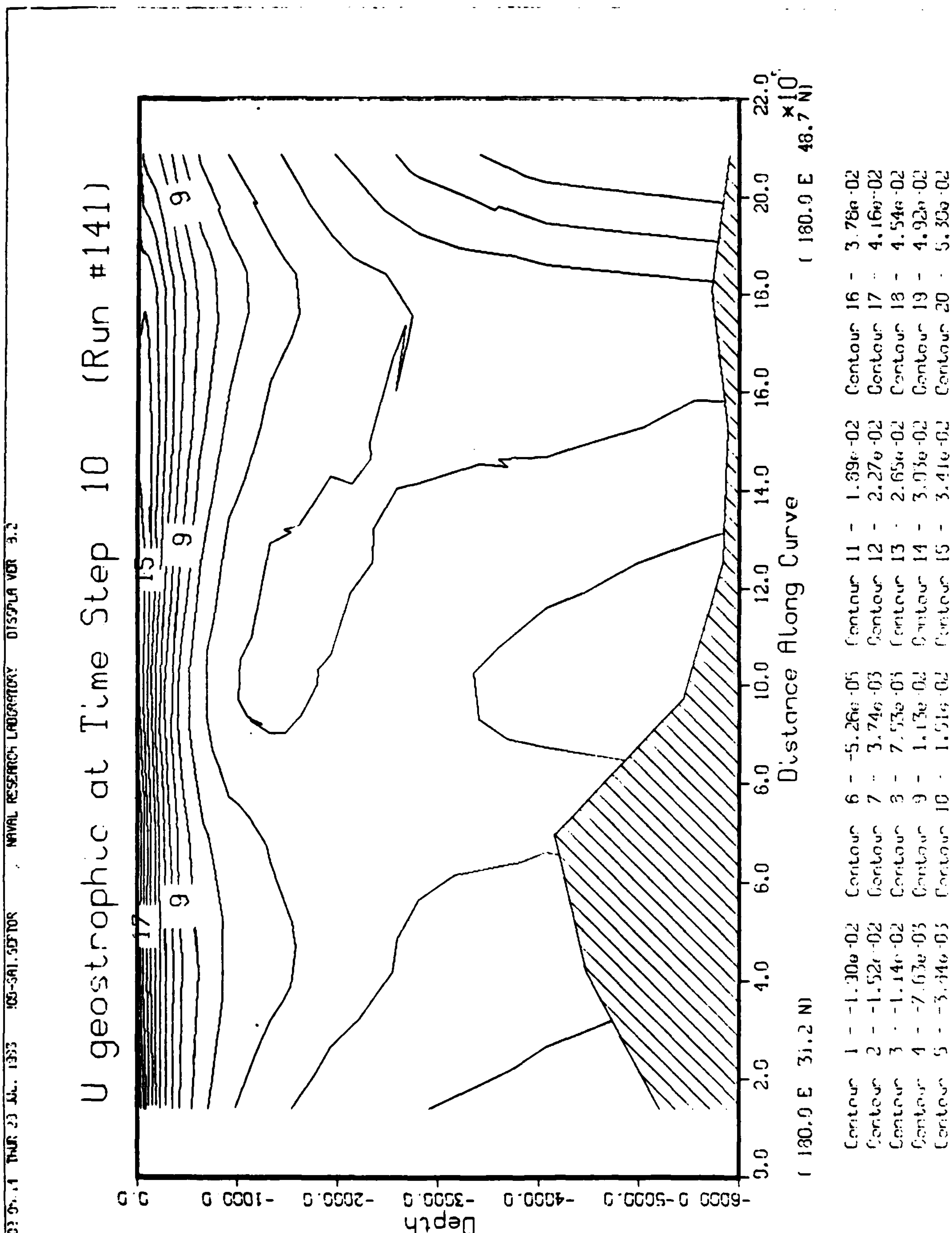


Figure 4

Sample Plot of Geostrophic Velocity

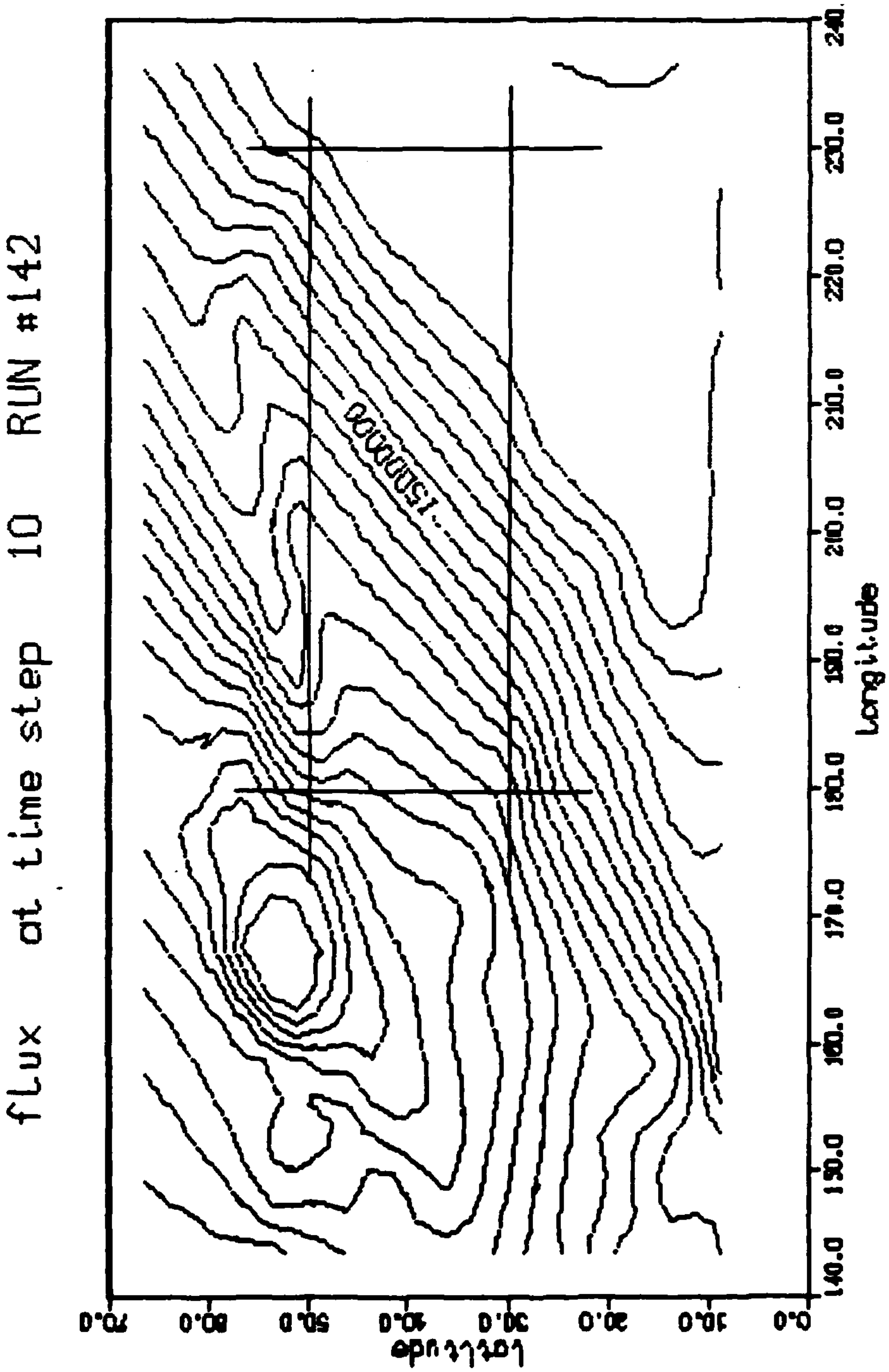


Figure 5
Sample plot of Barotropic Stream Function

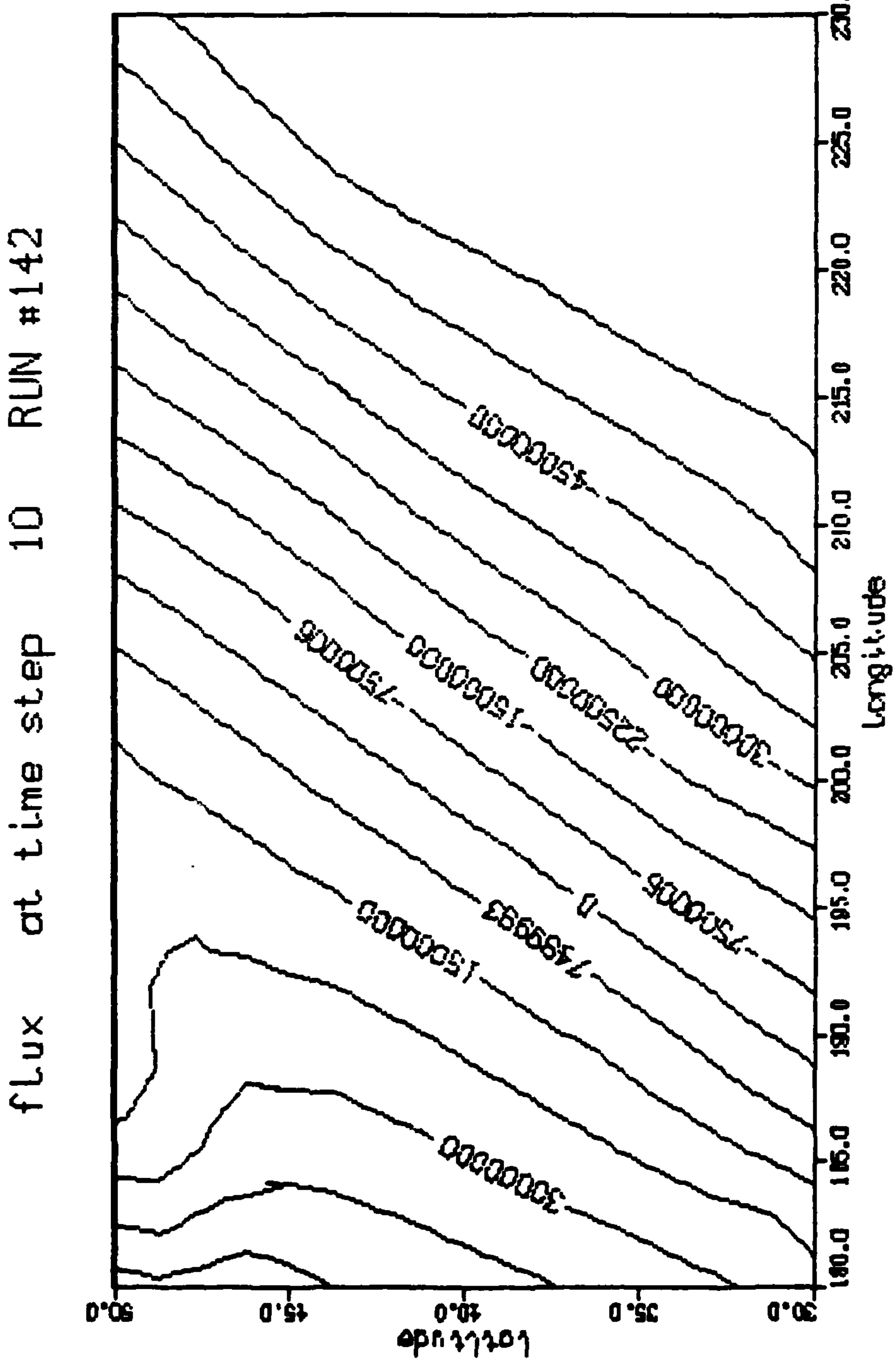


Figure 6

Sample Plot of Barotropic Stream Function (expanded view)

TEMPERATURE AT TIME STEP 10 (DEPTH= 100.0 M) RUN #136

THE CONTOURS ARE DEFINED AS FOLLOWS:

a = -4.557E-01	E = 1.127E+01	J = 2.299E+01
A = 1.889E+00	F = 1.361E+01	
B = 4.233E+00	G = 1.596E+01	
C = 6.578E+00	H = 1.830E+01	
D = 8.922E+00	I = 2.064E+01	

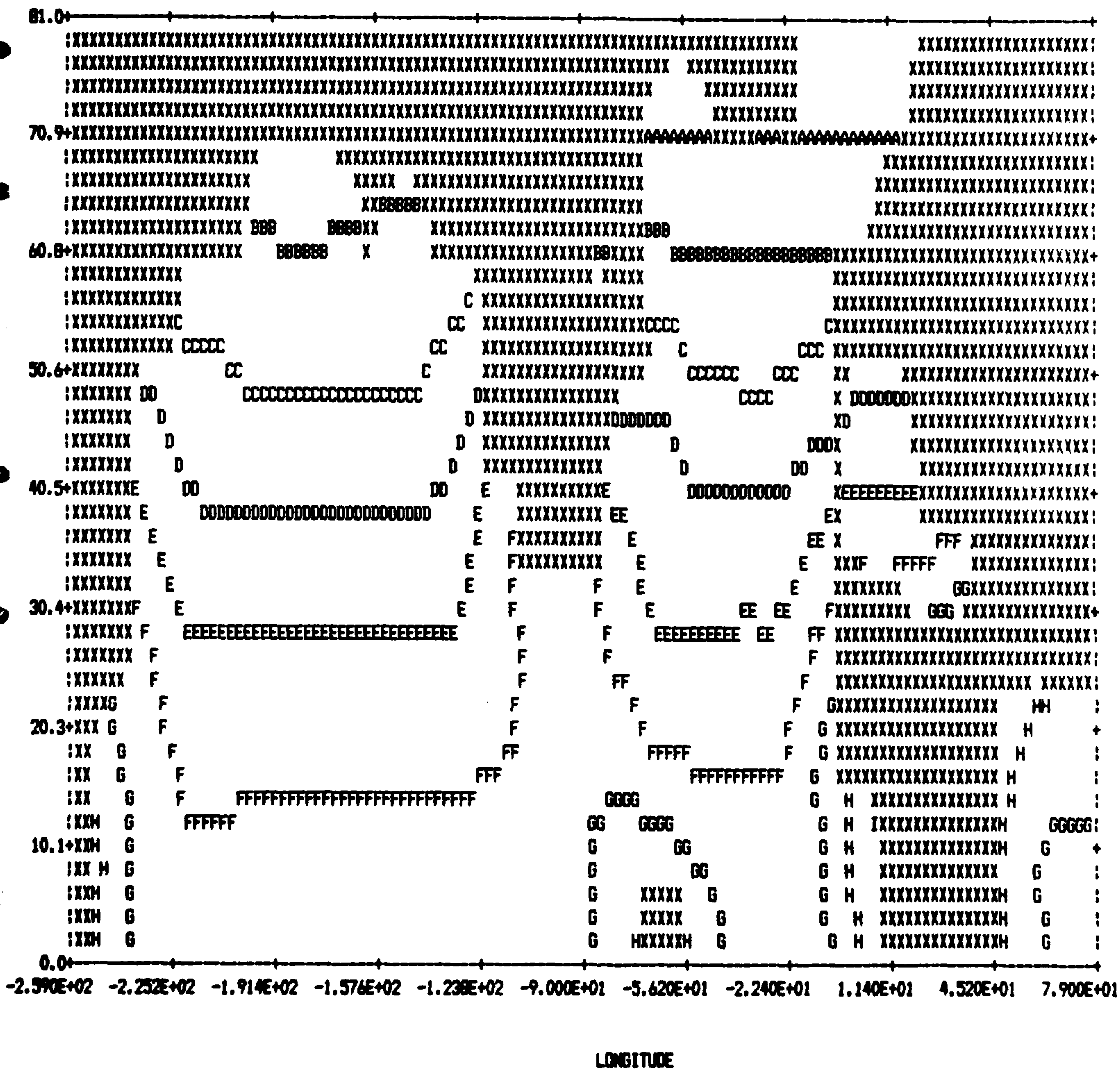


Figure 7
VAX Output Plot

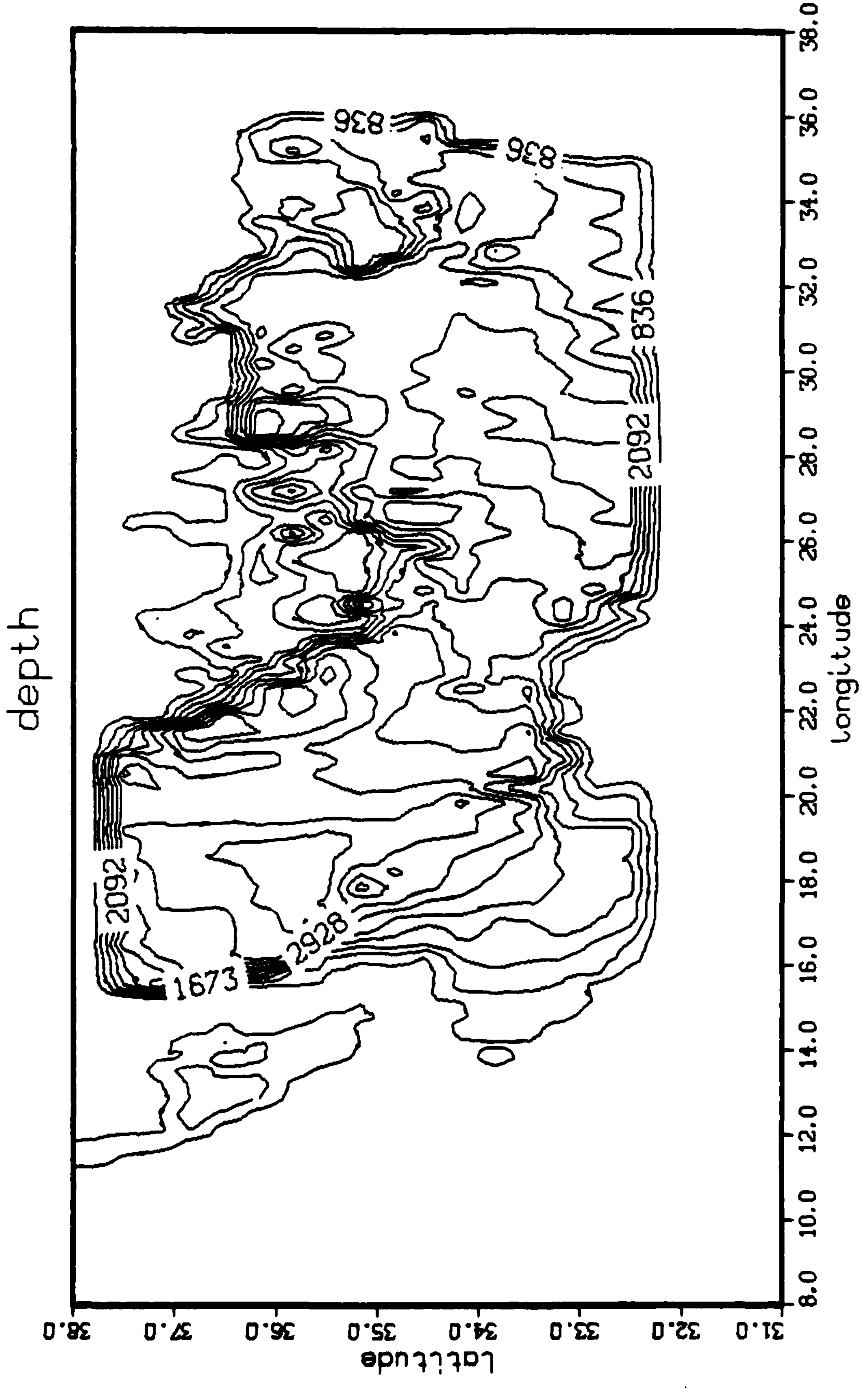

depth
longitude
latitude
8.0
10.0
12.0
14.0
16.0
18.0
20.0
22.0
24.0
26.0
28.0
30.0
32.0
34.0
36.0
38.0
31.0
32.0
33.0
34.0
35.0
36.0
37.0
38.0
836
2092
1673
2928

Figure 8

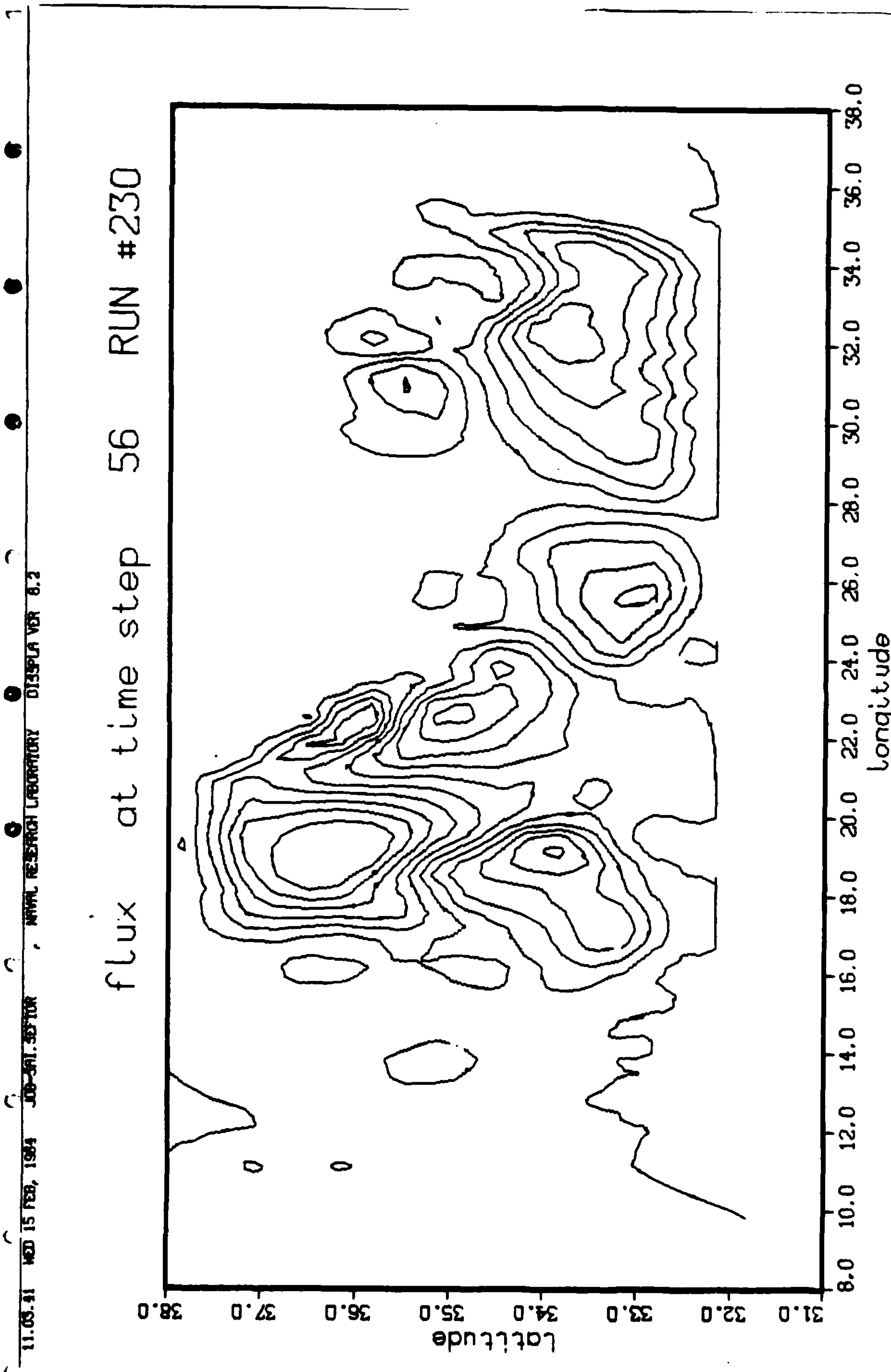
DEPTH

THE CONTOURS ARE DEFINED AS FOLLOWS:

0 =	0.000E 00	E =	1.162E 03	J =	2.324E 03	O =	3.487E 03
A =	2.324E 02	F =	1.395E 03	K =	2.557E 03	P =	3.719E 03
B =	4.649E 02	G =	1.627E 03	L =	2.789E 03	Q =	3.952E 03
C =	6.973E 02	H =	1.860E 03	M =	3.022E 03	R =	4.184E 03
D =	9.298E 02	I =	2.092E 03	N =	3.254E 03		

LONGITUDE

Figure 9



flux at time step 56 RUN #230
longitude
latitude
8.0
10.0
12.0
14.0
16.0
18.0
20.0
22.0
24.0
26.0
28.0
30.0
32.0
34.0
36.0
38.0
31.0
32.0
33.0
34.0
35.0
36.0
37.0
38.0
11.05.41 WED 15 FEB, 1984 JOB-JAT.SEFTOR , NAVAL RESEARCH LABORATORY DISSPLA VER 8.2
PLOT 2

Figure 10

FLUX AT TIME STEP 56 (RUN #230)

THE CONTOURS ARE DEFINED AS FOLLOWS:

i = -1.000E 07	d = -4.444E 06	A = 1.111E 06	F = 6.667E 06
h = -8.889E 06	c = -3.333E 06	B = 2.222E 06	G = 7.778E 06
g = -7.778E 06	b = -2.222E 06	C = 3.333E 06	H = 8.889E 06
f = -6.667E 06	a = -1.111E 06	D = 4.444E 06	I = 1.000E 07
e = -5.556E 06	0 = 0.000E 00	E = 5.556E 06	

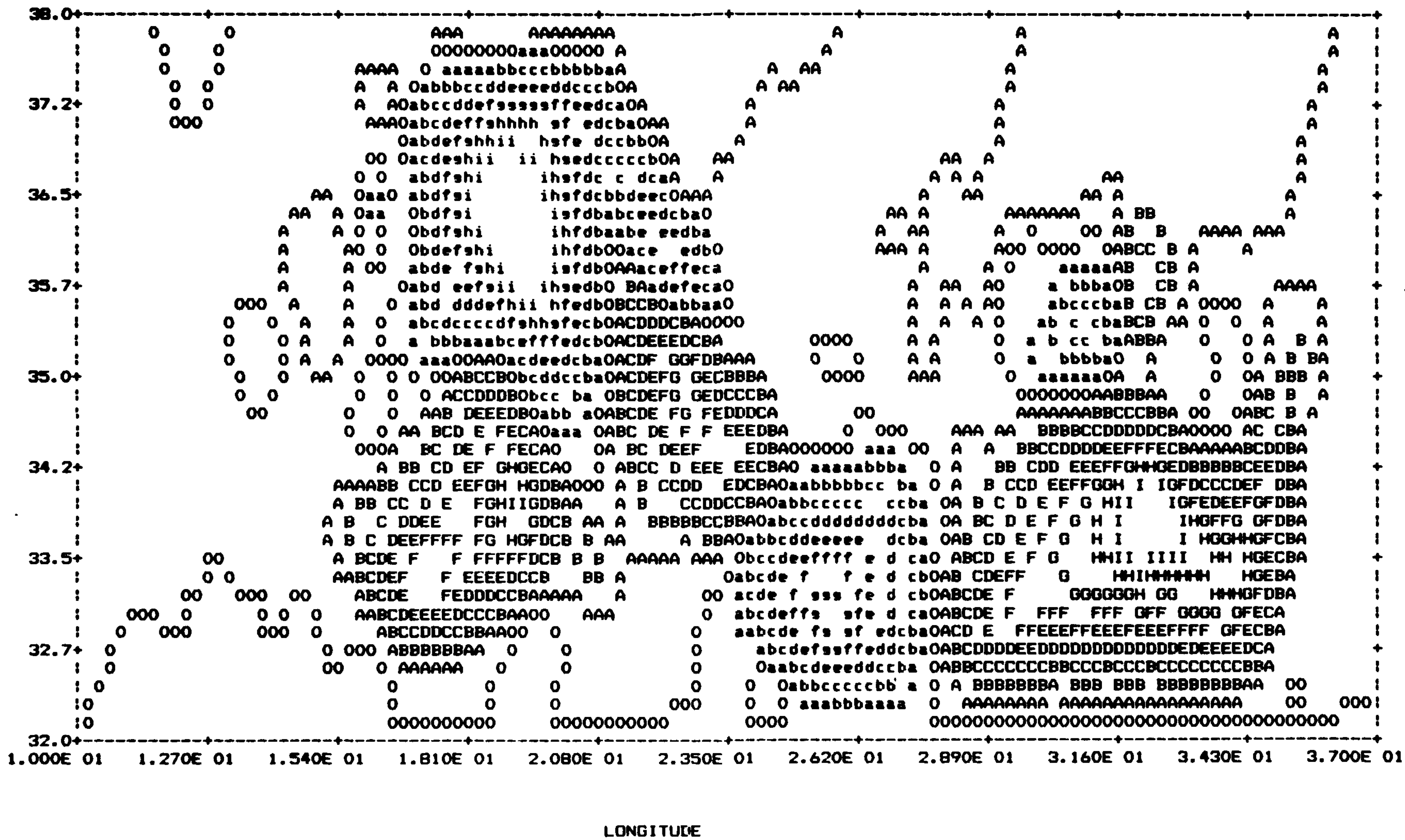


Figure 11

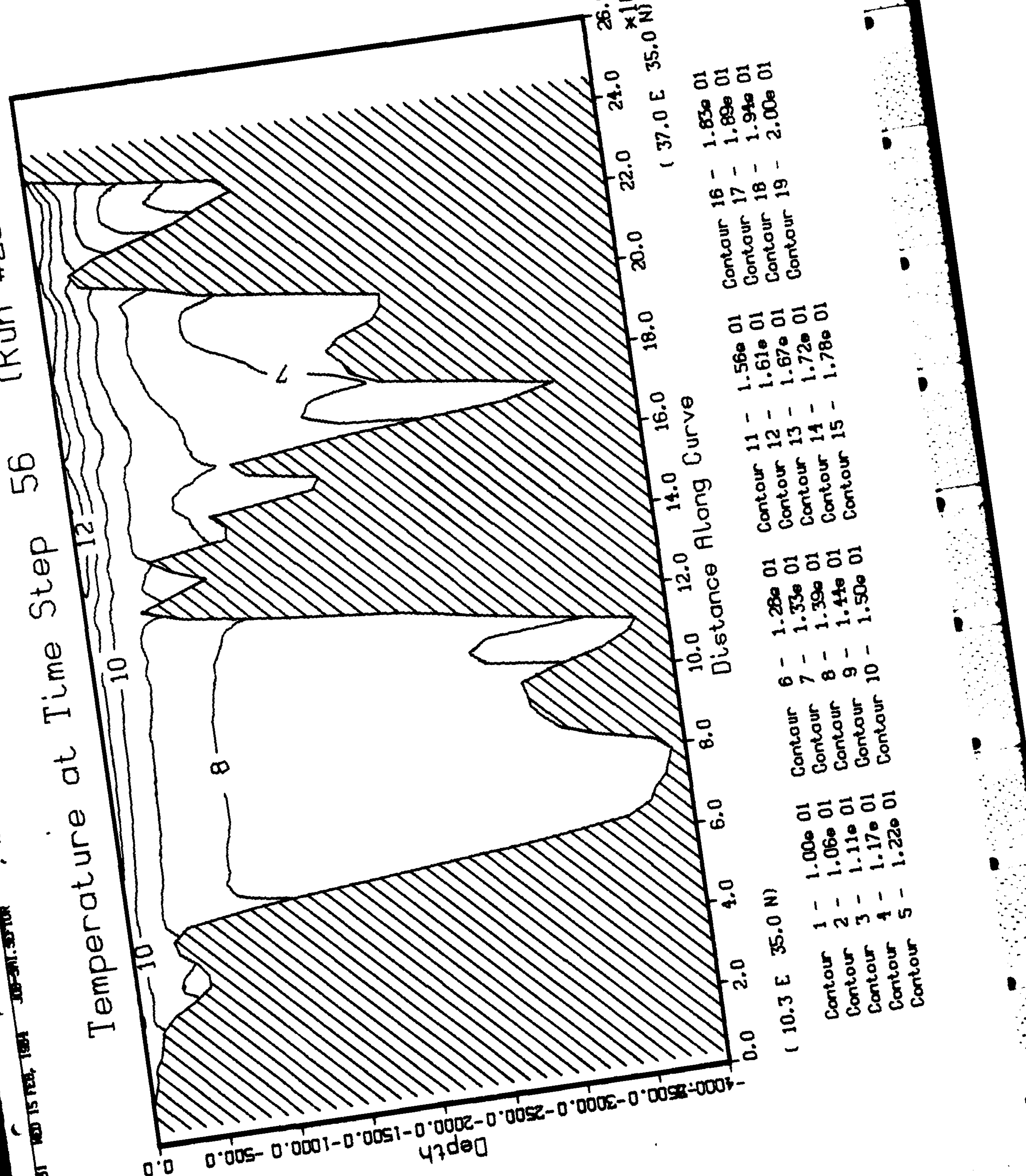

Temperature at Time Step 56 (Run #230)
PLOT 1 15.08.31 WED 15 FEB, 1984 JOB-SNT.3DTOR , NAVAL RESEARCH LABORATORY DTSPLA VER 8.2
Depth
0.0 -500.0 -1000.0 -1500.0 -2000.0 -2500.0 -3000.0 -3500.0 -4000.0
Distance Along Curve
0.0 2.0 4.0 6.0 8.0 10.0 12.0 14.0 16.0 18.0 20.0 22.0 24.0 26.0
x10^5
(10.3 E 35.0 N)
(37.0 E 35.0 N)
10
8
10
12
7
Contour 1 - 1.00e 01
Contour 2 - 1.06e 01
Contour 3 - 1.11e 01
Contour 4 - 1.17e 01
Contour 5 - 1.22e 01
Contour 6 - 1.28e 01
Contour 7 - 1.33e 01
Contour 8 - 1.39e 01
Contour 9 - 1.44e 01
Contour 10 - 1.50e 01
Contour 11 - 1.56e 01
Contour 12 - 1.61e 01
Contour 13 - 1.67e 01
Contour 14 - 1.72e 01
Contour 15 - 1.78e 01
Contour 16 - 1.83e 01
Contour 17 - 1.89e 01
Contour 18 - 1.94e 01
Contour 19 - 2.00e 01

Figure 12

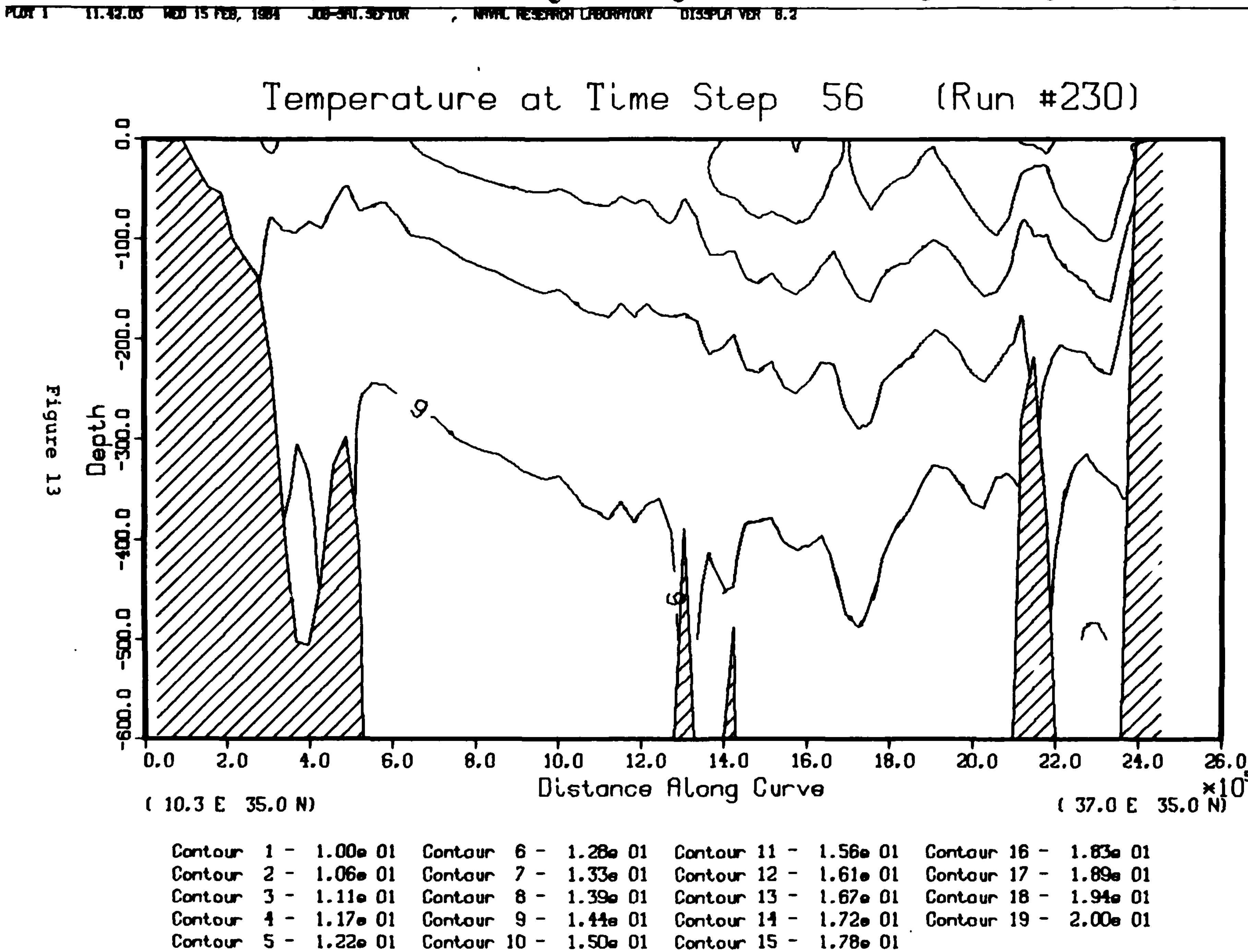

PLOT 1 11.42.03 WED 15 FEB, 1984 JOB=3RI.SEFTOR , NAVAL RESEARCH LABORATORY DISSPLA VER 8.2
Temperature at Time Step 56 (Run #230)
Depth
0.0
-100.0
-200.0
-300.0
-400.0
-500.0
-600.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
18.0
20.0
22.0
24.0
26.0
$\times 10^5$
Distance Along Curve
(10.3 E 35.0 N)
(37.0 E 35.0 N)
9
9
Contour 1 - 1.00e 01
Contour 2 - 1.06e 01
Contour 3 - 1.11e 01
Contour 4 - 1.17e 01
Contour 5 - 1.22e 01
Contour 6 - 1.28e 01
Contour 7 - 1.33e 01
Contour 8 - 1.39e 01
Contour 9 - 1.44e 01
Contour 10 - 1.50e 01
Contour 11 - 1.56e 01
Contour 12 - 1.61e 01
Contour 13 - 1.67e 01
Contour 14 - 1.72e 01
Contour 15 - 1.78e 01
Contour 16 - 1.83e 01
Contour 17 - 1.89e 01
Contour 18 - 1.94e 01
Contour 19 - 2.00e 01

Figure 13

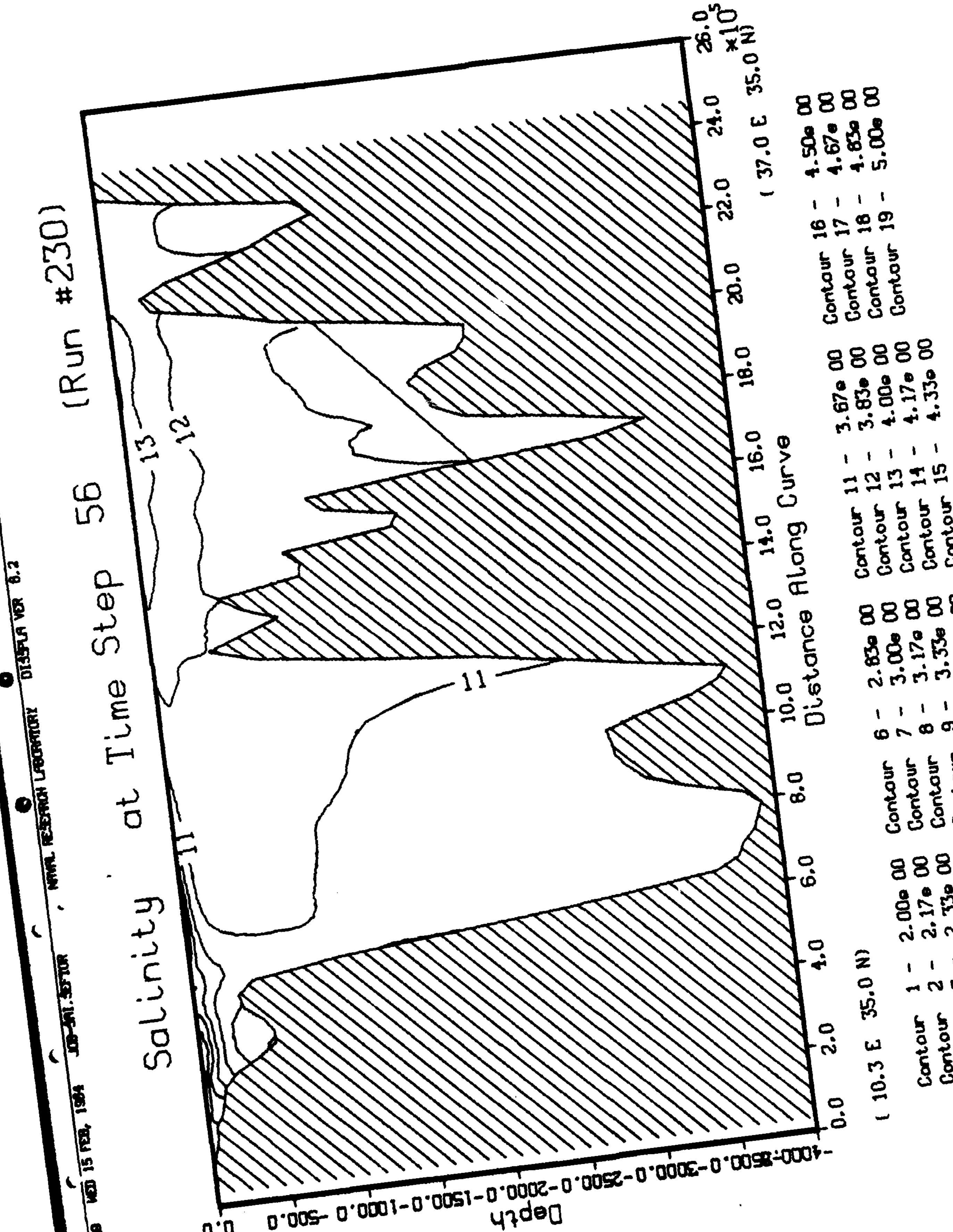

NAVAL RESEARCH LABORATORY
DISSPLA VER 8.2
Salinity at Time Step 56 (Run #230)
Distance Along Curve
Depth
0.0
-500.0
-1000.0
-1500.0
-2000.0
-2500.0
-3000.0
-3500.0
-4000.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
18.0
20.0
22.0
24.0
26.0
x10^5
(10.3 E 35.0 N)
(37.0 E 35.0 N)
11
11
12
13
Contour 1 - 2.00e 00
Contour 2 - 2.17e 00
Contour 3 - 2.33e 00
Contour 4 - 2.50e 00
Contour 5 - 2.67e 00
Contour 6 - 2.83e 00
Contour 7 - 3.00e 00
Contour 8 - 3.17e 00
Contour 9 - 3.33e 00
Contour 10 - 3.50e 00
Contour 11 - 3.67e 00
Contour 12 - 3.83e 00
Contour 13 - 4.00e 00
Contour 14 - 4.17e 00
Contour 15 - 4.33e 00
Contour 16 - 4.50e 00
Contour 17 - 4.67e 00
Contour 18 - 4.83e 00
Contour 19 - 5.00e 00

Figure 14

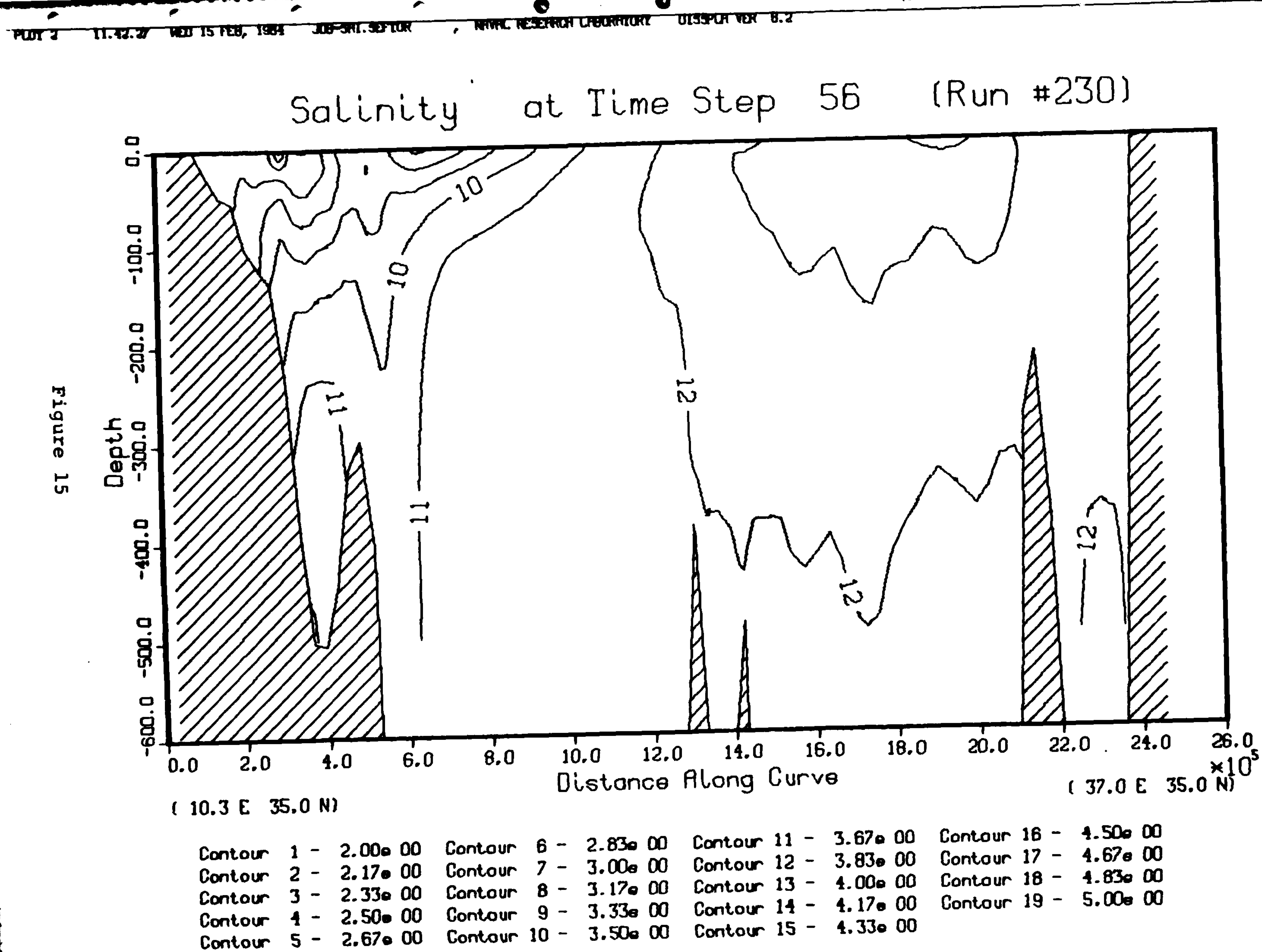
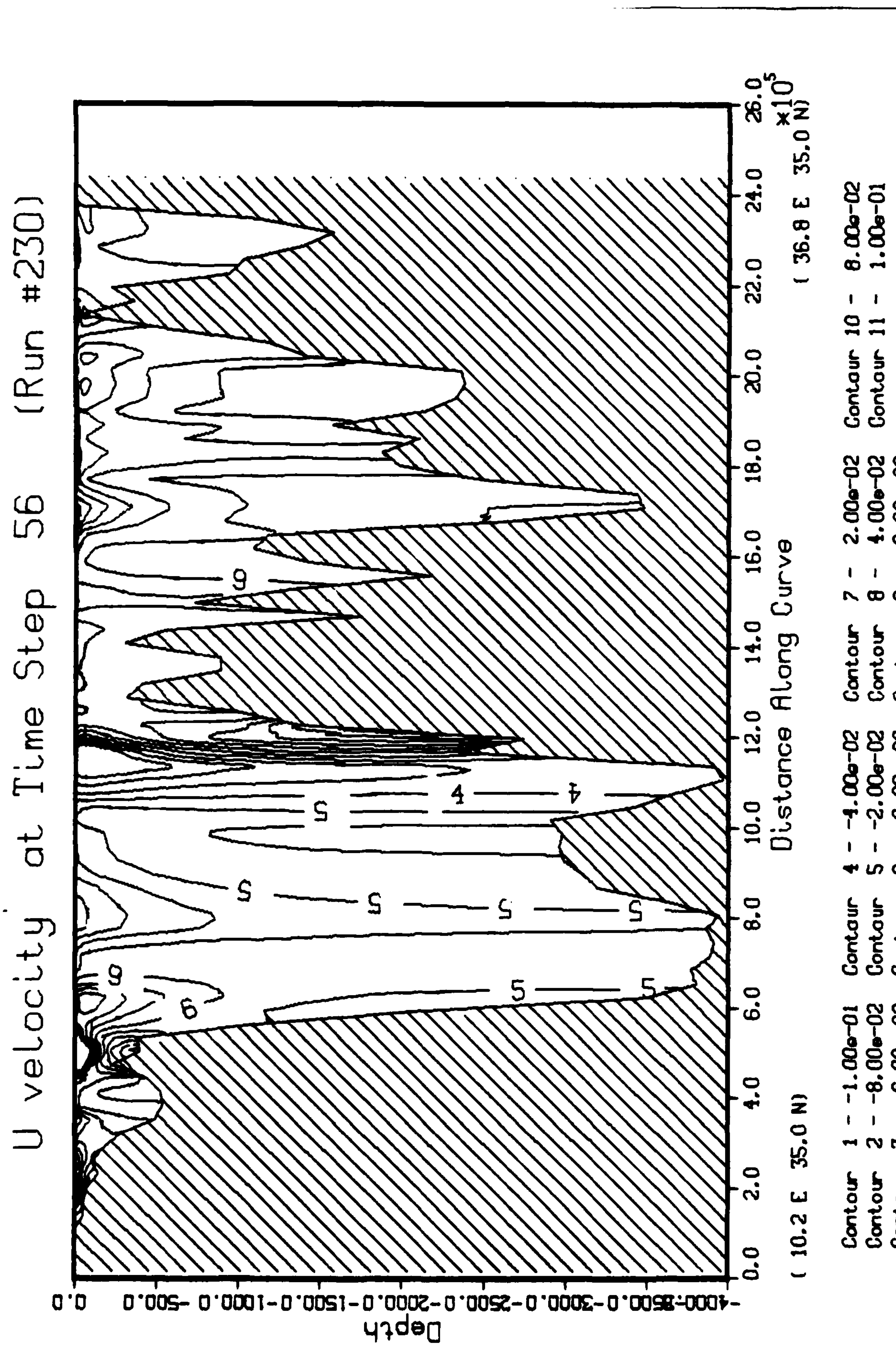

