AD-A064 788 SCIENCE APPLICATIONS INC MCLEAN VA F/G 20/4
TWO-DIMENSIONAL NUMERICAL MODEL OF THE NEAR-FIELD FLOW FOR AN O--ETC(U)
1977 G O ROBERTS, S A PIACSEK, J TOOMRE N00014-76-C-0610
UNCLASSIFIED SAI-76-626-WA NRL-GFD/OTEC-7-76 NL

1 OF 1
ADA
064788

END
DATE
FILMED
4-79
DDC

AD A064788

DDC FILE COPY

LEVEL

533,029

SCIENCE APPLICATIONS, INC.

APPROVED FOR PUBLIC RELEASE
DISTRIBUTION UNLIMITED

79 01 18

(18) NRL

(19) NRL-GFD/OTEC-7-76

UC-64

A035462

(6)

TWO-DIMENSIONAL NUMERICAL MODEL OF THE NEAR-FIELD FLOW FOR AN OCEAN THERMAL POWER PLANT. PART III. MANUAL FOR THE COMPUTER CODE NRFLØ2.

(10) Glyn O. Roberts, Steve A. Piacsek, Juri Toomre

~~Fluid Mechanics Division,~~
Science Applications, Inc.

408404

(15) NØØØ14-76-C-Ø61Ø, E(49-26)-1ØØ5

(11) 1977

(12) 54p.

Steve A. Piacsek and Juri Toomre

Geophysical Simulation Section, Code 7750
Naval Research Laboratory.

Report written in 1977

NRL 533 029

(14) SAI-76-626-WA

N00014-76-C-0610

previous report is SAI-76-625-WA (ADA 035462)

Supported by the Ocean Thermal Energy Conversion Program, Division of Solar Energy, Energy Research and Development Administration, under ERDA contract E (49-26) 1005.

408 404 79 01 18 048

TABLE OF CONTENTS

ABSTRACT

This report is a user's manual for our computer code NRFL02 which has been developed to calculate the near-field stratified turbulent flow driven by the intakes and outflows of an ocean thermal power plant. The code uses a two-dimensional geometry and a four-parameter first-order turbulence closure model. Sophisticated numerical methods enable convergent and accurate solutions to be obtained rapidly and economically. A large and flexible printer output package provides for the display and interpretation of the results.

1. INTRODUCTION

In Parts I and II of this report, we have described our two-dimensional computer model NRFLO2 for calculating the near-field external flow of an ocean thermal power plant (OTPP), and have presented numerical results for a proposed experimental simulation and for the Lockheed baseline OTPP design. Here, in Part III, we give a listing of the computer code and a brief description of its use.

The code is written in Fortran, and exists in both an IBM dialect and a CDC dialect. It uses card input for five lines of text, but otherwise the input parameters are specified in a brief main program which must be re-compiled for every run. In Sections 2 and 3 below, we describe the input parameters and the use of NRFLO2. Sections 4 and 5 present a brief description of the subroutines and their relationships, and a full listing of the code.

2. INPUT PARAMETERS

2.1 The Main Program

The parameter values for NRFL02 are not read as data, they are specified in a brief main program and passed to the subroutine PR, the master control program. Thus, in a typical run, only the main program needs to be compiled. The main program is listed at the beginning of Section 5 below. The input parameters are defined in terms of Part I of this report, and we will refer, for example, to equation (I/13) and page I/14.

2.2 The Non-Dimensionalization

The code NRFL02 makes the basic equations (I/13) non-dimensional by using the vertical domain size D (page I/14) as a length scale, scaling the temperature so that the ambient temperature $T_a(z)$ is zero at the bottom of the domain and one at the top, and choosing the time scale so that $g\alpha$ is unity. The units for the different variables are, therefore, as follows:

Length	D	;	(1a)
Time	$(D/g\alpha\Delta T)^{\frac{1}{2}}$	;	(1b)
Speed	$(Dg\alpha\Delta T)^{\frac{1}{2}}$	;	(1c)
Acceleration	$g\alpha\Delta T$	;	(1d)
Diffusivity	$(D^3 g\alpha\Delta T)^{\frac{1}{2}}$	;	(1e)
Stream-Function	$(D^3 g\alpha\Delta T)^{\frac{1}{2}}$	;	(1f)
Volume Flux	$(D^5 g\alpha\Delta T)^{\frac{1}{2}}$	;	(1g)
Kinetic Energy Density	$Dg\alpha\Delta T$	;	(1h)

where ΔT is the dimensional ambient temperature difference between the top and bottom of the computational domain.

The code input is given in non-dimensional terms, with the following exceptions. The array TFAHR(4) of OTPP temperatures, described in Section 2.3, and the top and bottom ambient temperatures TT and TB (Section 2.4) are specified in degrees Fahrenheit. The vertical domain size DSCALE and the region width WIDFT in the y-direction (Section 2.7) are given in ft. Note that for equations (1):

$$D = \text{DSCALE} \quad , \tag{2a}$$

$$\Delta T = \text{TT-TB} \quad , \tag{2b}$$

$$g = 32 \text{ ft/sec}^2 \quad , \tag{2c}$$

$$\alpha = 1.3 \times 10^{-4}/^{\circ}\text{F} \quad . \tag{2d}$$

These quantities do not affect the calculation, but only the output, which is presented in engineering units, in terms of feet, seconds, and degrees Fahrenheit.

For output purposes, z is measured up from the top of the domain (normally the water surface). Within the code, z is measured up from the bottom of the domain, and thus z = 1 at the top.

2.3 OTPP Design Parameters

The real arrays AMPU, AMPQ, TANALF, TFAHR, RPL, and ZCENT, of dimension 4, are initialized by data statements, and correspond to the dimensionless arrays a_k, E_k, α_k, T_k, r_k, and z_k in Section I/3, as follows:

$$a_k = \text{AMPU(k)} \quad ; \tag{3a}$$

$$E_k = \text{AMPQ(k)} \quad ; \tag{3b}$$

$$\tan \alpha_k = \text{TANALF(k)} \quad ; \tag{3c}$$

$$T_k = \text{(TFAHR(k) - TB)/(TT-TB)} \quad ; \tag{3d}$$

$$r_k = \text{RPL(k)} \quad ; \tag{3e}$$

$$z_k = \text{ZCENT(k)} \quad ; \tag{3f}$$

where z_k, being dimensionless, is measured up from the domain bottom, like z.

The quantities E_k, α_k, and T_k determine the inflow values of $\overline{E}$, $\overline{w}$, and $\overline{T}$ in equations (I/16), and are, therefore, irrelevant unless a_k is positive. Use is made of this fact to provide the option of determining the OTPP outflow temperatures by calculating the inflow temperatures from equation (I/17) and adding or subtracting a temperature increment.

The first two regions of negative $\overline{u}$ on the left boundary, starting from the bottom, are found, and the corresponding dimensionless mean plant inflow temperatures T_k are determined using equation (I/17). The dimensional plant inflow temperatures TIN(1) and TIN(2) are then determined using equation (3d), and TFAHR(2) and TFAHR(3) are replaced by the expressions

TFAHR(2) = TIN(1) + TFAHR(1) (4a)

TFAHR(3) = TIN(2) + TFAHR(4) . (4b)

Thus, to obtain fixed OTPP outflow temperatures, positive values of a_1 and a_4 should be used, since TFAHR(1) and TFAHR(4) remain unchanged. For an outflow 3°F warmer than the deepest OTPP inflow, a_2 should be positive and TFAHR(1) should be 3; also, a_1 should not be positive, so that TFAHR(1) is otherwise irrelevant. For an outflow 3°F cooler than the second deepest OTPP inflow, a_3 should be positive, TFAHR(4) should be -3, and a_4 should not be positive. If there is only one OTPP inflow region, then TIN(2) will be zero, and the original TFAHR(3) value will be replaced by TFAHR(4).

It has been suggested that the warm and cold outflows from the plant should be mixed. This can be accomplished consistently by making a_k positive for any two k values, and making the corresponding z_k and r_k values equal.

2.4 The Ambient Ocean

The dimensionless ambient temperature distribution is given by equation (I/18a), where

$$d_t = -ZTC \qquad (5a)$$

$$z_t = WID \qquad (5b)$$

and T_t and T_r are defined so that $T_a(0) = 0$ and $T_a(1) = 1$. The dimensionless ambient turbulent kinetic energy distribution is given by

$$E_a(z) = QAMB \exp\left\{(z - 1)/QDEP\right\} \quad . \qquad (6)$$

The dimensionless parameters QAMB and QDEP correspond to E_o and z_E in equation (I/18c).

2.5 The Turbulence Model

The parameters defining the turbulence model are:

$$L = TLEN \quad , \qquad (7a)$$

$$c_f = TCOF \quad , \qquad (7b)$$

$$c_s = CSTRAT \quad , \qquad (7c)$$

$$c_w = CW \quad . \qquad (7d)$$

Here L and TLEN are, of course, dimensionless; the dimensional turbulence length scale is TLEN x DSCALE ft.

In addition, the parameter PORC is set in this section of the main program. PORC replaces the constant $\frac{1}{2}$ in equation (I/22b), determining the porosity distribution which stops the reflection of internal waves by the right-hand boundary.

2.6 The Computational Mesh and Time-Step

The number of mesh points in the vertical direction is input as the parameter JJ in this section; JJ must be no greater than the vertical dimension in the array declarations in subroutines PR and MARCH (40 in the listing at card B28 in Section 5, but see Section 3.2). The number of mesh points in the horizontal direction is II, and is set in the subroutine PR.

The computational domain is rectangular, with

$$0 \leq z \leq 1 \quad , \qquad (8a)$$

$$0 \leq x \leq \gamma \quad , \qquad (8b)$$

in dimensionless terms, cf. page I/14. Here

$$\gamma = \text{WIDTH} \quad . \qquad (9)$$

The mesh intervals δx and δz increase in geometric progression, with the rate of increase defined by the input parameters

$$\text{XSTRCH} = \log(\delta x_{right}/\delta x_{left}) \quad , \qquad (10a)$$

$$\text{ZSTRCH} = \log(\delta z_{bottom}/\delta z_{top}) \quad . \qquad (10b)$$

The total dimensionless calculation time is

$$t_{end} = \text{TOTIME} \times \text{WIDTH}/u_{max} \quad , \qquad (11)$$

where u_{max} is the maximum of the imposed horizontal velocity on the left boundary. The time-step is determined from equation (I/34), with

$$c = \text{CFL} \quad . \qquad (12)$$

2.7 The Output Options

The code prints a diagnostic line of maxima and integrals, for every time step. In addition, it prints a variety of contour plots and curves NOUTPT times per run, including the final print which gives greater detail. Afterwards, NCOPYS copies of a result summary are printed; this result summary is illustrated by Figures I/3 and I/4.

DSCALE is the depth D of the computational region, in feet. WIDFT is the width of the region, in the y-direction. It is multiplied by the dimensional stream-function to give the volume flux in ft^3/sec.

3. USE OF NRFLO2

3.1 Choice of Input Parameters

Once a particular dimensional temperature profile (I/18a) and domain size D = DSCALE have been chosen, then TT and TB can be calculated, and the units (1) can be found. Because the stratification tends to produce horizontal flow, large D values are not required.

Since the external flow for the OTPP designs at present under consideration is not two-dimensional, the determination of the design parameters (Section 2.3) to be used in the model is difficult. An example is given in Part II of this report for the Lockheed baseline design; the greatest uncertainty is in the appropriate value of WIDFT. The arrays AMPU, RPL, and ZCENT must, of course, be scaled.

The ambient turbulence must be small enough to be innocuous, but must not be zero. Dimensionless QAMB values between 10^{-3} and 10^{-6} have been used, with QDEP from $\frac{1}{2}$ to 10. The results are not at all sensitive to these values.

The turbulence model parameters are a major problem. Our present choices

$$\text{TLEN} = L = \text{RPL} \times 0.4 \quad , \tag{13a}$$

$$\text{TCOF} = c_f = 0.5 \quad , \tag{13b}$$

$$\text{CSTRAT} = c_s = 0.1 \quad , \tag{13c}$$

$$\text{CW} = c_w = 0.1 \quad , \tag{13d}$$

are based on our experience in the simulation of turbulent submarine wakes. These parameters, and the corresponding parameters in NRFLO2 successors, must be tuned to give the best agreement with a wide range of experimental observations. The porosity coefficient TCOF should be about $\frac{1}{2}$; it is a part of the numerical method, not the turbulence model.

The domain WIDTH (9) should be between 2 and 4. The calculations in Part I, with the maximum possible port separation, used $\gamma = 2$ for display convenience; a slightly larger value would have given a better far-field result. Values of XSTRCH from 0.5 to 2 are appropriate; the calculation needs good horizontal resolution only near the left boundary. The number II of mesh points in the x-direction (see Section 3.2) should be such that the mesh interval δx at the left is about half of the smallest RPL value. The quantities JJ and ZSTRCH should similarly be chosen so that the mesh interval δz is about one third of any neighboring RPL values. Less resolution is needed in z ranges well away from any significant ZCENT values, and ZSTRCH magnitudes of 1 or more can sometimes be specified.

TOTIME values of 10 to 20 have been used in our computations. If the solutions do not become effectively steady in this time, then the oscillations appear to continue indefinitely; our problems with instability of the mean flow solutions have not yet been entirely solved. The time-step parameter CFL should be about 0.6; larger values occasionally result in numerical instability.

3.2 Dimension Statements

The computational arrays are dimensioned in the subroutines PR and MARCH, in the statements labeled B24 to B43 and D22 to D37 in the listing in Section 5. The value of JJ, specified in the main program, can be any number up to the corresponding dimension in the arrays (40 in the listing). The value of II must agree with the first dimension of the two-dimensional array declaration (12 in statements B28 to B31 and D22 to D25); therefore, II is set to 12 in statement B60. Values of II up to 25 can be used by changing the declarations of the two-dimensional arrays; larger values require changing the dimension statements for the one-dimensional arrays associated with the x-direction.

3.3 Input and Temporary Storage Streams

The required streams are listed in CDC form in the comment card at the beginning of the main program. Five cards of text (up to 60 characters per card) are read from stream 5 and printed on the output; the first card is printed as a heading for all the plots.

The streams 8 to 13 are used for temporary storage, with a printer stream format. Some of the output routines write to these streams, and the subroutine COMBIN then reads from specified streams, combines several streams as needed, and copies them to the printer stream.

The printer output is on stream 6. A printer line of up to 128 characters, plus the control character, is produced.

3.4 Output

The text from the five input cards is printed first, followed by a list of the input parameter values from the main program. This section of text is repeated in the result summary output, as at the bottom of Figures I/3 to I/8. The non-uniform meshes and intervals, and the time-step, the number of time-steps and the total dimensionless time, determined by the code, are also printed.

For every pair of leapfrog time-steps, the code outputs a diagnostic printer line giving various maxima and minima, together with the Fahrenheit OTPP inflow temperatures determined using equation (I/17) and equation (3d). This output can be used to investigate convergence to a steady solution, as well as for detecting errors or instability.

The total number of time-steps is split into NOUTPT equal parts, and after each section a number of printer plots are generated. The first three show the volume flux added to the ocean below depth z on the left boundary and on the right boundary, and below temperature T on the right boundary. This output is illustrated, for the strong-flow case of Part I, in Figure 1.

A plot of the surface temperature, as a function of x, is presented next, showing the temperature loss through turbulent mixing as the surface layers are drawn towards the OTPP inflow.

Contour plots are printed next, displaying seven variables as functions of x and z. The variables are as follows:

Turbulent kinetic energy density
$\overline{E}(ft^2/sec^2)$; (14a)

Turbulent diffusivity $K(ft^2/sec)$; (14b)

Diffusivity factor $c_s N^2 L^2/\overline{E}$; (14c)

Richardson number $N^2/\frac{1}{2}(\overline{u}_{i,j} + \overline{u}_{j,i})^2$; (14d)

Reynolds number
$\{\frac{1}{2}(u_{i,j} + u_{j,i})^2\}^{\frac{1}{2}}/K$; (14e)

Temperature $\overline{T}(^{o}F)$; (14f)

Volume flux function
WIDFT x $\psi(ft^3/sec)$. (14g)

These are contoured over the whole domain, with a specification of 12 contour intervals. At the end of the run, the seven plots are repeated for just the left half of the domain, double scale, using 20 contour intervals. This gives higher resolution in the region where smaller length scales are present.

