PB83-116103

Verification and Transfer of Thermal Pollution Model. Volume IV User's Manual for Three-Dimensional Rigid-Lid Model

Miami Univ.
Coral Gables, FL

Prepared for

National Aeronautics and Space Administration
Cocoa Beach, FL

May 82

U.S. Department of Commerce
National Technical Information Service
NTIS

PB83-116103

EPA-600/7-82-037d

May 1982

VERIFICATION AND TRANSFER OF THERMAL POLLUTION MODEL

VOLUME IV: USER'S MANUAL FOR THREE-DIMENSIONAL RIGID-LID MODEL

By

Samuel S. Lee, Subrata Sengupta,
Emmanuel V. Nwadike and Sumon K. Sinha
Department of Mechanical Engineering
University of Miami
Coral Gables, Florida 33124

NASA Contract No. NAS 10-9410

NASA Project Manager: Roy A. Bland

National Aeronautics and Space Administration
Kennedy Space Center
Kennedy Space Center, Florida 32899

EPA Interagency Agreement No. 78-DX-0166

EPA Project Officer: Theodore G. Brna

Industrial Environmental Research Laboratory
Office of Environmental Engineering and Technology
Research Triangle Park, North Carolina 27711

Prepared for:

U. S. Environmental Protection Agency
Office of Research and Development
Washington, D. C. 20460

ATTENTION

AS NOTED IN THE NTIS ANNOUNCEMENT, PORTIONS OF THIS REPORT ARE NOT LEGIBLE. HOWEVER, IT IS THE BEST REPRODUCTION AVAILABLE FROM THE COPY SENT TO NTIS.

TECHNICAL REPORT DATA
(Please read Instructions on the reverse before completing)

1. REPORT NO. EPA-600/7-82-037d	2. ORD Report	PB83-116103
4. TITLE AND SUBTITLE Verification and Transfer of Thermal Pollution Model; Volume IV. User's Manual for Three-dimensional Rigid-lid Model		5. REPORT DATE May 1982
		6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S) S.S.Lee, S.Sengupta, E.V.Nwadike, and S.K.Sinha		8. PERFORMING ORGANIZATION REPORT NO.
9. PERFORMING ORGANIZATION NAME AND ADDRESS The University of Miami Department of Mechanical Engineering P.O. Box 248294 Coral Gables, Florida 33124		10. PROGRAM ELEMENT NO.
		11. CONTRACT/GRANT NO. EPA IAG-78-DX-0166*
12. SPONSORING AGENCY NAME AND ADDRESS EPA, Office of Research and Development Industrial Environmental Research Laboratory Research Triangle Park, NC 27711		13. TYPE OF REPORT AND PERIOD COVERED Final; 3/78-9/80
		14. SPONSORING AGENCY CODE EPA/600/13

15. SUPPLEMENTARY NOTES IERL-RTP project officer is Theodore G.Brna, Mail Drop 61, 919/541-2683. (*) IAG with NASA, Kennedy Space Center, FL 32899, subcontracted to U. of Miami under NASA Contract NAS 10-9410.

16. ABSTRACT The six-volume report: describes the theory of a three-dimensional (3-D) mathematical thermal discharge model and a related one-dimensional (1-D) model, includes model verification at two sites, and provides a separate user's manual for each model. The 3-D model has two forms: free surface and rigid lid. The former, verified at Anclote Anchorage (FL), allows a free air/water interface and is suited for significant surface wave heights compared to mean water depth; e.g., estuaries and coastal regions. The latter, verified at Lake Keowee (SC), is suited for small surface wave heights compared to depth (e.g., natural or man-made inland lakes) because surface elevation has been removed as a parameter. These models allow computation of time-dependent velocity and temperature fields for given initial conditions and time-varying boundary conditions. The free-surface model also provides surface height variations with time. The 1-D model is considerably more economical to run but does not provide the detailed prediction of thermal plume behavior of the 3-D models. The 1-D model assumes horizontal homogeneity, but includes area-change and several surface-mechanism effects.

17. KEY WORDS AND DOCUMENT ANALYSIS

a. DESCRIPTORS	b. IDENTIFIERS/OPEN ENDED TERMS	c. COSATI Field/Group
Pollution	Pollution Control	13B
Thermal Diffusivity	Stationary Sources	20M
Mathematical Models		12A
Estuaries		08H,08J
Lakes		
Plumes		21B

18. DISTRIBUTION STATEMENT Release to Public	19. SECURITY CLASS (This Report) Unclassified	21. NO. OF PAGES 154
	20. SECURITY CLASS (This page) Unclassified	22. PRICE

EPA Form 2220-1 (9-73)

PREFACE

The three-dimensional rigid-lid model is intended to be used for hydrothermal predictions of closed basins subjected to a heated discharge together with various other inflows and outflows. This volume has been written in order to assist any prospective user in applying the model to specific sites. Derivation of the governing equations and various other details have been omitted. The programs are fairly general and only one subroutine and a data file has to be rewritten for specific cases.

This work was sponsored by the National Aeronautics and Space Administration (NASA-KSC) and the Environmental Protection Agency (EPA-RTP).

NOTICE

ABSTRACT

The three-dimensional rigid-lid model was developed by the thermal pollution group at the University of Miami and verified for accuracy at various sites. The model results have been found to be fairly accurate in all the verification runs. The model is intended to be used as a predictive tool in future sites and this manual has been written to enable any user to be able to apply it without difficulty.

CONTENTS

FIGURES

TABLES

SYMBOLS

A_H	Horizontal kinematic eddy viscosity
A_V	Vertical kinematic eddy viscosity
A_{ref}	Reference kinematic eddy viscosity
A_V^*	A_V/A_{ref}
B_H	Horizontal eddy thermal diffusivity
B_V	Vertical eddy thermal diffusivity
B_{ref}	Reference eddy thermal diffusivity
B_V^*	B_V/B_{ref}
C_p	Specific heat at constant pressure
Eu	Euler number
f	Coriolis parameter
g	Acceleration due to gravity
h	Depth relative to the mean water level
H	Reference depth
I	Grid index in x-direction or α-direction
J	Grid index in y-direction or β-direction
K	Grid index in z-direction or γ-direction
K_S	Surface heat transfer coefficient
L	Horizontal length scale
P	Pressure
P_S	Surface pressure
Pr	Turbulent Prandtl number, A_{ref}/B_{ref}
Pe	Péclet number
Q	Heat sources or sinks
Re	Reynolds number (turbulent)
Ri	Richardson number
T	Temperature
T_{ref}	Reference temperature
T_e	Equilibrium temperature
T_s	Surface temperature
t	Time
t_{ref}	Reference time
u	Velocity in x-direction
v	Velocity in y-direction
w	Velocity in z-direction
x	Horizontal coordinate
y	Horizontal coordinate
z	Vertical coordinate

Greek Letters

α	Horizontal coordinate in stretched system, = x
β	Horizontal coordinate in stretched system, = y
γ	Vertical coordinate in stretched system
σ	Constant in vertical diffusivity equation, or vertical coordinate in stretched system, = Z/H
Ω	Transformed vertical velocity
ρ	Density
τ_{xz}	Surface shear stress in x-direction
τ_{yz}	Surface shear stress in y-direction

$\overline{(\ )}$	Dimensional quantity
$(\tilde{\ })$	Dimensional mean quantity
$(\acute{\ })$	Dimensional quantity
$(\ )$	Dimensional quantity
$(\ )_{ref}$	Reference quantity

ACKNOWLEDGMENTS

This work was supported by a contract from the National Aeronautics and Space Administration (NASA-KSC) and the Environmental Protection Agency (EPA-RTP).

The authors express their sincere gratitude for the technical and managerial support of Mr. Roy A. Bland, the NASA-KSC project manager of this contract, and the NASA-KSC remote sensing group. Special thanks are also due to Dr. Theodore G. Brna, The EPA-RTP project manager, for his guidance and support of the experiments, and to Mr. S. B. Hager, Chief Engineer, Civil-Environmental Division, and Mr. William J. McCabe, Assistant Design Engineer, both from the Duke Power Company, Charlotte, North Carolina, and their data collection group for data acquisition. The support of Mr. Charles H. Kaplan of EPA was extremely helpful in the planning and reviewing of this project.

SECTION 1

INTRODUCTION

The need for mathematical modeling in predicting and monitoring thermal pollution was discussed in previous reports by Veziroglu et al. (1973, 1974). Predictive studies of ecosystems can only be made by mathematical models. A prior knowledge of the effects of disturbances is essential for environmental impact studies. Thus, the mathematical model is a crucial tool in decisions involving power plant siting, land development, etc.

The University of Miami team undertook development of a methodology using remote sensing and numerical modeling to study thermal pollution. The use of remotely-sensed data in modeling has been discussed by Sengupta et al. (1974). The remote sensing effort has been discussed in detail in previous publications. This volume has been written so as to enable a user to apply the mathematical model to new sites for predictive purposes.

The hydrodynamics and thermodynamics of an ecosystem are controlled by geometry, meteorological conditions and physical characteristics of the water such as density, salinity and turbidity. In this model the effects of salinity and turbidity have been neglected. Hence, the governing equations are composed of the three-dimensional Navier-Stokes equations and the energy equation. Various assumptions can be made for different situations leading to simplification or elimination of equations. The main simplifying assumption in this case is the rigid-lid assumption. This means that surface height fluctuations are not simulated by this model, and this is a reasonable assumption for most applications (e.g., Lakes).

The rigid-lid model has the following capabilities:

1. It predicts the wind-driven circulation.

2. It predicts the circulation caused by inflows and outflows to the domain.

3. It predicts the thermal effects in the domain.

4. It combines the aforementioned processes.

The calibration procedure consists of comparing ground-truth corrected airborne radiometer data with surface isotherms predicted by the model.

SECTION 2

RECOMMENDATIONS

Various numerical models have been developed to study the effects of heated discharge and meteorological conditions on bodies of water. Most of these models are one or two dimensional. These models have a high computational speed but only give horizontally or vertically averaged values of temperatures.

Three-dimensional models, however, have a much finer resolution but they consume larger computer time. The three-dimensional rigid-lid model can be used to obtain detailed temperature and velocity distributions in a domain where surface gravity waves are small compared to the depth of the domain. This model, as compared to free-surface models, runs faster since surface gravity waves are eliminated by the rigid-lid assumption.

A proper method of using this model would be to run a one-dimensional model initially to obtain a rough picture of the temperatures and then using this model to obtain a better resolution, the 1-D results being used as ambient conditions.

The following improvements have been suggested for the model.

1. Since all natural flows are turbulent, proper turbulent closures are needed to make the model meaningful. At present, the simplest possible closures, namely constant eddy viscosities and eddy diffusivities, have been used. However, better results may be obtained by using a higher order closure.

2. At present, the model uses uniform horizontal grids and stretched vertical grids. Nonuniform horizontal grids could be introduced for better resolution near the boundaries.

3. The program has been written to be run as a batch-job on the computer. It could be made interactive so as to enable the user to run it on a terminal. However, this would require some modifications in order to reduce the storage space.

SECTION 3

PROGRAM DESCRIPTION AND FLOW CHART

DESCRIPTION OF PROGRAM ALGORITHM

The governing equations for a body of water which are derived from the basic laws of conservation of mass, momentum and energy are shown in Table 1. These equations incorporate a vertically-stretched coordinate system so as to make the model general enough to handle any kind of bottom topography. The problem is set up as an initial value problem. The initial values of the water velocities and temperatures are specified and the model is run so as to give the values of the above quantities in subsequent time periods using an explicit scheme. The sequence of the calculations are as follows:

1. The initial values of the velocities and temperatures are read into the program, the region of interest within the basin being classified into interior, corner or boundary points. (Subroutines used are READ 3K, INITIA, INITIT, HEIGHT.)

2. The data, which includes the boundary conditions such as the various meteorological parameters like surface wind speed, air temperature, humidity and solar radiation are read into the program using subroutine READ2.

3. Depending on the site chosen, the various discharges (volume flow rate, velocities and temperatures) in and out of the basin are read into the model. These are incorporated in the subroutine INLET1.

4. The momentum, continuity and energy equations are now solved to determine the velocities and temperatures in the subsequent time steps. The predictive equation for pressure (viz., the Poisson equation) is solved iteratively to determine the pressures at various points of the domain. (Note: Because of the rigid-lid assumption, the surface or lid pressure is no longer atmospheric.)

THE PROGRAM FLOW CHART IS SHOWN IN FIGURE 1

The various subroutines used are as well as a brief description of their functions are shown in Tables 2 and 3.

<u>Symbols Used in Governing Equations</u>

(Quantities with bar are dimensional)

$\tilde{\rho}$ = density

$\tilde{T}$ = temperature

$\tilde{\omega} = \gamma(u\frac{\partial \tilde{h}}{\partial \dot{x}} + v\frac{\partial \tilde{h}}{\partial \dot{x}}) + \tilde{h}\tilde{\Omega}$

$\tilde{\Omega} = \frac{\partial \gamma}{\partial \tilde{t}}$

$\gamma = \tilde{Z}/\tilde{h}(n)y$

$\beta = \tilde{y}/L$

$\alpha = \tilde{x}/L$

$u = \tilde{u}/U_{ref}$

$v = \tilde{v}/U_{ref}$

$w = \tilde{w}/U_{ref}$

$t = \tilde{t}/t_{ref}$

$\varepsilon = H/L$

$P = \tilde{P}/P_{ref}U_{ref}^2$

$T = \dfrac{\tilde{T}-T_{ref}}{T_{ref}}$

$\rho = \dfrac{\tilde{\rho}-\rho_{ref}}{\rho_{ref}}$

$A_H^* = A_H/A_{ref}$ nondimensional horizontal eddy viscosity

$A_v^* = A_v/A_{ref}$ nondimensional vertical eddy viscosity

$B_H^* = B_H/B_{ref}$ nondimensional horizontal eddy viscosity

$B_v^* = B_v/B_{ref}$ nondimensional vertical eddy viscosity

$R_e = (U_{ref}L)/A_{ref}$, $R_B = U_{ref}/fL$, $P_r = A_{ref}/B_{ref}$

$P_e = R_e, P_r$, $E_u = gH/U_{ref}^2$

Table 1. Governing Equations

Continuity Equation:

$$\frac{\partial(hu)}{\partial\alpha} + \frac{\partial(hv)}{\partial\beta} + h\frac{\partial\Omega}{\partial\gamma} = 0$$

Momentum Equation:

$$\frac{\partial(hu)}{\partial t} + \frac{\partial(huu)}{\partial\alpha} + \frac{\partial(huv)}{\partial\beta} + h\frac{\partial(\Omega u)}{\partial\gamma} - \frac{h}{R_B}v$$

$$= -h\frac{\partial P_s}{\partial\alpha} - hB_x + \frac{1}{R_e}\frac{\partial}{\partial\alpha}(h\frac{\partial}{\partial\alpha}) + \frac{1}{R_e}\frac{\partial}{\partial\beta}(h\frac{\partial v}{\partial\beta})$$

$$+ \frac{1}{E^2R_e}\frac{1}{h}\frac{\partial}{\partial\gamma}(A_v^*\frac{\partial u}{\partial\gamma})$$

and

$$\frac{\partial(hv)}{\partial t} + \frac{\partial(huv)}{\partial\alpha} + \frac{\partial(hvv)}{\partial\beta} + h\frac{\partial(\partial v)}{\partial\gamma} + \frac{h}{R_B}u$$

$$= -h\frac{\partial P_s}{\partial\beta} - hB_y + \frac{1}{R_e}\frac{\partial}{\partial\alpha}(h\frac{\partial v}{\partial\alpha}) + \frac{1}{R_e}\frac{\partial}{\partial\beta}(h\frac{\partial v}{\partial\beta})$$

$$+ \frac{1}{E^2R_e}\frac{1}{h}\frac{\partial}{\partial\gamma}(A_v^*\frac{\partial v}{\partial\gamma})$$

Hydrostatic Equation:

$$\frac{\partial P}{\partial\gamma} = E_u(1+\rho)h$$

Energy Equation:

$$\frac{\partial(hT)}{\partial t} + \frac{\partial(huT)}{\partial\alpha} + \frac{\partial(hvT)}{\partial\beta} + h\frac{\partial(\Omega T)}{\partial\gamma}$$

$$= \frac{1}{P_e}\frac{\partial}{\partial\alpha}(h\frac{\partial T}{\partial\alpha}) + \frac{1}{P_e}\frac{\partial}{\partial\beta}(h\frac{\partial T}{\partial\beta}) + \frac{1}{P_eE^2}\frac{1}{h}\frac{\partial}{\partial\gamma}(B_v^*\frac{\partial T}{\partial\gamma})$$

SECTION 4

LIST OF PROGRAM SYMBOLS USED IN MAIN PROGRAM

DESCRIPTION OF MAIN VARIABLES

A. A - constant in equation of state, $\rho = A + BT + CT^2$

AREF - reference eddy viscosity

AA - value of 'V' at plume inlet

ABR - 1/Rossby number

AH - 1/Reynolds number

AI - coefficient in front of pressure term

AKT - $(K_s)(H_{ref})/(B_z)$

AP - coefficient in front of pressure term

ARBP - arbitrary pressure

AV - $\frac{1}{\varepsilon^2 R_E}$ where $\varepsilon = \frac{H}{L}$

A3 - normalized vertical eddy coefficient of viscosity

ANGLE - wind direction angle

B. B - constant in equation of state, $\rho = A + BT + CT^2$

BB - value of 'V' at plume inlet (at I=10)

BZ - $\rho C_p B_v$

BV - normalized vertical eddy diffusivity, normalized with respect to reference eddy diffusivity

C. C - constant in equation of state, $\rho = A + BT + CT^2$

CC - value of γ (constant)

CW - temperature gradient at vertical boundaries

CB - temperature gradient at the bottom

D. D - U at previous time step

D1TZ - $\frac{\partial T}{\partial Z}$

DPX - $\frac{\partial P}{\partial x}$

DPY - $\frac{\partial P}{\partial y}$

DPSX - $\frac{\partial P_s}{\partial x}$

DPSY - $\frac{\partial P_s}{\partial y}$

DT - time increment

DX - increment in x-direction

DY - increment in y-direction

DZ - increment in Z-direction

D1HUX - $\frac{\partial (hu)}{\partial x}$

D1HVY - $\frac{\partial (hv)}{\partial y}$

D1HUUX - $\frac{\partial (huu)}{\partial x}$

D1HUVY - $\frac{\partial (huv)}{\partial y}$

D1HVVY - $\frac{\partial (hvv)}{\partial y}$

D1UY - $\frac{\partial u}{\partial y}$

D1VX - $\frac{\partial v}{\partial x}$

D2UX - $\frac{\partial^2 u}{\partial x^2}$

D2VX - $\frac{\partial^2 v}{\partial x^2}$

D1VWX - $\frac{\partial (vw)}{\partial Z}$

D1UZ - $\frac{\partial u}{\partial Z}$

D2UZ - $\frac{\partial^2 u}{\partial Z^2}$

D1VZ - $\frac{\partial v}{\partial z}$

D2VZ - $\frac{\partial^2 v}{\partial Z^2}$

D1A3Z - $\frac{\partial A^3}{\partial Z}$

DLZ - $\frac{(DX^2)(DY^2)}{(DX)^2(DY)^2}$

E. E - V at previous time step

EPS - convergence criterion

EUL - Enler number

EX - residual error in pressure iteration

F. FH - forcing function in pressure equation

FW - factor in wind stress calculation formula

G. G - dummy variable for V (for future time step)

H. H - dummy variable for U (for future time step)

HI - nondimensional depth = $\frac{h}{H}$

HREF - reference depth

HX - $\frac{\partial H}{\partial \alpha}$

HY - $\frac{\partial H}{\partial \beta}$

I. IN - maximum number of grid points in x-direction

IWN - maximum number of half-grid points in x-direction, IWN = IN - 1

I - index of x-axis, main grid

ITN - index for number of iterations

IW - index for x-axis, half grid

IRUN - index for number of runs
= 0, first run
= 1, from second time onwards

ISGNX, ISGNY - determine signs of TAUX and TAUX respectively

J. J - index for y-axis, main grid

JW - index for y-axis, half grid

JWN - maximum number of half-grid points in y-direction
JWN - JN - 1

JN - maximum number of main grid points in y-direction

K. K - index for Z-axis

KSTORE - specified usage of tape for storing results

KN - maximum number of main grid points in Z-direction

KISS - surface heat transfer coefficient (nondimensional)

L - maximum length of the domain

LN - number of time steps to be computed

LLN - total number of time steps/LN

M. MAR - number to describe general location of a point in the main grid

MRH - number of describe general location of a point in the half grid

MAXIT - maximum number of iterations

O. OMEGA - relaxation factor

P. P - nondimensional pressure

PN - New pressure, nondimensional

PINTH - dummy variable for pressure (future time step)

R. R - dimensional density at main grid points

RE - Reynolds number

RB - Rossby number

RINTX - density integrated with respect to x

RINTY - density integrated with respect to y

RO - nondimensional density at main grid points

ROW - nondimensional density at half grid points

RREF - reference density (gm/cc)

RW - dimensional density at half grid points (gm/cc)

RADN - solar radiation (w/m^2)

T. T - nondimensional temperature at main grid points

TO - initial temperature (dimensional) (°C)

TAMB - ambient temperature (dimensional) (°C)

TAIR - air temperature (dimensional) (°C)

TAI - coefficient in front of convective terms in the energy equation, = 1.