PLOT 2 11.42.27 WED 15 FEB, 1984 JOB-SAL.SEFTOR , NAVAL RESEARCH LABORATORY DISSPLA VER 8.2
Salinity at Time Step 56 (Run #230)
Depth
0.0
-100.0
-200.0
-300.0
-400.0
-500.0
-600.0
10
10
11
11
12
12
12
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
18.0
20.0
22.0
24.0
26.0
×10^5
Distance Along Curve
(10.3 E 35.0 N)
(37.0 E 35.0 N)
Contour 1 - 2.00e 00
Contour 2 - 2.17e 00
Contour 3 - 2.33e 00
Contour 4 - 2.50e 00
Contour 5 - 2.67e 00
Contour 6 - 2.83e 00
Contour 7 - 3.00e 00
Contour 8 - 3.17e 00
Contour 9 - 3.33e 00
Contour 10 - 3.50e 00
Contour 11 - 3.67e 00
Contour 12 - 3.83e 00
Contour 13 - 4.00e 00
Contour 14 - 4.17e 00
Contour 15 - 4.33e 00
Contour 16 - 4.50e 00
Contour 17 - 4.67e 00
Contour 18 - 4.83e 00
Contour 19 - 5.00e 00

Figure 15



U velocity at Time Step 56 (Run #230)
Depth
0.0
-500.0
-1000.0
-1500.0
-2000.0
-2500.0
-3000.0
-3500.0
-4000.0
Distance Along Curve
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
18.0
20.0
22.0
24.0
26.0
$\times10^5$
(10.2 E 35.0 N)
(36.8 E 35.0 N)
Contour 1 - -1.00e-01
Contour 2 - -8.00e-02
Contour 3 - -6.00e-02
Contour 4 - -4.00e-02
Contour 5 - -2.00e-02
Contour 6 - 0.00e 00
Contour 7 - 2.00e-02
Contour 8 - 4.00e-02
Contour 9 - 6.00e-02
Contour 10 - 8.00e-02
Contour 11 - 1.00e-01

Figure 16

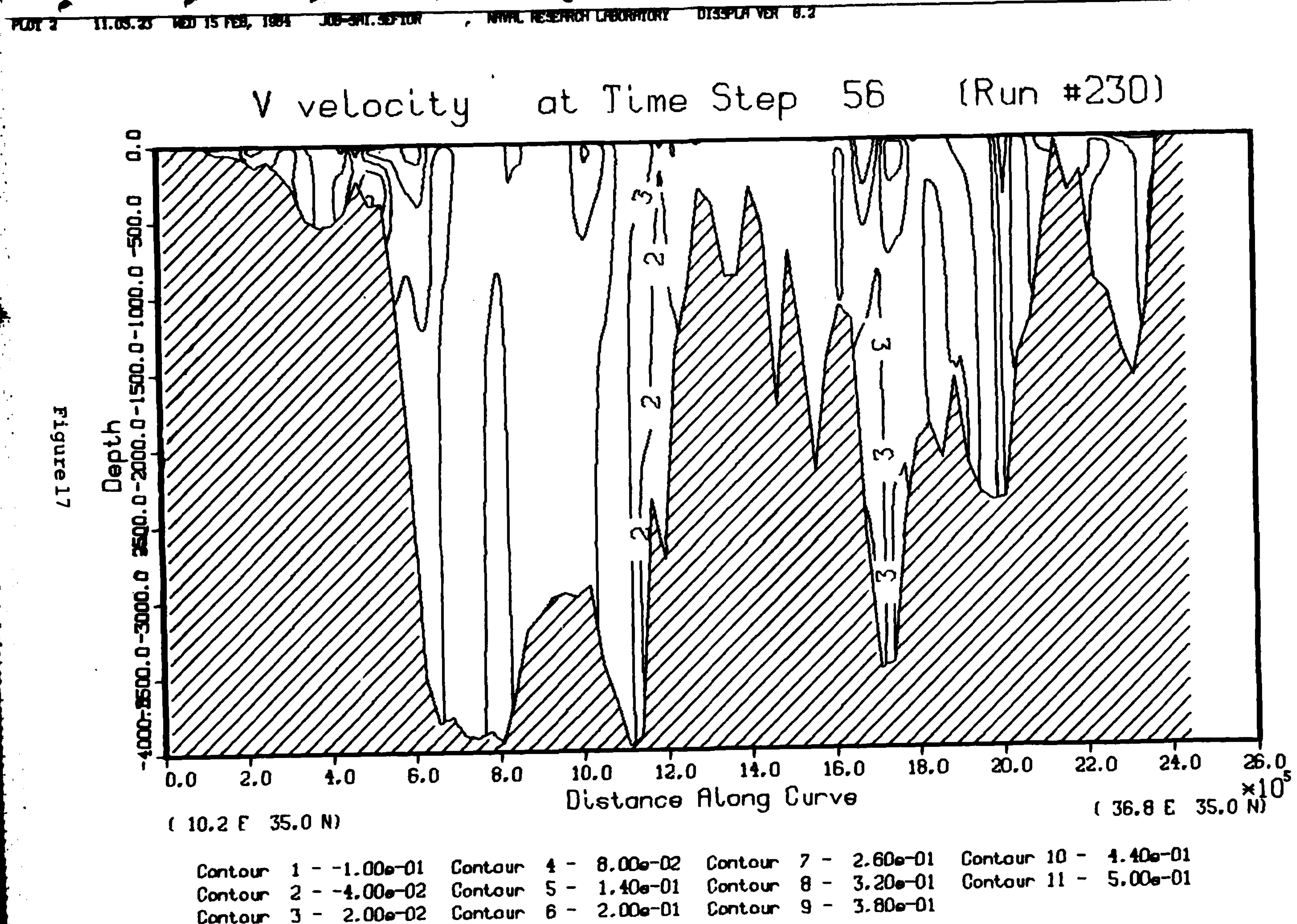

PLOT 2 11.05.23 WED 15 FEB, 1984 JOB-SAT.SEFIOR , NAVAL RESEARCH LABORATORY DISSPLA VER 8.2
V velocity at Time Step 56 (Run #230)
Depth
0.0
-500.0
-1000.0
-1500.0
-2000.0
-2500.0
-3000.0
-3500.0
-4000.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
18.0
20.0
22.0
24.0
26.0
×10^5
Distance Along Curve
(10.2 E 35.0 N)
(36.8 E 35.0 N)
Contour 1 - -1.00e-01
Contour 2 - -4.00e-02
Contour 3 - 2.00e-02
Contour 4 - 8.00e-02
Contour 5 - 1.40e-01
Contour 6 - 2.00e-01
Contour 7 - 2.60e-01
Contour 8 - 3.20e-01
Contour 9 - 3.80e-01
Contour 10 - 4.40e-01
Contour 11 - 5.00e-01

Figure 17

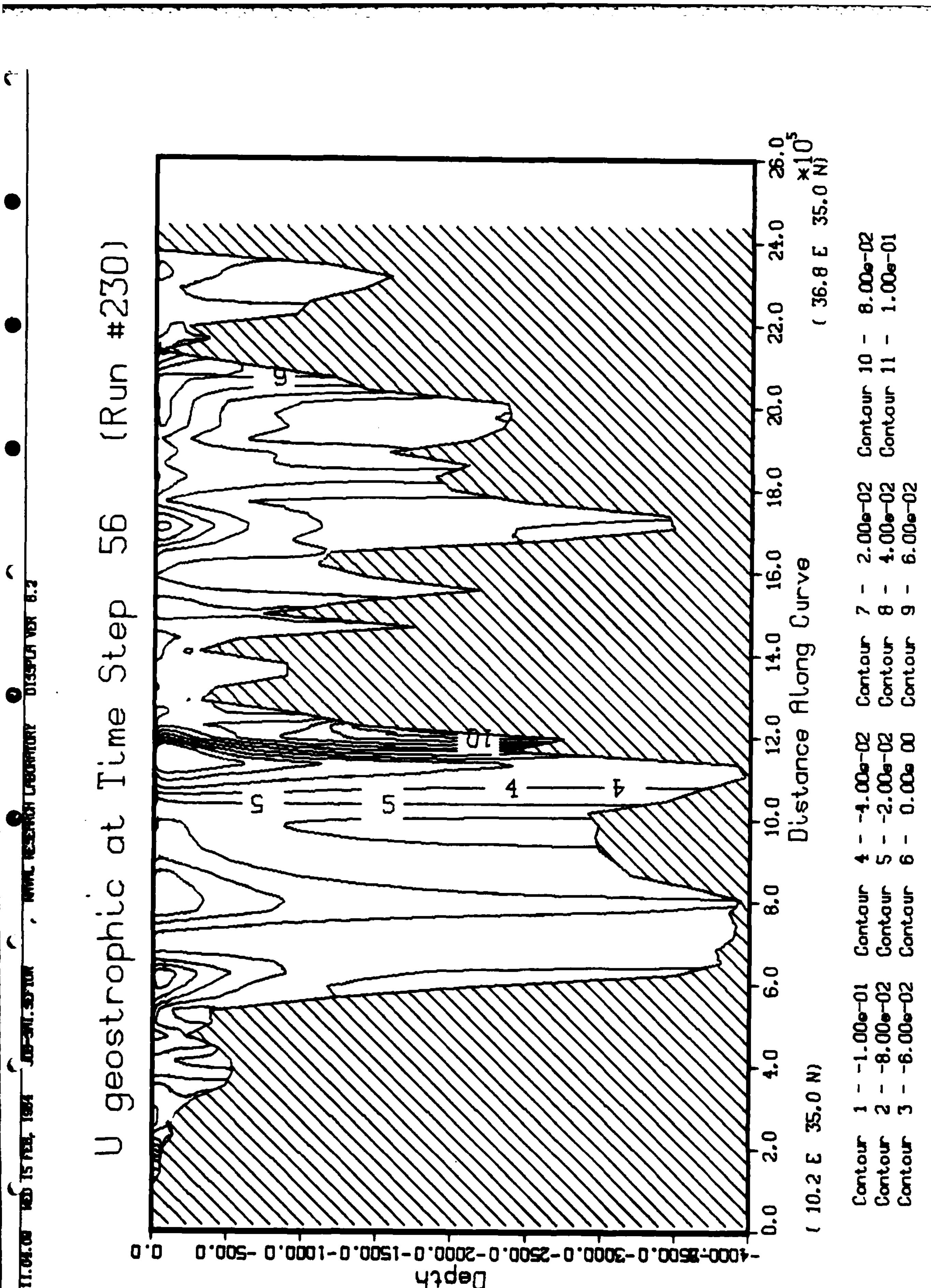
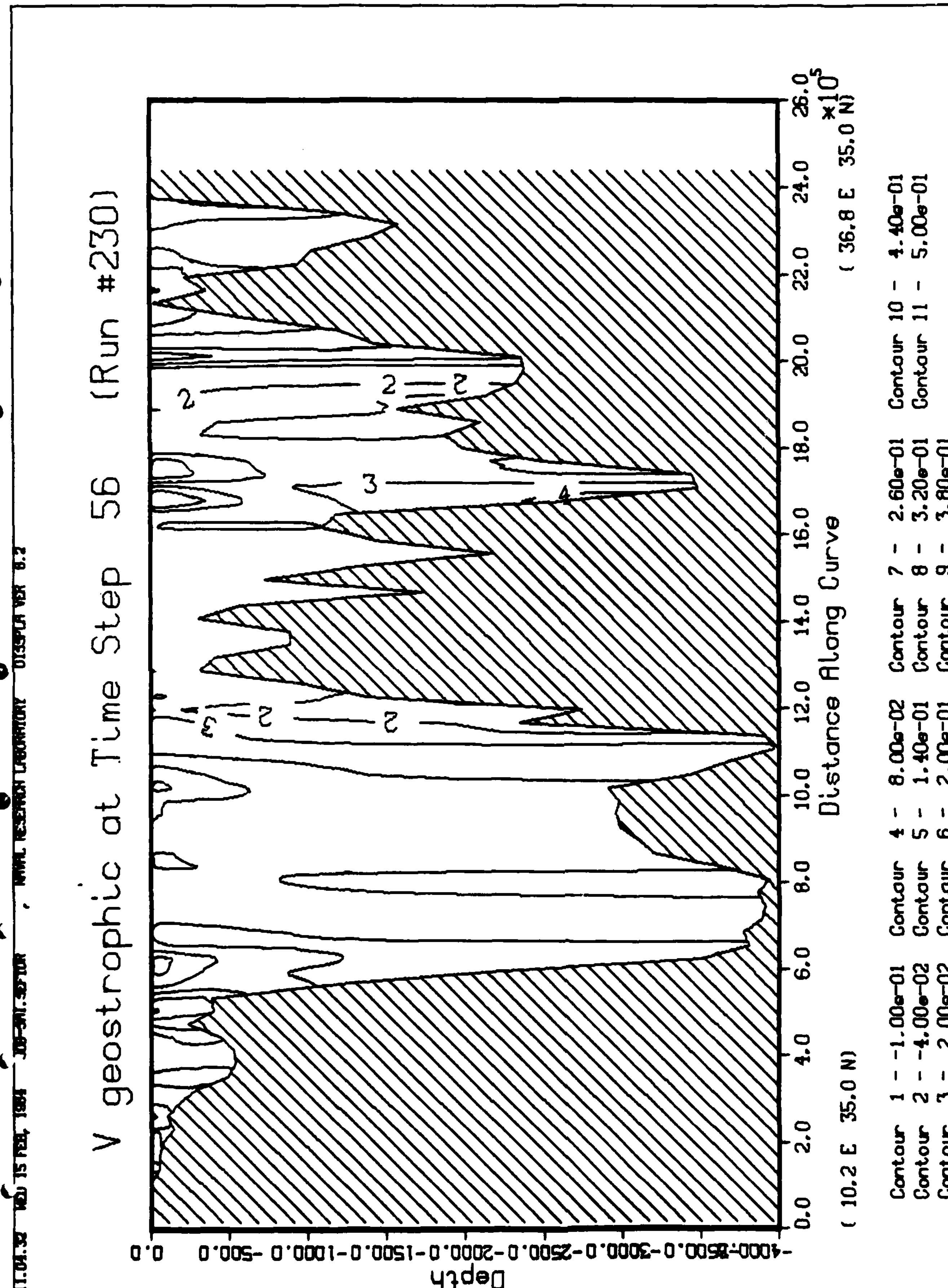

U geostrophic at Time Step 56 (Run #230)
Depth
0.0
-500.0
-1000.0
-1500.0
-2000.0
-2500.0
-3000.0
-3500.0
-4000.0
0.0
2.0
4.0
6.0
8.0
10.0
12.0
14.0
16.0
18.0
20.0
22.0
24.0
26.0
$\times 10^5$
Distance Along Curve
(10.2 E 35.0 N)
(36.8 E 35.0 N)
5
5
4
4
10
9
Contour 1 - -1.00e-01
Contour 2 - -8.00e-02
Contour 3 - -6.00e-02
Contour 4 - -4.00e-02
Contour 5 - -2.00e-02
Contour 6 - 0.00e 00
Contour 7 - 2.00e-02
Contour 8 - 4.00e-02
Contour 9 - 6.00e-02
Contour 10 - 8.00e-02
Contour 11 - 1.00e-01

Figure 18



V geostrophic at Time Step 56 (Run #230)
Depth
0.0 -500.0 -1000.0 -1500.0 -2000.0 -2500.0 -3000.0 -3500.0 -4000.0
Distance Along Curve
0.0 2.0 4.0 6.0 8.0 10.0 12.0 14.0 16.0 18.0 20.0 22.0 24.0 26.0
*10^5
(10.2 E 35.0 N)
(36.8 E 35.0 N)
Contour 1 - -1.00e-01
Contour 2 - -4.00e-02
Contour 3 - 2.00e-02
Contour 4 - 8.00e-02
Contour 5 - 1.40e-01
Contour 6 - 2.00e-01
Contour 7 - 2.60e-01
Contour 8 - 3.20e-01
Contour 9 - 3.80e-01
Contour 10 - 4.40e-01
Contour 11 - 5.00e-01
PLOT 4 11.04.32 WED 15 FEB, 1984 JOB-SNT.SCFTOR , NAVAL RESEARCH LABORATORY DISSPLA VER 8.2

Figure 19

Appendix A

Sigma Code Updates

```
*** SMS189 : DECK AANAML   IS EDITED FROM SPL      FILE
CREATED ON 03/23/82 AT 10:26:18          LAST UPDATED ON 12/05/83 AT 13:40:29 BY SMS VERSION 5.27
LANGUAGE:               USER INFORMATION:

MODSETS PREVIOUSLY APPLIED TO SPL: L081683A  L081683B  L120583A

  1        NAMELIST/ONE/AINV,CG,RC,LSU,RHOO,G,BETAH,BETAV,CV,ZV,DELT,CD,RARO      AANAML     1
  2       . , OMEGA,FO,FIXZS,ADUZS,FIXINT,ADSINT                                   AANAML     2
  3       . , TMAX,TINMN,SALT,UMAX,DT,NSTEP                                        AANAML     3
  4       . , SURFA,FIXTS,THTSLA,THTSPA,THTSSA,THTSLO,THTSPD,THTSSD                AANAML     4
  5       . , FIXUV,THUVLA,THUVPA,THUVSA,THUVLD,THUVPD,THUVSD                      AANAML     5
  6       . , THUVF,HOUR,VFIXUV,PC,CASE,SMOOTH,KS,ZSM                              AANAML     6
  7       . , IPL,NBATTN,DTMULT,DTMAX,DMPGPI                                       AANAML     7
  8       . , AUPMND,ASMZSM,OLDPI,DOFPII,DMPPII                                    AANAML     8
  9       . , NGRPPT,GSURF,NZSITR,ZSFERR,ZSFCHK                                    AANAML     9
 10       . , CJCOEF,DJCOEF,ZSINI,CZSINI,ZSFLCR                                    AANAML    10
 11       . , ZSFA,UVFDIV,REFUM,REFBF,ZSINMN                                       AANAML    11
 12       . , DELTTS,NPPAGE,REFBR,URMULT,TINDEP                                    AANAML    12
 13       . , IFNITR,FIA,FIERR,MOVCHK,FICHK,KVR,URAKN                              AANAML    13
 14       . , FORCIN,FTMTCH,VFIXTS,CONDEN                                          AANAML    14
 15       . , CRM,ZCRM,FLUXIN,DINFL,DBOTBL                                         L081683B   3
 16       . , NREGON,SAVSTR,SAVINC,SAVPLO,SAVPHI,SAVLLO,SAVLHI                     L120583A   3
 17        NAMELIST/TWO/DUM                                                        AANAML    15

          17 ACTIVE LINE(S)            1 INACTIVE LINE(S)

*E ALLMAC
```

```
SMS VERSION 5.27    LISTING OF DECK    A L L M A C                                        01/12/84          PAGE 1

*** SMSIN9 : DECK ALLMAC    IS EDITED FROM SPL       FILE
CREATED ON 06/18/79 AT 13:45:11           LAST UPDATED ON 12/27/83 AT 17:58:37 BY SMS VERSION 5.27
LANGUAGE: FORTRAN       USER INFORMATION:

MODSETS PREVIOUSLY APPLIED TO SPL: VTSFX170  LUVFX184  FIX184A   FIX192A   FIX194A   FIX194B   FIX201F   FIX211A   F256D
     F256E     F292A     F299A     DMPGPI    F310B     UPWINDI   DMPGPI2   F319A     PISWEEP1  PISWEEPS  PISWEEP6  F1207A
     F0104A    F0107A    F0131A    F0204A    F0205A    F0212C    F0221A    F0226D    F0306A    F0307C    F0314B    F0319C
     F0320A    F0331A    F0407C    F0407F    F0408A    F0408B    F0417A    F0417C    F0422B    F0430A    F0513A    F0530A
     F0627A    F0627B    F0798B    F0723A    F0723B    F0725C    F0807A    F0825A    F1010A    F1010B    F1020A    F1021A
     F1023B    F1023E    F1027D    F1223A    F010601A  F010601D  L071501A  L072801A  L080501C  L080601A  L081301A  L032302A
     L040802A  L092382A  L072603B  L072703B  L081503A  L081603A  L082903B  L120503A  L122003A  L122703A

      *                                                                                  L081503A     3
      *     STUDY GULF OF MEXICO                                                         L081503A     4
      *                                                                                  L081503A     5
            PARAMETER IF=52,  JF=35,  KF=5                                               L081503A     6
          . , IFP1=IF+1,JFP1=JF+1,KFP1=KF+1                                              ALLMAC       2
          . , IFM1=IF-1,JFM1=JF-1,KFM1=KF-1                                              ALLMAC       3
          . , IFM2=IF - 2,JFM2=JF - 2                                                    LUVFX184     3
            PARAMETER NREG=10,NREG2=NREG*2                                               L120503A     3
      *                                                                                  ALLMAC       4
            REAL*4 KV,KM,NSQ,LSQ,LAM                                                     F0530A       3
          . , KV322,KE,KVM                                                               F0723A       3
          . , LAMBAR                                                                     L072801A     3
            LOGICAL FIXZS,FIXINT,DELTIS                                                  F0430A       3
          . , FIXTS,FIXUV,VFIXUV,DMPGPI                                                  DMPGPI       3
          . , OLDPI,DMPPII,UVFDIV,FORCIN                                                 F1020A       3
          . , FTMTCH,VFIXTS,CONDEN                                                       L081301A     3
            INTEGER PL,ZSFCHK,OJCOLF,REFBM                                               F0314B       3
          . , REFBF,FICHK,SAVSTR,SAVINC,SAVILO,SAVIHI,SAVJLO,SAVJHI                      L120503A     5
            REAL*8 REFBA,REFBR                                                           F0417C       3
      *                                                                                  ALLMAC       9
      *                                                                                  F0417A       3
      *   BEWARE:  THE ORDER OF VARIABLE IN THE COMMON BLOCKS IS NOT RANDOM!!!           F0417A       4
      *                                                                                  F0417A       5
            COMMON/BASICO/BEGARK                                                         L032302A     3
      *                                                                                  L032302A     4
            COMMON/BASIC1/TSS(IFP1,JFP1,KF,2),USV(IF,JF,KF,2)                            ALLMAC      10
          . , D1(IFP1,JFP1),D2(IFP1,JFP1),D3(IFP1,JFP1),D4(IFP1,JFP1)                    F292A        5
          . , D5(IFP1,JFP1),D6(IFP1,JFP1),D7(IFP1,JFP1),D8(IFP1,JFP1)                    F292A        6
          . , D9(IFP1,JFP1),D10(IFP1,JFP1)                                               F292A        7
          . , TPP(IF,JF,2),TPL(IFP1,JFP1,2),TLL(IF,JF,2),ELLS2(IFP1,JF)                  F0417A       7
          . , EPPS2(IF,JFP1),TSLS2(IF,JF),TSPS2(IF,JF),TFLUX(IF,JF)                      F0825A       3
          . , EPLS2(IFP1,JFP1)                                                           F0417A       9
          . , S(JFP1),C(JFP1),SBAR(JF),CBAR(JF)                                          ALLMAC      13
          . , UPHI1(JFP1),UPBAR(JF),UPBAR1(JF),DLAM1(IFP1)                               F292A        9
          . , DLBAR(IF),DLBAR1(IF),ALFZ(IFP1,JFP1)                                       ALLMAC      15
          . , CI(JFP1),CBAR1(JF),ZBARK(IF,JF,2),DELKZ(IF,JF,2)                           F0205A       6
          . , DZBRK1(IF,JF),ZBRIJK(IFP1,JFP1,2),ALFL(IF,JF),ALFP(IF,JF)                  ALLMAC      10
          . , ZBOT(IF,JF)                                                                F0417A      11
          . , GSL(IF,JFP1),FSP(IF,JFP1),GSP(IFP1,JF),FSL(IFP1,JF)                        ALLMAC      25
          . , GSS(IFP1,JFP1,2),FSS2(IF,JF)                                               ALLMAC      26
          . , UVSAV1(IF,JF,2),UVSAV2(IFP1,JFP1,2)                                        ALLMAC      27
          . , UVSAV3(IFP1,JFP1,2),PISAV1(IF,JFP1,2),ZSFCHR                               F0204A       5
          . , TIMIEP,NSQ(IF,JF,KF),ALFZI(IFP1,JFP1),IFNITR,FIA,FIERR                     F0627B       3
          . , DDTCHK,FICHK,KVM,NOPCVS(JF,KF),UMARN,FIXCHP                                F0825A       5
```

```
     *                                                                            ALLMAC      29
           COMMON/BASIC2//,AV(IF,JF)                                              ALLMAC      30
     . ,   KV(IF,JF,KFP1),KH(IFP1,JFP1,KF),RHO(IF,JF,KF)                          F0205A      12
     . ,   ZSUSTD(IF,JF),ZSUOTS(IF,JF)                                            F0417A      13
     . ,   PIAV(IF,JF,3),PITSI(IF,JFP1,2),PISUR(IFP1,JFP1)                        F0417A      14
     . ,   AIRV,CG,DEL,HL,LSQ,RHOO,G,BETAM,BETAV                                  ALLMAC      35
     . ,   F(IF,JF),LV,ZV,FS(IF,JF),DELT,LD,RARO                                  ALLMAC      36
     . ,   SURF(IFP1,JFP1),UR(IF,JF),FO                                           ALLMAC      37
     . ,   PHI(JF),LAM(IF),Z(IF,JF,4),UASVA(IF,JF,2)                              F256E        3
     . ,   FIXZS,ADSZS,FIXINT,ADSINT,SURFA                                        F0430A       6
     . ,   ZSUR(IF,JF),UZBKK2(IF,JF),FIXTS,THISLA,THISPA,THISSA                   F0417A      16
     . ,   CEVD(IF,JF,2,KFM1),FIXUV,THUVLA,THUVPA,THUVSA,THUVF                    F0205A      16
     . ,   ANS(IFP1,JFP1,KF,2)                                                    FIX192A      5
     . ,   KVOZZ(IF,JF),FSS(IF,JF),ALFZIJ(IF,JF)                                  FIX194B      5
     . ,   VFIXUV,THISLD,THISPD,THISSD,THUVLD,THUVPD,THUVSD                       FIX211A      7
     . ,   NSTEP,PC(60),IPC(3,40),GRADZB(IF,JF),DTMULT,DTMAX,DMPGPI               DMPGPI2      3
     . ,   KE,AUPWND,ASMZSM,ZSM(IF,JF),PI(IF,JF,KF),OLDPI                         PISWEEP1    13
     . ,   ZBPIF(IF,JF),DOFPII,DMPPII,NGRPPT,GSURF,NZSITR,XXXX2,ZSFERR            F0807A       5
     . ,   CJCOEF,OJCOEF,PEINF,ZSFLCR,ZSFA,UVFDIV                                 F0306A       5
     . ,   TTOT,TSUTOT,STOT,SSUTOT,REFBM,REFBF,REFBOT                             F0320A       5
     . ,   MINT(IF,JF),DELTTS,MPPAGE,MIMEAN,MIOSCL                                F0422B       3
     *                                                                            F0320A       6
           COMMON/BASIC3/REFBA(15),REFBR                                          F0417C       5
     . ,   FORCIN,FTMTCH,QRMULT,UMAX,VFIXTS,XXXX3                                 L0715B1A     5
     . ,   IRUN,LAMBAR(IFP1),PHIBAR(JFP1)                                         L0728B1A     5
     . ,   CONDEN,XLSPAC,XPSPAC                                                   L0727B3B     3
     . ,   CKM,ZCKM,FLUXIN(4),DINFL,DBOTBL,ZBOTMM(IF,JF)                          L0829B3B     3
     . ,   NREGON       ,SAVSIR(NREG ),SAVINC(NREG ),SAVPLO(NREG2)                L1220B3A     4
     . ,   SAVPHI(NREG2),SAVLLO(NREG2),SAVLHI(NREG2),SAVILO(2,NREG )              L1220B3A     5
     . ,   SAVIHI(2,NREG),SAVJLO(2,NREG),SAVJHI(2,NREG)                           L1220B3A     6
     *                                                                            L1227B3A     3
     *                                  EDGE-J I-T OR S                           L1227B3A     4
           COMMON/BASIC4/EXTS1(IFP1,KFM1,2,2),EXTS2(JFP1,KFM1,2,2)                L1227B3A     5
     . ,               EXUV1(IF   ,KFM1,2,2),EXUV2(JF   ,KFM1,2,2)                L1227B3A     6
     . ,               EXPI1(I=   ,KFM1,2  ),EXPI2(JF   ,KFM1,2  )                L1227B3A     7
     *                                                                            L0323B2A     6
           COMMON/BASICH/VAR1,VAR2,VAR3,VAR4,VAR5,VAR5B,ARY6(60),VAR7,VAR8        L0408B2A     3
     . ,    VAR9,VAR10,VAR11,VAR12,VAR13,VAR14,VAR15,VAR16,VAR17                  L0323B2A     8
     . ,   VAR18,VAR19,VAR20,VAR21,VAR22,VAR23,VAR24,VAR25                        L0323B2A     9
     *                                                                            L0323B2A    10
           COMMON/BASIC9/ENDMRK                                                   L0323B2A    11
     *                                                                            FIX184A      4
```

86 ACTIVE LINE(S)   134 INACTIVE LINE(S)

BE CASE6

```
*** SMS109 : DECK CASE6     IS EDITED FROM SPL       FILE
CREATED ON 03/31/83 AT 11:46:10          LAST UPDATED ON 06/15/83 AT 14:41:13 BY SMS VERSION 5.27
LANGUAGE:               USER INFORMATION:

MODSETS PREVIOUSLY APPLIED TO SPL: L040483B  L040483C  L040583A  L061583A

            SUBROUTINE CASE6                                              CASE6      1
      *                                                                   CASE6      2
      *     THIS ROUTINE MERGES OTS AND CLIMATOLOGY DATA FOR THE          CASE6      3
      *         INITILIZATION                                             CASE6      4
      *                                                                   CASE6      5
      $CALL ALLMAC                                                        CASE6      6
      *                                                                   CASE6      7
            PARAMETER I20=20,NOTS=7,NCLIM=10,NFIELD=NOTS+NCLIM            CASE6      8
            DIMENSION D63(63,63),T1A(I20),T2A(I20),Z1A(I20),Z2A(I20)      CASE6      9
           * ,                  T1B(I20),T2B(I20),Z1B(I20),Z2B(I20)       CASE6     10
           * , DUMIJK(IF,JF,KF),DEPTH(NFIELD),DATA(I20),DEPTHS(NCLIM)     CASE6     11
            INTEGER TYPE,CATNO(NFIELD),CATNOS(NCLIM)                      CASE6     12
            EQUIVALENCE (D63,LEVD),(DUMIJK,RH)                            CASE6     13
            LOGICAL FIRST/T/                                              CASE6     14
            DATA CATNO/"B10+","P14+","P15+","P16+","P17+","P18+","P19+"   CASE6     15
           * ,          "T  S","T040","T060","T080","T100","T150","T200"  CASE6     16
           * ,          "T300","T400","T500"/                             CASE6     17
            DATA DEPTH/0.,60.96,121.92,182.88,243.84,304.80,365.76,0.,400. CASE6    18
           * ,          600.,800.,1000.,1500.,2000.,3000.,4000.,5000./    CASE6     19
            DATA CATNOS/"S000","S005","S010","S020","S060","S100","S200"  CASE6     20
           * ,          "S300","S400","S500"/                             CASE6     21
            DATA DEPTHS/0.,50.,100.,200.,600.,1000.,2000.,3000.,4000.,5000./ CASE6  22
            DATA HLM/200./                                                CASE6     23
      *                                                                   CASE6     24
            IF (NFIELD .GT. KF*2) CALL STOPP                              CASE6     25
           .(" ARRAY ANS TOO SMALL AS WORKING SPACE IN CASE6 $")          CASE6     26
            IF (IF*JF*KFMI .LT. 3969) CALL STOPP                          CASE6     27
           .("LEVD IN CASE5 IS TOO SMALL TO HOLD A 63 BY 63 ARRAYS")      CASE6     28
      *                                                                   CASE6     29
      * *** DO THE TEMPERATURE                                            CASE6     30
      *                                                                   CASE6     31
            M=0                                                           CASE6     32
        100 M=M+1                                                         CASE6     33
            IF (M.GT.NFIELD) GO TO 150                                    CASE6     34
            IF (M.NE.1) GO TO 105                                         CASE6     35
            READ (18) NCATNO,IS,IR,JS,JR                                  CASE6     36
            IF (NCATNT.NE.NOTS) CALL STOPP                                CASE6     37
           .(" WRONG NUMBER OF OTS PARAMETER FIELDS IN CASE6 $")          CASE6     38
        105 CONTINUE                                                      CASE6     39
            IF (M.NE.NOTS+1) GO TO 110                                    CASE6     40
            READ (18) NPAR,IS,IR,JS,JR                                    CASE6     41
            IF (NPAR .NE. NCLIM+NCLIM) CALL STOPP                         L040483B   3
           .(" WRONG NUMBER OF CLIM PARAMETER FIELDS IN CASE6 $")         CASE6     43
        110 CONTINUE                                                      CASE6     44
      *                                                                   CASE6     45
            IE=IR + IS - 1                                                CASE6     46
            JE=JR + JS - 1                                                CASE6     47
      *                                                                   CASE6     48
            READ (19) TYPE                                                CASE6     49
            IF (TYPE .NE. CATNO(M)) CALL STOPP                            CASE6     50
```

```
      .(" WRONG OTS OR CLIM DATA FIELD IN CASE6 ")                      CASE6       51
*                                                                       CASE6       52
      READ (10) ((D63(I,J),I=IS,IE),J=JS,JE)                            CASE6       53
*                                                                       CASE6       54
*     PUT INTO WORK ARRAY (ANS) BY INTERPOLATION                        CASE6       55
*                                                                       CASE6       56
      CALL COORDC(D63,63,PHIBAR,LAMBAR,ANS(1,1,N,1),IFP1,D1,D5,D7,D9    CASE6       57
     . , IF,JF,IFP1,JFP1,1,1,FIRST)                                     CASE6       58
      IF (FIRST) FIRST=.FALSE.                                          CASE6       59
      GO TO 100                                                         CASE6       60
*                                                                       CASE6       61
*     ADD UP DIFFERENCES TO GET T, NOT DELTA T                          CASE6       62
*                                                                       CASE6       63
  150 CONTINUE                                                          CASE6       64
*                                                                       CASE6       65
      DO 160 N=2,7                                                      CASE6       66
      DO 160 J=2,JF                                                     CASE6       67
      DO 160 I=2,IF                                                     CASE6       68
      ANS(I,J,N,1)=ANS(I,J,N-1,1) + ANS(I,J,N,1)                        CASE6       69
  160 CONTINUE                                                          CASE6       70
*                                                                       CASE6       71
* *** FIND DATA AT THE CORRECT DEPTHS USING A SPLINE INTERPOLATION.     CASE6       72
*       THIS PROCEDURE WILL BE VERY INEFFICIENT.  BUT IT DOESN'T        CASE6       73
*       MATTER SINCE WE ONLY DO IT ONCE PER RUN.                        CASE6       74
*                                                                       CASE6       75
*     THIS PROCEDURE INVOLVES THE BLENDING OF OTS AND CLIMATOLOGY.      CASE6       76
*       THE OTS IS USED WITHOUT MODIFICATION.  THE CLIMATOLOGY IS       CASE6       77
*       CORRECTED.                                                      CASE6       78
*                                                                       CASE6       79
*     LOOP OVER MESH                                                    CASE6       80
*                                                                       CASE6       81
      DO 280 I=2,IF                                                     CASE6       82
      DO 280 J=2,JF                                                     CASE6       83
*                                                                       CASE6       84
      DO 220 N=1,NFIELD                                                 CASE6       85
  220 DATA(N)=ANS(I,J,N,1)                                              CASE6       86
*                                                                       CASE6       87
      CALL CUSPLN(DEPTH          ,DATA          ,NOTS ,Y1A,Y2A,Z1A,Z2A  CASE6       88
     . ,          2,DATA(NOTS)   ,0)                                    CASE6       89
      CALL CUSPLN(DEPTH(NOTS+1),DATA(NOTS+1),NCLIM,Y1B,Y2B,Z1B,Z2B      CASE6       90
     . ,          2,DATA(NFIELD),0)                                     CASE6       91
*                                                                       CASE6       92
      DO=DEPTH(NOTS)                                                    L061503A     3
      DELTT=DATA(NOTS) - SPLINE(DO,DEPTH(NOTS+1),DATA(NOTS+1)           L040503A     3
     .                                          ,Y2B,NCLIM)             CASE6       95
      DELTP=Y1A(NOTS) - .1*(SPLINE(DO+5.,DEPTH(NOTS+1),DATA(NOTS+1)     L040403C     3
     . , Y2B,NCLIM) -       SPLINE(DO-5.,DEPTH(NOTS+1),DATA(NOTS+1)     CASE6       97
     . , Y2B,NCLIM))                                                    CASE6       98
*                                                                       CASE6       99
      DO 260 K=2,KFM1                                                   CASE6      100
      ZLL=DUM1JK(I,J,K)                                                 CASE6      101
*                                                                       CASE6      102
      IF (ZLL - DO) 240,240,250                                         CASE6      103
  240 CONTINUE                                                          CASE6      104
      T63(I,J,K,1)=SPLINE(ZLL,DEPTH,DATA,Y2A,NOTS)                      CASE6      105
      GO TO 260                                                         CASE6      106
*                                                                       CASE6      107
  250 CONTINUE                                                          CASE6      108
```

```
      EF=(1. + (ZEE-DO)/HCH) * EXP((DO-ZEE)/HCH)                              CASE6     109
      GE=(ZEE-DO)*EXP((DO-ZEE)/HCH)                                           CASE6     110
*                                                                             CASE6     111
      TSS(I,J,K,1)=SPLINE(ZEE,DEPTH(NOTS+1),DATA(NOTS+1),Y2B,NCLIM)           CASE6     112
     *           +DELTT*EF + DELTP*GE                                         L040583A    5
*                                                                             CASE6     114
  260 CONTINUE                                                                CASE6     115
*                                                                             CASE6     116
  200 CONTINUE                                                                CASE6     117
*                                                                             CASE6     118
*                                                                             CASE6     119
* *** DO THE SALINITY                                                         CASE6     120
*                                                                             CASE6     121
      N=0                                                                     CASE6     122
  300 N=N+1                                                                   CASE6     123
      IF (N.GT.NCLIM) GO TO 302                                               CASE6     124
*                                                                             CASE6     125
      READ (10) TYPE                                                          CASE6     126
      IF (TYPE.NE.CATNOS(N)) CALL STOPP                                       CASE6     127
     *(" WRONG SALINITY FIELD IN CASE6 $")                                    CASE6     128
*                                                                             CASE6     129
      READ (10) ((D63(I,J),I=IS,IE),J=JS,JE)                                  CASE6     130
*                                                                             CASE6     131
      CALL COORDC(D63,63,PHIBAR,LAMBAR,ANS(1,1,N,1),IFP1,D1,D5,D7,D9          CASE6     132
     * ,         IF,JF,IFP1,JFP1,1,1,FIRST)                                   CASE6     133
      GO TO 300                                                               CASE6     134
*                                                                             CASE6     135
  302 CONTINUE                                                                CASE6     136
*                                                                             CASE6     137
      DO 315 I=2,IF                                                           CASE6     138
      DO 315 J=2,JF                                                           CASE6     139
*                                                                             CASE6     140
      DO 305 N=1,NCLIM                                                        CASE6     141
  305 DATA(N)=ANS(I,J,N,1) - 35.                                              CASE6     142
*                                                                             CASE6     143
      CALL CUSPLN(DEPTHS,DATA,NCLIM,Y1A,Y2A,Z1A,Z2A,2,DEPTHS(NCLIM)           CASE6     144
     * ,          0)                                                          CASE6     145
*                                                                             CASE6     146
      DO 310 K=2,KFM1                                                         CASE6     147
  310 TSS(I,J,K,2)=SPLINE(DUMIJK(I,J,K),DEPTHS,DATA,Y2A,NCLIM)                CASE6     148
*                                                                             CASE6     149
  315 CONTINUE                                                                CASE6     150
*                                                                             CASE6     151
* *** APPLY NO FLUX LATERAL BOUNDARY CONDITIONS, EQN. 3.36 AND 3.37           CASE6     152
*                                                                             CASE6     153
      DO 340 K=2,KFM1                                                         CASE6     154
*                                                                             CASE6     155
*                                                                             CASE6     156
      DO 335 L=1,2                                                            CASE6     157
      DO 325 J=2,JF                                                           CASE6     158
      TSS(1,J,K,L)= TSS(3,J,K,L)                                              CASE6     159
      TSS(IFP1,J,K,L)= TSS(IFM1,J,K,L)                                        CASE6     160
  325 CONTINUE                                                                CASE6     161
*                                                                             CASE6     162
      DO 330 I=1,IFP1                                                         CASE6     163
      TSS(I,1,K,L)= TSS(I,3,K,L)                                              CASE6     164
      TSS(I,JFP1,K,L)= TSS(I,JFM1,K,L)                                        CASE6     165
  330 CONTINUE                                                                CASE6     166
```

```
      335 CONTINUE
*
      340 CONTINUE
*
*         NOW APPLY NO FLUX CONDITION AT TOP AND BOTTOM
*
          DO 350 L=1,2
          DO 350 I=1,IFP1
          DO 350 J=1,JFP1
          TSS(I,J,1,L)=TSS(I,J,2,L)
          TSS(I,J,KF,L)=TSS(I,J,KFM1,L)
      350 CONTINUE
*
*         SET INITIAL VELOCITIES TO ZERO
*
          USV=0.
*
*         SET WIND AND SOLAR FLUX TO ZERO  (THEY CAN BE
*           OVERRIDDEN BY THE FORCING)
*
          UASVA=0.
          UR=0.
*
*
          RETURN
          END
```