FIGURE 1. Illustration of the Environmental Impact Plots

The three printer plots show the volume flux in ft^3/sec added to the "ocean" below depth z ft on the left and right boundaries, and below temperature T on the right boundary, for the NRFLO2 simulation of the proposed strong-flow experiment reported in Part I. On the left boundary, a flux 1.44 ft^3/sec (for 1 ft of tank width) is added in the bottom 1 ft, and an equal flux is removed in the 1 ft interval at the surface. As a result of the turbulent transport, entrainment, and recirculation in the near-field, the far-field impact is the addition of 0.606 ft^3/sec below a depth of 1.9 feet in the temperature range from 43^oF to 59^oF, and the removal of an equal flux from the shallower water, between 59^oF and 80^oF.

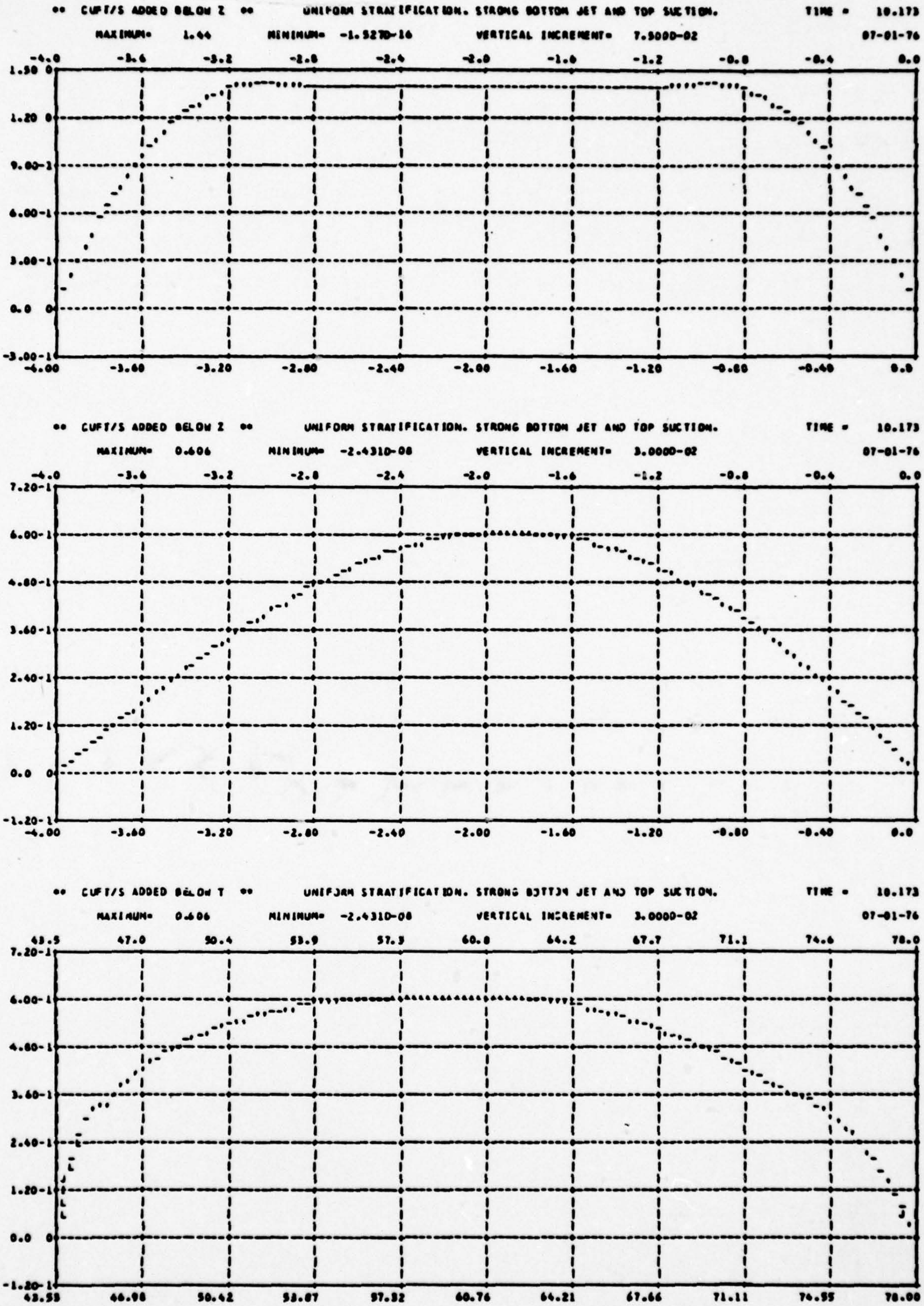

FIGURE 1. Illustration of the Environmental Impact Plots

Finally, the results are summarized using the format of Figures I/3 and I/4. One copy of this format is generated at the intermediate stages, and NCOPYS copies are generated at the end, so that a copy can be photographed, and so that copies can be sent to OTPP designers direct from the printer.

4. BRIEF DESCRIPTION OF THE SUBROUTINES

4.1 NRFLO2 and PR

In the program listing in Section 5, each subroutine has a different alphabet identifier. In this section, we give the identifier in brackets (note that this identifier is not an argument). Where appropriate, we refer also to the individual card identifier.

The main program NRFLO2(A) must be compiled for every run. It calls PR(B), the master control program, which does all the work. PR and the subroutine MARCH(D) must be recompiled every time II, the number of mesh points in the x-direction, is changed (see Section 3.2). The other subroutines need only be compiled once.

The master control program PR receives the input parameters from the main program (some under different names, because they are passed elsewhere in common blocks). It reads the text cards, finds the date, and writes (B75) the text and the list of input parameter values to the printer and to stream 10 (for later printing). It sets up the mesh (B100) and the boundary and initial conditions (B158), with the subprogram FUNC(C) used for the function f(x) in equation (I/14). It calculates the time-step and the time-step counts (B230), and sets the auxiliary arrays for the calculation (B248) and the arrays for solving Poisson's equation (I/31) [using the subroutine ROPT(I)].

The leapfrog time-stepping is done by the subroutine MARCH(D). There are two arrays for each of the four variables $\overline{u}$, $\overline{w}$, $\overline{T}$, and $\overline{E}$, one for the even time-steps and one for the odd. MARCH is called twice for each pair of time-steps, to replace first one set of arrays and then the other set with its new values. The main loop extends from B293 to the end, with the initialization and parasite removal (page I/27) followed by the double call to MARCH, the call to OUTPUT(M) which controls all the printing, and the modification of the plant outflow temperatures according to equations (4) and (3d).

4.2 MARCH and Associated Subroutines

The subroutine MARCH(D) extrapolates the pressure for equation (I/29) and evaluates the turbulent diffusivity (D45). It prepares arrays classifying the mesh points on the left and right boundaries as inflow, slow outflow, or fast outflow (D61), for applying the boundary conditions in Sections I/3 and I/4 and the prescription on page I/24 for fast outflow.

The subroutines FCSTD(E), FCSTU(F), and FSCTW(G) advance $\overline{E}$, $\overline{T}$, $\overline{u}$, and $\overline{w}$ according to equations (I/26), see card D102. These subroutines use TRID1(H) for the tridiagonal implicit solutions. The pressure operations described on page (I/26) are applied starting with card D114, using the subroutines TSTEP(K) for equations (I/30) and POISB(J) for the Poisson equation (I/31). The subroutine BOUND(L) is then called, to apply the boundary conditions on the top and sides using the arrays determined earlier. This completes the time-step.

4.3 OUTPUT and Associated Subroutines

The subroutine OUTPUT(M) is called from PR every time-step pair, and generates the output described in Section 3.4. It evaluates the various maxima and minima (M29), the divergence (M51), and the inflow temperatures (M59), and prints the diagnostic line. Control is then returned to PR unless the count is satisfied for further output (M93).

The inflow temperatures are written to the end of stream 10, for later printing with the results summary (M100). The mesh arrays are made dimensional (M114) and the speed scale (1c) is found (M130). The Figure 1 environmental impact plots and the surface temperature plot are printed (M129), using the subroutine XPLOT(R), which calls MAXMIN(S) and SHORT7(T). The seven contour plots are printed in the section from M150 to M219, using the subroutine CONTOR(N). The loop starting at M153 produces two series of seven plots at the end of the calculation. The second series displays only the left-half of the computational region, with more contours and with twice the resolution of the first series. The subroutine BD(P) is used to prepare some of the seven functions for CONTOR, by applying symmetry boundary conditions to specify undetermined boundary values.

The plots in the results summary of $\overline{T}$ and $\overline{u}$ at the left and right boundaries, together with the ambient temperature, are produced by the subroutine VPLOT(Q). The speed and temperature plotting ranges are evaluated after card M222, using the subroutine MAXMIN(S). All four plots in the result summary, as shown in Figures I/3 and I/4, use the same call M263, with a loop M245 for left and right and a loop M259 for small and large. The plots are written on streams 8, 9, 11, and 12; the scales are then written beneath the plots (M264).

The results summary is written starting at M279; if multiple copies are appropriate, it is written to stream 13 (M281) and NCOPYS copies are written later to stream 6 (M344).

First, the temperature and the volume flux function (14g) are contoured again, this time with a specification of 9 contour intervals (M285). Then, the small size VPLOT output on streams 8 and 9 is combined with the results summary on stream 10, using the subroutine COMBIN(O), to generate the format in the lower panel of Figure I/3 (M308). Next, the diffusivity is contoured (M313). The large-scale VPLOT output on streams 11 and 12 is then combined, as in the center panel of Figure I/4 (M323). Finally, the results summary on stream 10 is split up between streams 8 and 9 (M325) and recombined (M342) to produce the format in the lower panel of Figure I/4.

This completes the results summary, and after restoring the dimensionless meshes (M353) control is returned to PR.

5. NRFLO2 LISTING

The code listing is reproduced photographically on the following pages, in an IBM dialect. Each card has an identifier, the letter being different for each subroutine. The subroutines are listed below, with their identifiers, names of the calling subroutines, and the purpose.