TAH - $\frac{1}{P_e}$ where $P_e = R_e \times P_r$

TAV - $\frac{1}{P_e \varepsilon^2}$ where $\varepsilon = \frac{H}{L}$

TE - equilibrium temperature (dimensional) (°C)

TTOT - total time elapsed

TAUX - $\partial u/\partial \gamma$ (nondimensional)

TAUY - $\partial v/\partial \gamma$ (nondimensional)

TEM - dimensional temperature at main grid points

TEMW - dimensional temperature at half-grid points

TREF - reference temperature

TW - nondimensional temperature at half-grid points

TLL - temperature at the discharge point (nondimensional)

TSU - water surface temperature (nondimensional)

TDEW - dewpoint temperature (dimensional)

U. U - velocity in x-direction (nondimensional)

V. V - velocity in y-direction (nondimensional)

VVIS - vertical eddy viscosity (nondimensional)

W. W - velocity in Z-direction (nondimensional)

WH - W at half-grid points

WHLDT - time derivative of WH at lid (i.e., $\frac{\partial}{\partial t}(WH)/Z = 0$)

X. XINT - Integral of x terms on the right-hand side of Poisson's equation

X - horizontal coordinate across discharge

Y. YINT - Integral of y terms on the right-hand side of Poisson's equation

Y - horizontal coordinate across discharge

Z. Z - vertical coordinate

MARKER MATRICES

The following number convention is used for the MAR = matrix system, which classifies points (or nodes) on the main grid system = (Refer to Figure).

MAR = 0, points outside the region of interest.

MAR = 1, point on the far y-boundary.

MAR = 2, point on the near y-boundary.

MAR = 3, point on the near x-boundary.

MAR - 4, point on the far x-boundary.

MAR = 5, outside corner on near x-boundary and far y-boundary.

MAR = 6, inside corner on far x-boundary and far y-boundary.

MAR = 7, outside corner on near x-boundary and near y-boundary.

MAR = 8, inside corner on near x-boundary and near y-boundary.

MAR = 9, outside corner on far x-boundary and near y-boundary.

MAR = 10, outside corner on far x-boundary and far y-boundary.

MAR = 11, points in the interior of the region of interest.

The following number convention is used to describe the MRH (matrix for the half-grid system).

MRH = 1, corner at far x-boundary and far y-boundary.

MRH - 2, points on near y-boundary.

MRH = 3, points on near x-boundary.

MRH = 4, corner at near x and near y-boundaries.

MRH = 6, far corner on x-axis.

MRH = 7, corner at far x and y-boundaries.

MRH = 9, interior grid points.

SECTION 5

PREPARATION OF RUNS

This section presents the steps to be followed in order to run the model for a particular location.

1. The boundaries are chosen depending on the particular situation, the general idea being to include all inflows and outflows. If a heated discharge enters the body of water the region of interest must be chosen so as to include this since it is a major factor in determining the size and spread of the resulting plume.

2. The grid size is chosen depending on the resolution required. The user should remember that the choice of the grid size directly determines the maximum allowable time step since this is directly related by the various stability criteria. (See choice of time step in Section 6.)

3. Specify number of full-grid points IN, JN, KN and number of half-grid points IWN, JWN. Since the actual domain may be smaller than the total rectangular region, INxJNxKN, the marker matrices MAR and MRH are used to specify the domain so that points outside the domain of interest skip the subsequent calculations.

4. IRUN is specified (= 0 for the first run, = 1 for subsequent runs). KSTORE is specified to indicate whether any tape has been assigned to store results of the run.

   KSTORE = 0 if no tape has been assigned.

   = 1 if tape has been assigned.

   LLN is specified to denote the number of hours of simulation to be carried out.

5. The depths at various places within the domain are specified using subroutine HEIGHT. The various inflows and outflows to the domain are specified using INLET1. (For details please refer to Biscayne Bay run, Sengupta et al. (1975).)

6. The various data like solar radiation, wind speed, wind direction and dewpoint temperature are specified in a data file which is made by the main program.

   For further details see the next section.

SECTION 6

INPUT DATA

The data that is required for the execution of the main program is listed in Table 3 in the order it appears. Note, the data input symbols have already been defined in Section 4. Moreover, the following remarks should be observed.

* Free format is used for all data input.
* Distinction must be made for integer and real number.
* The order of the cards must be followed.

SECTION 7

PLOTTING PROGRAMS

The plotting programs for the 3-D rigid-lid model are distinct from the main program and subroutines used to run it. The user has an option of either using a tape (Unit 8) during running the main program TMAINN to store the results or just run it without storing the results. For making subsequent continuation runs of TMAINN all that is required is the result of the last hour in the previous run. For plotting, however, one needs the results of all the hours for which results are to be plotted. These results are used as input data to run the various plotting programs.

DESCRIPTION OF PLOT PROGRAMS

The following are the main plotting programs.

PLOT - plots surface isotherms.

PLUV - plots u, v components of the velocities (i.e., 'K' sections).

PLUW - plots u, w components of the velocities (i.e., 'j' sections).

PLVW - plots v, w components of the velocities (i.e., 'i' sections).

SUBROUTINES

The various plot programs and subroutines are shown in Table 4.

Other subroutines seen in these programs (e.g., ARROHD, FLINE, etc.) are standard FORTRAN subroutines used for plotting, using a CALCOMP x,y plotter, and are hence ommitted in the above listing.

REFERENCES

Lee, S., Sengupta, S., Nwadike, E. V. and S. K. Sinha. Verification of Three-Dimensional Rigid-Lid Model at Lake Keowee. Technical Report 1980, NASA Contract NAS10-9410.

Sengupta, S., Lee, S. S. and R. Bland. Numerical Modeling of Circulation in Biscayne Bay. Transaction of the American Geophysical Union, June 1975.

Sengupta, S. and W. Lick. A Numerical Model for Wind-Driven Circulation and Temperature Fields in Lakes and Ponds. FTAS/TR-74-98, 1974.

Wilson, B. W. Note on Surface Wind Stresses Over Water at Low and High Wind Speeds. Journal of Geophysical Research, Vol. 65, No. 10, 1960.

APPENDICES

APPENDIX A

EXAMPLE CASE

INTRODUCTION

The area of interest is Lake Keowee in South Carolina, which was formed from 1968 through 1971 by damming the Little and Keowee rivers. The lake is located about 40 km west of Greenville and constitutes Duke Power Company's Keowee-Toxaway complex.

Lake Keowee has two arms connected by a canal (maximum depth 30.5 m). There are three power plants on the lake, namely, the Oconee Nuclear Station, Keowee hydro station and Jocassee-pumped storage station. The Oconee Nuclear Station is a three unit steam-electric station with an installed capacity of generating 2580 MW. The Oconee Nuclear Station draws in condenser-cooling water from the lower arm of Lake Keowee and discharges the heated effluent to the upper arm of the lake. The intake structure for the condenser-cooling water allows water from 20 to 27 m depth (full pond) to pass through. The discharge structure has an opening from 9 to 12 meters below the water surface (full pond) through which the CCW returns directly to the upper branch of the lake.

Lake Jocassee is located north of Lake Keowee and is used as a reservoir for Jocassee-pumped storage station. Lake Keowee also serves as the lower pond for this station. The Jocassee station has reversible turbines with a maximum generating flow (into Lake Keowee) of about 820 m^3/sec and a maximum pumping flow (out of Lake Keowee into Lake Jocassee) of about 775 m^3/sec, the net flow into Lake Keowee from Jocassee being about 15.5 m^3/sec.

Lake Keowee has a full pond elevation of 243.8 m above MSL. At full pond it has a volume of approximately 1.18×10^9 m^3, an area of 74 km^2, a mean depth of 15.8 m and a shoreline of about 480 km. The outflow from Lake Keowee is through Keowee hydro station and may vary from approximately 1.4 m^3/sec (leakage) to 560 m^3/sec. Maximum allowable draw-down of the lake is 7.6 m.

A map of the area of interest is shown in Figure 3.

PROBLEM STATEMENT

The objective of the present work is to find the three-dimensional temperature and velocity distributions in the region where the effects of the thermal discharge are noticeable. The effects of Jocassee-pumped storage station, Keowee hydro station as well as the meteorological conditions have been incorporated.

The region of interest is chosen to include the effects of the Oconee Nuclear Station discharge, the outflow through Keowee dam and the impact of the Jocassee-pumped storage station on the velocity and temperature distributions in Lake Keowee. The depth of the domain is cut off at 16 meters, since this is the level at which the thermocline occurs. Hence, for running the model, a constant depth region is considered. The plan view of the domain is shown in Figure 4. (Note: For variable depth refer to Biscayne Bay simulation studies by the University of Miami thermal pollution group.) In this figure, AB is an open boundary which takes care of the flow from or to the Jocassee-pumped storage station. 'C' shows the position of the flow in the canal connecting the two arms of the lake. 'D' is the discharge point for the Oconee Nuclear Station and 'E' is the outflow from Keowee hydro station.

The inclusion of the above results in a domain 2895.6 m x 2438.4 m in the horizontal plane. The horizontal grid size (in x and y directions) is 152.4 m x 152.4 m, giving a total of 20 x 17 (= 340) nodes in the horizontal plane, out of which 293 lie in the region of interest. The 16 m constant depth region of interest is divided into 4 equal slices of 4 m each, giving at total of 5 nodes in the vertical (Z)/direction. Hence, there are 293 x 5 nodes (grid points) in the region of interest. This region is specified using the MAR and MRH marker matrices (Figure 5 and Figure 6).

Boundary Conditions

On the Jocassee effect boundary, the flow velocity (varying with time) is specified. Open-boundary condition ($\frac{\partial T}{\partial y} = 0$) is specified for the temperature.

The same is done for the Keowee hydro boundary. The only difference is that the values specified are at three points in the vertical plane (i.e., at K = 1, 2 and 3) since this region covers the discharge area.

For the Oconee Nuclear Station, the discharge velocity as well as the discharge temperature is specified at the discharge point.

Open-boundary conditions are specified for the temperature and velocity at the canal. This, however, leads to a possible violation of mass balance in the region of interest. This mass unbalance will actually show up as a variation in the water level in the lake which is beyond the capability of the rigid-lid model.

At all solid boundaries as well as the artificial bottom (since the bottom is cut off at 16 m) perfect insulation (temperature gradient = 0) and zero velocity conditions are assumed.

At the surface, the vertical component of the velocity is specified as zero (rigid-lid constraint). Surface wind shear stress and heat transfer coefficient are specified.

Initial Conditions

The initial values of the water velocities are assumed to be zero. The initial temperature of the lake is assumed to be equal to the ambient water temperature (determined by running a one-dimensional model) and is taken to be uniform throughout the domain.

CALCULATION OF PARAMETERS AND INPUT DATA

Reference Quantities

Reference length = L = maximum length of the domain = 2895.6 m.

Reference horizontal eddy viscosity A_{ref} = 0.002 L 4/3

= 38311.48 cm²sec.

For better agreement with data the value chosen is 60,000 cm²/sec.

Reference depth = H = 16 m.

Reference vertical A_v = 0.002 x (H) 4/3.

Eddy viscosity = 37.43 cm²/sec.

Reference velocity = V_{ref} = 30 cm/sec.

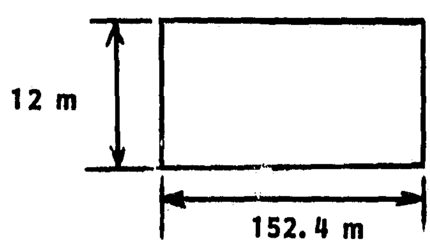
Reference temperature = T_{ref} = 10.0°C.

Reference time = L/V_{ref} = 9652 sec.

Calculation of Inflows and OUtflows into the Domain (Used in INLET1)

Oconee Nuclear Station Discharge Velocity--

The discharge is considered to take place through a point at a depth of 12 m (k = 3). The discharge velocity is calculated as follows:



The total discharge into the basin is equal to:

$$(100 \frac{cm}{m} \times V \times 152.4 \times 12) = Q$$

where Q = average discharge in m^3/sec

$$\therefore V = \frac{8144.1}{60} \text{ m/sec}$$

$$= 7.42207 \text{ cm/sec}$$

The average values of Q over 24 hrs is taken since the variation is negligible.

Nondimensional discharge velocity $= \frac{V}{V_{ref}} = \frac{V}{30} = 0.24740$

Keowee Hydro Discharge Velocity--

The outflow through the Keowee hydro station is through a channel 152.4 m x 12 m.

The volume flowrate Q = (152.4 x 12 x V) m^3/sec

where V = discharge velocity (m/sec)

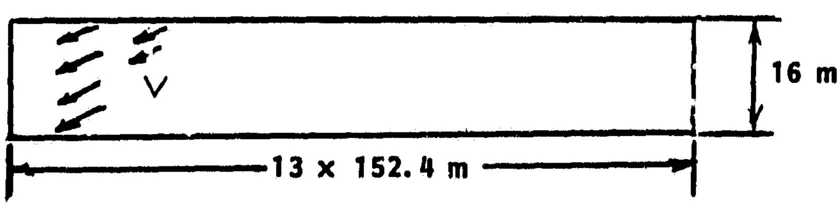
$$\therefore V = [Q/(152.4 \times 12)] \text{ m/sec} = \left(\frac{Q}{152.4 \times 12 \times 100}\right) \text{ cm/sec}$$

Q is specified as a function of time in INLET1.

The procedure for nondimensionalization is similar.

Jocassee Flow Velocity--

The entire flow to or from the Jocassee-pumped storage station is assumed to take place through the entire upper boundary (AB in Figure 4). The flow through this area (shown below) is assumed to be uniform and is assumed to take place simultaneously with the outflow through the Jocassee station.



$V = Q/[(16 \times 13 \times 152.4) \times 100]$ cm/sec.

Q = flow through Jocassee (m^3/sec).

Q is positive when Jocassee is generating (i.e., the flow is into the region of interest) and negative when pumping (i.e., flow out of region of interest).

SAMPLE INPUT

The following are the inputs to TMAINN contained in the data file IPUT (which includes values calculated earlier).

Input #	No. of Data In Card	Symbol		Value
1	3	IRUN	=	0
		KSTORE	=	1
		LLN	=	3
2	2	VVIS	=	37.43/60,000 = 0.00062384
		ABR	=	0.78
3	4	AI	=	1.0
		AH	=	$\frac{60,000}{30 \times 2895 \times 100} = 0.01228172$
		AV	=	$(\frac{2895.6^2}{16})$ AH = 402.08304
		AP	=	1.0
4	4	EPS	=	0.001
		MAXIT	=	60
		OMEGA	=	1.8
		ARBP	=	1.0
5	3	DX	=	152.4/2895.6 = 0.05263
		DY	=	0.05263
		DZ	=	4/16 = 0.25
6	3	TAI	=	1.0
		TAH	=	AH = 0.01228172
		TAV	=	AV = 402.08304
7	3	A	=	1.000428
		B	=	-0.000019

Input #	No. of Data In Card	Symbol	Value
		C	= −0.0000046
8	1	TO	= 10.0
9	3	EUL	$= \frac{980 \times (16\times100)}{30^2} = 1742.222$
		CW	= 0.0
		CB	= 0.0
10	2	AA	= 0.24740
		CC	= 16/16 = 1.0
11	1	TLL	$= \frac{31.7 - 10}{10}$
12	1	TAU	= 0.0152 cm^2/sec
13	1	DT	Criterion (convective) $= \Delta t < \frac{\Delta x}{U} = \frac{152.4 \times 100}{30}$ $= 504$ secs > 504 secs Hence, convective criterion dominates; choose ΔT = 300 secs $DT = \frac{\Delta T}{t_{ref}} = \frac{300}{9652} = 0.03108164$ Note: choose best time step by trial and error
14	1	CTTOT	$= t_{ref}/3600 = 2.681111$
15	1	ISOTOP	= 0
16	6	WS	See Table 5
		TSU	
		TDEW	
		RADN	
		ISGNX	
		ISGNY	
17	1	ANGLE	See Table 5

LAKE KEOWEE APPLICATION-EXECUTION DECK

The following execution deck is for use in the UNIVAC 1100 computer at the University of Miami. These may have to be modified if a different computer is used.

(ALL PROGRAMS AND SUBPROGRAMS COMPILED AND STORED IN FILE)

First Run

1. @ ASG, AX FILE.

   (THE FILE IS ASSIGNED FOR THE RUN)

2. @ ASG,T 8, 16N, TAPENAME.

   (A TAPE FILE NAMES '8' IS BEING ASSIGNED. THE TAPE IS 9-TRACK, AND THE REEL NUMBER IS 'TAPENAME')

3. @ PRT,S FILE. TMAINN

   (THE MAIN PROGRAM IS PRINTED)

4. @ PACK FILE.

   (THE FILE IS PACKED)

5. @ PREP FILE.

   (ENTRY POINT TABLE IS PREPARED)

6. @ MAP,S
7. IN FILE. TMAINN
8. LIB FILE.
9. END
10. @ XQT
11. 0

    (VALUE FOR IRUN,FIRST RUN: IRUN=0)

12. 24

    (NUMBER OF HOURS REQUIRED, MINIMUM=1 HOUR, MAX=24)

13. 0

(0 IF MAGNETIC TAPE IS REQUIRED TO STORE RESULT, IF NOT, ANY NUMBER)

14. @ ADD FILE. INPUT

(INPUT DATA FILE FOR THE PARTICULAR RUN)

15. @ FIN

EXECUTION DECK FOR PLOT PROGRAMS

1. @ ASG,AX FILE.

2. @ ASG,T 8., 16N, TAPENAME.

3. @ ASG,T 11., 16N, PLOTTAPE.