192 ACTIVE LINE(S)   5 INACTIVE LINE(S)

SE CASE7

```
*** SMS109 : DECK CASE7     IS EDITED FROM SPL       FILE
CREATED ON 06/16/83 AT 11:56:13          LAST UPDATED ON 06/16/83 AT 11:56:13 BY SMS VERSION 5.27
LANGUAGE:               USER INFORMATION:

            SUBROUTINE CASE7                                                CASE7      1
      *                                                                     CASE7      2
      *     THIS ROUTINE MERGES EOTS AND CLIMATOLOGY DATA FOR THE           CASE7      3
      *          INITILIZATION                                              CASE7      4
      *                                                                     CASE7      5
      $CALL ALLMAC                                                          CASE7      6
      *                                                                     CASE7      7
            PARAMETER I20=25,NOTS=11,NCLIM=10,NFIELD=NOTS+NCLIM             CASE7      8
            DIMENSION D63(63,63),V1A(I20),V2A(I20),Z1A(I20),Z2A(I20)        CASE7      9
           . .                  V1B(I20),V2B(I20),Z1B(I20),Z2B(I20)         CASE7     10
           . . DUMIJK(IF,JF,KF),DEPTH(NFIELD),DATA(I20),DEPTHS(NCLIM)       CASE7     11
            INTEGER TYPE,CATNO(NFIELD),CATNOS(NCLIM)                        CASE7     12
            EQUIVALENCE (D63,CEVD),(DUMIJK,KM)                              CASE7     13
            LOGICAL FIRST/T/                                                CASE7     14
            DATA CATNO/"AAIZ","AAQZ","AARZ","AASZ","AATZ","AAUZ","AAVZ"     CASE7     15
           . .          "AAWZ","AAXZ","AAYZ","AAZZ"                         CASE7     16
           . .          "T  S","T040","T060","T080","T100","T150","T200"    CASE7     17
           . .          "T300","T400","T500"/                               CASE7     18
            DATA DEPTH/0.,25.,50.,75.,100.,125.,150.,200.,250.,300.,400.    CASE7     19
           . .          0.,400.                                             CASE7     20
           . .        600.,800.,1000.,1500.,2000.,3000.,4000.,5000./        CASE7     21
            DATA CATNOS/"S000","S005","S010","S020","S060","S100","S200"    CASE7     22
           . .          "S300","S400","S500"/                               CASE7     23
            DATA DEPTHS/0.,50.,100.,200.,600.,1000.,2000.,3000.,4000.,5000./ CASE7    24
            DATA HLM/200./                                                  CASE7     25
      *                                                                     CASE7     26
            IF (NFIELD .GT. KF+2*2*IF*JF/(IFP1*JFP1)) CALL STOPP            CASE7     27
           .(" ARRAY ANS TOO SMALL AS WORKING SPACE IN CASE7 $")            CASE7     28
            IF (IF*JF*KFM1 .LT. 3969) CALL STOPP                            CASE7     29
           .("CEVD IN CASES IS TOO SMALL TO HOLD A 63 BY 63 ARRAY$")        CASE7     30
      *                                                                     CASE7     31
      * *** DO THE TEMPERATURE                                              CASE7     32
      *                                                                     CASE7     33
            N=0                                                             CASE7     34
        100 N=N+1                                                           CASE7     35
            IF (N.GT.NFIELD) GO TO 150                                      CASE7     36
            IF (N.NE.1) GO TO 105                                           CASE7     37
            READ (10) NCATNO,IS,IR,JS,JR                                    CASE7     38
            IF (NCATNO.NE.NOTS) CALL STOPP                                  CASE7     39
           .(" WRONG NUMBER OF EOTS PARAMETER FIELDS IN CASE7 $")           CASE7     40
        105 CONTINUE                                                        CASE7     41
            IF (N.NE.NOTS+1) GO TO 110                                      CASE7     42
            READ (10) NPAR,IS,IR,JS,JR                                      CASE7     43
            IF (NPAR .NE. NCLIM+NCLIM) CALL STOPP                           CASE7     44
           .(" WRONG NUMBER OF CLIM PARAMETER FIELDS IN CASE7 $")           CASE7     45
        110 CONTINUE                                                        CASE7     46
      *                                                                     CASE7     47
            IE=IR + IS - 1                                                  CASE7     48
            JE=JR + JS - 1                                                  CASE7     49
      *                                                                     CASE7     50
            READ (10) TYPE                                                  CASE7     51
            IF (TYPE .NE. CATNO(N)) CALL STOPP                              CASE7     52
```

```
     .(" WRONG EOTS OR CLIM DATA FIELD IN CASE7 $")
*
      READ (18) ((D63(I,J),I=IS,IE),J=JS,JE)
*
*     PUT INTO WORK ARRAY (ANS) BY INTERPOLATION
*
      CALL COORDC(D63,63,PHIBAR,LAMBAR,ANS(1,1,N,1),IFP1,D1,D5,D7,D9
     . , IF,JF,IFP1,JFP1,1,1,FIRST)
      IF (FIRST) FIRST=.FALSE.
      GO TO 100
*
*     ADD UP DIFFERENCES TO GET T, NOT DELTA T
*
  150 CONTINUE
*
      DO 160 N=2,NOTS
      DO 160 J=2,JF
      DO 160 I=2,IF
      ANS(I,J,N,1)=ANS(I,J,N-1,1) + ANS(I,J,N,1)
  160 CONTINUE
*
* *** FIND DATA AT THE CORRECT DEPTHS USING A SPLINE INTERPOLATION.
*       THIS PROCEDURE WILL BE VERY INEFFICIENT.  BUT IT DOESN'T
*       MATTER SINCE WE ONLY DO IT ONCE PER RUN.
*
*     THIS PROCEDURE INVOLVES THE BLENDING OF EOTS AND CLIMATOLOGY.
*       THE EOTS IS USED WITHOUT MODIFICATION,  THE CLIMATOLOGY IS
*       CORRECTED.
*
*     LOOP OVER MESH
*
      DO 280 I=2,IF
      DO 280 J=2,JF
*
      DO 220 N=1,NFIELD
  220 DATA(N)=ANS(I,J,N,1)
*
      CALL CUSPLN(DEPTH         ,DATA         ,NOTS ,Y1A,Y2A,Z1A,Z2A
     . ,            2,DATA(NOTS)  ,0)
      CALL CUSPLN(DEPTH(NOTS+1),DATA(NOTS+1),NCLIM,Y1B,Y2B,Z1B,Z2B
     . ,            2,DATA(NFIELD),0)
*
      D0=DEPTH(NOTS)
      DELTT=DATA(NOTS) - SPLINE(D0,DEPTH(NOTS+1),DATA(NOTS+1)
     .                                          ,Y2B,NCLIM)
      DELTP=Y1A(NOTS) - .1*(SPLINE(D0+5.,DEPTH(NOTS+1),DATA(NOTS+1)
     . , Y2B,NCLIM) -       SPLINE(D0-5.,DEPTH(NOTS+1),DATA(NOTS+1)
     . , Y2B,NCLIM))
*
      DO 260 K=2,KFM1
      ZEE=DUMIJK(I,J,K)
*
      IF (ZEE - D0) 240,240,250
  240 CONTINUE
      T8S(I,J,K,1)=SPLINE(ZEE,DEPTH,DATA,Y2A,NOTS)
      GO TO 260
*
  250 CONTINUE
```

```
      EF=(1. + (ZEE-D0)/HCH) * EXP((D0-ZEE)/HCH)
      GE=(ZEE-D0)*EXP((D0-ZEE)/HCH)
*
      TSS(I,J,K,1)=SPLINE(ZEE,DEPTH(NOTS+1),DATA(NOTS+1),Y2B,NCLIN)
     *            +DELTT*EF + DELTP*GE
*
  260 CONTINUE
*
  200 CONTINUE
*
*
* *** DO THE SALINITY
*
      N=0
  300 N=N+1
      IF (N.GT.NCLIN) GO TO 302
*
      READ (10) TYPE
      IF (TYPE.NE.CATNOS(N)) CALL STOPP
     .(' WRONG SALINITY FIELD IN CASE7 $')
*
      READ (10) ((D63(I,J),I=IS,IE),J=JS,JE)
*
      CALL COORDC(D63,63,PHIBAR,LAMBAR,ANS(1,1,N,1),IFP1,D1,D5,D7,D9
     * ,         IF,JF,IFP1,JFP1,1,1,FIRST)
      GO TO 300
*
  302 CONTINUE
*
      DO 315 I=2,IF
      DO 315 J=2,JF
*
      DO 305 N=1,NCLIN
  305 DATA(N)=ANS(I,J,N,1) - 35.
*
      CALL LUSPLN(DEPTHS,DATA,NCLIN,Y1A,Y2A,Z1A,Z2A,2,DEPTHS(NCLIN)
     * ,          0)
*
      DO 310 K=2,KFM1
  310 TSS(I,J,K,2)=SPLINE(DUMIJK(I,J,K),DEPTHS,DATA,Y2A,NCLIN)
*
  315 CONTINUE
*
* *** APPLY NO FLUX LATERAL BOUNDARY CONDITIONS, EQN. 3.36 AND 3.37
*
      DO 340 K=2,KFM1
*
*
      DO 335 L=1,2
      DO 325 J=2,JF
      TSS(1,J,K,L)= TSS(3,J,K,L)
      TSS(IFP1,J,K,L)= TSS(IFM1,J,K,L)
  325 CONTINUE
*
      DO 330 I=1,IFP1
      TSS(I,1,K,L)= TSS(I,3,K,L)
      TSS(I,JFP1,K,L)= TSS(I,JFM1,K,L)
  330 CONTINUE
```

```
      335 CONTINUE                                               CASE7   169
*                                                                CASE7   170
      340 CONTINUE                                               CASE7   171
*                                                                CASE7   172
*         NOW APPLY NO FLUX CONDITION AT TOP AND BOTTOM          CASE7   173
*                                                                CASE7   174
          DO 350 L=1,2                                           CASE7   175
          DO 350 I=1,IFP1                                        CASE7   176
          DO 350 J=1,JFP1                                        CASE7   177
          TSS(I,J,1,L)=TSS(I,J,2,L)                              CASE7   178
          TSS(I,J,KF,L)=TSS(I,J,KFM1,L)                          CASE7   179
      350 CONTINUE                                               CASE7   180
*                                                                CASE7   181
*         SET INITIAL VELOCITIES TO ZERO                         CASE7   182
*                                                                CASE7   183
          USV=0.                                                 CASE7   184
*                                                                CASE7   185
*         SET WIND AND SOLAR FLUX TO ZERO  (THEY CAN BE          CASE7   186
*           OVERRIDDEN BY THE FORCING)                           CASE7   187
*                                                                CASE7   188
          UABVA=0.                                               CASE7   189
          QR=0.                                                  CASE7   190
*                                                                CASE7   191
*                                                                CASE7   192
          RETURN                                                 CASE7   193
          END                                                    CASE7   194

          194 ACTIVE LINE(S)          0 INACTIVE LINE(S)

*E CASE8
```

```
*** SMS189 : DECK CASE0     IS EDITED FROM SPL       FILE
CREATED ON 12/20/83 AT 09:59:12          LAST UPDATED ON 12/27/83 AT 17:50:30 BY SMS VERSION 5.27
LANGUAGE:                USER INFORMATION:

MODSETS PREVIOUSLY APPLIED TO SPL: L122783A

           SUBROUTINE CASE0                                                   CASE0       1
      *                                                                       CASE0       2
      *    THIS IS THE MASTER INITIALIZATION ROUTINE FOR A NESTED             CASE0       3
      *      RUN.  IT ACTS AS A MEMORY MANAGER FOR CASE0A, WHERE              CASE0       4
      *      THE ACTUAL INTERPOLATION IS PERFORMED.                           CASE0       5
      *                                                                       CASE0       6
      $CALL ALLMAC                                                            CASE0       7
      *                                                                       CASE0       8
           DIMENSION DUMIJK(IF,JF,KF),DUMK(IF,JF,KF),COUM(IF,JF,KFM1,2)       CASE0       9
         . , OLBAR(1),OPBAR(1),OLAM(1),OPHI(1),OZBOT(IFP1,JFP1)               CASE0      10
           EQUIVALENCE (DUMIJK,KM),(DUMK,KV),(CEVD,COUM)                      CASE0      11
         . , (OLBAR,D6),(OPBAR,D7),(OLAM,D8),(OPHI,D9),(OZBOT,D5)             CASE0      12
      *                                                                       CASE0      13
           INTEGER OIRUN,ON,OSVSTR,OSVINC,ONI1,ONI2,ONJ1,ONJ2,OKF             CASE0      14
         . ,        OKFM1                                                     CASE0      15
      *                                                                       CASE0      16
           DATA LU/81/,LUS/82/                                                L122783A    3
      *                                                                       CASE0      18
           READ (LU) OIRUN,ON                                                 CASE0      19
           READ (LU) OSVSTR,OSVINC,OSVPLO,OSVPHI,OSVLLO,OSVLHI                CASE0      20
           IF (OSVLLO .GT. LAMBAR( 2)) GO TO 40                               CASE0      21
           IF (OSVLHI .LT. LAMBAR(IF)) GO TO 40                               CASE0      22
           IF (OSVPLO .GT. PHIBAR( 2)) GO TO 40                               CASE0      23
           IF (OSVPHI .LT. PHIBAR(JF)) GO TO 40                               CASE0      24
           GO TO 50                                                           CASE0      25
      *                                                                       CASE0      26
        40 CONTINUE                                                           CASE0      27
           PRINT 901,OSVLLO,LAMBAR( 2)                                        CASE0      28
         . ,          OSVLHI,LAMBAR(IF)                                       CASE0      29
         . ,          OSVPLO,PHIBAR( 2)                                       CASE0      30
         . ,          OSVPHI,PHIBAR(JF)                                       CASE0      31
           CALL STOPP(" WRONG LIMITS ON NESTED RUN $")                        CASE0      32
       901 FORMAT (" OSVLLO= ",1PE12.5," MUST BE LESS    THAN ",1PE12.5,/     CASE0      33
         . ,       " OSVLHI= ",1PE12.5," MUST BE GREATER THAN ",1PE12.5,/     CASE0      34
         . ,       " OSVPLO= ",1PE12.5," MUST BE LESS    THAN ",1PE12.5,/     CASE0      35
         . ,       " OSVPHI= ",1PE12.5," MUST BE GREATER THAN ",1PE12.5)      CASE0      36
        50 CONTINUE                                                           CASE0      37
      *                                                                       CASE0      38
           PRINT 902,ON,OIRUN,OSVSTR,OSVINC                                   CASE0      39
       902 FORMAT (////," THIS IS SUBREGION ",I1," OF RUN ",I4                CASE0      40
         . , //," THIS DATA WAS STORED STARTING AT STEP ",I4,", AND"          CASE0      41
         . , " WAS SAVED EVERY ",I3," STEPS")                                 CASE0      42
           READ (LU) ONI1,ONJ1,ONI2,ONJ2,OKF,DT                               CASE0      43
           READ (LU) (OLBAR(I),I=1,ONI1),(OPBAR(J),J=1,ONJ1)                  CASE0      44
         . ,          ( OLAM(I),I=1,ONI2),( OPHI(J),J=1,ONJ2)                 CASE0      45
           WRITE (LUS) ONI1,ONJ1,ONI2,ONJ2,OKF                                L122783A    5
           WRITE (LUS) (OLBAR(I),I=1,ONI1),(OPBAR(J),J=1,ONJ1)                L122783A    6
         . ,          ( OLAM(I),I=1,ONI2),( OPHI(J),J=1,ONJ2)                 L122783A    7
           READ (LU) OFIXZS,OAUXZS,OZSFA,OZSFER,ONZSIT                        CASE0      46
         . ,          OFIXIN,OAUXIN,OFIA ,OFILRR,OIFNIT                       CASE0      47
```

```
         IF (ONI2*ONJ2 .GT. IFP1*JFP1) CALL STOPP                       CASE8      48
        .('  DS TOO SMALL IN CASE8')                                     CASE8      49
         READ (LU) ((O2BOT(I,J),I=1,ONI2),J=1,ONJ2)                     CASE8      50
   *                                                                     CASE8      51
   *     CHECK ON ARRAY WORK SPACE                                       CASE8      52
   *                                                                     CASE8      53
         OKFM1=OKF - 1                                                   CASE8      54
         IC=MAX(ONI1*ONJ1*OKFM1 , ONI2*ONJ2*OKFM1)                      CASE8      55
         IF (IC .GT. IF*JF*KFM1*2) CALL STOPP                           CASE8      56
        .('  WORK SPACE CEVD IS TOO SMALL IN CASE8')                     CASE8      57
   *                                                                     CASE8      58
   *     THE FOLLOWING MESS IS NECESSITATED BY ASC FORTRAN               CASE8      59
   *                                                                     CASE8      60
         CALL CASE8A(                                                    CASE8      61
        .   COUM,ONI1,ONJ1,OKFM1                                         CASE8      62
        . , COUM(1,1,1,2),ONI2,ONJ2,OKFM1                                CASE8      63
        . , OLBAR,ONI1                                                   CASE8      64
        . , OPBAR,ONJ1                                                   CASE8      65
        . , OLAM,ONI2                                                    CASE8      66
        . , OPHI,ONJ2                                                    CASE8      67
        . , ANS,ONI1,ONJ1,KF,2                                           CASE8      68
        . , D10,ONI1,ONJ1                                                CASE8      69
        . , ANS,ONI2,ONJ2,KF,2                                           CASE8      70
        . , DUMIJK,IF,JF,KF                                              CASE8      71
        . , DUMK,IF,JF,KF                                                CASE8      72
        . , TSS,IFP1,JFP1,KF,2                                           CASE8      73
        . , ZSUM,IF,JF                                                   CASE8      74
        . , USV,IF,JF,KF,2                                               CASE8      75
        . , LAMBAR,IFP1                                                  CASE8      76
        . , PHIBAR,JFP1                                                  CASE8      77
        . , LAM,IF                                                       CASE8      78
        . , PHI,JF     ,    LU,LJS)                                      L122783A    9
   *                                                                     CASE8      80
   *     APPLY NO FLUX CONDITIONS AT TOP AND BOTTOM                      CASE8      81
   *                                                                     CASE8      82
         DO 100 L=1,2                                                    CASE8      83
         DO 100 I=2,IF                                                   CASE8      84
         DO 100 J=2,JF                                                   CASE8      85
         TSS(I,J,1 ,L)=TSS(I,J,2   ,L)                                   CASE8      86
         TSS(I,J,KF,L)=TSS(I,J,KFM1,L)                                   CASE8      87
     100 CONTINUE                                                        CASE8      88
   *                                                                     CASE8      89
   *     (NOTE: USV( , ,1, ) WILL REALLY BE SET BY SURFACE FORCING       CASE8      90
   *                                                                     CASE8      91
         DO 110 L=1,2                                                    CASE8      92
         DO 110 I=2,IFM1                                                 CASE8      93
         DO 110 J=2,JFM1                                                 CASE8      94
         USV(I,J,1 ,L)=USV(I,J,2   ,L)                                   CASE8      95
         USV(I,J,KF,L)=USV(I,J,KFM1,L)                                   CASE8      96
     110 CONTINUE                                                        CASE8      97
   *                                                                     CASE8      98
   *     LATERAL BOUNDARY CONDITIONS WILL BE SET IN BOUNDB               CASE8      99
   *                                                                     CASE8     100
   *                                                                     CASE8     101
   *     THE WIND AND SOLAR FLUX CAN BE OVERRIDDEN BY THE FORCING        CASE8     102
   *                                                                     CASE8     103
         UASVA=0.                                                        CASE8     104
         UR=0.                                                           CASE8     105
```

```
SHS VERSION 5.27    LISTING OF DECK    C A S E 8                              01/12/84       PAGE 3
      108    *                                                                    CASE8      106
      109    *                                                                    CASE8      107
      110            RETURN                                                       CASE8      108
      111            END                                                          CASE8      109

                112 ACTIVE LINE(S)          2 INACTIVE LINE(S)
SE CASE8A
```

```
*** SMS109 : DECK CASE8A   IS EDITED FROM SPL       FILE
CREATED ON 12/20/83 AT 09:59:14          LAST UPDATED ON 12/27/83 AT 17:58:40 BY SMS VERSION 5.27
LANGUAGE:                USER INFORMATION:

MODSETS PREVIOUSLY APPLIED TO SPL: L122083B  L122783A

         SUBROUTINE CASE8A(                                                CASE8A       1
        .    OZBIJK,ONI1,ONJ1,OKFM1                                        CASE8A       2
        . ,  OZBK,ONI2,ONJ2,OKFM1                                          CASE8A       3
        . ,  OLBAR,ONI1                                                    CASE8A       4
        . ,  OPBAR,ONJ1                                                    CASE8A       5
        . ,  OLAM,ONI2                                                     CASE8A       6
        . ,  OPHI,ONJ2                                                     CASE8A       7
        . ,  OTSS,ONI1,ONJ1,KF,LOTS                                        CASE8A       8
        . ,  OZSUR,ONI1,ONJ1                                               CASE8A       9
        . ,  OUSV,ONI2,ONJ2,KF,LOUV                                        CASE8A      10
        . ,  ZBIJK,IF,JF,KF                                                CASE8A      11
        . ,  ZBK,IF,JF,KF                                                  CASE8A      12
        . ,  TSS,IFP1,JFP1,KF,LTS                                          CASE8A      13
        . ,  ZSUR,IF,JF                                                    CASE8A      14
        . ,  USV,IF,JF,KF,LUV                                              CASE8A      15
        . ,  LAMBAR,IFP1                                                   CASE8A      16
        . ,  PHIBAR,JFP1                                                   CASE8A      17
        . ,  LAM,IF                                                        CASE8A      18
        . ,  PHI,JF        , LU,LUS)                                       L122783A     3
    *                                                                      CASE8A      20
    *                                                                      CASE8A      21
         INTEGER ONI1,ONJ1,OKFM1,OKFM2,ONI2,ONJ2                           CASE8A      22
        . ,  ONI1M1,ONJ1M1,ONI2M1,ONJ2M1                                   CASE8A      23
         REAL LAMBAR,LAM                                                   CASE8A      24
    *                                                                      CASE8A      25
         DIMENSION OZBIJK(ONI1,ONJ1,OKFM1),OZBK(ONI2,ONJ2,OKFM1)           CASE8A      26
        . ,  OLBAR(ONI1),OPBAR(ONJ1),OLAM(ONI2),OPHI(ONJ2)                 CASE8A      27
        . ,  OTSS(ONI1,ONJ1,KF,LOTS),OZSUR(ONI1,ONJ1)                      CASE8A      28
        . ,  OUSV(ONI2,ONJ2,KF,LOUV),ZBIJK(IF,JF,KF),ZBK(IF,JF,KF)         CASE8A      29
        . ,  TSS(IFP1,JFP1,KF,LTS),ZSUR(IF,JF),USV(IF,JF,KF,LUV)           CASE8A      30
        . ,  LAMBAR(IFP1),PHIBAR(JFP1),LAM(IF),PHI(JF)                     CASE8A      31
    *                                                                      CASE8A      32
         DIMENSION RES(20),A(4),B(4),IOF(4),JOF(4),K4(4),C4(4),ONC4(4)     CASE8A      33
        . ,  USE(4)                                                        CASE8A      34
         DATA IOF/0,1,0,1/,JOF/0,0,1,1/                                    CASE8A      35
         LOGICAL ERROR                                                     CASE8A      36
    *                                                                      CASE8A      37
    *    WARNING OTSS AND OUSV ARE EXTERNALLY EQUIVALENCED                 CASE8A      38
    *                                                                      CASE8A      39
         ONI1M1=ONI1 - 1                                                   CASE8A      40
         ONI2M1=ONI2 - 1                                                   CASE8A      41
         ONJ1M1=ONJ1 - 1                                                   CASE8A      42
         ONJ2M1=ONJ2 - 1                                                   CASE8A      43
         OKFM2=OKFM1 - 1                                                   CASE8A      44
    *                                                                      CASE8A      45
    *    READ OLD DEPTH VALUES                                             CASE8A      46
    *                                                                      CASE8A      47
         READ (LU) (((OZBIJK(I,J,K),I=1,ONI1),J=1,ONJ1),K=2,OKFM1)         CASE8A      48
         READ (LU) (((OZBK  (I,J,K),I=1,ONI2),J=1,ONJ2),K=2,OKFM1)         CASE8A      49
         WRITE (LUS) (((OZBIJK(I,J,K),I=1,ONI1),J=1,ONJ1),K=2,OKFM1)       L122783A     5
```

```
      WRITE (LUS) (((OZBK  (I,J,K),I=1,ONI2),J=1,ONJ2),K=2,OKFM1)        L122783A    6
*                                                                         CASE8A     50
*     READ OLD T+S AND ZSUR                                               CASE8A     51
*                                                                         CASE8A     52
      READ (LU) ((((OTSS(I,J,K,L),I=1,ONI1),J=1,ONJ1),K=1,KF)             CASE8A     53
     *,, L=1,LOTS)                                                        CASE8A     54
      READ (LU) ((OZSUR(I,J),I=1,ONI1),J=1,ONJ1)                          CASE8A     55
*                                                                         CASE8A     56
*     DO THE INTERPOLATION                                                CASE8A     57
*                                                                         CASE8A     58
      DO 200 I=2,IF                                                       CASE8A     59
*                                                                         CASE8A     60
      DO 110 IO=1,ONIIM1                                                  CASE8A     61
      IF (LAMBAR(I).GT.OLBAR(IO) .AND. LAMBAR(I).LE.OLBAR(IO+1))          CASE8A     62
     .  GO TO 115                                                         CASE8A     63
  110 CONTINUE                                                            CASE8A     64
      PRINT 966,(OLBAR(IP),IP=1,ONIIM1)                                   CASE8A     65
  966 FORMAT (/,"  OLBAR= ",/,(10(2X,1PE11.4)))                           CASE8A     66
      ENCODE(80,901,MES) I,LAMBAR(I)                                      CASE8A     67
  901 FORMAT (" LAMBAR(",I2,")=",1PE10.3," NOT ON OLD MESH ..."           CASE8A     68
     . ,"CASE8A$")                                                        CASE8A     69
      CALL STOPP(MES)                                                     CASE8A     70
  115 CONTINUE                                                            CASE8A     71
      A0=( LAMBAR(I)-OLBAR(IO) )  /  ( OLBAR(IO+1)-OLBAR(IO) )            CASE8A     72
      OMA0=1. - A0                                                        CASE8A     73
      A(1)=OMA0                                                           CASE8A     74
      A(2)=A0                                                             CASE8A     75
      A(3)=OMA0                                                           CASE8A     76
      A(4)=A0                                                             CASE8A     77
*                                                                         CASE8A     78
*                                                                         CASE8A     79
      DO 190 J=2,JF                                                       CASE8A     80
*                                                                         CASE8A     81
      DO 120 JO=1,ONJ1M1                                                  CASE8A     82
      IF (PHIBAR(J).GE.OPBAR(JO) .AND. PHIBAR(J).LE.OPBAR(JO+1))          CASE8A     83
     .   GO TO 125                                                        CASE8A     84
  120 CONTINUE                                                            CASE8A     85
      PRINT 967,(OPBAR(JP),JP=1,ONJ1M1)                                   CASE8A     86
  967 FORMAT (/,"  OPBAR= ",/,(10(2X,1PE11.4)))                           CASE8A     87
      ENCODE (80,902,MES) J,PHIBAR(J)                                     CASE8A     88
  902 FORMAT (" PHIBAR(",I2,")=",1PE10.3," NOT ON OLD MESH ... "          CASE8A     89
     . , "CASE8A$")                                                       CASE8A     90
      CALL STOPP (MES)                                                    CASE8A     91
  125 CONTINUE                                                            CASE8A     92
      B0=( PHIBAR(J)-OPBAR(JO) )  /  ( OPBAR(JO+1)-OPBAR(JO) )            CASE8A     93
      OMB0=1. - B0                                                        CASE8A     94
      B(1)=OMB0                                                           CASE8A     95
      B(2)=OMB0                                                           CASE8A     96
      B(3)=B0                                                             CASE8A     97
      B(4)=B0                                                             CASE8A     98
*                                                                         CASE8A     99
*                                                                         CASE8A    100
*        (I,J+1)   3------------------------------4  (I+1,J+1)           CASE8A    101
*                  I                              I                       CASE8A    102
*                  I                              I                       CASE8A    103
*                  I                              I                       CASE8A    104
*                  I                              I                       CASE8A    105
*                  I                              I                       CASE8A    106
```

```
*         (I,J)      1-------------------------------2  (I+1,J)
*
*   X=X(1)* OMA* OMB + X(2)*   A* OMB + X(3)* OMA*   B + X(4)*   A*   B
*
*   X=X(1)*A(1)*B(1) + X(2)*A(2)*B(2) + X(3)*A(3)*B(3) + X(4)*A(4)*B(4)
*
*
* *** DO INTERPOLATION FOR ZSUR
*
      ZSUR(I,J)=0.
      DO 127 N=1,4
      ZSUR(I,J)=ZSUR(I,J) + OZSUR(I0+IOF(N),J0+JOF(N)) * A(N) * B(N)
  127 CONTINUE
*
* *** DO INTERPOLATION FOR T AND S
*
      KFM1=KF - 1
      DO 180 K=2,KFM1
*
*     FIND K AND INTERPOLATION FACTORS FOR 4 CORNERS
*
      DO 140 N=1,4
*
      USE(N)=1.
      DO 130 K0=2,OKFM2
      IF (ZBIJK(I,J,K).GE.OZBIJK(I0+IOF(N),J0+JOF(N),K0  ) .AND.
     .    ZBIJK(I,J,K).LE.OZBIJK(I0+IOF(N),J0+JOF(N),K0+1) ) GO TO 135
  130 CONTINUE
      USE(N)=0.
      K0=2
  135 CONTINUE
      K4(N)=K0
      C4(N)=( ZBIJK(I,J,K) -       OZBIJK(I0+IOF(N),J0+JOF(N),K0) ) /
     .       (OZBIJK(I0+IOF(N),J0+JOF(N),K0+1) -
     .                             OZBIJK(I0+IOF(N),J0+JOF(N),K0) )
      OMC4(N)=1. - C4(N)
  140 CONTINUE
*
*     DO INTERPOLATION
*
      IF (USE(1)+USE(2)+USE(3)+USE(4) .EQ. 0.) GO TO 155
*
      DO 150 L=1,2
      TSS(I,J,K,L)=0.
      DO 150 N=1,4
      TSS(I,J,K,L)=TSS(I,J,K,L) +
     . ( OMC4(N)*OTSS(I0+IOF(N),J0+JOF(N),K4(N)  ,L)
     . +  C4(N)*OTSS(I0+IOF(N),J0+JOF(N),K4(N)+1,L) ) * A(N) * B(N)
     .                                               * USE(N)
  150 CONTINUE
*
      FACT=0.
      DO 152 N=1,4
  152 FACT=FACT + A(N) * B(N) * USE(N)
*
      IF (FACT .NE. 0.) GO TO 154
      LOGICAL DEBUG/.FALSE./
```

```
      IF (.NOT.DEBUG) GO TO 155
      PRINT 961,(A(NP),B(NP),USE(NP),NP=1,4)
  961 FORMAT ('    A, B, AND USE:',/
     . ,(3(2X,1PE11.4)))
      PRINT 964,PHIBAR(J),OPBAR(J0),OPBAR(J0+1)
      PRINT 965,LAMBAR(I),OLBAR(I0),OLBAR(I0+1)
  964 FORMAT (' PHI, OPHI, OPHI+1 =',3(2X,1PE11.4))
  965 FORMAT (' LAM, OLAM, OLAM+1 =',3(2X,1PE11.4))
      PRINT 962,I,J,K,ZBIJK(I,J,K)
  962 FORMAT (/,' ZBIJK(' , I2 , ',' , I2 , ',' , I2 , ')=',1PE11.4)
      DO 160 NP=1,4
      PRINT 963,NP,(OZBIJK(I0+IOF(NP),J0+JOF(NP),KK),KK=2,OKFM1)
  963 FORMAT (//,'  N= ',I2,'      OZBIJK=',/
     . , (7(2X,1PE11.4)))
  160 CONTINUE
      CALL STOPP('  FACT = 0 IN CASE8A $')
  154 CONTINUE
*
      FACT=1. / FACT
      DO 153 L=1,2
  153 TGS(I,J,K,L)=TGS(I,J,K,L) * FACT
*
      GO TO 150
*
  155 CONTINUE
      DO 156 L=1,2
      CALL CASE8B(TGS(I,J,K,L),ZBIJK(I,J,K),OZBIJK,ONI1,ONJ1,OKFM1
     . , OTGS(1,1,1,L),ONI1,ONJ1,KF,IOF,JOF,I0,J0,2,OKFM1,A,B
     . , ERROR)
      IF (ERROR) GO TO 157
  156 CONTINUE
      GO TO 150
  157 CONTINUE
      PRINT 940,I,J,K,ZBIJK(I,J,K)
      DO 999 N=1,4
      PRINT 941,(OZBIJK(I0+IOF(N),J0+JOF(N),KP),KP=2,OKFM1)
  999 CONTINUE
*
  940 FORMAT ('  ZBIJK(', I2 , ',' , I2 , ',' , I2 ,')= ',1PE11.4)
  941 FORMAT (/,'    OZBIJK=',5(3X,1PE11.4)
     . , /,    ('              ',5(3X,1PE11.4)))
      CALL STOPP(' POINT TOO DEEP IN CASE8A$')
*
  150 CONTINUE
*
  180 CONTINUE
*
  190 CONTINUE
*
  200 CONTINUE
*
*
*     READ OLD USV
*
      READ (LU) ((((OUSV(I,J,K,L),I=1,ONI2),J=1,ONJ2),K=1,KF)
     . , L=1,LMTS)
*
*     DO THE INTERPOLATION
```

```
*
      IFM1=IF - 1
      DO 700 I=2,IFM1
*
      DO 610 IO=1,OMI2M1
      IF (LAM(I).GT.OLAM(IO) .AND. LAM(I).LE.OLAM(IO+1))
     .  GO TO 615
  610 CONTINUE
      ENCODE(80,911,MES) I,LAM(I)
  911 FORMAT (' LAM(',I2,')=',1PE10.3,' NOT ON OLD MESH ...'
     . ,'CASE8A6')
      CALL STOPP(MES)
  615 CONTINUE
      AO=( LAM(I)-OLAM(IO) )  /  ( OLAM(IO+1)-OLAM(IO) )
      OMAO=1. - AO
      A(1)=OMAO
      A(2)=AO
      A(3)=OMAO
      A(4)=AO
*
*
      JFM1=JF - 1
      DO 690 J=2,JFM1
*
      DO 620 JO=1,OMJ2M1
      IF (PHI(J).GE.OPHI(JO) .AND. PHI(J).LE.OPHI(JO+1))
     .  GO TO 625
  620 CONTINUE
      ENCODE (80,912,MES) J,PHI(J)
  912 FORMAT (' PHI(',I2,')=',1PE10.3,' NOT ON OLD MESH ... '
     . , 'CASE8A5')
      CALL STOPP (MES)
  625 CONTINUE
      BO=( PHI(J)-OPHI(JO) )  /  ( OPHI(JO+1)-OPHI(JO) )
      OMBO=1. - BO
      B(1)=OMBO
      B(2)=OMBO
      B(3)=BO
      B(4)=BO
*
*
      DO 680 K=2,KFM1
*
*     FIND K AND INTERPOLATION FACTORS FOR 4 CORNERS
*
      DO 640 N=1,4
*
      USE(N)=1.
      DO 630 KO=2,OKFM2
      IF (ZBK(I,J,K).GE.OZBK(IO+IOF(N),JO+JOF(N),KO  ) .AND.
     .    ZBK(I,J,K).LE.OZBK(IO+IOF(N),JO+JOF(N),KO+1) ) GO TO 635
  630 CONTINUE
      USE(N)=0.
      KO=2
  635 CONTINUE
      K4(N)=KO
      C4(N)=( ZBK(I,J,K) -     OZBK(IO+IOF(N),JO+JOF(N),KO) ) /
     .      (OZBK(IO+IOF(N),JO+JOF(N),KO+1) -
```

```
SRS VERSION 5.27    LISTING OF DECK    C A S E 8 A                                        01/12/84        PAGE 6
          *                        OZBK(IO+IOF(N),JO+JOF(N),KO) )                            CASE8A     281
           OML4(N)=1. - L4(N)                                                                CASE8A     282
      640 CONTINUE                                                                           CASE8A     283
     *                                                                                       CASE8A     284
     *     DO INTERPOLATION                                                                  CASE8A     285
     *                                                                                       CASE8A     286
           IF (USE(1)+USE(2)+USE(3)+USE(4) .EQ. 0.) GO TO 655                                CASE8A     287
     *                                                                                       CASE8A     288
           DO 650 L=1,2                                                                      CASE8A     289
           USV(I,J,K,L)=0.                                                                   CASE8A     290
           DO 650 N=1,4                                                                      CASE8A     291
           USV(I,J,K,L)=USV(I,J,K,L) +                                                       CASE8A     292
          . ( OML4(N)*OUSV(IO+IOF(N),JO+JOF(N),K4(N)  ,L)                                    CASE8A     293
          .  +  L4(N)*OUSV(IO+IOF(N),JO+JOF(N),K4(N)+1,L) ) * A(N) * B(N)                    CASE8A     294
          .                                                  * USE(N)                        CASE8A     295
      650 CONTINUE                                                                           CASE8A     296
     *                                                                                       CASE8A     297
           FACT=0.                                                                           CASE8A     298
           DO 652 N=1,4                                                                      CASE8A     299
      652 FACT=FACT + A(N) * B(N) * USE(N)                                                   CASE8A     300
     *                                                                                       L1220838     3
           IF (FACT .EQ. 0.) GO TO 655                                                       L1220838     4
     *                                                                                       CASE8A     301
           FACT=1. / FACT                                                                    CASE8A     302
           DO 653 L=1,2                                                                      CASE8A     303
      653 USV(I,J,K,L)=USV(I,J,K,L) * FACT                                                   CASE8A     304
     *                                                                                       CASE8A     305
           GO TO 658                                                                         CASE8A     306
     *                                                                                       CASE8A     307
      655 CONTINUE                                                                           CASE8A     308
           DO 656 L=1,2                                                                      CASE8A     309
           USV(I,J,K,L)=0.                                                                   CASE8A     310
      656 CONTINUE                                                                           CASE8A     311
     *                                                                                       CASE8A     312
      658 CONTINUE                                                                           CASE8A     313
     *                                                                                       CASE8A     314
      660 CONTINUE                                                                           CASE8A     315
     *                                                                                       CASE8A     316
      690 CONTINUE                                                                           CASE8A     317
     *                                                                                       CASE8A     318
      700 CONTINUE                                                                           CASE8A     319
     *                                                                                       CASE8A     320
     *                                                                                       CASE8A     321
           RETURN                                                                            CASE8A     322
           END                                                                               CASE8A     323

               327 ACTIVE LINE(S)           1 INACTIVE LINE(S)

 IE CASE8B
```

```
*** SMS109 I DECK CASE08   IS EDITED FROM SPL      FILE
CREATED ON 12/12/83 AT 17:15:13          LAST UPDATED ON 12/20/83 AT 09:59:16 BY SMS VERSION 5.27
LANGUAGE:               USER INFORMATION:

MODSETS PREVIOUSLY APPLIED TO SPL: L121303A  L122003A

      SUBROUTINE CASE08(X,DEPTHI,Z,IZ,JZ,KZ,V,IV,JV,KV,IOF,JOF           CASE08      1
     * , I0,J0,K1,K2,A,B,ERROR)                                         CASE08      2
*                                                                       CASE08      3
*     THIS SUBROUTINE DOES A SPLINE EXTRAPOLATION FOR POINTS OUT OF     CASE08      4
*       RANGE                                                           CASE08      5
*                                                                       CASE08      6
      PARAMETER P=20                                                    CASE08      7
      LOGICAL ERROR                                                     CASE08      8
*                                                                       CASE08      9
      DIMENSION Z(IZ,JZ,KZ),V(IV,JV,KV),IOF(4),JOF(4)                   CASE08     10
     * , A(4),B(4)                                                      L121303A    3
*                                                                       CASE08     11
      DIMENSION Y1(P),Y2(P),Z1(P),Z2(P),DEPTH(P),DATA(P),USE(4)         CASE08     12
     * , X4(4)                                                          CASE08     13
*                                                                       CASE08     14
      NK=K2 - K1 + 1                                                    CASE08     15
      IF (NK .GT. P) CALL STOPP(" PARAMETER TOO SMALL IN CASE08")       CASE08     16
      KOF=K1 - 1                                                        CASE08     17
*                                                                       CASE08     18
      DO 30 N=1,4                                                       CASE08     19
*                                                                       CASE08     20
      DO 20 K=K1,K2                                                     CASE08     21
      KK=K - KOF                                                        CASE08     22
      DEPTH(KK)=Z(I0+IOF(N),J0+JOF(N),K)                                CASE08     23
      DATA (KK)=V(I0+IOF(N),J0+JOF(N),K)                                CASE08     24
   20 CONTINUE                                                          CASE08     25
*                                                                       CASE08     26
      CALL CUSPLN(DEPTH,DATA,NK,Y1,Y2,Z1,Z2,1,DEPTH(NK),0)              CASE08     27
*                                                                       CASE08     28
      X4(N)=SPLINE(DEPTHI,DEPTH,DATA,Y2,NK)                             CASE08     29
*                                                                       CASE08     30
      USE(N)=1.                                                         CASE08     31
      IF (DEPTHI .GT. 2*DEPTH(NK)) USE(N)=EXP(-DEPTHI/DEPTH(NK))        L122003A    3
*                                                                       CASE08     33
   30 CONTINUE                                                          CASE08     34
*                                                                       CASE08     35
      ERROR=.FALSE.                                                     CASE08     36
      IF (USE(1)+USE(2)+USE(3)+USE(4) .NE. 0.) GO TO 50                 CASE08     37
      ERROR=.TRUE.                                                      CASE08     38
      RETURN                                                            CASE08     39
*                                                                       CASE08     40
   50 CONTINUE                                                          CASE08     41
      X    =0.                                                          CASE08     42
      FACT=0.                                                           CASE08     43
      DO 60 N=1,4                                                       CASE08     44
      X    =X     + X4(N) * A(N) * B(N) * USE(N)                        CASE08     45
      FACT=FACT +         A(N) * B(N) * USE(N)                          CASE08     46
   60 CONTINUE                                                          CASE08     47
*                                                                       CASE08     48
      X=X / FACT                                                        CASE08     49
```

```
SMS VERSION 5.27    LISTING OF DECK    C A S E 8 B                                   01/12/84          PAGE 2

        51    *                                                                          CASE8B        50
        52    *                                                                          CASE8B        51
        53          RETURN                                                               CASE8B        52
        54          END                                                                  CASE8B        53

              54 ACTIVE LINE(S)            1 INACTIVE LINE(S)

6E ENDND
```

SRS VERSION 5.27   LISTING OF DECK   E X B N D   01/12/84   PAGE 1

*** SRS109 : DECK EXBND   IS EDITED FROM SPL   FILE
CREATED ON 12/27/83 AT 17:58:41   LAST UPDATED ON 01/04/84 AT 13:39:42 BY SRS VERSION 5.27
LANGUAGE: USER INFORMATION:

MODSETS PREVIOUSLY APPLIED TO SPL: L010404A

```
            SUBROUTINE EXBND                                                    EXBND      1
      *                                                                         EXBND      2
      *     THIS ROUTINE READS BOUNDARY DATA WRITTEN BY THE COARSE RUN,         EXBND      3
      *      AND FINDS THE  VALUES ON THE BOUNDARY FOR USE IN THE               EXBND      4
      *      APPLICATION OF THE BOUNDARY CONDITIONS.   IT ACTS AS A            EXBND      5
      *      MEMORY MANAGER FOR EXBNDA, WHERE THE ACTION OCCURS.                EXBND      6
      *                                                                         EXBND      7
      $CALL ALLMAC                                                              EXBND      8
      *                                                                         EXBND      9
            DIMENSION DUMIJK(IF,JF,KF),DUMK(IF,JF,KF),CDUM(IF,JF,KFM1,2)        EXBND     10
           . , OLBAR(1),OPBAR(1),OLAM(1),OPHI(1)                                EXBND     11
            EQUIVALENCE (DUMIJK,KN),(DUMK,KV),(CEVD,CDUM)                       EXBND     12
           . , (OLBAR,D6),(OPBAR,D7),(OLAM,D8),(OPHI,D9)                        EXBND     13
      *                                                                         EXBND     14
            INTEGER ONI1,ONI2,ONJ1,ONJ2,OKF,OKFM1                               EXBND     15
      *                                                                         EXBND     16
            DATA LU/01/,LUS/02/                                                 EXBND     17
      *                                                                         EXBND     18
            REWIND LUS                                                          EXBND     19
            READ (LUS) ONI1,ONJ1,ONI2,ONJ2,OKF                                  EXBND     20
            READ (LUS) (OLBAR(I),I=1,ONI1),(OPBAR(J),J=1,ONJ1)                  EXBND     21
           . ,          ( OLAM(I),I=1,ONI2),( OPHI(J),J=1,ONJ2)                 EXBND     22
      *                                                                         EXBND     23
      *     CHECK ON ARRAY WORK SPACE                                           EXBND     24
      *                                                                         EXBND     25
            OKFM1=OKF - 1                                                       EXBND     26
            IC=MAX(ONI1*ONJ1*OKFM1 , ONI2*ONJ2*OKFM1)                           EXBND     27
            IF (IC .GT. IF*JF*KFM1*2) CALL STOPP                                EXBND     28
           .("  WORK SPACE CEVD IS TOO SMALL IN EXBND$")                        EXBND     29
      *                                                                         EXBND     30
      *     THE FOLLOWING MESS IS NECESSITATED BY ASC FORTRAN                   EXBND     31
      *                                                                         EXBND     32
            CALL EXBNDA(                                                        EXBND     33
           .   CDUM,ONI1,ONJ1,OKFM1                                             EXBND     34
           . , CDUM(1,1,1,2),ONI2,ONJ2,OKFM1                                    EXBND     35
           . , OLBAR,ONI1    ,    OPBAR,ONJ1                                    EXBND     36
           . , OLAM,ONI2    ,    OPHI,ONJ2                                      EXBND     37
           . , AMS,ONI1,ONJ1,KF,2                                               EXBND     38
           . , AMS,ONI2,ONJ2,KF,2                                               EXBND     39
           . , AMS,ONI1,ONJ1,KF                                                 L010404A   3
           . , DUMIJK,IF,JF,KF                                                  EXBND     41
           . , DUMK,IF,JF,KF                                                    EXBND     42
           . , EXTS1,IFP1,KFM1                                                  EXBND     43
           . , EXTS2,JFP1,KFM1                                                  EXBND     44
           . , EXUV1,IF,KFM1                                                    EXBND     45
           . , EXUV2,JF,KFM1                                                    EXBND     46
           . , EXPI1,IF,KFM1                                                    EXBND     47
           . , EXPI2,JF,KFM1                                                    EXBND     48
           . , LAMBAR,IFP1    ,    PHIBAR,JFP1                                  EXBND     49
           . , LAM,IF    ,    PHI,JF    ,    LU,LUS)                            EXBND     50
```

```
SRS VERSION 5.27    LISTING OF DECK    E X B N D                                    01/12/84        PAGE 2

       50    *                                                                        EXBND       51
       51    *                                                                        EXBND       52
       52         RETURN                                                              EXBND       53
       53         END                                                                 EXBND       54

              54 ACTIVE LINE(S)            1 INACTIVE LINE(S)

*E EXBNDA
```

SMS VERSION 5.27   LISTING OF DECK   E X B N D A   01/12/84   PAGE 1

*** SMS109 : DECK EXBNDA   IS EDITED FROM SPL   FILE
CREATED ON 12/27/83 AT 17:50:43   LAST UPDATED ON 01/05/84 AT 13:32:20 BY SMS VERSION 5.27
LANGUAGE: USER INFORMATION:

MODSETS PREVIOUSLY APPLIED TO SPL: L010504A L010504B

```
      SUBROUTINE EXBNDA(
     . . OZBIJK,ONI1,ONJ1,OKFM1
     . . OZOK,ONI2,ONJ2,OKFM1
     . . OLBAR,ONI1    ,   OPBAR,ONJ1
     . . OLAM,ONI2    ,   OPHI,ONJ2
     . . OTUS,ONI1,ONJ1,KF,LOTS
     . . OUSV,ONI2,ONJ2,KF,LOUV
     . . OPI,ONI1,ONJ1,KF
     . . ZBIJK,IF,JF,KF
     . . ZOK,IF,JF,KF
     . . EXTS1,IFP1,KFM1
     . . EXTS2,JFP1,KFM1
     . . EXUV1,IF,KFM1
     . . EXUV2,JF,KFM1
     . . EXPI1,IF,KFM1
     . . EXPI2,JF,KFM1
     . . LAMBAR,IFP1    ,    PHIBAR,JFP1
     . . LAM,IF    ,    PHI,JF    ,    LU,LUS)
C
C
      INTEGER ONI1,ONJ1,OKFM1,OKFM2,ONI2,ONJ2
     . . ONI1M1,ONJ1M1,ONI2M1,ONJ2M1
      REAL LAMBAR,LAM
C
      DIMENSION OZBIJK(ONI1,ONJ1,OKFM1),OZOK(ONI2,ONJ2,OKFM1)
     . . OLBAR(ONI1),OPBAR(ONJ1),OLAM(ONI2),OPHI(ONJ2)
     . . OTUS(ONI1,ONJ1,KF,LOTS),OPI(ONI1,ONJ1,KF)
     . . OUSV(ONI2,ONJ2,KF,LOUV),ZBIJK(IF,JF,KF),ZOK(IF,JF,KF)
     . . EXTS1(IFP1,KFM1,2,2),EXTS2(JFP1,KFM1,2,2)
     . . EXUV1(IF,KFM1,2,2),EXUV2(JF,KFM1,2,2)
     . . EXPI1(IF,KFM1,2),EXPI2(JF,KFM1,2)
     . . LAMBAR(IFP1),PHIBAR(JFP1),LAM(IF),PHI(JF)
     . . ANSR(2)
C
      DIMENSION MES(20),A(4),B(4),IOF(4),JOF(4),K4(4),C4(4),ONC4(4)
     . . USE(4)
      DATA IOF/0,1,0,1/,JOF/0,0,1,1/,TRASH/77F000000/
      LOGICAL ERROR
C
C     WARNING OTUS AND OUSV ARE EXTERNALLY EQUIVALENCED
C
      ONI1M1=ONI1 - 1
      ONI2M1=ONI2 - 1
      ONJ1M1=ONJ1 - 1
      ONJ2M1=ONJ2 - 1
      OKFM2=OKFM1 - 1
C
C     READ OLD DEPTH VALUES
C
      READ (LUS) (((OZBIJK(I,J,K),I=1,ONI1),J=1,ONJ1),K=2,OKFM1)
```

```
      READ (LUS) (((OZBK  (I,J,K),I=1,ONI2),J=1,ONJ2),K=2,OKFN1)     EXBNDA    51
*                                                                    EXBNDA    52
*     READ OLD T$S                                                   EXBNDA    53
*                                                                    EXBNDA    54
      OT$S=TRASH                                                     EXBNDA    55
      CALL RWBND(OT$S,ONI1,ONJ1,KF,LOTS,1,ONI1,1,ONJ1,1,KF,1,LOTS    EXBNDA    56
     . , LU,.TRUE.)                                                  EXBNDA    57
*                                                                    EXBNDA    58
*     DO THE INTERPOLATION                                           EXBNDA    59
*                                                                    EXBNDA    60
      DO 200 I=2,IF                                                  EXBNDA    61
*                                                                    EXBNDA    62
      DO 110 IO=1,ONI1N1                                             EXBNDA    63
      IF (LAMBAR(I).GT.OLBAR(IO) .AND. LAMBAR(I).LE.OLBAR(IO+1))     EXBNDA    64
     .  GO TO 115                                                    EXBNDA    65
  110 CONTINUE                                                       EXBNDA    66
      PRINT 966,(OLBAR(IP),IP=1,ONI1N1)                              EXBNDA    67
  966 FORMAT (/," OLBAR= ",/,(10(2X,1PE11.4)))                       EXBNDA    68
      ENCODE(80,901,MES) I,LAMBAR(I)                                 EXBNDA    69
  901 FORMAT (" LAMBAR(",I2,")=",1PE10.3," NOT ON OLD MESH ..."      EXBNDA    70
     . ,"EXBNDA$")                                                   EXBNDA    71
      CALL STOPP(MES)                                                EXBNDA    72
  115 CONTINUE                                                       EXBNDA    73
      IF (LAMBAR(I) .EQ. OLBAR(ONI1N1)) IO=IO + 1                    L010584A   3
      AO=( LAMBAR(I)-OLBAR(IO) )  /  ( OLBAR(IO+1)-OLBAR(IO) )       EXBNDA    74
      OMAO=1. - AO                                                   EXBNDA    75
      A(1)=OMAO                                                      EXBNDA    76
      A(2)=AO                                                        EXBNDA    77
      A(3)=OMAO                                                      EXBNDA    78
      A(4)=AO                                                        EXBNDA    79
*                                                                    EXBNDA    80
*                                                                    EXBNDA    81
      IF (I.NE.2 .AND. I.NE.IF) GO TO 116                            EXBNDA    82
      JI=1                                                           EXBNDA    83
      GO TO 117                                                      EXBNDA    84
  116 CONTINUE                                                       EXBNDA    85
      JI=JF - 2                                                      EXBNDA    86
  117 CONTINUE                                                       EXBNDA    87
      DO 190 J=2,JF,JI                                               EXBNDA    88
*                                                                    EXBNDA    89
      DO 120 JO=1,ONJ1N1                                             EXBNDA    90
      IF (PHIBAR(J).GT.OPBAR(JO) .AND. PHIBAR(J).LE.OPBAR(JO+1))     L010584A   5
     .   GO TO 125                                                   EXBNDA    92
  120 CONTINUE                                                       EXBNDA    93
      PRINT 967,(OPBAR(JP),JP=1,ONJ1N1)                              EXBNDA    94
  967 FORMAT (/," OPBAR= ",/,(10(2X,1PE11.4)))                       EXBNDA    95
      ENCODE (80,902,MES) J,PHIBAR(J)                                EXBNDA    96
  902 FORMAT (" PHIBAR(",I2,")=",1PE10.3," NOT ON OLD MESH ... "     EXBNDA    97
     . , "EXBNDA$")                                                  EXBNDA    98
      CALL STOPP (MES)                                               EXBNDA    99
  125 CONTINUE                                                       EXBNDA   100
      IF (PHIBAR(J) .EQ. OPBAR(ONJ1N1)) JO=JO + 1                    L010584A   7
      BO=( PHIBAR(J)-OPBAR(JO) )  /  ( OPBAR(JO+1)-OPBAR(JO) )       EXBNDA   101
      OMBO=1. - BO                                                   EXBNDA   102
      B(1)=OMBO                                                      EXBNDA   103
      B(2)=OMBO                                                      EXBNDA   104
      B(3)=BO                                                        EXBNDA   105
      B(4)=BO                                                        EXBNDA   106
```

```
*
*
*       (I,J+1)     3--------------------------------4  (I+1,J+1)
*                   |                                |
*                   |                                |
*                   |                                |
*                   |                                |
*                   |                                |
*       (I,J)       1--------------------------------2  (I+1,J)
*
*
*   X=X(1)* OMA* OMB + X(2)*   A* OMB + X(3)* OMA*   B + X(4)*   A*   B
*
*   X=X(1)*A(1)*B(1) + X(2)*A(2)*B(2) + X(3)*A(3)*B(3) + X(4)*A(4)*B(4)
*
* *** DO INTERPOLATION FOR T AND S
*
      KFM1=KF - 1
      DO 100 K=2,KFM1
*
*     FIND K AND INTERPOLATION FACTORS FOR 4 CORNERS
*
      DO 140 N=1,4
*
      USE(N)=1.
      DO 130 K0=2,KFM2
      IF (ZBIJK(I,J,K).GE.OZBIJK(I0+IOF(N),J0+JOF(N),K0  ) .AND.
     *    ZBIJK(I,J,K).LE.OZBIJK(I0+IOF(N),J0+JOF(N),K0+1) ) GO TO 135
  130 CONTINUE
      USE(N)=0.
      K0=2
  135 CONTINUE
      K4(N)=K0
      C4(N)=( ZBIJK(I,J,K) -      OZBIJK(I0+IOF(N),J0+JOF(N),K0) ) /
     *       (OZBIJK(I0+IOF(N),J0+JOF(N),K0+1) -
     *                            OZBIJK(I0+IOF(N),J0+JOF(N),K0) )
      OMC4(N)=1. - C4(N)
  140 CONTINUE
*
*     DO INTERPOLATION
*
      IF (USE(1)+USE(2)+USE(3)+USE(4) .EQ. 0.) GO TO 155
*
      DO 150 L=1,2
      ANSR(L)=0.
      DO 150 N=1,4
      ANSR(L)=ANSR(L) +
     * ( OMC4(N)*OTBS(I0+IOF(N),J0+JOF(N),K4(N)  ,L)
     *  +  C4(N)*OTBS(I0+IOF(N),J0+JOF(N),K4(N)+1,L) ) * A(N) * B(N)
     *                                               * USE(N)
  150 CONTINUE
*
      FACT=0.
      DO 152 N=1,4
  152 FACT=FACT + A(N) * B(N) * USE(N)
*
      IF (FACT .NE. 0.) GO TO 154
```

```
      LOGICAL DEBUG/.FALSE./
      IF (.NOT.DEBUG) GO TO 155
      PRINT 961,(A(NP),B(NP),USE(NP),NP=1,4)
  961 FORMAT ("     A, B, AND USE:",/
     . ,(3(2X,1PE11.4)))
      PRINT 964,PHIBAR(J),OPBAR(J0),OPBAR(J0+1)
      PRINT 965,LAMBAR(I),OLBAR(I0),OLBAR(I0+1)
  964 FORMAT (" PHI, OPHI, OPHI+1 =",3(2X,1PE11.4))
  965 FORMAT (" LAM, OLAM, OLAM+1 =",3(2X,1PE11.4))
      PRINT 962,I,J,K,ZBIJK(I,J,K)
  962 FORMAT (/," ZBIJK(" , I2 , "," , I2 , "," , I2 , ")=",1PE11.4)
      DO 160 NP=1,4
      PRINT 963,NP,(OZBIJK(I0+IOF(NP),J0+JOF(NP),KK),KK=2,OKFN1)
  963 FORMAT (//,"  N= ",I2,"     OZBIJK=",/
     . , (7(2X,1PE11.4)))
  160 CONTINUE
      CALL STOPP("  FACT = 0 IN EXBNDA 5")
  154 CONTINUE
*
      FACT=1. / FACT
      DO 153 L=1,2
  153 ANSR(L)=ANSR(L) * FACT
*
      GO TO 150
*
  155 CONTINUE
      DO 156 L=1,2
      CALL CASEBB(ANSR(L),ZBIJK(I,J,K),OZBIJK,ONI1,ONJ1,OKFN1
     . , OTSS(1,1,1,L),ONI1,ONJ1,KF,IOF,JOF,I0,J0,2,OKFN1,A,B
     . , ERROR)
      IF (ERROR) GO TO 157
  156 CONTINUE
      GO TO 150
  157 CONTINUE
      PRINT 940,I,J,K,ZBIJK(I,J,K)
      DO 999 N=1,4
      PRINT 941,(OZBIJK(I0+IOF(N),J0+JOF(N),KP),KP=2,OKFN1)
  999 CONTINUE
*
  940 FORMAT ("  ZBIJK(", I2 , "," , I2 , "," , I2 ,")= ",1PE11.4)
  941 FORMAT (/,"    OZBIJK=",5(3X,1PE11.4)
     . , /,   ("            ",5(3X,1PE11.4)))
      CALL STOPP(" POINT TOO DEEP IN EXBNDA1")
*
  150 CONTINUE
*
      DO 170 L=1,2
      IF (I.NE.2) GO TO 162
*
*     LEFT EDGE
*
      EXTS2(J,K,1,L)=ANSR(L)
      GO TO 170
  162 IF (I.EQ.IF) GO TO 168
      IF (J.NE.2) GO TO 164
*
*     BOTTOM EDGE
*
```

```
      EXTS1(I,K,1,L)=ANSR(L)
      GO TO 170
*
*     TOP EDGE
*
  164 EXTS1(I,K,2,L)=ANSR(L)
      GO TO 170
*
*     RIGHT EDGE
*
  168 EXTS2(J,K,2,L)=ANSR(L)
*
  170 CONTINUE
*
  180 CONTINUE
*
  190 CONTINUE
*
  200 CONTINUE
*
*
*     READ OLD UUV
*
      OUUV=TRASH
      CALL RHBND(OUUV,ONI2,ONJ2,KF,LOUV,1,ONI2,1,ONJ2,1,KF,1,LOUV
     . , LU,.TRUE.)
*
*     DO THE INTERPOLATION
*
      IFM1=IF - 1
      DO 700 I=2,IFM1
      XLAM=LAM(I)
*
*     THIS IS NOT CORRECT AT THE CORNERS
*
      IF (I.EQ.2    ) XLAM=LAMBAR(2)
      IF (I.EQ.IFM1) XLAM=LAMBAR(IF)
*
      DO 610 IO=1,ONI2M1
      IF (XLAM.GT.OLAM(IO) .AND. XLAM.LE.OLAM(IO+1))
     .  GO TO 615
  610 CONTINUE
      ENCODE(80,911,MES) I,XLAM
  911 FORMAT (' LAM(',I2,')=',1PE10.3,' NOT ON OLD MESH ...'
     . ,'EXBNDA$')
      CALL STOPP(MES)
  615 CONTINUE
      IF (XLAM .EQ. OLAM(ONI2M1)) IO=IO + 1
      A0=( XLAM-OLAM(IO) )  /  ( OLAM(IO+1)-OLAM(IO) )
      OMA0=1. - A0
      A(1)=OMA0
      A(2)=A0
      A(3)=OMA0
      A(4)=A0
*
*
      JFM1=JF - 1
      IF (I.NE.2 .AND. I.NE.IFM1) GO TO 616
```

```
      JI=1                                                              EXBNDA   280
      GO TO 617                                                         EXBNDA   281
  616 CONTINUE                                                          EXBNDA   282
      JI=JFM1 - 2                                                       EXBNDA   283
  617 CONTINUE                                                          EXBNDA   284
      DO 690 J=2,JFM1,JI                                                EXBNDA   285
      XPHI=PHI(J)                                                       EXBNDA   286
*                                                                       EXBNDA   287
*     THIS IS NOT CORRECT NEAR THE CORNERS                              EXBNDA   288
*                                                                       EXBNDA   289
      IF (J.EQ.2    ) XPHI=PHIBAR(2)                                    EXBNDA   290
      IF (J.EQ.JFM1) XPHI=PHIBAR(JF)                                    EXBNDA   291
*                                                                       EXBNDA   292
      DO 620 J0=1,ONJ2M1                                                EXBNDA   293
      IF (XPHI.GT.OPHI(J0) .AND. XPHI.LE.OPHI(J0+1))                    L0105048   5
     *   GO TO 625                                                      EXBNDA   295
  620 CONTINUE                                                          EXBNDA   296
      ENCODE (80,912,MES) J,XPHI                                        EXBNDA   297
  912 FORMAT (' PHI(',I2,')=',1PE10.3,' NOT ON OLD MESH ... '           EXBNDA   298
     * , 'EXBNDA$')                                                     EXBNDA   299
      CALL STOPP (MES)                                                  EXBNDA   300
  625 CONTINUE                                                          EXBNDA   301
      IF (XPHI .EQ. OPHI(ONJ2M1)) J0=J0 + 1                             L0105048   7
      B0=( XPHI-OPHI(J0) )  /  ( OPHI(J0+1)-OPHI(J0) )                  EXBNDA   302
      OMB0=1. - B0                                                      EXBNDA   303
      B(1)=OMB0                                                         EXBNDA   304
      B(2)=OMB0                                                         EXBNDA   305
      B(3)=B0                                                           EXBNDA   306
      B(4)=B0                                                           EXBNDA   307
*                                                                       EXBNDA   308
*                                                                       EXBNDA   309
      DO 680 K=2,KFM1                                                   EXBNDA   310
*                                                                       EXBNDA   311
*     FIND K AND INTERPOLATION FACTORS FOR 4 CORNERS                    EXBNDA   312
*                                                                       EXBNDA   313
      DO 640 N=1,4                                                      EXBNDA   314
*                                                                       EXBNDA   315
      USE(N)=1.                                                         EXBNDA   316
      DO 630 K0=2,OKFM2                                                 EXBNDA   317
      IF (ZBK(I,J,K).GE.OZBK(I0+IOF(N),J0+JOF(N),K0  ) .AND.            EXBNDA   318
     *    ZBK(I,J,K).LE.OZBK(I0+IOF(N),J0+JOF(N),K0+1) ) GO TO 635      EXBNDA   319
  630 CONTINUE                                                          EXBNDA   320
      USE(N)=0.                                                         EXBNDA   321
      K0=2                                                              EXBNDA   322
  635 CONTINUE                                                          EXBNDA   323
      K4(N)=K0                                                          EXBNDA   324
      C4(N)=( ZBK(I,J,K) -      OZBK(I0+IOF(N),J0+JOF(N),K0) ) /        EXBNDA   325
     *      (OZBK(I0+IOF(N),J0+JOF(N),K0+1) -                           EXBNDA   326
     *                         OZBK(I0+IOF(N),J0+JOF(N),K0) )           EXBNDA   327
      OMC4(N)=1. - C4(N)                                                EXBNDA   328
  640 CONTINUE                                                          EXBNDA   329
*                                                                       EXBNDA   330
*     DO INTERPOLATION                                                  EXBNDA   331
*                                                                       EXBNDA   332
      IF (USE(1)+USE(2)+USE(3)+USE(4) .EQ. 0.) GO TO 655                EXBNDA   333
*                                                                       EXBNDA   334
      DO 650 L=1,2                                                      EXBNDA   335
      ANSR(L)=0.                                                        EXBNDA   336
```

```
         DO 650 N=1,4
         ANSR(L)=ANSR(L) +
        . ( OMG4(N)*DUIV(I0+IOF(N),J0+JOF(N),K4(N)  ,L)
        .  +   C4(N)*DUIV(I0+IOF(N),J0+JOF(N),K4(N)+1,L) ) * A(N) * B(N)
        .                                                   * USE(N)
     650 CONTINUE
    *
         FACT=0.
         DO 652 N=1,4
     652 FACT=FACT + A(N) * B(N) * USE(N)
    *
         IF (FACT .EQ. 0.) GO TO 655
    *
         FACT=1. / FACT
         DO 653 L=1,2
     653 ANSR(L)=ANSR(L) * FACT
    *
         GO TO 658
    *
     655 CONTINUE
         DO 656 L=1,2
         ANSR(L)=0.
     656 CONTINUE
    *
     658 CONTINUE
    *
         DO 670 L=1,2
         IF (I.NE.2) GO TO 662
    *
    *    LEFT EDGE
    *
         EXUV2(J,K,1,L)=ANSR(L)
         GO TO 670
     662 IF (I.EQ.IF) GO TO 668
         IF (J.NE.2) GO TO 664
    *
    *    BOTTOM EDGE
    *
         EXUV1(I,K,1,L)=ANSR(L)
         GO TO 670
    *
    *    TOP EDGE
    *
     664 EXUV1(I,K,2,L)=ANSR(L)
         GO TO 670
    *
    *    RIGHT EDGE
    *
     668 EXUV2(J,K,2,L)=ANSR(L)
    *
     670 CONTINUE
    *
     680 CONTINUE
    *
     690 CONTINUE
    *
     700 CONTINUE
    *
```

```
      *     DO INTERPOLATION FOR PI                                      EXBNDA    395
      *                                                                  EXBNDA    396
            OPI=TRASH                                                    EXBNDA    397
            CALL RWBND(OPI,ONI1,ONJ1,KF,1,1,ONI1,1,ONJ1,1,KF,1,1         EXBNDA    398
          . , LU,.TRUE.)                                                 EXBNDA    399
      *                                                                  EXBNDA    400
      *     DO THE INTERPOLATION                                         EXBNDA    401
      *                                                                  EXBNDA    402
            DO 1200 I=2,IF                                               EXBNDA    403
      *                                                                  EXBNDA    404
            DO 1110 IO=1,ONI1M1                                          EXBNDA    405
            IF (LAMBAR(I).GT.OLBAR(IO) .AND. LAMBAR(I).LE.OLBAR(IO+1))   EXBNDA    406
          .  GO TO 1115                                                  EXBNDA    407
       1110 CONTINUE                                                     EXBNDA    408
            PRINT 1966,(OLBAR(IP),IP=1,ONI1M1)                           EXBNDA    409
       1966 FORMAT (/," OLBAR= ",/,(10(2X,1PE11.4)))                     EXBNDA    410
            ENCODE(80,1901,MES) I,LAMBAR(I)                              EXBNDA    411
       1901 FORMAT (" LAMBAR(",I2,")=",1PE10.3," NOT ON OLD MESH ..."    EXBNDA    412
          . ,"EXBNDAS")                                                  EXBNDA    413
            CALL STOPP(MES)                                              EXBNDA    414
       1115 CONTINUE                                                     EXBNDA    415
            IF (LAMBAR(I) .EQ. OLBAR(ONI1M1)) IO=IO + 1                  L0105048    9
            AO=( LAMBAR(I)-OLBAR(IO) )  /  ( OLBAR(IO+1)-OLBAR(IO) )     EXBNDA    416
            OMAO=1. - AO                                                 EXBNDA    417
            A(1)=OMAO                                                    EXBNDA    418
            A(2)=AO                                                      EXBNDA    419
            A(3)=OMAO                                                    EXBNDA    420
            A(4)=AO                                                      EXBNDA    421
      *                                                                  EXBNDA    422
      *                                                                  EXBNDA    423
            IF (I.NE.2 .AND. I.NE.IF) GO TO 1116                         EXBNDA    424
            JI=1                                                         EXBNDA    425
            GO TO 1117                                                   EXBNDA    426
       1116 CONTINUE                                                     EXBNDA    427
            JI=JF - 2                                                    EXBNDA    428
       1117 CONTINUE                                                     EXBNDA    429
            DO 1190 J=2,JF,JI                                            EXBNDA    430
      *                                                                  EXBNDA    431
            DO 1120 JO=1,ONJ1M1                                          EXBNDA    432
            IF (PHIBAR(J).GT.OPBAR(JO) .AND. PHIBAR(J).LE.OPBAR(JO+1))   L0105048   11
          .  GO TO 1125                                                  EXBNDA    434
       1120 CONTINUE                                                     EXBNDA    435
            PRINT 1967,(OPBAR(JP),JP=1,ONJ1M1)                           EXBNDA    436
       1967 FORMAT (/," OPBAR= ",/,(10(2X,1PE11.4)))                     EXBNDA    437
            ENCODE (80,1902,MES) J,PHIBAR(J)                             EXBNDA    438
       1902 FORMAT (" PHIBAR(",I2,")=",1PE10.3," NOT ON OLD MESH ... "   EXBNDA    439
          . , "EXBNDAS")                                                 EXBNDA    440
            CALL STOPP (MES)                                             EXBNDA    441
       1125 CONTINUE                                                     EXBNDA    442
            IF (PHIBAR(J) .EQ. OPBAR(ONJ1M1)) JO=JO + 1                  L0105048   13
            BO=( PHIBAR(J)-OPBAR(JO) )  /  ( OPBAR(JO+1)-OPBAR(JO) )     EXBNDA    443
            OMBO=1. - BO                                                 EXBNDA    444
            B(1)=OMBO                                                    EXBNDA    445
            B(2)=OMBO                                                    EXBNDA    446
            B(3)=BO                                                      EXBNDA    447
            B(4)=BO                                                      EXBNDA    448
      *                                                                  EXBNDA    449
      *                                                                  EXBNDA    450
```

```
*       (I,J+1)   3-------------------------------4  (I+1,J+1)
*                 |                               |
*                 |                               |
*                 |                               |
*                 |                               |
*                 |                               |
*       (I,J)     1-------------------------------2  (I+1,J)
*
*
*   X=X(1)* OMA* OMB + X(2)*   A* OMB + X(3)* OMA*   B + X(4)*   A*   B
*
*   X=X(1)*A(1)*B(1) + X(2)*A(2)*B(2) + X(3)*A(3)*B(3) + X(4)*A(4)*B(4)
*
*
* *** DO INTERPOLATION FOR PI
*
      KFM1=KF - 1
      DO 1100 K=2,KFM1
*
*     FIND K AND INTERPOLATION FACTORS FOR 4 CORNERS
*
      DO 1140 N=1,4
*
      USE(N)=1.
      DO 1130 KO=2,OKFM2
      IF (ZBK(I,J,K).GE.OZBK(IO+IOF(N),JO+JOF(N),KO  ) .AND.
     *    ZBK(I,J,K).LE.OZBK(IO+IOF(N),JO+JOF(N),KO+1) ) GO TO 1135
 1130 CONTINUE
      USE(N)=0.
      KO=2
 1135 CONTINUE
      K4(N)=KO
      C4(N)=( ZBK(I,J,K) -       OZBK(IO+IOF(N),JO+JOF(N),KO) ) /
     *       (OZBK(IO+IOF(N),JO+JOF(N),KO+1) -
     *                            OZBK(IO+IOF(N),JO+JOF(N),KO) )
      OMC4(N)=1. - C4(N)
 1140 CONTINUE
*
*     DO INTERPOLATION
*
      IF (USE(1)+USE(2)+USE(3)+USE(4) .EQ. 0.) GO TO 1155
*
      L=1
      ANSR(L)=0.
      DO 1150 N=1,4
      ANSR(L)=ANSR(L) +
     * ( OMC4(N)*OPI(IO+IOF(N),JO+JOF(N),K4(N)   )
     *  +   C4(N)*OPI(IO+IOF(N),JO+JOF(N),K4(N)+1) ) * A(N) * B(N)
     *                                                  * USE(N)
 1150 CONTINUE
*
      FACT=0.
      DO 1152 N=1,4
 1152 FACT=FACT + A(N) * B(N) * USE(N)
*
      IF (FACT .EQ. 0.) GO  3 1155
*
      FACT=1. / FACT
```

```
      L=1
      ANSR(L)=ANSR(L) * FACT
*
      GO TO 1150
*
 1155 CONTINUE
      L=1
      CALL CASEBB(ANSR(L),ZBK(I,J,K),OZBK,ONI2,ONJ2,OKFN1
     * , OPI(1,1,1),ONI1,ONJ1,KF,IOF,JOF,IO,JO,2,OKFN1,A,B
     * , ERROR)
      IF (ERROR) GO TO 1157
 1156 CONTINUE
      GO TO 1150
 1157 CONTINUE
      PRINT 1940,I,J,K,ZBK(I,J,K)
      DO 1999 N=1,4
      PRINT 1941,(OZBK(IO+IOF(N),JO+JOF(N),KP),KP=2,OKFN1)
 1999 CONTINUE
*
 1940 FORMAT ("  ZBK(", I2 , "," , I2 , "," , I2 ,")= ",1PE11.4)
 1941 FORMAT (/,"    OZBK=",5(3X,1PE11.4)
     * , /,    ("                ",5(3X,1PE11.4)))
      CALL STOPP(" POINT TOO DEEP IN EXBNDAS")
*
 1150 CONTINUE
*
      L=1
      IF (I.NE.2) GO TO 1162
*
*     LEFT EDGE
*
      EXPI2(J,K,1)=ANSR(L)
      GO TO 1170
 1162 IF (I.EQ.IF) GO TO 1160
      IF (J.NE.2) GO TO 1164
*
*     BOTTOM EDGE
*
      EXPI1(I,K,1)=ANSR(L)
      GO TO 1170
*
*     TOP EDGE
*
 1164 EXPI1(I,K,2)=ANSR(L)
      GO TO 1170
*
*     RIGHT EDGE
*
 1160 EXPI2(J,K,2)=ANSR(L)
*
 1170 CONTINUE
*
 1180 CONTINUE
*
 1190 CONTINUE
*
 1200 CONTINUE
*
```

```
      573     *                                                        EXBNDA     567
      574            RETURN                                            EXBNDA     568
      575            END                                               EXBNDA     569

             575 ACTIVE LINE(S)         3 INACTIVE LINE(S)

 BE FLUNST
```

```
*** SMS189 : DECK FLUXST   IS EDITED FROM SPL      FILE
CREATED ON 08/16/83 AT 15:14:31          LAST UPDATED ON 02/28/84 AT 10:58:37 BY SMS VERSION 5.27
LANGUAGE:             USER INFORMATION:

MODSETS PREVIOUSLY APPLIED TO SPL: L081683F  L081883A  L081883B  L082983B  L020384A  L021384A  L022884A

              SUBROUTINE FLUXST                                               L021384A
        *                                                                     L021384A
        *     SETUP IMPOSED FLUX                                              L021384A
        *                                                                     L021384A
        *     IF DINFLS > 1.E10 THEN USE THE OLD FORMULATION (NO SIGN CHANGE) L021384A
        *                                                                     L021384A
        *     IF DINFLS < 1.E10 THEN ADJUST SOMETHING TO GET:                 L021384A
        *                   (INFLOW/OUTFLOW)=ABS(DINFLR)                      L021384A
        *                                                                     L021384A
        *     IF DINFLR > 0    THEN ADJUST DINFLS                             L021384A
        *     IF DINFLR < 0    THEN ADJUST DINFL                              L021384A
        *                                                                     L021384A
        $CALL ALLMAC                                                          L021384A
        *                                                                     L021384A
              INTEGER ITOLER/2/,ICK(4)/0,0,0,0/,MITER/5/                      L021384A
              LOGICAL FUDGE                                                   L021384A
        *                                                                     L021384A
        *     ON K LOOPS, WE ARE WORKING AT K=K+1, SO RANGE IS REALLY         L021384A
        *       FROM 2 -> KF                                                  L021384A
        *                                                                     L021384A
        *     USE ONLY INTERNAL POINTS  (LOOP TO KFM2)                        L021384A
        *                                                                     L021384A
              KFM2=KFM1 - 1                                                   L021384A
        *                                                                     L021384A
              PWEST=0.                                                        L021384A
              PEAST=0.                                                        L021384A
              PSOUTH=0.                                                       L021384A
              PNORTH=0.                                                       L021384A
        *                                                                     L021384A
        *****                                                                 L021384A
        ***** WEST EDGE                                                       L021384A
        *****                                                                 L021384A
              IF (FLUXIN(1) .EQ. 0.) GO TO 190                                L021384A
        *                                                                     L021384A
        *     I AT EDGE SHOULD REALLY BE 1.5, BUT ALL QUANTITIES ARE          L021384A
        *       SYMMETRIC, SO WE CAN USE I=2                                  L021384A
        *                                                                     L021384A
              I=2                                                             L021384A
        *                                                                     L021384A
              ICK(1)=0                                                        L021384A
              R1=0.                                                           L021384A
              S1=0.                                                           L021384A
              IF (DINFLR(1).LT.0.) S1=1000.                                   L021384A
        103   CONTINUE                                                        L021384A
              ICK(1)=ICK(1) + 1                                               L021384A
        *                                                                     L021384A
              K=0                                                             L021384A
              CALL ZSETC(K+2)                                                 L021384A
              CALL FDEP2(K+1,1)                                               L021384A
        *                                                                     L021384A
```

```
            FWEST=0.                                            L021384A   53
            PWEST=0.                                            L021384A   54
      *                                                         L021384A   55
            DO 165 K=1,KFM2                                     L021384A   56
      *                                                         L021384A   57
            CALL ZSET3(K+2)                                     L021384A   58
            CALL KDEP2(K+1,1)                                   L021384A   59
      *                                                         L021384A   60
            CON=1. / DINFL(1)                                   L021384A   61
            DO 105 J=1,JF                                       L021384A   62
            D1(I,J)=ZBARK(I,J,1) * CON                          L021384A   63
      105   D2(I,J)=ZBOTHH(I,J) - ZBARK(I,J,1)                  L021384A   64
      *                                                         L021384A   65
            CON=1. / DBOTBL(1)                                  L021384A   66
            DO 110 J=1,JF                                       L021384A   67
            D1(I,J)=D1(I,J) + 1.                                L021384A   68
      110   D2(I,J)=D2(I,J) * CON                               L021384A   69
      *                                                         L021384A   70
            IF (DINFLS(1) .GT. 1.E10) GO TO 119                 L021384A   71
      *                                                         L021384A   72
            DO 112 J=1,JF                                       L021384A   73
            D5(I,J)=DINFLS(1) - ZBARK(I,J,2)                    L021384A   74
      112   D6(I,J)=DINFLS(1) + ZBARK(I,J,2)                    L021384A   75
      *                                                         L021384A   76
            DO 114 J=1,JF                                       L021384A   77
      114   D6(I,J)=D6(I,J) * D1(I,J)                           L021384A   78
      *                                                         L021384A   79
            DO 116 J=1,JF                                       L021384A   80
      116   D1(I,J)=D5(I,J) / D6(I,J)                           L021384A   81
            GO TO 122                                           L021384A   82
      *                                                         L021384A   83
      119   CONTINUE                                            L021384A   84
            DO 120 J=1,JF                                       L021384A   85
      120   D1(I,J)=1. / D1(I,J)                                L021384A   86
      122   CONTINUE                                            L021384A   87
      *                                                         L021384A   88
            DO 130 J=1,JF                                       L021384A   89
            D3(I,J)=D1(I,J) * D2(I,J)                           L021384A   90
      130   D4(I,J)=ABS(D1(I,J)) + D2(I,J)                      L022884A    4
      *                                                         L021384A   92
            DO 140 J=1,JF                                       L021384A   93
      140   D3(I,J)=D3(I,J) / D4(I,J)                           L021384A   94
      *                                                         L021384A   95
            DO 150 J=1,JF                                       L021384A   96
      150   D4(I,J)=D3(I,J) * ALFL(I,J)                         L021384A   97
      *                                                         L021384A   98
            DO 160 J=1,JF                                       L021384A   99
            FWEST=FWEST +        D4(I,J)                        L021384A  100
      160   PWEST=PWEST + AMAX1(D4(I,J),0.)                     L021384A  101
      *                                                         L021384A  102
      165   CONTINUE                                            L021384A  103
      *                                                         L021384A  104
            IF (DINFLS(1) .GT. 1.E10) GO TO 190                 L021384A  105
      *                                                         L021384A  106
            RATIO=  FWEST / (FWEST - PWEST)                     L021384A  107
            IF (FUDGE(RATIO,DINFLR(1),ITOLER,.TRUE.)) GO TO 185 L021384A  108
            IF (IC(1) .EQ. MITER) GO TO 185                     L021384A  109
      *                                                         L021384A  110
```

```
*     ADJUST                                                L021384A  111
*                                                           L021384A  112
      RO=R1                                                 L021384A  113
      SO=S1                                                 L021384A  114
      S1=DINFLS(1)                                          L021384A  115
      IF (DINFLR(1) .LT. 0.) S1=DINFL(1)                    L021384A  116
      R1=RATIO                                              L021384A  117
      R2=ABS(DINFLR(1))                                     L021384A  118
      CON=S1 + (R2-R1) * (S1-SO) / (R1-RO)                  L021384A  119
      IF (DINFLR(1) .GT. 0.) DINFLS(1)=CON                  L021384A  120
      IF (DINFLR(1) .LT. 0.) DINFL (1)=CON                  L021384A  121
      GO TO 103                                             L021384A  122
*                                                           L021384A  123
185   CONTINUE                                              L021384A  124
      DINFLR(1)=RATIO                                       L021384A  125
*                                                           L021384A  126
190   CONTINUE                                              L021384A  127
*                                                           L021384A  128
*****                                                       L021384A  129
***** EAST EDGE                                             L021384A  130
*****                                                       L021384A  131
      IF (FLUXIN(2) .EQ. 0.) GO TO 290                      L021384A  132
*                                                           L021384A  133
*     1 AT EDGE SHOULD REALLY BE IF-.5, BUT ALL QUANTITIES ARE  L021384A  134
*       SYMMETRIX, SO WE CAN USE I=IF-1                     L021384A  135
*                                                           L021384A  136
      I=IFM1                                                L021384A  137
      ICK(2)=0                                              L021384A  138
      R1=0.                                                 L021384A  139
      S1=0.                                                 L021384A  140
      IF (DINFLR(2).LT.0.) S1=1000.                         L021384A  141
203   CONTINUE                                              L021384A  142
      ICK(2)=ICK(2) + 1                                     L021384A  143
*                                                           L021384A  144
      K=0                                                   L021384A  145
      CALL ZSET3(K+2)                                       L021384A  146
      CALL KDEP2(K+1,1)                                     L021384A  147
*                                                           L021384A  148
      FEAST=0.                                              L021384A  149
      FEAST=0.                                              L021384A  150
*                                                           L021384A  151
      DO 265 K=1,KFM2                                       L021384A  152
*                                                           L021384A  153
      CALL ZSET3(K+2)                                       L021384A  154
      CALL KDEP2(K+1,1)                                     L021384A  155
*                                                           L021384A  156
      CON=1. / DINFL(2)                                     L021384A  157
      DO 205 J=1,JF                                         L021384A  158
      D1(I,J)=ZBARK(I,J,2) * CON                            L021384A  159
205   D2(I,J)=ZBOTHH(I,J) - ZBARK(I,J,2)                    L021384A  160
*                                                           L021384A  161
      CON=1. / DNOTBC(2)                                    L021384A  162
      DO 210 J=1,JF                                         L021384A  163
      D1(I,J)=D1(I,J) + 1.                                  L021384A  164
210   D2(I,J)=D2(I,J) * CON                                 L021384A  165
*                                                           L021384A  166
      IF (DINFLS(2) .GT. 1.E10) GO TO 219                   L021384A  167
*                                                           L021384A  168
```

```
             DO 212 J=1,JF                                      L021384A   169
             D5(I,J)=DINFLS(2) - ZBAR(I,J,2)                    L021384A   170
      212    D6(I,J)=DINFLS(2) + ZBAR(I,J,2)                    L021384A   171
      *                                                         L021384A   172
             DO 214 J=1,JF                                      L021384A   173
      214    D6(I,J)=D6(I,J) * D1(I,J)                          L021384A   174
      *                                                         L021384A   175
             DO 216 J=1,JF                                      L021384A   176
      216    D1(I,J)=D5(I,J) / D6(I,J)                          L021384A   177
             GO TO 222                                          L021384A   178
      *                                                         L021384A   179
      219    CONTINUE                                           L021384A   180
             DO 220 J=1,JF                                      L021384A   181
      220    D1(I,J)=1. / D1(I,J)                               L021384A   182
      222    CONTINUE                                           L021384A   183
      *                                                         L021384A   184
             DO 230 J=1,JF                                      L021384A   185
             D3(I,J)=D1(I,J) * D2(I,J)                          L021384A   186
      230    D4(I,J)=ABS(D1(I,J)) + D2(I,J)                     L022884A     6
      *                                                         L021384A   188
             DO 240 J=1,JF                                      L021384A   189
      240    D3(I,J)=D3(I,J) / D4(I,J)                          L021384A   190
      *                                                         L021384A   191
             DO 250 J=1,JF                                      L021384A   192
      250    D4(I,J)=D3(I,J) * ALFL(I,J)                        L021384A   193
      *                                                         L021384A   194
             DO 260 J=1,JF                                      L021384A   195
             FEAST=FEAST +        D4(I,J)                       L021384A   196
      260    PEAST=PEAST + AMAX1(D4(I,J),0.)                    L021384A   197
      *                                                         L021384A   198
      265    CONTINUE                                           L021384A   199
      *                                                         L021384A   200
             IF (DINFLS(2) .GT. 1.E10) GO TO 290                L021384A   201
      *                                                         L021384A   202
             RATIO=  FEAST / (FEAST - PEAST)                    L021384A   203
             IF (FODGE(RATIO,DINFLR(2),ITOLER,.TRUE.)) GO TO 285 L021384A  204
             IF (IC(2) .EQ. MITER) GO TO 285                    L021384A   205
      *                                                         L021384A   206
      *      ADJUST                                             L021384A   207
      *                                                         L021384A   208
             R0=R1                                              L021384A   209
             S0=S1                                              L021384A   210
             S1=DINFLS(2)                                       L021384A   211
             IF (DINFLR(2) .LT. 0.) S1=DINFL(2)                 L021384A   212
             R1=RATIO                                           L021384A   213
             R2=ABS(DINFLR(2))                                  L021384A   214
             CON=S1 + (R2-R1) * (S1 -S0) / (R1-R0)              L021384A   215
             IF (DINFLR(2) .GT. 0.) DINFLS(2)=CON               L021384A   216
             IF (DINFLR(2) .LT. 0.) DINFL(2)=CON                L021384A   217
             GO TO ...                                          L021384A   218
      *                                                         L021384A   219
      285    CONTINUE                                           L021384A   220
             DINFLR(2)=RATIO                                    L021384A   221
      *                                                         L021384A   222
      290    CONTINUE                                           L021384A   223
      *****                                                     L021384A   224
      ***** SOUTH FLOW                                          L021384A   225
      *****                                                     L021384A   226
```

```
      IF (FLUXIN(3) .EQ. 0.) GO TO 390
*
*     J AT EDGE SHOULD REALLY BE 1.5, BUT ALL QUANTITIES ARE
*       SYMMETRIX, SO WE CAN USE J = 2
*
      J=2
*
      ICK(3)=0
      R1=0.
      S1=0.
      IF (DINFLK(3).LT.0.) S1=1000
303   CONTINUE
      ICK(3)=ICK(3) + 1
*
      K=0
      CALL ZSET3(K+2)
      CALL KDEP2(K+1,1)
*
      FSOUTH=0.
      FSOUTH=0.
*
      DO 365 K=1,KFM2
*
      CALL ZSET3(K+2)
      CALL KDEP2(K+1,1)
*
      CON=1. / DINFL(3)
      DO 305 I=1,IF
      D1(I,J)=ZBARK(I,J,2) * CON
305   D2(I,J)=ZBOTHH(I,J) - ZBARK(I,J,2)
*
      CON=1. / HBOTBL(3)
      DO 310 I=1,IF
      D1(I,J)=D1(I,J) + 1.
310   D2(I,J)=D2(I,J) * CON
*
      IF (DINFLS(3) .GT. 1.E10) GO TO 319
*
      DO 312 I=1,IF
      D5(I,J)=DINFLS(3) - ZBARK(I,J,2)
312   D6(I,J)=DINFLS(3) + ZBARK(I,J,2)
*
      DO 314 I=1,IF
314   D6(I,J)=D6(I,J) * D1(I,J)
*
      DO 316 I=1,IF
316   D1(I,J)=D5(I,J) / D6(I,J)
      GO TO 322
*
319   CONTINUE
      DO 320 I=1,IF
320   D1(I,J)=1. / D1(I,J)
322   CONTINUE
*
      DO 330 I=1,IF
      D3(I,J)=D1(I,J) * D2(I,J)
330   D4(I,J)=ABS(D1(I,J)) + D2(I,J)
*
```

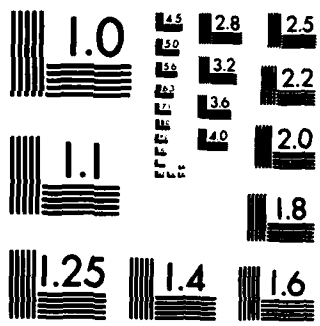
Card identifier column, as printed beside every line: L021384A.

```
           DO 340 I=1,IF
     340   D3(I,J)=D3(I,J) / D4(I,J)
     *
           DO 350 I=1,IF
     350   D4(I,J)=D3(I,J) * ALFP(I,J)
     *
           DO 360 I=1,IF
           FSOUTH=FSOUTH +        D4(I,J)
     360   PSOUTH=PSOUTH + AMAX1(D4(I,J),0.)
     *
     365   CONTINUE
     *
           IF (DINFLS(3) .GT. 1.E10) GO TO 390
     *
           RATIO= - PSOUTH / (FSOUTH - PSOUTH)
           IF (FUDGE(RATIO,DINFLR(3),ITOLER,.TRUE.)) GO TO 385
           IF (ICK(3) .EQ. MITER) GO TO 385
     *
     *     ADJUST
     *
           R0=R1
           S0=S1
           S1=DINFLS(3)
           IF (DINFLR(3) .LT. 0.) S1=DINFL(3)
           R1=RATIO
           R2=ABS(DINFLR(3))
           CON=S1 + (R1-R1) * (S1-S0) / (R1-R0)
           IF (DINFLR(3) .GT. 0.) DINFLS(3)=CON
           IF (DINFLR(3) .LT. 0.) DINFL (3)=CON
           GO TO 303
     *
     385   CONTINUE
           DINFLR(3)=RATIO
     *
     390   CONTINUE
     *****
     ***** NORTH EDGE
     *****
           IF (FLUXIN(4) .EQ. 0.) GO TO 490
     *
     *     J AT EDGE SHOULD REALLY BE JF-.5, BUT ALL QUANTITIES ARE
     *       SYMMETRIX, SO WE CAN USE J = JF-1
     *
           J=JFM1
     *
           ICK(4)=0
           R1=0.
           S1=0.
           IF (DINFLR(4).LT.0.) S1=1000.
     405   CONTINUE
           ICK(4)=ICK(4) + 1
     *
           [illegible]
           [illegible]
           [illegible]
     *
           ENORTH=0.
           FNORTH=0.
```

```
      *                                                L021384A  343
           DO 465 K=1,KFM2                             L021384A  344
      *                                                L021384A  345
           CALL ZSETS(K+2)                             L021384A  346
           CALL KDEP2(K+1,1)                           L021384A  347
      *                                                L021384A  348
           CON=1. / DINFL(4)                           L021384A  349
           DO 405 I=1,IF                               L021384A  350
           D1(I,J)=ZBARK(I,J,2) * CON                  L021384A  351
      405  D2(I,J)=ZBOTHH(I,J) - ZBARK(I,J,2)          L021384A  352
      *                                                L021384A  353
           CON=1. / DBOTBL(4)                          L021384A  354
           DO 410 I=1,IF                               L021384A  355
           D1(I,J)=D1(I,J) + 1.                        L021384A  356
      410  D2(I,J)=D2(I,J) * CON                       L021384A  357
      *                                                L021384A  358
           IF (DINFLS(4) .GT. 1.E10) GO TO 419         L021384A  359
      *                                                L021384A  360
           DO 412 I=1,IF                               L021384A  361
           D5(I,J)=DINFLS(4) - ZBARK(I,J,2)            L021384A  362
      412  D6(I,J)=DINFLS(4) + ZBARK(I,J,2)            L021384A  363
      *                                                L021384A  364
           DO 414 I=1,IF                               L021384A  365
      414  D6(I,J)=D6(I,J) * D1(I,J)                   L021384A  366
      *                                                L021384A  367
           DO 416 I=1,IF                               L021384A  368
      416  D1(I,J)=D5(I,J) / D6(I,J)                   L021384A  369
           GO TO 422                                   L021384A  370
      *                                                L021384A  371
      419  CONTINUE                                    L021384A  372
           DO 420 I=1,IF                               L021384A  373
      420  D1(I,J)=1. / D1(I,J)                        L021384A  374
      422  CONTINUE                                    L021384A  375
      *                                                L021384A  376
           DO 430 I=1,IF                               L021384A  377
           D3(I,J)=D1(I,J) * D2(I,J)                   L021384A  378
      430  D4(I,J)=ABS(D1(I,J)) + D2(I,J)              L022884A   10
      *                                                L021384A  380
           DO 440 I=1,IF                               L021384A  381
      440  D3(I,J)=D3(I,J) / D4(I,J)                   L021384A  382
      *                                                L021384A  383
           DO 450 I=1,IF                               L021384A  384
      450  D4(I,J)=D3(I            J)                  L021384A  385
      *                                                L021384A  386
           DO 460 I=1,IF                               L021384A  387
           FNORTH=FNORTH +      D4(I,J)                L021384A  388
      460  PNORTH=FNORTH + AMAX1(D4(I,J),0.)           L021384A  389
      *                                                L021384A  390
      465  CONTINUE                                    L021384A  391
      *                                                L021384A  392
           IF (DINFLS(4) .GE. 1.E10) GO TO 490         L021384A  393
      *                                                L021384A  394
           RATIO   FNORTH   FNORTH   FNORTH            L021384A  395
           IF ([illegible]) GO TO 485                  L021384A  396
           IF ([illegible]) GO TO 485                  L021384A  397
      *                                                L021384A  398
      *    ADJUST                                      L021384A  399
      *                                                L021384A  400
```

AD-A146 548 SIGMA CODE TESTING(U) SCIENCE APPLICATIONS INC MCLEAN VA J L SEFTOR ET AL. 28 MAR 84 SAI-84/1073 N00014-83-C-0289 2/2

UNCLASSIFIED F/G 8/10 NL



'OPY RESOLUTION TEST CHART

```
         R0=R1
         S0=S1
         S1=DINFLS(4)
         IF (DINFLR(4) .LT. 0.) S1=DINFL(4)
         R1=RATIO
         R2=ABS(DINFLR(4))
         CON=S1 + (R2-R1) * (S1-S0) / (R1-R0)
         IF (DINFLR(4) .GT. 0.) DINFLS(4)=CON
         IF (DINFLR(4) .LT. 0.) DINFL (4)=CON
         GO TO 403
*
 485     CONTINUE
         DINFLR(4)=RATIO
*
 490     CONTINUE
*
         IF (PWEST  .NE. 0.) FLUXIN(1)=FLUXIN(1) / (-PWEST )
         IF (PEAST  .NE. 0.) FLUXIN(2)=FLUXIN(2) / ( PEAST )
         IF (PSOUTH .NE. 0.) FLUXIN(3)=FLUXIN(3) / (-PSOUTH)
         IF (PNORTH .NE. 0.) FLUXIN(4)=FLUXIN(4) / ( PNORTH)
*
         PRINT 900
         PRINT 901, PWEST,FLUXIN(1), PEAST,FLUXIN(2)
     . .           PSOUTH,FLUXIN(3),PNORTH,FLUXIN(4)
  900 FORMAT (///,'  INPUT FLUXES ARE POSITIVE FOR OUTFLOW.  THESE'
     . . ' FLUXES ARE CORRECTED FOR MESH DIRECTION.')
  901 FORMAT (///,' FLUXST:          AREA IN   FLUX/AREA',/
     . .            '          WEST ',1PE10.3,1X,1PE10.3,/
     . .            '          EAST ',1PE10.3,1X,1PE10.3,/
     . .            '         SOUTH ',1PE10.3,1X,1PE10.3,/
     . .            '         NORTH ',1PE10.3,1X,1PE10.3)
*
         PRINT 973,DINFL,DBOTBL,DINFLS,DINFLR,IC
  973 FORMAT (//,T20,'   WEST',T35,'   EAST',T50,'  SOUTH'
     . . T65,'  NORTH',/
     . . T10,' DINFL:',T20,4(1PE10.3,5X),/
     . . T10,'DBOTBL:',T20,4(1PE10.3,5X),/
     . . T10,'DINFLS:',T20,4(1PE10.3,5X),/
     . . T10,'DINFLR:',T20,4(1PE10.3,5X),/
     . . T10,'# ITER:',T20,4(    I10,5X),//)
*
*
         RETURN
         END
```