IDENTIFIER AND NAME		CALLED FROM	PURPOSE
A	NRFLO2		Initialize parameters for PR
B	PR	NRFLO2	Master control program
C	FUNC	PR	Function f(x) in equation (I/14)
D	MARCH	PR	Advance one leapfrog time-step
E	FCSTD	MARCH	Advance $\overline{E}$ and $\overline{T}$
F	FCSTU	MARCH	Advance $\overline{u}$
G	FCSTW	MARCH	Advance $\overline{w}$
H	TRID1	FCSTD FCSTU FCSTW	Tridiagonal implicit solution
I	ROPT	PR	Set arrays for POISB
J	POISB	MARCH	Poisson solver
K	TSTEP	MARCH	Add pressure gradient to flow
L	BOUND	MARCH	Apply boundary conditions
M	OUTPUT	PR	Output control program
N	CONTOR	OUTPUT	Contour program
O	COMBIN	OUTPUT	Combine output streams
P	BD	OUTPUT	Prepare array boundaries for CONTOR
Q	VPLOT	OUTPUT	Vertical function plots
R	XPLOT	OUTPUT	Horizontal function plots
S	MAXMIN	OUTPUT XPLOT	Maximum and minimum
T	SHORT7	XPLOT	Short format for numbers

```
C     PROGRAM NRFLO2(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE8,TAPE9,
C    1 TAPE10)
C  ***  INITIALIZE ALL PARAMETERS AND CONSTANTS  ***
      IMPLICIT REAL*8(A-H,O-Z)
C
C * * * *  OCEAN ENGINE PARAMETERS * * *
      REAL AMPJ(4),AMPQ(4),TANALF(4),TFAHR(4),RPL(4),ZCENT(4)
      DATA AMPU/-.12,.12,2*0./
      DATA AMPQ/0.,.2,2*0./
      DATA TANALF/4*0./
      DATA TFAHR/-3.,70.,2*0./
      DATA RPL/4*.2/
      DATA ZCENT/.8,.4,2*0./
C
C
C AMBIENT OCEAN
      TT=80
      TB=67
      ZTC=-.2
      WID=.3
      QAMB=.001
      QDEP=.5
C
C
C TURBULENCE MODEL PARAMETERS
      TLEN=.03
      TCOF=.3
      CSTRAT=.1
      CW=.1
      PORC=.5
C
C
C MESH
C XSTRCH=LOG(RIGHT  DX / LEFT DX )
C ZSTRCH=LOG(BOTTOM DZ / TOP  DZ )
C
      JJ=12
      WIDTH=2
      XSTRCH=1
      ZSTRCH=0
C
C
C TIME AND STEP
      TOTIME=10
      CFL=.6
C
C
C OUTPUT
      NOUTPT=2
      DSCALE=250
      WIDFT=100
      NCOPYS=3
      CALL PR (JJ,AMPU,AMPQ,TANALF,TFAHR,RPL,ZCENT,TT,TB,ZTC,WID,QAMB,
     1QDEP,TLEN,TCOF,CSTRAT,CW,PORC,WIDTH,XSTRCH,ZSTRCH,TOTIME,CFL,NOUTP
     2T,DSCALE,WIDFT,NCOPYS)
      STOP
      END


      SUBROUTINE PR (JJT,AMPJ,AMPQ,TANALF,TFAHR,RPL,ZCENT,TT,TB,ZTC,WI
     1D,QAMB,QDEP,TLEN,TCOF,TCST,TCW,PORC,WIDTH,XSTRCH,ZSTRCH,TOTIME,CFL
     2,NOUTPT,DSCALE,WIDFT,NCOPYS)
C  ***  MASTER CONTROL PROGRAM  ***
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON /N1/ DT,TIME
      COMMON /N2/ COF,RLAM,PI,GAP,DEP,CSTRAT,CW
      COMMON /N3/ II,JJ,III,JJ1,II2,JJ2
      COMMON /N4/ AAR,AAZ,KAPPA
      COMMON /N7/ A1,A2,A3,B1,B2,B3,WI,WK
      COMMON /N12/ F,E,ADV,AV2,S,P,PE
      COMMON /N13/ ALP,BET,GAM,DEN,H,SO,ZS,SO1,SO2,ZS1,ZS2,ZFTU,ZFTM,Z
     1FQU,ZFQM,XFTU,XFTM,XFQU,XFQM,XPOR
      COMMON /N14/ DX1,DX2,DXP,DXM,DXT,DXG,DXD,DXU,DXW,DZ1,DZ2,DZP,DZM
     1,DZT,DZG,DZD,DZU,DZW,ZBL,ZBR,ZWL,ZWR
      COMMON /N15/ R,X,Z,Y
      COMMON /N16/ B,BF,UBND,FAC
C
C     REAL    ABS, SQRT
      REAL T,ZT,Q,ZQ,F,E,ADV,AV2
```

```
      REAL U,ZU,W,ZW,S,P,PE
C
C     DIMENSION STATEMENTS IN MAIN AND MARCH
C 1D ARRAYS CAN BE LEFT WITH LARGER DIMENSION, BUT FIRST DIMENSION
C  OF 2D ARRAYS MUST BE RIGHT.
C
      DIMENSION U(12,40), ZU(12,40), W(12,40), ZW(12,40), S(12,40), P(
     112,40), PE(12,40)
      DIMENSION T(12,40), ZT(12,40), Q(12,40), ZQ(12,40), F(12,40), E(
     112,40), AV2(12,40), ADV(12,40)
      DIMENSION R(25), X(25), Z(40), Y(40)
      DIMENSION DX1(25), DX2(25), DXP(25), DXM(25), DXT(25), DXS(25),
     1DXD(25), DXU(25), DXW(25)
      DIMENSION DZ1(40), DZ2(40), DZP(40), DZM(40), DZT(40), DZS(40),
     1DZD(40), DZU(40), DZW(40), ZBL(40), ZBR(40), ZWL(40), ZWR(40)
      DIMENSION A1(25), A2(25), A3(25), B1(40), B2(40), B3(40), WI(25)
     1, WK(40)
      DIMENSION B(40), FAC(40), BF(40)
      DIMENSION UBND(40)
      DIMENSION ALP(40), BET(40), GAM(40), DEN(40), H(40), S0(40), ZS(
     140), S01(40), S02(40), ZS1(40), ZS2(40), XFTU(40), XFTM(40), XFQU(
     240), XFQM(40), ZFTU(40), ZFTM(40), ZFQU(40), ZFQM(40), XPOR(40)
C
C
      REAL AMPJ(4),AMPQ(4),TANALF(4),TFAHR(4),RPL(4),ZCENT(4)
      COMMON /TXTC/ ITXT(60),DAY,JTXT(60)
      DIMENSION TIN(2)
C
C
C TXT IS TEXT FOR EACH RUN, ON XPLOT AND CONTOR AND VPLOT OUTPUT.
C READ OFF DATA CARD, (60A1) .
      READ (5,10) ITXT,JTXT
10    FORMAT (60A1/(15A4))
      CALL IDAY (DAY)
C
      II=12
      JJ=20
      II=8
      II=12
      JJ=JJT
      TIME=TOTIME
      GAP=WIDTH
      RLAM=TLEN
      COF=TCOF
      CSTRAT=TCST
      CW=TCW
C
C END INITIALIZATIONS
C
C
C     * * * * * * *    PRINT INITIAL CONDITIONS * * * * * * * * *
C
      DO 20 I=6,10,4
20    WRITE (I,30) ITXT,JTXT,AMPU,AMPQ,TANALF,TFAHR,RPL,ZCENT,TT,TB,ZT
     1C,WID,QAMB,QDEP,RLAM,COF,CSTRAT,CW,PORC,GAP,XSTRCH,ZSTRCH,II,JJ,TI
     2ME,CFL,NOUTPT,DSCALE,WIDFT,DAY
30    FORMAT (1H1//1X60A1//4(1X,15A4/)/////' AMPU      ',4F12.4/' AMPQ
     1   ',4F12.3/' TANALF   ',4F12.3/' TFAHR ',4F12.1/' RPL      ',4F12.3
     2/' ZCENT    ',4F12.3/' TT,TB,ZTC,WID ',F4.1,F12.1,F14.3,F12.3,/' QA
     3MB,QDEP ',1P,D24.1,0P,F9.2/' TLEN,CF,CS,CW,PORC',5F7.3,/' GAP,XSTR
     4CH,ZSTRCH,II,JJ',3F8.3,2I4/' TIME,CFL,NOUTPT,DSCALE,WIDFT',F6.1,F6
     5.2,I3,2F7.1/1X,A8,////////)
C
C
      CALL INDUMP
      CALL NOUNDF
      PI=4*DATAN(1.0D0)
      KAPPA=2
      KAPPA=3
      II1=II-1
      II2=II-2
      JJ1=JJ-1
      JJ2=JJ-2
      I2=II2
      J1=JJ1
C
C
C SET UP MESH
      FMULT=1+XSTRCH/II2
      ZMULT=1-ZSTRCH/JJ2
```

```
      DZ=1
      DX=DZ
      DEP=1
C
C
      RFM=FMULT**.5D0
      DX1(1)=DX/FMULT
      DX2(1)=DX/RFM
      X(1)=0
      R(1)=-DX/RFM/(1+RFM)
      DO 40 I=2,II
      DX2(I)=DX2(I-1)*FMULT
      DX1(I)=DX1(I-1)*FMULT
      X(I)=X(I-1)+DX1(I)
40    R(I)=R(I-1)+DX2(I-1)
      FACT=GAP/X(II1)
      DO 50 I=1,II
      X(I)=X(I)*FACT
      R(I)=R(I)*FACT
      DX1(I)=DX1(I)*FACT
50    DX2(I)=DX2(I)*FACT
      PRINT 60
60    FORMAT (//' DX1,DX2,X,R',/)
      PRINT 70, (DX1(I),I=1,II)
      PRINT 70, (DX2(I),I=1,II)
      PRINT 70, (X(I),I=1,II)
      PRINT 70, (R(I),I=1,II)
70    FORMAT (/(10F12.5))
C
C
      RFM=ZMULT**.5D0
      DZ1(1)=DZ/ZMULT
      DZ2(1)=DZ/RFM
      Y(1)=0
      Z(1)=-DZ/RFM/(1+RFM)
      DO 80 J=2,JJ
      DZ1(J)=DZ1(J-1)*ZMULT
      DZ2(J)=DZ2(J-1)*ZMULT
      Y(J)=Y(J-1)+DZ1(J)
      Z(J)=Z(J-1)+DZ2(J-1)
80    CONTINUE
      FACT=1/Y(JJ1)
      DO 90 J=1,JJ
      DZ1(J)=DZ1(J)*FACT
      DZ2(J)=DZ2(J)*FACT
      Y(J)=Y(J)*FACT
90    Z(J)=Z(J)*FACT
C
      PRINT 100
100   FORMAT (//' DZ1,DZ2,Y,Z',/)
      PRINT 70, (DZ1(J),J=1,JJ)
      PRINT 70, (DZ2(J),J=1,JJ)
      PRINT 70, (Y(J),J=1,JJ)
      PRINT 70, (Z(J),J=1,JJ)
C
C
C     * * * * * * * *   SET UP INITIAL CONDITIONS   * * * * * * *
C
C LEFT  AND INITIAL U IS UBND(J)
C LEFT W, Q, T, PERMANENT IN ARRAYS, WHERE U.GT.0
C RIGHT AND INITIAL T IS B (J)
C RIGHT AND INITIAL Q IS BF(J)
C RIGHT AND INITIAL W IS ZERO
C
      DARGT=DATAN((DEP-ZTC)/WID)
      DARGB=DATAN((0.-ZTC)/WID)
      ZERO=0
      UMAX=0
      DO 120 J=1,JJ
      UBND(J)=0
      W(1,J)=0
      A=1.D-10
      C=0
      D=0
      DO 110 K=1,4
      AA=FUNC((Z(J)-ZCENT(K))/RPL(K))
      AB=FUNC((Y(J)-ZCENT(K))/RPL(K))
      AU=AMPU(K)/RPL(K)
      AV=DMAX1(AU,ZERO)
      UBND(J)=UBND(J)+AU*AA
```

```
      A=A+AV*AA
      C=C+AV*AV*AV*AMPQ(K)*AA
      D=D+AV*(TFAHR(K)-TB)/(TT-TB)*AA
      W(1,J)=W(1,J)+AV*AB*TANALF(K)
  110 CONTINUE
      Q(1,J)=C/A
      T(1,J)=D/A
      U(1,J)=UBND(J)
      UMAX=DMAX1(UMAX,DABS(UBND(J)))
      B(J)=(DATAN((Z(J)-ZTC)/WID)-DARGB)/(DARGT-DARGB)
      BF(J)=QAMB*DEXP(-(DEP-Z(J))/QDEP)
      IF (U(1,J).LE.0.) Q(1,J)=BF(J)
      IF (Q(1,J).EQ.0.) Q(1,J)=BF(J)
      IF (U(1,J).LE.0.) T(1,J)=B(J)
      DO 120 I=2,II
      U(I,J)=UBND(J)
      T(I,J)=B(J)
      Q(I,J)=BF(J)
      W(I,J)=0
  120 CONTINUE
      BVMAX=0
      DO 130 J=2,JJ1
      BVMAX=DMAX1(BVMAX,((B(J+1)-B(J))/DZ2(J)))
      A=DMAX1(((U(II1,J+1)-U(II1,J-1))/DZ1(J)/2)**2-(T(II,J+1)-T(II,J-
     1))/DZ1(J)/2,ZERO)
  130 Q(II,J)=BF(J)+A*RLAM*RLAM/COF
      BVMAX=DSQRT(BVMAX)
      Q(II,JJ)=Q(II,JJ1)
      Q(II,1)=Q(II,2)
      DO 140 J=1,JJ
      DO 140 I=2,II1
  140 Q(I,J)=Q(II,J)
      DO 150 J=1,JJ
      DO 150 I=1,II
      ZU(I,J)=U(I,J)
      ZW(I,J)=W(I,J)
      ZT(I,J)=T(I,J)
      ZQ(I,J)=Q(I,J)
      ADV(I,J)=0.
      F(I,J)=0.
      E(I,J)=0.
      S(I,J)=0.
      AV2(I,J)=0.
      P(I,J)=0
      PE(I,J)=P(I,J)
  150 CONTINUE
C
C
C TIME STEP AND COUNTS
C
      DT=CFL/(UMAX/DX1(2)+BVMAX)
      TIME=TIME*GAP/UMAX
      NKM=TIME/NOUTPT/DT/2+.5
      NKM=2*NKM
      NN=NKM*NOUTPT
      DT=TIME/NN
      NLF=20
      NLF=9
      NC=0
      NL=NLF
      NK=0
      WRITE (6,160) TIME,DT,NN,NKM,NLF
  160 FORMAT (//' TOTIME ,  DT',2F12.4/' NN,NKM,NLF',3I6//)
      TIME=0.0
C
C
C INITIALISE TIME STEPPING ARRAYS
C
      TP3=2.3
      DO 170 I=1,II1
      DXP(I)=DT/(DX1(I+1)*DX2(I))
      DXM(I)=DT/(DX1(I)*DX2(I))
      DXT(I)=DT/(2*DX1(I))
      DXG(I)=2*DT/DX2(I)
      DXU(I)=DT/(4*DX2(I))
      DXW(I)=DT/(4*DX1(I))
      XPOR(I)=1-PORC*DT*BVMAX*DEXP(-DSINH(DSINH(DMIN1(6-6*R(I)/GAP,TP3
     1))))
  170 CONTINUE
C
```

```
      DO 180 J=1,JJ1
      DZP(J)=DT/(DZ1(J+1)*DZ2(J))
      DZM(J)=DT/(DZ1(J)*DZ2(J))
      DZT(J)=DT/(2*DZ1(J))
      DZG(J)=2*DT/DZ2(J)
      DZU(J)=DT/(4*DZ1(J))
      DZW(J)=DT/(4*DZ2(J))
180   CONTINUE
C
C
C FOR POISSON
      AAR=PI*PI/GAP/GAP/8
      AAZ=DMAX1(4/DX1(2)**2-AAR,4/DZ1(2)**2+AAR,4/DZ1(JJ1)**2+AAR)
      DO 190 I=2,II1
      A1(I)=1.0D0/(DX1(I)*DX2(I))
      A2(I)=1.0D0/(DX1(I)*DX2(I-1))
      A3(I)=A1(I)+A2(I)-AAR
190   CONTINUE
      DO 200 J=2,JJ1
      B1(J)=1.0D0/(DZ1(J)*DZ2(J))
      B2(J)=1.0D0/(DZ1(J)*DZ2(J-1))
      B3(J)=B1(J)+B2(J)+AAR
200   CONTINUE
C
C
      CALL ROPT
C
C
C   * * * * * * *    MARCHING PROCESS BEGINS   * * * * * * *
C
C
210   CONTINUE
      IF (NL-NLF) 260,220,260
C
C INITIALISE ZF AND RE-INITIALISE TO REMOVE PARASITES.
C P IS BACK EXTRAPOLATED.
C
220   CONTINUE
      NL=0
      DO 240 I=1,II
      DO 230 J=2,JJ1
      PST=PE(I,J)
      PE(I,J)=2*PST-P(I,J)
      P(I,J)=PST
230   CONTINUE
      DO 240 J=1,JJ
      ZU(I,J)=U(I,J)
      ZW(I,J)=W(I,J)
      ZT(I,J)=T(I,J)
      ZQ(I,J)=Q(I,J)
240   CONTINUE
      CALL MARCH (U,ZU,W,ZW,T,ZT,Q,ZQ)
      DO 250 J=1,JJ
      DO 250 I=1,II