(A MAGNETIC TAPE FILE NAMED '11' IS BEING ASSIGNED. THE TAPE IS 7-TRACK AND THE REEL NUMBER IS 'PLOTTAPE'. THE PLOTS ARE STORED ON THIS TAPE)

4. @ PRT,S FILE.PLOTTER

(THE PLOT PROGRAM IS PRINTED)

5. @ PACK FILE.

6. @ PREP FILE.

7. @ MAP,S

8. IN FILE.PLOTTER

9. LIB FILE.

10. END

11. @ XQT

12. @ ADD FILE. INPUT

13. @ FIN

Table 2. Subroutines Required in Main Program TMAINN

No.	Name	Description	Remarks
1	DVISV	Computes D1VY, D2VY, D1VX and D2VX.	Called by subroutine INTE. Schemes used similar to DVISU.
2	DVISU	Computes D1UX, D2UX, and D1UY.	Called by INTE. $\frac{\partial \mu}{\partial \alpha}$, $\frac{\partial \mu}{\partial \beta}$ are computed at interior, boundary or corner pts by scheme similar to the one used in DINERU.
3	DVVY	Computes D1HVVY.	Called by INTE. $\frac{\partial}{\partial \beta}$ (hvv) is computed for interior, boundary or corner by a scheme similar to the one used in DINERU.
4	DUVY	Computes D1HUVY.	Called by INTE. $\frac{\partial}{\partial \beta}$ (huv) is computed for interior, boundary and corner pts by a scheme similar to the one used in DINERU.
5	DINERU	Computes D1HUUX and D1HUVX.	Called by INTE. The results are used in Poisson equation for pressure.
6	TPRINK	Prints temperatures at a grid point.	Called by TMAINN.
7	PRUV	Prints the values of U and V at all main grid points.	Called by TMAINN.
8	PRITEX	Prints the No. of iterations (ITN) and final residual error in solving the Poisson equation	Called by TMAINN.
9	TPRIN1	Prints the input parameters.	Called by TMAINN.
10	STORE2	Stores values of input parameters and physical quantities on tape #8	Called by TMAINN.
11	RWR	Computes real vertical velocities from modified vertical velocities used in equations at integral grid points.	Called by TMAINN.

Table 2. Subroutines Required in Main Program TMAINN (Continued)

No.	Name	Description	Remarks
12	RWRH	Computes real vertical velocities at half-grid points.	Called by TMAINN.
13	DENSTY	Uses the equation of state and computes density field from the temperature field.	Called by TMAINN.
14	TEQB	Allows for vertical mixing at a particular grid point. Program is called by TMAINN.	If the temp at the grid pt just above it is less and the difference is more than a specified maximum, the two temperatures are averaged.
15	OLDT	Sets the values of temperature field at time step 'n' equal to the temperature field at (n+1) after all computations for time step 'n' are completed.	
16	TEMB2	Computes temperatures at the boundary points in the domain of interest.	Called by TMAINN.
17	TEMI4	Computes temperatures at the interior points of the domain of interest.	Called by TMAINN.
18	RWH	Computes vertical velocities at half-grid points.	Called by TMAINN.
19	OLDUV	Sets the values of D and E equal to U and V respectively in order to retain values of U and V at one time step lag.	Called by TMAINN.
20	UVTOP	Computes U and V at the top using wind stress boundary conditions.	Called by TMAINN. Computations are made for MAR = 11 only (internal grid points).
21	UVT	Computes U and V for variable density at successive time steps.	Called by TMAINN.

Table 2. Subroutines Required in Main Program TMAINN (Continued)

No.	Name	Description	Remarks
22	PRE1L	Computes pressure for far field from Poisson's Equation at half-grid points.	Called by TMAINN.
23	FORCE	Computes R.H.S. of Poisson's Equation at half-grid points.	Called by TMAINN.
24	DPSXY	Computes DPSX and DPSY.	Called by TMAINN.
25	ROINTY	Computes Y_p in the Poisson's Equation.	Called by TMAINN.
26	ROINTX	Computes X_p in the Poisson's Equation.	Called by TMAINN.
27	CORINT	Adds integral of Coriolis' component XINT and YINT.	Called by TMAINN.
28	INTE	Computes XINT, YINT, DPSX, and DPSY.	Called by INTE.
29	WHATIJ	Computes the values of W at I, J from the values of WH at IW, JW.	Called by TMAINN.
30	WHTOP	Sets the value of WH equal to zero at the surface.	Called by TMAINN.
31	ERROR	Calculates "Hirt and Harlow" correction term at half-grid points and at the surface (WHLDT).	Called by TMAINN.
32	READ2	Reads in input parameters and physical quantities stored on tape #7.	Corresponds to store 2. Called in by TMAINN.
33	INLET1	Puts in velocities u and v pheme discharge, etc. into the model.	Called by TMAINN.
34	HEIGHT	Inputs depths of the basin into the model.	This subroutine is for a constant depths model. Called by TMAINN.

Table 2. Subroutines Required in Main Program TMAINN (Continued)

No.	Name	Description	Remarks
35	INITIT	Sets initial temperature field.	Sets the temperature field equal to ref temp at all grid points. Called by TMAINN.
36	INITIA	Initializes values of U, V, WH, W, D, E and PINTH.	Called by TMAINN.
37	READ 3K	Classifies region of interest into interior, corner and boundary points using matrix MAR.	Called by TMAINN.
38	IPUT	Data files containing values of input data for the respective days.	

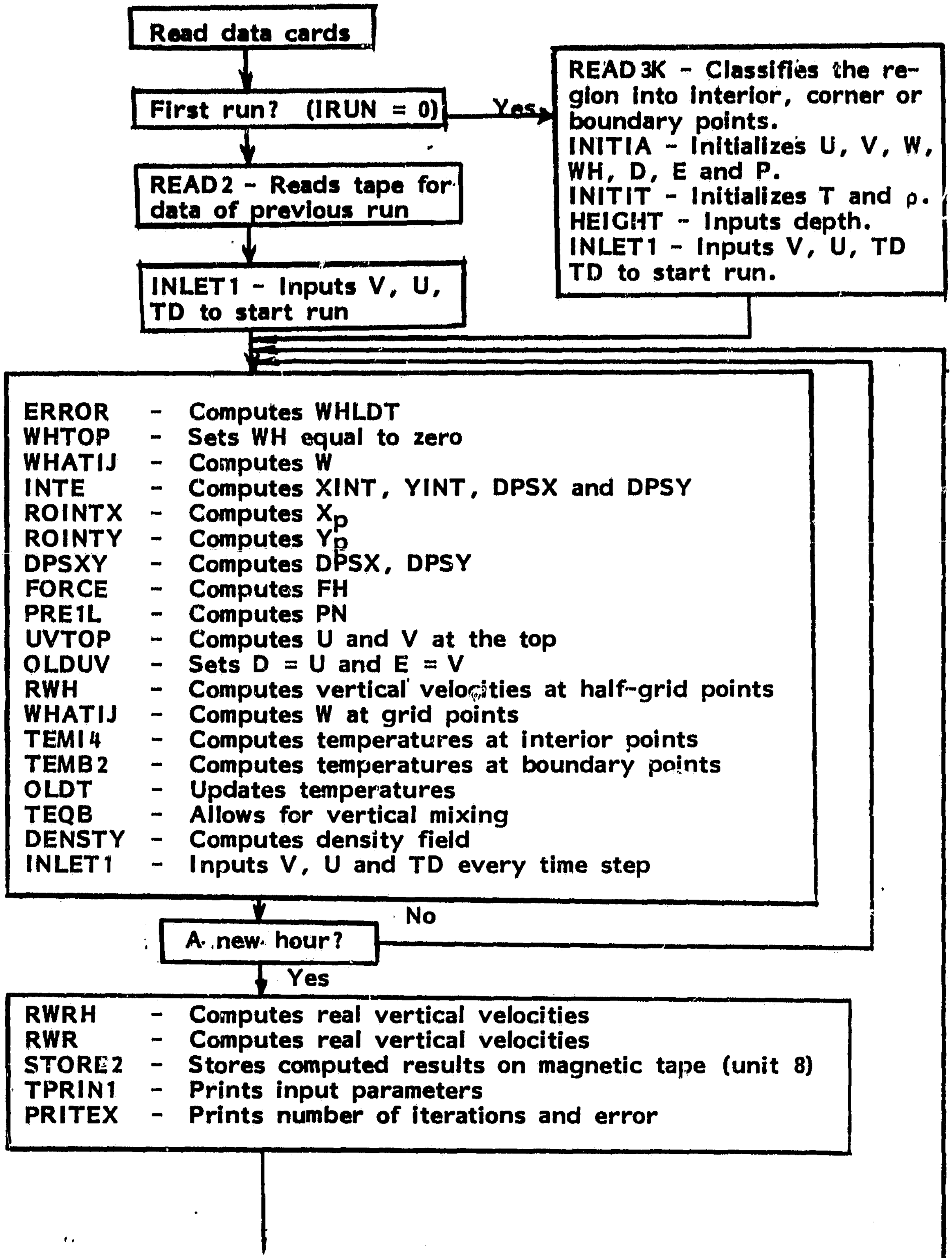


Figure 1. Flow chart (main program)

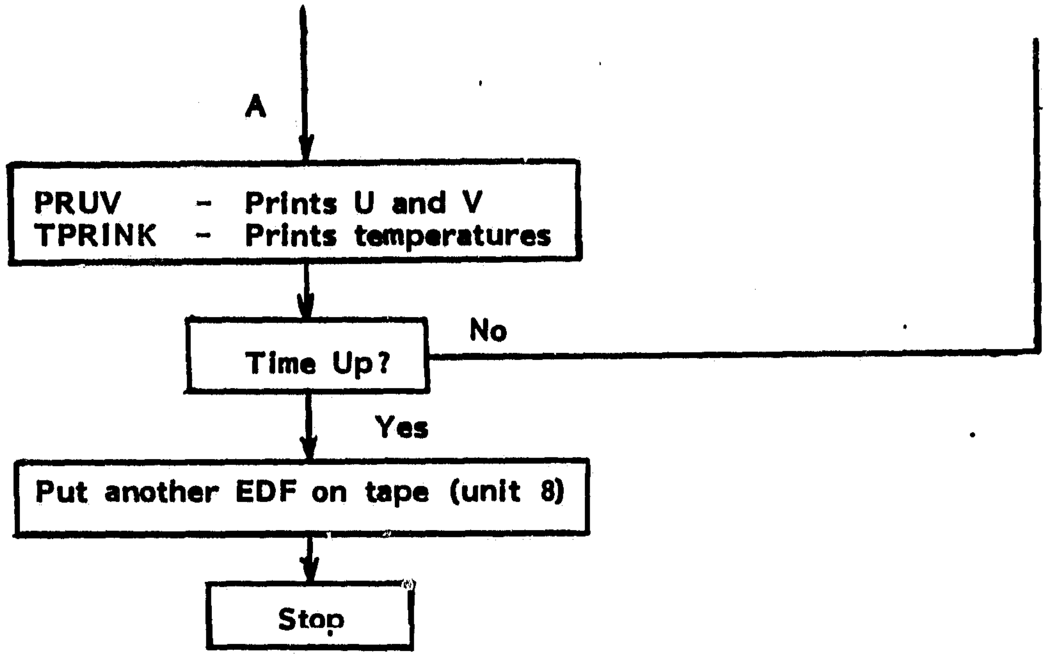


Figure 1 (Continued). Flow chart (main program)

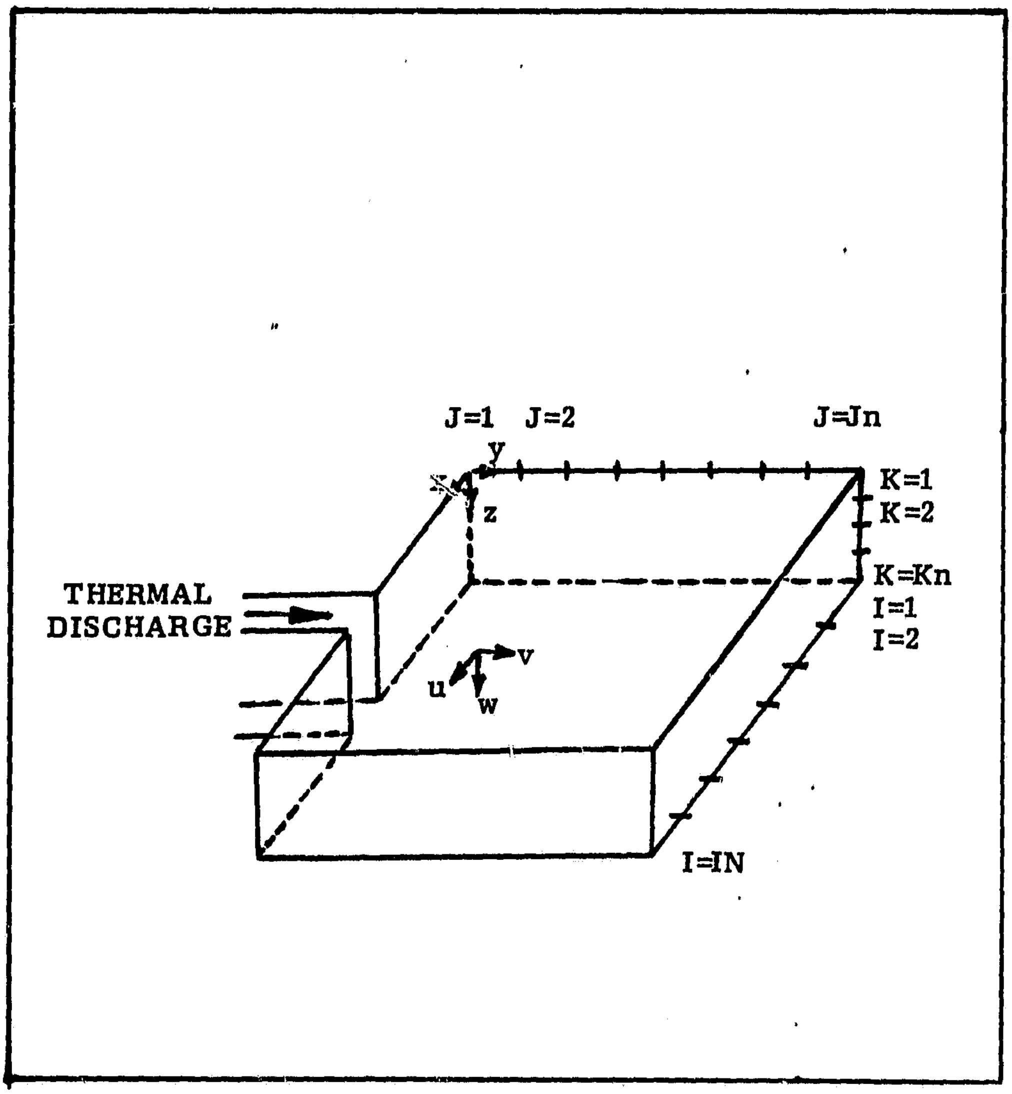


Figure 2. Coordinate and grid system

Table 3. Input Data to TMAINN

Input #	No. of Data In Card	Symbol	Definition /Value
1	3	IRUN	= 0 for first run
		LLN	= No of hours of simulation
		KSTORE	= 0 if no tape is assigned = 1 if tape is assigned
2	2	VVIS	= Nondimensional vertical eddy viscosity
		ABR	= 1/Rossby No. = $\frac{fL}{U_{ref}}$
3	4	AI	= Coefficient in front of inertia term = 1.0
		AH	= 1/Reynolds No. = $\frac{\text{Ref eddy hoz viscosity}}{U_{ref} \cdot L}$
	4	AV	= $(1/\varepsilon^2 Re)$ $(\varepsilon = H/L)$
		AP	= Coefficient in fron of pressure term = 1.0
4	4	EPS	= Convergence factor = 0.001
		MAXIT	= Maximum number of iterations for Poisson Equation
		OMEGA	= Relaxation factor = 1.8
		ARBP	= Arbitrary pressure = 1.0
5	3	DX	= Horizontal grid spacing (x dir.)
		DY	= Horizontal grid spacing (y dir.) = $\Delta y/L$
		DZ	= Vertical grid spacing (z dir.) = $\Delta z/H$
6	3	TAI	= Coefficient of convective terms in energy equation = 1.0
		TAH	= Horizontal eddy diffusivity = AH (usually)

Table 3. Input Data to TMAINN (Continued)

Input #	No. of Data In Card	Symbol	Definition/Value
		TAV	= Vertical eddy diffusivity = AV (usually)
7	3	A	= 1.000428 These are coefficients = -0.000019 in the equation of = -0.0000046 state for water where $\rho = A + BT + CT^2$ (gm/cc)
8	1	TO	= Reference temperature (°C)
9	3	EUL	= Euler No. = $\frac{gH}{(U_{ref})^2}$
		CW	= Temperature gradient at vertical boundary
		CB	= Temperature gradient at the bottom
10	2	AA	= Nondimensional discharge velocity = (discharge velocity) $/U_{ref}$
		CC	= No dimensional depth = h/H_{ref}
11	1	TLL	= Nondimensional discharge temperature = $(T_D - T_o)/T_o$
12	1	TAU	= Surface shear stress (from Wilson Curve) (Refer to Figure 7)
13	1	DT	= Nondimensional time step = $\Delta T(L/U_{ref})$
14	1	CTTOT	= Converts nondimensional time to hours
15	1	ISTOP	= Number of hours of previous run
16	6	WS	= Wind speed (m/sec)
		TSU	= Air temperature (°C)
		TDEW	= Dewpoint temperature (°C)
		RADN	= Incident solar radiation (w/m²)

Table 3. Input Data to TMAINN (Continued)

Input #	No. of Data in Card	Symbol	Definition/Value
		ISGNX	= +1 if x component of W_s is negative = -1 if x component of W_s is positive = +1 if y component of W_s is negative = -1 if y component of W_s is positive
17	1	ANGLE	= Direction of W_s (degrees) with respect to the x axis

Table 4. Plotting Programs

No.	Name	Program Description	Remarks
1	PLOT	Plots surface isotherms	
2	PLUV	Plots velocities, K section	
3	PLUW	Plots velocities, J section	
4	PLVW	Plots velocities, I section	
5	ECHKON	Calculates equal temperature points	Called by PLOT
6	CONLIN	Draws the isotherms	Called by ECHKON
7	ENDER	Writes the values of the temperature on the isotherms	Called by ECHKON