442 ACTIVE LINE(S)   197 INACTIVE LINE(S)

SMS VERSION 5.27   LISTING OF DECK   P P A R A M   01/12/84   PAGE 1

```
*** SMS109 1 DECK PPARAM    IS EDITED FROM SPL        FILE
CREATED ON 03/23/82 AT 10:26:52          LAST UPDATED ON 08/29/83 AT 14:30:47 BY SMS VERSION 5.27
LANGUAGE:             USER INFORMATION:

MODSETS PREVIOUSLY APPLIED TO SPL: L040202A  L040702A  L040702B  L092302A  L072603B  L081603A  L081603B  L082903A

          SUBROUTINE PPARAM(DT)                                                      L040702A       3
    *                                                                                PPARAM         2
    *     THIS SUBROUTINE PRINTS THE RUN PARAMETERS                                  PPARAM         3
    *                                                                                PPARAM         4
    *CALL ALLMAC                                                                     PPARAM         5
    *                                                                                L040202A       3
    *CALL ASTUPC                                                                     L040202A       4
    *                                                                                PPARAM         6
          PRINT 970,AINV   ,BETAN ,BETAV ,FIXINT,THTSLA,THTSLD                       PPARAM         8
       .            ,CASE  ,CG    ,CV    ,FIXTS ,THTSPA,THTSPD                       PPARAM         9
       .            ,CD    ,DELT  ,DT    ,FIXUV ,THTSSA,THTSSD                       PPARAM        10
       .            ,FORCIN,G     ,HOUR  ,FIXZS ,THUVLA,THUVLD                       PPARAM        11
       .            ,LSQ   ,NSTEP ,OMEGA ,VFIXTS,THUVPA,THUVPD                       PPARAM        12
       .            ,RARO  ,RC    ,RHOO  ,SMOOTH,THUVSA,THUVSD                       PPARAM        13
       .            ,SALT  ,TMAX  ,TIMAM ,VFIXUV,ADBZS ,ADBINT                       PPARAM        14
       .            ,UMAX  ,ZV    ,TINDEP,THUVF ,KS    ,ZSM                          PPARAM        15
      970 FORMAT (///T5,"AINV   =",1PE12.5,T25,"BETAN =",0PF5.2,T45,"BETAV ="        PPARAM        16
       . , F5.2,T70,"FIXINT=",L1,T85,"THTSLA=",F5.2,T105,"THTSLD=",F5.2              PPARAM        17
       . , /,T5,"CASE   =",I2,T25,"CG     =",F5.2,T45,"CV     =",1PE10.3,T70         PPARAM        18
       . , "FIXTS =",L1,T85,"THTSPA=",0PF5.2,T105,"THTSPD=",F5.2,/,T5                PPARAM        19
       . , "CD     =",1PE10.3,T25,"DELT   =",0PF7.2,T45,"DT     =",F10.1             PPARAM        20
       . , T70,"FIXUV =",L1,T85,"THTSSA=",F5.2,T105,"THTSSD=",F5.2                   PPARAM        21
       . , /,T5,"FORCIN=",L1,T25,"G      =",0PF6.2,T45,"HOUR   ="                    PPARAM        22
       . , F6.2,T70,"FIXZS =",L1,T85,"THUVLA=",F5.2,T105,"THUVLD="                   PPARAM        23
       . , F5.2,/,T5,"LSQ    =",1PE10.3,T25,"NSTEP =",I5,T45,"OMEGA ="               PPARAM        24
       . , 1PE12.5,T70,"VFIXTS=",1X,L1,T85,"THUVPA=",0PF5.2,T105,"THUVPD="           PPARAM        25
       . , F5.2,/,T5,"RARO   =",1PE12.5,T25,"RC     =",0PF7.3,T45,"RHOO   ="         PPARAM        26
       . , F10.7,T70,"SMOOTH=",L1,T85,"THUVSA=",F5.2,T105,"THUVSD="                  PPARAM        27
       . , F5.2,/,T5,"SALT   =",F7.2,T25,"TMAX   =",F7.2,T45,"TIMAM =",I4            PPARAM        28
       . , T70,"VFIXUV=",L1,T85,"ADBZS >",F6.3,T105,"ADBINT=",F6.3,/,T5              PPARAM        29
       . , "UMAX   =",F7.2,T25,"ZV     =",1PE10.3,T45,"TINDEP=",0PF7.1               PPARAM        30
       . , T70,"THUVF =",0PF5.2,T85                                                  PPARAM        31
       . , "KS     =",0PF5.2,T105,"ZSM    =",F7.1)                                   PPARAM        32
          PRINT 971,NBATTM,DTMULT,DTMAX ,DAPGPI                                      PPARAM        33
       . , AUPUND,ASRZSM,OLDPI,DAPPII,DOFPII                                         PPARAM        34
       . , NGRPPT,GSURF,NZSITR,ZSFERR                                                PPARAM        35
       . , CJCOEF,OJCOEF,ZSINI,CZSINI                                                PPARAM        36
       . , ZSFCHK,ZSFA,UVFDIV,REFBN,REFBF                                            PPARAM        37
       . , ZSINMN,DELTTS,NPPAGE,REFBR,QRMULT,SURFA                                   PPARAM        38
       . , IFNITR,FIA,FIERR,FICHK,DOVCHK,KVM,UMAXM                                   PPARAM        39
       . , FTATCH,CONDEN,XLSPAC,XPSPAC                                               L092302A       3
      971 FORMAT (/,T5,"NBATTM=",1PE12.5,T25,"DTMULT=",0PF5.2,T45,"DTMAX ="          PPARAM        41
       . , 1PE12.5,T70,"DAPGPI=",L1                                                  PPARAM        42
       . , T85,"AUPUND=",0PF5.2                                                      PPARAM        43
       . , T105,"ASRZSM=",0PF8.3                                                     PPARAM        44
       . , /,T5,"OLDPI =",L1                                                         PPARAM        45
       . , T25,"DAPPII=",L1,T45,"DOFPII=",0PF8.1                                     PPARAM        46
       . , T70,"NGRPPT=",I2,T85,"GSURF =",0PF6.2                                     PPARAM        47
       . , T105,"NZSITR=",I3,/,                        T25,"ZSFERR=",0PF7.3          PPARAM        48
       . , T45,"CJCOEF=",0PF6.3,T70,"OJCOEF=",A4                                     PPARAM        49
```

```
     . . T05,"ZSINI =",I3,T105,"CZSINI=",1PE10.3                  PPARAM     50
     . . /,T5,"ZSFCHK=",I6                                        PPARAM     51
     . . T45,"ZSFA  =",1PE10.3,T70,"UVFDIV=",L1                   PPARAM     52
     . . T05,"REFOR =",I2,T105,"REFOF =",I6                       PPARAM     53
     . . /,T5,"ZSINNN=",I4,T25,"DELTTS=",L1,T45,"NPPAGE=",I3      PPARAM     54
     . . T70,"REFOR =",0PF6.3,T05,"QRMULT=",F6.3,T105,"SURFA =",1PE10.3   PPARAM     55
     . . /,T5,"IFNITR=",I3,T25,"FIA   =",1PE10.3,T45,"FIERR =",0PF7.3    PPARAM     56
     . . T70,"ZICHK =",I3,T05,"DDTCHK=",1PE10.3,T105,"KVN   =",1PE10.3   PPARAM     57
     . . /,T5,"UMAXN =",0PF7.3,T25,"FTNTCH=",1X,L1,T45,"CONDEN=",1X,L1   PPARAM     58
     . . T70,"XLSPAC=",1PE10.3,T05,"YPSPAC=",1PE10.3)             L092302A    5
      PRINT 972,CAN,ZCAN,FLUXIN,DINFL,DDOTDL                      L0016030    3
  972 FORMAT (/,T5,"CAN   =",1PE10.3,T25,"ZCAN  =",1PE10.3        L0029030    3
     . . T45,"FLUXIN=",1PE10.3,T70,1PE10.3,T93,1PE10.3,T113       L0029030    4
     . . 1PE10.3,/,T5,"DINFL =",1PE10.3,T25,"DDOTDL=",1PE10.3     L0029030    5
     . )                                                          L0016030    7
      IMAX=10                                                     PPARAM     60
      PRINT 910,(PC(III),III=1,IMAX)                              PPARAM     61
  910 FORMAT  (1H0,10X,"PRINT OPTIONS:",/                         PPARAM     62
     . . T11,"P1 =",I6,"  (PRINT INCREMENT FOR TIME)"             PPARAM     63
     . . T71,"P2 =",I6,"  (PRINT INCREMENT FOR T)",/              PPARAM     64
     . . T11,"P3 =",I6,"  (PRINT INCREMENT FOR S)"                PPARAM     65
     . . T71,"P4 =",I6,"  (PRINT INCREMENT FOR U)",/              PPARAM     66
     . . T11,"P5 =",I6,"  (PRINT INCREMENT FOR V)"                PPARAM     67
     . . T71,"P6 =",I6,"  (PRINT INCREMENT FOR Z SURFACE)",/      PPARAM     68
     . . T11,"P7 =",I6,"  (PRINT INCREMENT FOR KM)"               PPARAM     69
     . . T71,"P8 =",I6,"  (PRINT INCREMENT FOR KV)",/             PPARAM     70
     . . T11,"P9 =",I6,"  (PRINT INCREMENT FOR T SURFACE)"        PPARAM     71
     . . T71,"P10=",I6,"  (0 TO SEE INITIAL CONDITIONS)",/        PPARAM     72
     . . T11,"P11=",I6,"  (0 TO SEE NAMELIST 9)"                  PPARAM     73
     . . T71,"P12=",I6,"  (STORE INCREMENT FOR SOUND SPEED)",/    PPARAM     74
     . . T11,"P13=",I6,"  (STORE INCREMENT FOR SURFACE TEMPERATURE)"   PPARAM     75
     . . T71,"P14=",I6,"  (0 TO SEE BOTTOM DEPTH)",/              PPARAM     76
     . . T11,"P15=",I6,"  (0 TO SEE Z BAR IJ)"                    PPARAM     77
     . . T71,"P16=",I6,"  (0 TO SEE INITIAL SURFACE TEMPERATURE)",/   PPARAM     78
     . . T11,"P17=",I6,"  (STORE INCREMENT FOR T)"                PPARAM     79
     . . T71,"P18=",I6,"  (STORE INCREMENT FOR S)"                PPARAM     80
     . )                                                          PPARAM     81
      IMAX=IMAX + 1                                               PPARAM     82
      IMAX2=IMAX + 17                                             PPARAM     83
      PRINT 911,(PC(III),III=IMAX,IMAX2)                          PPARAM     84
  911 FORMAT (1X                                                  PPARAM     85
     . . T11,"P19=",I6,"  (PRINT INCREMENT FOR (RHO-RHOR)/RHOO)"  PPARAM     86
     . . T71,"P20=",I6,"  (PRINT INCREMENT FOR NSQ)",/            PPARAM     87
     . . T11,"P21=",I6,"  (PRINT INCREMENT FOR PI ARRAY)"         PPARAM     88
     . . T71,"P22=",I6,"  (PRINT INCREMENT FOR N-I)",/            PPARAM     89
     . . T11,"P23=",I6,"  (PRINT INCREMENT FOR DELTA T)"          PPARAM     90
     . . T71,"P24=",I6,"  (PRINT INCREMENT FOR DELTA S)",/        PPARAM     91
     . . T11,"P25=",I6,"  (PRINT INCREMENT FOR QCHECK)"           PPARAM     92
     . . T71,"P26=",I6,"  (0 TO SEE Z BAR K)",/                   PPARAM     93
     . . T11,"P27=",I6,"  (0 TO SEE Z BAR IJK)"                   PPARAM     94
     . . T71,"P28=",I6,"  (STORE INCREMENT FOR U)",/              PPARAM     95
     . . T11,"P29=",I6,"  (STORE INCREMENT FOR V)"                PPARAM     96
     . . T71,"P30=",I6,"  (PRINT INCREMENT FOR R/RC - VERT. SLICE)",/  PPARAM     97
     . . T11,"P31=",I6,"  (STORE INCREMENT FOR PI)"               PPARAM     98
     . . T71,"P32=",I6,"  (PRINT INCREMENT FOR FLUX)",/           PPARAM     99
     . . T11,"P33=",I6,"  (STORE INCREMENT FOR FLUX)"             PPARAM    100
     . . T71,"P34=",I6,"  (PRINT INCREMENT FOR FORCING: SOLAR FLUX)",/  PPARAM    101
     . . T11,"P35=",I6,"  (PRINT INCREMENT FOR FORCING: HEAT FLUX)"   PPARAM    102
```

```
      . . T71,"P36=",I6,"  (PRINT INCREMENT FOR FORCING: WIND(U))"      PPARAM     103
      . )                                                               PPARAM     104
       IMAX=IMAX2 + 1                                                   PPARAM     105
       IMAX2=IMAX + 2                                                   L0726030     3
       PRINT 912,(PC(III),III=IMAX,IMAX2)                               PPARAM     107
   912 FORMAT (1X                                                       PPARAM     108
      . . T11,"P37=",I6,"  (PRINT INCREMENT FOR FORCING: WIND(V))"      PPARAM     109
      . . T71,"P38=",I6,"  (STORE INCREMENT FOR GEOSTROPHIC VEL (U))",/ L0726030     5
      . . T11,"P39=",I6,"  (STORE INCREMENT FOR GEOSTROPHIC VEL (V))"   L0726030     6
      . . /)                                                            PPARAM     110
 *                                                                      PPARAM     111
       RETURN                                                           PPARAM     112
       END                                                              PPARAM     113

            121 ACTIVE LINE(S)           11 INACTIVE LINE(S)

 AE REGNST
```

*** SMS109 : DECK REGNST   IS EDITED FROM SPL   FILE
CREATED ON 12/20/83 AT 09:59:17   LAST UPDATED ON 01/05/84 AT 13:32:24 BY SMS VERSION 5.27
LANGUAGE: USER INFORMATION:

MODSETS PREVIOUSLY APPLIED TO SPL: L010504B

```
      SUBROUTINE REGNST(PHI,LAM,PBAR,LBAR,SAVPLO,SAVPHI,SAVLLO
     * , SAVLHI,SAVILO,SAVIHI,SAVJLO,SAVJHI,IF,JF,IFP1,JFP1,NREGON
     * , NREG)
*
*     THIS SUBROUTINE SETS THE REGIONS OVER WHICH DATA WILL BE
*       STORED FOR A SUBSEQUENT NESTED RUN.
*     T-S DATA WILL BE WRITTEN FROM SAVILO(1) TO SAVIHI(1) WHILE
*     U-S DATA WILL BE WRITTEN FROM SAVILO(2) TO SAVIHI(2)
*           (AND SIMILARLY FOR J)
*
      REAL*4 PHI(JF),LAM(IF),PBAR(JFP1),LBAR(IFP1),SAVPLO(1)
     * , SAVPHI(1),SAVLLO(1),SAVLHI(1)
      INTEGER*4 SAVILO(2,1),SAVIHI(2,1),SAVJLO(2,1),SAVJHI(2,1)
*
      DO  50 J=2,JF
   50 PBAR(J)=PHI(J) + PHI(J-1)
*
      DO  60 J=2,JF
   60 PBAR(J)=PBAR(J) * .5
*
      DO  70 I=2,IF
   70 LBAR(I)=LAM(I) + LAM(I-1)
*
      DO  80 I=2,IF
   80 LBAR(I)=LBAR(I) * .5
*
      PBAR(1    )=2. * PBAR(2 ) - PBAR(3    )
      PBAR(JFP1)=2. * PBAR(JF) - PBAR(JF-1)
      LBAR(1    )=2. * LBAR(2 ) - LBAR(3    )
      LBAR(IFP1)=2. * LBAR(IF) - LBAR(IF-1)
*
*     LOOP OVER REGIONS
*
      DO 1000 N=1,NREGON
*
      IF (SAVLLO(N).GT.SAVLHI(N) .OR. SAVPLO(N).GT.SAVPHI(N))
     *CALL STOPP(" ERROR IN SAVL OR SAVP INPUT ... REGNST$")
*
      DO 110 I=2,IF
      II=IF + 2 - I
      IF (LBAR(II) .LT. SAVLLO(N)) GO TO 115
  110 CONTINUE
      CALL STOPP(" ERROR IN SAVLLO IN REGNST$")
  115 CONTINUE
      SAVLLO(  N+NREG)=SAVLLO(N)
      SAVLLO(  N      )=LBAR(II)
      SAVILO(1,N      )=II
*
      IFM1=IF - 1
      DO 117 I=2,IFM1
```

```
      II=IFM1 + 2 - I                                          REGNST      51
      IF (LAM(II) .LT. SAVLLO(N+NREG)) GO TO 119               L0105048     3
  117 CONTINUE                                                 REGNST      53
      CALL STOPP(" ERROR IN SAVLLO IN REGNST(2)$")             REGNST      54
  119 CONTINUE                                                 REGNST      55
      SAVILO(2,N)=II                                           REGNST      56
*                                                              REGNST      57
      DO 120 I=2,IF                                            REGNST      58
      IF (LBAR(I) .GT. SAVLHI(N)) GO TO 125                    REGNST      59
  120 CONTINUE                                                 REGNST      60
      CALL STOPP(" ERROR IN SAVLHI IN REGNST$")                REGNST      61
  125 CONTINUE                                                 REGNST      62
      SAVLHI(  N+NREG)=SAVLHI(N)                               REGNST      63
      SAVLHI(  N      )=LBAR(I)                                REGNST      64
      SAVIHI(1,N      )=I                                      REGNST      65
*                                                              REGNST      66
      DO 127 I=2,IFM1                                          REGNST      67
      IF (LAM(I) .GT. SAVLHI(N+NREG)) GO TO 129                L0105048     5
  127 CONTINUE                                                 REGNST      69
      CALL STOPP(" ERROR IN SAVLHI IN REGNST(2)$")             REGNST      70
  129 CONTINUE                                                 REGNST      71
      SAVIHI(2,N)=I                                            REGNST      72
*                                                              REGNST      73
      DO 130 J=2,JF                                            REGNST      74
      JJ=JF + 2 - J                                            REGNST      75
      IF (PBAR(JJ) .LT. SAVPLO(N)) GO TO 135                   REGNST      76
  130 CONTINUE                                                 REGNST      77
      CALL STOPP(" ERROR IN VALUE OF SAVPLO IN REGNST$")       REGNST      78
  135 CONTINUE                                                 REGNST      79
      SAVPLO(  N+NREG)=SAVPLO(N)                               REGNST      80
      SAVPLO(  N      )=PBAR(JJ)                               REGNST      81
      SAVJLO(1,N      )=JJ                                     REGNST      82
*                                                              REGNST      83
      JFM1=JF - 1                                              REGNST      84
      DO 137 J=2,JFM1                                          REGNST      85
      JJ=JFM1 + 2 - J                                          REGNST      86
      IF (PHI(JJ) .LT. SAVPLO(N+NREG)) GO TO 139               L0105048     7
  137 CONTINUE                                                 REGNST      88
      CALL STOPP(" ERROR IN VALUE OF SAVPLO IN REGNST(2)$")    REGNST      89
  139 CONTINUE                                                 REGNST      90
      SAVJLO(2,N)=JJ                                           REGNST      91
*                                                              REGNST      92
      DO 140 J=2,JF                                            REGNST      93
      IF (PBAR(J) .GT. SAVPHI(N)) GO TO 145                    REGNST      94
  140 CONTINUE                                                 REGNST      95
      CALL STOPP(" ERROR IN VALUE OF SAVPHI IN REGNST$")       REGNST      96
  145 CONTINUE                                                 REGNST      97
      SAVPHI(  N+NREG)=SAVPHI(N)                               REGNST      98
      SAVPHI(  N      )=PBAR(J)                                REGNST      99
      SAVJHI(1,N      )=J                                      REGNST     100
*                                                              REGNST     101
      DO 147 J=2,JFM1                                          REGNST     102
      IF (PHI(J) .GT. SAVPHI(N+NREG)) GO TO 149                L0105048     9
  147 CONTINUE                                                 REGNST     104
      CALL STOPP(" ERROR IN VALUE OF SAVPHI IN REGNST(2)$")    REGNST     105
  149 CONTINUE                                                 REGNST     106
      SAVJHI(2,N)=J                                            REGNST     107
*                                                              REGNST     108
```

```
 1000 CONTINUE
*
      CONI=57.29578
      PRINT 901
      DO 1010 N=1,NREGON
      P1=SAVLLO(N+NREG) * CONI
      P2=SAVLLO(N     ) * CONI
      P3=SAVLHI(N+NREG) * CONI
      P4=SAVLHI(N     ) * CONI
      P5=SAVPLO(N+NREG) * CONI
      P6=SAVPLO(N     ) * CONI
      P7=SAVPHI(N+NREG) * CONI
      P8=SAVPHI(N     ) * CONI
      PRINT 902,N,P1,P2,SAVILO(1,N)
     * *                P3,P4,SAVIHI(1,N)
     * *                P5,P6,SAVJLO(1,N)
     * *                P7,P8,SAVJHI(1,N)
 1010 CONTINUE
*
  901 FORMAT (///,T30,"*** SAVE REGION SET UP ***",//
     * *"           N",T13 ,"LLO IN",T24 ,"LLO USE",T35 ,"ILO"
     * *                T43 ,"LHI IN",T54 ,"LHI USE",T65 ,"IHI"
     * *                T73 ,"PLO IN",T84 ,"PLO USE",T95 ,"JLO"
     * *                T103,"PHI IN",T114,"PHI USE",T125,"JHI")
  902 FORMAT (4X,I3,2X,4(1X,1PE9.2,1X,1PE12.5,2X,I3,2X))
*
*
      RETURN
      END
```

137 ACTIVE LINE(S)   4 INACTIVE LINE(S)

IE ANDND

```
SMS VERSION 5.27    LISTING OF DECK    R H O M D                                              01/12/84          PAGE 1

*** SMS109 : DECK RHOMD    IS EDITED FROM SPL      FILE
CREATED ON 12/20/83 AT 09:59:10          LAST UPDATED ON 12/27/83 AT 17:58:47 BY SMS VERSION 5.27
LANGUAGE:                USER INFORMATION:

MODSETS PREVIOUSLY APPLIED TO SPL: L122783A

              SUBROUTINE RHOMD(A,IA,JA,KA,LA,I1,I2,J1,J2,K1,K2,L1,L2                   RHOMD        1
             * , LU,READ)                                                              RHOMD        2
         *                                                                             RHOMD        3
              DIMENSION A(IA,JA,KA,LA)                                                 RHOMD        4
              LOGICAL READ                                                             RHOMD        5
         *                                                                             RHOMD        6
              IF (.NOT. READ) GO TO 500                                                RHOMD        7
         *                                                                             RHOMD        8
              READ (LU) (((A(I,J1  ,K,L),I=I1,I2),K=K1,K2),L=L1,L2)                    L122783A     3
              READ (LU) (((A(I,J1+1,K,L),I=I1,I2),K=K1,K2),L=L1,L2)                    L122783A     4
              READ (LU) (((A(I,J2-1,K,L),I=I1,I2),K=K1,K2),L=L1,L2)                    L122783A     5
              READ (LU) (((A(I,J2  ,K,L),I=I1,I2),K=K1,K2),L=L1,L2)                    L122783A     6
         *                                                                             L122783A     7
              READ (LU) (((A(I1  ,J,K,L),J=J1,J2),K=K1,K2),L=L1,L2)                    L122783A     8
              READ (LU) (((A(I1+1,J,K,L),J=J1,J2),K=K1,K2),L=L1,L2)                    L122783A     9
              READ (LU) (((A(I2-1,J,K,L),J=J1,J2),K=K1,K2),L=L1,L2)                    L122783A    10
              READ (LU) (((A(I2  ,J,K,L),J=J1,J2),K=K1,K2),L=L1,L2)                    L122783A    11
         *                                                                             L122783A    12
              RETURN                                                                   RHOMD       10
         *                                                                             RHOMD       11
          500 CONTINUE                                                                 RHOMD       12
              WRITE (LU) (((A(I,J1  ,K,L),I=I1,I2),K=K1,K2),L=L1,L2)                   RHOMD       13
              WRITE (LU) (((A(I,J1+1,K,L),I=I1,I2),K=K1,K2),L=L1,L2)                   RHOMD       14
              WRITE (LU) (((A(I,J2-1,K,L),I=I1,I2),K=K1,K2),L=L1,L2)                   RHOMD       15
              WRITE (LU) (((A(I,J2  ,K,L),I=I1,I2),K=K1,K2),L=L1,L2)                   RHOMD       16
         *                                                                             RHOMD       17
              WRITE (LU) (((A(I1  ,J,K,L),J=J1,J2),K=K1,K2),L=L1,L2)                   RHOMD       18
              WRITE (LU) (((A(I1+1,J,K,L),J=J1,J2),K=K1,K2),L=L1,L2)                   RHOMD       19
              WRITE (LU) (((A(I2-1,J,K,L),J=J1,J2),K=K1,K2),L=L1,L2)                   RHOMD       20
              WRITE (LU) (((A(I2  ,J,K,L),J=J1,J2),K=K1,K2),L=L1,L2)                   RHOMD       21
         *                                                                             RHOMD       22
              RETURN                                                                   RHOMD       23
              END                                                                      RHOMD       24

             33 ACTIVE LINE(S)          1 INACTIVE LINE(S)

 OE SAVREG
```

```
*** SMS109 : DECK SAVREG   IS EDITED FROM SPL      FILE
CREATED ON 12/20/83 AT 09:59:19          LAST UPDATED ON 01/05/84 AT 13:50:45 BY SMS VERSION 5.27
LANGUAGE:               USER INFORMATION:

MODSETS PREVIOUSLY APPLIED TO SPL: L010504A  L010504C

      SUBROUTINE SAVREG(ISTP)                                          SAVREG      1
*                                                                      SAVREG      2
*     THIS ROUTINE WRITES OUT DATA FOR USE AS INITIAL AND              SAVREG      3
*       BOUNDARY CONDITIONS FOR A NESTED RUN                           SAVREG      4
*                                                                      SAVREG      5
*CALL ALLMAC                                                           SAVREG      6
*                                                                      SAVREG      7
      IF (NREGON .EQ. 0) RETURN                                        SAVREG      8
*                                                                      SAVREG      9
      DO 1000 N=1,NREGON                                               SAVREG     10
*                                                                      SAVREG     11
      IF (ISTP .LT. SAVSTR(N)) GO TO 1000                              SAVREG     12
*                                                                      SAVREG     13
      I1 =SAVILO(1,N)                                                  SAVREG     14
      I2 =SAVIHI(1,N)                                                  SAVREG     15
      J1 =SAVJLO(1,N)                                                  SAVREG     16
      J2 =SAVJHI(1,N)                                                  SAVREG     17
      II1=SAVILO(2,N)                                                  SAVREG     18
      II2=SAVIHI(2,N)                                                  SAVREG     19
      JJ1=SAVJLO(2,N)                                                  SAVREG     20
      JJ2=SAVJHI(2,N)                                                  SAVREG     21
*                                                                      SAVREG     22
      LU=70 + N                                                        SAVREG     23
*                                                                      SAVREG     24
      IF (ISTP .NE. SAVSTR(N)) GO TO 500                               SAVREG     25
*                                                                      SAVREG     26
*     SAVE INITIAL DATA                                                SAVREG     27
*                                                                      SAVREG     28
      WRITE (LU) ((((T4S(I,J,K,L),I=I1,I2),J=J1,J2),K=1,KF),L=1,2)     SAVREG     29
      WRITE (LU) ((ZSUR(I,J),I=I1,I2),J=J1,J2)                         SAVREG     30
*                                                                      SAVREG     31
      WRITE (LU) ((((U4V(I,J,K,L),I=II1,II2),J=JJ1,JJ2),K=1,KF)        SAVREG     32
     . , L=1,2)                                                        SAVREG     33
*                                                                      SAVREG     34
*                                                                      SAVREG     35
  500 CONTINUE                                                         SAVREG     36
      IF (MOD(ISTP,SAVINC(N)) .NE. 0) GO TO 1000                       SAVREG     37
*                                                                      SAVREG     38
      CALL RWBND(T4S,IFP1,JFP1,KF,2,I1,I2,J1,J2,1,KF,1,2               L010504A    3
     . , LU,.FALSE.)                                                   L010504A    4
*                                                                      L010504A    5
      CALL RWBND(U4V,IF,JF,KF,2,II1,II2,JJ1,JJ2,1,KF,1,2               L010504A    6
     . , LU,.FALSE.)                                                   L010504A    7
*                                                                      L010504A    8
      CALL RWBND(PI,IF,JF,KF,1,I1,I2,J1,J2,1,KF,1,1                    L010504A    9
     . , LU,.FALSE.)                                                   L010504A   10
*                                                                      L010504A   11
*     CALL RWBND(ZSUR,IF,JF,1,1,I1,I2,J1,J2,1,1,1,1                    L010504A   12
*    . , LU,.FALSE.)                                                   L010504A   13
*                                                                      SAVREG     50
```

```
SNS VERSION 5.27    LISTING OF DECK    S A V R E G                                    01/12/84         PAGE 2

       50   *      WRITE (LU) REFBN                                                      LO10504C     3
       51   *      WRITE (LU) (REFBA(I),I=1,REFBN)                                       LO10504C     4
       52   *                                                                            SAVREG      53
       53   *                                                                            SAVREG      54
       54    1000 CONTINUE                                                               SAVREG      55
       55   *                                                                            SAVREG      56
       56   *                                                                            SAVREG      57
       57          RETURN                                                                SAVREG      58
       58          END                                                                   SAVREG      59

              59 ACTIVE LINE(S)           13 INACTIVE LINE(S)

 DE SETUPA
```

```
SMS VERSION 5.27    LISTING OF DECK    S E T U P A                                  01/12/84          PAGE 1

*** SMS109 : DECK SETUPA   IS EDITED FROM SPL        FILE
CREATED ON 07/26/79 AT 16:23:26          LAST UPDATED ON 12/05/83 AT 13:40:40 BY SMS VERSION 5.27
LANGUAGE:               USER INFO  ATION:

MODSETS PREVIOUSLY APPLIED TO SPL: FIX211A   FIX211B   FIX215A   FIX215B   FIX221A   FIX225A   FIX226C   FIX226D   F255A
        F255C     F256A     F296B     F298A     F299A     F299B     F303A     F303B     F305A     DMPGPI    UPUIND1   F313A
        DMPGP12   F313B     SLFADJ1   F319A     F319C     PISWEEP4  PISWEEP6  F1206D    F1207A    F1210B    F1212B    F0104A
        F0107A    F0131A    F0204A    F0205A    F0211C    F0211D    F0212C    F0213A    F0214D    F0219A    F0221A    F0222A
        F0225C    F0226D    F0306A    F0310A    F0314B    F0318A    F0319C    F0321C    F0326A    F0326B    F0331A    F0331C
        F0401A    F0403A    F0407C    F0400B    F0409A    F0409C    F0410A    F0410B    F0411A    F0416B    F0417C    F0418A
        F0422C    F0422D    F0430A    F0430D    F0513A    F0515A    F0515B    F0515C    F0530A    F0627B    F0700B    F0723A
        F0723B    F0725C    F0807A    F0812A    F0825B    F1009F    F1020A    F1020E    F1021A    F1023C    F1023E    F0106601A
        F0106601D L0715B1A  L001101E  L001301A  L032302A  L040202A  L040702A  L041302A  L092302A  L072603B  L072703A  L001603A
        L082903C  L120583A

              SUBROUTINE SETUPA(OTH,ICASE)                                                      L001101E      3
        *                                                                                       SETUP         2
        $CALL ALLMAC                                                                            SETUP         3
        *                                                                                       SETUP         4
        $CALL ASTUPC                                                                            F0515A        6
        *                                                                                       SETUP        12
        $CALL AAMAML                                                                            L032302A      3
        *                                                                                       L032302A      4
              EQUIVALENCE (IICASE,VARIS)                                                        L032302A      5
        *                                                                                       SETUP        22
              NOIN=.FALSE.                                                                      SETUP        23
              EMPTY=.FALSE.                                                                     SETUP        24
              ILABEL=0                                                                          SETUP        25
              CALL POSITN(1,2)                                                                  F0211C        3
              BACKSPACE 1                                                                       F0211C        4
              BACKSPACE 1                                                                       F0211C        5
              IRUN=0                                                                            F0212C        8
              READ (1,991,END=4) IRUN                                                           F0212C       10
            4 CONTINUE                                                                          F0212C       11
              IRUN=IRUN + 1                                                                     F0211C        7
              CALL S$DATE(DATE,1)                                                               SETUP        26
              CALL S$TIME(TIME,1)                                                               SETUP        27
              PRINT 939,IRUN,(DATE(L),L=2,3),(TIME2(L),L=1,2)                                   F0211C        9
          939 FORMAT (1H1,T5,"RUN ",I3,T105,2A4," (",A2,":",A2,")")                             F0211C       10
            5 CONTINUE                                                                          SETUP        30
              READ (5,930,END=8) MARK,LABEL                                                     SETUP        31
              IF (MARK(1).EQ.AMPER) GO TO 6                                                     SETUP        32
              ILABEL=ILABEL + 1                                                                 SETUP        33
              PRINT 935,MARK,LABEL                                                              SETUP        34
              IF (RITLOG) GO TO 5                                                               F0212C       13
              IF (MARK(1).NE.STAR) GO TO 5                                                      F0212C       14
              WRITE(1,991)IRUN,(DATE(L),L=2,3),(TIME2(L),L=1,2),MARK,LABEL                      F0212C       15
          991 FORMAT (I3,1X,2A4,"(",A2,":",A2,") ",2A2,19A4)                                    F0211C       16
              RITLOG=.TRUE.                                                                     F0212C       17
              GO TO 5                                                                           SETUP        35
            6 BACKSPACE 5                                                                       SETUP        36
              GO TO 9                                                                           SETUP        37
            8 CONTINUE                                                                          SETUP        38
              NOIN=.TRUE.                                                                       SETUP        39
              EMPTY=.TRUE.                                                                      SETUP        40
            9 CONTINUE                                                                          SETUP        41
```

```
      IF (.NOT.RITLOG)                                           F0212C   19
     .WRITE(1,991)IRUN,(DATE(L),L=2,3),(TIME2(L),L=1,2)          F0211C   17
  930 FORMAT  (2A2,19A4)                                         SETUP    42
  935 FORMAT (26X,2A2,19A4)                                      SETUP    43
  940 FORMAT (2A2)                                               SETUP    44
*                                                                SETUP    45
*     SET DEFAULT VALUES                                         SETUP    46
*                                                                SETUP    47
      AINV=1.5695E-7                                             SETUP    48
      CG=0.1E0                                                   SETUP    49
      RC=.23E0                                                   SETUP    50
      LSQ=10.E0                                                  SETUP    51
      LSQ=LSQ * LSQ                                              SETUP    52
      RHO0=1.028104SE0                                           SETUP    53
      G=9.8E0                                                    SETUP    54
      GSURF=9.8E0                                                F0107A    6
      BETAN=1.5E0                                                SETUP    55
      BETAV=1.3E0                                                SETUP    56
      CV=1.E-4                                                   SETUP    57
      ZV=1.E2                                                    SETUP    58
      DELT=20.E0                                                 SETUP    59
      CD=3.E-3                                                   SETUP    60
      SURFA=0.                                                   F0430A    5
      AARO=1.21E-3                                               SETUP    62
      FIXZS=.TRUE.                                               SETUP    63
      FIXINT=.TRUE.                                              SETUP    64
      ADJZS=1.                                                   SETUP    65
      ADJINT=1.                                                  SETUP    66
      FIXTS=.TRUE.                                               SETUP    67
      VFIXTS=.TRUE.                                              F1023E    5
      THTSLA=.5E0                                                FIX211A   7
      THTSPA=.5E0                                                FIX211A   8
      THTSSA=.5E0                                                FIX211A   9
      THTSLD=1.E0                                                FIX211B   4
      THTSPD=1.E0                                                FIX211B   5
      THTSSD=1.E0                                                FIX211B   6
      FIXUV=.TRUE.                                               SETUP    71
      VFIXUV=.TRUE.                                              SETUP    72
      THUVLA=.5E0                                                FIX211A  14
      THUVPA=.5E0                                                FIX211A  15
      THUVSA=.5E0                                                FIX211A  16
      THUVLD=4.E0                                                FIX211B   8
      THUVPD=4.E0                                                FIX211B   9
      THUVSD=.5E0                                                FIX211A  19
      THUVF=.5                                                   F0225C    3
      TMAX=27.                                                   SETUP    77
      TINHN=0                                                    F0326A    9
      SALT=1.E0                                                  SETUP    79
      UMAX=10.E0                                                 SETUP    80
      DT=600.E0                                                  SETUP    85
      HOUR=0.                                                    SETUP    86
      NSTEP=0                                                    F0225C    5
      OMEGA=7.292166E-5                                          SETUP    88
      F0=1.458433E-5                                             SETUP    89
      CASE=1                                                     SETUP    90
      SMOOTH=.FALSE.                                             F290A     9
      KS=1.5                                                     F290A    10
      ZSM=2000.                                                  F290A    11
```

```
      NWATTN=500.                                             DMPGPI2    6
      DTMULT=1.                                               DMPGPI     6
      DIMAX=0.                                                DMPGPI     7
      DMPGPI=.FALSE.                                          DMPGPI     8
      AWPUNO=0.                                               F0409C     3
      ASMZSM=+100.00                                          F0409C     4
      OLDPI=.TRUE.                                            F1020A     6
      DMPPII=0.2                                              F1212B     3
      DMPPII=.TRUE.                                           F1207A     6
      NGAPPT=6                                                F0104A     6
      NZSITR=15                                               F0306A     6
      ZSFERR=0.                                               F0204A     6
      ZSFCHK=0                                                F0205A     6
      CJCOEF=.5
*          DECS=-9936722222;   INCS=-9007375661;               F0306A    10
*          DEC_=-9936723041;   INC_=-9007377281;               F0306A    11
      OJCOEF=-9007373566                                      F0306A    12
      ZSINI=0                                                 F0214B     7
      CZSINI=1.E-4                                            F0326A    11
      ZSINMN=22                                               F0326A    12
      ZSFLCR=1.E20                                            F0306A    14
      ZSFA=100.                                               F0226B     7
      VVFDIV=.TRUE.                                           F0306A    16
      REFOR=10                                                F0314B     8
      REFBF=100000                                            F0319C     6
      DELTTS=.FALSE.                                          F0400B     8
      NPPAGE=2                                                F0400B     9
      REFBR=2.300                                             F0417C     6
      QRMULT=1.                                               F0430B     6
      TIMREP=200.                                             F0513A     6
      IFNITR=10                                               F0627B     6
      FIA=100.                                                F0627B     7
      FIERR=.1                                                F0627B     8
      KVR=5.E4                                                F0007A     6
      BOTCHK=200.                                             F0700B     5
      FICHK=0                                                 F0700B     6
      UMAXM=.1                                                F0725C     6
      FORCIN=.FALSE.                                          F1020A     8
      FTATCH=.TRUE.                                           F1021A     6
      CONDEM=.FALSE.                                          L001301A   5
      CKM=0.                                                  L001603A   3
      ZCKM=1.E6                                               L001603A   4
      DO 12 I=1,4                                             L001603A   5
   12 FLMINC(I)=0.                                            L001603A   6
      DINFL=100.                                              L001603A   7
      DOOTBL=1.E3                                             L001603A   8
      NREGON=0                                                L120503A   3
*                                                             FIX215A    6
*     GET BOTTOM DATA                                         FIX215A    7
*                                                             FIX215A    8
      READ (2,910) LMIN,LMAX,PMIN,PMAX,IFC,JFC,XLSPAC,XPSPAC  L092302A   3
  910 FORMAT (4F7.1,2I7,1P2E10.3)                             L092302A   4
      IF (IF.NE.IFC  .OR.  JF.NE.JFC)                         FIX215A   11
     .CALL STOPP(' ERROR IN SETUP, WRONG VALUE OF IF OR JF RETURNED FROM  FIX215A  12
     . FILE 26')                                              FIX215A   13
      XLMIN=- LMIN                                            FIX215A   14
      PRINT 919,LMIN,LMAX,PMIN,PMAX                           F303A      7
  919 FORMAT (///,' THE DOMAIN RUNS FROM ',F8.3               F0411A     3
```

```
     . , " DEGREES EAST TO ",F8.3," DEGREES EAST, AND "                 F0411A       4
     . , F8.3," DEGREES NORTH TO ",F8.3," DEGREES NORTH.")              F0411A       5
      DO 15 I=1,IF                                                      FII2150      3
   15 READ (2,920)(ZBOT(I,J),J=1,JF)                                    F0331A       5
C                                                                       L0029O3C     3
C     ZBOT WILL EVENTUALLY (IN SETUPB) BE BARRED WRT I AND J, SO        L0029O3C     4
C       WE NEED TO SAVE ZBOT AT THE ORIGINAL LOCATIONS (NM-)            L0029O3C     5
C                                                                       L0029O3C     6
      DO 16 J=1,JF                                                      L0029O3C     7
      DO 16 I=1,IF                                                      L0029O3C     8
   16 ZBOTNM(I,J)=ZBOT(I,J)                                             L0029O3C     9
  920 FORMAT (10F7.1)                                                   F1009F       3
      DO 10 I=1,3                                                       F299A        5
      DO 10 J=1,40                                                      F299A        6
   10 IPC(I,J)=1                                                        F299A        7
      DO 20 II=1,60                                                     F299A        8
   20 PC(II)=-1                                                         SETUP       99
C                                                                       SETUP      100
C                                                                       SETUP      101
      IF (MARK(2).NE.ON     .OR.   NOIN) GO TO 30                       SETUP      102
      READ (5,ONE)                                                      SETUP      103
      READ(5,940,END=25) MARK                                           SETUP      104
      BACKSPACE 5                                                       SETUP      105
      GO TO 30                                                          SETUP      106
   25 NOIN=.TRUE.                                                       SETUP      107
   30 CONTINUE                                                          SETUP      108
      IF (MARK(2).NE.TW   .OR.   NOIN) GO TO 50                         F299B        6
      READ (5,TWO)                                                      F299B        7
      READ (5,940,END=45) MARK                                          F299B        8
      BACKSPACE 5                                                       F299B        9
      GO TO 50                                                          F299B       10
   45 NOIN=.TRUE.                                                       F299B       11
   50 CONTINUE                                                          F299B       12
      IF (HOUR.NE.0) DT=HOUR * 3600.                                    SETUP      109
      IF (PC(1).LT.0) PC(1)=NSTEP                                       SETUP      117
      IF (PC(6).LT.0) PC(6)=NSTEP                                       SETUP      118
      IF (PC(9).LT.0) PC(9)=NSTEP                                       SETUP      119
      IF (PC(12).LE.0) PC(12)=0                                         F255A        3
      IF (PC(13).LT.0) PC(13)=0                                         SETUP      120
      IF (PC(14).LT.0) PC(14)=0                                         SETUP      121
      IF (PC(17).LT.0) PC(17)=0                                         F255A        5
      IF (PC(18).LT.0) PC(18)=0                                         F255A        6
      IF (PC(25).LT.0) PC(25)=1                                         F05150       3
      IF (PC(26).LT.0) PC(26)=1                                         F05150       4
      IF (PC(27).LT.0) PC(27)=1                                         F05150       5
      IF (PC(28).LT.0) PC(28)=0                                         F05150       6
      IF (PC(29).LT.0) PC(29)=0                                         F05150       7
      IF (PC(31).LT.0) PC(31)=0                                         F0012A       3
      IF (PC(33).LT.0) PC(33)=0                                         F00250       3
      IF (PC(38).EQ.0 .AND. PC(39).EQ.0) GO TO 60                       L0727O3A     3
      IF (.NOT.OLDPI) CALL STOPP                                        L0727O3A     4
     .(" GEOSTROPHIC VELOCITY ONLY IF  OLDPI=T .. SETUPA6")             L0727O3A     5
      IF (     DMPGPI) CALL STOPP                                       L0727O3A     6
     .(" GEOSTROPHIC VELOCITY ONLY IF DMPGPI=F .. SETUPA6")             L0727O3A     7
   60 CONTINUE                                                          L0727O3A     8
C                                                                       F0310A       5
C     IF BAROTROPIC CASE                                                F0310A       6
C                                                                       F0310A       7
```

```
          IF (G.EQ.0.)                              F0310A      8
         .CALL S4JSLV(1,1,1,1,CB,I)                 F0310A      9
          LASTEP=999                                L041302A    3
          DO 40 II=1,60                             F299A      10
       40 IF (PC(II) .LT. 0) PC(II)=LASTEP          L041302A    5
          CALL PPARAM(DT)                           L040702A    3
   *                                                F0515A      8
          DTX=DT                                    F0515A      9
          ICASE=CASE                                L081101E    5
          IICASE=CASE                               L032302A    9
   *                                                F0515A     10
   *      DO SETUP PART B NEXT                      F0515A     11
   *                                                F0515A     12
          RETURN                                    F0515A     13
          END                                       F0515A     14
```