      ZU(I,J)=(U(I,J)+ZU(I,J))/2.0
      ZW(I,J)=(W(I,J)+ZW(I,J))/2.0
      ZT(I,J)=(T(I,J)+ZT(I,J))/2.
      ZQ(I,J)=(Q(I,J)+ZQ(I,J))/2.
250   CONTINUE
C
260   CONTINUE
C
C
      CALL MARCH (ZU,U,ZW,W,ZT,T,ZQ,Q)
      CALL MARCH (U,ZU,W,ZW,T,ZT,Q,ZQ)
      TIME=TIME+DT*2.
      NL=NL+2
      NC=NC+2
      NK=NK+2
C
C   * * * * * * *    FORM OUTPUT   * * * * * * *
C
      CALL OUTPUT (U,ZU,W,ZW,T,Q,F,E,S,ADV,R,X,Z,Y,DX1,DZ1,RLAM,DT,TIM
     1E,GAP,DEP,II,JJ,NC,NK,NKM,I2,J1,B,DX2,DZ2,TT,TB,ZQ,DSCALE,NN,TIN,W
     2IDFT,ADV,AV2,NCOPYS)
C
C
C MODIFY PLANT OUTFLOW TEMPERATURES.
C
      TFAHR(2)=TIN(1)+TFAHR(1)
```

```
      TFAHR(3)=TIN(2)+TFAHR(4)
      DO 280 J=1,JJ
      IF (U(1,J).LE.0.) GO TO 280
      A=1.D-10
      D=0
      DO 270 K=1,4
      AA=FUNC((Z(J)-ZCENT(K))/RPL(K))
      AU=AMPU(K)/RPL(K)
      AV=DMAX1(AU,ZERO)
      A=A+AV*AA
      D=D+AV*(TFAHR(K)-TB)/(TT-TB)*AA
270   CONTINUE
      T(1,J)=D/A
      ZT(1,J)=D/A
280   CONTINUE
C
C
      IF (NC-NN) 210,290,290
C
290   CONTINUE
C
      STOP
      END


      FUNCTION FUNC (A)
C *** FUNCTION FOR PLANT INFLOWS AND OUTFLOWS ***
      IMPLICIT REAL*8(A-H,O-Z)
      B=1.DO-A*A
      IF (B.LT.0.DO) B=0.DO
      FUNC=B*B
      RETURN
      END


      SUBROUTINE MARCH (U,ZU,W,ZW,T,ZT,Q,ZQ)
C *** MARCH ONE HALF TIME STEP ***
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON /N1/ DT,TIME
      COMMON /N2/ COF,RLAM,PI,GAP,DEP,CSTRAT,CW
      COMMON /N3/ II,JJ,III,JJ1,II2,JJ2
      COMMON /N7/ A1,A2,A3,B1,B2,B3,WI,WK
      COMMON /N12/ F,E,ADV,AV2,S,P,PE
      COMMON /N13/ ALP,BET,GAM,DEN,H,SO,ZS,SO1,SO2,ZS1,ZS2,ZFTU,ZFTM,Z
     1FQU,ZFQM,XFTU,XFTM,XFQU,XFQM,XPOR
      COMMON /N14/ DX1,DX2,DXP,DXM,DXT,DXG,DXD,DXU,DXW,DZ1,DZ2,DZP,DZM
     1,DZT,DZG,DZD,DZU,DZW,ZBL,ZBR,ZWL,ZWR
      COMMON /N15/ R,X,Z,Y
      COMMON /N16/ B,BF,UBND,FAC
C
      REAL ABS,SQRT
      REAL T,ZT,Q,ZQ,F,E,ADV,AV2
      REAL U,ZU,W,ZW,S,P,PE
C
C
      DIMENSION U(12,40), ZU(12,40), W(12,40), ZW(12,40), S(12,40), P(
     112,40), PE(12,40)
      DIMENSION T(12,40), ZT(12,40), Q(12,40), ZQ(12,40), F(12,40), E(
     112,40), AV2(12,40), ADV(12,40)
      DIMENSION R(25), X(25), Z(40), Y(40)
      DIMENSION DX1(25), DX2(25), DXP(25), DXM(25), DXT(25), DXG(25),
     1DXD(25), DXU(25), DXW(25)
      DIMENSION DZ1(40), DZ2(40), DZP(40), DZM(40), DZT(40), DZG(40),
     1DZD(40), DZU(40), DZW(40), ZBL(40), ZBR(40), ZWL(40), ZWR(40)
      DIMENSION A1(25), A2(25), A3(25), B1(40), B2(40), B3(40), WI(25)
     1, WK(40)
      DIMENSION B(40), FAC(40), BF(40)
      DIMENSION UBND(40)
      DIMENSION ALP(40), BET(40), GAM(40), DEN(40), H(40), SO(40), ZS(
     140), SO1(40), SO2(40), ZS1(40), ZS2(40), XFTU(40), XFTM(40), XFQU(
     240), XFQM(40), ZFTU(40), ZFTM(40), ZFQU(40), ZFQM(40), XPOR(40)
C
C
      DT2=2*DT
      JJ1=JJ-1
C
      CALL WR (Q,'QBEF')
      CALL WR (T,'TBEF')
      DO 20 I=1,II
```

```
      DO 10 J=2,JJ1
      PST=PE(I,J)
      PE(I,J)=P(I,J)
      P(I,J)=2*P(I,J)-PST
      F(I,J)=RLAM*SQRT(Q(I,J))/(1.+RLAM*RLAM*(T(I,J+1)-T(I,J-1))/DZ1(J
     1)/2.*CSTRAT/Q(I,J))
C                                            02/)))J,I(W(SBA+))J,I(U(SBA(
C
      IF (F(I,J).LT.0.) F(I,J)=0.
10    CONTINUE
      F(I,1)=F(I,2)
      F(I,JJ)=F(I,JJ1)
20    CONTINUE
      CALL WR (F,'F    ')
C
C ZBL AND ZBR ARE 1 FOR INFLOW, 0 FOR SLOW OUTFLOW, -1 FOR FAST OUTFLOW.
C ZWL AND ZWR ARE 1 FOR INFLOW, 0 FOR SLOW OUTFLOW, -1 FOR FAST OUTFLOW.
      AA=0
      C1=0
      D=0
      ZERO=0.
      ONE=1.
      DO 30 J=1,JJ1
      A=U(1,J)
      C=F(2,J)/DX1(2)+A/2.5/100
      ZBL(J)=0.D0
      IF (A.GT.ZERO) ZBL(J)=ONE
      IF (C.LT.ZERO) ZBL(J)=-ONE
      A=A+AA
      AA=A-AA
      C=C+C1
      C1=C-C1
      ZWL(J)=0.D0
      IF (A.GT.ZERO) ZWL(J)=ONE
      IF (C.LT.ZERO) ZWL(J)=-ONE
      A=U(II1,J)
      C=F(II1,J)/DX1(II1)-A/2.5/100
      ZBR(J)=0.D0
      IF (A.LT.ZERO) ZBR(J)=ONE
      IF (C.LT.ZERO) ZBR(J)=-ONE
      A=A+AA
      AA=A-AA
      C=C+C1
      C1=C-C1
      ZWR(J)=0.D0
      IF (A.LT.ZERO) ZWR(J)=ONE
      IF (C.LT.ZERO) ZWR(J)=-ONE
30    CONTINUE
C
C
      IF (TIME.EQ.0.) CALL BOUND (ZJ,ZW,ZQ,ZT,ZBL,ZBR,ZWL,ZWR,B,BF,II,
     1JJ,DZ1)
      IF (TIME.EQ.0.) CALL BOUND (U,W,Q,T,ZBL,ZBR,ZWL,ZWR,B,BF,II,JJ,D
     1Z1)
C
C
      CALL FCSTD (U,W,T,ZT,Q,ZQ,F,ADV,AV2,ALP,BET,GAM,DEN,H,SO1,SO2,ZS
     11,ZS2,ZFTU,ZFTM,ZFQU,ZFQM,XFTU,XFTM,XFQU,XFQM,DXD,DXP,DXM,DXT,E,S,
     2ZBL,ZBR,DZD,DZP,DZM,DZT,DX1,DX2,DZ1,DZ2,FAC,BF,B,II,JJ)
      CALL FCSTU (U,ZU,W,F,E,ADV,ALP,BET,GAM,DEN,H,SO,ZS,DXP,DZP,ZBL,Z
     1BR,DXM,DZM,DXU,DZU,II,JJ,P,DX2)
      CALL FCSTW (W,ZW,U,T,F,E,ADV,ALP,BET,GAM,DEN,H,SO,ZS,DXP,DZP,Q,X
     1POR,ZWL,ZWR,DXM,DZM,DXW,DZW,II,JJ,P,DZ2)
      CALL WR (ZT,'T    ')
      CALL WR (ZQ,'Q    ')
      CALL WR (ZU,'U1   ')
      CALL WR (ZW,'W1   ')
C
      SIGN=-1.
      CALL TSTEP (ZU,ZW,P,DXG,DZG,II,JJ,SIGN)
C
      DO 40 I=2,II1
      DO 40 J=2,JJ1
      S(I,J)=(ZU(I,J)-ZU(I-1,J))/DX1(I)+(ZW(I,J)-ZW(I,J-1))/DZ1(J)
      S(I,J)=S(I,J)/DT2
40    CONTINUE
C
      CALL POISB (P,S,E,A1,A2,A3,B1,B2,B3,WI,WK,II,JJ)
C
C
```

```
      SIGN=1.
      CALL TSTEP (ZU,ZW,P,DXG,DZG,II,JJ,SIGN)
      CALL BOUND (ZU,ZW,ZQ,ZT,ZBL,ZBR,ZWL,ZWR,B,BF,II,JJ,DZ1)
      DO 50 I=2,II1
      DO 50 J=2,JJ1
      S(I,J)=(ZU(I,J)-ZU(I-1,J))/DX1(I)+(ZW(I,J)-ZW(I,J-1))/DZ1(J)
      S(I,J)=S(I,J)/DT2
50    CONTINUE
C
C
      CALL WR (P,'P   ')
      CALL WR (ZU,'UF  ')
      CALL WR (ZW,'WF  ')
      RETURN
      END

      SUBROUTINE FCSTD (U,W,T,ZT,Q,ZQ,F,ADV,AV2,ALP,BET,GAM,DEN,H,SO1,
     1SO2,ZS1,ZS2,ZFTU,ZFTM,ZFQU,ZFQM,XFTU,XFTM,XFQU,XFQM,DXD,DXP,DXM,DX
     2T,E,S,ZBL,ZBR,DZD,DZP,DZM,DZT,DX1,DX2,DZ1,DZ2,FAC,BF,B,ML,NL)
C *** FORECAST T AND Q ***
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON /N1/ DT,TIME
      COMMON /N2/ COF,RLAM,PI,GAP,DEP,CSTRAT,CW
C
      REAL ABS,SQRT
      REAL T,ZT,Q,ZQ,F,ADV,AV2,U,W,E,S
C
      DIMENSION U(ML,NL), W(ML,NL), E(ML,NL), S(ML,NL)
      DIMENSION T(ML,NL), Q(ML,NL), F(ML,NL), ALP(NL), BET(NL), GAM(NL
     1), DEN(NL), H(NL), SO1(NL), SO2(NL), ZS1(NL), ZS2(NL), ZFTU(NL), Z
     2FTM(NL), ZFQU(NL), ZFQM(NL), XFTU(NL), XFTM(NL), XFQU(NL), XFQM(NL
     3), DXD(ML), DXP(ML), DXM(ML), DXT(ML), DZD(NL), DZP(NL), DZM(NL),
     4DZT(NL), DX1(ML), DX2(ML), DZ1(NL), DZ2(NL), FAC(NL), BF(NL), ADV(
     5ML,NL), AV2(ML,NL), ZT(ML,NL), ZQ(ML,NL), ZBL(NL), ZBR(NL), B(NL)
C
C
      ZERO=0
      TWO=2
      II=ML
      JJ=NL
      II1=II-1
      II2=II-2
      JJ1=JJ-1
      JJ2=JJ-2
C
C
C Z IMPLICIT
      DO 10 J=2,JJ1
      XFTU(J)=(F(2,J)+F(1,J))/2.0D0*(ZT(2,J)-ZT(1,J))
      XFQU(J)=(F(2,J)+F(1,J))/2.0D0*(ZQ(2,J)-ZQ(1,J))
10    CONTINUE
C
      DO 70 I=2,II1
C
      DZD(1)=(F(I,2)+F(I,1))/2.0D0
      DO 20 J=2,JJ1
      DZD(J)=(F(I,J+1)+F(I,J))/2.0D0
20    CONTINUE
      DO 30 J=2,JJ1
      FA=(F(I+1,J)+F(I,J))/2.0D0
      XFTM(J)=XFTU(J)
      XFQM(J)=XFQU(J)
      XFTU(J)=FA*(ZT(I+1,J)-ZT(I,J))
      XFQU(J)=FA*(ZQ(I+1,J)-ZQ(I,J))
30    CONTINUE
C
      GAM(1)=-DZD(1)*DZM(1)
      DO 40 J=2,JJ1
      AA=(U(I,J)-U(I-1,J))/DX1(I)
      AB=(U(I,J+1)+U(I-1,J+1)-U(I,J-1)-U(I-1,J-1))/DZ1(J)+(W(I+1,J)+W(
     1I+1,J-1)-W(I-1,J)-W(I-1,J-1))/DX1(I)
      ADV(I,J)=DXT(I)*(U(I,J)*T(I+1,J)-U(I-1,J)*T(I-1,J))+DZT(J)*(W(I,
     1J)*T(I,J+1)-W(I,J-1)*T(I,J-1))
C
C UPSTREAM FINITE DIFFERENCE REPRESENTATION FOR Q.
      UQ=U(I,J)+U(I-1,J)
      WQ=W(I,J)+W(I,J-1)
      IQ=1
```

```
      JQ=1
      IF (UQ.LT.0) IQ=-1
      IF (WQ.LT.0) JQ=-1
C     AV2(I,J)=DXT(I)*(U(I,J)*Q(I+1,J)-U(I-1,J)*Q(I-1,J))+DZT(J)*(W(I,
C    1J)*Q(I,J+1)-W(I,J-1)*Q(I,J-1))
      WB=(W(I,J)+W(I,J-1))*.5D0
      AV2(I,J)=UQ*DXT(I)*IQ*(Q(I,J)-Q(I-IQ,J))+WQ*DZT(J)*JQ*(Q(I,J)-Q(
     1I,J-JQ))-DT*F(I,J)*((4.*AA*AA+AB*AB/16.-(T(I,J+1)-T(I,J-1))/2./DZ1
     2(J))-Q(I,J)*COF/RLAM/RLAM)-WB*WB*CW*F(I,J)*DT*(T(I,J+1)-T(I,J-1))/
     32./DZ1(J)/Q(I,J)
      AA=0.D0
      AB=0.D0
      A=DT*F(I,J)*((4.*AA*AA+AB*AB/16.+(T(I,J+1)-T(I,J-1))/2./DZ1(J))/
     1Q(I,J)+COF/RLAM/RLAM)+UQ*IQ*DXT(I)+WQ*JQ*DZT(J)
      ALP(J)=-DZD(J-1)*DZP(J-1)
      GAM(J)=-DZD(J)*DZM(J)
      BET(J)=1.0D0-GAM(J)-ALP(J)
      E(I,J)=A
      SO1(J)=ZT(I,J)+DXM(I)*XFTU(J)-DXP(I-1)*XFTM(J)-ADV(I,J)
      SO2(J)=ZQ(I,J)+DXM(I)*XFQU(J)-DXP(I-1)*XFQM(J)-AV2(I,J)+2*A*Q(I,
     1J)
40    CONTINUE
C
      BET(2)=BET(2)+ALP(2)
      BET(JJ1)=BET(JJ1)+GAM(JJ1)
C
      CALL TRIDI (ALP,BET,GAM,SO1,ZS1,JJ)
      DO 50 J=2,JJ1
      BET(J)=BET(J)+E(I,J)*TWO
50    CONTINUE
      CALL TRIDI (ALP,BET,GAM,SO2,ZS2,JJ)
C
      DO 60 J=2,JJ1
      ZT(I,J)=ZS1(J)
      S(I,J)=ZQ(I,J)
      ZQ(I,J)=DMAX1(ZS2(J),ZQ(I,J)/TWO)
60    CONTINUE
C
      ZT(I,1)=ZT(I,2)
      ZT(I,JJ)=ZT(I,JJ1)
      ZQ(I,1)=ZQ(I,2)
      ZQ(I,JJ)=ZQ(I,JJ1)
70    CONTINUE
C
C
      CALL WR (ZT,'TINT')
      CALL WR (ZQ,'QINT')
      CALL BD (E,ML,NL)
      CALL WR (E,'E=A ')
C
C X IMPLICIT
      DO 80 I=2,II1
      ZFTU(I)=(F(I,2)+F(I,1))/2.0D0*(ZT(I,2)-ZT(I,1))
      ZFQU(I)=(F(I,2)+F(I,1))/2.0D0*(ZQ(I,2)-ZQ(I,1))
80    CONTINUE
C
C
      DO 140 J=2,JJ1
C
C
      DO 90 I=2,II1
      ZFTM(I)=ZFTU(I)
      ZFQM(I)=ZFQU(I)
      FA=(F(I,J+1)+F(I,J))/2.0D0
      ZFTU(I)=FA*(ZT(I,J+1)-ZT(I,J))
      ZFQU(I)=FA*(ZQ(I,J+1)-ZQ(I,J))
90    CONTINUE
C
      DO 100 I=1,II1
      DXD(I)=(F(I+1,J)+F(I,J))/2.0D0
100   CONTINUE
C
      DO 110 I=2,II1
      GAM(I)=-DXD(I)*DXM(I)
      ALP(I)=-DXD(I-1)*DXP(I-1)
      BET(I)=1.0D0-GAM(I)-ALP(I)
      SO1(I)=ZT(I,J)+DZM(J)*ZFTU(I)-DZP(J-1)*ZFTM(I)-ADV(I,J)
      SO2(I)=ZQ(I,J)+DZM(J)*ZFQU(I)-DZP(J-1)*ZFQM(I)-AV2(I,J)+E(I,J)*(
     12.*Q(I,J)-S(I,J))
110   CONTINUE
```

```
C
C LEFT
      IF (ZBL(J).GT.ZERO) SO1(2)=SO1(2)-ALP(2)*ZT(1,J)
      IF (ZBL(J).GT.ZERO) SO2(2)=SO2(2)-ALP(2)*ZQ(1,J)
      IF (ZBL(J).EQ.ZERO) BET(2)=BET(2)+ALP(2)
      IF (ZBL(J).LT.ZERO) BET(2)=BET(2)+ALP(2)*2.D0
      IF (ZBL(J).LT.ZERO) GAM(2)=GAM(2)-ALP(2)
C
C RIGHT
      IF (ZBR(J).GT.ZERO) SO1(II1)=SO1(II1)-GAM(II1)*B(J)
      IF (ZBR(J).GT.ZERO) SO2(II1)=SO2(II1)-GAM(II1)*BF(J)
      IF (ZBR(J).EQ.ZERO) BET(II1)=BET(II1)+GAM(II1)
      IF (ZBR(J).LT.ZERO) BET(II1)=BET(II1)+GAM(II1)*2.D0