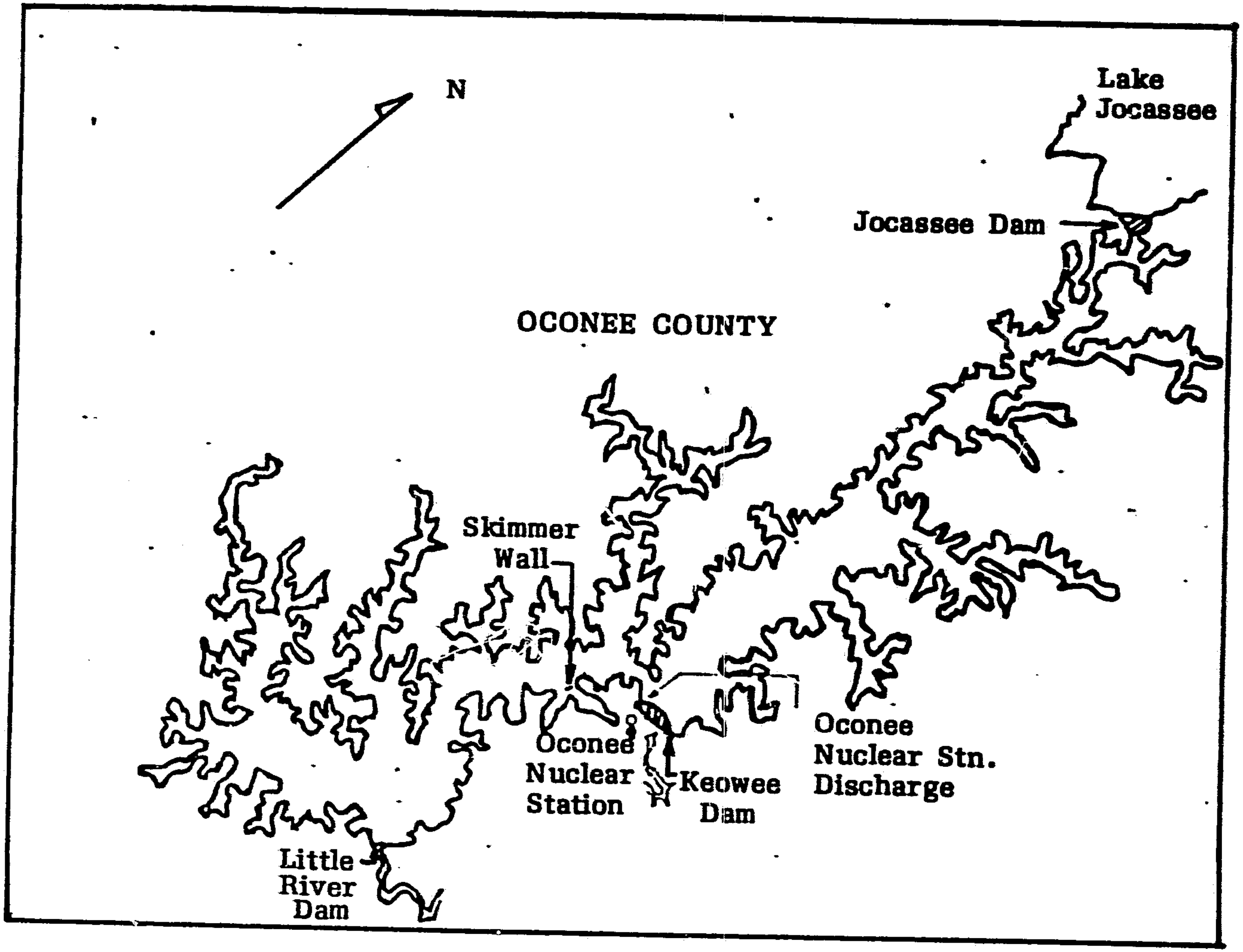
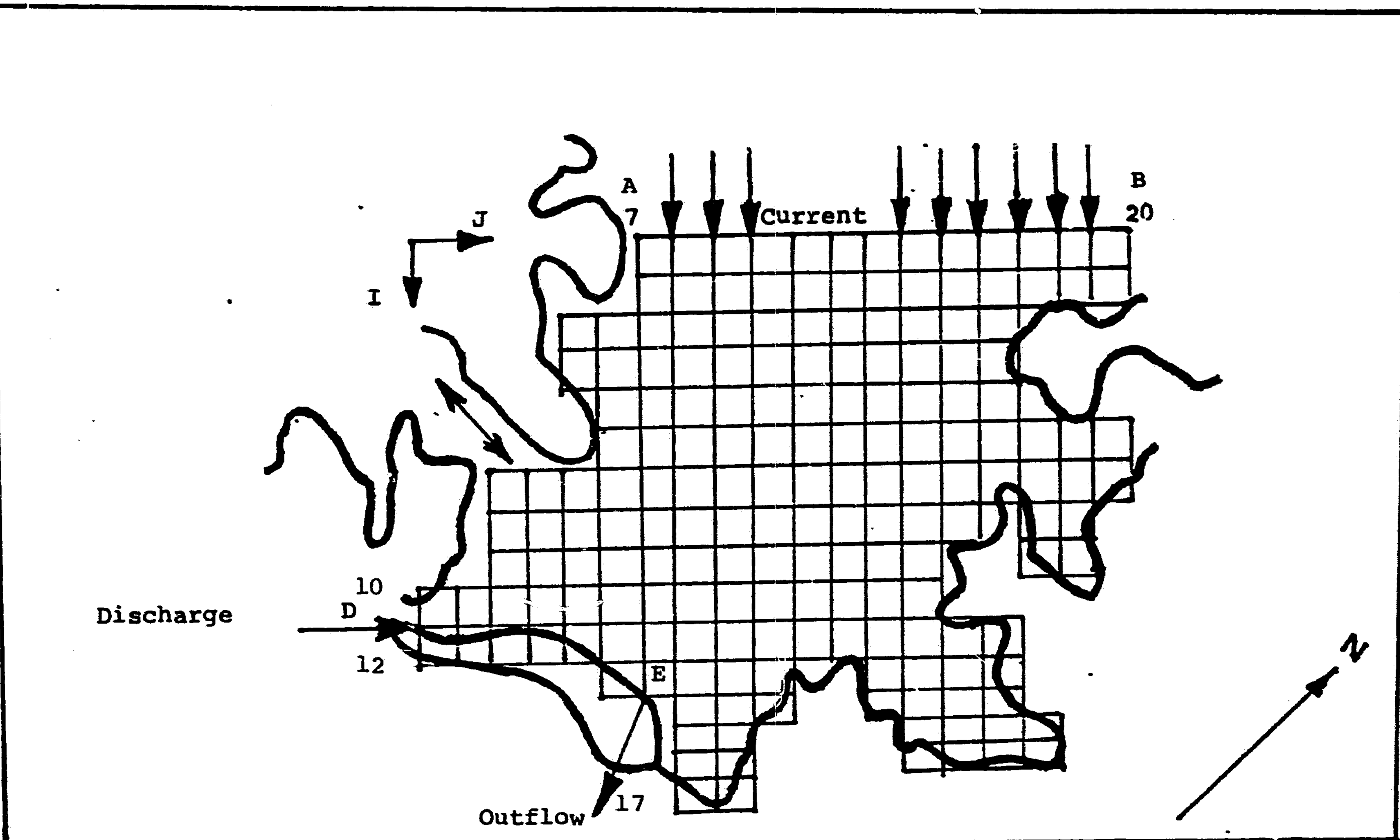


Figure 3. Lake Keowee



Fiqure 4. Lake Keowee (region of interest) showing inputs and outputs (for 3-D model)

1	0	0	0	0	0	0	7	3	3	3	3	3	3	3	3	3	3	3	3	5
2	0	0	0	0	0	0	2	11	11	11	11	11	11	11	11	11	11	11	11	1
3	0	0	0	0	7	3	8	11	11	11	11	11	11	11	11	11	6	4	4	10
4	0	0	0	0	2	11	11	11	11	11	11	11	11	11	11	11	1	0	0	0
5	0	0	0	0	9	8	11	11	11	11	11	11	11	11	11	11	1	0	0	0
6	0	0	0	0	0	2	11	11	11	11	11	11	11	11	11	11	6	3	3	5
7	0	0	7	3	3	8	11	11	11	11	11	11	11	11	11	11	11	11	11	1
8	0	0	2	11	11	11	11	11	11	11	11	11	11	11	11	6	8	11	6	10
9	0	0	2	11	11	11	11	11	11	11	11	11	11	11	6	10	2	11	1	0
10	7	3	8	11	11	11	11	11	11	11	11	11	11	11	1	0	9	4	10	0
11	2	11	11	11	11	11	11	11	11	11	11	11	11	11	6	5	5	0	0	0
12	9	4	4	4	4	8	11	11	11	11	6	4	8	11	11	11	1	0	0	0
13	0	0	0	0	0	9	4	8	11	11	1	0	2	11	11	11	1	0	0	0
14	0	0	0	0	0	0	0	2	11	6	10	0	9	8	11	11	6	5	0	0
15	0	0	0	0	0	0	0	2	11	1	0	0	0	2	11	11	11	1	0	0
16	0	0	0	0	0	0	0	2	11	1	0	0	0	9	4	4	4	10	0	0
17	0	0	0	0	0	0	0	9	4	10	0	0	0	0	0	0	0	0	0	0

Figure 5. MAR marker matrix

1	0	0	0	0	0	0	4	10	10	10	10	10	10	10	10	10	10	10	3
2	0	0	0	0	0	0	2	9	9	9	9	9	9	9	9	9	8	8	7
3	0	0	0	0	4	10	9	9	9	9	9	9	9	9	9	1	0	0	0
4	0	0	0	0	6	9	9	9	9	9	9	9	9	9	9	1	0	0	0
5	0	0	0	0	0	2	9	9	9	9	9	9	9	9	9	1	0	0	0
6	0	0	0	0	0	2	9	9	9	9	9	9	9	9	9	9	10	10	3
7	0	0	4	10	10	9	9	9	9	9	9	9	9	9	9	8	9	9	7
8	0	0	2	9	9	9	9	9	9	9	9	9	9	9	7	0	2	1	0
9	0	0	2	9	9	9	9	9	9	9	9	9	9	1	0	0	6	7	0
10	4	10	9	9	9	9	9	9	9	9	9	9	9	1	0	0	0	0	0
11	6	8	8	8	8	9	9	9	9	9	8	8	9	9	10	3	0	0	0
12	0	0	0	0	0	6	8	9	9	1	0	0	2	9	9	1	0	0	0
13	0	0	0	0	0	0	0	2	9	7	0	0	6	9	9	1	0	0	0
14	0	0	0	0	0	0	0	2	1	0	0	0	0	2	9	9	3	0	0
15	0	0	0	0	0	0	0	2	1	0	0	0	0	6	3	8	7	0	0
16	0	0	0	0	0	0	0	6	7	0	0	0	0	0	0	0	0	0	0

Figure 6. MRH marker matrix

Table 5. Meteorological Data for Lake Keowee (February 27, 1979)

Time (hrs from midnight)	Wind Speed (m/s)	Air Temp (°C)	Dewpoint Temp (°C)	Solar Radiation (w/m²)	Wind Direction (Degrees)
1	1.833	-0.33	-2.78	0.0	15°
2	1.073	-0.72	-1.67	0.0	75°
3	2.325	-1.61	-1.61	0.0	60°
4	1.565	-2.22	-2.28	0.0	15°
5	2.056	-1.83	-1.89	0.0	50°
6	1.788	-2.17	-2.22	0.0	85°
7	2.012	-2.72	-2.78	20.94	85°
8	2.280	-1.67	-2.78	195.39	60°
9	0.626	0.01	-3.33	369.85	5°
10	1.386	3.06	-2.22	544.31	75°
11	1.609	5.83	-2.22	655.31	15°
12	1.788	8.83	-1.39	725.75	40°
13	3.129	11.06	-2.78	746.68	80°
14	2.593	12.28	-5.00	704.81	70°
15	1.520	13.39	-5.56	579.20	80°
16	1.207	13.89	-5.56	383.81	75°
17	1.565	13.83	-5.61	146.55	55°
18	1.609	13.72	-3.33	20.94	15°
19	2.056	11.72	-4.44	0.0	30°
20	1.162	9.72	-2.78	0.0	25°
21	1.772	8.33	5.28	0.0	55°
22	2.861	7.78	5.56	0.0	55°
23	2.995	7.00	5.28	0.0	50°
24	1.386	5.28	3.89	0.0	60°

Table 6. Inflows and Outflows to Lake Keowee

Time Feb. 27, 1978	Oconee Discharge (m^3/min)	Oconee Discharge Temp (°C)	Net Jocassee Flow (C.F.S.)	Keowee Hydro Flow (C.F.S.)
12.00 a.m.	7505.3	18.6	-14395	48
1.00	7498.1	18.5	-18754	48
2.00	7492.0	18.4	-18805	48
3.00	7492.0	18.5	-18713	48
4.00	7491.6	18.3	-18698	48
5.00	7494.3	18.3	-18688	48
6.00	7488.2	18.3	-15939	48
7.00	7481.8	18.2	3484	3668
8.00	7485.6	18.3	16823	17540
9.00	7488.2	18.2	13503	8488
10.00	7497.7	18.3	5470	8096
11.00	7504.1	18.3	100	2680
12.00 p.m.	7503.4	18.4	100	48
1.00	7506.0	18.5	100	48
2.00	7506.4	18.5	100	48
3.00	7503.4	18.4	100	48
4.00	7501.9	18.4	100	48
5.00	7507.5	18.4	100	48
6.00	7511.0	18.4	100	48
7.00	7516.2	18.4	100	48
8.00	7518.9	18.3	100	48
9.00	7520.4	18.2	100	48
10.00	7516.6	18.2	100	48
11.00	7509.4	18.2	100	48
12.00 a.m.	7507.2	18.2	-4382	48

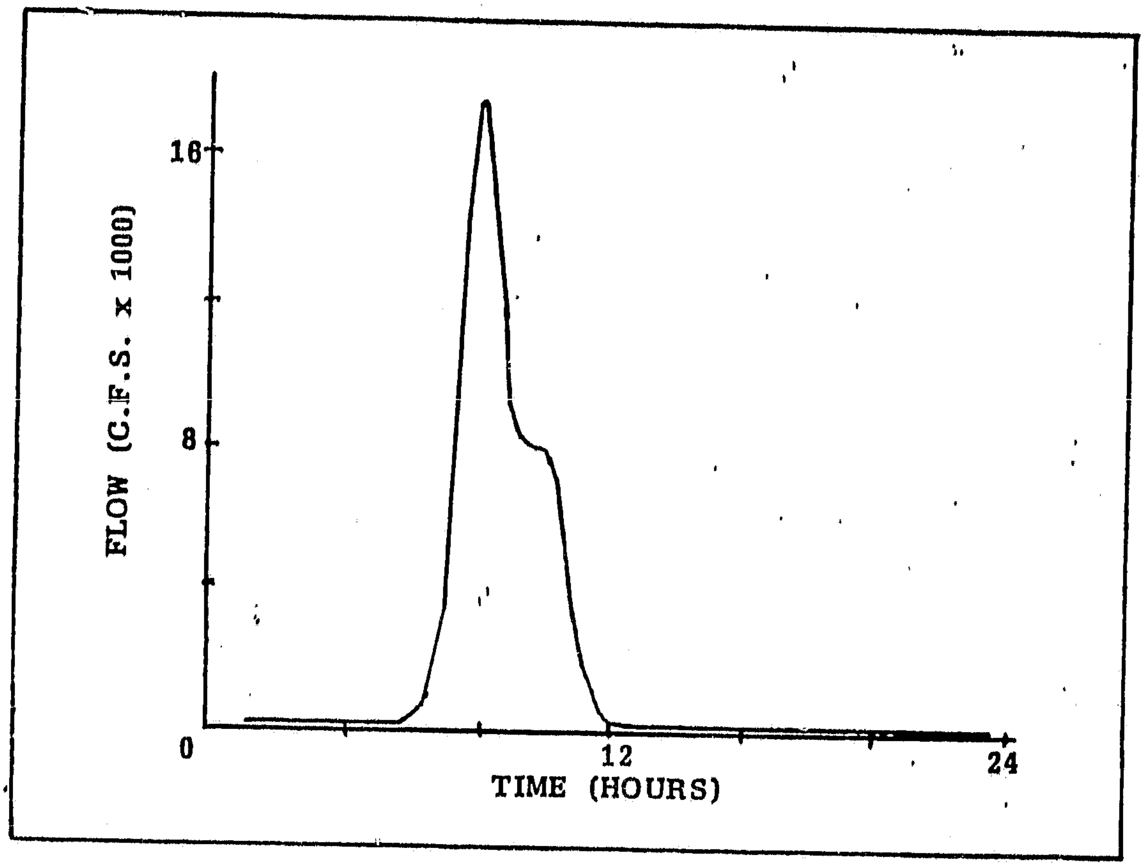


Figure 7. Keowee hydro discharge (February 27, 1979)

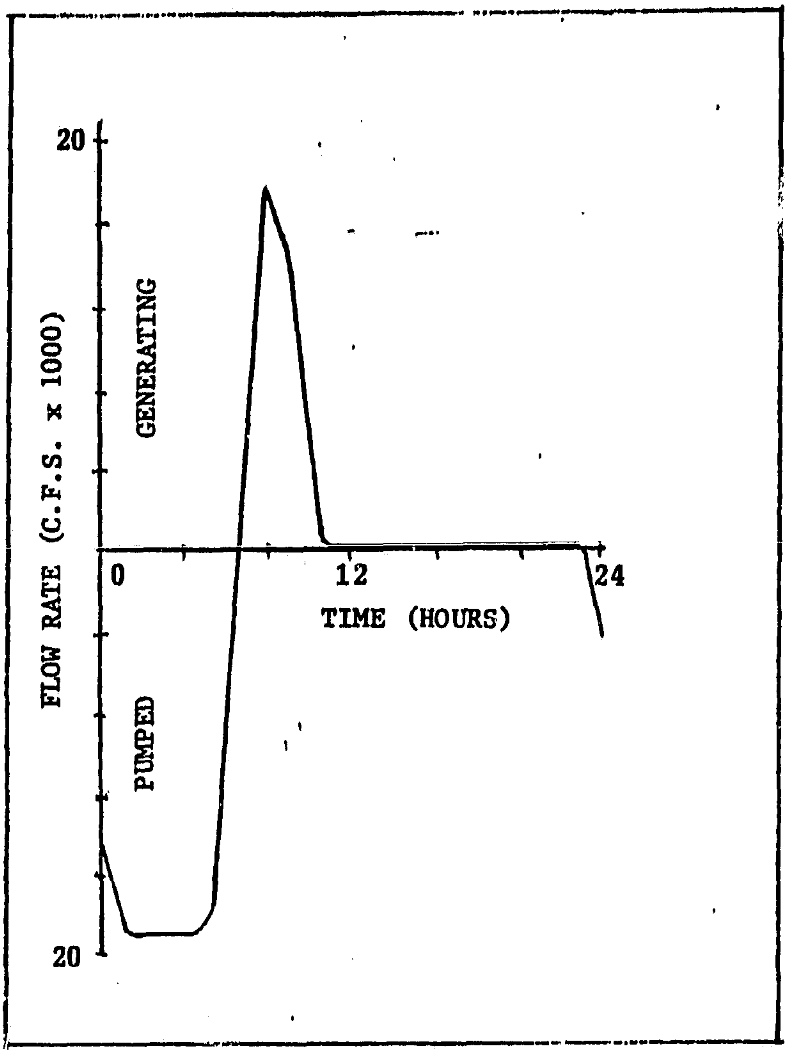


Figure 8. Jocassee-pumped storage station discharge data (February 27, 1979)