230 ACTIVE LINE(S)   296 INACTIVE LINE(S)

SE SETUPB

```
*** SAS189 : DECK SETUPB   IS EDITED FROM SPL      FILE
CREATED ON 05/15/80 AT 08:40:13          LAST UPDATED ON 12/20/83 AT 09:59:21 BY SAS VERSION 5.27
LANGUAGE:            USER INFORMATION:

MODSETS PREVIOUSLY APPLIED TO SPL: F05158    F0515D    F0720F    F1020A    F1023C    F1223B    F010681A  F010701A  L071501A
     L072001B  L092382A  L033103A  L001603A  L001603D  L120503A  L120503B  L120603B  L122003A

            SUBROUTINE SETUPB(DT)                                                                F1020A      3
      *                                                                                          SETUPB      2
      $CALL ALLMAC                                                                               SETUPB      3
      $CALL ASTUPC                                                                               SETUPB      4
      *                                                                                          F05158      3
            DIMENSION DUMK(IF,JF,KF),DUMIJK(IF,JF,KF)                                            F05158      4
            REAL*4 LBAR(IFP1),PBAR(JFP1)                                                         L120503A    3
            EQUIVALENCE (DUMK,KV),(DUMIJK,KM)                                                    F05158      5
            EQUIVALENCE (LBAR,D2),(PBAR,D1)                                                      L120503A    5
      *                                                                                          F1020A      5
            NAMELIST/MINE/PHI,LAM,F,UAGVA,SURF,QR,ZBOT                                           F1020A      6
           . , ZSUR,FS,REFDA                                                                     F1020A      7
      *                                                                                          SETUPB      5
            ITEST=0                                                                              SETUPB      6
            DO 56 J=1,40                                                                         SETUPB      7
            DO 52 I=1,3                                                                          SETUPB      8
            IF (IPC(I,J).NE.1) GO TO 54                                                          SETUPB      9
         52 CONTINUE                                                                             SETUPB     10
            GO TO 56                                                                             SETUPB     11
         54 ITEST=ITEST + 1                                                                      SETUPB     12
            IF (ITEST.EQ.1) PRINT 900                                                            SETUPB     13
            PRINT 901,J,(IPC(II,J),II=1,3)                                                       SETUPB     14
         56 CONTINUE                                                                             SETUPB     15
        900 FORMAT (1H0,5X,"PRINT INCREMENT CONTROLS",/)                                         SETUPB     16
        901 FORMAT (5X, "IPC(1," ,I2, ")=(" ,I3, "," ,I2, "," ,I2,")")                           F0720F      3
      *                                                                                          SETUPB     18
      *                                                                                          SETUPB     19
            IF (PC(14).EQ.0)                                                                     SETUPB     20
           .CALL PRT(-1," BOTTOM DEPTH  ",ZBOT,IF,JF,1.1,.FALSE.                                 SETUPB     21
           . , IPC(1,14))                                                                        SETUPB     22
      *                                                                                          SETUPB     23
      * *** HERE, IF REQUESTED, THE INPUT BOTTOM TOPOGRAPHY CAN BE SMOOTHED **                   SETUPB     24
      *                                                                                          SETUPB     25
            IF (.NOT.SMOOTH) GO TO 58                                                            SETUPB     26
            CALL ZSMUTH(KS,ZSM,ZBOT,D3,D4,IF,JF                                                  SETUPB     27
           . , ZTOTS,ZTOTE,ZMAXX,NTT,DTT,TTAU)                                                   SETUPB     28
            PRINT 950,ZTOTS                                                                      SETUPB     29
        950 FORMAT (1H0," THE SUM OF THE INTERIOR DEPTHS BEFORE IS",1PE10.3)                     SETUPB     30
            PRINT 951,KS,ZSM,ZMAXX,NTT,DTT,TTAU                                                  SETUPB     31
        951 FORMAT (1H0," KS= ",F3.1,5X,"ZSMOOTH= ",F7.1,5X                                      SETUPB     32
           . ,"ZMAX= ",F7.1,5X,"NT= ",I4,5X,"DT= ",1PE9.2                                        SETUPB     33
           . , 5X,"TAU= ",1PE9.2)                                                                SETUPB     34
            PRINT 952,ZTOTE                                                                      SETUPB     35
        952 FORMAT(1H0," THE SUM OF THE INTERIOR DEPTHS AFTER IS ",1PE10.3,/)                    SETUPB     36
            CALL PRT(-1," SMOOTHED DEPTHS",ZBOT,IF,JF,1.1,.FALSE.                                SETUPB     37
           . , IPC(1,14))                                                                        SETUPB     38
         58 CONTINUE                                                                             SETUPB     39
      *                                                                                          SETUPB     40
      * *** SET UP MESH ***********************************************************             SETUPB     41
```

```
*                                                                   SETUPB     42
*   REMEMBER THE LATERAL BOUNDARIES LIE BETWEEN MESH LINES (IE.     SETUPB     43
*     THERE IS ALWAYS A ROW OF MESH POINTS OUTSIDE THE BOUNDARY).   SETUPB     44
*     HENCE, THE FACTORS OF 1.5 (RATHER THAN 1) BELOW.              SETUPB     45
*                                                                   SETUPB     46
    CON=1.745329E-2                                                 SETUPB     49
    PRINT 960                                                       L092302A    4
    CALL MESHST(PHI,JF,PMIN,PMAX,XPSPAC,0)                          L092302A    5
    DO 60 J=1,JF                                                    SETUPB     51
 60 PHI(J)=PHI(J) * CON                                             SETUPB     53
    PRINT 961                                                       L092302A    8
    CALL MESHST(LAM,IF,LMIN,LMAX,XLSPAC,0)                          L092302A    9
    DO 70 I=1,IF                                                    SETUPB     55
 70 LAM(I)=LAM(I) * CON                                             SETUPB     57
    DEL=PHI(2) - PHI(1)                                             L092302A   12
    DO 75 J=3,JF                                                    L092302A   13
    DELCHK=PHI(J) - PHI(J-1)                                        L092302A   14
    IF (DELCHK .LT. DEL) DEL=DELCHK                                 L092302A   15
 75 CONTINUE                                                        L092302A   16
*                                                                   L120503B    3
    DO 76 N=1,NREGON                                                L120503B    4
    SAVPLO(N)=SAVPLO(N) * CON                                       L120503B    5
    SAVPHI(N)=SAVPHI(N) * CON                                       L120503B    6
    SAVLLO(N)=SAVLLO(N) * CON                                       L120503B    7
    SAVLHI(N)=SAVLHI(N) * CON                                       L120503B    8
 76 CONTINUE                                                        L120503B    9
    DEL=CON * DEL / AINV                                            L092302A   17
*                                                                   SETUPB     58
    DO 80 I=1,IF                                                    SETUPB     59
    DO 80 J=1,JF                                                    SETUPB     60
 80 F(I,J)=2.E0 * OMEGA * SIN(PHI(J))                               SETUPB     61
*                                                                   L072001B    3
*   SET UP Z BAR K AND Z BAR IJK IN RV-RN ARRAY SPACE.  THEY        L072001B    4
*     MAY BE USED IN CASE ROUTINES                                  L072001B    5
*                                                                   L072001B    6
    CALL DEPINI                                                     L072001B    7
*                                                                   L120503A    7
*   SET UP THE SUBREGIONS, IF NECESSARY                             L120503A    8
*                                                                   L120503A    9
    IF (NREGON .EQ. 0) GO TO 78                                     L120503A   10
    CALL REGNST(PHI,LAM,D1,D2,SAVPLO,SAVPHI,SAVLLO,SAVLHI           L120503A   11
   . , SAVILO,SAVIHI,SAVJLO,SAVJHI,IF,JF,IFP1,JFP1,NREGON,NREG)     L120503A   12
*                                                                   L120503A   13
    DO 77 N=1,NREGON                                                L120503A   14
*                                                                   L120503A   15
    LU=N + 70                                                       L120503A   16
    KFN=KF                                                          L120503A   17
    I1 =SAVILO(1,N)                                                 L122003A    4
    I2 =SAVIHI(1,N)                                                 L122003A    5
    J1 =SAVJLO(1,N)                                                 L122003A    6
    J2 =SAVJHI(1,N)                                                 L122003A    7
*                                                                   L122003A    8
    II1=SAVILO(2,N)                                                 L122003A    9
    II2=SAVIHI(2,N)                                                 L122003A   10
    JJ1=SAVJLO(2,N)                                                 L122003A   11
    JJ2=SAVJHI(2,N)                                                 L122003A   12
*                                                                   L122003A   13
    NI1= I2 -  I1 + 1                                               L122003A   14
```

```
      NJ1= J2 - J1 + 1                                          L122003A   15
      NI2=II2 - II1 + 1                                         L122003A   16
      NJ2=JJ2 - JJ1 + 1                                         L122003A   17
*                                                               L120503A   27
      WRITE (LU) IRUN,N                                         L120503A   28
      WRITE (LU) SAVSTR(N),SAVINC(N),SAVPLO(N),SAVPHI(N)        L120503A   29
     . ,        SAVLLO(N),SAVLHI(N)                             L120503A   30
      WRITE (LU) NI1,NJ1,NI2,NJ2,KFM,DT                         L122003A   19
      WRITE (LU) (LBAR(I),I= I1, I2)                            L122003A   21
     . ,        (PBAR(J),J= J1, J2)                             L122003A   22
     . ,        ( LAM(I),I=II1,II2)                             L122003A   23
     . ,        ( PHI(J),J=JJ1,JJ2)                             L122003A   24
      WRITE (LU) FIXZS ,ADDZS ,ZSFA,ZSFERR,NZSITR               L120503A   36
     . ,        FIXINT,ADDINT,FIA ,FIERR ,IFNITR                L120503A   37
      WRITE (LU) ((ZBOT(I,J),I=II1,II2)                         L122003A   26
     . ,                    J=JJ1,JJ2)                          L122003A   27
      IF (I1.LT.2 .OR. I2.GT.IF  .OR.  J1.LT.2 .OR. J2.GT.JF)   L120503A   40
     .CALL STOPP(" SUBREGION NOT IN INTERIOR ... SETUPB$")      L120503A   41
      WRITE (LU) (((DUMIJK(I,J,K),I= I1, I2),J= J1, J2),K=2,KFM1)  L122003A   29
      WRITE (LU) (((DUMK  (I,J,K),I=II1,II2),J=JJ1,JJ2),K=2,KFM1)  L122003A   30
*                                                               L120503A   44
   77 CONTINUE                                                  L120503A   45
C                                                               L120503A   46
   78 CONTINUE                                                  L120503A   47
*                                                               SETUPB     62
* *** CALL SPECIFIC INPUT ROUTINE ************************************  SETUPB     63
*                                                               SETUPB     64
*                1   2   3   4   5   6   7   8                  L120503A   49
      GO TO (100, 90,120,130,140,145,150,152),CASE              L120503A   50
   90 CONTINUE                                                  L071501A    5
      CALL STOPP("BAD VALUE OF CASE$")                          F1020A     10
  100 CALL CASE1(TINMAN,TMAX,SALT)                              F1023C      3
      GO TO 160                                                 F1020A     12
  120 CALL CASE3(DT)                                            F1020A     15
      GO TO 160                                                 F1020A     16
  130 CALL CASE4(TMAX,DT)                                       F010703A    3
      GO TO 160                                                 F1020A     18
  140 CALL CASE5                                                F1020A     19
      GO TO 160                                                 L033103A    6
  145 CALL CASE6                                                L033103A    7
      GO TO 160                                                 L033103A    8
  150 CALL CASE7                                                L033103A    9
      GO TO 160                                                 L120503A   52
  152 CALL CASE8                                                L120603B    3
  160 CONTINUE                                                  F1020A     22
*                                                               F05158      7
*     OPTIONAL PRINTS    Z BAR K    AND    Z BAR IJK            L072001B    9
*                                                               F05158      9
      IF (PC(26).EQ.0)                                          F05158     10
     .CALL PRT(0,"Z BAR K$",DUMK,IF,JF,KFM1,1,.TRUE.,IPC(1,26)) F05158      4
      IF (PC(27).EQ.0)                                          F05158     12
     .CALL PRT(0,"Z BAR IJK$",DUMIJK,IF,JF,KFM1,1,.TRUE.,IPC(1,27))  F05158   6
*                                                               SETUPB     71
*                                                               SETUPB     72
* *** SET UP SOME OF THE ARRAYS REQUIRED IN THE CALCULATION ***********  SETUPB     73
*                                                               SETUPB     74
*     FIND ARRAY USED IN CALCULATION OF DEPTHS AT SIGMA LEVELS  SETUPB     75
*                                                               SETUPB     76
```

```
      DELTI=1. / DELT
      DO 185 J=1,JF
      DO 185 I=1,IF
  185 ZSAV(I,J)=ZBOT(I,J) * DELTI   +   1.
*
*     FIND Z BAR IJ
*
      DO 190 I=2,IF
      DO 190 J=1,JF
  190 D3(I,J)=ZBOT(I,J) + ZBOT(I-1,J)
*
      DO 195 I=1,IF
      DO 195 J=1,JF
  195 DUM(I,J)=0.
*
      DO 200 I=2,IF
      DO 200 J=2,JF
  200 DUM(I,J)=(D3(I,J) + D3(I,J-1)) * .25E0
*
      DO 205 J=2,JF
  205 DUM(1,J)=DUM(3,J)
*
      DO 210 I=1,IF
  210 DUM(I,1)=DUM(I,3)
*
*     FIND QUANTITY THAT IS REQUIRED IN ROUTINE IGRPIE
*
      DELTI=1. / DELT
      DO 215 J=1,JF
      DO 215 I=1,IF
  215 D3(I,J)=DUM(I,J) * DELTI
*
      DO 220 J=1,JF
      DO 220 I=1,IF
  220 D3(I,J)=D3(I,J) + 1.
*
      DO 225 J=1,JF
      DO 225 I=1,IF
  225 D3(I,J)=ALOG(D3(I,J))
*
      DO 230 J=1,JF
      DO 230 I=1,IF
  230 ZDPIF(I,J)=1. / D3(I,J)
*
*     FIND QUANTITY THAT IS REQUIRED IN ROUTINE UVDOTB
*
*
      CONI=(-1. / HEATTH**4) * .125
      DO 235 J=2,JFM1
      DO 235 I=2,IFM1
      GRADZB(I,J)=AMAX1(DUM(I,J),DUM(I+1,J),DUM(I,J+1),DUM(I+1,J+1))
      ZWORK      =AMIN1(DUM(I,J),DUM(I+1,J),DUM(I,J+1),DUM(I+1 J+1))
      GRADZB(I,J)=GRADZB(I,J) - ZWORK
  235 GRADZB(I,J)=GRADZB(I,J) * CONI
*
*     BOTTOM TOPOGRAPHY IS SYMMETRIC
*
      DO 240 J=2,JFM1
```

```
      GRADZB(1,J)=GRADZB(2,J)
  240 GRADZB(IF,J)=GRADZB(IFM1,J)
*
      DO 245 I=1,IF
      GRADZB(I,1)=GRADZB(I,2)
  245 GRADZB(I,JF)=GRADZB(I,JFM1)
*
*     FIND THE MODIFIED DEPTH VALUES THAT WILL BE USED IN PISURF
*       FOR THE ZS STABILIZATION.
*
      IF (.NOT.DNPGPI) GO TO 257
      CONB=-0.
      DO 250 J=1,JF
      DO 250 I=1,IF
  250 D10(I,J)=GRADZB(I,J) * CONB
*
*     NEGATIVE 1/3 TO INVERT THE EXPRESSION
*
      CON3=-1. / 3.
      CALL RAISE(D9,D10,CON3,1,IF,1,JF,IFP1,JFP1,IFP1,JFP1)
*
      DO 255 J=1,JF
      DO 255 I=1,IF
  255 D10(I,J)=AMIN1(ZBOT(I,J),D9(I,J))
*
      CALL SMZSH(D10,IFP1,JFP1,ASMZSH,ZSH,IF,JF,D9,IFP1,JFP1)
      GO TO 258
  257 CALL SMZSH(ZBOT,IF,JF,ASMZSH,ZSH,IF,JF,D9,IFP1,JFP1)
  258 CONTINUE
      IF (ASMZSH.LE.0.)
     .CALL PRT(-1,'SMOOTHED ZS (FOR FIXING) DEPTHS',ZSH,IF,JF,1,2
     . , .FALSE.,0)
*
*     THE DEPTH (Z-B BAR IJ) IS NEEDED IN THE SURFACE FIXING
*
      DO 260 J=2,JF
      DO 260 I=1,IF
  260 D3(I,J)=ZBOT(I,J) + ZBOT(I,J-1)
*
      DO 263 J=2,JF
      DO 263 I=2,IF
  263 ZBOT(I,J)=(D3(I-1,J) + D3(I,J)) * .25
*
      DO 265 J=1,JF
      DO 265 I=1,IF
      FS(I,J)=0.
  265 CONTINUE
*
*     INITIALIZE Z SURFACE
*
      I=IABS(ZSINI)
      CALL ZSSET(ZSUR,IF,JF,I,CZSINI,ZSINMN)
      IF (ZSINI.GT.0)
     .CALL ZSPRT(-1,'INITIALIZED SURFACE VALUES IN METERS:',ZSUR
     . , IF,JF,.FALSE.,9,D4)
*
*     READ IN COEFFICIENTS TO BE USED IN REFERENCE BUOYANCY
*
```

```
      IF (REFBN.EQ.0) GO TO 267                                        SETUPB  193
      READ (4) ITEST                                                   SETUPB  194
      IF (ITEST.NE.REFBN) GO TO 266                                    SETUPB  195
      READ (4) (REFBA(I),I=1,REFBN)                                    SETUPB  196
      GO TO 267                                                        SETUPB  197
  266 CONTINUE                                                         SETUPB  198
      DO 269 I=1,REFBN                                                 SETUPB  199
  269 REFBA(I)=0.D0                                                    SETUPB  200
      PRINT 901                                                        SETUPB  201
  901 FORMAT (//,2(" *****",/)                                         SETUPB  202
     . ," ***** WARNING:  REFBN DOES NOT MATCH VALUE FROM INPUT"       SETUPB  203
     . ," DISK FILE.  ALL COEFFICIENTS WILL BE SET TO ZERO.",/         SETUPB  204
     . ,2(" *****",/))                                                 SETUPB  205
  267 CONTINUE                                                         SETUPB  206
C                                                                      SETUPB  207
C     ZERO OUT RHO SO THAT MAX VALUES CAN BE PICKED OUT BY QCHECK      SETUPB  208
C                                                                      SETUPB  209
      DO 260 K=2,KFR1                                                  SETUPB  210
      DO 260 J=1,JF                                                    SETUPB  211
      DO 260 I=1,IF                                                    SETUPB  212
  260 RHO(I,J,K)=0.                                                    SETUPB  213
C                                                                      SETUPB  214
C ***                                                          ***     SETUPB  215
C                                                                      SETUPB  216
      IF (PC(15).NE.0) GO TO 270                                       SETUPB  217
      CALL PRT(-1,"     Z BAR IJ     ",DUM,IF,JF,1,1,.FALSE.           SETUPB  218
     . , IPC(1,15))                                                    SETUPB  219
  270 CONTINUE                                                         SETUPB  220
      IF (PC(16) .NE. 0) GO TO 280                                     SETUPB  221
      DELTI=1. / DELT                                                  SETUPB  222
      DO 275 J=2,JF                                                    SETUPB  223
      DO 275 I=2,IF                                                    SETUPB  224
  275 DUM(I,J)=DUM(I,J) * DELTI                                        SETUPB  225
      CALL TKSURF(TSS,TS,D3,D4,IFP1,JFP1,KF,DUM,IF,JF,0)               SETUPB  226
      CALL PRT(-1," INIT SURF TEMP ",TS,IFP1,JFP1,1,1,.FALSE.          SETUPB  227
     . , IPC(1,16))                                                    SETUPB  228
  280 CONTINUE                                                         SETUPB  229
C                                                                      SETUPB  230
C                                                                      F1020A   24
C     SURF ARRAY CAN BE SET IN ROUTINE FORCE                           F1020A   25
C                                                                      F1020A   26
      IF (FORCIN) GO TO 290                                            F1020A   27
C                                                                      SETUPB  231
C     SET SURFACE BOUNDARY CONDITION ARRAY.  WHAT IT REPRESENTS        SETUPB  232
C       DEPENDS ON THE VALUE OF SURFA:                                 SETUPB  233
C           SURFA=0          SURFACE FLUX OF HEAT SET TO ZERO          SETUPB  234
C           SURFA > 1.E10    SURFACE TEMPERATURE IMPOSED               SETUPB  235
C           SURFA OTHERWISE  MIXED CONDITION                           SETUPB  236
C     (NOTE THAT TSURF INITIAL IS JUST T(2))                           SETUPB  237
C                                                                      SETUPB  238
      IF (SURFA.NE.0.) GO TO 285                                       SETUPB  239
      DO 283 J=2,JF                                                    SETUPB  240
      DO 283 I=2,IF                                                    SETUPB  241
  283 SURF(I,J)=0.                                                     SETUPB  242
      GO TO 290                                                        SETUPB  243
C                                                                      SETUPB  244
  285 CONTINUE                                                         SETUPB  245
      DO 289 J=2,JF                                                    SETUPB  246
```

```
      DO 289 I=2,IF                                      SETUPB   247
  289 SURF(I,J)=TGS(I,J,2,1)                             SETUPB   248
      IF (SURFA.GE.1.E10) GO TO 290                      SETUPB   249
*                                                        SETUPB   250
      SURFAI=1. / SURFA                                  SETUPB   251
      DO 292 J=2,JF                                      SETUPB   252
      DO 292 I=2,IF                                      SETUPB   253
  292 DS(I,J)=QR(I,J) * SURFAI                           SETUPB   254
*                                                        SETUPB   255
      DO 295 J=2,JF                                      SETUPB   256
      DO 295 I=2,IF                                      SETUPB   257
  295 SURF(I,J)=SURF(I,J) - DS(I,J)                      SETUPB   258
*                                                        SETUPB   259
  290 CONTINUE                                           SETUPB   260
*                                                        SETUPB   261
      IF (MARK(2).NE.NI   .OR.   NOIN) GO TO 310         SETUPB   262
      READ (5,NINE)                                      SETUPB   263
      READ (5,940,END=310) MARK                          SETUPB   264
      BACKSPACE 5                                        SETUPB   265
      GO TO 310                                          SETUPB   266
  305 NOIN=.TRUE.                                        SETUPB   267
  310 CONTINUE                                           SETUPB   268
      IF (PC(11).EQ.0) WRITE (6,NINE)                    SETUPB   269
*                                                        SETUPB   270
*     SET PISUR BASED ON INITIAL ZSUR                    SETUPB   271
*                                                        SETUPB   272
      X=-GSURF                                           SETUPB   273
      DO 312 J=2,JF                                      SETUPB   274
      DO 312 I=2,IF                                      SETUPB   275
  312 PISUR(I,J)=ZSUR(I,J) * X                           SETUPB   276
*                                                        SETUPB   277
      PRINT 946                                          SETUPB   278
  946 FORMAT (//,' THE INPUT DATA APPEARS AS FOLLOWS:',/)  SETUPB   279
      REWIND 5                                           SETUPB   280
      IF (.NOT.EMPTY) GO TO 315                          SETUPB   281
      PRINT 947                                          SETUPB   282
      PRINT 947                                          SETUPB   283
  947 FORMAT (//,' THERE IS NO INPUT DATA FOR THIS RUN')   SETUPB   284
      RETURN                                             SETUPB   285
  315 CONTINUE                                           SETUPB   286
      IF (ILABEL.EQ.0) GO TO 330                         SETUPB   287
      DO 300 LL=1,ILABEL                                 SETUPB   288
  300 READ 930,MARK,LABEL                                SETUPB   289
  330 CONTINUE                                           SETUPB   290
  320 READ (5,945,END=350) LABEL                         SETUPB   291
      PRINT 945,LABEL                                    SETUPB   292
      GO TO 320                                          SETUPB   293
  350 CONTINUE                                           SETUPB   294
  945 FORMAT (19A4)                                      SETUPB   295
  930 FORMAT (2A2,19A4)                                  SETUPB   296
  940 FORMAT (2A2)                                       SETUPB   297
  960 FORMAT (20X,'LATITUDE MESH:')                      L092302A  19
  961 FORMAT (20X,'LONGITUDE MESH:')                     L092302A  20
*                                                        SETUPB   298
      RETURN                                             SETUPB   299
      END                                                SETUPB   300
```

SAS VERSION 5.27   LISTING OF DECK   S E T U P B   01/12/84   PAGE 8

395 ACTIVE LINE(S)   54 INACTIVE LINE(S)

NE UUDOTB

```
*** SMS189 : DECK UVBND     IS EDITED FROM SPL       FILE
CREATED ON 06/18/79 AT 13:46:32          LAST UPDATED ON 02/13/84 AT 13:18:33 BY SMS VERSION 5.27
LANGUAGE: FORTRAN      USER INFORMATION:

MODSETS PREVIOUSLY APPLIED TO SPL: FIX200A   FIX201A   FIX201F   F292A    PISWEEP1  F0428B   F0530A   F1024D   F1027C
     F011281A  F011281B  L032382A  L081683A  L081883B  L082983B  L010484A  L020384A  L021384B

          SUBROUTINE UVBND                                                    UVBND       1
      *                                                                       UVBND       2
      *   THIS ROUTINE SETS THE EXTRA UV POINTS WHICH LIE OUTSIDE THE         UVBND       3
      *     BOUNDARY.                                                         UVBND       4
      *                                                                       UVBND       5
      $CALL ALLMAC                                                            UVBND       6
      *                                                                       UVBND       7
          DIMENSION D12(IFP1,JFP1,2)                                          F292A       3
          EQUIVALENCE (D12,D5)                                                UVBND       9
        . , (CASE,VAR15)                                                      L032382A    4
          EQUIVALENCE (ISTP,VAR14)                                            L081683A    3
          INTEGER CASE                                                        L010484A    3
      *                                                                       UVBND      10
      *   DO THE BOTTOM ACCORDING TO EQN. 3.34 (INTERIOR LATERAL POINTS).     UVBND      11
      *                                                                       UVBND      12
      *   IF THE VERTICAL FIXING WAS DONE, THEN THE QUANTITY THAT             FIX200A     7
      *     WE CALCULATE IN DBTM WOULD HAVE BEEN PLACED IN D9.                FIX201A     6
      *   IT IS BASED ON THE OLD UV'S, NOT THE UPDATED ONES.                  F0428B      3
      *     THEREFORE, DBTM MUST BE CALLED AGAIN.                             F0428B      4
      *                                                                       FIX200A    17
        200 CALL DBTM(IFM1,JFM1,KFM1,IF,JF,KF,IFP1,JFP1                       F292A       5
        . , U$V,F,NSQ(1,1,KF),FO,DELKZ,D1,D3,D9)                              F0530A      3
      *                                                                       UVBND      64
          DO 300 L=1,2                                                        UVBND      65
          DO 300 I=2,IFM1                                                     UVBND      66
          DO 300 J=2,JFM1                                                     UVBND      67
        300 U$V(I,J,KF,L)=U$V(I,J,KFM1,L) $ D9(I,J)                           FIX201A    12
      *                                                                       UVBND      69
      *   NOW DO THE TOP.  WE SET TAU-SIGMA=TAU-Z, WE USE THE SURFACE         UVBND      70
      *     CONDITIONS (EQN. 3.31) FOR TAU-SIGMA, AND WE SOLVE EQNS. 3.19     UVBND      71
      *     FOR THE DELTA-K U$V.                                              UVBND      72
      *                                                                       UVBND      73
      *   WE DON'T HAVE DZBRKT AT THE CORRECT K LEVEL, SO WE MUST             UVBND      74
      *     RECALCULATE IT.                                                   UVBND      75
      *     NOTE:  Z3 - Z1= Z3 - (2 * Z2 - Z3)  BY EQN. 3.25A.                UVBND      76
      *                                                                       UVBND      77
      *                                  SINCE THE WIND STRESS IS AT          F1027C      3
      *                                  THE INCORRECT TIME, APPLY A          F1027C      4
      *                                  SIMPLY CONDITION                     F1027C      5
      *                                                                       F1027C      6
          DO 450 L=1,2                                                        UVBND     103
          DO 450 I=2,IFM1                                                     UVBND     104
          DO 450 J=2,JFM1                                                     UVBND     105
        450 U$V(I,J,1,L)=U$V(I,J,2,L)                                         F1027C      8
      *                                                                       UVBND     107
      *   NOW DO THE LATERAL BOUNDARIES.                                      UVBND     108
      *                                                                       UVBND     109
      *   APPLY BOUNDARY CONDITIONS FOR ...                                   UVBND     110
      *     (NO FLUX LATERAL BOUNDARIES)                                      UVBND     111
```

```
      $                                                              UVBND      112
            DO 5 K=1,KF                                              UVBND      113
      $                                                              UVBND      114
      $$$$U                                                          UVBND      115
            L=1                                                      UVBND      116
      $                                                              F011281A     6
            IF (CASE.EQ.4) GO TO 12                                  F011281A     7
            DO 10 J=2,JFM1                                           UVBND      117
            USV(1,J,K,L)= - USV(2,J,K,L)                             UVBND      118
            USV(IF,J,K,L)= - USV(IFM1,J,K,L)                         UVBND      119
         10 CONTINUE                                                 UVBND      120
            GO TO 18                                                 F011281A     9
         12 CONTINUE                                                 F011281A    10
            DO 15 J=2,JFM1                                           F011281A    11
            USV(1,J,K,L)=USV(2,J,K,L)                                F011281A    12
            USV(IF,J,K,L)=USV(IFM1,J,K,L)                            F011281A    13
         15 CONTINUE                                                 F011281A    14
      $                                                              F011281A    15
         18 CONTINUE                                                 F011281A    16
      $                                                              UVBND      121
            DO 20 I=1,IF                                             UVBND      122
            USV(I,1,K,L)=USV(I,2,K,L)                                UVBND      123
            USV(I,JF,K,L)=USV(I,JFM1,K,L)                            UVBND      124
         20 CONTINUE                                                 UVBND      125
      $                                                              UVBND      126
      $$$$V                                                          UVBND      127
            L=2                                                      UVBND      128
            DO 30 J=2,JFM1                                           UVBND      129
            USV(1,J,K,L)=USV(2,J,K,L)                                UVBND      130
            USV(IF,J,K,L)=USV(IFM1,J,K,L)                            UVBND      131
         30 CONTINUE                                                 UVBND      132
      $                                                              UVBND      133
            DO 40 I=1,IF                                             UVBND      134
            USV(I,1,K,L)= - USV(I,2,K,L)                             UVBND      135
            USV(I,JF,K,L)= - USV(I,JFM1,K,L)                         UVBND      136
         40 CONTINUE                                                 UVBND      137
      $                                                              UVBND      138
          5 CONTINUE                                                 UVBND      139
      $                                                              L081683A     5
      $     IMPLEMENT IMPOSED SIDE FLUXES                            L081683A     6
      $                                                              L081683A     7
            K=0                                                      L081683A     8
            CALL ZSET3(K+2)                                          L081683A     9
            CALL KDEP2(K+1,1)                                        L081683A    10
      $                                                              L081683A    11
            DO 1000 K=1,KFM1                                         L081683A    12
      $                                                              L081683A    13
            CALL ZSET3(K+2)                                          L081683A    14
            CALL KDEP2(K+1,1)                                        L081683A    15
      $                                                              L082983B     3
      $     WE ARE WORKING AT K=K+1, SO RANGE IS REALLY FROM 2 -> KF L082983B     4
      $                                                              L082983B     5
      $     USE ONLY INTERNAL POINTS:                                L082983B     6
      $                                                              L082983B     7
            IF (K+1 .EQ. KF) GO TO 1000                              L082983B     8
      $                                                              L081683A    16
      $     WEST EDGE                                                L081683A    17
      $                                                              L081683A    18
```

```
          IF (FLUXIN(1) .EQ. 0.) GO TO 190                      L081683A   19
     *                                                          L020384A    3
     *    I AT EDGE SHOULD REALLY BE 1.5, BUT ALL QUANTITIES ARE   L020384A    4
     *      SYMMETRIX, SO WE CAN USE I=2                        L020384A    5
     *                                                          L020384A    6
          I=2                                                   L020384A    7
     *                                                          L021384B    3
          CON=1. / DINFL(1)                                     L021384B    4
          DO 105 J=1,JF                                         L021384B    5
          D1(I,J)=ZBARK(I,J,1) * CON                            L021384B    6
     105  D2(I,J)=ZBOTHH(I,J) - ZBARK(I,J,1)                    L021384B    7
     *                                                          L021384B    8
          CON=1. / DBOTBL(1)                                    L021384B    9
          DO 110 J=1,JF                                         L021384B   10
          D1(I,J)=D1(I,J) + 1.                                  L021384B   11
     110  D2(I,J)=D2(I,J) * CON                                 L021384B   12
     *                                                          L021384B   13
          IF (DINFLS(1) .GT. 1.E10) GO TO 119                   L021384B   14
     *                                                          L021384B   15
          DO 112 J=1,JF                                         L021384B   16
          D5(I,J)=DINFLS(1) - ZBARK(I,J,2)                      L021384B   17
     112  D6(I,J)=DINFLS(1) + ZBARK(I,J,2)                      L021384B   18
     *                                                          L021384B   19
          DO 114 J=1,JF                                         L021384B   20
     114  D6(I,J)=D6(I,J) * D1(I,J)                             L021384B   21
     *                                                          L021384B   22
          DO 116 J=1,JF                                         L021384B   23
     116  D1(I,J)=D5(I,J) / D6(I,J)                             L021384B   24
          GO TO 122                                             L021384B   25
     *                                                          L021384B   26
     119  CONTINUE                                              L021384B   27
          DO 120 J=1,JF                                         L021384B   28
     120  D1(I,J)=1. / D1(I,J)                                  L021384B   29
     122  CONTINUE                                              L021384B   30
     *                                                          L020384A   21
          DO 130 J=1,JF                                         L020384A   22
          D3(I,J)=D1(I,J) * D2(I,J)                             L020384A   23
      130 D4(I,J)=D1(I,J) + D2(I,J)                             L020384A   24
     *                                                          L020384A   25
          DO 140 J=1,JF                                         L020384A   26
      140 D3(I,J)=D3(I,J) / D4(I,J)                             L020384A   27
     *                                                          L020384A   28
          CON=2. * FLUXIN(1) * (1.-EXP(-FLOAT(ISTP)/30.))       L020384A   29
          DO 150 J=1,JF                                         L020384A   30
      150 D3(I,J)=D3(I,J) * CON                                 L020384A   31
     *                                                          L020384A   32
          DO 160 J=1,JF                                         L020384A   33
      160 U$V(1,J,K,1)=D3(I,J) - U$V(2,J,K,1)                   L020384A   34
     *                                                          L020384A   35
      190 CONTINUE                                              L081683A   20
     *                                                          L081683A   21
     *    EAST EDGE                                             L081683A   22
     *                                                          L081683A   23
          IF (FLUXIN(2) .EQ. 0.) GO TO 290                      L081683A   24
     *                                                          L081683A   25
     *    I AT EDGE SHOULD REALLY BE IF-.5, BUT ALL QUANTITIES ARE   L081683A   26
     *      SYMMETRIX, SO WE CAN USE I=IF-1                     L081683A   27
     *                                                          L081683A   28
```

```
          I=IFM1                                              L081683A   29
*                                                             L021384B   32
          CON=1. / DINFL(2)                                   L021384B   33
          DO 205 J=1,JF                                       L021384B   34
          D1(I,J)=ZBARK(I,J,2) * CON                          L021384B   35
    205   D2(I,J)=ZBOTHH(I,J) - ZBARK(I,J,2)                  L021384B   36
*                                                             L021384B   37
          CON=1. / DBOTBL(2)                                  L021384B   38
          DO 210 J=1,JF                                       L021384B   39
          D1(I,J)=D1(I,J) + 1.                                L021384B   40
    210   D2(I,J)=D2(I,J) * CON                               L021384B   41
*                                                             L021384B   42
          IF (DINFLS(2) .GT. 1.E10) GO TO 219                 L021384B   43
*                                                             L021384B   44
          DO 212 J=1,JF                                       L021384B   45
          D5(I,J)=DINFLS(2) - ZBARK(I,J,2)                    L021384B   46
    212   D6(I,J)=DINFLS(2) + ZBARK(I,J,2)                    L021384B   47
*                                                             L021384B   48
          DO 214 J=1,JF                                       L021384B   49
    214   D6(I,J)=D6(I,J) * D1(I,J)                           L021384B   50
*                                                             L021384B   51
          DO 216 J=1,JF                                       L021384B   52
    216   D1(I,J)=D5(I,J) / D6(I,J)                           L021384B   53
          GO TO 222                                           L021384B   54
*                                                             L021384B   55
    219   CONTINUE                                            L021384B   56
          DO 220 J=1,JF                                       L021384B   57
    220   D1(I,J)=1. / D1(I,J)                                L021384B   58
    222   CONTINUE                                            L021384B   59
*                                                             L081683A   43
          DO 230 J=1,JF                                       L081683A   44
          D3(I,J)=D1(I,J) * D2(I,J)                           L081683A   45
      230 D4(I,J)=D1(I,J) + D2(I,J)                           L081683A   46
*                                                             L081683A   47
          DO 240 J=1,JF                                       L081683A   48
      240 D3(I,J)=D3(I,J) / D4(I,J)                           L081683A   49
*                                                             L081683A   50
          CON=2. * FLUXIN(2) * (1.-EXP(-FLOAT(ISTP)/30.))     L081683A   51
          DO 250 J=1,JF                                       L081683A   52
      250 D3(I,J)=D3(I,J) * CON                               L081683A   53
*                                                             L081683A   54
          DO 260 J=1,JF                                       L081683A   55
      260 U*V(IF,J,K,1)=D3(I,J) - U*V(IFM1,J,K,1)             L081683A   56
*                                                             L081683A   57
      290 CONTINUE                                            L081683A   58
*                                                             L081683A   59
*         SOUTH EDGE                                          L081683A   60
*                                                             L081683A   61
          IF (FLUXIN(3) .EQ. 0.) GO TO 390                    L081683A   62
*                                                             L081683A   63
*         J AT EDGE SHOULD REALLY BE 1.5, BUT ALL QUANTITIES ARE  L081683A   64
*           SYMMETRIX, SO WE CAN USE J = 2                    L081683A   65
*                                                             L081683A   66
          J=2                                                 L081683A   67
*                                                             L021384B   61
          CON=1. / DINFL(3)                                   L021384B   62
          DO 305 I=1,IF                                       L021384B   63
          D1(I,J)=ZBARK(I,J,2) * CON                          L021384B   64
```

```
 305  D2(I,J)=ZBOTHM(I,J) - ZBARK(I,J,2)                          L021384B   65
 *                                                                 L021384B   66
      CON=1. / DBOTBL(3)                                           L021384B   67
      DO 310 I=1,IF                                                L021384B   68
      D1(I,J)=D1(I,J) + 1.                                         L021384B   69
 310  D2(I,J)=D2(I,J) * CON                                        L021384B   70
 *                                                                 L021384B   71
      IF (DINFLS(3) .GT. 1.E10) GO TO 319                          L021384B   72
 *                                                                 L021384B   73
      DO 312 I=1,IF                                                L021384B   74
      D5(I,J)=DINFLS(3) - ZBARK(I,J,2)                             L021384B   75
 312  D6(I,J)=DINFLS(3) + ZBARK(I,J,2)                             L021384B   76
 *                                                                 L021384B   77
      DO 314 I=1,IF                                                L021384B   78
 314  D6(I,J)=D6(I,J) * D1(I,J)                                    L021384B   79
 *                                                                 L021384B   80
      DO 316 I=1,IF                                                L021384B   81
 316  D1(I,J)=D5(I,J) / D6(I,J)                                    L021384B   82
      GO TO 322                                                    L021384B   83
 *                                                                 L021384B   84
 319  CONTINUE                                                     L021384B   85
      DO 320 I=1,IF                                                L021384B   86
 320  D1(I,J)=1. / D1(I,J)                                         L021384B   87
 322  CONTINUE                                                     L021384B   88
 *                                                                 L081683A   81
      DO 330 I=1,IF                                                L081683A   82
      D3(I,J)=D1(I,J) * D2(I,J)                                    L081683A   83
  330 D4(I,J)=D1(I,J) + D2(I,J)                                    L081683A   84
 *                                                                 L081683A   85
      DO 340 I=1,IF                                                L081683A   86
  340 D3(I,J)=D3(I,J) / D4(I,J)                                    L081683A   87
 *                                                                 L081683A   88
      CON=2. * FLUXIN(3) * (1.-EXP(-FLOAT(ISTP)/30.))              L081683A   89
      DO 350 I=1,IF                                                L081683A   90
  350 D3(I,J)=D3(I,J) * CON                                        L081683A   91
 *                                                                 L081683A   92
      DO 360 I=1,IF                                                L081683A   93
  360 U*V(I,1,K,2)=D3(I,J) - U*V(I,2,K,2)                          L081683A   94
 *                                                                 L081683A   95
  390 CONTINUE                                                     L081683A   96
 *                                                                 L081683A   97
 *    NORTH EDGE                                                   L081683A   98
 *                                                                 L081683A   99
      IF (FLUXIN(4) .EQ. 0.) GO TO 490                             L081683A  100
 *                                                                 L020384A   37
 *    J AT EDGE SHOULD REALLY BE JF-.5, BUT ALL QUANTITIES ARE     L020384A   38
 *      SYMMETRIX, SO WE CAN USE J = JF-1                          L020384A   39
 *                                                                 L020384A   40
      J=JFM1                                                       L020384A   41
 *                                                                 L021384B   90
      CON=1. / DINFL(4)                                            L021384B   91
      DO 405 I=1,IF                                                L021384B   92
      D1(I,J)=ZBARK(I,J,2) * CON                                   L021384B   93
 405  D2(I,J)=ZBOTHM(I,J) - ZBARK(I,J,2)                           L021384B   94
 *                                                                 L021384B   95
      CON=1. / DBOTBL(4)                                           L021384B   96
      DO 410 I=1,IF                                                L021384B   97
      D1(I,J)=D1(I,J) + 1.                                         L021384B   98
```

```
410  D2(I,J)=D2(I,J) * CON                                  L021384B    99
*                                                           L021384B   100
     IF (DINFLS(4) .GT. 1.E10) GO TO 419                    L021384B   101
*                                                           L021384B   102
     DO 412 I=1,IF                                          L021384B   103
     D5(I,J)=DINFLS(4) - ZBARK(I,J,2)                       L021384B   104
412  D6(I,J)=DINFLS(4) + ZBARK(I,J,2)                       L021384B   105
*                                                           L021384B   106
     DO 414 I=1,IF                                          L021384B   107
414  D6(I,J)=D6(I,J) * D1(I,J)                              L021384B   108
*                                                           L021384B   109
     DO 416 I=1,IF                                          L021384B   110
416  D1(I,J)=D5(I,J) / D6(I,J)                              L021384B   111
     GO TO 422                                              L021384B   112
*                                                           L021384B   113
419  CONTINUE                                               L021384B   114
     DO 420 I=1,IF                                          L021384B   115
420  D1(I,J)=1. / D1(I,J)                                   L021384B   116
422  CONTINUE                                               L021384B   117
*                                                           L020384A    55
     DO 430 I=1,IF                                          L020384A    56
     D3(I,J)=D1(I,J) * D2(I,J)                              L020384A    57
  430 D4(I,J)=D1(I,J) + D2(I,J)                             L020384A    58
*                                                           L020384A    59
     DO 440 I=1,IF                                          L020384A    60
  440 D3(I,J)=D3(I,J) / D4(I,J)                             L020384A    61
*                                                           L020384A    62
     CON=2. * FLUXIN(4) * (1.-EXP(-FLOAT(ISTP)/30.))        L020384A    63
     DO 455 I=1,IF                                          L020384A    64
  455 D3(I,J)=D3(I,J) * CON                                 L020384A    65
*                                                           L020384A    66
     DO 460 I=1,IF                                          L020384A    67
  460 U*V(I,JF,K,2)=D3(I,J) - U*V(I,JFM1,K,2)               L020384A    68
*                                                           L020384A    69
  490 CONTINUE                                              L081683A   101
*                                                           L081683A   102
 1000 CONTINUE                                              L081683A   103
*                                                           L081683A   104
*                                                           UVBND      140
*                                                           UVBND      141
     RETURN                                                 UVBND      142
     END                                                    UVBND      143
```

323 ACTIVE LINE(S)   161 INACTIVE LINE(S)

```
*** SMS109 : DECK UVDOTB   IS EDITED FROM SPL      FILE
CREATED ON 07/13/79 AT 10:43:39          LAST UPDATED ON 07/27/83 AT 10:25:28 BY SMS VERSION 5.27
LANGUAGE:             USER INFORMATION:

MODSETS PREVIOUSLY APPLIED TO SPL: XFONT190  F292A    DMPGPI    F306A    DMPGPI2   REFIX317  PISWEEP6  L0726838  L0727038

          SUBROUTINE UVDOTB(K)                                          UVDOTB      1
      *                                                                 UVDOTB      2
      *   THIS ROUTINE IS PART "B" OF A TWO PART ROUTINE.  PART "A"     UVDOTB      3
      *   MUST BE CALLED FIRST.                                         UVDOTB      4
      *                                                                 UVDOTB      5
      $CALL ALLMAC                                                      UVDOTB      6
      *                                                                 UVDOTB      7
          DIMENSION D12(IFP1,JFP1,2),D22(IFP1,JFP1,2),D32(IFP1,JFP1,2)  F292A       3
        . , VOLI(IFP1,JFP1),ANSR(IFM2,JFM2,2,KF),GRADPI(IF,JF,2)        PISWEEP6    3
        . , UVVG(IF,JF,2)                                               L0727038    3
      *                                                                 UVDOTB     10
          EQUIVALENCE (D12,D1),(D22,D3),(D32,D5)                        UVDOTB     11
        . , (VOLI,D9),(ANS,ANSR),(GRADPI,PISAV1)                        PISWEEP6    5
        . , (D7,UVVG)                                                   L0727038    5
      *                                                                 L0727038    6
      *   NOTE: EQUIVALENCE FOR UVVG OVERLAPS USE OF D7 AND D8          L0727038    7
      *     => CAN'T USE DMPGPI=T                                       L0727038    8
      *                                                                 L0727038    9
      *                                                                 UVDOTB     13
      *                                                                 UVDOTB     16
      *>>>>>>>>>>                                                       UVDOTB     17
               M=1                                                      UVDOTB     18
      *>>>>>>>>>>                                                       UVDOTB     19
      *                                                                 UVDOTB     20
          IF (K.EQ.1) GO TO 8005                                        UVDOTB     21
      *                                                                 UVDOTB     22
      *   O TH    TERM                                                  UVDOTB     23
      *                                                                 UVDOTB     24
      *MMM                                                              UVDOTB     25
          DO 810 I=1,IF                                                 UVDOTB     26
          DO 810 J=2,JF                                                 UVDOTB     27
      810 D7(I,J)=ALFP(I,J) + ALFP(I,J-1)                               UVDOTB     28
      *MMM                                                              UVDOTB     29
          DO 820 I=2,IFM1                                               UVDOTB     30
          DO 820 J=2,JF                                                 UVDOTB     31
      820 D8(I,J)=TPL(I+1,J,M) + TPL(I,J,M)                             UVDOTB     32
      *                                                                 UVDOTB     33
          DO 830 I=2,IFM1                                               UVDOTB     34
          DO 830 J=2,JF                                                 UVDOTB     35
      830 D8(I,J)=D8(I,J) * .5E0                                        UVDOTB     36
      *                                                                 UVDOTB     37
          DO 840 I=2,IFM1                                               UVDOTB     38
          DO 840 J=2,JF                                                 UVDOTB     39
      840 D12(I,J,1)=D7(I,J) * D8(I,J)                                  UVDOTB     40
      *                                                                 UVDOTB     41
          DO 850 I=2,IFM1                                               UVDOTB     42
          DO 850 J=2,JF                                                 UVDOTB     43
      850 D12(I,J,2)=D7(I,J) * TPP(I,J,M)                               UVDOTB     44
      *                                                                 UVDOTB     45
          DO 860 L=1,2                                                  UVDOTB     46
```

```
      DO 860 I=2,IFM1                                          UVDOTB    47
      DO 860 J=2,JFM1                                          UVDOTB    48
  860 D22(I,J,L)=D12(I,J+1,L) - D12(I,J,L)                     UVDOTB    49
*                                                              UVDOTB    50
      DO 870 L=1,2                                             UVDOTB    51
      DO 870 I=2,IFM1                                          UVDOTB    52
      DO 870 J=2,JFM1                                          UVDOTB    53
  870 D22(I,J,L)=D22(I,J,L) * .5E0                             UVDOTB    54
*                                                              UVDOTB    55
      DO 880 L=1,2                                             UVDOTB    56
      DO 880 I=2,IFM1                                          UVDOTB    57
      DO 880 J=2,JFM1                                          UVDOTB    58
  880 ANSR(I-1,J-1,L,K)=ANSR(I-1,J-1,L,K) + D22(I,J,L)         UVDOTB    59
*                                                              UVDOTB    60
*     9 TH     TERM                                            UVDOTB    61
*                                                              UVDOTB    62
*     CALCULATE AHEAD                                          UVDOTB    63
*                                                              UVDOTB    64
 8005 CONTINUE                                                 UVDOTB    65
*                                                              UVDOTB    66
      DO 930 I=2,IFM1                                          UVDOTB    67
      DO 930 J=2,JFM1                                          UVDOTB    68
  930 D12(I,J,1)=ALFZIJ(I,J) * TSL02(I,J)                      UVDOTB    69
*                                                              UVDOTB    70
      DO 940 I=2,IFM1                                          UVDOTB    71
      DO 940 J=2,JFM1                                          UVDOTB    72
  940 D12(I,J,2)=ALFZIJ(I,J) * TSP02(I,J)                      UVDOTB    73
      IF (K.EQ.1) GO TO 8006                                   UVDOTB    74
*                                                              UVDOTB    75
      DO 950 L=1,2                                             UVDOTB    76
      DO 950 I=2,IFM1                                          UVDOTB    77
      DO 950 J=2,JFM1                                          UVDOTB    78
  950 D22(I,J,L)=D12(I,J,L) - UVSAV3(I,J,L)                    UVDOTB    79
*                                                              UVDOTB    80
 8006 CONTINUE                                                 UVDOTB    81
*                                                              UVDOTB    82
      DO 960 L=1,2                                             UVDOTB    83
      DO 960 I=2,IFM1                                          UVDOTB    84
      DO 960 J=2,JFM1                                          UVDOTB    85
  960 UVSAV3(I,J,L)=D12(I,J,L)                                 UVDOTB    86
      IF (K.EQ.1) RETURN                                       UVDOTB    87
*                                                              UVDOTB    88
      DO 970 L=1,2                                             UVDOTB    89
      DO 970 I=2,IFM1                                          UVDOTB    90
      DO 970 J=2,JFM1                                          UVDOTB    91
  970 ANSR(I-1,J-1,L,K)=ANSR(I-1,J-1,L,K) + D22(I,J,L)         UVDOTB    92
*                                                              UVDOTB    93
*     FIND THE VOLUME (LEAVING THE APPROPRIATE QUANTITY IN D10 REFIX317   3
*       FOR USE IN RESET).                                     REFIX317   4
*                                                              UVDOTB    95
      CALL VOL1(DELKZ,ALFZ,D3,D10,VOL1,1                       REFIX317   6
     * , IF,JF,N,IFP1,JFP1,IFM1,JFM1)                          UVDOTB    97
*                                                              UVDOTB    98
      DO 980 L=1,2                                             UVDOTB    99
      DO 980 I=2,IFM1                                          UVDOTB   100
      DO 980 J=2,JFM1                                          UVDOTB   101
  980 ANSR(I-1,J-1,L,K)=ANSR(I-1,J-1,L,K) + VOL1(I,J)          UVDOTB   102
*                                                              UVDOTB   103
```

```
C      NEXT TERM --- START ON BRACKET
C
       DO 1000 L=1,2
       DO 1000 I=1,IF
       DO 1000 J=1,JF
  1000 D12(I,J,L)=UUV(I,J,K,1) * UUV(I,J,K,3-L)
C****
       DO 1010 I=2,IFM1
       DO 1010 J=2,JF
  1010 D7(I,J)=TPL(I+1,J,M) + TPL(I,J,M)
C****
       DO 1020 I=2,IFM1
       DO 1020 J=2,JFM1
  1020 DD(I,J)=D7(I,J+1) + D7(I,J)
C
       DO 1030 I=2,IFM1
       DO 1030 J=2,JFM1
  1030 D32(I,J,1)=DD(I,J) * .25E0
C****
       DO 1040 I=2,IFM1
       DO 1040 J=2,JFM1
  1040 D7(I,J)=TLL(I+1,J,M) + TLL(I,J,M)
C
       DO 1050 I=2,IFM1
       DO 1050 J=2,JFM1
  1050 D32(I,J,2)=D7(I,J) * .5E0
C
C
       DO 1060 L=1,2
       DO 1060 I=2,IFM1
       DO 1060 J=2,JFM1
  1060 D12(I,J,L)=D12(I,J,L) - D32(I,J,L)
C
       DO 1070 J=2,JFM1
  1070 D7(1,J)=CBARI(J) * SBAR(J)
C
       DO 1080 J=2,JFM1
  1080 D7(1,J)=D7(1,J) * AINV
C
       DO 1130 L=1,2
       DO 1130 I=2,IFM1
       DO 1130 J=2,JFM1
  1130 D12(I,J,L)=D12(I,J,L) * D7(1,J)
C
       DO 1140 I=2,IFM1
       DO 1140 J=2,JFM1
  1140 ANSR(I-1,J-1,1,K)=ANSR(I-1,J-1,1,K) + D12(I,J,1)
C
       DO 1150 I=2,IFM1
       DO 1150 J=2,JFM1
  1150 ANSR(I-1,J-1,2,K)=ANSR(I-1,J-1,2,K) - D12(I,J,2)
C
C      NEXT TERM
C
       DO 1170 L=1,2
       DO 1170 I=2,IFM1
       DO 1170 J=2,JFM1
  1170 D12(I,J,L)=F(I,J) * UUV(I,J,K,3-L)
```

```
C                                                                UVDOTB   162
      DO 1100 I=2,IFM1                                           UVDOTB   163
      DO 1100 J=2,JFM1                                           UVDOTB   164
 1100 ANSR(I-1,J-1,1,K)=ANSR(I-1,J-1,1,K) + D12(I,J,1)          UVDOTB   165
C                                                                UVDOTB   166
      DO 1190 I=2,IFM1                                           UVDOTB   167
      DO 1190 J=2,JFM1                                           UVDOTB   168
 1190 ANSR(I-1,J-1,2,K)=ANSR(I-1,J-1,2,K) - D12(I,J,2)          UVDOTB   169
C                                                                UVDOTB   170
C     LAST TERM IN U DOT EQUATION                                UVDOTB   171
C                                                                UVDOTB   172
      IF (.NOT.OLDPI) GO TO 1341                                 PISWEEP6   7
C****                                                            UVDOTB   173
      DO 1310 I=2,IFM1                                           UVDOTB   174
      DO 1310 J=2,JF                                             UVDOTB   175
 1310 D1(I,J)=ZBRIJK(I+1,J,N) - ZBRIJK(I,J,N)                    UVDOTB   176
C****                                                            UVDOTB   177
      DO 1320 I=2,IFM1                                           UVDOTB   178
      DO 1320 J=2,JFM1                                           UVDOTB   179
 1320 D2(I,J)=D1(I,J+1) + D1(I,J)                                UVDOTB   180
C****                                                            UVDOTB   181
      DO 1330 I=2,IFM1                                           UVDOTB   182
      DO 1330 J=2,JFM1                                           UVDOTB   183
 1330 D2(I,J)=D2(I,J) * .5E0                                     UVDOTB   184
C****                                                            UVDOTB   185
      DO 1340 I=2,IFM1                                           UVDOTB   186
      DO 1340 J=2,JFM1                                           UVDOTB   187
 1340 D2(I,J)=D2(I,J) * PIAV(I,J,3)                              UVDOTB   188
 1341 CONTINUE                                                   PISWEEP6   9
C****                                                            UVDOTB   189
      DO 1350 I=2,IF                                             UVDOTB   190
      DO 1350 J=1,JF                                             UVDOTB   191
 1350 D5(I,J)=DELRZ(I,J,N) + DELRZ(I-1,J,N)                      UVDOTB   192
C****                                                            UVDOTB   193
      DO 1360 I=2,IFM1                                           UVDOTB   194
      DO 1360 J=1,JF                                             UVDOTB   195
 1360 D6(I,J)=D5(I+1,J) + D5(I,J)                                UVDOTB   196
C****                                                            UVDOTB   197
      DO 1370 I=2,IFM1                                           UVDOTB   198
      DO 1370 J=2,JF                                             UVDOTB   199
 1370 D5(I,J)=D6(I,J) + D6(I,J-1)                                UVDOTB   200
C****                                                            UVDOTB   201
      DO 1380 I=2,IFM1                                           UVDOTB   202
      DO 1380 J=2,JFM1                                           UVDOTB   203
 1380 D6(I,J)=D5(I,J+1) + D5(I,J)                                UVDOTB   204
C****                                                            UVDOTB   205
      DO 1390 I=2,IFM1                                           UVDOTB   206
      DO 1390 J=2,JFM1                                           UVDOTB   207
 1390 D5(I,J)=D6(I,J) * .0625E0                                  UVDOTB   208
      IF (.NOT.OLDPI) GO TO 1413                                 PISWEEP6  11
C****                                                            UVDOTB   209
      DO 1400 I=2,IFM1                                           UVDOTB   210
      DO 1400 J=2,JFM1                                           UVDOTB   211
 1400 D1(I,J)=D5(I,J) * PIAV(I,J,1)                              UVDOTB   212
C                                                                UVDOTB   213
      DO 1410 I=2,IFM1                                           UVDOTB   214
      DO 1410 J=2,JFM1                                           UVDOTB   215
 1410 D2(I,J)=D1(I,J) - D2(I,J)                                  UVDOTB   216
```

```
      *                                                        PISWEEP6   13
      GO TO 1417                                               PISWEEP6   14
 1413 CONTINUE                                                 PISWEEP6   15
      DO 1415 J=2,JFM1                                         PISWEEP6   16
      DO 1415 I=2,IFM1                                         PISWEEP6   17
 1415 D2(I,J)=GRADPI(I,J,1) * D5(I,J)                          PISWEEP6   18
 1417 CONTINUE                                                 PISWEEP6   19
      *                                                        UVDOTB    217
      A=1.E0 / AINV                                            UVDOTB    218
      DO 1420 J=2,JFM1                                         UVDOTB    219
 1420 D1(1,J)=A * BDBAR(J)                                     UVDOTB    220
      *                                                        UVDOTB    221
      DO 1430 I=2,IFM1                                         UVDOTB    222
      DO 1430 J=2,JFM1                                         UVDOTB    223
 1430 D2(I,J)=D2(I,J) * D1(1,J)                                UVDOTB    224
      *                                                        UVDOTB    225
      DO 1435 I=2,IFM1                                         UVDOTB    226
      DO 1435 J=2,JFM1                                         UVDOTB    227
 1435 D2(I,J)=D2(I,J) * VOLI(I,J)                              UVDOTB    228
      *                                                        L0726030    3
      IF (.NOT. OLDPI) GO TO 1438                              L0726030    4
      DO 1437 I=2,IFM1                                         L0726030    5
      DO 1437 J=2,JFM1                                         L0726030    6
 1437 UVVG(I,J,2)=D2(I,J) / F(I,J)                             L0726030    7
 1438 CONTINUE                                                 L0726030    8
      *                                                        DAPGPI      3
      IF (.NOT.DAPGPI) GO TO 1490                              DAPGPI      4
      DO 1440 J=2,JF                                           DAPGPI      5
      DO 1440 I=2,IFM1                                         DAPGPI      6
 1440 D7(I,J)=ZBRIJK(I,J,M) + ZBRIJK(I+1,J,M)                  DAPGPI      7
      *                                                        DAPGPI      8
      DO 1445 J=2,JFM1                                         DAPGPI      9
      DO 1445 I=2,IFM1                                         DAPGPI     10
 1445 D8(I,J)=D7(I,J) + D7(I,J+1)                              DAPGPI     11
      *                                                        DAPGPI     12
      DO 1450 J=2,JFM1                                         DAPGPI     13
      DO 1450 I=2,IFM1                                         DAPGPI     14
 1450 D8(I,J)=D8(I,J) * .25                                    DAPGPI     15
      *                                                        DAPGPI     16
      DO 1455 J=2,JFM1                                         DAPGPI     17
      DO 1455 I=2,IFM1                                         DAPGPI     18
 1455 D8(I,J)=D8(I,J) + ZBARK(I,J,M)                           DAPGPI     19
      *                                                        DAPGPI     20
      DO 1460 J=2,JFM1                                         DAPGPI     21
      DO 1460 I=2,IFM1                                         DAPGPI     22
 1460 D7(I,J)=D8(I,J) * D8(I,J)                                DAPGPI     23
      *                                                        DAPGPI     24
      DO 1465 J=2,JFM1                                         DAPGPI     25
      DO 1465 I=2,IFM1                                         DAPGPI     26
 1465 D8(I,J)=D7(I,J) * D8(I,J)                                DAPGPI     27
      *                                                        DAPGPI     28
      DO 1470 J=2,JFM1                                         DAPGPI     29
      DO 1470 I=2,IFM1                                         DAPGPI     30
 1470 D8(I,J)=D8(I,J) * GRADZB(I,J)                            DAPGPI2     3
      DO 1473 J=2,JFM1                                         F306A       3
      DO 1473 I=2,IFM1                                         F306A       4
 1473 D8(I,J)=EXP(D8(I,J))                                     F306A       5
      *                                                        DAPGPI     32
```

```
        DO 1475 J=2,JFM1                                          DAPGPI      33
        DO 1475 I=2,IFM1                                          DAPGPI      34
   1475 D2(I,J)=D2(I,J) * DB(I,J)                                 DAPGPI      35
   1490 CONTINUE                                                  DAPGPI      36
*                                                                 UVDOTB     229
        DO 1495 I=2,IFM1                                          DAPGPI      38
        DO 1495 J=2,JFM1                                          DAPGPI      39
   1495 ANSR(I-1,J-1,1,K)=ANSR(I-1,J-1,1,K) - D2(I,J)             DAPGPI      40
*                                                                 UVDOTB     233
        IF (.NOT.OLDPI) GO TO 1563                                PISWEEP6    21
*HHH                                                              UVDOTB     234
        DO 1510 I=2,IF                                            UVDOTB     235
        DO 1510 J=2,JFM1                                          UVDOTB     236
   1510 D1(I,J)=ZBRIJK(I,J+1,K) - ZBRIJK(I,J,K)                   UVDOTB     237
*HHH                                                              UVDOTB     238
        DO 1520 I=2,IFM1                                          UVDOTB     239
        DO 1520 J=2,JFM1                                          UVDOTB     240
   1520 D2(I,J)=D1(I+1,J) + D1(I,J)                               UVDOTB     241
*HHH                                                              UVDOTB     242
        DO 1530 I=2,IFM1                                          UVDOTB     243
        DO 1530 J=2,JFM1                                          UVDOTB     244
   1530 D1(I,J)=D2(I,J) * .5E0                                    UVDOTB     245
*                                                                 UVDOTB     246
        DO 1540 I=2,IFM1                                          UVDOTB     247
        DO 1540 J=2,JFM1                                          UVDOTB     248
   1540 D1(I,J)=D1(I,J) * PIAV(I,J,3)                             UVDOTB     249
*                                                                 UVDOTB     250
        DO 1550 I=2,IFM1                                          UVDOTB     251
        DO 1550 J=2,JFM1                                          UVDOTB     252
   1550 D2(I,J)=DS(I,J) * PIAV(I,J,2)                             UVDOTB     253
*                                                                 UVDOTB     254
        DO 1560 I=2,IFM1                                          UVDOTB     255
        DO 1560 J=2,JFM1                                          UVDOTB     256
   1560 D2(I,J)=D2(I,J) - D1(I,J)                                 UVDOTB     257
*                                                                 PISWEEP6    23
        GO TO 1567                                                PISWEEP6    24
   1563 CONTINUE                                                  PISWEEP6    25
        DO 1565 J=2,JFM1                                          PISWEEP6    26
        DO 1565 I=2,IFM1                                          PISWEEP6    27
   1565 D2(I,J)=GRADPI(I,J,2) * DS(I,J)                           PISWEEP6    28
   1567 CONTINUE                                                  PISWEEP6    29
*                                                                 UVDOTB     258
        DO 1570 I=2,IFM1                                          UVDOTB     259
        DO 1570 J=2,JFM1                                          UVDOTB     260
   1570 D1(I,J)=CBAR(J) * A                                       UVDOTB     261
*                                                                 UVDOTB     262
        DO 1580 I=2,IFM1                                          UVDOTB     263
        DO 1580 J=2,JFM1                                          UVDOTB     264
   1580 D3(I,J)=D1(I,J) * DLBAR(I)                                UVDOTB     265
*                                                                 UVDOTB     266
        DO 1590 I=2,IFM1                                          UVDOTB     267
        DO 1590 J=2,JFM1                                          UVDOTB     268
   1590 D2(I,J)=D2(I,J) * D3(I,J)                                 UVDOTB     269
*                                                                 UVDOTB     270
        DO 1592 I=2,IFM1                                          L0726030    10
        DO 1592 J=2,JFM1                                          L0726030    11
   1592 D2(I,J)=D2(I,J) * VOLI(I,J)                               L0726030    12
*                                                                 L0726030    13
```

```
      IF (.NOT.OLDPI) GO TO 1595                          L0726030    14
      DO 1593 I=2,IFR1                                    L0726030    15
      DO 1593 J=2,JFR1                                    L0726030    16
 1593 UAVG(I,J,1)=D2(I,J) / F(I,J)                        L0726030    17
*                                                         L0726030    18
      DO 1594 I=2,IFR1                                    L0726030    19
      DO 1594 J=2,JFR1                                    L0726030    20
 1594 UAVG(I,J,1)=-UAVG(I,J,1)                            L0726030    21
*                                                         L0726030    22
      WRITE (12) UAVG                                     L0726030    23
*                                                         L0726030    24
 1595 CONTINUE                                            L0726030    25
*                                                         DAPGPI      42
      IF (.NOT.DAPGPI) GO TO 1598                         DAPGPI      43
      DO 1597 J=2,JFR1                                    DAPGPI      44
      DO 1597 I=2,IFR1                                    DAPGPI      45
 1597 D2(I,J)=D2(I,J) + D8(I,J)                           DAPGPI      46
 1598 CONTINUE                                            DAPGPI      47
*                                                         UVDOTB     274
      DO 1600 I=2,IFR1                                    UVDOTB     275
      DO 1600 J=2,JFR1                                    UVDOTB     276
 1600 ANSA(I-1,J-1,2,K)=ANSA(I-1,J-1,2,K) - D2(I,J)       UVDOTB     277
*                                                         UVDOTB     303
*                                                         UVDOTB     304
      RETURN                                              UVDOTB     305
      END                                                 UVDOTB     306
```