      IF (ZBR(J).LT.ZERO) ALP(II1)=ALP(II1)-GAM(II1)
C
      CALL TRID1 (ALP,BET,GAM,SO1,ZS1,II)
      DO 120 I=2,II1
      BET(I)=BET(I)+E(I,J)
120   CONTINUE
      CALL TRID1 (ALP,BET,GAM,SO2,ZS2,II)
C
      DO 130 I=2,II1
      ZT(I,J)=ZS1(I)
      ZERO=Q(I,J)/2
      EPS=1.D-20
      ZQ(I,J)=DMAX1(ZS2(I),ZERO,EPS)
130   CONTINUE
140   CONTINUE
      RETURN
      END


      SUBROUTINE FCSTU (U,ZU,W,F,E,ADV,ALP,BET,GAM,DEN,H,SO,ZS,DXP,DZP
     1,ZBL,ZBR,DXM,DZM,DXU,DZU,ML,NL,P,DX2)
C  ***  FORECAST U  ***
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON /N1/ DT,TIME
C
C
      REAL F,E,ADV
      REAL U,ZU,W,P
C
      DIMENSION U(ML,NL), ZU(ML,NL), W(ML,NL), F(ML,NL), E(ML,NL), ADV
     1(ML,NL), ALP(NL), BET(NL), GAM(NL), DEN(NL), H(NL), SO(NL), ZS(NL)
     2, DXP(ML), DZP(NL), DXM(ML), DZM(NL), DXU(ML), DZU(NL), ZBL(NL), Z
     3BR(NL), P(ML,NL), DX2(ML)
      DIMENSION ZL(200)
C
C
      II=ML
      II1=II-1
      II2=II-2
      JJ=NL
      JJ1=JJ-1
      JJ2=JJ-2
C
      DO 10 J=1,JJ1
      DO 10 I=1,II1
      E(I,J)=(F(I+1,J+1)+F(I,J+1)+F(I+1,J)+F(I,J))/4.0
10    CONTINUE
C
C
C Z IMPLICIT
      DO 20 J=2,JJ1
      ZL(J)=ZU(1,J)
20    CONTINUE
C
      DO 50 I=2,II1
      DO 30 J=2,JJ1
      GAM(J)=-DZM(J)*E(I,J)
      ALP(J)=-DZP(J-1)*E(I,J-1)
      BET(J)=1.0D0-GAM(J)-ALP(J)
      DKP=DXP(I)*F(I+1,J)
      DKM=DXM(I)*F(I,J)
      TUX=(U(I+1,J)+U(I,J))*U(I+1,J)-(U(I,J)+U(I-1,J))*U(I-1,J)
      TUZ=(W(I+1,J)+W(I,J))*U(I,J+1)-(W(I+1,J-1)+W(I,J-1))*U(I,J-1)
      ADV(I,J)=DXU(I)*TUX+DZU(J)*TUZ+DT*(P(I+1,J)-P(I,J))/DX2(I)-16.*D
     1XU(I)*DZU(J)/DT*(E(I,J)*(W(I+1,J)-W(I,J))-E(I,J-1)*(W(I+1,J-1)-W(I
     2,J-1)))
```

```
      FX=DKP*ZU(I+1,J)+DKM*ZL(J)-(DKP+DKM)*ZU(I,J)
      SO(J)=ZU(I,J)-ADV(I,J)+FX*2.
      ZL(J)=ZU(I,J)
30    CONTINUE
C
      BET(2)=BET(2)+ALP(2)
      BET(JJ1)=BET(JJ1)+GAM(JJ1)
C
      CALL TRID1 (ALP,BET,GAM,SO,ZS,JJ)
C
      DO 40 J=2,JJ1
      ZU(I,J)=ZS(J)
40    CONTINUE
      ZU(I,1)=ZU(I,2)
      ZU(I,JJ)=ZU(I,JJ1)
50    CONTINUE
C
C
      CALL WR (ZU,'UINT')
C X IMPLICIT
      DO 60 I=2,II1
      ZL(I)=ZU(I,1)
60    CONTINUE
C
      DO 90 J=2,JJ1
      DO 70 I=2,II1
      ALP(I)=-DXM(I)*F(I,J)*2.
      GAM(I)=-DXP(I)*F(I+1,J)*2.
      BET(I)=1.0D0-ALP(I)-GAM(I)
      DKP=DZM(J)*E(I,J)
      DKM=DZP(J-1)*E(I,J-1)
      FZ=DKP*ZU(I,J+1)+DKM*ZL(I)-(DKP+DKM)*ZU(I,J)
      SO(I)=ZU(I,J)-ADV(I,J)+FZ
      ZL(I)=ZU(I,J)
70    CONTINUE
      SO(2)=SO(2)-ALP(2)*ZU(1,J)
      IF (ZBR(J).GE.O.) ALP(III1)=ALP(III1)+GAM(III1)
      IF (ZBR(J).LT.O.) ALP(III1)=ALP(III1)-GAM(III1)
      IF (ZBR(J).LT.O.) BET(III1)=BET(III1)+GAM(III1)*2.DO
C
      CALL TRID1 (ALP,BET,GAM,SO,ZS,II)
C
      DO 80 I=2,II1
      ZU(I,J)=ZS(I)
80    CONTINUE
90    CONTINUE
C
C
      RETURN
      END


      SUBROUTINE FCSTW (W,ZW,U,T,F,E,ADV,ALP,BET,GAM,DEN,H,SO,ZS,DXP,D
     1ZP,Q,XPOR,ZWL,ZWR,DXM,DZM,DXW,DZW,ML,NL,P,DZ2)
C   ***  FORECAST W   ***
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON /N1/ DT,TIME
      COMMON /N2/ COF,RLAM,PI,GAP,DEP,CSTRAT,CW
C
      REAL U,P,W,ZW,Q
      REAL F,E,T,ADV
C
      DIMENSION Q(ML,NL)
      DIMENSION W(ML,NL), ZW(ML,NL), U(ML,NL), T(ML,NL), F(ML,NL), E(M
     1L,NL), ADV(ML,NL), ALP(NL), BET(NL), GAM(NL), DEN(NL), H(NL), SO(N
     2L), ZS(NL), DXP(ML), DZP(NL), DXM(ML), DZM(NL), DXW(ML), DZW(NL),
     3ZWL(NL), ZWR(NL), P(ML,NL), DZ2(NL), XPOR(ML)
      DIMENSION ZL(200)
C
      II=ML
      II1=II-1
      II2=II-2
      JJ=NL
      JJ1=JJ-1
      JJ2=JJ-2
C
      DTH=DT/2.0D0
C
C
```

```
C Z IMPLICIT
      DO 10 J=2,JJ2
      ZL(J)=ZW(1,J)
10    CONTINUE
C
      DO 40 I=2,II1
      DO 20 J=2,JJ2
      GAM(J)=-DZP(J)*F(I,J+1)*2.D0
      ALP(J)=-DZM(J)*F(I,J)*2.D0
      BET(J)=1.0D0-ALP(J)-GAM(J)
      DKP=DXM(I)*E(I,J)
      DKM=DXP(I-1)*E(I-1,J)
      FX=DKP*ZW(I+1,J)+DKM*ZL(J)-(DKP+DKM)*ZW(I,J)
      TUX=(U(I,J+1)+U(I,J))*W(I+1,J)-(U(I-1,J+1)+U(I-1,J))*W(I-1,J)
      TUZ=(W(I,J+1)+W(I,J))*W(I,J+1)-(W(I,J)+W(I,J-1))*W(I,J-1)
      ADV(I,J)=DXW(I)*TUX+DZW(J)*TUZ-DTH*(T(I,J+1)+T(I,J)-T(II,J+1)-T(
     1II,J))+DT*(P(I,J+1)-P(I,J))/DZ2(J)-16.*DXW(I)*DZW(J)/DT*(E(I,J)*(U
     2(I,J+1)-U(I,J))-E(I-1,J)*(U(I-1,J+1)-U(I-1,J)))
      SO(J)=ZW(I,J)-ADV(I,J)+FX-CW*(F(I,J)+F(I,J+1))*(T(I,J+1)-T(I,J))
     1/(Q(I,J)+Q(I,J+1))*DT/DZ2(J)*ZW(I,J)
      ZL(J)=ZW(I,J)
20    CONTINUE
C
      CALL TRIDI (ALP,BET,GAM,SO,ZS,JJ1)
C
      DO 30 J=2,JJ2
      ZW(I,J)=ZS(J)*XPOR(I)
30    CONTINUE
40    CONTINUE
C
C
      CALL WR (ZW,'WINT')
C X IMPLICIT
      DO 50 I=2,II1
      ZL(I)=0.0
50    CONTINUE
C
      DO 80 J=2,JJ2
      DO 60 I=2,II1
      GAM(I)=-DXM(I)*E(I,J)
      ALP(I)=-DXP(I-1)*E(I-1,J)
      BET(I)=1.0D0-ALP(I)-GAM(I)+CW*(F(I,J)+F(I,J+1))*(T(I,J+1)-T(I,J)
     1)/(Q(I,J)+Q(I,J+1))*DT/DZ2(J)
      DKP=DZP(J)*F(I,J+1)
      DKM=DZM(J)*F(I,J)
      FZ=DKP*ZW(I,J+1)+DKM*ZL(I)-(DKP+DKM)*ZW(I,J)
      SO(I)=ZW(I,J)-ADV(I,J)+FZ*2.D0
      ZL(I)=ZW(I,J)
60    CONTINUE
C
C LEFT
      IF (ZWL(J).GT.0.D0) SO(2)=SO(2)-ALP(2)*W(1,J)
      IF (ZWL(J).EQ.0.D0) BET(2)=BET(2)+ALP(2)
      IF (ZWL(J).LT.0.D0) BET(2)=BET(2)+ALP(2)*2.D0
      IF (ZWL(J).LT.0.D0) GAM(2)=GAM(2)-ALP(2)
C
C RIGHT
      IF (ZWR(J).EQ.0.D0) BET(II1)=BET(II1)+GAM(II1)
      IF (ZWR(J).LT.0.D0) BET(II1)=BET(II1)+GAM(II1)*2.D0
      IF (ZWR(J).LT.0.D0) ALP(II1)=ALP(II1)-GAM(II1)
C
      CALL TRIDI (ALP,BET,GAM,SO,ZS,II)
C
      DO 70 I=2,II1
      ZW(I,J)=ZS(I)*XPOR(I)
70    CONTINUE
80    CONTINUE
C
      RETURN
      END


      SUBROUTINE TRIDI (A,B,C,S,Z,N)
C  ***  TRIDIAGONAL EQUATION SOLVER  ***
C
C SOLVES TRIDIAGONAL SYSTEM, I=2 TO N-1.
C A,B,C UNCHANGED, RHS S DESTROYED, SOLUTION IN Z.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(N), B(N), C(N), S(N), Z(N)
```

```
      NM=N-1
      D=B(2)
      S(2)=S(2)/D
      DO 10 I=3,NM
      Z(I-1)=C(I-1)/D
      D=B(I)-A(I)*Z(I-1)
10    S(I)=(S(I)-A(I)*S(I-1))/D
      Z(NM)=S(NM)
      DO 20 J=4,N
      I=N+2-J
20    Z(I)=S(I)-Z(I)*Z(I+1)
      RETURN
      END


      SUBROUTINE ROPT
C  ***  INITIALIZE POISSON SOLVER  ***
      IMPLICIT REAL*8(A-H,P-Z)
C
C
      COMMON /N4/ AAR,AAZ,KAPPA
      COMMON /N5/ RR(64),LL
      DIMENSION ALP(8), BET(8), D(50)
C
C
      ALP(1)=AAR
      BET(1)=AAZ
      RA=BET(1)/ALP(1)
      DO 10 I=1,KAPPA
      ALP(I+1)=(ALP(I)*BET(I))**.5D0
      BET(I+1)=(ALP(I)+BET(I))*.5D0
10    CONTINUE
      AL=ALP(KAPPA+1)
      BE=BET(KAPPA+1)
C
C NEWTON'S ITERATION FOR R, FOR EXTRAPOLATED INITIAL APPROXIMATION
C
      RK=DMIN1(2*AL,(AL*BE)**.5D0)
      DO 20 J=1,10
      AK=(BE-RK)/(BE+RK)
      A=(BE-RK)/(BE+RK)
      BK=(RK-AL)/(RK+AL)
      B=(1-A)**1.5D0/A+3-1/BK
      AD=-2*BE/(BE+RK)**2
      BD=2*AL/(AL+RK)**2
      FD=BD/BK/BK-(1-AK)**.5D0*(1+AK/2)/AK/AK*AD
C4    R=(AL*BE)**.5D0     IF NOT EXTRAPOLATING
20    RK=RK-B/FD
      S=1/(1/BK-3)
      AS=AK
C
C FIND ALL KK R VALUES
C RR(K) ARE THE ITERATION PARAMETERS
C
      LL=2**KAPPA
      RR(1)=RK
      K1=LL
      DO 40 I=1,KAPPA
      J=KAPPA+1-I
      AL=ALP(J)
      BE=BET(J)
      K2=K1/2
      DO 30 J=1,LL,K1
      RR(J)=RR(J)+(RR(J)**2-AL*BE)**.5D0
30    RR(J+K2)=AL*BE/RR(J)
40    K1=K2
      RETURN
      END


      SUBROUTINE POISB (G,S1,S2,A1,A2,A3,B1,B2,B3,WI,WK,II,KK)
C  ***  POISSON EQUATION SOLVER  ***
      IMPLICIT REAL*8(A-H,O-Z)
C
      REAL G,S1,S2
C FIXED BY GJR. P(II)=-P(II)
C
      COMMON /N5/ RR(64),LIM
      DIMENSION G(II,1,KK), S1(II,1,KK), S2(II,1,KK)
      DIMENSION A1(II), A2(II), A3(II), B1(KK), B2(KK), B3(KK), WI(KK)
```

```
     1, WK(KK)
      J=1
      N=1
      II1=II-1
      II2=II-2
      KK1=KK-1
      KK2=KK-2
      DO 10 I=2,II1
      G(I,1,1)=G(I,1,2)
      G(I,1,KK)=G(I,1,KK1)
10    CONTINUE
      DO 160 L=1,LIM
      DO 20 I=2,II1
      DO 20 K=2,KK1
      S2(I,1,K)=(RR(L)-B3(K))*G(I,1,K)+B1(K)*G(I,1,K+1)+B2(K)*G(I,1,K-
     11)-S1(I,1,K)
20    CONTINUE
      BBB1=+A3(2)+RR(L)-A2(2)
      DBBB1=1.0/BBB1
      WI(2)=-A1(2)*DBBB1
      DO 50 K=2,KK1
      G(2,1,K)=S2(2,1,K)*DBBB1
      DO 30 I=3,II2
      BAB=+A3(I)+RR(L)
      DENA=1.0/(BAB+A2(I)*WI(I-1))
      WI(I)=-A1(I)*DENA
30    G(I,1,K)=(S2(I,1,K)+A2(I)*G(I-1,1,K))*DENA
      BBB2=+A3(II1)+RR(L)+A1(II1)
      G(II1,1,K)=(S2(II1,1,K)+A2(II1)*G(II2,1,K))/(BBB2+A2(II1)*WI(II2
     1))
      S2(II1,1,K)=G(II1,1,K)
      DO 40 IL=2,II2
      I=II2-IL+2
      S2(I,1,K)=G(I,1,K)-WI(I)*S2(I+1,1,K)
40    CONTINUE
50    CONTINUE
      DO 60 I=2,II1
      DO 60 K=2,KK1
      G(I,1,K)=S2(I,1,K)
60    CONTINUE
      DO 70 I=2,II1
      G(I,N,1)=G(I,1,2)
      G(I,N,KK)=G(I,1,KK1)
70    CONTINUE
      DO 80 K=2,KK1
      G(1,1,K)=G(2,1,K)
      G(II,1,K)=-G(II1,1,K)
80    CONTINUE
      DO 90 K=2,KK1
      DO 90 I=2,II1
      S2(I,1,K)=(RR(L)-A3(I))*G(I,1,K)+A1(I)*G(I+1,1,K)+A2(I)*G(I-1,1,
     1K)-S1(I,1,K)
90    CONTINUE
      BBB1=+B3(2)+RR(L)-B2(2)
      DBBB1=1.0/BBB1
      WK(2)=-B1(2)*DBBB1
      DO 120 I=2,II1
      G(I,1,2)=S2(I,1,2)*DBBB1
      DO 100 K=3,KK2
      BAB=+B3(K)+RR(L)
      DENA=1.0/(BAB+B2(K)*WK(K-1))
      WK(K)=-B1(K)*DENA
100   G(I,1,K)=(S2(I,1,K)+B2(K)*G(I,1,K-1))*DENA
      BBB2=+B3(KK1)+RR(L)-B1(KK1)
      G(I,1,KK1)=(S2(I,1,KK1)+B2(KK1)*G(I,1,KK2))/(BBB2+B2(KK1)*WK(KK2
     1))
      S2(I,1,KK1)=G(I,1,KK1)
      DO 110 KL=2,KK2
      K=KK2-KL+2
      S2(I,1,K)=G(I,1,K)-WK(K)*S2(I,1,K+1)
110   CONTINUE
120   CONTINUE
      DO 130 I=2,II1
      DO 130 K=2,KK1
      G(I,1,K)=S2(I,1,K)
130   CONTINUE
      DO 140 I=2,II1
      G(I,1,1)=G(I,1,2)
      G(I,1,KK)=G(I,1,KK1)
140   CONTINUE
```

```
      DO 150 K=2,KK1
      G(1,N,K)=G(2,N,K)
      G(II,N,K)=-G(III1,N,K)
150   CONTINUE
160   CONTINUE
      RETURN
      END


      SUBROUTINE TSTEP (ZU,ZW,P,DXG,DZG,ID,JD,SIGN)
C *** MODIFY FLOW BY PRESSURE GRADIENT  ***
      IMPLICIT REAL*8(A-H,O-Z)
C
      REAL ZU,ZW,P
C
      DIMENSION ZU(ID,JD), ZW(ID,JD), P(ID,JD), DXG(ID), DZG(JD)
C
      II=ID
      JJ=JD
      III1=II-1
      II2=II-2
      JJ1=JJ-1
      JJ2=JJ-2
C
C
      DO 10 J=2,JJ1
      DO 10 I=2,III1
      ZU(I,J)=ZU(I,J)-DXG(I)*(P(I+1,J)-P(I,J))*SIGN
10    CONTINUE
C
      DO 20 J=2,JJ2
      DO 20 I=2,III1