APPENDIX B

FORTRAN SOURCE PROGRAM LISTING

LIST OF MAIN PROGRAM AND SUBROUTINES

```
ASA*NASA(1).THAINN FOR CREATED ON 5 MAY 80 AT 10:40:24
      C**********************************************************************
      C*  THIS IS THE MAIN PROGRAM FOR THE 3-D RIGID-LID MODEL              *
      C**********************************************************************
      C
      C
            PARAMETER IN=17,JN=20,KN=5,IWN=16,JWN=19,KNM1=4
            DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IWN,JWN,KN),
           CWR(IN,JN,KN),WRH(IWN,JWN,KN),P(IWN,JWN),D(IN,JN,KN),E(IN,JN,KN),
           CWHLDT(IWN,JWN),XINT(IN,JN),YINT(IN,JN),H(IN,JN,KN),G(IN,JN,KN),
           CHI(IN,JN),HX(IN,JN),HY(IN,JN),HAR(IN,JN),MRH(IWN,JWN),FH(T-N,JKN)
           CDPSX(IN,JN),DPSY(IN,JN)
            DIMENSION A3(KN)
            DIMENSION T(IN,JN,KN),IP(IN,JN,KN),ID(IN,JN,KN),RO(IN,JN,WN),
           CRINTX(IN,JN,KN),RINTY(IN,JN,KN),WD(IN,JN,KN),TACTUL(IN,JN,KN)
            DIMENSION TW(IWN,JWN,KN),ROW(IWN,JWN,KN)
            DIMENSION UA(IN,JN,KN),VA(IN,JN,KN)
            REAL KISS
            INM1=IN-1
            READ 1,IRUN
            READ 1,LLN
            READ 1,KSTORE
      1     FORMAT()
            TTOT1=0.
            HTTOT=0.0
            READ 2, VVIS,AGR
            READ 2, AI,AH,AV,AP
            READ 2, EPS,MAXIT,OMEGA,ARBP
            READ 2, DX,DY,DZ
            READ 2, TAI,TAH,TAV
            READ 2, A,B,C
            READ 2, TO
            READ 2, EUL,CW,CB
            READ 2,AA,CC
      C     READ 2,TLL
            A3(1)=VVIS
            A3(2)=VVIS
            A3(3)=VVIS
            A3(4)=VVIS
            A3(5)=VVIS
            B3=VVIS
            READ 2,TAU
            READ 2,DT
            READ 2,CTTOT
            READ 2,ISTOP
      2     FORMAT ()
            DL2=DX*DX
            TREF=TO
            RREF=A+B*TO+C*TO*TO
            IF(IRUN.GT.0) GO TO 3
            CALL READ3K(I,J,IN,JN,IW,JW,IWN,JWN,HAR,MRH)
            CALL INITIA(IN,JN,KN,IWN,JWN,U,V,W,WH,D,E,
           CP,I,J,K,IW,JW,ARBP)
            CALL INITIT(I,J,K,IN,JN,KN,IW,JW,IWN,JWN,A,B,C,T,RO,HAR,MRH,TREF,
           CRREF,TW,ROW,TO)
            CALL HEIGHT(I,J,K,IN,JN,KN,HI,HX,HY,CC)
            CALL INLET1(I,J,K,IN,JN,KN,U,V,H,G,T,TD,AA,TLL,DT,HTTOT)
            GO TO 4
      3     CONTINUE
            CALL READ2(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,HI,
           CHAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,A,
           C,CB,CW,A,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,TAME,TTOT)
            PRINT 333, TTOT,TAUX,TAUY
      333   FORMAT(4X,3E16.6)
            HTTOT=CTTOT*TTOT
            CALL INLET1(I,J,K,IN,JN,KN,U,V,H,G,T,TO,AA,TLL,DT,HTTOT)
      4     CONTINUE
            DL2=DX*DX
            ISTOP=14
            IF(HTTOT.EQ.0.0) GO TO 656
            DO 61 LLSE=1,ISTOP
            READ 2,WS,TSU,TDEW,RADN,ISGNX,ISGNY
            READ 2,ANGLE
            PRINT 161,WS,TSU,TDEW,RADN,ANGLE,HTTOT,ISGNX,ISGNY
      161   FORMAT(1X,6(',',F12.6),2(',',I5))
      61    CONTINUE
      656   CONTINUE
            DO 6 LL=1,LLN
            READ 2,WS,TSU,TDEW,RADN,ISGNX,ISGNY
```

```
      READ 2,ANGLE
      PRINT 161,WS,TSU,TOEW,RADN,ANGLE,HTTOT,ISGNX,ISGNY
222   TSU=T(IN-3,JN-5,1)
      TSU=TREF*(1.+TSU)
      TTOT=TTOT+DT
      TTOT1=TTOT1+DT
      TM=(TSU+TOEW)/2.
COMMENT : THE NEXT 6 LINES ARE USED TO CALCULATE THE
C       : EQUILIBRIUM TEMPERATURE.
C
      FW=9.2+0.46*WS**2
      BETAS=0.35+0.015*TM+0.0012*..)**2
      AISS=4.5+0.05*TSU+BETAS*FW+0.047*FW
      TAMB=TOE.+RADN/KISS
      TE=(TAMB-TREF)/TREF
C
COMMENT : THE SURFACE HEAT EXCHANGE COEFFICIENT IS
C       : NON-DIMENSIONLIZED.
C
      AKT=KISS*0.00191
C
COMMENT : ANGLE (DEGREES) = ANGLE X 0.01745329 (RADIANS).
C
      TAUX=(0.154)*SIN(ANGLE*0.01745329)*ISGNX
      TAUY=(0.154)*COS(ANGLE*0.01745329)*ISGNY
      PRINT 161,TSU,TM,FW,BETAS,TAMB,KISS,ISTOP,LL
      CALL ERROR(IWN,JWN,IW,JW,DT,WH,WHLDT,KN,MRH)
      CALL WHTOP(IW,JW,IWN,JWN,KN,WH,K,MRM)
      CALL WHATIJ(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
      CALL INTE(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,MAR,XINT,YINT,A3,A1,
     CAH,AV,TAUX,TAUY,DX,DY,DZ,D,E,DT,DPSX,DPSY,AP)
      CALL CORINT(I,J,K,IN,JN,KN,ABR,U,V,XINT,YINT,DZ,HI,MAR)
      CALL ROINTX(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,
     CMAR,RINTX,HX,XINT)
      CALL ROINTY(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,MAR,
     CRINTY,HY,YINT)
      CALL DPSXY(I,J,IN,JN,IW,JW,IWN,JWN,DPSX,DPSY,P,DX,DY,MAR)
      CALL FORCE(I,J,IW,JW,XINT,YINT,WHLDT,DX,DY,HI,HX,HY,MRH,
     CDPSX,DPSY,FH,AP,IN,JN,IWN,JWN,RINTX,RINTY,U,V,EUL,ABR,MAR,KN)
      CALL PREIL(EPS,MAXIT,IN,JN,P,ITN,DPSX,DPSY,FH,DL2,OMEGA,
     CMRH,I,J,K,IW,JW,DX,DY,CX,IWN,JWN,ARBP)
      CALL UVT(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,D,E,H,G,DX,DY,DZ,
     CRINTX,RINTY,EUL,W,DT,A1,AP,AH,AV,A3,HI,HX,HY,P,MAR)
      CALL UVTOP(H,G,TAUX,TAUY,I,J,K,DZ,IN,JN,KN,HI,MAR)
      CALL OLDUV(I,J,K,IN,JN,KN,U,V,D,E)
      CALL OLDUV(I,J,K,IN,JN,KN,H,G,U,V)
      CALL RWH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,DX,DY,DZ,MRH)
      CALL WHATIJ(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
15    CONTINUE
      DO 20 I=1,IN
      DO 20 J=1,JN
      WD(I,J,1)=W(I,J,1)
20    CONTINUE
      DO 30 I=1,IN
      DO 30 J=1,JN
      W(I,J,1)=0.0
30    CONTINUE
35    CONTINUE
      CALL TEM1(I,J,K,IN,JN,KN,U,V,T,TD,DX,
     CCB,
     CDY,DZ,W,DT,TAI,TAH,TAV,B3,HI,HX,HY,MAR,AKT,TREF,TAMB)
      CALL TEM2(I,J,K,IN,JN,KN,TD,DX,DY,DZ,MAR,CB,HI,AKT,CW,TAMB,
     CHX,HY,T,TREF,TAV,TAI,TAH,B3,DT)
      CALL OLDT(I,J,K,IN,JN,KN,T,TP)
      CALL OLDT(I,J,K,IN,JN,KN,TD,T)
      CALL TEQB(I,J,K,IN,JN,KN,T,MAR)
      CALL DENSTY(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,A,B,C,MAR,MRH,Y,YW,
     CRO,ROW,RREF,TREF)
      DO 2000 I=9,11
      K=1
C     PRINT 9200,IL,I,K,(V(I,J,K),J=1,JNJ)
2000  CONTINUE
C9200  FORMAT(/' L=',I3,3X,'I=',I3,3X,'K=',I3/' V-VELOCITY'/(5X,8E15.7))
      DO 40 I=1,IN
      DO 40 J=1,JN
      W(I,J,1)=WD(I,J,1)
40    CONTINUE
      CALL INLET1(I,J,K,IN,JN,KN,U,V,H,G,T,TD,AA,TLL,DT,HTTOT)
      TIME=TTOT1*CTTOT
```

```
      PRTT=1.
      IF(TIME.GT.[illegible]T) GO TO 444
      GO TO 222
  444  TTOT1=0.0
      CALL RWRM(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,HX,HY,
     CDX,DY,DZ,WRM,WHH)
      CALL RWR(I,J,K,IN,JN,KN,U,V,W,WR,HI,HX,HY,DZ,MAR)
      IF(KSTORE.GT.0) GO TO 1000
      CALL STORE2(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,
     CHI,MAR,MRW,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,
     CTAV,AKT,CB,CW,A,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,TAMB,TTOT)
 1000 CONTINUE
      HTTOT=CTTOT+TTOT
      PRINT 92,HTTOT
      PRINT 93,TAUX,TAUY
   93 FORMAT(1X,'TAUX=',F11.6,4X,'TAUY=',F11.6)
   92 FORMAT(1X,'TOTAL TIME THUS FAR =',F5.1,'HRS',/)
      CALL TPRIN1(TAI,TAH,TAV,CB,CW,AKT,TREF,RREF,EUL,A,B,C,TE,TU)
      CALL PRITE(ITR,EX)
      CALL PRUV(I,J,K,IN,JN,KN,U,V,UA,VA,MAR)
      CALL TPRINK(I,J,K,IN,JN,KN,T,RO,TREF,MAR,TACTUL)
      ISTOP=ISTOP+1
    6 CONTINUE
      END FILE 8
      END
```

```
ASA*NASA(1).CORINT FOR CREATED ON 5 MAY 80 AT 10:48:36
      C**************************************************************
      C  THIS SUBROUTINE ADDS INTEGRAL OF CORIOLIS COMPONENT TO XINT  *
      C  & YINT.                                                      *
      C**************************************************************
            SUBROUTINE CORINT(I,J,K,IN,JN,KN,ABR,U,V,XINT,YINT,DZ,HI,MAR)
            DIMENSION U(IN,JN,KN),V(IN,JN,KN),XINT(IN,JN),YINT(IN,JN),HI(IN,
           CJN),MAR(IN,JN)
            DO 10 I=1,IN
            DO 10 J=1,JN
            IF (MAR(I,J).LT.11) GO TO 9
            DO 8 K=2,KN
            XINT(I,J)=XINT(I,J)-ABR*HI(I,J)*(V(I,J,K-1)+V(I,J,K))*DZ/2
            YINT(I,J)=YINT(I,J)+ABR*HI(I,J)*(U(I,J,K-1)+U(I,J,K))*DZ/2
      8     CONTINUE
      9     CONTINUE
      10    CONTINUE
            RETURN
            END
```

```
NASA*NASA(1).DENSTY FOR CREATED ON 15 MAY 74 AT 11:36:18
C*************************************************************************
C          THE FOLLOWING PROGRAM CALCULATES THE DENSITY FIELD FROM
C               THE TEMPERATURE FIELD
C*************************************************************************
      SUBROUTINE DENSTY(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,A,B,C,
     CMAR,MRW,
     CT,TW,RO,ROW,RREF,TREF)
      DIMENSION RO(IN,JN,KN),T(IN,JN,KN)
      DIMENSION ROW(IWN,JWN,KN),TW(IWN,JWN,KN)
      DIMENSION MAR(IN,JN),MRW(IWN,JWN)
      DO 10 I=1,IN
      DO 10 J=1,JN
      IF (MAR(I,J).EQ.0) GO TO 12
      DO 11 K=1,KN
      TEM=T(I,J,K)*TREF+TREF
      R=A+B*TEM+C*TEM*TEM
      RO(I,J,K)=(R-RREF)/RREF
   11 CONTINUE
   12 CONTINUE
   10 CONTINUE
      DO 20 IW=1,IWN
      DO 20 JW=1,JWN
      IF (MRW(IW,JW).EQ.0) GO TO 22
      DO 21 K=1,KN
      TEMW=TW(IW,JW,K)*TREF+TREF
      RW=A+B*TEMW+C*TEMW*TEMW
      ROW(IW,JW,K)=(RW-RREF)/RREF
   21 CONTINUE
   22 CONTINUE
   20 CONTINUE
      RETURN
      END
```

```
ASA*NASA(1).DINERU FOR CREATED ON 5 MAY 80 AT 11:00:12
      C***************************************************************
      C   THIS SUBROUTINE COMPUTES UIHUUX,DIHVVY WHICH ARE USED IN     *
      C   THE POISSON EQUATION FOR PRESSURE.                           *
      C***************************************************************
      C
      C
            SUBROUTINE DINERU(I,J,K,IN,JN,KN,U,V,HI,DX,DY,DIHUUX,DIHUVX,MAR)
            DIMENSION U(IN,JN,KN),V(IN,JN,KN),HI(IN,JN),MAR(IN,JN)
            IF(MAR(I,J).EQ.0) GO TO 50
            IF(MAR(I,J).EQ.1) GO TO 31
            IF(MAR(I,J).EQ.2) GO TO 32
            IF(MAR(I,J).EQ.3) GO TO 33
            IF(MAR(I,J).EQ.4) GO TO 34
            IF(MAR(I,J).EQ.5) GO TO 35
            IF(MAR(I,J).EQ.6) GO TO 36
            IF(MAR(I,J).EQ.7) GO TO 37
            IF(MAR(I,J).EQ.8) GO TO 38
            IF(MAR(I,J).EQ.9) GO TO 39
            IF(MAR(I,J).EQ.10)GO TO 40
            DIHUUX=(U(I+1,J,K)*U(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
           C*U(I-1,J,K)*HI(I-1,J))/(2*DX)
            DIHUVX=(U(I+1,J,K)*V(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
           C*V(I-1,J,K)*HI(I-1,J))/(2*DX)
            GO TO 50
      31    CONTINUE
            DIHUUX=(U(I+1,J,K)*U(I+1,J,K)*HI(I+1,U)-U(I-1,J,K)
           C*U(I-1,J,K)*HI(I-1,J))/(2*DX)
            DIHUVX=(J(I+1,J,K)*V(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
           C*V(I-1,J,K)*HI(I-1,J))/(2*DX)
            GO TO 50
      32    CONTINUE
            DIHUUX=(U(I+1,J,K)*U(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
           C*U(I-1,J,K)*HI(I-1,J))/(2*DX)
            DIHUVX=(J(I+1,J,K)*V(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
           C*V(I-1,J,K)*HI(I-1,J))/(2*DX)
            GO TO 50
      33    CONTINUE
            DIHUUX=(4*HI(I+1,J)*U(I+1,J,K)*U(I+1,J,K)-3*HI(I,J)*U(I,J,K)
           C*U(I,J,K)-HI(I+2,J)*U(I+2,J,K)*U(I+2,J,K))/(2*DX)
            DIHUVX=(4*HI(I+1,J)*U(I+1,J,K)*V(I+1,J,K)-3*HI(I,J)*U(I,J,K)
           C*V(I,J,K)-HI(I+2,J)*U(I+2,J,K)*V(I+2,J,K))/(2*DX)
            GO TO 50
      34    CONTINUE
            DIHUUX=(3*HI(I,J)*U(I,J,K)*U(I,J,K)-4*HI(I-1,J)*U(I-1,J,K)
           C*U(I-1,J,K)+HI(I-2,J)*U(I-2,J,K)*U(I-2,J,K))/(2*DX)
            DIHUVX=(3*HI(I,J)*U(I,J,K)*V(I,J,K)-4*HI(I-1,J)*U(I-1,J,K)
           C*V(I-1,J,K)+HI(I-2,J)*U(I-2,J,K)*V(I-2,J,K))/(2*DX)
            GO TO 50
      35    CONTINUE
            DIHUUX=(4*HI(I+1,J)*U(I+1,J,K)*U(I+1,J,K)-3*HI(I,J)*U(I,J,K)
           C*U(I,J,K)-HI(I+2,J)*U(I+2,J,K)*U(I+2,J,K))/(2*DX)
            DIHUVX=(4*HI(I+1,J)*U(I+1,J,K)*V(I+1,J,K)-3*HI(I,J)*U(I,J,K)
           C*V(I,J,K)-HI(I+2,J)*U(I+2,J,K)*V(I+2,J,K))/(2*DX)
            GO TO 50
      36    CONTINJE
            DIHUUX=(U(I+1,J,K)*U(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
           C*U(I-1,J,K)*HI(I-1,J))/(2*DX)
            DIHUVX=(U(I+1,J,K)*V(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
           C*V(I-1,J,K)*HI(I-1,J))/(2*DX)
            GO TO 50
      37    CONTINUE
            DIHUUX=(4*HI(I+1,J)*U(I+1,J,K)*U(I+1,J,K)-3*HI(I,J)*U(I,J,K)
           C*U(I,J,K)-HI(I+2,J)*U(I+2,J,K)*U(I+2,J,K))/(2*DX)
            DIHUVX=(4*HI(I+1,J)*U(I+1,J,K)*V(I+1,J,K)-3*HI(I,J)*U(I,J,K)
           C*V(I,J,K)-HI(I+2,J)*U(I+2,J,K)*V(I+2,J,K))/(2*DX)
            GO TO 50
      38    CONTINUE
            CIHUUX=(U(I+1,J,K)*U(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
           C*U(I-1,J,K)*HI(I-1,J))/(2*DX)
            DIHUVX=(U(I+1,J,K)*V(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
           C*V(I-1,J,K)*HI(I-1,J))/(2*DX)
            GO TO 50
      39    CONTINUE
            DIHUUX=(3*HI(I,J)*U(I,J,K)*U(I,J,K)-4*HI(I-1,J)*U(I-1,J,K)
           C*U(I-1,J,K)+HI(I-2,J)*U(I-2,J,K)*U(I-2,J,K))/(2*DX)
            DIHUVX=(3*HI(I,J)*U(I,U,K)*V(I,J,K)-4*HI(I-1,J)*U(I-1,J,K)
           C*V(I-1,J,K)+HI(I-2,J)*U(I-2,J,K)*V(I-2,J,K))/(2*DX)
            GO TO 50
```

```
   40 CONTINUE
      DIHUUX=(3*HI(I,J)*U(I,J,K)*U(I,J,K)-4*HI(I-1,J)*U(I-1,J,K)
     C*U(I-1,J,K)+HI(I-2,J)*U(I-2,J,K)*U(I-2,J,K))/(2*DX)
      DIHUVX=(3*HI(I,J)*U(I,J,K)*V(I,J,K)-4*HI(I-1,J)*U(I-1,J,K)
     C*V(I-1,J,K)+HI(I-2,J)*U(I-2,J,K)*V(I-2,J,K))/(2*DX)
   50 CONTINUE
      RETURN
      END
```