366 ACTIVE LINE(S)   40 INACTIVE LINE(S)

SE UVGEO

```
*** SAS109 : DECK UVGEO    IS EDITED FROM SPL       FILE
CREATED ON 07/26/83 AT 13:03:37        LAST UPDATED ON 07/27/83 AT 10:25:31 BY SAS VERSION 5.27
LANGUAGE:          USER INFORMATION:

MODSETS PREVIOUSLY APPLIED TO SPL: L0727830

            SUBROUTINE UVGEO                                          UVGEO       1
      *                                                               UVGEO       2
      *     THIS ROUTINE TAKES THE 2-D DATA WRITTEN OUT BY UVGND, AND UVGEO       3
      *       CREATES A 3-D ARRAY                                     UVGEO       4
      *                                                               UVGEO       5
      @CALL ALLMAC                                                    UVGEO       6
      *                                                               UVGEO       7
            DIMENSION U6V3D(IF,JF,KF,2)                               UVGEO       8
           * , U6VG(IF,JF,2)                                          L0727830    3
            EQUIVALENCE (U6V3D,ANS)                                   UVGEO       9
           * , (D7,U6VG)                                              L0727830    5
      *                                                               UVGEO      10
            REWIND 12                                                 UVGEO      11
      *                                                               UVGEO      12
            DO 100 K=2,KFM1                                           UVGEO      13
            READ (12) U6VG                                            UVGEO      14
      *                                                               UVGEO      15
            DO 90 L=1,2                                               UVGEO      16
            DO 90 J=2,JFM1                                            UVGEO      17
            DO 90 I=2,IFM1                                            UVGEO      18
            U6V3D(I,J,K,L)=U6VG(I,J,L)                                UVGEO      19
         90 CONTINUE                                                  UVGEO      20
        100 CONTINUE                                                  UVGEO      21
      *                                                               UVGEO      22
      *     BOUNDARY CONDITIONS                                       UVGEO      23
      *                                                               UVGEO      24
            DO 200 L=1,2                                              UVGEO      25
            DO 200 J=2,JFM1                                           UVGEO      26
            DO 200 I=2,IFM1                                           UVGEO      27
            U6V3D(I,J,KF,L)=U6V3D(I,J,KFM1,L)                         UVGEO      28
            U6V3D(I,J, 1,L)=U6V3D(I,J,   2,L)                         UVGEO      29
        200 CONTINUE                                                  UVGEO      30
      *                                                               UVGEO      31
      *     APPLY BOUNDARY CONDITIONS:  EQNS. 3.36 AND 3.37           UVGEO      32
      *       (NO FLUX LATERAL BOUNDARY.)                             UVGEO      33
      *                                                               UVGEO      34
            DO 200 K=1,KF                                             UVGEO      35
      *                                                               UVGEO      36
      ****U                                                           UVGEO      37
            L=1                                                       UVGEO      38
      *                                                               UVGEO      39
            DO 220 J=2,JFM1                                           UVGEO      40
            U6V3D( 1,J,K,L)= - U6V3D(   2,J,K,L)                      UVGEO      41
            U6V3D(IF,J,K,L)= - U6V3D(IFM1,J,K,L)                      UVGEO      42
        220 CONTINUE                                                  UVGEO      43
      *                                                               UVGEO      44
            DO 230 I=1,IF                                             UVGEO      45
            U6V3D(I, 1,K,L)=U6V3D(I,    2,K,L)                        UVGEO      46
            U6V3D(I,JF,K,L)=U6V3D(I,JFM1,K,L)                         UVGEO      47
        230 CONTINUE                                                  UVGEO      48
```

```
      *
      *****V
            L=2
            DO 240 J=2,JFM1
            UUV3D( 1,J,K,L)=UUV3D(    2,J,K,L)
            UUV3D(IF,J,K,L)=UUV3D(IFM1,J,K,L)
  240 CONTINUE
      *
            DO 250 I=1,IF
            UUV3D(I, 1,K,L)= - UUV3D(I,    2,K,L)
            UUV3D(I,JF,K,L)= - UUV3D(I,JFM1,K,L)
  250 CONTINUE
      *
  200 CONTINUE
      *
            REWIND 12
      *
            RETURN
            END
```

69 ACTIVE LINE(S)   0 INACTIVE LINE(S)

Appendix B

TAPER JSL to Produce Combined Data Set and Listing of TAPER2

CREATED ON 02/15/84 AT 15:37:12 LAST UPDATED ON 02/15/84 AT 15:37:12 BY SMS VERSION 5.27
LANGUAGE: USER INFORMATION:

```
/ COM
/ COM       MACROS:
/ COM
/ COM    GET IS LIKE ASG
/ COM   RGET IS LIKE REL AND THEN ASG
/ COM    PUT IS LIKE CATV
/ COM    SEE IS LIKE FOSYS
/ COM
/ COM     PT IS THE PATH FOR THE TAPER OBJECT LIBRARY
/ COM
/ LIMIT MIN=2
/ RGET OBJLIB,PT/OBJLIB
/ REL SYS.LMOD
/ LNK
LIBRARY OBJLIB
INCLUDE TAPER2
/ REL FT31F001,FT32F001,FT33F001,FT34F001,FT35F001,FT36F001
/ REL FT41F001,FT42F001,FT43F001,FT44F001,FT45F001,FT46F001
/ REL FT51F001,FT52F001,FT53F001,FT54F001,FT55F001,FT56F001
/ REL FT71F001,FT72F001
/ RGET FT61F001,DOD/NAVY/NORDA/WARNA1/FNOCTAPE/SAIOCN1
/ SET N=1
/        REL FT06F001
/      FD FT06F001,BAND=2/20/2
/ FD FT71F001,BAND=4/50/4
/ FXQT OPT=(I),LTP=(99,99,N),ADDMEM=24K,CPTIME=6000
 &ONE
     IS=05, IE=28, JS=13, JE=51,
     YEAR=76,MONTH=10,DAY=29,HOUR=0,
     DT=24.,NT=1,NCATST=7,
     SIGMAD=T,
     CATNO='B10+','P14+','P15+','P16+','P17+','P18+','P19+',
 &END
/ COM
/ COM    OUTPUT ON FT71F001
/ COM
/ IF TERM.NE.0,ERR
/ SEE N,FT06F001,NAME=FILE6ONE
/ RGET OBJLIB,PT/OBJLIB
/ REL SYS.LMOD
/ LNK
LIBRARY OBJLIB
INCLUDE TAPER3
/ REL FT31F001,FT32F001,FT33F001,FT34F001,FT35F001,FT36F001
/ REL FT41F001,FT42F001,FT43F001,FT44F001,FT45F001,FT46F001
/ REL FT51F001,FT52F001,FT53F001,FT54F001,FT55F001,FT56F001
/ COM
/ COM  TAPER3 WRITES ON FT72 -> CHANGE SO THAT WE WILL APPEND
/ COM
/ REL FT72F001
/ RENAME FT71F001,FT72F001
/ FD FT72F001,POS=MOD
/ RGET FT61F001,AFFIL/IND/SAI/SEFTJ1/USER/NEDN/CLIMAT
/ SET N=1
```

```
/         REL FT06F001                                        TWODATA 55
/      FD FT06F001,BAND=2/20/2                                TWODATA 56
/ FXOT OPT=(I),LTP=(99,99,N),ADDMEM=24K,CPTIME=4000           TWODATA 57
 &ONE                                                         TWODATA 58
      IS=05, IE=28, JS=13, JE=51,                             TWODATA 59
      NPN=20, SIGNAD=T,                                       TWODATA 60
 &END                                                         TWODATA 61
T  SEAJ                                                       TWODATA 62
T0400 J                                                       TWODATA 63
T0600 J                                                       TWODATA 64
T0800 J                                                       TWODATA 65
T1000 J                                                       TWODATA 66
T1500 J                                                       TWODATA 67
T2000 J                                                       TWODATA 68
T3000 J                                                       TWODATA 69
T4000 J                                                       TWODATA 70
T5000 J                                                       TWODATA 71
S0000 J                                                       TWODATA 72
S0050 J                                                       TWODATA 73
S0100 J                                                       TWODATA 74
S0200 J                                                       TWODATA 75
S0600 J                                                       TWODATA 76
S1000 J                                                       TWODATA 77
S2000 J                                                       TWODATA 78
S3000 J                                                       TWODATA 79
S4000 J                                                       TWODATA 80
S5000 J                                                       TWODATA 81
/ IF TERM.NE.0,ERR                                            TWODATA 82
/ PUT FT72F001,US/INITLDTA                                    TWODATA 83
/ SEE N,FT06F001,NAME=FILE6TWO                                TWODATA 84
/         REL FT31F001,FT32F001,FT33F001,FT34F001,FT35F001,FT36F001  TWODATA 85
/ REL FT41F001,FT42F001,FT43F001,FT44F001,FT45F001,FT46F001   TWODATA 86
/ REL FT51F001,FT52F001,FT53F001,FT54F001,FT55F001,FT56F001   TWODATA 87
/ERR NOP                                                      TWODATA 88

        88 ACTIVE LINE(S)              0 INACTIVE LINE(S)

*** SMS167 : PDS DIRECTORY FOR NEWSPL   FILE SUCCESSFULLY UPDATED FOR DECK TWODATA
*
```

```
SMS VERSION 5.27    LISTING OF DECK    T A P E R 2                                                          01/12/84          PAGE 1

*** SMS109 : DECK TAPER2   IS EDITED FROM SPL       FILE
CREATED ON 07/31/80 AT 10:13:40           LAST UPDATED ON 04/04/83 AT 11:46:25 BY SMS VERSION 5.27
LANGUAGE:              USER INFORMATION:

MODSETS PREVIOUSLY APPLIED TO SPL: F07310    F0909A    F0911A    F0922A    F1008A    F1103A    L0522016  L072701A  L080401A
     L033103A  L040403A

             PROGRAM TAPER2(NTAPES)                                                              TAPER2        1
       *                                                                                         TAPER2        2
       *     THIS PROGRAM, GIVEN A TIME AND SPACE WINDOW, AND A REQUIRED                          TAPER2        3
       *       OUTPUT TIME INTERVAL, WILL PROVIDE DATA FROM FNWC GENERATED                        TAPER2        4
       *       TAPES.                                                                            TAPER2        5
       *     THE FILE ALLOCATION IS AS FOLLOWS                                                   TAPER2        6
       *       61-69    INPUT TAPES                                                              TAPER2        7
       *       31-50    DATA IN SPACE-TIME WINDOW (ANY TIME INTERVAL)                            TAPER2        8
       *       71       OUTPUT FILE                                                              TAPER2        9
       *                                                                                         TAPER2       10
       $CALL AATAPE                                                                              TAPER2       11
       *                                                                                         TAPER2       12
             PARAMETER NCATP=17                                                                  TAPER2       13
       *                                                                                         TAPER2       14
             REAL*4 D(63,63,NCATP)                                                               TAPER2       15
            . , DZ(63,63,2)                                                                      F0909A        3
             DATA CATNO/'AAMZ','AABZ','AACZ','AADZ','AAEZ','AAFZ','AAIZ','AAQZ'                  TAPER2       16
            . , 'AARZ','AASZ','AATZ','AAWZ','AAVZ','AAWZ','AAXZ','AAYZ','AAZZ'/                  TAPER2       17
            . , INDEX/NCATP*0/                                                                   TAPER2       18
       *                                                                                         TAPER2       19
             NAMELIST/ONE/YEAR,MONTH,DAY,HOUR,IS,IE,JS,JE,NT,DT                                  TAPER2       20
            . , CHKMED,NODISC,SUMMRY                                                             L0522016      3
            . , NCATST,CATNO,SIGNAD                                                              L033103A      3
       *                                                                                         TAPER2       22
             CALL RSSTOP                                                                         TAPER2       23
       *                                                                                         TAPER2       24
             NCATNO=NCATP                                                                        TAPER2       25
       *                                                                                         TAPER2       26
             DO 10 N=1,NCATNO                                                                    TAPER2       27
          10 SORT(N)=.FALSE.                                                                     TAPER2       28
             ABTRN=.FALSE.                                                                       TAPER2       29
             LU=5                                                                                TAPER2       30
             CHKMED=.FALSE.                                                                      TAPER2       31
             NODISC=.FALSE.                                                                      F1008A        7
             SUMMRY=.FALSE.                                                                      L0522016      5
             NCATST=0                                                                            L080401A      5
             SIGNAD=.FALSE.                                                                      L033103A      5
             READ(LU,ONE)                                                                        TAPER2       32
             IF (NCATST.NE.0) NCATNO=NCATST                                                      L080401A      7
             IF (CHKMED .OR. SUMMRY) GO TO 25                                                    L0522016      7
       *                                                                                         TAPER2       33
       *     DO DATA CHECK                                                                       TAPER2       34
       *                                                                                         TAPER2       35
             IF (YEAR.LT.85 .AND.  YEAR .GT.65        .AND.                                      TAPER2       36
            .    MONTH.GE.1   .AND.  MONTH.LE.12        .AND.                                    TAPER2       37
            .    DAY   .GE.1  .AND.  DAY   .LE.31       .AND.                                    TAPER2       38
            .    HOUR  .GE.0  .AND.  HOUR  .LE.24       .AND.                                    TAPER2       39
            .    IS    .GE.1  .AND.  IS    .LE.62       .AND.                                    TAPER2       40
            .    IE    .GE.2  .AND.  IE    .LE.63       .AND.                                    TAPER2       41
```

```
     .  JS   .GE.1  .AND.  JS   .LE.62         .AND.          TAPER2     42
     .  JE   .GE.2  .AND.  JE   .LE.63         .AND.          TAPER2     43
     .  NT   .GE.1  .AND.  NT   .LE.1000       .AND.          F1103A      3
     .  DT   .GE.1. .AND.  DT   .LE.24.)                      F0731B      3
     . GO TO 20                                               TAPER2     46
      WRITE (6,ONE)                                           TAPER2     47
      CALL STOPP("ERROR IN DATA ENTRY$")                      TAPER2     48
   20 CONTINUE                                                TAPER2     49
*                                                             TAPER2     50
*     FIND STARTING TIME                                      TAPER2     51
*                                                             TAPER2     52
      KS=(((YEAR*100)+MONTH)*100+DAY)*100+HOUR                TAPER2     53
*                                                             TAPER2     54
*     FIND ENDING TIME                                        TAPER2     55
*                                                             TAPER2     56
      IT=IFIX(FLOAT(NT-1) * DT)                               TAPER2     57
      KE=TIMEA(KS,IT)                                         TAPER2     58
*                                                             TAPER2     59
*     READ TAPES, WRITE OUT INFORMATION TO DIFFERENT FILES FOR ALL    TAPER2     60
*       DATA IN THE TIME-SPACE WINDOW.                        TAPER2     61
*                                                             TAPER2     62
   25 CONTINUE                                                L0522081G   9
      ITYPE=4                                                 F0911A      3
      CALL NEDMC(NTAPES,ITYPE,D2,D2(1,1,2))                   F0911A      4
*                                                             TAPER2     64
      DO 30 N=1,NCATNO                                        TAPER2     65
      NF=30 + N                                               TAPER2     66
      REWIND NF                                               TAPER2     67
   30 CONTINUE                                                TAPER2     68
*                                                             TAPER2     69
*     SEE IF EVERYTHING IS IN THE CORRECT ORDER               TAPER2     70
*                                                             TAPER2     71
      DO 200 N=1,NCATNO                                       TAPER2     72
      IF (.NOT.SORT(N)) GO TO 200                             TAPER2     73
      CALL FSORT(30+N,54,DTIMES(1,N),D,IS,IE,JS,JE,INDEX(N))  F0922A      3
  200 CONTINUE                                                TAPER2     77
*                                                             TAPER2     78
      NTREAL=INDEX(1)                                         F0731B      5
      DO 250 N=2,NCATNO                                       TAPER2     79
      IF (INDEX(N).EQ.INDEX(1)) GO TO 250                     TAPER2     80
      GO TO 260                                               TAPER2     81
  250 CONTINUE                                                TAPER2     82
      GO TO 280                                               TAPER2     83
  260 PRINT 932                                               TAPER2     84
  932 FORMAT ("  WARNING:  THERE ARE NOT AN EQUAL NUMBER OF EOTS"     TAPER2     85
     . , " PARAMETERS IN EACH CATAGORY.")                     TAPER2     86
      DO 270 N=2,NCATNO                                       TAPER2     88
  270 IF (INDEX(N).LT.NTREAL) NTREAL=INDEX(N)                 TAPER2     89
  280 CONTINUE                                                TAPER2     90
*                                                             TAPER2     91
*                                                             TAPER2     92
      DO 500 NI=1,NTREAL                                      TAPER2     93
*                                                             TAPER2     94
      DO 300 N=1,NCATNO                                       TAPER2     95
      NF=30 + N                                               TAPER2     96
  300 READ(NF,END=310) ((D(I,J,N),I=IS,IE),J=JS,JE)           TAPER2     97
      GO TO 320                                               TAPER2     98
  310 ENCODE (80,930,LABEL) N                                 TAPER2     99
```

```
  930 FORMAT('TAPER HIT AN END OF FILE FOR CAT NO ',I2,'$')
      GO TO 1000
  320 CONTINUE
*
      DO 330 N=2,NCATNO
      IF (DTIMES(N1,1) .EQ. DTIMES(N1,N)) GO TO 330
      GO TO 340
  330 CONTINUE
*
*     EVERYTHING IS OK
*
      GO TO 440
  340 CONTINUE
*
*     ALL TIMES ARE NOT EQUAL
*
      TIME=DTIMES(N1,1)
      DO 400 N=2,NCATNO
  400 IF (DTIMES(N1,N) .GT. TIME) TIME=DTIMES(N1,N)
*
      DO 430 N=1,NCATNO
      IF (DTIMES(N1,N) .EQ. TIME) GO TO 430
*
  410 CONTINUE
*
*     THIS FILE HAS A TIME THAT IS TOO EARLY, SO SHIFT
*
      NF=30 + N
      READ(NF,END=310) ((D(I,J,N),I=IS,IE),J=JS,JE)
      NS=N1
      INDEX(N)=INDEX(N) - 1
      NE=INDEX(N)
      DO 420 NN=NS,NE
  420 DTIMES(NN,N)=DTIMES(NN+1,N)
*
      PRINT 934,N
  934 FORMAT (' FILE ',I2,' WILL BE SKIPPED ONE ENTRY')
*
      IF (DTIMES(N1,N) .EQ. TIME) GO TO 430
      IF (DTIMES(N1,N) .LT. TIME) GO TO 410
*
      ENCODE(80,931,LABEL) TIME,N
  931 FORMAT('FILE AT CORRECT TIME (',I8,') CANNOT BE FOUND FOR CAT NO '
     * , I2,'$')
      GO TO 1000
*
  430 CONTINUE
*
*
  440 CONTINUE
*
*     WRITE OUT
*
      IF (NODISC) GO TO 470
      I=DTIMES(N1,1)
      J=JULIAN(I)
      HR=MOD(I,100)
      I=I/100
```

```
           DAY=MOD(I,100)                                              TAPER2      157
           I=I/100                                                     TAPER2      158
           MO=MOD(I,100)                                               TAPER2      159
           YR=I/100                                                    TAPER2      160
           IR=IE - IS + 1                                              TAPER2      161
           JR=JE - JS + 1                                              TAPER2      162
     *                                                                 L033103A      7
           IF (SIGMOD) GO TO 465                                       L033103A      8
     *                                                                 L033103A      9
           IF (N1.EQ.1) WRITE(71) NTREAL,J,YR,MO,DAY,HR,IS,IR,JS,JR    TAPER2      163
                        WRITE(71)        J,YR,MO,DAY,HR,IS,IR,JS,JR    F1000A       11
     *                                                                 TAPER2      165
           DO 460 J=JS,JE                                              TAPER2      166
       460 WRITE(71) ((D(I,J,N),N=1,NCATNO),I=IS,IE)                   TAPER2      167
     *                                                                 F1000A       13
           GO TO 500                                                   F1000A       14
     *                                                                 L033103A     11
     *     SIGMA CODE OUTPUT                                           L033103A     12
     *                                                                 L033103A     13
       465 CONTINUE                                                    L033103A     14
     *                                                                 L033103A     15
           WRITE (71) NCATNO,IS,IR,JS,JR,J,YR,MO,DAY,HR                L033103A     16
     *                                                                 L033103A     17
           DO 467 N=1,NCATNO                                           L033103A     18
           WRITE (71) CATNO(N)                                         L033103A     19
           WRITE (71) ((D(I,J,N),I=IS,IE),J=JS,JE)                     L033103A     20
       467 CONTINUE                                                    L033103A     21
     *                                                                 L033103A     22
           GO TO 500                                                   L033103A     23
       470 CONTINUE                                                    F1000A       15
           DO 480 N=1,NCATNO                                           F1000A       16
           WRITE (LUO,926) CATNO(N),DTIMES(N1,1)                       F1000A       17
       480 CALL FPRT(D(1,1,N),IS,IE,JS,JE,2,0)                         F1000A       18
       926 FORMAT (////,10X,"CATALOG NUMBER ",A4," AT TIME ",I8," IST",//)  F1000A  19
     *                                                                 TAPER2      168
       500 CONTINUE                                                    TAPER2      169
     *                                                                 TAPER2      170
           IF (NTREAL .LT. 2) GO TO 525                                L040403A      3
           DO 520 N=2,NTREAL                                           TAPER2      171
           I=TIMES( DTIMES(N,1),DTIMES(N-1,1) )                        TAPER2      172
           IF (FLOAT(I).LE.DT) GOTO 520                                TAPER2      173
           II=TIMEA(DTIMES(N-1,1),IFIX(DT))                            TAPER2      174
           PRINT 933,II                                                TAPER2      175
       933 FORMAT(" MISSING EOTS FIELDS AT TIME: ",I8)                 TAPER2      176
       520 CONTINUE                                                    TAPER2      177
       525 CONTINUE                                                    L040403A      5
     *                                                                 TAPER2      178
     *                                                                 TAPER2      179
     *     SUMMARY LISTING                                             TAPER2      180
     *                                                                 TAPER2      181
           GO TO 1010                                                  TAPER2      182
      1000 ABTHM=.TRUE.                                                TAPER2      183
      1010 LUO=6                                                       TAPER2      184
           NTM1=NT - 1                                                 TAPER2      185
           WRITE (LUO,921) KS,KE,IS,IE,JS,JE,NTM1,DT                   TAPER2      186
           DO 1100 N=1,NCATNO                                          TAPER2      187
           WRITE (LUO,923) N,CATNO(N),INDEX(N),SORT(N)                 TAPER2      188
      1100 CONTINUE                                                    TAPER2      189
```

```
      921 FORMAT("1",20X,20(1H*),"   TAPE R(EAD) SUMMARY LISTING   ",20(1H*)      TAPER2   190
        . . //,10X,"THE TIME WINDOW RUNS FROM ",I8," TO ",I8,".",                  TAPER2   191
        . . /,10X,"AND THE SPACE WINDOW RUNS FROM I=",I2," TO "                    TAPER2   192
        . . I2,",  AND J=",I2," TO ",I2,"."                                        TAPER2   193
        . . //,10X,I2," INTERVALS OF ",F4.1," HOURS WERE REQUESTED."               TAPER2   194
        . . ///,T15,"#",T25,"CATALOG NO",T39,"# OF ENTRIES"                        TAPER2   195
        . . T54,"NEEDS SORTING",T71,"ORDER OUT"                                    TAPER2   196
        . . T86,"SCALED BY"                                                        TAPER2   197
        . . //)                                                                    TAPER2   198
      923 FORMAT (T14,I2,T20,A4,T42,I3,T60,L1                                      TAPER2   199
        . . T74,I2,T85,1PE10.3                                                     TAPER2   200
        . )                                                                        TAPER2   201
          IF (ABTRM) CALL STOPP(LABEL)                                             TAPER2   202
    .                                                                              TAPER2   203
    .                                                                              TAPER2   204
          STOP                                                                     TAPER2   205
          END                                                                      TAPER2   206

          240 ACTIVE LINE(S)           13 INACTIVE LINE(S)
```

Appendix C

Plotting Routines

```
SMS VERSION 5.27    LISTING OF DECK    C O N I N T                                   01/12/84           PAGE 1

*** SMS109 : DECK CONINT   IS EDITED FROM SPL      FILE
CREATED ON 08/01/83 AT 11:06:24          LAST UPDATED ON 08/01/83 AT 11:06:24 BY SMS VERSION 5.27
LANGUAGE:              USER INFORMATION:

        1          SUBROUTINE CONINT(X,Y,N,S1,S2)                                                CONINT       1
        2          DIMENSION X(N),Y(N)                                                           CONINT       2
        3          COMMON/CINT1/CVALUE                                                           CONINT       3
        4          CALL CONCRV(X,Y,N,CVALUE)                                                     CONINT       4
        5          RETURN                                                                        CONINT       5
        6          END                                                                           CONINT       6

              6 ACTIVE LINE(S)            0 INACTIVE LINE(S)

RE HSLAB
```

```
*** SMS109 : DECK HSLAB    IS EDITED FROM SPL      FILE
CREATED ON 08/27/80 AT 16:13:45          LAST UPDATED ON 08/08/83 AT 12:23:33 BY SMS VERSION 5.27
LANGUAGE:           USER INFORMATION:

MODSETS PREVIOUSLY APPLIED TO SPL: F00270    F00270    F0029A    F0029B    F1103C    F010601E  L092702A  L000103A  L000103B
     L000103C  L000203A  L000303A  L000503A  L000603A  L000603B  L000603C  L000603D  L000803A

          SUBROUTINE HSLAB(T,IF,JF,ZB,IZB,JZB,L,P                                F00270      3
         * , LABEL,ISTP,NPLOT,PAPERP,IRUN)                                       F010601E    3
    *                                                                            HSLAB       3
    *     THIS ROUTINE PLOTS A HORIZONTAL SLAB OF A DATA QUANTITY                HSLAB       4
    *                                                                            HSLAB       5
          PARAMETER ISIZE=40,JSIZE=30                                            L092702A    3
          REAL*4 L                                                               HSLAB       7
          EXTERNAL CONINT                                                        L000103B    3
          DIMENSION T(IF,JF),L(IF),P(JF),ZB(IZB,JZB)                             F00270      5
         * , CON(40),CONZ(20)                                                    F0029A      3
         * , LABEL(4),DEGL(ISIZE),DEGP(JSIZE)                                    HSLAB       10
          INTEGER  NTIT(20),PATIT(4)                                             HSLAB       11
          COMMON WORK(5000)                                                      L000103A    3
          COMMON/CINT1/CVALUE                                                    L000103A    4
          LOGICAL PAPERP                                                         L000103A    5
    *                                                                            HSLAB       12
          NAMELIST/PLTDT4/H,ISEP,NLINE,AMAG,XL,XU,YL,YU,NTIK,NCON,ZU,ZL          HSLAB       13
         * , NCONZ,ZBZL,ZBZU                                                     HSLAB       14
         * , CHMIN,CHANG                                                         L000803A    3
    *                                                                            HSLAB       15
          IF (IF.GT.ISIZE .OR. JF.GT.JSIZE)                                      F00270      7
         *CALL STOPP("ARRAYS TOO SMALL IN HSLAB$")                               F00270      8
    *                                                                            HSLAB       18
          IF (ISTP.NE.0) GO TO 80                                                HSLAB       19
    *                                                                            HSLAB       20
    *     DO INITILIZIATION                                                      HSLAB       21
    *                                                                            HSLAB       22
          CONST=57.29578                                                         HSLAB       23
          DO 10 I=1,IF                                                           F00270      10
       10 DEGL(I)=L(I) * CONST                                                   HSLAB       25
    *                                                                            HSLAB       26
          DO 20 J=1,JF                                                           F00270      12
       20 DEGP(J)=P(J) * CONST                                                   HSLAB       28
    *                                                                            HSLAB       29
    *     SET DEFAULTS                                                           HSLAB       30
    *                                                                            HSLAB       31
          H=100.                                                                 HSLAB       32
          ISEP=1                                                                 HSLAB       33
          NLINE=0                                                                HSLAB       34
          AMAG=1.                                                                HSLAB       35
          CHMIN=1.5                                                              L000803A    5
          CHANG=50.                                                              L000803A    6
    *                                                                            HSLAB       36
    *     PLACE BOUNDARIES CORRECTLY                                             HSLAB       37
    *                                                                            HSLAB       38
          IFM1=IF - 1                                                            F00270      3
          JFM1=JF - 1                                                            F00270      4
          XL=(DEGL(1    ) + DEGL(2    ))  *  .5                                  HSLAB       39
          XU=(DEGL(IFM1) + DEGL(IF   ))  *  .5                                   HSLAB       40
```

```
          YL=(DEGP(1   ) + DEGP(2   )) * .5                                  NSLAB      41
          YU=(DEGP(JFM1) + DEGP(JF  )) * .5                                  NSLAB      42
          NTIK=5                                                             NSLAB      43
          NCON=3                                                             NSLAB      44
          ZU=1.                                                              NSLAB      45
          ZL=-1.                                                             NSLAB      46
          NCONZ=0                                                            F00290      3
          ZBZL=-1.                                                           L000603A    3
          ZBZU=-1.                                                           L000603A    4
          ZBZLP=0.                                                           L000603A    5
          ZBZUP=0.                                                           L000603D    4
          READ (5,PLTDT4)                                                    NSLAB      50
          DO 25 I=1,IF                                                       L000603B    3
          DO 25 J=1,JF                                                       L000603B    4
          IF (DEGL(I).LT.XL .OR. DEGL(I).GT.XU) GO TO 25                     L000603C    3
          IF (DEGP(J).LT.YL .OR. DEGP(J).GT.YU) GO TO 25                     L000603C    4
          ZBZUP=AMAX1(ZBZUP,ZB(I,J))                                         L000603A   12
          ZBZLP=AMIN1(ZBZLP,ZB(I,J))                                         L000603D    6
       25 CONTINUE                                                           L000603A   13
     *                                                                       L000603A   14
          IF (ZBZU .LT. 0.) ZBZU=ZBZUP                                       L000603A   15
          IF (ZBZL .LT. 0.) ZBZL=ZBZLP                                       L000603D    8
     *                                                                       L000603D    9
          IF (NLINE.EQ.0) NLINE=ISEP + 42                                    NSLAB      51
          IF (NCON.GT.40 .OR. NCONZ.GT.20)                                   F0029A      5
         .CALL STOPP("YOU HAVE ASKED FOR TOO MANY CONTOURS IN NSLAB$")       F0029A      6
     *                                                                       NSLAB      52
          XC=1. / FLOAT(NCON - 1)                                            NSLAB      53
          DO 30 N=1,NCON                                                     NSLAB      54
       30 CON(N)=( ZU * FLOAT(N-1) + ZL * FLOAT(NCON-N) ) * XC               NSLAB      55
     *                                                                       NSLAB      56
          IF (NCONZ.LE.0) GO TO 45                                           F00290      5
          XC=1. / FLOAT(NCONZ - 1)                                           NSLAB      57
          DO 40 N=1,NCONZ                                                    NSLAB      58
       40 CONZ(N)=( ZBZU * FLOAT(N-1) + ZBZL * FLOAT(NCONZ-N) ) * XC         NSLAB      59
     *                                                                       NSLAB      60
          IF (PAPERP) GO TO 43                                               L000503A    3
     *                                                                       L000503A    4
          CALL BGNPL(NPLOT)                                                  L000503A    5
          CALL BASALF("L/CSTAN")                                             L000503A    6
          CALL MIXALF("STAND")                                               L000503A    7
          CALL BLOWUP(1.25)                                                  L000503A    8
          CALL NOCHEK                                                        L000503A    9
          CALL GRACE(0.)                                                     L000503A   10
     *                                                                       L000503A   11
          XTIT=9.                                                            L000503A   12
          YTIT=5.                                                            L000503A   13
          CALL TITLE(1H ,-1,0,0,0,0,XTIT,YTIT)                               L000503A   14
          CALL ENDGR(1)                                                      L000503A   15
          CALL OREL(0.,1.)                                                   L000503A   16
          CALL TITLE("DEPTH$",-100,"LONGITUDE$",100,"LATITUDE$",100          L000503A   17
         . , XTIT,YTIT)                                                      L000503A   18
          CALL GRAF(XL,"SCALE",XU,YL,"SCALE",YU)                             L000503A   19
          CALL FRAME                                                         L000503A   20
          CALL BCOMON(5000)                                                  L000503A   21
     *                                                                       L000503A   22
          CALL CONMIN(CHMIN)                                                 L000803A    8
          CALL CONANG(CHANG)                                                 L000803A    9
```

```
      CALL CONLIN(0,'SOLID',   'LABELS',1,1)                        L0005O3A   25
      CALL CONLIN(1,'SOLID','NOLABELS',1,1)                         L0005O3A   26
      CALL CONMGN                                                    L0005O3A   27
      DO 42 N=1,NCONZ                                                L0005O3A   28
      CVALUE=CONZ(N)                                                 L0005O3A   29
      CALL CNTOUR(ZB,IZB,DEGL,1,IF,DEGP,1,JF,CONZ(N),CONINT,TT)     L0005O3A   30
   42 CONTINUE                                                       L0005O3A   31
*                                                                    L0005O3A   32
      CALL CONEND                                                    L0005O3A   33
      CALL CONTUR(1,'LABELS','DRAW')                                 L0005O3A   34
*                                                                    L0005O3A   35
      CALL ENDPL(NPLOT)                                              L0005O3A   36
      GO TO 45                                                       L0005O3A   37
*                                                                    L0005O3A   38
   43 CONTINUE                                                       L0005O3A   39
*                                                                    L0005O3A   40
      CALL CONTOR(ZB,IZB,DEGL,DEGP,IF,JF,CONZ,NCONZ                  MSLAB      41
     . , ISEP,NLINE,AMAG                                             MSLAB      62
     . , XL,XU,YL,YU,6,NTIK                                          MSLAB      63
     . , DUM,DUM,0,0.,1.                                             MSLAB      64
     . , DUM,DUM,DUM,DUM,DUM,DUM,DUM                                 F0029A      8
     . , 'DEPTH$','LONGITUDE$',' $',' $',' $')                       MSLAB      65
   45 CONTINUE                                                       F0029B      7
*                                                                    MSLAB      66
      RETURN                                                         MSLAB      67
*                                                                    MSLAB      68
   80 CONTINUE                                                       MSLAB      69
*                                                                    MSLAB      70
      READ (3) T                                                     MSLAB      71
*                                                                    MSLAB      72
*     DO PLOTTING                                                    MSLAB      73
*                                                                    MSLAB      74
      IF (NCON.EQ.0) GO TO 600                                       L0005O3A   42
      ENCODE(80,970,MTIT) LABEL,ISTP,IRUN                            F0106B1E    5
      ENCODE (16,971,PXTIT)                                          MSLAB      76
  970 FORMAT(4A6,' AT TIME STEP ',I4,'   (RUN #',I3,')$')            F0106B1E    7
  971 FORMAT (' LONGITUDE $')                                        MSLAB      78
      IF (PAPERP) GO TO 500                                          L0001O3A    7
*                                                                    L0001O3A    8
*                                                                    L0001O3A    9
      CALL BGNPL(NPLOT)                                              L0001O3A   10
      CALL BASALF('L/CSTAN')                                         L0001O3A   11
      CALL MIXALF('STAND')                                           L0001O3A   12
      CALL BLOWUP(1.25)                                              L0001O3A   13
      CALL NOCHEK                                                    L0001O3A   14
      CALL GRACE(0.)                                                 L0001O3A   15
*                                                                    L0001O3A   16
      XTIT=9.                                                        L0001O3C    3
      YTIT=5.                                                        L0001O3C    4
      CALL TITLE(1H ,-1,0,0,0,0,XTIT,YTIT)                           L0001O3C    5
      CALL ENDGR(1)                                                  L0001O3C    6
      CALL OREL(0.,1.)                                               L0001O3C    7
      CALL TITLE(MTIT,-100,'LONGITUDE$',100,'LATITUDE$',100          L0001O3C    8
     . , XTIT,YTIT)                                                  L0001O3C    9
      CALL GRAF(XL,'SCALE',XU,YL,'SCALE',YU)                         L0003O3A    3
      CALL FRAME                                                     L0001O3A   20
      CALL BCOMON(5000)                                              L0001O3A   21
*                                                                    L0001O3A   22
```

```
      CALL CONMIN(3.)                                               L000103C   11
      CALL CONANG(5.)                                               L000103C   12
      CALL CONLIN(0,'SOLID',  'LABELS',1,1)                         L000103C   13
      CALL CONLIN(1,'SOLID','NOLABELS',1,1)                         L000103C   14
      CALL CONRGN                                                   L000103A   23
      DO 100 N=1,NCON                                               L000103A   24
      CVALUE=CON(N)                                                 L000103A   25
      CALL CNTOUR(T,IF,DEGL,1,IF,DEGP,1,JF,CON(N),CONINT,TT)        L000103A   26
  100 CONTINUE                                                      L000103A   27
*                                                                   L000103A   28
      CALL CONEND                                                   L000103A   29
      CALL CONTUR(1,'LABELS','DRAW')                                L000103A   30
*                                                                   L000103A   31
      CALL ENDPL(NPLOT)                                             L000103A   32
      GO TO 600                                                     L000203A    3
*                                                                   L000103A   33
  500 CONTINUE                                                      L000103A   34
*                                                                   NSLAB      79
      CALL CONTOR(T,IF,DEGL,DEGP,IF,JF,CON,NCON                     F00270     14
     . , ISEP,NLINE,ARAG                                            NSLAB      01
     . , XL,XU,YL,YU,6,NTIK                                         NSLAB      02
     . , DUM,DUM,0,0.,1.                                            NSLAB      03
     . , DUM,DUM,DUM,DUM,DUM,DUM,DUM                                F0029A     10
     . , NTIT,PXTIT,' 0',' 0',' 0')                                 NSLAB      04
*                                                                   NSLAB      05
  600 CONTINUE                                                      L000203A    5
*                                                                   NSLAB      06
      RETURN                                                        NSLAB      07
      END                                                           NSLAB      08
```