      ZW(I,J)=ZW(I,J)-DZG(J)*(P(I,J+1)-P(I,J))*SIGN
20    CONTINUE
C
      DO 30 I=2,II2
      ZU(I,1)=ZU(I,2)
      ZU(I,JJ)=ZU(I,JJ1)
30    CONTINUE
C
C
      DO 60 J=2,JJ2
C
      IF (ZU(1,J)) 50,50,40
C
40    CONTINUE
C
      ZW(1,J)=0.0
      ZW(II,J)=ZW(III1,J)
C
50    CONTINUE
C
      ZW(1,J)=ZW(2,J)
      ZW(II,J)=ZW(III1,J)
C
60    CONTINUE
C
C
      RETURN
      END


      SUBROUTINE BOUND (U,W,Q,T,ZBL,ZBR,ZWL,ZWR,B,BF,II,JJ,DZ1)
C *** APPLIES BOUNDARY CONDITIONS TO U,W,Q AND T  ***
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /N2/ COF,RLAM,PI,GAP,DEP,CSTRAT
      DIMENSION U(II,JJ), W(II,JJ), T(II,JJ), Q(II,JJ), ZBL(JJ), ZWL(J
     1J), ZBR(JJ), ZWR(JJ), B(JJ), BF(JJ), DZ1(JJ)
      REAL U,W,Q,T
      JJ1=JJ-1
      III1=II-1
      II2=II-2
      ZERO=0
      U(III1,1)=U(III1,2)
      U(III1,JJ)=U(III1,JJ1)
      T(III1,1)=T(III1,2)
      T(III1,JJ)=T(III1,JJ1)
C
C SIDES
      DO 10 J=2,JJ1
```

```
C
C LEFT
      IF (ZBL(J).EQ.0.) Q(1,J)=Q(2,J)
      IF (ZBL(J).EQ.0.) T(1,J)=T(2,J)
      IF (ZWL(J).EQ.0.) W(1,J)=W(2,J)
      IF (ZBL(J).LT.0.) T(1,J)=T(2,J)*2.-T(3,J)
      IF (ZBL(J).LT.0.) Q(1,J)=Q(2,J)*2.-Q(3,J)
      IF (ZBL(J).LT.0.) Q(1,J)=AMAX1(Q(1,J),Q(2,J)/4.)
      IF (ZWL(J).LT.0.) W(1,J)=W(2,J)*2.-W(3,J)
C
C RIGHT
      A=DMAX1(((U(III,J+1)-U(III,J-1))/DZ1(J)/2)**2-(T(III,J+1)-T(III,
     1J-1))/DZ1(J)/2,ZERO)
      IF (ZBR(J).GT.0.) Q(II,J)=BF(J)+A*RLAM*RLAM/COF
      IF (ZBR(J).GT.0.) T(II,J)=B(J)
      IF (ZWR(J).GT.0.) W(II,J)=0.
      IF (ZBR(J).GE.0.) U(II,J)=U(II2,J)
      IF (ZWR(J).EQ.0.) W(II,J)=W(III,J)
      IF (ZBR(J).EQ.0.) Q(II,J)=Q(III,J)
      IF (ZBR(J).EQ.0.) T(II,J)=T(III,J)
      IF (ZBR(J).LT.0.) T(II,J)=T(III,J)*2.-T(II2,J)
      IF (ZBR(J).LT.0.) Q(II,J)=Q(III,J)*2.-Q(II2,J)
      IF (ZBR(J).LT.0.) Q(II,J)=AMAX1(Q(II,J),Q(III,J)/4.)
      IF (ZBR(J).LT.0.) U(II,J)=U(III,J)*2.-U(II2,J)
      IF (ZWR(J).LT.0.) W(II,J)=W(III,J)*2.-W(II2,J)
10    CONTINUE
C
C TOP AND BOTTOM
      DO 20 I=1,II
      U(I,1)=U(I,2)
      T(I,1)=T(I,2)
      Q(I,1)=Q(I,2)
      U(I,JJ)=U(I,JJ1)
      T(I,JJ)=T(I,JJ1)
      Q(I,JJ)=Q(I,JJ1)
20    CONTINUE
      RETURN
      END


      SUBROUTINE OUTPUT (U,ZU,W,ZW,T,Q,F,E,S,ADV,R,X,Z,Y,DX1,DZ1,RLAM,
     1DT,TIME,GAP,DEP,II,JJ,NC,NK,NKM,I2,J1,B,DX2,DZ2,TT,TB,ZQ,DSCALE,NN
     2,TIN,WIDFT,G,H,NCOPYS)
C  ***  OUTPUT ROUTINE CALLED EVERY TIME STEP   ***
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /N2/ COF,XXXX,PI,YYY,ZZZ,CSTRAT
      COMMON /TXTC/ ITXT(60),DAY,JTXT(60)
      COMMON /STREAM/ IS
      REAL U,ZU,W,ZW,S,P,ZQ
      REAL T,Q,E,F,ADV,G,H
      DIMENSION F(II,JJ), E(II,JJ), G(II,JJ), H(II,JJ)
      DIMENSION U(II,JJ), W(II,JJ), T(II,JJ), Q(II,JJ), S(II,JJ), R(II
     1), X(II), Z(JJ), Y(JJ), DX1(II), DZ1(JJ), ZU(II,JJ), ZW(II,JJ), B(
     2JJ), DX2(II), DZ2(JJ), ZQ(II,JJ)
      DIMENSION AL(4), AR(4), ICH(4), ADV(JJ,II)
      DATA ICH/1HT,1HU,1HA,1HT/
      DIMENSION F1(75), F2(75), F3(75), F4(75), F5(75), F6(75)
      DIMENSION IFROM(3), ICHFR(3)
      DIMENSION SX(8), TIN(2), TX(3)
      DIMENSION ITSAR(14), TAR(5)
      DATA ITSAR/1,2,3,4,6,8,12,18,24,36,48,72,96,120/
C
      III=II-1
      II2=II-2
      JJ1=JJ-1
      JJ2=JJ-2
      TD=TT-TB
C
      DO 10 I=1,8
10    SX(I)=0.
      SX(6)=1.D10
      SX(7)=1.D10
      SX(8)=1.D10
      SUM7=0.0
C
      DO 40 J=1,JJ1
      DO 40 I=1,III
      TX(1)=T(I,J)
      TX(2)=Q(I,J)
      TX(3)=ZQ(I,J)
```

```
      TX(4)=U(I,J)
      TX(5)=W(I,J)
      TX(4)=DABS(TX(4))
      TX(5)=DABS(TX(5))
      DO 20 K=1,5
20    SX(K)=DMAX1(SX(K),TX(K))
      DO 30 K=6,8
      SX(K)=DMIN1(SX(K),TX(K-5))
30    CONTINUE
40    CONTINUE
      DO 50 J=2,JJ1
      DO 50 I=2,II1
      D=(U(I,J)-U(I-1,J))/DX1(I)+(W(I,J)-W(I,J-1))/DZ1(J)
      SUM7=SUM7+D*D
50    CONTINUE
      DIV=SUM7
C
C
      TIN(1)=0
      TIN(2)=0
      A=0
      C=0
      IT=1
      JT=0
      DO 70 J=2,JJ1
      IF (U(1,J).GE.0.) GO TO 60
      JT=1
      A=A+U(1,J)*(T(2,J)*1.5-T(3,J)*.5)*DZ1(J)
      C=C+U(1,J)*DZ1(J)
      IF (J.EQ.JJ1) GO TO 60
      GO TO 70
60    IF (JT.EQ.0) GO TO 70
      TIN(IT)=A/C*TD+TB
      JT=0
      A=0
      C=0
      IT=IT+1
      IF (IT.GE.3) GO TO 80
70    CONTINUE
80    CONTINUE
C
      IF (NK.EQ.2) PRINT 90
      IF (NC.EQ.0) PRINT 90
90    FORMAT (/'   NC    TMAX         QMAX        ZQMAX        UMAX         WMA
     1X         TMIN         QMIN         ZQMIN        DIV          TIN1         TIN2
     2'/)
C
      PRINT 100, NC,SX,DIV,TIN
100   FORMAT (I6,1P,11D11.4)
      IF (NK.EQ.NKM) PRINT 90
C
C
      IF (NK-NKM) 480,110,110
C
C
110   CONTINUE
      NK=0
C
C
C WRITE ANSWERS AT END OF STREAM 10 FOR PRINTING
C
      REWIND 10
      READ (10,120)
120   FORMAT (29(/))
C
      IF (TIN(1).EQ.0) GO TO 150
      IF (TIN(2).EQ.0) WRITE (10,130) TIN(1)
      IF (TIN(2).NE.0) WRITE (10,140) TIN
130   FORMAT (10X,'MEAN PLANT INFLOW TEMPERATURE IS',F5.1)
140   FORMAT (6X,'MEAN PLANT INFLOW TEMPERATURES ARE',2F6.1)
150   CONTINUE
C
C
C SCALES IN FEET
      DO 160 I=1,II
      X(I)=X(I)*DSCALE
      R(I)=R(I)*DSCALE
160   CONTINUE
      DO 170 J=1,JJ
      Z(J)=Z(J)*DSCALE-DSCALE
```

```
      Y(J)=Y(J)*DSCALE-DSCALE
170   CONTINUE
C
C XPLOTS
C
      PRINT 470
      ISEP=1
      JLINE=25
C ENVIRONMENTAL IMPACT PLOT
      USC=DSQRT(DSCALE*TD*32*1.3E-4)
      F2(1)=0
      DO 180 J=1,JJ1
      F1(J)=(T(II,J)+T(II,J+1))/2.*TD+TB
180   F2(J+1)=F2(J)+U(II1,J+1)*DZ2(J+1)*DSCALE*WIDFT*USC
      TRT=F1(JJ1)
      TRB=F1(1)
      CALL XPLOT (F2,F1,JJ1,1,30,TRB,TRT,'CUFT/S ADDED BELOW T   **','T
     1IME =',TIME)
C
C
      XL=0.0
      XU=GAP*DSCALE
C
      DO 190 I=1,II
      F1(I)=T(I,JJ1)*TD+TB
190   CONTINUE
      CALL XPLOT (F1,R,II,ISEP,JLINE,XL,XU,'T(I,JJ1)                   ',
     1'TIME =',TIME)
C
C DOUBLE PRINT COUNT AT END.
      IP=1
      IF (NC.GE.NN) IP=2
      DO 250 IPC=1,IP
      IS=6
      PRINT 470
      NCONT=12
      IF (IPC.EQ.2) NCONT=20
      ISEP=1
      NLIN=-1
      AMAG=1
      ZER=0
      XU=GAP*DSCALE/IPC
      ZL=-DSCALE
C
      CALL CONTOR (Q,R,Z,II,JJ,NCONT,ISEP,NLIN,AMAG,ZER,XU,ZL,ZER,'TUR
     1BULENCE ISOLINES  **   ','TIME  ',TIME)
C
      CALL CONTOR (F,R,Z,II,JJ,NCONT,ISEP,NLIN,AMAG,ZER,XU,ZL,ZER,'TUR
     1BULENT DIFFUSIVITY  **','TIME  ',TIME)
C
C
      DO 200 I=2,II1
      DO 200 J=2,JJ1
200   E(I,J)=CSTRAT*RLAM*RLAM*(T(I,J+1)-T(I,J-1))/2./DZ1(J)/Q(I,J)
      CALL BD (E,II,JJ)
C
      CALL CONTOR (E,R,Z,II,JJ,NCONT,ISEP,NLIN,AMAG,ZER,XU,ZL,ZER,'CST
     1RAT * N**2 * L**2 / Q ','TIME  ',TIME)
      DO 220 I=2,II1
      DO 210 J=2,JJ1
      AA=(U(I,J)-U(I-1,J))/DX1(I)
      AB=(U(I,J+1)+U(I-1,J+1)-U(I,J-1)-U(I-1,J-1))/DZ1(J)+(W(I+1,J)+W(
     1I+1,J-1)-W(I-1,J)-W(I-1,J-1))/DX1(I)
      S(I,J)=4.*AA*AA+AB*AB/16+1.D-10
      A=S(I,J)
      E(I,J)=DSQRT(A)/(F(I,J)+1.E-20)
      S(I,J)=(T(I,J+1)-T(I,J-1))/2./DZ1(J)/S(I,J)
210   CONTINUE
220   CONTINUE
C
      CALL BD (S,II,JJ)
      CALL BD (E,II,JJ)
C
      CALL CONTOR (S,R,Z,II,JJ,NCONT,ISEP,NLIN,AMAG,ZER,XU,ZL,ZER,'RIC
     1HARDSON NUMBER  **    ','TIME  ',TIME)
C
      CALL CONTOR (E,R,Z,II,JJ,NCONT,ISEP,NLIN,AMAG,ZER,XU,ZL,ZER,' RE
     1YNOLDS  NUMBER  **    ','TIME  ',TIME)
C
      DO 230 J=1,JJ
```

```
      DO 230 I=1,II
230   S(I,J)=TB+TD*T(I,J)
      CALL CONTOR (S,R,Z,II,JJ,NCONT,ISEP,NLIN,AMAG,ZER,XU,ZL,ZER,'TEM
     1PERATURE  **            ','TIME  ',TIME)
C
C
      DO 240 I=1,II
      E(I,1)=0.0
      SUM=0.0
      DO 240 J=2,JJ
      SUM=SUM+U(I,J)*DZ1(J)
      E(I,J)=SUM
240   CONTINUE
C
      CALL CONTOR (E,X,Y,II,JJ,NCONT,ISEP,NLIN,AMAG,ZER,XU,ZL,ZER,'STR
     1EAM LINES  **           ','TIME  ',TIME)
C
C
250   CONTINUE
C
C
C VPLOT RANGES.
      I=SX(4)*12*USC+.9
      USCP=I/10.
      USCM=-USCP
      DO 260 J=1,JJ
260   F1(J)=T(1,J)
      CALL MAXMIN (F1,Z,JJ,TMX,TMN,TAB,ZMX,ZMN,ZAB)
      IF (TMX.LT.1) TMX=1
      IF (TMN.GT.0) TMN=0
      ITL=TB+TD*TMN+.01
      ITU=TB+TD*TMX+.99
      ITD=ITU-ITL
      DO 270 I=1,14
      ITS=ITSAR(I)
      IF (ITD.LE.ITS) GO TO 280
270   CONTINUE
280   CONTINUE
      ITL=ITL-(ITS-ITD)/2
      ITU=ITL+ITS
      DO 290 I=1,5
290   TAR(I)=ITL+ITS*(I-1)/4.
C
C VPLOT CALL LOOP.
      DO 340 I=1,II1,II2
      DO 300 J=1,JJ
      ADV(J,1)=T(I,J)
      ADV(J,2)=U(I,J)
      ADV(J,3)=B(J)
300   CONTINUE
C
      AL(2)=-USCP/USC
      AR(2)=USCP/USC
      AL(3)=(ITL-TB)/TD
      AR(3)=(ITU-TB)/TD
      AL(1)=AL(3)
      AR(1)=AR(3)
C
      DO 330 IPC=1,2
      IS=9
      IF (I.EQ.1) IS=8
      IF (IPC.EQ.2) IS=IS+3
      CALL VPLOT (ADV,JJ,3,Z,ZL,ZER,ICH,AL,AR,5*IPC,7,4*IPC,6,IS)
      IF (IPC.EQ.1) WRITE (IS,310) TAR,USCM,USCP
      IF (IPC.EQ.2) WRITE (IS,320) TAR,USCM,USCP
310   FORMAT (/' DEG',F6.1,3F6.1,F5.1/' FT/S',F5.1,9X,'0.0',F11.1)
320   FORMAT (/' DEG',F6.1,3F12.1,F11.1/' FT/S',F5.1,21X,'0.0',F23.1)
330   CONTINUE
340   CONTINUE
C
      IFROM(1)=3
      IFROM(2)=10
      IFROM(3)=9
      ICHFR(1)=36
      ICHFR(2)=60
      ICHFR(3)=32
C
C
C START MULTIPLE COPY FORMAL OUTPUT.
      IS=6
```

```
      IF (NC.EQ.NN.AND.NCOPYS.GT.1) IS=13
C
      WRITE (IS,470)
C
      NCONT=0
      XU=GAP*DSCALE
C
      DO 350 J=1,JJ
      DO 350 I=1,II
      F(I,J)=F(I,J)*USC*DSCALE
350   S(I,J)=TB+TD*T(I,J)
      CALL CONTOR (S,R,Z,II,JJ,NCONT,ISEP,NLIN,AMAG,ZER,XU,ZL,ZER,'TEM
     1PERATURE  **          ','TIME  ',TIME)
C
C
      DO 360 I=1,II
      E(I,1)=0.0
      SUM=0.0
      DO 360 J=2,JJ
      SUM=SUM+U(I,J)*DZ1(J)*USC*DSCALE*WIDFT
      E(I,J)=SUM
360   CONTINUE
C
      J1=JJ1
      CALL CONTOR (E,X,Y,II,J1,NCONT,ISEP,NLIN,AMAG,ZER,XU,ZL,ZER,'STR
     1EAM LINES (CUFT/S)  **','TIME  ',TIME)
C
      CALL COMBIN (42,IS,132,3,IFROM,ICHFR)
C END FIRST PAGE PAIR
C
      WRITE (IS,470)
C
      CALL CONTOR (F,R,Z,II,JJ,NCONT,ISEP,NLIN,AMAG,ZER,XU,ZL,ZER,'DIF
     1FUSIVITY (FT**2/S)  **','TIME  ',TIME)
C
      WRITE (IS,370)
370   FORMAT (11X,'**  TEMPERATURE T AND HORIZONTAL MOTION U AT LEFT A
     1ND RIGHT BOUNDARIES, WITH THE AMBIENT TEMPERATURE A  **'/)
      IFROM(1)=11
      IFROM(2)=12
      ICHFR(1)=72
      ICHFR(2)=56
      CALL COMBIN (75,IS,132,2,IFROM,ICHFR)
C
      REWIND 10