ASA*NASA(1).DPSXY FOR CREATED ON 5 MAY 80 AT 11:05:25

```
C*************************************************************
C   THIS SUBROUTINE CALCULATES DPSX AND DPSY USED IN COMP-   *
C   UTING THE R.H.S OF POISSON'S EQUATION AT HALF GRID PO-   *
C   INTS                                                      *
C*************************************************************
C
      SUBROUTINE DPSXY(I,J,IN,JN,IW,JW,IWN,JWN,DPSX,DPSY,P,DX,DY,MAR)
      DIMENSION P(IWN,JWN),DPSX(IN,JN),DPSY(IN,JN),MAR(IN,JN)
      DO 10 I=1,IN
      DO 10 J=1,JN
      IW=I
      JW=J
      IF (MAR(I,J).LT.11) GO TO 9
      DPSX(I,J)=(P(IW,JW)-P(IW-1,JW)+P(IW,JW-1)-P(IW-1,JW-1))/(2*DX)
      DPSY(I,J)=(P(IW,JW)-P(IW,JW-1)+P(IW-1,JW)-P(IW-1,JW-1))/(2*DY)
    9 CONTINUE
   10 CONTINUE
      RETURN
      END
```

```
ASA*NASA(1).DUVY FOR CREATED ON 5 MAY 80 AT 11:08:08
C*******************************************************
C   THIS SUBROUTINE COMPUTES DIHUVY USED BY SUB. INTE   *
C*******************************************************
      SUBROUTINE DUVY(I,J,K,IN,JN,KN,U,V,HI,DY,DIHUVY,MAR)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),HI(IN,JN),MAR(IN,JN)
      IF(MAR(I,J).EQ.0) GO TO 50
      IF(MAR(I,J).EQ.1) GO TO 31
      IF(MAR(I,J).EQ.2) GO TO 32
      IF(MAR(I,J).EQ.3) GO TO 33
      IF(MAR(I,J).EQ.4) GO TO 34
      IF(MAR(I,J).EQ.5) GO TO 35
      IF(MAR(I,J).EQ.6) GO TO 36
      IF(MAR(I,J).EQ.7) GO TO 37
      IF(MAR(I,J).EQ.8) GO TO 38
      IF(MAR(I,J).EQ.9) GO TO 39
      IF(MAR(I,J).EQ.10)GO TO 40
      DIHUVY=(U(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-U(I,J-1,K)
     C*V(I,J-1,K)*HI(I,J-1))/(2*DY)
      GO TO 50
   31 CONTINUE
      DIHUVY=(3*HI(I,J)*U(I,J,K)*V(I,J,K)-4*HI(I,J-1)*U(I,J-1,K)
     C*V(I,J-1,K)+HI(I,J-2)*U(I,J-2,K)*V(I,J-2,K))/(2*DY)
      GO TO 50
   32 CONTINUE
      DIHUVY=(4*HI(I,J+1)*U(I,J+1,K)*V(I,J+1,K)-3*HI(I,J)*
     CU(I,J,K)*V(I,J,K)-HI(I,J+2)*U(I,J+2,K)*V(I,J+2,K))/(2*DY)
      GO TO 50
   33 CONTINUE
      DIHUVY=(U(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-U(I,J-1,K)
     C*V(I,J-1,K)*HI(I,J-1))/(2*DY)
      GO TO 50
   34 CONTINUE
      DIHUVY=(U(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-U(I,J-1,K)
     C*V(I,J-1,K)*HI(I,J-1))/(2*DY)
      GO TO 50
   35 CONTINUE
      DIHUVY=(3*HI(I,J)*U(I,J,K)*V(I,J,K)-4*HI(I,J-1)*U(I,J-1,K)
     C*V(I,J-1,K)+HI(I,J-2)*U(I,J-2,K)*V(I,J-2,K))/(2*DY)
      GO TO 50
   36 CONTINUE
      DIHUVY=(U(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-U(I,J-1,K)
     C*V(I,J-1,K)*HI(I,J-1))/(2*DY)
      GO TO 50
   37 CONTINUE
      DIHUVY=(4*HI(I,J+1)*U(I,J+1,K)*V(I,J+1,K)-3*HI(I,J)*
     CU(I,J,K)*V(I,J,K)-HI(I,J+2)*U(I,J+2,K)*V(I,J+2,K))/(2*DY)
      GO TO 50
   38 CONTINUE
      DIHUVY=(U(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-U(I,J-1,K)
     C*V(I,J-1,K)*HI(I,J-1))/(2*DY)
      GO TO 50
   39 CONTINUE
      DIHUVY=(4*HI(I,J+1)*U(I,J+1,K)*V(I,J+1,K)-3*HI(I,J)*
     CU(I,J,K)*V(I,J,K)-HI(I,J+2)*U(I,J+2,K)*V(I,J+2,K))/(2*DY)
      GO TO 50
   40 CONTINUE
      DIHUVY=(3*HI(I,J)*U(I,J,K)*V(I,J,K)-4*HI(I,J-1)*U(I,J-1,K)
     C*V(I,J-1,K)+HI(I,J-2)*U(I,J-2,K)*V(I,J-2,K))/(2*DY)
   50 CONTINUE
      RETURN
      END
```

```
ASA 'NASA(1).DVISU FOR CREATED ON 5 MAY 80 AT 11:10:49
      C*****************************************************************
      C   THIS SUBROUTINE COMPUTES D1UX,D2UX & D1UY                     *
      C*****************************************************************
      C
            SUBROUTINE DVISU(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1UX,D2UX,D1UY,D2U
           CMAR)
            DIMENSION U(IN,JN,KN),V(IN,JN,KN),HI(IN,JN),MAR(IN,JN)
            IF(MAR(I,J).EQ.0) GO TO 50
            IF(MAR(I,J).EQ.1) GO TO 31
            IF(MAR(I,J).EQ.2) GO TO 32
            IF(MAR(I,J).EQ.3) GO TO 33
            IF(MAR(I,J).EQ.4) GO TO 34
            IF(MAR(I,J).EQ.5) GO TO 35
            IF(MAR(I,J).EQ.6) GO TO 36
            IF(MAR(I,J).EQ.7) GO TO 37
            IF(MAR(I,J).EQ.8) GO TO 38
            IF(MAR(I,J).EQ.9) GO TO 39
            IF(MAR(I,J).EQ.10)GO TO 40
            D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
            D1UY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
            D2UX=(U(I+1,J,K)-2*U(I,J,K)+U(I-1,J,K))/(DX*DX)
            D2UY=(U(I,J+1,K)-2*U(I,J,K)+U(I,J-1,K))/(DY*DY)
            GO TO 50
   31       CONTINUE
            D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
            D2UX=(U(I+1,J,K)-2*U(I,J,K)+U(I-1,J,K))/(DX*DX)
            D1UY=(3*U(I,J,K)+U(I,J-2,K)-4*U(I,J-1,K))/(2*DY)
            D2UY=(U(I,J,K)+U(I,J-2,K)-2*U(I,J-1,K))/(DY*DY)
            GO TO 50
   32       CONTINUE
            D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
            D2UX=(U(I+1,J,K)-2*U(I,J,K)+U(I-1,J,K))/(DX*DX)
            D1UY=(4*U(I,J+1,K)-3*U(I,J,K)-U(I,J+2,K))/(2*DY)
            D2UY=(U(I,J+2,K)+U(I,J,K)-2*U(I,J+1,K))/(DY*DY)
            GO TO 50
   33       CONTINUE
            D1UY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
            D2UY=(U(I,J+1,K)-2*U(I,J,K)+U(I,J-1,K))/(DY*DY)
            D1UX=(4*U(I+1,J,K)-3*U(I,J,K)-U(I+2,J,K))/(2*DX)
            D2UX=(U(I+2,J,K)-2*U(I+1,J,K)+U(I,J,K))/(DX*DX)
            GO TO 50
   34       CONTINUE
            D1UY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
            D2UY=(U(I,J+1,K)-2*U(I,J,K)+U(I,J-1,K))/(DY*DY)
            D1UX=(3*U(I,J,K)-4*U(I-1,J,K)+U(I-2,J,K))/(2*DX)
            D2UX=(U(I,J,K)-2*U(I-1,J,K)+U(I-2,J,K))/(DX*DX)
            GO TO 50
   35       CONTINUE
            D1UY=(3*U(I,J,K)+U(I,J-2,K)-4*U(I,J-1,K))/(2*DY)
            D2UY=(U(I,J,K)+U(I,J-2,K)-2*U(I,J-1,K))/(DY*DY)
            D1UX=(4*U(I+1,J,K)-3*U(I,J,K)-U(I+2,J,K))/(2*DX)
            D2UX=(U(I+2,J,K)-2*U(I+1,J,K)+U(I,J,K))/(DX*DX)
            GO TO 50
   36       CONTINUE
            D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
            D1UY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
            D2UX=(U(I+1,J,K)-2*U(I,J,K)+U(I-1,J,K))/(DX*DX)
            D2UY=(U(I,J+1,K)-2*U(I,J,K)+U(I,J-1,K))/(DY*DY)
            GO TO 50
   37       CONTINUE
            D1UY=(4*U(I,J+1,K)-3*U(I,J,K)-U(I,J+2,K))/(2*DY)
            D2UY=(U(I,J+2,K)+U(I,J,K)-2*U(I,J+1,K))/(DY*DY)
            D1UX=(4*U(I+1,J,K)-3*U(I,J,K)-U(I+2,J,K))/(2*DX)
            D2UX=(U(I+2,J,K)-2*U(I+1,J,K)+U(I,J,K))/(DX*DX)
            GO TO 50
   38       CONTINUE
            D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
            D1UY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
            D2UX=(U(I+1,J,K)-2*U(I,J,K)+U(I-1,J,K))/(DX*DX)
            D2UY=(U(I,J+1,K)-2*U(I,J,K)+U(I,J-1,K))/(DY*DY)
            GO TO 50
   39       CONTINUE
            D1UY=(4*U(I,J+1,K)-3*U(I,J,K)-U(I,J+2,K))/(2*DY)
            D2UY=(U(I,J+2,K)+U(I,J,K)-2*U(I,J+1,K))/(DY*DY)
            D1UX=(3*U(I,J,K)-4*U(I-1,J,K)+U(I-2,J,K))/(2*DX)
            D2UX=(U(I,J,K)-2*U(I-1,J,K)+U(I-2,J,K))/(DX*DX)
            GO TO 50
   40       CONTINUE
```

```
      D1UY=(3*U(I,J,K)+U(I,J-2,K)-4*U(I,J-1,K))/(2*DY)
      D2UY=(U(I,J,K)+U(I,J-2,K)-2*U(I,J-1,K))/(DY*DY)
      D1UX=(3*U(I,J,K)-4*U(I-1,J,K)+U(I-2,J,K))/(2*DX)
      D2UX=(U(I,J,K)-2*U(I-1,J,K)+U(I-2,J,K))/(DX*DX)
   50 CONTINUE
      RETURN
      END
```

```
RSA-NASA(1).DVISV FOR CREATED ON 5 MAY 80 AT 11:13:50
C ************************************************************
C   THIS SUBROUTINE COMPUTES D1VY,D2VY,D1VX & D2VX             *
C ************************************************************
C
      SUBROUTINE DVISV(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1VX,D2VX,D1VY,D2VY
     CMAR)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),HI(IN,JN),MAR(IN,JN)
      IF(MAR(I,J).EQ.0) GO TO 50
      IF(MAR(I,J).EQ.1) GO TO 31
      IF(MAR(I,J).EQ.2) GO TO 32
      IF(MAR(I,J).EQ.3) GO TO 33
      IF(MAR(I,J).EQ.4) GO TO 34
      IF(MAR(I,J).EQ.5) GO TO 35
      IF(MAR(I,J).EQ.6) GO TO 36
      IF(MAR(I,J).EQ.7) GO TO 37
      IF(MAR(I,J).EQ.8) GO TO 38
      IF(MAR(I,J).EQ.9) GO TO 39
      IF(MAR(I,J).EQ.10)GO TO 40
      D1VX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
      D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
      D2VX=(V(I+1,J,K)-2*V(I,J,K)+V(I-1,J,K))/(DX*DX)
      D2VY=(V(I,J+1,K)-2*V(I,J,K)+V(I,J-1,K))/(DY*DY)
      GO TO 50
 31   CONTINUE
      D1VX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
      D2VX=(V(I+1,J,K)-2*V(I,J,K)+V(I-1,J,K))/(DX*DX)
      D1VY=(3*V(I,J,K)-4*V(I,J-1,K)+V(I,J-2,K))/(2*DY)
      D2VY=(V(I,J,K)+V(I,J-2,K)-2*V(I,J-1,K))/(DY*DY)
      GO TO 50
 32   CONTINUE
      D1VX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
      D2VX=(V(I+1,J,K)-2*V(I,J,K)+V(I-1,J,K))/(DX*DX)
      D1VY=(4*V(I,J+1,K)-3*V(I,J,K)-V(I,J+2,K))/(2*DY)
      D2VY=(V(I,J+2,K)+V(I,J,K)-2*V(I,J+1,K))/(DY*DY)
      GO TO 50
 33   CONTINUE
      D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
      D2VY=(V(I,J+1,K)-2*V(I,J,K)+V(I,J-1,K))/(DY*DY)
      D1VX=(4*V(I+1,J,K)-3*V(I,J,K)-V(I+2,J,K))/(2*DX)
      D2VX=(V(I+2,J,K)-2*V(I+1,J,K)+V(I,J,K))/(DX*DX)
      GO TO 50
 34   CONTINUE
      D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
      D2VY=(V(I,J+1,K)-2*V(I,J,K)+V(I,J-1,K))/(DY*DY)
      D1VX=(3*V(I,J,K)-4*V(I-1,J,K)+V(I-2,J,K))/(2*DX)
      D2VX=(V(I,J,K)-2*V(I-1,J,K)+V(I-2,J,K))/(DX*DX)
      GO TO 50
 35   CONTINUE
      D1VY=(3*V(I,J,K)-4*V(I,J-1,K)+V(I,J-2,K))/(2*DY)
      D2VY=(V(I,J,K)+V(I,J-2,K)-2*V(I,J-1,K))/(DY*DY)
      D1VX=(4*V(I+1,J,K)-3*V(I,J,K)-V(I+2,J,K))/(2*DX)
      D2VX=(V(I+2,J,K)-2*V(I+1,J,K)+V(I,J,K))/(DX*DX)
      GO TO 50
 36   CONTINUE
      D1VX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
      D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
      D2VX=(V(I+1,J,K)-2*V(I,J,K)+V(I-1,J,K))/(DX*DX)
      D2VY=(V(I,J+1,K)-2*V(I,J,K)+V(I,J-1,K))/(DY*DY)
      GO TO 50
 37   CONTINUE
      D1VY=(4*V(I,J+1,K)-3*V(I,J,K)-V(I,J+2,K))/(2*DY)
      D2VY=(V(I,J+2,K)+V(I,J,K)-2*V(I,J+1,K))/(DY*DY)
      D1VX=(4*V(I+1,J,K)-3*V(I,J,K)-V(I+2,J,K))/(2*DX)
      D2VX=(V(I+2,J,K)-2*V(I+1,J,K)+V(I,J,K))/(DX*DX)
      GO TO 50
 38   CONTINUE
      D1VX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
      D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
      D2VX=(V(I+1,J,K)-2*V(I,J,K)+V(I-1,J,K))/(DX*DX)
      D2VY=(V(I,J+1,K)-2*V(I,J,K)+V(I,J-1,K))/(DY*DY)
      GO TO 50
 39   CONTINUE
      D1VY=(4*V(I,J+1,K)-3*V(I,J,K)-V(I,J+2,K))/(2*DY)
      D2VY=(V(I,J+2,K)+V(I,J,K)-2*V(I,J+1,K))/(DY*DY)
      D1VX=(3*V(I,J,K)-4*V(I-1,J,K)+V(I-2,J,K))/(2*DX)
      D2VX=(V(I,J,K)-2*V(I-1,J,K)+V(I-2,J,K))/(DX*DX)
      GO TO 50
 40   CONTINUE
```

```
      DIVY=(3*V(I,J,K)-4*V(I,J-1,K)+V(I,J-2,K))/(2*DY)
      D2VY=(V(I,J,K)+V(I,J-2,K)-2*V(I,J-1,K))/(DY*DY)
      DIVX=(3*V(I,J,K)-4*V(I-1,J,K)+V(I-2,J,K))/(2*DX)
      D2VX=(V(I,J,K)-2*V(I-1,J,K)+V(I-2,J,K))/(DX*DX)
   50 CONTINUE
      RETURN
      END
```

```
NASA(1).DVVY FOR C[illegible]D ON 5 MAY 80 AT 11:16:10
      C*****************************************************
      C   THIS SUBROU[illegible] COMPUTES DIHVVY               *
      C*****************************************************
      C
            SUBROUTINE DVVY(I,J,K,IN,JN,KN,U,V,HI,DY,DIHVVY,MAR)
            DIMENSION U(IN,JN,KN),V(IN,JN,KN),HI(IN,JN),MAR(IN,JN)
            IF(MAR(I,J).EQ.0) GO TO 50
            IF(MAR(I,J).EQ.1) GO TO 31
            IF(MAR(I,J).EQ.2) GO TO 32
            IF(MAR(I,J).EQ.3) GO TO 33
            IF(MAR(I,J).EQ.4) GO TO 34
            IF(MAR(I,J).EQ.5) GO TO 35
            IF(MAR(I,J).EQ.6) GO TO 36
            IF(MAR(I,J).EQ.7) GO TO 37
            IF(MAR(I,J).EQ.8) GO TO 38
            IF(MAR(I,J).EQ.9) GO TO 39
            IF(MAR(I,J).EQ.10)GO TO 40
            DIHVVY=(V(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-V(I,J-1,K)*
           CV(I,J-1,K)*HI(I,J-1))/(2*DY)
            GO TO 50
   31       CONTINUE
            DIHVVY=(3*HI(I,J)*V(I,J,K)*V(I,J,K)+HI(I,J-2)*V(I,J-2,K)
           C*V(I,J-2,K)-4*HI(I,J-1)*V(I,J-1,K)*V(I,J-1,K))/(2*DY)
            GO TO 50
   32       CONTINUE
            DIHVVY=(4*HI(I,J+1)*V(I,J+1,K)*V(I,J+1,K)-3*HI(I,J)*V(I,J,K)
           C*V(I,J,K)-HI(I,J+2)*V(I,J+2,K)*V(I,J+2,K))/(2*DY)
            GO TO 50
   33       CONTINUE
            DIHVVY=(V(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-V(I,J-1,K)*
           CV(I,J-1,K)*HI(I,J-1))/(2*DY)
            GO TO 50
   34       CONTINUE
            DIHVVY=(V(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-V(I,J-1,K)*
           CV(I,J-1,K)*HI(I,J-1))/(2*DY)
            GO TO 50
   35       CONTINUE
            DIHVVY=(3*HI(I,J)*V(I,J,K)*V(I,J,K)+HI(I,J-2)*V(I,J-2,K)
           C*V(I,J-2,K)-4*HI(I,J-1)*V(I,J-1,K)*V(I,J-1,K))/(2*DY)
            GO TO 50
   36       CONTINUE
            DIHVVY=(V(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-V(I,J-1,K)*
           CV(I,J-1,K)*HI(I,J-1))/(2*DY)
            GO TO 50
   37       CONTINUE
            DIHVVY=(4*HI(I,J+1)*V(I,J+1,K)*V(I,J+1,K)-3*HI(I,J)*V(I,J,K)
           C*V(I,J,K)-HI(I,J+2)*V(I,J+2,K)*V(I,J+2,K))/(2*DY)
            GO TO 50
   38       CONTINUE
            DIHVVY=(V(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-V(I,J-1,K)*
           CV(I,J-1,K)*HI(I,J-1))/(2*DY)
            GO TO 50
   39       CONTINUE
            DIHVVY=(4*HI(I,J+1)*V(I,J+1,K)*V(I,J+1,K)-3*HI(I,J)*V(I,J,K)
           C*V(I,J,K)-HI(I,J+2)*V(I,J+2,K)*V(I,J+2,K))/(2*DY)
            GO TO 50
   40       CONTINUE
            DIHVVY=(3*HI(I,J)*V(I,J,K)*V(I,J,K)+HI(I,J-2)*V(I,J-2,K)
           C*V(I,J-2,K)-4*HI(I,J-1)*V(I,J-1,K)*V(I,J-1,K))/(2*DY)
   50       CONTINUE
            RETURN
            END
```

```
ASA*NASA(1).DIZZ SYM CREATED ON 15 AUG 79 AT 10:04:10
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C          THIS PROGRAM CALCULATES THE Z DERIVATIVES
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      SUBROUTINE DIZZ(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,DX,DY,DZ,D1UWZ,
     CA3,TAUX,TAUY,D1VWZ,D1UZ,D2UZ,D1VZ,D2VZ,D1A3Z)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),HI(IN,JN),
     CHX(IN,JN),HY(IN,JN)
      DIMENSION A3(KN)
      IF(K.EQ.1)GO TO 61
      IF(K.EQ.KN)GO TO 62
      D1UZ=(U(I,J,K+1)-U(I,J,K-1))/(2*DZ)
      D1VZ=(V(I,J,K+1)-V(I,J,K-1))/(2*DZ)
      D2UZ=(U(I,J,K+1)-2.*U(I,J,K)+U(I,J,K-1))/(DZ**2)
      D2VZ=(V(I,J,K+1)-2.*V(I,J,K)+V(I,J,K-1))/(DZ**2)
      D1A3Z=(A3(K+1)-A3(K-1))/(2.*DZ)
      D1UWZ=(U(I,J,K+1)*W(I,J,K+1)-U(I,J,K-1)*W(I,J,K-1))/(2.*DZ)
      D1VWZ=(V(I,J,K+1)*W(I,J,K+1)-V(I,J,K-1)*W(I,J,K-1))/(2.*DZ)
      GO TO 63
61    CONTINUE
      D1UZ=HI(I,J)*TAUX
      D1VZ=HI(I,J)*TAUY
      D2UZ=2.*(U(I,J,K+1)-U(I,J,K))/(DZ**2)-2.*(TAUX*HI(I,J)/DZ)
      D2VZ=2.*(V(I,J,K+1)-V(I,J,K))/(DZ**2)-2.*(TAUY*HI(I,J)/DZ)
      D1A3Z=(4.*A3(K+1)-3.*A3(K)-A3(K+2))/(2.*DZ)
      D1UWZ=(4.*U(I,J,K+1)*W(I,J,K+1)-3.*U(I,J,K)*W(I,J,K)-U(I,J,K+2)*W
     CI,J,K+2))/(2.*DZ)
      D1VWZ=(4.*V(I,J,K+1)*W(I,J,K+1)-3.*V(I,J,K)*W(I,J,K)-V(I,J,K+2)*W
     CI,J,K+2))/(2.*DZ)
      GO TO 63
62    CONTINUE
      D1UZ=(3.*U(I,J,K)-4.*U(I,J,K-1)+U(I,J,K-2))/(2.*DZ)
      D1VZ=(3.*V(I,J,K)-4.*V(I,J,K-1)+V(I,J,K-2))/(2.*DZ)
      D2UZ=(U(I,J,K-2)+U(I,J,K)-2.*U(I,J,K-1))/(DZ**2)
      D2VZ=(V(I,J,K-2)+V(I,J,K)-2.*V(I,J,K-1))/(DZ**2)
      D1A3Z=(3.*A3(K)-4.*A3(K-1)+A3(K-2))/(2*DZ)
      D1UWZ=(3.*U(I,J,K)*W(I,J,K)-4.*U(I,J,K-1)*W(I,J,K-1)
     C+U(I,J,K-2))/(2.*DZ)
      D1VWZ=(3.*V(I,J,K)*W(I,J,K)-4.*V(I,J,K-1)*W(I,J,K-1)
     C+V(I,J,K-2))/(2.*DZ)
63    CONTINUE
      RETURN
      END
```

```
ASA*NASA(1).ERROR FOR CREATED ON 15-MAY-74-AT 16:07:35
      C**********************************************************************
      C      THIS PROGRAM CALCULATES THE HIRT AND HARLOW CORRECTION TERM AT THE
      C                                    SURFACE
      C**********************************************************************
             SUBROUTINE ERROR(IWN,JWN,IW,JW,DT,WH,WHLDT,KN,MRH)
             DIMENSION WHLDT(IWN,JWN),WH(IWN,JWN,KN)
             DIMENSION MRH(IWN,JWN)
      C WHLDT IS THE TIME DERIVATIVE OF W AT HALF GRID POINTS AT LID
             DO 3100 IW=1,IWN
             DO 3100 JW=1,JWN
             IF (MRH(IW,JW).EQ.0) GO TO 3000
             WHLDT(IW,JW)=-WH(IW,JW,1)/DT
       3000  CONTINUE
       3100  CONTINUE
             RETURN
             END
```

```
NASA*NASA(1).FORCE FOR CREATED ON 5 MAY 60 AT 11:18:29
      C*************************************************
      C   THIS SUBROUTINE COMPUTES THE R.H.S OF POISSONS*
      C  EQUATION AT HALF GRID POINTS.                  *
      C*************************************************
      C
            SUBROUTINE FORCE(I,J,IW,JW,XINT,YINT,WHLDT,DX,DY,HI,HX,HY,
           CMRH,
           CDPSX,DPSY,FH,AP,IN,JN,IWN,JWN,RINTX,RINTY,U,V,EUL,ABR,MAR,KN)
            DIMENSION XINT(IN,JN),YINT(IN,JN),WHLDT(IWN,JWN),HI(IN,JN),HX(I
           CJN),HY(IN,JN),DPSX(IN,JN),DPSY(IN,JN),FH(IWN,JWN)
            DIMENSION MRH(IWN,JWN),RINTX(IN,JN,KN),RINTY(IN,JN,KN)
            DIMENSION U(IN,JN,KN),V(IN,JN,KN),MAR(IN,JN)
            K=1
            DO 100 I=1,IN
            DO 100 J=1,JN
            IF(MAR(I,J).LT.11) GO TO 90
            DPSX(I,J)=DPSX(I,J)-EUL*RINTX(I,J,K)+V(I,J,K)*ABR
            DPSY(I,J)=DPSY(I,J)-EUL*RINTY(I,J,K)-ABR*U(I,J,K)
      90    CONTINUE
      100   CONTINUE
            DO 10 IW=1,IWN
            DO 10 JW=1,JWN
            I=IW
            J=JW
            IF (MRH(IW,JW).EQ.0) GO TO 9
            DPSXH=(DPSX(I,J)+DPSX(I+1,J)+DPSX(I,J+1)+DPSX(I+1,J+1))/4.0
            DPSYH=(DPSY(I,J)+DPSY(I+1,J)+DPSY(I,J+1)+DPSY(I+1,J+1))/4.0
            HXH=(HX(I,J)+HX(I+1,J)+HX(I,J+1)+HX(I+1,J+1))/4.0
            HYH=(HY(I,J)+HY(I+1,J)+HY(I,J+1)+HY(I+1,J+1))/4.0
            DXINT=(XINT(I+1,J)+XINT(I+1,J+1)-XINT(I,J)-XINT(I,J+1))/(2*DX)
            DYINT=(YINT(I,J+1)+YINT(I+1,J+1)-YINT(I,J)-YINT(I+1,J))/(2*DY)
            HH=(HI(I,J)+HI(I+1,J)+HI(I,J+1)+HI(I+1,J+1))/4.0
            FH(IW,JW)=(1./AP)*(-(1./HH)*(DXINT+DYINT)-WHLDT(IW,JW)-(AP/HH)*
           C(HXH*DPSXH+HYH*DPSYH))
      9     CONTINUE
      10    CONTINUE
            RETURN
            END
```