194 ACTIVE LINE(S)   26 INACTIVE LINE(S)

$E OCPLOT

```
SMS VERSION 5.27    LISTING OF DECK    O C P L O T                                          01/12/84          PAGE 1

*** SMS109 : DECK OCPLOT   IS EDITED FROM SPL       FILE
CREATED ON 08/16/79 AT 16:29:22           LAST UPDATED ON 07/27/83 AT 15:21:09 BY SMS VERSION 5.27
LANGUAGE:              USER INFORMATION:

MODSETS PREVIOUSLY APPLIED TO SPL: F255B    F256A    F256B    F257A    F261A    F261C    F263B    F268A    F269B
     F275A     F276A     F282A     F283A     F283B     F284A     F285C     F0111D    F0114A    F0114D    F0515F    F0521C
     F0521G    F0529C    F0529D    F0530E    F0714A    F0714B    F0731C    F0013C    F0014B    F0022A    F0027A    F0027B
     F0027C    F0916B    F1009G    F1229A    F1231A    F010601C  F010601E  L0915028  L0923028  L0924028C L0411030  L072703C
     L0727030

          PROGRAM OCPLOT(ISW)                                                           OCPLOT        1
          PARAMETER IS=40,JS=30,KS=10                                                   L0915028      3
         . , IS2=IS + 2                                                                 F261A         3
          DIMENSION D1(IS),D2(JS),D3(IS,JS,KS),D5(IS,JS),D6(IS,JS)                      F275A         3
         . , D13(IS),D14(JS),ZT(IS,JS),ZS(IS,JS),ZC(IS,JS)                              F283A         3
         . , ZBOT(IS,JS)                                                                F0022A        3
          LOGICAL READT,READS,PAPERP/F/                                                 F0521C        3
         . , TOPLAY                                                                     F0530E        4
          REAL*4 LMIN,LMAX,LDIF                                                         F0529D        3
          INTEGER PC(60),IB(60)                                                         F0515F        3
          CALL RSSTOP                                                                   OCPLOT        4
          READ (3) KF,DELT,PC,IB,NSTEP,XLSPCT,XPSPCT                                    L0923028      3
          IRUN=KF / 1000                                                                F010601E      3
          KF=MOD(KF,1000)                                                               F010601E      4
          NSTEPO=NSTEP / 10000                                                          L0411030      3
          NSTEP =MOD(NSTEP,10000)                                                       L0411030      4
          NSTEPO=NSTEPO + 1                                                             L0411030      5
          READ (2,910) LMIN,LMAX,PMIN,PMAX,IF,JF,XLSPAC,XPSPAC                          L0923028      5
      910 FORMAT (4F7.1,2I7,1P2E10.3)                                                   L0923028      6
          IF (XLSPCT.NE.XLSPAC .OR. XPSPCT.NE.XPSPAC) CALL STOPP                        L0923028      7
         .(" MESH SPACING INCONSISTENT BETWEEN SIGMA RUN AND ZDATA $")                  L0923028      8
    *                                                                                   F0714A        3
    *     SPLIT UP HERE DEPENDING ON DATA TYPE                                          F0714A        4
    *                                                                                   F0714A        5
          IF (ISW.GT.10) GO TO 1000                                                     F0714A        6
    *                                                                                   F0714A        7
    *                                                                                   F0714A        8
          IFP1=IF + 1                                                                   OCPLOT        8
          JFP1=JF + 1                                                                   OCPLOT        9
          IF (IFP1.GT.IS .OR. JFP1.GT.JS .OR. KF.GT.KS)                                 F255B        11
         .CALL STOPP("ARRAYS ARE TOO SMALL IN OCPLOT$")                                 F255B        12
          NPLOT=0                                                                       F269B         3
    *                                                                                   F261A         8
          DO 50 I=1,IF                                                                  F261A         9
       50 READ(2,920) (ZBOT(I,J),J=1,JF)                                                F0022A        5
      920 FORMAT (10F7.1)                                                               F1009G        3
          DO 60 I=2,IF                                                                  F261A        12
          DO 60 J=1,JF                                                                  F261A        13
       60 D6(I,J)=ZBOT(I,J) + ZBOT(I-1,J)                                               F0022A        7
          DO 70 I=2,IF                                                                  F261A        15
          DO 70 J=2,JF                                                                  F261A        16
       70 D5(I,J)=(D6(I,J) + D6(I,J-1)) * .25                                           F261A        17
          DO 80 J=2,JF                                                                  F261A        18
          D5(1,J)=D5(3,J)                                                               F0114D        4
       80 D5(IFP1,J)=D5(IF-1,J)                                                         F0114D        5
          DO 90 I=1,IFP1                                                                F261A        21
```

```
      DS(I,1)=DS(I,3)                                                F01140    7
   90 DS(I,JFP1)=DS(I,JF-1)                                          F01140    8
*                                                                    F0529C    3
*     SET UP THE MESH                                                F0529C    4
*       (NOTE:  THIS IS THE MESH LOCATED AT THE TS POINTS, AND IS    F0529C    5
*                 NOT THE MESH FROM THE CODE.)                       F0529C    6
*                                                                    F0529C    7
      CONST=1./57.29578                                              F0529C    8
      PRINT 950                                                      L0923028 10
      CALL MESHST(D1,IFP1,LMIN,LMAX,KLSPAC,1)                        L0923028 11
      PRINT 951                                                      L0923028 12
      CALL MESHST(D2,JFP1,PMIN,PMAX,KPSPAC,1)                        L0923028 13
      DO 93 I=1,IFP1                                                 F0529C   17
   93 D1(I)=D1(I) * CONST                                            L0923028 15
      DO 96 J=1,JFP1                                                 F0529C   19
   96 D2(J)=D2(J) * CONST                                            L092402C  3
*                                                                    L0923028 18
  950 FORMAT (20X,"LONGITUDE MESH:")                                 L0923028 19
  951 FORMAT (20X," LATITUDE MESH:")                                 L0923028 20
      IF (ISW.NE.1 .AND. ISW.NE.2 .AND. ISW.NE.3)                    F0521C    5
     .CALL STOPP("BAD VALUE OF LOAD TIME PARAMTER IN OCPLOTS")       F2558    14
      IF (ISW.EQ.1 .OR. ISW.EQ.2) CALL COMPRS                        F0731C    3
      IF (ISW.EQ.3) PAPERP=.TRUE.                                    F0521C    7
      ISTP=0                                                         F261A    25
      CALL SURFT(D6,D13,D14,IFP1,JFP1,LMIN,LMAX,PMIN,PMAX,ISTP       F0114A    3
     . , NPLOT,D5,IS,JS,IRUN,KLSPAC,KPSPAC)                          L0923028 22
      CALL STSLCE(IFP1,JFP1,KF,D1,LMIN,LMAX,D2,PMIN,PMAX             F276A     3
     . , D5,IS,JS,DELT,IRUN)                                         F010601E  8
      CALL SETVP(D1,IFP1,D2,JFP1,D5,IS,JS,KF,DELT,IRUN)              F010601E 10
      CALL HSLICE(D3,IF,JF,KF,ZBOT,D5,IS,JS,D1,IFP1,D2,JFP1          F0022A    9
     . , "    (P)RESSURE    ",ISTP,NPLOT,DELT,PAPERP,IRUN)           F010601E 12
*                                                                    F2558    18
*     IN THE CALLS TO SLICE THE LAST 6 ARGUMENTS ARE:  PLOT NUMBER,  F203A     6
*       DELT, WHICH DERIVATIVE=0 AT THE BOTTOM, STEP NUMBER, DATA    F203A     7
*       TYPE,  AND THE OPERATION TO BE PERFORMED.                    F203A     8
*                                                                    F2558    19
      TOPLAY=.TRUE.                                                  F0530E    6
      DO  200 ISTP=NSTEP0,NSTEP                                      L0411030  7
      READT=.FALSE.                                                  F203A    10
      READS=.FALSE.                                                  F203A    11
*                                                                    F2558    22
      IF (PC(13).EQ.0) GO TO 100                                     F2558    23
      IF (MOD(ISTP,PC(13)).EQ.0 .AND. ISTP.GE.IB(13))                F2558    24
     .CALL SURFT(D6,D13,D14,IFP1,JFP1,LMIN,LMAX,PMIN,PMAX,ISTP,NPLOT F0114A    6
     . , D5,IS,JS,IRUN,KLSPAC,KPSPAC)                                L0923028 24
  100 CONTINUE                                                       F2558    26
*                                                                    F2558    27
      IF (PC(17).EQ.0) GO TO 120                                     F2558    28
      IF (MOD(ISTP,PC(17)).NE.0 .OR. ISTP.LT.IB(17)) GO TO 120       F203A    13
      CALL SLICE(D3,IFP1,JFP1,KF,D1,D2,ZT,"  (T)EMPERATURE ",NPLOT   F203A    14
     . , DELT,1,ISTP,1,1,PAPERP,TOPLAY)                              F0530E    8
      READT=.TRUE.                                                   F203A    16
      CALL VPRFL(D3,IFP1,JFP1,KF,"  (T)EMPERATURE "                  F205C     5
     . , NPLOT,DELT,ISTP,D5,IS,JS,PAPERP)                            F0521C   14
  120 CONTINUE                                                       F2558    33
*                                                                    F2558    34
      IF (PC(18).EQ.0) GO TO 130                                     F2558    35
      IF (MOD(ISTP,PC(18)).NE.0 .OR. ISTP.LT.IB(18)) GO TO 130       F203A    18
```

```
      CALL SLICE(D3,IFP1,JFP1,KF,D1,D2,ZS,'    (S)ALINITY    ',NPLOT       F203A      19
     . , DELT,1,ISTP,2,1,PAPERP,TOPLAY)                                    F0530E     10
      READS=.TRUE.                                                         F203A      21
      CALL VPRFL(D3,IFP1,JFP1,KF,'    (S)ALINITY    '                      F205C       8
     . , NPLOT,DELT,ISTP,DS,IS,JS,PAPERP)                                  F0521C     10
  130 CONTINUE                                                             F2550      30
                                                                           F2550      40
      IF (PC(12).EQ.0) GO TO 140                                           F2550      41
      IF (MOD(ISTP,PC(12)).NE.0 .OR. ISTP.LT.ID(12)) GO TO 140             F203B       3
      IF (.NOT.READT)                                                      F203A      24
     .CALL SLICE(D3,IFP1,JFP1,KF,D1,D2,ZT,0,0,DELT,1,ISTP,1,2              F0521C     20
     . , PAPERP,TOPLAY)                                                    F0530E     12
      IF (.NOT.READS)                                                      F203A      26
     .CALL SLICE(D3,IFP1,JFP1,KF,D1,D2,ZS,0,0,DELT,1,ISTP,2,2              F0521C     23
     . , PAPERP,TOPLAY)                                                    F0530E     14
      CALL SSPEED(ZT,ZS,IFP1,JFP1,ZC)                                      F204A       3
      CALL SLICE(D3,IFP1,JFP1,KF,D1,D2,ZC,'(S)OUND (S)PEED '               F203A      29
     . , NPLOT,DELT,2,ISTP,3,3,PAPERP,TOPLAY)                              F0530E     16
  140 CONTINUE                                                             F2550      45
*                                                                          F0013C      3
      IF (PC(31).EQ.0) GO TO 150                                           F0013C      4
      IF (MOD(ISTP,PC(31)).NE.0 .OR. ISTP.LT.ID(31)) GO TO 150             F0013C      5
      CALL HSLICE(D3,IF,JF,KF,Z00T,DS,IS,JF,D1,IFP1,D2,JFP1                F0022A     11
     . , '    (P)RESSURE    ',ISTP,NPLOT,DELT,PAPERP,IRUN)                 F010401E   16
  150 CONTINUE                                                             F0013C      8
*                                                                          F2550      46
  200 CONTINUE                                                             F2550      47
      IF (.NOT.PAPERP) CALL DONEPL                                         F0521G      3
      STOP                                                                 OCPLOT     11
*                                                                          F0714A     12
*     FOR U-V TYPE DATA                                                    F0714A     13
*                                                                          F0714A     14
 1000 CONTINUE                                                             F0714A     15
      ISW=MOD(ISW,10)                                                      F0714A     16
*                                                                          F0714A     17
*                                                                          F0714A     18
      IF (IF.GT.IS .OR. JF.GT.JS .OR. KF.GT.KS)                            F0714A     19
     .CALL STOPP('ARRAYS ARE TOO SMALL IN OCPLOT$')                        F0714A     20
      NPLOT=0                                                              F0714A     21
*                                                                          F0714A     22
      DO 1050 I=1,IF                                                       F0714A     23
 1050 READ (2,920) (US(I,J),J=1,JF)                                        F0714A     24
*                                                                          F0714A     25
*     SET UP THE MESH                                                      F0714A     26
*                                                                          F0714A     27
      CONST=1. / 57.29578                                                  F0714A     28
      ALMIN=LMIN * CONST                                                   F0714A     29
      ALMAX=LMAX * CONST                                                   F0714A     30
      APMIN=PMIN * CONST                                                   F0714A     31
      APMAX=PMAX * CONST                                                   F0714A     32
      PDIF=APMAX - APMIN                                                   F0714A     33
      LDIF=ALMAX - ALMIN                                                   F0714A     34
      XIFMI=1. / (IF - 2.)                                                 F0714A     35
      XJFMI=1. / (JF - 2.)                                                 F0714A     36
      DO 1095 I=1,IF                                                       F0714A     37
 1095 D1(I)=ALMIN + LDIF * (FLOAT(I)-1.5) * XIFMI                          F0714A     38
      DO 1096 J=1,JF                                                       F0714A     39
 1096 D2(J)=APMIN + PDIF * (FLOAT(J)-1.5) * XJFMI                          F0714A     40
```

```
      IF (ISW.NE.1 .AND. ISW.NE.2 .AND. ISW.NE.3)                          F0714A    41
     .CALL STOPP('BAD VALUE OF LOAD TIME PARAMTER IN DCPLOTS')             F0714A    42
      IF (ISW.EQ.1 .OR. ISW.EQ.2) CALL COMPRS                              F0731C     5
      IF (ISW.EQ.3) PAPERP=.TRUE.                                          F0714A    45
      CALL SURFT(D6,D13,D14,IFP1,JFP1,LMIN,LMAX,PMIN,PMAX,-1               F0916B     3
     . , NPLOT,D5,IS,JS,IRUN,XLSPAC,XPSPAC)                                L0923828  26
      CALL STSLCE(IF,JF,KF,D1,LMIN,LMAX,D2,PMIN,PMAX                       F0714A    46
     . , D5,IS,JS,DELT,IRUN)                                               F010601E  20
      CALL SETVP(D3,IF,D2,JF,D5,IS,JS,KF,DELT,IRUN)                        F010601E  22
      ISTP=0                                                               F0027C     3
      CALL HSLAB(D3,IF,JF,D5,IS,JS,D1,D2                                   F0027C     4
     . , '             FLUX   ',ISTP,NPLOT,PAPERP,IRUN)                    F010601E  24
*                                                                          F0714A    49
*     IN THE CALLS TO SLICE THE LAST 6 ARGUMENTS ARE:  PLOT NUMBER,        F0714A    50
*       DELT, WHICH DERIVATIVE=0 AT THE BOTTOM, STEP NUMBER, DATA          F0714A    51
*       TYPE.  AND THE OPERATION TO BE PERFORMED.                          F0714A    52
*                                                                          F0714A    53
      TOPLAT=.FALSE.                                                       F0714A    54
      DO 1200 ISTP=NSTEP0,NSTEP                                            L0411030   9
*                                                                          F0714A    56
      IF (PC(28).EQ.0) GO TO 1120                                          F0714A    57
      IF (MOD(ISTP,PC(28)).NE.0 .OR. ISTP.LT.IB(28)) GO TO 1120            F0714B     4
      CALL SLICE(D3,IF,JF,KF,D1,D2,IT,'  (U) VELOCITY   ',NPLOT            F0714A    59
     . , DELT,1,ISTP,1,1,PAPERP,TOPLAT)                                    F0714A    60
      CALL VPRFL(D3,IF,JF,KF,'  (U) VELOCITY   '                           F0714A    61
     . , NPLOT,DELT,ISTP,D5,IS,JS,PAPERP)                                  F0714A    62
 1120 CONTINUE                                                             F0714A    63
*                                                                          F0714A    64
      IF (PC(29).EQ.0) GO TO 1130                                          F0714A    65
      IF (MOD(ISTP,PC(29)).NE.0 .OR. ISTP.LT.IB(29)) GO TO 1130            F0714B     6
      CALL SLICE(D3,IF,JF,KF,D1,D2,IS,' (V) VELOCITY    ',NPLOT            F0714A    67
     . , DELT,1,ISTP,2,1,PAPERP,TOPLAT)                                    F0714A    68
      CALL VPRFL(D3,IF,JF,KF,' (V) VELOCITY    '                           F0714A    69
     . , NPLOT,DELT,ISTP,D5,IS,JS,PAPERP)                                  F0714A    70
 1130 CONTINUE                                                             F0714A    71
*                                                                          F0027A     3
      IF (PC(33).EQ.0) GO TO 1140                                          F0027A     4
      IF (MOD(ISTP,PC(33)).NE.0 .OR. ISTP.LT.IB(33)) GO TO 1140            F0027A     5
      CALL HSLAB(D3,IF,JF,D5,IS,JS,D1,D2                                   F0027B     3
     . , '             FLUX   ',ISTP,NPLOT,PAPERP,IRUN)                    F010601E  26
 1140 CONTINUE                                                             L0727030   3
*                                                                          L072703C   3
      IF (PC(30).EQ.0) GO TO 1150                                          L072703C   4
      IF (MOD(ISTP,PC(30)).NE.0 .OR. ISTP.LT.IB(30)) GO TO 1150            L072703C   5
      CALL SLICE(D3,IF,JF,KF,D1,D2,IT,'(U) GEOSTROPHIC ',NPLOT             L072703C   6
     . , DELT,1,ISTP,1,1,PAPERP,TOPLAT)                                    L072703C   7
      CALL VPRFL(D3,IF,JF,KF,'(U) GEOSTROPHIC '                            L072703C   8
     . , NPLOT,DELT,ISTP,D5,IS,JS,PAPERP)                                  L072703C   9
 1150 CONTINUE                                                             L072703C  10
*                                                                          L072703C  11
      IF (PC(39).EQ.0) GO TO 1160                                          L072703C  12
      IF (MOD(ISTP,PC(39)).NE.0 .OR. ISTP.LT.IB(39)) GO TO 1160            L072703C  13
      CALL SLICE(D3,IF,JF,KF,D1,D2,IS,'(V) GEOSTROPHIC ',NPLOT             L072703C  14
     . , DELT,1,ISTP,2,1,PAPERP,TOPLAT)                                    L072703C  15
      CALL VPRFL(D3,IF,JF,KF,'(V) GEOSTROPHIC '                            L072703C  16
     . , NPLOT,DELT,ISTP,D5,IS,JS,PAPERP)                                  L072703C  17
 1160 CONTINUE                                                             L072703C  18
*                                                                          L072703C  19
```

```
SNS VERSION 5.27    LISTING OF DECK    O C P L O T                                   01/12/84        PAGE 5

          *                                                                          F0714A     73
       1200 CONTINUE                                                                 F0714A     74
            IF (.NOT.PAPERP) CALL DONEPL                                             F0714A     75
            STOP                                                                     F0714A     76
            END                                                                      OCPLOT     12

        225 ACTIVE LINE(S)          119 INACTIVE LINE(S)
OE SETUP
```

```
*** SRS109 : DECK SETVP    IS EDITED FROM SPL       FILE
CREATED ON 10/12/79 AT 17:22:43         LAST UPDATED ON 04/18/83 AT 11:31:43 BY SRS VERSION 5.27
LANGUAGE:               USER INFORMATION:

MODSETS PREVIOUSLY APPLIED TO SPL: F2008     F208C     F209A     F0516A     F1113D     F010601E  L041003D

      SUBROUTINE SETVP(L,IFP1,P,JFP1,Z,ISZ,JSZ,KF,DELT,IRUND)                 F010601E     3
      IMPLICIT REAL*4 (L)                                                     SETVP        2
*CALL CONVPR                                                                  SETVP        3
      DIMENSION L(IFP1),P(JFP1),Z(ISZ,JSZ)                                    SETVP        4
      LOGICAL NOLOGO,NOCHK,NINPUT                                             SETVP        5
      NAMELIST/PLTDT3/SPOTL,SPOTP,NOLOGO,NOCHK,GRCE,HMAX                      SETVP        6
     . , XTIT,YTIT,SIZE,NSPOT                                                 SETVP        7
     . , NOGRID                                                               F1113D       3
      IRUN=IRUND                                                              F010601E     5
      DO 50 N=1,10                                                            SETVP        8
      SPOTL(N)=0.                                                             SETVP        9
   50 SPOTP(N)=0.                                                             SETVP       10
      NSPOT=0                                                                 SETVP       11
      NOLOGO=.TRUE.                                                           SETVP       12
      NOCHK=.TRUE.                                                            SETVP       13
      GRCE=0.                                                                 SETVP       14
      HMAX=0.                                                                 SETVP       15
      XTIT=5.                                                                 L041003D     3
      YTIT=0.                                                                 L041003D     4
      SIZE=1.25                                                               SETVP       18
      NOGRID=.FALSE.                                                          F1113D       5
      READ(5,PLTDT3)                                                          SETVP       19
      IF (NSPOT.EQ.0) RETURN                                                  F0516A       3
      NINPUT=.FALSE.                                                          SETVP       20
      IF (HMAX.NE.0.) NINPUT=.TRUE.                                           SETVP       21
      CONST=57.29578                                                          F2008        3
      CONSTI=1. / CONST                                                       F2008        4
      DO 60 N=1,NSPOT                                                         F2008        5
      SPOTL(N)=SPOTL(N) * CONSTI                                              F2008        6
   60 SPOTP(N)=SPOTP(N) * CONSTI                                              F2008        7
*                                                                             SETVP       22
*     FIND WHERE THE SPOT(S) LIE ON THE MESH                                  SETVP       23
*                                                                             SETVP       24
      DELTI=1. /  DELT                                                        SETVP       25
      IF=IFP1 - 1                                                             SETVP       26
      JF=JFP1 - 1                                                             SETVP       27
      KFM1=KF - 1                                                             SETVP       28
      DO 150 N=1,NSPOT                                                        SETVP       29
      DO 110 I=1,IF                                                           SETVP       30
      IF (SPOTL(N).GE.L(I) .AND. SPOTL(N).LE.L(I+1)) GO TO 115                SETVP       31
  110 CONTINUE                                                                SETVP       32
      CALL STOPP("SPOTL DOES NOT LIE ON MESH IN SETVP$")                      SETVP       33
  115 IS(N)=I                                                                 SETVP       34
      PARTI(N)=(SPOTL(N)-L(I)) / (L(I+1)-L(I))                                SETVP       35
      DO 120 J=1,JF                                                           SETVP       36
      IF (SPOTP(N).GE.P(J) .AND. SPOTP(N).LE.P(J+1)) GO TO 125                SETVP       37
  120 CONTINUE                                                                SETVP       38
      CALL STOPP("SPOTP DOES NOT LIE ON MESH IN SETVP$")                      SETVP       39
  125 JS(N)=J                                                                 SETVP       40
      PARTJ(N)=(SPOTP(N)-P(J)) / (P(J+1)-P(J))                                SETVP       41
```

```
SMS VERSION 5.27    LISTING OF DECK    S E T V P                                  01/12/84          PAGE 2
        150 CONTINUE                                                               SETVP   42
            DO 200 K=1,KFM1                                                        SETVP   43
            KK=K + .5                                                              SETVP   44
        200 SIG(K)=(KK - 2.) / (KF - 2.)                                           SETVP   45
     *                                                                             SETVP   46
            IF (MINPUT) GO TO 230                                                  SETVP   47
            HMAX=AMAX1(Z(I,J),Z(I+1,J))                                            F280C    3
            HMAX=AMAX1(HMAX,Z(I,J+1))                                              F280C    4
            HMAX=AMAX1(HMAX,Z(I+1,J+1))                                            F280C    5
            HMAX=-HMAX                                                             F280C    6
        230 CONTINUE                                                               SETVP   51
            HOD=-HMAX + DELTI                                                      SETVP   52
            DO 250 KK=1,20                                                         SETVP   53
            SIGMA=FLOAT(KK-1) / 19.                                                SETVP   54
        250 THESH(KK)=-(((1.+HOD)**SIGMA)-1.) + DELT                               SETVP   55
     *                                                                             SETVP   56
     *                                                                             SETVP   57
            RETURN                                                                 SETVP   58
            END                                                                    SETVP   59

          69 ACTIVE LINE(S)            0 INACTIVE LINE(S)
```

IE VPRFL

```
*** SMS109 : DECK VPRFL     IS EDITED FROM SPL       FILE
CREATED ON 10/12/79 AT 17:22:50          LAST UPDATED ON 04/18/83 AT 09:32:26 BY SMS VERSION 5.27
LANGUAGE:            USER INFORMATION:

MODSETS PREVIOUSLY APPLIED TO SPL: F200A     F200D     F200E     F200F     F0521C    F1113A    F1113B    F1229C    F010601E
     F010801E  L041803C

      SUBROUTINE VPRFL(T,IFP1,JFP1,KF,LABEL                                  F0521C      3
     . , NPLOT,DELT,IC,Z,ISZ,JSZ,PAPERP)                                     F0521C      4
      IMPLICIT REAL*4 (L)                                                    VPRFL       2
*CALL COMVPR                                                                 VPRFL       3
      DIMENSION T(IFP1,JFP1,KF),T1(KSIZE),T2(KSIZE),Z1(KSIZE),Z2(KSIZE)      F1229C      3
     . , T4(20,4),TW(KSIZE),LABEL(4),LTIT(20),Z(ISZ,JSZ)                     F010601E    3
     . , NTIT(20),IRLAB(10)                                                  F010801E    3
      LOGICAL NOCHK,NOLOGO,PAPERP                                            F0521C      8
*                                                                            F1229C      6
      IF (KF.GT.KSIZE)                                                       F1229C      7
     .CALL STOPP("ARRAYS TOO SMALL IN VPRFL*")                               F1229C      8
*                                                                            F1229C      9
      KFM1=KF - 1                                                            VPRFL       7
      DELTI=1. / DELT                                                        VPRFL       8
      CONST=57.29578                                                         F200A       3
*                                                                            VPRFL       9
      IF (NSPOT.EQ.0) RETURN                                                 VPRFL      10
      DO 500 NS=1,NSPOT                                                      VPRFL      11
      DO 160 N=1,4                                                           VPRFL      12
      I=IS(NS) + N - 1 - (N/3) * 2                                           VPRFL      13
      J=JS(NS) + (N/3)                                                       VPRFL      14
      DO 110 KK=1,KFM1                                                       VPRFL      15
  110 TW(KK)=T(I,J,KK)                                                       VPRFL      16
      CALL CUSPLN(SIG,TW,KFM1,T1,T2,Z1,Z2,1,1.,0)                            VPRFL      17
      LOGZBI=1. / ALOG(1. + Z(I,J) * DELTI)                                  VPRFL      18
*                                                                            F200D       3
*     TRESH IS NEGATIVE                                                      F200D       4
      DO 120 KK=1,20                                                         VPRFL      19
      TSIG=ALOG(1. - TRESH(KK) * DELTI) * LOGZBI                             F200D       6
  120 T4(KK,N)=SPLINE(TSIG,SIG,TW,T2,KFM1)                                   VPRFL      21
  160 CONTINUE                                                               VPRFL      22
*                                                                            VPRFL      23
      DO 200 KK=1,20                                                         VPRFL      24
      TW(KK)=T4(KK,1)*(1.-PARTI(NS))*(1.-PARTJ(NS))                          VPRFL      25
     .       +T4(KK,2)*(   PARTI(NS))*(1.-PARTJ(NS))                         VPRFL      26
     .       +T4(KK,3)*(1.-PARTI(NS))*(   PARTJ(NS))                         VPRFL      27
     .       +T4(KK,4)*(   PARTI(NS))*(   PARTJ(NS))                         VPRFL      28
  200 CONTINUE                                                               VPRFL      29
*                                                                            VPRFL      30
      IF (PAPERP) GO TO 240                                                  F0521C     10
      NPLOT=NPLOT + 1                                                        VPRFL      31
      CALL BGNPL(NPLOT)                                                      VPRFL      32
      CALL BASALF("L/CSTAN")                                                 VPRFL      33
      CALL MIXALF("STAND")                                                   VPRFL      34
      CALL BLOWUP(SIZE)                                                      VPRFL      35
      IF (NOCHK) CALL NOCHEK                                                 VPRFL      36
      CALL GRACE(GRLE)                                                       VPRFL      37
      ENCODE(52,950,LTIT) LABEL                                              VPRFL      38
  950 FORMAT (4A4,"$")                                                       VPRFL      39
```

```
SAS VERSION 5.27    LISTING OF DECK    V P R F L                                          01/12/84          PAGE 2

           CALL TITLE(IN ,1,LTIT,100,"(D)EPTHS",100,HTIT,YTIT)                            L041003C        3
           ENCODE(52,951,LTIT) LABEL,IC                                                   VPRFL          42
       951 FORMAT (4A4,"AT (T)IME (S)TEP ",I3,"$")                                        VPRFL          43
           CALL HEADIN(LTIT,-100,4,2)                                                     F200F           5
           LOUT=SPOTL(NS) + CONST                                                         F200B           5
           ROUT=SPOTP(NS) + CONST                                                         F200B           6
           ENCODE(52,952,LTIT) LOUT,POUT                                                  F200B           7
       952 FORMAT (ACC)AT LOCATION ",F6.1,"(E) : ",F6.1,"(N))$")                          VPRFL          46
           CALL HEADIN(LTIT,100,2,2)                                                      VPRFL          47
       240 CONTINUE                                                                       F0521C         12
           TMAX=-50.                                                                      VPRFL          48
           TMIN=100.                                                                      VPRFL          49
           DO 250 KK=1,20                                                                 VPRFL          50
           IF (TMAX.LT.TW(KK)) TMAX=TW(KK)                                                VPRFL          51
       250 IF (TMIN.GT.TW(KK)) TMIN=TW(KK)                                                VPRFL          52
           IF (PAPERP) GO TO 400                                                          F0521C         14
           CALL GRAF(TMIN,"SCALE",TMAX,HMAX,"SCALE",0.)                                   L041003C        5
           CALL CURVE(TW,THESH,20,0)                                                      L041003C        6
           ENCODE(40,953,IRLAB) IRUN                                                      F010001E        8
       953 FORMAT ("(C)X(R)UN #",I4,"()$")                                                F010001E        9
           CALL MESSAG(IRLAB,100,.25,-.8)                                                 F010001E       10
           IF (.NOT.NOLOGO) CALL LOGO                                                     VPRFL          55
           CALL ENDPL(NPLOT)                                                              VPRFL          56
           GO TO 500                                                                      F0521C         16
      *                                                                                   F0521C         17
      *    PAPER PLOT                                                                     F0521C         18
      *                                                                                   F0521C         19
       400 CONTINUE                                                                       F0521C         20
           ENCODE(80,970,HTIT)LABEL,IC,IRUN                                               F010601E        8
       970 FORMAT ("** ",4A4," AT TIME STEP ",I4,"   (RUN #",I3,")$")                     F010601E        9
           CALL BSCALE(TMIN,TMAX,HMAX,0.)                                                 L041003C        8
           IF (NOGRID) CALL XGRID                                                         F11130          3
           CALL BPLOT(TW,THESH,20,HTIT)                                                   L041003C       10
           LOUT=SPOTL(NS) + CONST                                                         F0521C         25
           POUT=SPOTP(NS) + CONST                                                         F0521C         26
           PRINT 971,LOUT,POUT                                                            F0521C         27
       971 FORMAT (/,"0 AT LOCATION ",F6.1,"(E) : ",F6.1,"(N)")                           F0521C         28
       500 CONTINUE                                                                       VPRFL          57
           RETURN                                                                         VPRFL          58
           END                                                                            VPRFL          59

       89 ACTIVE LINE(S)              23 INACTIVE LINE(S)
```

Appendix D

Main Program for UVPLOT

```
*** SMS109 : DECK MAIN      IS EDITED FROM SPL       FILE
CREATED ON 01/12/84 AT 11:31:06         LAST UPDATED ON 01/12/84 AT 11:33:14 BY SMS VERSION 5.27
LANGUAGE:              USER INFORMATION:

         *PARAMETER IS=40,JS=30,KS=10
         * , IS2=IS * 2
          DIMENSION D1(IS),D2(JS),D3(IS,JS,KS),DS(IS,JS),FLUX(32,24)
         * ,ZBOT(IS,JS),LABEL1(4),LABEL2(4)
         *        ,LABEL3(4),LABEL4(4)
          REAL*4 LMIN,LMAX
          INTEGER PC(60),IB(60)
          CALL RSTOP
          ENCODE(16,210,LABEL1)
          ENCODE(16,220,LABEL2)
          ENCODE(16,230,LABEL3)
          ENCODE(16,240,LABEL4)
   210    FORMAT("   U VELOCITY    ")
   220    FORMAT("   V VELOCITY    ")
   230    FORMAT("  WG VELOCITY    ")
   240    FORMAT("  WG VELOCITY    ")
          READ (3) KF,DELT,PC,IB,NSTEP,XLSPCT,XPSPCT
          IRUN=KF / 1000
          KF=MOD(KF,1000)
          NSTEP0=NSTEP/10000
          NSTEP=MOD(NSTEP,10000)
          NSTEP0=NSTEP0+1
          READ (2,910) LMIN,LMAX,PMIN,PMAX,IF,JF,XLSPAC,XPSPAC
      910 FORMAT (4F7.1,2I7,1P2E10.3)
          WRITE(10) NSTEP,IRUN,LMIN,LMAX,PMIN,PMAX,IF,JF
          IF (XLSPCT.NE.XLSPAC .OR. XPSPCT.NE.XPSPAC) CALL STOPP
         *(" MESH SPACING INCONSISTENT BETWEEN SIGMA RUN AND ZDATA $")
C
C
          IFP1=IF + 1
          JFP1=JF + 1
          IF (IFP1.GT.IS .OR. JFP1.GT.JS .OR. KF.GT.KS)
         *CALL STOPP("ARRAYS ARE TOO SMALL IN OCPLOTS")
C
          DO 50 I=1,IF
       50 READ(2,920) (ZBOT(I,J),J=1,JF)
      920 FORMAT (10F7.1)
          DO 60 I=1,IF
          DO 60 J=1,JF
   60     DS(I,J)=ZBOT(I,J)
C
C         SET UP THE MESH
C           (NOTE:  THIS IS THE MESH LOCATED AT THE TS POINTS, AND IS
C                     NOT THE MESH FROM THE CODE.)
C
          CONST=1. / 57.29578
          PRINT 950
          CALL MESHST(D1,IF,LMIN,LMAX,XLSPAC,0)
          PRINT 951
          CALL MESHST(D2,JF,PMIN,PMAX,XPSPAC,0)
          DO 93 I=1,IF
       93 U1(I)=D1(I) * CONST
```

```
          DO 96 J=1,JF
       96 D2(J)=D2(J) * CONST
    C
      950 FORMAT (20X,'LONGITUDE MESH:')
      951 FORMAT (20X,' LATITUDE MESH:')
          ISTP=0
          CALL MLEVEL(D3,IF,JF,KF,IF,JF,ZBOT,DS,IS,JS,D1,IF,D2,
         1        JF,LABEL1,ISTP,DELT,IRUN)
    C
          DO 200 ISTP=NSTEP0,NSTEP
    C
    C
          IF (PC(28).EQ.0) GO TO 120
          IF (MOD(ISTP,PC(28)).NE.0 .OR. ISTP.LT.IB(28)) GO TO 120
          CALL MLEVEL(D3,IF,JF,KF,IF,JF,ZBOT,DS,IS,JS,D1,IF,D2,
         1        JF,LABEL1,ISTP,DELT,IRUN)
      120 CONTINUE
    C
          IF (PC(29).EQ.0) GO TO 130
          IF (MOD(ISTP,PC(29)).NE.0 .OR. ISTP.LT.IB(29)) GO TO 130
          CALL MLEVEL(D3,IF,JF,KF,IF,JF,ZBOT,DS,IS,JS,D1,IF,D2,
         1       JF,LABEL2,ISTP,DELT,IRUN)
      130 CONTINUE
          IF(PC(33).EQ.0) GO TO 140
          IF(MOD(ISTP,PC(33)).NE.0 .OR. ISTP.LT.IB(33)) GO TO 140
          READ(3) FLUX
    140   CONTINUE
          IF(PC(30).EQ.0) GO TO 150
          IF(MOD(ISTP,PC(30)).NE.0 .OR. ISTP.LT.IB(30))  GO TO 150
          CALL MLEVEL(D3,IF,JF,KF,IF,JF,ZBOT,DS,IS,JS,D1,IF,D2,JF,
         *          LABEL3,ISTP,DELT,IRUN)
    150   CONTINUE
          IF(PC(39).EQ.0) GO TO 160
          IF(MOD(ISTP,PC(39)).NE.0 .OR. ISTP.LT.IB(39)) GO TO 160
          CALL MLEVEL(D3,IF,JF,KF,IF,JF,ZBOT,DS,IS,JS,D1,IF,D2,JF,
         *          LABEL4,ISTP,DELT,IRUN)
    160   CONTINUE
    C
    C
      200 CONTINUE
          STOP
          END
```

94 ACTIVE LINE(S)   0 INACTIVE LINE(S)

*** SMS167 : PDS DIRECTORY FOR NEWSPL   FILE SUCCESSFULLY UPDATED FOR DECK MAIN

Appendix E

Listings Required for Mediterranean Tests

CREATED ON 02/15/84 AT 11:19:03   LAST UPDATED ON 02/15/84 AT 11:19:03 BY SMS VERSION 5.27
LANGUAGE: USER INFORMATION:

```
/ FXL LIBRARY=6/OBJLIB
      PROGRAM BOTTOM
*
*
*                   ********* SOLUTION DOMAIN **********
*                   ************************************
*                   ************************************
*                   ************************************
*
*      :       :       :       :       :       :       :
*     T(1)    T(2)                            T(I)     T(IFP1)
*
*          :       :       :       :       :       :
*         Z(1)    Z(2)                           Z(IF)
*
*
*     THE ACTUAL SOLUTION DOMAIN HAS IF-2 INTERVALS AND RUNS FROM
*       T(2) TO T(IF).   (T(1) AND T(IFP1) ARE EXTRA POINTS.)
*
*     Z IS STAGGERED WRT T AND IS DEFINED FROM A HALF MESH CELL
*       TO THE LEFT OF THE DOMAIN TO A HALF MESH CELL ON THE RIGHT.
*
*     WE PASS TO ROUTINE INTRP ACTUAL LOCATIONS IN DEGREES, WHERE
*       WE NEED THE DEPTH (Z).   THAT IS AN AREA WHICH IS LARGER
*       THAN THE DOMAIN BY A HALF MESH CELL ON ALL SIDES.   SINCE
*       THE DATA BASE POINTS ARE LOCATED ON THE HALF DEGREE.   IT
*       IS BETTER TO PASS VALUES LOCATED AT N + 1/2.
*
*     SUBROUTINE MESHST IS IN THE MAIN SIGMA CODE
*
      PARAMETER PIF=83,PJF=20
      DIMENSION Z(PIF,PJF),DREC(301,91),LAM(PIF),PHI(PJF)
      REAL *4 LAM,LMAX,LMIN,LDIF
      NAMELIST/DOMAIN/LMAX,LMIN,PMAX,PMIN,IF,JF,XLSPAC,XPSPAC
*
      CALL R*STOP
      XLSPAC=0.
      XPSPAC=0.
      READ (5,DOMAIN)
      IF (IF.GT.PIF  .OR.  JF.GT.PJF)   STOP 888
      PRINT 902,LMIN,LMAX,PMIN,PMAX,IF,JF
  902 FORMAT (1H1,' THE DOMAIN RUNS FROM ',F6.1
     . , ' DEGREES EAST TO ',F6.1,' DEGREES EAST, AND '
     . ,F6.1,' DEGREES NORTH TO ',F6.1,' DEGREES NORTH.',/
     . , 1H0,' THE MESH IS (IF=) ',I3,' BY (JF=) ',I3,'.')
      IF (XLSPAC.NE.0.) PRINT 905,XLSPAC
      IF (XPSPAC.NE.0.) PRINT 906,XPSPAC
  905 FORMAT (//,' THE CENTRAL LONGITUDE SPACING IS ',1PE10.3
     . , ' DEGREES')
  906 FORMAT (//,' THE CENTRAL LATITU DE SPACING IS ',1PE10.3
     . , ' DEGREES')
      REWIND 1
      REWIND 2
      READ (1) DREC
```

```
      CALL MESHGT(LAM,IF,LMIN,LMAX,XLSPAC,0)
      CALL MESHGT(PHI,JF,PMIN,PMAX,XPSPAC,0)
      IFM1=IF - 1
      JFM1=JF - 1
      DO 200 I=2,IFM1
      DO 200 J=2,JFM1
  200 CALL INTRP(Z(I,J),LAM(I),PHI(J),DREC)
*
  100 CONTINUE
*
*     FIX UP BOUNDARIES
*
      DO 210 J=2,JFM1
      Z(2   ,J)=1.
  210 Z(IFM1,J)=1.
*
      DO 220 I=2,IFM1
  220 Z(I,2)=1.
C
      DO 230 I=15,IFM1
  230 Z(I,JFM1)=1.
*
*     DO BOUNDARIES
*
      DO 300 J=2,JFM1
      Z(1,J)=Z(2,J)
  300 Z(IF,J)=Z(IFM1,J)
      DO 350 I=1,IF
      Z(I,1)=Z(I,2)
  350 Z(I,JF)=Z(I,JFM1)
*
*     MAKE LAND INTO SHALLOW WATER
*
      DO 400 J=1,JF
      DO 400 I=1,IF
  400 IF (Z(I,J).LE.1.) Z(I,J)=1.
*
*     PUT IT UP
*
C     DO 510 II=1,2
C     LU=4*II - 2
        LU=2
      WRITE(LU,904)LMIN,LMAX,PMIN,PMAX,IF,JF,XLSPAC,XPSPAC
  904 FORMAT (4F7.1,2I7,1P2E10.3)
      DO 500 I=1,IF
  500 WRITE(LU,903)(Z(I,J),J=1,JF)
  903 FORMAT (18F7.1)
  510 CONTINUE
*
      CALL PRT(0,'DEPTHS0',Z,PIF,PJF,1,1,.TRUE.,0)
*
      STOP
      END
      SUBROUTINE INTRP (Z,LAM,PHI,D)
*
*     THE DATA IS LOCATED AT 10' INTERVALS.  THE LOWER LEFT
*       CORNER IS AT -10 DEG E. AND +30 DEG N.
*
```

```
*      IL AND JL ARE THE INDICES OF THE NEXT LOWER DATA BASE POINT.  SI
*        AND SJ ARE THE FRACTIONS THAT THE ACTUAL POINT IS ABOVE IL AND
*        JL.  THAT IS IF SI=.5, THEN THE ACTUAL POINT IS HALF WAY BETWEEN
*        IL AND IL+1.
*
       DIMENSION D(301,91)
       REAL*4 LAM,LCOR
       DATA LCOR/-10./,PCOR/30./
*
       XL=(LAM-LCOR) * 6.
       IL=IFIX(XL)
       IU=IL + 1
       IF (IL.LT.1 .OR. IU.GT.301) CALL STOPP
      .(' YOUR REGION DOES NOT LIE COMPLETELY ON DATA SET $')
       SI=XL - FLOAT(IL)
*
       XP=(PHI-PCOR) * 6.
       JL=IFIX(XP)
       JU=JL + 1
       IF (JL.LT.1 .OR. JU.GT.91) CALL STOPP
      .(' YOUR REGION DOES NOT LIE COMPLETELY ON DATA SET $')
       SJ=XP - FLOAT(JL)
*
       SIR=1. - SI
       SJR=1. - SJ
       Z=    D(IL,JL) * SIR * SJR
       Z=Z + D(IL,JU) * SI  * SJR
       Z=Z + D(IU,JL) * SIR * SJ
       Z=Z + D(IU,JU) * SI  * SJ
*
*
       RETURN
       END
/ IF E.NE.0,SKIP
/ RGET FT01F001,S/MTOPO
/ REL FT02F001,FT06F001,SEE
/ FD FT02F001,LREC=126,BKSZ=2520,FORG=PS,RCFM=FBS
/ OXT
 &DOMAIN
       LMIN=10.,  LMAX=37.,
       PMIN=32.,  PMAX=38.,
       IF=83,     JF=20,
       XLSPAC=0., XPSPAC=0.,
 &END
/ PUT FT02F001,S/ZDATA
/ PUT SEE,S/T1
/SKIP NOP
```

159 ACTIVE LINE(S)   0 INACTIVE LINE(S)

*** SMG167 : PDS DIRECTORY FOR NEWSPL   FILE SUCCESSFULLY UPDATED FOR DECK TOPOG

$OK LOCATION,ACTION=ADD

CREATED ON 02/15/84 AT 11:19:04 LAST UPDATED ON 02/15/84 AT 11:19:04 BY SMS VERSION 5.27
LANGUAGE: USER INFORMATION:

```
/ LIMIT MIN=2                                                       LOCATION
/ FXL LIBRARY=PT/OBJLIB                                             LOCATION
      DIMENSION XLAT(63,63),XLONG(63,63),D(63,63)                   LOCATION
      CALL R#STOP                                                   LOCATION
      RD=180. / (4.*ATAN(1.))                                       LOCATION
*                                                                   LOCATION
      CALL GRIDC(1,63,1,63,XLONG,XLAT,D)                            LOCATION
      XLONG=XLONG * RD                                              LOCATION
      XLAT =XLAT  * RD                                              LOCATION
*                                                                   LOCATION
      D=0.                                                          LOCATION
      DO 10 J=1,63                                                  LOCATION
      DO 10 I=1,63                                                  LOCATION
      IF (( XLAT(I,J).GE.32.  .AND.  XLAT(I,J).LE.38. ) .AND.       LOCATION
     .   (XLONG(I,J).GE.10. .AND. XLONG(I,J).LE.37.)) D(I,J)=1.     LOCATION
   10 CONTINUE                                                      LOCATION
*                                                                   LOCATION
      IS=0                                                          LOCATION
   15 IS=IS + 1                                                     LOCATION
      DO 20 J=1,63                                                  LOCATION
      IF (D(IS,J).EQ.1.) GO TO 25                                   LOCATION
   20 CONTINUE                                                      LOCATION
      GO TO 15                                                      LOCATION
   25 CONTINUE                                                      LOCATION
*                                                                   LOCATION
      IE=64                                                         LOCATION
   30 IE=IE - 1                                                     LOCATION
      DO 35 J=1,63                                                  LOCATION
      IF (D(IE,J).EQ.1.) GO TO 45                                   LOCATION
   35 CONTINUE                                                      LOCATION
      GO TO 30                                                      LOCATION
   45 CONTINUE                                                      LOCATION
*                                                                   LOCATION
      JS=0                                                          LOCATION
   50 JS=JS + 1                                                     LOCATION
      DO 55 I=IS,IE                                                 LOCATION
      IF (D(I,JS).EQ.1.) GO TO 60                                   LOCATION
   55 CONTINUE                                                      LOCATION
      GO TO 50                                                      LOCATION
   60 CONTINUE                                                      LOCATION
*                                                                   LOCATION
      JE=64                                                         LOCATION
   65 JE=JE - 1                                                     LOCATION
      DO 70 I=IS,IE                                                 LOCATION
      IF (D(I,JE).EQ.1.) GO TO 75                                   LOCATION
   70 CONTINUE                                                      LOCATION
      GO TO 65                                                      LOCATION
   75 CONTINUE                                                      LOCATION
*                                                                   LOCATION
      JM=JS + JE                                                    LOCATION
      DO 200 I=IS,IE                                                LOCATION
*                                                                   LOCATION
      DO 90 J=JS,JE                                                 LOCATION
      JB=J                                                          LOCATION
```

```
         IF (D(I,J).EQ.1.) GO TO 95                              LOCATION  55
      90 CONTINUE                                                LOCATION  56
   $                                                             LOCATION  57
      95 CONTINUE                                                LOCATION  58
         DO 100 J=JS,JE                                          LOCATION  59
         JL=JM - J                                               LOCATION  60
         IF (D(I,JL).EQ.1.) GO TO 105                            LOCATION  61
     100 CONTINUE                                                LOCATION  62
   $                                                             LOCATION  63
     105 CONTINUE                                                LOCATION  64
         PRINT 902,I,JB,JL                                       LOCATION  65
     902 FORMAT (' AT I= ',I2,' J RUNS FROM ',I2,' TO ',I2)      LOCATION  66
     200 CONTINUE                                                LOCATION  67
   $                                                             LOCATION  68
   $     LEAVE AN EXTRA ONE ALL THE WAY AROUND FOR THE INTERPOLATION LOCATION  69
   $                                                             LOCATION  70
         IS=IS - 1                                               LOCATION  71
         IE=IE + 1                                               LOCATION  72
         JS=JS - 1                                               LOCATION  73
         JE=JE + 1                                               LOCATION  74
   $                                                             LOCATION  75
         PRINT 901,IS,IE,JS,JE                                   LOCATION  76
     901 FORMAT ('    IS=',I2,'     IE=',I2,'    JS=',I2,'    JE=',I2)  LOCATION  77
         END                                                     LOCATION  78
   / QXT                                                         LOCATION  79
   / COM ANSWER:                                                 LOCATION  80
```

80 ACTIVE LINE(S)   0 INACTIVE LINE(S)

$$$ SMS167 : PDS DIRECTORY FOR NEWSPL   FILE SUCCESSFULLY UPDATED FOR DECK LOCATION

$DK INITIAL,ACTION=ADD

```
CREATED ON 02/15/84 AT 11:19:05          LAST UPDATED ON 02/15/84 AT 11:19:05 BY SMB VERSION 5.27
LANGUAGE:               USER INFORMATION:

     / GET OBJLIB,PT/OBJLIB                                           INITIAL     1
     / LNK                                                            INITIAL     2
     LIBRARY OBJLIB                                                   INITIAL     3
     INCLUDE TAPER3                                                   INITIAL     4
     / TAPEREAD SEEOPT=0,TAPEPATH=N/CLIMAT,CATPATH=S/INITLDTA,;       INITIAL     5
           RUNTIME=2000,FILESIZE=20,LOADV=$                           INITIAL     6
      &ONE                                                            INITIAL     7
           IS=45,  IE=50,  JS=31,  JE=40,                             INITIAL     8
           NPN=20,  SIGNAD=T,                                         INITIAL     9
      &END                                                            INITIAL    10
     T  SEAA                                                          INITIAL    11
     T0400 A                                                          INITIAL    12
     T0600 A                                                          INITIAL    13
     T0800 A                                                          INITIAL    14
     T1000 A                                                          INITIAL    15
     T1500 A                                                          INITIAL    16
     T2000 A                                                          INITIAL    17
     T3000 A                                                          INITIAL    18
     T4000 A                                                          INITIAL    19
     T5000 A                                                          INITIAL    20
     S0000 A                                                          INITIAL    21
     S0050 A                                                          INITIAL    22
     S0100 A                                                          INITIAL    23
     S0200 A                                                          INITIAL    24
     S0600 A                                                          INITIAL    25
     S1000 A                                                          INITIAL    26
     S2000 A                                                          INITIAL    27
     S3000 A                                                          INITIAL    28
     S4000 A                                                          INITIAL    29
     S5000 A                                                          INITIAL    30

              30 ACTIVE LINE(S)             0 INACTIVE LINE(S)

*** SMB167 : PDS DIRECTORY FOR NEWSPL  FILE SUCCESSFULLY UPDATED FOR DECK INITIAL

$OK FORCING,ACTION=ADD
```

SMS VERSION 5.27   LISTING OF DECK   F O R C I N G   02/15/84   PAGE 1

CREATED ON 02/15/84 AT 11:19:06   LAST UPDATED ON 02/15/84 AT 11:19:06 BY SMS VERSION 5.27
LANGUAGE: USER INFORMATION:

```
/ RGET OBJLIB.PT/OBJLIB                                              FORCING    1
/ LNK                                                                FORCING    2
LIBRARY OBJLIB                                                       FORCING    3
INCLUDE TAPER                                                        FORCING    4
/ TAPEREAD TAPEPATH=N/SAIATM2,CATPATH=S/FORCING,:                    FORCING    5
     RUNTIME=4000,FILESIZE=20,LOADV=$                                FORCING    6
 &ONE                                                                FORCING    7
     YEAR=77,  MONTH=1,  DAY=7,  HOUR=0,                             FORCING    8
     IS=45,  IE=50,  JS=31,  JE=40,                                  FORCING    9
     NT=56,  DT=6.,                                                  FORCING   10
     SIGMAD=T,   A27A28=T,                                           FORCING   11
 &END                                                                FORCING   12
```

12 ACTIVE LINE(S)   0 INACTIVE LINE(S)

$$$ SMS167 : PDS DIRECTORY FOR NEWGPL   FILE SUCCESSFULLY UPDATED FOR DECK FORCING

END

FILMED

11-84

DTIC