      DO 380 I=1,11
      READ (10,410) TX
380   WRITE (3,410) TX
      DO 390 I=1,17
      READ (10,410) TX
390   WRITE (3,410) TX
      DO 400 I=1,6
      READ (10,410,END=420) TX
400   WRITE (3,410) TX
410   FORMAT (1X,8A8)
420   CONTINUE
C
      IFROM(1)=8
      IFROM(2)=9
      ICHFR(1)=68
      ICHFR(2)=60
      CALL COMBIN (20,IS,132,2,IFROM,ICHFR)
C
      IF (IS.EQ.6) GO TO 440
      DO 430 I=1,NCOPYS
430   CALL COMBIN (285,6,128,1,13,128)
      IS=6
440   CONTINUE
C
      PRINT 470
C   END MULTIPLE COPY FORMAL OUTPUT.
C
C RETURN TO DIMENSIONLESS SCALES
      DO 450 I=1,II
      X(I)=X(I)/DSCALE
      R(I)=R(I)/DSCALE
450   CONTINUE
      DO 460 J=1,JJ
      Y(J)=Y(J)/DSCALE+1
      Z(J)=Z(J)/DSCALE+1
```

```
460       CONTINUE
C
470       FORMAT (1H1////)
C
480       CONTINUE
C
          RETURN
          END


          SUBROUTINE CONTOR (O,XV,YQ,NX,NY,NCONT,ISEP,NLIN,AMAG,XL,XU,YL,Y
     1U,TITLE,SUBTIT,VARI)
C
C       *** GLYN'S LINE CONTOUR ROUTINE   *****
C O(I,J) IS VALUE AT (XV(I),YQ(J)), WITH XV AND YQ MONOTONIC INCREASING
C DIMENSIONS ARE NX AND NY
C APPROXIMATELY NCONT CONTOURS ARE DRAWN
C OUTPUT SPLIT INTO ISEP PARALLEL PAGES
C VERTICAL SCALE DETERMINED BY APPROXIMATELY NLIN HORIZONTAL ROWS,
C IF NLIN.LT.0 , PHYSICAL SCALING, WITH VERTICAL STRETCHED BY AMAG
C CONTOURS REGION  XL.LE.X.LE.XU ,  YL.LE.Y.LE.YU
C TITLE IS 24 CHARACTERS AT 8 PER WORD.
C SUBTIT IS 8 CHARACTERS, E.G. *TIME =  * , WITH VARI THE TIME VALUE.
C MAIN PROGRAM SHOULD CONTAIN --
C       COMMON /TXTC/ITXT(60),DAY
C       READ(5,1)ITXT
C1      FORMAT(60A1)
C       CALL IDAY(DAY)
C -- TO READ FURTHER TEXT OFF A CARD AND SET DATE IN DAY.
C BLANK COMMON IS USED FOR WORKING SPACE (962 WORDS) AND SO PASSED
C  ARRAYS SHOULD NOT BE IN BLANK COMMON.
C
          IMPLICIT REAL*8(A-H,O-Z)
C
          DIMENSION IC(120,2)
          DIMENSION LG(121), CHAR(60), TITLE(3), LD(121)
          COMMON /TXTC/ ITXT(60),DAY
          COMMON IC,LD,LC,G,H
          COMMON /STREAM/ IS
          INTEGER BLANK,MINUS,PLUS,BAR,CHAR
          DIMENSION XV(NX), YQ(NY), G(120,4), XI(11)
          REAL F,G,IC,H(4)
          REAL O(NX,NY)
          YV(J)=YQ(J)
          F(I,J)=O(I,J)
C STATEMENT FUNCTION CAN BE USED TO TURN A OVER
          DATA BLANK/1H /,MINUS/1H-/,PLUS/1H+/
          DATA BAR/1H|/
C       DATA BAR/1HI/
          DATA CHAR/1HT,1HS,1HR,1HQ,1HP,1HO,1HN,1HM,1HL,1HK,1HJ,1HI,1HH,1H
     1G,1HF,1HE,1HD,1HC,1HB,1HA,1HO,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,
     21HO,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1HO,1H1,1H2,1H3,1H4,1H5,1H
     36,1H7,1H8,1H9,1HO,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
          YG=YU-YL
C ******************************************************************
          IF (ISEP.GT.4) ISEP=(ISEP+1)/4*4
C ******************************************************************
          XG=(XU-XL)/ISEP
          DO 390 ISP=1,ISEP
          XB=XL+XG*ISP-XG
          DO 10 I=1,121
          LD(I)=BLANK
10        LG(I)=MINUS
          DO 20 I=1,121,12
20        LG(I)=PLUS
          DO 30 I=1,11
30        XI(I)=XB+XG*(I-1)/10
C
C       FIND MAXIMUM, MINIMUM, AND CONTOUR ORIGIN AND INCREMENT
          FMX=F(1,1)
          FMN=FMX
          DO 40 J=1,NY
          DO 40 I=1,NX
          FV=F(I,J)
          FMX=DMAX1(FMX,FV)
40        FMN=DMIN1(FMN,FV)
          IF (FMX.EQ.FMN) WRITE (IS,400) TITLE,SUBTIT,VARI,FMX
          IF (FMX.EQ.FMN) RETURN
C  NOTE FORMAT CHANGED FROM D TO G IN INDEPENDENT CHANGE.
          A=DLOG10((FMX-FMN)/NCONT)+.1505
```

```
      I=A
      IF (A.LT.0) I=I-1
      B=I
      C=A-I
      FINC=10.**B
      IF (C.LT..1505) GO TO 50
      FINC=10.**B*1.5
      IF (C.LT..3010) GO TO 50
      FINC=10.**B*2
      IF (C.LT..4516) GO TO 50
      FINC=10.**B*3
      IF (C.LT..6505) GO TO 50
      FINC=10.**B*5
      IF (C.LT..8490) GO TO 50
      FINC=10.**B*7.5
50    CONTINUE
      J=FMN/FINC
      K=FMX/FINC
      I=J
      IF (K-J.GT.25) I=(J+K)/2
      IF (K.LT.26.AND.J.GT.-26) I=0
      FOR=I*FINC
C
      IF (XT.GT.XV(NX).OR.XB.LT.XV(1)) GO TO 110
      IF (YU.GT.YV(NY).OR.YL.LT.YV(1)) GO TO 110
      JB=1
      IB=1
      DO 60 J=1,NY
      IF (YV(J).LT.YL) JB=J+1
      IF (YV(J).GE.YU) GO TO 70
60    CONTINUE
70    JT=J-1
      XT=XB+XG
      DO 80 I=1,NX
      IF (XV(I).LE.XB) IB=I+1
      IF (XV(I).GE.XT) GO TO 90
80    CONTINUE
90    IT=I-1
      DO 100 J=JB,JT
      DO 100 I=IB,IT
      IF (DABS(F(I,J)-FOR)/FINC.GE..5) GO TO 110
100   CONTINUE
      WRITE (IS,410) TITLE,SUBTIT,VARI,XB,XT,YL,YU
      GO TO 390
110   CONTINUE
C
      NLINE=NLIN
      IL=NLINE/7.+.5
      IF (NLINE.LE.3) IL=72*YG/XG*AMAG/7+.5
      IL=MAX0(IL,1)
      NLINE=7*IL
      CALL SHORT7 (YU,FR,IX)
      WRITE (IS,420) TITLE,ITXT,SUBTIT,VARI,FMX,FMN,FOR,FINC,DAY,XI
      WRITE (IS,450) YU,LG
C
C     SET UP G
      DO 120 I=1,120
      IC(I,2)=1
120   CONTINUE
      NYU=3
      NXL=1
      DO 130 J=1,NY
130   IF (YU.GT.YV(J)) NYU=J+1
      DO 140 I=1,NX
140   IF (XB.GT.XV(I)) NXL=I
      J=0
      JY=MIN0(NY+1,NYU+2)
      MM=1
150   DO 210 M=MM,4
      N=JY-M
      IX=MAX0(0,NXL-2)
      IX=MIN0(IX,NX-4)
      DO 200 I=1,120
      X=XB+(I-.5)*XG/120
160   IF (X.LE.XV(IX+3).OR.IX+4.EQ.NX) GO TO 170
      IX=IX+1
      GO TO 160
170   E=0
      DO 190 K=1,4
      D=F(IX+K,N)
```

```
      DO 190 L=1,4
      IF (L.EQ.K) GO TO 180
      D=D*(X-XV(IX+L))/(XV(IX+K)-XV(IX+L))
180   CONTINUE
190   E=E+D
200   G(I,M)=E
      JJ=J
      IF (J.NE.0) GO TO 220
210   CONTINUE
C
C     MAIN PROGRAM
      JJ=1
220   DO 360 J=JJ,NLINE
      IERC=0
      Y=YU-YG*(J-.5)/NLINE
      IF (Y.GE.YV(JY-3).OR.JY.EQ.5) GO TO 240
      JY=JY-1
      DO 230 K=1,3
      DO 230 I=1,120
230   G(I,K)=G(I,K+1)
      MM=4
      GO TO 150
240   DO 270 I=1,120
      E=0
      DO 260 K=1,4
      D=G(I,K)
      DO 250 L=1,4
      IF (L.EQ.K) GO TO 250
      D=D*(Y-YV(JY-L))/(YV(JY-K)-YV(JY-L))
250   CONTINUE
260   E=E+D
      IC(I,1)=IC(I,2)
      IC(I,2)=(E-FOR)/FINC+22
270   CONTINUE
      IF (J.EQ.1) GO TO 360
      IQ=0
      DO 310 I=2,120
      LG(I)=BLANK
      H(1)=IC(I-1,1)
      H(2)=IC(I-1,2)
      H(3)=IC(I,2)
      H(4)=IC(I,1)
      DO 280 K=1,3
      M=1+K
      DO 280 L=M,4
      IF (H(K).LE.H(L)) GO TO 280
      HH=H(K)
      H(K)=H(L)
      H(L)=HH
280   CONTINUE
      IL=(H(1)+H(2))/2.
      IU=(H(3)+H(4))/2.-1
      IF (IU.LE.60.AND.IL.GE.1) GO TO 300
      IF (IERC.LT.1) WRITE (IS,290) IU,IL,I,E
      IERC=1
290   FORMAT (' IU,IL,I,E= ',3I14,D14.4)
      GO TO 310
300   CONTINUE
      IF (IL.GT.IU) GO TO 310
      LG(I)=CHAR(IL)
      IF (IL.EQ.IU) GO TO 310
      LD(I)=CHAR(IU)
      IQ=1
310   CONTINUE
C
C     PRINT LINE, POSSIBLY WITH Y VALUE
      JLINE=J-1
      IF (JLINE/7*7.NE.JLINE) GO TO 320
      Y=YU-YG*JLINE/NLINE
      CALL SHORT7 (Y,FR,IX)
      WRITE (IS,450) Y,LG
      GO TO 330
320   LG(1)=BAR
      LG(121)=BAR
      WRITE (IS,430) LG
      LG(1)=PLUS
      LG(121)=PLUS
330   IF (IQ.EQ.0) GO TO 350
      IQ=0
      WRITE (IS,440) LD
```

```
      DO 340 I=1,121
340   LD(I)=BLANK
350   CONTINUE
360   CONTINUE
C
C     BOTTOM UNDERLINE
      DO 370 I=1,121
370   LG(I)=MINUS
      DO 380 I=1,121,12
380   LG(I)=PLUS
      CALL SHORT7 (YL,FR,IX)
      WRITE (IS,450) YL,LG
      WRITE (IS,460) XI
390   CONTINUE
      CALL MS (O,NX,NY)
      RETURN
C
400   FORMAT (///21X,3A8,60X,A8,1X,G12.3,//20H  FUNCTION=CONSTANT=,G12
     1.4///)
410   FORMAT (///21X,3A8,60X,A8,1X,G12.3,//40H  BLANK PAGE FOR CONTOR.
     1   XB,XT,YL,YU=,4G12.3///)
420   FORMAT (1H ,7X,4H**  ,3A8,7X,60A1,10X,A8,1X,F7.3//14X,8HMAXIMUM=
     1,1P,G12.3,3X,8HMINIMUM=,G12.3,3X,13HCONTOUR ZERO=,G12.3,3X,18HCONT
     2OUR INCREMENT=,G12.3,3X,A8//1X,0P,11(F8.1,4X))
430   FORMAT (8X,121A1)
440   FORMAT (1H+,7X,121A1)
450   FORMAT (1X,F7.1,121A1)
460   FORMAT (1X,11(F8.1,4X)////)
      END


      SUBROUTINE COMBIN (NLINE,IDEST,ICH,NFROM,IFROM,ICHAR)
      DIMENSION IFROM(NFROM), ICHAR(NFROM)
      INTEGER PLUS,BLANK,ONE
      DATA PLUS/1H+/,BLANK/1H /
      DIMENSION LG(140), LB(140), IBEG(10), IEND(10), IFILE(10)
      DATA LB/1H+,139*1H /
C
C COMBINES NLINE PRINT LINES FROM THE BEGINNING OF STREAMS IFROM(NFROM),
C    ICHAR(NFROM) CHARACTERS FROM EACH. REWINDS STREAMS AT THE END.
C DEALS CORRECTLY WITH OVERPRINTING + AT LINE BEGINNING, OTHER BEGINNING
C    CHARACTERS ARE TREATED AS BLANKS EXCEPT ON IFROM(1).
C OUTPUTS ICH CHARACTERS PER LINE, TO STREAM IDEST.
C NFROM AND ICHAR REDUCED IF NECESSARY TO GIVE LESS THAN ICH CHARACTERS.
C IBM STREAMS //GO.FTXXF001 DD UNIT=SYSDA,SPACE=(TRK,(1,5)),
C               //  DCB=(RECFM=FBA,BLKSIZE=1330,LRECL=133)
C
C
C INITIALISE.
      N=0
      DO 30 J=1,NFROM
      IBEG(J)=N+1
      N=N+ICHAR(J)
      IF (N.LT.ICH) GO TO 10
      NFROM=J
      N=ICH
      ICHAR(J)=ICH-IBEG(J)+1
10    IEND(J)=N
      IFILE(J)=1
      K=IFROM(J)
      IB=IBEG(J)
      IE=IEND(J)
      REWIND K
C
      READ (K,20) LM,(LG(I),I=IB,IE)
      IF (J.EQ.1) LL=LM
C
20    FORMAT (140A1)
30    CONTINUE
      IC=N
C
C MAIN LOOP
      DO 90 ILINE=1,NLINE
      WRITE (IDEST,20) LL,(LG(I),I=1,IC)
      LL=BLANK
      DO 80 J=1,NFROM
      IF (IFILE(J).EQ.0) GO TO 80
      IE=IEND(J)
      IB=IBEG(J)
      K=IFROM(J)
```

```
40      READ (K,20,END=60) LM,(LG(I),I=IB,IE)
        IF (LM.NE.PLUS) GO TO 50
        WRITE (IDEST,20) (LB(I),I=1,IB),(LG(I),I=IB,IE)
        GO TO 40
50      IF (J.EQ.1) LL=LM
        GO TO 30
60      DO 70 I=IB,IE
70      LG(I)=BLANK
        IFILE(J)=0
80      CONTINUE
90      CONTINUE
C END MAIN LOOP.
C
        DO 100 J=1,NFROM
        K=IFROM(J)
        REWIND K
100     CONTINUE
        RETURN
        END


        SUBROUTINE BD (A,II,JJ)
C  ***  APPLY SYMMETRY BOUNDARY CONDITIONS FOR ARRAY PLOTTING  ***
        DIMENSION A(II,JJ)
        II1=II-1
        JJ1=JJ-1
        DO 10 J=2,JJ1
        A(1,J)=A(2,J)
10      A(II,J)=A(II1,J)
        DO 20 I=1,II
        A(I,1)=A(I,2)
20      A(I,JJ)=A(I,JJ1)
        RETURN
        END


        SUBROUTINE VPLOT (A,N,M,XV,XB,XT,CH,AL,AR,NVTK,NVCPT,NHTK,NHCPT,
     1IS)
C