ASA-NASA(1).HEIGHT FOR CREATED ON 5 MAY 80 AT 11:21:35

```
C**************************************************************************
C       THIS PROGRAM PUTS CONSTANT DEPTH FOR CC=1.0 IN THE DATA
C******************************************      **********************
C
        SUBROUTINE HEIGHT(I,J,K,IN,JN,KN,HI,HX,HY,CC)
        DIMENSION HI(IN,JN),HX(IN,JN),HY(IN,JN)
        DO 100 I=1,IN
        DO 100 J=1,JN
        HI(I,J)=CC
        HX(I,J)=0.0
        HY(I,J)=0.0
100     CONTINUE
        DO 200 I=1,IN
        PRINT 101,I,(HI(I,J),J=1,JN)
200     CONTINUE
101     FORMAT(/' I=',I3/,' DEPTH'/5X,9E14.7)
        RETURN
        END
```

.ASA*NASA(1).INITIA FOR CREATED ON 14 MAY 74 AT 15:51:03

```
C***********************************************************************
C      THIS PROGRAM INITIALIZES THE VALUES OF U,V,WH,W,D,E,PINTH
C***********************************************************************
      SUBROUTINE INITIA(IN,JN,KN,IWN,JWN,U,V,W,WH,D,E,PINTH,I,J,K,IW,'
     CARBP)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IWN,JWN,KN),
     CD(IN,JN,KN),E(IN,JN,KN),
     CPINTH(IWN,JWN)
C INITIAL CONDITIONS ON U AND V
10    DO 100 I=1,IN
11    DO 100 J=1,JN
      DO 100 K=1,KN
      U(I,J,K)=0
      V(I,J,K)=0
      W(I,J,K)=0
      D(I,J,K)=0.0
      E(I,J,K)=0.0
100   CONTINUE
C INITIAL CONDITIONS ON WH AND PH
      DO 200 IW=1,IWN
      DO 200 JW=1,JWN
      PINTH(IW,JW)=ARBP
      DO 200 K=1,KN
      WH(IW,JW,K)=0
200   CONTINUE
      RETURN
      END
```

```
NASA*NASA(1).INITIT FOR CREATED ON 15 MAY 74 AT 11:35:25
      C****************************************************************
      C       THIS PROGRAM INITIALIZES TEMP AND DENSITY
      C****************************************************************
            SUBROUTINE INITIT(I,J,K,IN,JN,KN,IW,JW,IWN,JWN,A,B,C,T,RO,
           CMAR,MRH,
           CTREF,RREF,
           CTW,ROW,TO)
            DIMENSION T(IN,JN,KN),RO(IN,JN,KN),TW(IWN,JWN,KN),ROW(IWN,JWN,
            DIMENSION MAR(IN,JN),MRH(IWN,JWN)
            TOD=(TO-TREF)/TREF
            R=A+B*TO+C*TO*TO
            ROD=(R-RREF)/RREF
            DO 10 I=1,IN
            DO 10 J=1,JN
            IF (MAR(I,J).EQ.0) GO TO 12
            DO 11 K=1,KN
            T(I,J,K)=TOD
            RO(I,J,K)=ROD
         11 CONTINUE
         12 CONTINUE
         10 CONTINUE
            DO 20 IW=1,IWN
            DO 20 JW=1,JWN
            IF (MRH(IW,JW).EQ.0) GO TO 22
            DO 21 K=1,KN
            TW(IW,JW,K)=TOD
            ROW(IW,JW,K)=ROD
         21 CONTINUE
         22 CONTINUE
         20 CONTINUE
            RETURN
            END
```

```
NASA*NASA(1).INLET1 ELT CREATED ON 5 MAY 80 AT 11:34:12
C*********************************************************************
C               THIS SURROUTINE FOR INLET AND OUTLETS FOR
C                                DOMAIN
C*********************************************************************
      SUBROUTINE INLET1(I,J,K,IN,JN,KN,U,V,H,G,T,TD,AA,TLL,DT,HTTOT)
      DIMENSION H(IN,JN,KN),G(IN,JN,KN),U(IN,JN,KN)
      DIMENSION V(IN,JN,KN),T(IN,JN,KN),TD(IN,JN,KN)
      TSDT=HTTOT
      TSDT1=HTTOT
C
COMMENT : THIS PROGRAM HAS BEEN WRITTEN SPECIFICALLY FOR
C       : LAKE KEOWEE. USERS MUST CHANGE TO SUIT SPECIFIC
C       : SITE. WATCH OUT FOR COMMENTS TO START OR STOP
C       : CHANGE.
C
      JNM1=JN-1
      KNM1=KN-1
C
COMMENT : START CHANGE.
C
      V(11,1,3)=AA
      G(11,1,3)=AA
      H(11,1,3)=0.0
      T(11,1,3)=TLL
      TD(11,1,3)=TLL
10    CONTINUE
      SF=0.00322579
      PI=3.141592653
      AJOCSE=0.5475
      BJOCSE=11.4525
      IF(TSDT.GE.0.0.AND.TSDT.LE.1.0)SV=SF*(-14.395-(18.75-14
     C.395)*TSDT)
      IF(TSDT.GE.1.0.AND.TSDT.LE.5.0)SV=SF*(-18.754)
      IF(TSDT.GE.5.0.AND.TSDT.LE.8.0)SV=SF*(((16.823+
     C18.754)/3.)*(TSDT-5.0)+18.754)
      IF(TSDT.GE.8.0.AND.TSDT.LE.11.0)SV=-SF*(((16.823-
     C0.1)/3.)*(TSDT-8.0)-16.823)
      IF(TSDT.GE.11.0.AND.TSDT.LE.23.0)SV=SF*0.1
      IF(TSDT.GE.23.0.AND.TSDT.LE.24.0)SV=-SF*(((4.5+0.1))
     C*TSDT-0.1)
C
COMMENT : STOP CHANGE.
C
      DO 20 K=1,KNM1
      DO 20 J=8,JNM1
      U(1,J,K)=SV
      H(1,J,K)=SV
      V(1,J,K)=0.0
      G(1,J,K)=0.0
      T(1,J,K)=T(2,J,K)
      TD(1,J,K)=TD(2,J,K)
20    CONTINUE
      DO 30 K=1,KNM1
C
COMMENT : START CHANGE.
C
      U(7,5,K)=2.*U(8,5,K)-U(9,5,K)
      H(7,5,K)=2.*U(8,5,K)-U(9,5,K)
      V(7,5,K)=0.0
      G(7,5,K)=0.0
C
COMMENT : STOP CHANGE.
C
30    CONTINUE
      SX=0.05789526
      IF(TSDT.GE.0.0.AND.TSDT.LE.6.0)SV1=0.048*SX
      IF(TSDT.GE.6.0.AND.TSDT.LE.8.0)SV1=SX*(((17.54-0.048)/2.)*
     C(TSDT-6.0)+0.048)
      IF(TSDT.GE.8.0.AND.TSDT.LE.12.0)SV1=SX*(((.048-17.54)/4.
     C)*(TSDT-8.0)+17.54)
      IF(TSDT.GE.12.0.AND.TSDT.LE.24.0)SV1=SX*0.048
45    DO 40 K=1,3
      U(13,7,K)=SV1
      H(13,7,K)=SV1
      T(13,7,K)=T(12,7,K)
      TD(13,7,K)=TD(12,7,K)
40    CONTINUE
      T(13,7,4)=T(12,7,4)
```

```
      TO(13,7,4)=TO(12,7,4)
      T(13,7,5)=T(12,7,5)
      TO(13,7,5)=TO(12,7,5)
      CONTINUE
      RETURN
      END
```

```
AGA*NASA(1).INTE FOR CREATED ON 5 MAY 80 AT 11:36:17
      C**************************************************************
      C   THIS SUBROUTINE COMPUTES XINT,YINT,DPSX & DPSY.
      C**************************************************************
      C
            SUBROUTINE INTE(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,MAR,XINT,YINT,A3
           C,AI,AH,AV,TAUX,TAUY
           C,DX,DY,DZ,D,E,DT,DPSX,DPSY,AP)
            DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),MAR(IN,JN),HI(IN,J
            DIMENSION HX(IN,JN),HY(IN,JN)
            DIMENSION A3(KN)
            DIMENSION XINT(IN,JN),YINT(IN,JN)
            DIMENSION DPSX(IN,JN),DPSY(IN,JN)
            DIMENSION D(IN,JN,KN),E(IN,JN,KN)
            DO 200 I=1,IN
            DO 200 J=1,JN
            IF(MAR(I,J).EQ.0) GO TO 200
            YINT(I,J)=0.0
            XINT(I,J)=0.0
            DO 190 K=1,KN
            CALL DINERU(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1HUUX,D1HUVX,MAR)
            CALL DUVY(I,J,K,IN,JN,KN,U,V,HI,DY,D1HUVY,MAR)
            CALL DVVY(I,J,K,IN,JN,KN,U,V,HI,DY,D1HVVY,MAR)
            CALL DVISU(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1UX,D2UX,D1UY,D2UY,MAR)
            CALL DVISV(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1VX,D2VX,D1VY,D2VY,MAR)
            CALL D1ZZ(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,DX,DY,DZ,D1UWZ,A3,
           CTAUX,TAUY,D1VWZ,D1UZ,D2UZ,D1VZ,D2VZ,D1A3Z)
            IF (K.EQ.1) GO TO 1000
            IF (K.EQ.KN) GO TO 1010
            XSUM=(AI*(D1HUUX+D1HUVY+HI(I,J)*D1UWZ)
           C-AH*(D2UX*HI(I,J)+D2UY*HI(I,J))
           C-AH*(D1UX*HX(I,J)+D1UY*HY(I,J))
           C-AV*(1.0/HI(I,J))*(A3(K)*D2UZ+D1A3Z*D1UZ))*DZ
            YSUM=(AI*(D1HUVX+D1HVVY+HI(I,J)*D1VWZ)
           C-AH*(D2VX*HI(I,J)+D2VY*HI(I,J))
           C-AH*(D1VX*HX(I,J)+D1VY*HY(I,J))
           C-AV*(1.0/HI(I,J))*(A3(K)*D2VZ+D1A3Z*D1VZ))*DZ
            GO TO 1100
       1000 CONTINUE
            XSUM=(AI*(D1HUUX+D1HUVY+HI(I,J)*D1UWZ)
           C-AH*(D2UX*HI(I,J)+D2UY*HI(I,J))
           C-AH*(D1UX*HY(I,J)+D1UY*HY(I,J))
           C-AV*(1.0/HI(I,J))*(A3(K)*D2UZ+D1A3Z*D1UZ))*DZ/2.0
            YSUM=(AI*(D1HUVX+D1HVVY+HI(I,J)*D1VWZ)
           C-AH*(D2VX*HI(I,J)+D2VY*HI(I,J))
           C-AH*(D1VX*HX(I,J)+D1VY*HY(I,J))
           C-AV*(1.0/HI(I,J))*(A3(K)*D2VZ+D1A3Z*D1VZ))*DZ/2.0
            D1UT=(U(I,J,K)-D(I,J,K))/DT
            D1VT=(V(I,J,K)-E(I,J,K))/DT
            Q=2.0/DZ
            DPSX(I,J)=(1./AP)*(1./HI(I,J))*(-XSUM*Q-HI(I,J)*D1UT)
            DPSY(I,J)=(1./AP)*(1./HI(I,J))*(-YSUM*Q-HI(I,J)*D1VT)
            GO TO 1100
       1010 CONTINUE
            XSUM=(AI*(D1HUUX+D1HUVY+HI(I,J)*D1UWZ)
           C-AH*(D2UX*HI(I,J)+D2UY*HI(I,J))
           C-AH*(D1UX*HX(I,J)+D1UY*HY(I,J))
           C-AV*(1.0/HI(I,J))*(A3(K)*D2UZ+D1A3Z*D1UZ))*DZ/2.0
            YSUM=(AI*(D1HUVX+D1HVVY+HI(I,J)*D1VWZ)
           C-AH*(D2VX*HI(I,J)+D2VY*HI(I,J))
           C-AH*(D1VX*HX(I,J)+D1VY*HY(I,J))
           C-AV*(1.0/HI(I,J))*(A3(K)*D2VZ+D1A3Z*D1VZ))*DZ/2.0
       1100 CONTINUE
            XINT(I,J)=XSUM+XINT(I,J)
            YINT(I,J)=YSUM+YINT(I,J)
        190 CONTINUE
        200 CONTINUE
            RETURN
            END
```

```
NASA.NASA(1). J1"T SYM CREATED ON 19 NOV 77 AT 09:57:00
0.00108,0.78
1.0,0.0092,301.56,1.0
0.001,100,1.8,1.0
0.0526,0.0526,0.25
1.0,0.0092,301.56
1.000428,-0.000019,-0.0000046
10.0
1742.22,0.0,0,0
0.22787,1.0
0,0,0,0,0,0,7,3,3,3,3,3,3,3,3,3,3,3,3,5
0,0,0,0,0,0,2,11,11,11,11,11,11,11,11,11,11,11,11,11,1
0,0,0,0,7,3,8,11,11,11,11,11,11,11,11,11,11,6,4,4,10
0,0,0,0,2,11,11,11,11,11,11,11,11,11,11,11,11,1,0,0,0
0,0,0,0,9,8,11,11,11,11,11,11,11,11,11,11,1,0,0,0
0,0,0,0,0,2,11,11,11,11,11,11,11,11,11,11,6,3,3,5
0,0,7,3,3,5,11,11,11,11,11,11,11,11,11,11,1,11,11,1
0,0,2,11,11,11,11,11,11,11,11,11,11,11,11,11,6,8,11,6,10
0,0,2,11,11,11,11,11,11,11,11,11,11,11,11,6,10,2,11,1,0
7,3,8,11,11,11,11,11,11,11,11,11,11,11,11,1,0,9,4,10,0
2,11,11,11,11,11,11,11,11,11,11,11,11,11,11,6,3,5,0,0,0
9,4,4,4,4,8,11,11,11,11,6,4,8,11,11,11,1,0,0,0
0,0,0,0,0,9,4,8,11,11,1,0,2,11,11,11,1,0,0,0
0,0,0,0,0,0,0,2,11,6,10,0,9,8,11,11,6,5,0,0
0,0,0,0,0,0,0,2,11,1,0,0,0,2,11,11,11,1,0,0
0,0,0,0,0,0,0,2,11,1,0,0,0,9,4,4,4,10,0,0
0,0,0,0,0,0,0,9,4,10,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,4,10,10,10,10,10,10,10,10,10,10,10,3
0,0,0,0,0,0,2,9,9,9,9,9,9,9,9,7,8,8,7
0,0,0,0,4,10,9,9,9,9,9,9,9,9,9,1,0,0,0
0,0,0,0,6,9,9,9,9,9,9,9,9,9,1,0,0,0
0,0,0,0,0,2,9,9,9,9,9,9,9,9,9,1,0,0,0
0,0,0,0,0,2,9,9,9,9,9,9,9,9,9,7,10,10,3
0,0,4,10,10,9,9,9,9,9,9,9,9,9,8,9,9,7
0,0,2,9,9,9,9,9,9,9,9,9,9,7,0,2,1,0
0,0,2,9,9,9,9,9,9,9,9,9,1,0,0,6,7,0
4,10,9,9,9,9,7,9,9,9,9,9,1,0,0,0,0,0
6,6,8,8,8,9,9,9,9,9,8,8,9,9,10,3,0,0,0
0,0,0,0,0,6,8,9,9,1,0,0,2,9,9,1,0,0,0
0,0,0,0,0,0,0,2,9,7,0,0,6,9,9,1,0,0,0
0,0,0,0,0,0,0,2,1,0,0,0,0,2,9,9,3,0,0
0,0,0,0,0,0,0,2,1,0,0,0,0,6,8,8,7,0,0
0,0,0,0,0,0,0,6,7,0,0,0,0,0,0,0,0,0,0
1.633,-0.33,-2.78,0.0,-1,1
15.0
1.073,-0.72,-1.67,0.0,-1,1
75.0
2.325,-1.61,-1.61,0.0,-1,1
60.0
1.565,-2.22,-2.28,0.0,-1,1
15.0
2.056,-1.83,-1.09,0.0,-1,1
50.0
1.788,-2.17,-2.22,0.0,-1,1
85.0
2.012,-2.72,-2.78,20.94,1,1
85.0
2.280,-1.67,-2.78,195.39,-1,-1
60.0
0.626,0.01,-3.33,369.85,1,-1
5.0
1.386,3.06,-2.22,544.31,-1,-1
75.0
1.609,5.83,-2.22,655.96,1,-1
15.0
1.788,8.83,-1.39,725.75,1,1
40.0
3.129,11.06,-2.78,746.68,1,1
80.0
2.593,12.28,-5.0,704.81,1,1
70.0
1.520,13.39,-5.56,579.20,-1,1
80.0
1.207,13.89,-5.56,383.81,1,1
75.0
1.565,13.83,-5.61,146.55,1,-1
55.0
1.609,13.72,-3.33,20.94,-1,1