C ***** GLYN'S VPLOT PLOTTER ROUTINE ******
C PLOTS M FUNCTIONS OF X, DOWN THE PRINTER PAGE.
C X INCREASES UP THE PAGE FROM XB TO XT.
C A(J,K) IS THE VALUE OF THE K'TH FUNCTION AT XV(J), J=1 TO N.
C XV(J) IS MONOTONIC INCREASING.
C THE  LEFT MARGIN REPRESENTS THE VALUE AL(K).
C THE RIGHT MARGIN REPRESENTS THE VALUE AR(K).
C THE K'TH FUNCTION IS PLOTTED USING THE CHARACTER CH(K).
C THE PLOTTING AREA HAS NVTK TICK MARKS VERTICALLY, AND
C    NVCPT CHARACTERS PER TICK.
C THE PLOTTING AREA HAS NHTK TICK MARKS HORIZONTALLY, AND
C    NHCPT CHARACTERS PER TICK.
C THE TOTAL CHARACTERS HORIZONTALLY IS NHTK*NHCPT, AND IS LESS THAN 132.
C
C BLANK COMMON IS USED FOR WORKING SPACE ( 266 WORDS) AND SO PASSED
C  ARRAYS SHOULD NOT BE IN BLANK COMMON.
C
        IMPLICIT REAL*8(A-H,O-Z)
        COMMON LG(133,2)
        REAL A
        DIMENSION A(N,M), XV(N), CH(M), AL(M), AR(M)
        DATA BLANK/1H /,MINUS/1H-/,PLUS/1H+/,BAR/1H|/
        INTEGER CH,BLANK,PLUS,BAR
        IH=NHTK*NHCPT
        IV=NVTK*NVCPT
        IH2=IH+2
        IV1=IV+1
        NM=N-1
C       WRITE (IS,10)
        ILP=2
        LG(1,1)=BLANK
        LG(1,2)=PLUS
        DO 120 I=1,IV1
        IF (I.NE.1.AND.I.NE.IV1) GO TO 30
        DO 10 J=2,IH2
        LG(J,1)=MINUS
10      LG(J,2)=BLANK
        DO 20 J=2,IH2,NHCPT
20      LG(J,1)=PLUS
        GO TO 30
30      DO 40 K=1,ILP
```

```
      DO 40 J=2,IH2
40    LG(J,K)=BLANK
      LG(IH2,1)=BAR
      LG(2,1)=BAR
      IHH=(2+IH2)/2
      IF (I/2*2.EQ.I.AND.NHTK/2*2.EQ.NHTK) LG(IHH,1)=BAR
50    CONTINUE
      ILP=1
      X=XT-(I-1)*(XT-XB)/IV
      DO 60 J=1,NM
      IF ((X-XV(J))*(X-XV(J+1)).LE.0) GO TO 70
60    CONTINUE
      IF ((X-XV(2))/(X-XV(1)).GT.1) J=2
70    J=MAX0(J,2)
      J=MIN0(J,N-2)
      IX=J-2
      DO 100 J=1,M
C
      E=0
      DO 90 K=1,4
      D=A(IX+K,J)
      DO 80 L=1,4
      IF (L.EQ.K) GO TO 80
      D=D*(X-XV(IX+L))/(XV(IX+K)-XV(IX+L))
80    CONTINUE
90    E=E+D
      K=2.5+(E-AL(J))/(AR(J)-AL(J))*IH
      IF (K.GT.IH2.OR.K.LT.2) GO TO 100
      IL=1
      IF (LG(K,1).NE.BLANK) IL=2
      LG(K,IL)=CH(J)
      ILP=MAX0(ILP,IL)
100   CONTINUE
      DO 120 L=1,ILP
      IF ((I-1)/NVCPT*NVCPT.EQ.I-1.AND.L.EQ.1) GO TO 110
      WRITE (IS,130) (LG(K,L),K=1,IH2)
      GO TO 120
110   LG(IH2,1)=PLUS
      LG(2,1)=PLUS
      WRITE (IS,140) X,(LG(K,L),K=2,IH2)
120   CONTINUE
130   FORMAT (A1,7X,133A1)
140   FORMAT (1X,F7.1,133A1)
      RETURN
      END


      SUBROUTINE XPLOT (F,XV,NX,ISEP,NLIN,XL,XU,TITLE,SUBTIT,VARI)
C
C     ***  GLYN'S XPLOT PLOTTER ROUTINE   *****
C F(I) IS VALUE AT XV(I), WITH XV(I) MONOTONIC INCREASING.
C THE DIMENSIONS ARE NX.
C DOMAIN XL TO XU SPLIT INTO ISEP PAGES HORIZONTALLY FOR HIGH RESOLUTION
C APPROXIMATELY NLIN LINES ARE DRAWN VERTICALLY ON THE PRINTER PAGE.
C TITLE IS 24 CHARACTERS AT 8 PER WORD.
C SUBTIT IS 9 CHARACTERS, E.G. *TIME =  * , WITH VARI THE TIME VALUE.
C MAIN PROGRAM SHOULD CONTAIN --
C     COMMON /TXTC/ITXT(60),DAY
C     READ(5,1)ITXT
C1    FORMAT(60A1)
C     CALL IDAY(DAY)
C -- TO READ FURTHER TEXT OFF A CARD AND SET DATE IN DAY.
C BLANK COMMON IS USED FOR WORKING SPACE (1568 WORDS) AND SO PASSED
C  ARRAYS SHOULD NOT BE IN BLANK COMMON.
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER JB(3,121),JT(3,121)
      REAL G,B
      DIMENSION F(NX), G(721), IH(121), XI(11), XV(NX)
      INTEGER KK(3)
      EQUIVALENCE (KK(1),KK1), (KK(2),KK2), (KK(3),KK3)
      INTEGER Q(2,2,2),EQU,APOS,UND,BLANK,MINUS,PLUS,BAR,DOT,LG(121)
      DIMENSION TITLE(3)
      DIMENSION ISTRM(10)
      COMMON /TXTC/ ITXT(60),DAY
      COMMON G,JB,JT,IH
      DATA Q/1HM,1H9,1HH,1HJ,1HP,1HI,1HL,1H /,EQU/1H=/
      DATA APOS/1H'/,UND/1H_/,DOT/1H./,BAR/1H|/
      DATA BLANK/1H /,MINUS/1H-/,PLUS/1H+/
      DATA ISTRM/5,21,31,4,11,41,42,43,22,23/,NSTRM/10/
```

```
      NST=1
      IF (NLIN.EQ.-1) NST=NSTRM
      IF (NLIN.EQ.-1) NLIN=15
      DO 330 NS=1,NST
      IS=ISTRM(NS)
      XG=(XU-XL)/ISEP
      DO 330 ISP=1,ISEP
      XB=XL+XG*ISP-XG
      DO 10 I=1,121
10    LG(I)=MINUS
      DO 20 I=1,121,12
20    LG(I)=PLUS
      DO 30 I=1,11
30    XI(I)=XB+XG*(I-1)/10
C
      NM=NX-1
C     FIND MAXIMUM, MINIMUM, AND CONTOUR ORIGIN AND INCREMENT
      CALL MAXMIN (F,XV,NX,FMX,FMN,FAB,XMX,XMN,XAB)
      IF (1.E4*(FMX-FMN).GT.DABS(FMX)+DABS(FMN)) GO TO 40
      WRITE (IS,340) TITLE,SUBTIT,VARI,FMX
      RETURN
40    CONTINUE
      A=DLOG10((FMX-FMN)/NLIN)+.1505
      I=A
      IF (A.LT.0) I=I-1
      B=I
      C=A-I
      FINC=10.**B
      IF (C.LT..1505) GO TO 50
      FINC=10.**B*1.5
      IF (C.LT..3010) GO TO 50
      FINC=10.**B*2
      IF (C.LT..4516) GO TO 50
      FINC=10.**B*3
      IF (C.LT..6505) GO TO 50
      FINC=10.**B*5
      IF (C.LT..8490) GO TO 50
      FINC=10.**B*7.5
50    CONTINUE
      FINC=FINC*4
      I=FMN/FINC-.09375+1000
      I=I-1000
      FOR=I*FINC
      N=(FMX-FOR)/FINC+1.15625
      N=N*4-1
      FINC=FINC/4
      FTOP=FOR+FINC*(N+1)
      CALL SHORT7 (FTOP,FR,IX)
      WRITE (IS,350) TITLE,ITXT,SUBTIT,VARI,FMX,FMN,FINC,DAY,XI,FR,IX,
     1LG
C
C     SET UP G
      IQ=121
60    CONTINUE
      DO 110 I=1,IQ
      X=XB+(I-1)*XG/(IQ-1)
      DO 70 J=1,NM
      IF ((X-XV(J))*(X-XV(J+1)).LE.0) GO TO 80
70    CONTINUE
      IF ((X-XV(2))/(X-XV(1)).GT.1) J=2
80    J=MAX0(J,2)
      J=MIN0(J,NX-2)
      IX=J-2
C
      E=0
      DO 100 K=1,4
      D=F(IX+K)
      DO 90 L=1,4
      IF (L.EQ.K) GO TO 90
      D=D*(X-XV(IX+L))/(XV(IX+K)-XV(IX+L))
90    CONTINUE
100   E=E+D
110   G(I)=E
      IF (IQ.NE.721) GO TO 130
      DO 120 K=1,3
      DO 120 J=2,120
      I=6*J+2*K-9
      B=G(I)
      E=G(I+1)-B
      D=B-G(I-1)
```

```
      IF (E*D.LT.0) B=B-(E+D)**2/8/(E-D)
      JB(K,J)=N+2-(AMAX1(G(I+1),G(I-1),B)-FOR)/FINC
120   JT(K,J)=N+1-(AMIN1(G(I+1),G(I-1),B)-FOR)/FINC
      GO TO 160
130   CONTINUE
      DO 140 I=1,121
140   IH(I)=4*N-4*(G(I)-FOR)/FINC+4.5
      JQ=0
      DO 150 I=2,121
150   JQ=MAX0(JQ,IH(I)-IH(I-1),IH(I-1)-IH(I))
      IQ=721
      IF (JQ.GT.12) GO TO 60
160   CONTINUE
C     MAIN PROGRAM
      DO 300 J=1,N
      DO 170 I=1,121
170   LG(I)=BLANK
      DO 180 I=1,121,12
180   LG(I)=BAR
      IF (J/4*4.NE.J) GO TO 210
      DO 190 I=1,121
190   LG(I)=MINUS
      DO 200 I=1,121,12
200   LG(I)=PLUS
210   CONTINUE
      Y=FTOP-J*FINC
      K=0
      DO 220 I=1,121
      P3=.3
      A=DABS(IH(I)-4*J-P3)
      IF (.2.LT.A.AND.A.LT.1.8) K=K+1
      IF (IH(I).EQ.4*J+2) LG(I)=UND
      IF (IH(I).EQ.4*J+1) LG(I)=DOT
      IF (IH(I).EQ.4*J-1) LG(I)=APOS
      IF (IH(I).EQ.4*J) LG(I)=MINUS
220   IF (IH(I).EQ.4*J.AND.J/4*4.EQ.J) LG(I)=EQJ
      IF (JQ.LE.12) GO TO 260
      DO 250 I=2,120
      DO 230 K=1,3
      KK(K)=2
230   IF (J.GE.JB(K,I).AND.J.LE.JT(K,I)) KK(K)=1
      IF (KK(1)+KK(2)+KK(3).EQ.6) GO TO 250
      P3=.3
      IF (DABS(IH(I)-4*J-P3).GT.1.8) GO TO 240
      IF ((MAX0(JT(1,I),JT(2,I),JT(3,I)).LE.J.OR.MIN0(JB(1,I),JB(2,I),
     1JB(3,I)).GE.J)) GO TO 250
240   CONTINUE
      LG(I)=D(KK1,KK2,KK3)
250   CONTINUE
260   CONTINUE
C     PRINT LINE, POSSIBLY WITH Y VALUE
      IF (J/4*4.NE.J) GO TO 290
      CALL SHORT7 (Y,FR,IX)
      WRITE (IS,360) FR,IX,LG
      IF (K.LT.3.AND.JQ.LE.12) GO TO 300
      DO 270 I=1,121
270   LG(I)=MINUS
      DO 280 I=1,121,12
280   LG(I)=PLUS
      WRITE (IS,370) LG
      GO TO 300
290   CONTINUE
      WRITE (IS,380) LG
300   CONTINUE
C
C
C     BOTTOM UNDERLINE
      DO 310 I=1,121
310   LG(I)=MINUS
      DO 320 I=1,121,12
320   LG(I)=PLUS
      CALL SHORT7 (FOR,FR,IX)
      WRITE (IS,390) FR,IX,LG,XI
330   CONTINUE
      RETURN
C
C
340   FORMAT (1H-,20X,3A8,60X,A8,1X,G12.3,/25H0XPLOT FUNCTION=CONSTANT
     1=,G12.3/)
350   FORMAT (1H ,7X,4H**  ,3A8,7X,60A1,10X,A8,1X,F7.3//14X,8HMAXIMUM=
```

```
     1,1P,G12.3,4X,8HMINIMUM=,G12.3,7X,19HVERTICAL INCREMENT=,G12.3,23X,
     2A9//1X,0P,11(F8.1,4X)/1X,F5.2,I2,121A1)
 360  FORMAT (1X,F5.2,I2,121A1)
 370  FORMAT (1H+,7X,121A1)
 380  FORMAT (8X,121A1)
 390  FORMAT (1X,F5.2,I2,121A1/1X,11(F8.2,4X)/////)
      END

      SUBROUTINE MAXMIN (F,X,N,FMAX,FMIN,FABS,XMAX,XMIN,XABS)
C *** MAXIMUM OF FUNCTION, USING SECOND ORDER INTERPOLATION ***
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION F(N), X(N)
      NM=N-1
      XMAX=X(1)
      XMIN=X(1)
      FMAX=F(1)
      FMIN=F(1)
      IF (F(N).GT.FMAX) XMAX=X(N)
      IF (F(N).GT.FMAX) FMAX=F(N)
      IF (F(N).LT.FMIN) XMIN=X(N)
      IF (F(N).LT.FMIN) FMIN=F(N)
      DO 10 I=2,NM
      D=F(I)-F(I-1)
      E=F(I+1)-F(I)
      IF (D.EQ.E) GO TO 10
      IF (D*E.GT.0.AND.I.NE.2.AND.I.NE.NM) GO TO 10
      IF (D*E*10/3.GT.D*D+E*E) GO TO 10
      G=F(I)-(E+D)**2/(E-D)/8
      Y=X(I)-(E+D)/4/(E-D)*(X(I+1)-X(I-1))
      IF (G.LT.FMIN) XMIN=Y
      IF (G.LT.FMIN) FMIN=G
      IF (G.GT.FMAX) XMAX=Y
      IF (G.GT.FMAX) FMAX=G
 10   CONTINUE
      XABS=XMAX
      FABS=DABS(FMAX)
      IF (FABS+FMIN.LT.0) XABS=XMIN
      IF (FABS+FMIN.LT.0) FABS=-FMIN
      RETURN
      END

      SUBROUTINE SHORT7 (F,FR,IX)
      IMPLICIT REAL*8(A-G)
      B=F
      A=1
      IF (F.GT.1.D-10) GO TO 10
      A=-1
      F=-F
      IF (F.GT.1.D-10) GO TO 10
      IX=0
      FR=0
      F=B
      RETURN
 10   I=1000+DLOG10(F)+1.E-3
      IX=I-1000
      FR=A*F/10.D0**IX
      F=B
      RETURN
      END
```

REFERENCES

1. Roberts, Glyn O., Piacsek, S. A., and Toomre, Juri, "Two-Dimensional Numerical Model of the Near-Field Flow for an Ocean Thermal Power Plant, Part I. The Theoretical Approach and a Laboratory Simulation," NRL-GFD/OTEC 5-76, Science Applications, Inc. and Geophysical Simulation Section, Code 7750, Naval Research Laboratory.

2. Roberts, Glyn O., Piacsek, S. A., and Toomre, Juri, "Two-Dimensional Numerical Model of the Near-Field Flow for an Ocean Thermal Power Plant, Part II. Simulation of the Lockheed Baseline Design," NRL-GFD/OTEC 6-76, Science Applications, Inc. and Geophysical Simulation Section, Code 7750, Naval Research Laboratory.

TWO-DIMENSIONAL NUMERICAL MODEL
OF THE NEAR-FIELD FLOW FOR AN
OCEAN THERMAL POWER PLANT.

PART III. MANUAL FOR THE
COMPUTER CODE NRFLO2.

SAI-76-626-WA

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