15.0
```

```
2.056,11.72,-4.44,0.0,-1,-1
30.0
1.162,9.72,-2.78,0.0,-1,-1
25.0
2.772,8.33,5.28,0.0,1,-1
55.0
2.861,7.78,5.56,0.0,1,-1
55.0
2.995,7.00,5.28,0.0,1,-1
50.0
1.386,5.28,3.89,0.0,1,-1
60.0
```

```
ASA*NASA(1).OLDT FOR CREATED ON 5 MAY 80 AT 11:41:11
      C*****************************************************
      C   THIS SUBROUTINE SETS THE VALUES OF THE TEMPERATURE
      C*****************************************************
      C
            SUBROUTINE OLDT(I,J,K,IN,JN,KN,TO,T)
            DIMENSION T(IN,JN,KN),TO(IN,JN,KN)
            DO 10 I=1,IN
            DO 10 J=1,JN
            DO 10 K=1,KN
            T(I,J,K)=TO(I,J,K)
      10    CONTINUE
            RETURN
            END
```

```
NASA*NASA(1).OLDUV FOR CREATED ON 14 MAY 74 AT 15:49:33
      C*************************************************************************
      C     THIS PROGRAM SETS THE VALUES OF D AND E EQUAL TO U AND V RESPEC
      C          IN ORDER TO RETAIN VALUES OF U AND V AT ONE TIME STEP LAG
      C*************************************************************************
            SUBROUTINE OLDUV(I,J,K,IN,JN,KN,U,V,D,E)
            DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN)
            DO 831 K=1,KN
            DO 831 I=1,IN
            DO 831 J=1,JN
            D(I,J,K)=U(I,J,K)
            E(I,J,K)=V(I,J,K)
      831   CONTINUE
            RETURN
            END
```

```
ASA*NASA(1).PREIL FOR CREATED ON 5 MAY 80 AT 11:43:26
C**************************************************************
C  THIS SUBROUTINE COMPUTES PRESSURE FOR FAR FIELD
C**************************************************************
C
      SUBROUTINE PREIL(EPS,MAXIT,IN,JN,P,ITN,DPSX,DPSY,FH,DL2,OMEGA,
     CMRH,I,J,K,IW,JW,DX,DY,EX,IWN,JWN,ARBP)
      DIMENSION P(IWN,JWN),FH(IWN,JWN),DPSX(IN,JN),DPSY(IN,JN)
      DIMENSION MRH(IWN,JWN)
      ITN=0
    1 EX=0.0
      DD=ARBP
      ITN=ITN+1
      DO 10 IWD=1,IWN
      DO 10 JW=1,JWN
      IW=(IWN+1)-IWD
      I=IW
      J=JW
      IF (MRH(IW,JW).EQ.0) GO TO 57
      IF (MRH(IW,JW).EQ.1) GO TO 11
      IF (MRH(IW,JW).EQ.2) GO TO 12
      IF (MRH(IW,JW).EQ.3) GO TO 13
      IF (MRH(IW,JW).EQ.4) GO TO 14
      IF (MRH(IW,JW).EQ.5) GO TO 18
      IF (MRH(IW,JW).EQ.6) GO TO 16
      IF (MRH(IW,JW).EQ.7) GO TO 17
      IF (MRH(IW,JW).EQ.8) GO TO 18
      IF (MRH(IW,JW).EQ.10) GO TO 19
      PN=.25*(P(IW-1,JW)+P(IW+1,JW)+P(IW,JW-1)+P(IW,JW+1)-DL2*FH(IW,JW
      GO TO 50
   11 CONTINUE
      PN=.25*(P(IW-1,JW)+P(IW+1,JW)+P(IW,JW-1)+P(IW,JW)+(DPSY(I,J+1)
     C+DPSY(I+1,J+1))*DY/2.-DL2*FH(IW,JW))
      GO TO 50
   12 CONTINUE
      PN=.25*(P(IW-1,JW)+P(IW+1,JW)+P(IW,JW+1)+P(IW,JW)
     C-(DPSY(I,J)+DPSY(I+1,J))*DY/2-DL2*FH(IW,JW))
      GO TO 50
   13 CONTINUE
      PN=.25*(P(IW+1,JW)+P(IW,JW-1)+P(IW,JW)-(DPSX(I,J)+DPSX(I,J+1))
     C*DX/2+P(IW,JW)+(DPSY(I,J+1)+DPSY(I+1,J+1))*DY/2-DL2*FH(IW,JW))
      GO TO 50
   14 CONTINUE
      PN=.25*(P(IW+1,JW)+P(IW,JW+1)+P(IW,JW)-(DPSX(I,J)+DPSX(I,J+1))*
     CDX/2+P(IW,JW)-(DPSY(I,J)+DPSY(I+1,J))*DY/2-DL2*FH(IW,JW))
      GO TO 50
   15 CONTINUE
      PN=ARBP
      GO TO 50
   16 CONTINUE
      PN=.25*(P(IW,JW+1)+P(IW-1,JW)+P(IW,JW)+(DPSX(I+1,J+1)+DPSX(I+1,J
     C*DX/2+P(IW,JW)-(DPSY(I,J)+DPSY(I+1,J))*DY/2-DL2*FH(IW,JW))
      GO TO 50
   17 CONTINUE
      PN=.25*(P(IW-1,JW)+P(IW,JW-1)+P(IW,JW)+(DPSX(I+1,J)+DPSX(I+1,J+)
     C*DX/2+P(IW,JW)+(DPSY(I,J+1)+DPSY(I+1,J+1))*DY/2-DL2*FH(IW,JW))
      GO TO 50
   18 CONTINUE
      PN=.25*(P(IW,JW+1)+P(IW-1,JW)+P(IW,JW-1)+
     CP(IW,JW)+(DPSX(I+1,J)+DPSX(I+1,J+1))*DX/2-
     CDL2*FH(IW,JW))
      GO TO 50
   19 CONTINUE
      PN=.25*(P(IW+1,JW)+P(IW,JW-1)+P(IW,JW+1)+DD-DL2*FH(IW,JW))
   50 CONTINUE
      PNEW=OMEGA*PN+(1-OMEGA)*P(IW,JW)
      IF(ABS(PNEW).LT.(10.**(-16))) GO TO 51
      DIFF=ABS((PNEW-P(IW,JW))/PNEW)
      IF (DIFF.LT.EX) GO TO 51
      EX=DIFF
   51 P(IW,JW)=PNEW
   57 CONTINUE
   10 CONTINUE
      IF(EX.LT.EPS) GO TO 20
      IF(ITN.LT.MAXIT) GO TO 1
   20 CONTINUE
      RETURN
      END
```

```
ASA*NASA(1).PRITEX FOR CREATED ON 14 MAY 74 AT 15:49:42
      C**********************************************************************
      C     THIS PROGRAM PRINTS OUT THE VALUES OF NUMBER OF ITERATIONS ANDF
      C                                RESIDUAL ERROR IN SOLVING POISSON
      C**********************************************************************
            SUBROUTINE PRITFX(ITN,EX)
            PRINT 5500,ITN,EX
      5500  FORMAT(/' ITN=',I4,5X,' EX=',E15.7)
            RETURN
            END
```

```
NASA(1).PRUV SYM CREATED ON 5 MAY 80 AT 11:48:16
      C**********************************************************
      C   THIS SUBROUTINE PRINTS THE VALUE OF U AND V AT ALL MAIN
      C   GRID POINTS.
      C**********************************************************
      C
            SUBROUTINE PRUV(I,J,K,IN,JN,KN,U,V,UA,VA,MAR)
            DIMENSION U(IN,JN,KN),V(IN,JN,KN),MAR(IN,JN),
           CUA(IN,JN,KN),VA(IN,JN,KN)
            DO 9100 K=1,KN
            DO 9100 J=1,JN
            DO 9100 I=1,IN
            UA(I,J,K)=U(I,J,K)*30.
            VA(I,J,K)=V(I,J,K)*30.
      C     IF(MAR(I,J).EQ.0)UA=1000000.00
      C     IF(MAR(I,J).EQ.0)VA=1000000.00
      9100    CONTINUE
            DO 150 K=1,KN
            WRITE(6,105)K
            DO 140 I=1,IN
            WRITE(6,106)(UA(I,J,K),J=1,JN)
      140     CONTINUE
      150     CONTINUE
            DO 151 K=1,KN
            WRITE(6,107)K
            DO 141 I=1,IN
            WRITE(6,106)(VA(I,J,K),J=1,JN)
      141     CONTINUE
      151     CONTINUE
      105     FORMAT('1','U-VELOCITY FOR K='I5)
      107     FORMAT('1','V-VELOCITY FOR K='I5)
      106     FORMAT(//,22F6.2)
            RETURN
            END
```

```
ASA*NASA(1).READ2 FOR CREATED ON 1 MAR 79 AT 12:38:36
      C*****************************************************************
      C     THIS PROGRAM READS TAPE FOR DATA I FOR THE VARIABLE DENSITY CAS
      C*****************************************************************
            SUBROUTINE READ2(U,V,WH,PINTH,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,
           CHX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,
           CTAI,TAH,TAV,AKT,CB,CW,A,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,TAM
           CTTOT)
            DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
           CWH(IWN,JWN,KN),PINTH(IWN,JWN)
            DIMENSION HX(IN,JN),HY(IN,JN),HI(IN,JN),MAR(IN,JN),MRH(IWN,JWN),
           CW(IN,JN,KN),WR(IN,JN,KN),WRH(IWN,JWN,KN)
            DIMENSION T(IN,JN,KN),RO(IN,JN,KN),TW(IWN,JWN,KN),ROW(IWN,JWN,KM
            READ (7,END=1)  (((U(I,J,K),K=1,KN),J=1,JN),I=1,IN),
           C(((V(I,J,K),K=1,KN),J=1,JN),I=1,IN),
           C(((D(I,J,K),K=1,KN),J=1,JN),I=1,IN),
           C(((E(I,J,K),K=1,KN),J=1,JN),I=1,IN),
           C(((WH(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
           C(((W(I,J,K),K=1,KN),J=1,JN),I=1,IN),
           C(((WR(I,J,K),K=1,KN),J=1,JN),I=1,IN),
           C(((WRH(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
           C((PINTH(IW,JW),JW=1,JWN),IW=1,IWN)
           C,((HI(I,J),J=1,JN),I=1,IN),((HX(I,J),J=1,JN),I=1,IN),((HY(I,J),J
           C1,JN),I=1,IN),((MAR(I,J),J=1,JN),I=1,IN),((MRH(IW,JW),JW=1,JWN),
           CIW=1,IWN),(((T(I,J,K),K=1,KN),J=1,JN),I=1,IN),
           C(((RO(I,J,K),K=1,KN),J=1,JN),I=1,IN),
           C(((TW(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
           C(((ROW(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
           CTAI,TAH,TAV,AKT,CB,CW,A,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,TAM
           CAI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,TTOT
    1       CONTINUE
            RETURN
            END
```

```
ASA*NASA(1).READ3K ELT CREATED ON 5 MAY 80 AT 11:50:20
C*****************************************************************
C   THIS SUBROUTINE READS AND PRINTS THE MAR & MRH MATRICES
C*****************************************************************
C
      SUBROUTINE READ3K(I,J,IN,JN,IW,JW,IWN,JWN,MAR,MRH)
      DIMENSION MAR(IN,JN),MRH(IWN,JWN)
      DO 300 I=1,IN
      READ 2,(MAR(I,J),J=1,JN)
      PRINT 3,I,(MAR(I,J),J=1,JN)
300   CONTINUE
      DO 400 IW=1,IWN
      READ 2,(MRH(IW,JW),JW=1,JWN)
      PRINT 3,IW,(MRH(IW,JW),JW=1,JWN)
400   CONTINUE
2     FORMAT()
3     FORMAT(/15X,30I4)
      RETURN
      END
```

```
A547NASA(1).ROINTX FOR CREATED ON 5 MAY 80 AT 11:52:21
      C*********************************************************
      C   THIS SUBROUTINE COMPUTES XP IN THE POISSONS EQUATION
      C*********************************************************
      C
            SUBROUTINE ROINTX(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,
           CMAR,RINTX,HX,XINT)
            DIMENSION RINTX(IN,JN,KN),RO(IN,JN,KN),XINT(IN,JN),HI(IN,JN),
           CMAR(IN,JN),HX(IN,JN)
            DO 100 I=1,IN
            DO 100 J=1,JN
            IF (MAR(I,J).EQ.0) GO TO 101
            RINTX(I,J,1)=0.0
            DO 110 K=2,KN
            IF (MAR(I,J).EQ.1) GO TO 11
            IF (MAR(I,J).EQ.2) GO TO 12
            IF (MAR(I,J).EQ.3) GO TO 13
            IF (MAR(I,J).EQ.4) GO TO 14
            IF (MAR(I,J).EQ.5) GO TO 15
            IF (MAR(I,J).EQ.6) GO TO 16
            IF (MAR(I,J).EQ.7) GO TO 17
            IF (MAR(I,J).EQ.8) GO TO 18
            IF (MAR(I,J).EQ.9) GO TO 19
            IF (MAR(I,J).EQ.10) GO TO 20
            RX=DZ*(RO(I+1,J,K)+RO(I+1,J,K-1)-RO(I-1,J,K)-RO(I-1,J,K-1))/(4*D
            GO TO 102
      11    CONTINUE
            RX=DZ*(RO(I+1,J,K)+RO(I+1,J,K-1)-RO(I-1,J,K)-RO(I-1,J,K-1))/(4*D
            GO TO 102
      12    CONTINUE
            RX=DZ*(RO(I+1,J,K)+RO(I+1,J,K-1)-RO(I-1,J,K)-RO(I-1,J,K-1))/(4*D
            GO TO 102
      13    CONTINUE
            RX=DZ*(4*(RO(I+1,J,K)+RO(I+1,J,K-1))-3*(RO(I,J,K)+RO(I,J,K-1))
           C-(RO(I+2,J,K)+RO(I+2,J,K-1)))/(4*DX)
      14    CONTINUE
            RX=DZ*(3*(RO(I,J,K)+RO(I,J,K-1))+(RO(I-2,J,K)+RO(I-2,J,K-1))
           C-4*(RO(I-1,J,K)+RO(I-1,J,K-1)))/(4*DX)
            GO TO 102
      15    CONTINUE
            RX=DZ*(4*(RO(I+1,J,K)+RO(I+1,J,K-1))-3*(RO(I,J,K)+RO(I,J,K-1))
           C-(RO(I+2,J,K)+RO(I+2,J,K-1)))/(4*DX)
            GO TO 102
      16    CONTINUE
            RX=DZ*(RO(I+1,J,K)+RO(I+1,J,K-1)-RO(I-1,J,K)-RO(I-1,J,K-1))/(4*D
            GO TO 102
      17    CONTINUE
            RX=DZ*(4*(RO(I+1,J,K)+RO(I+1,J,K-1))-3*(RO(I,J,K)+RO(I,J,K-1))
           C-(RO(I+2,J,K)+RO(I+2,J,K-1)))/(4*DX)
            GO TO 102
      18    CONTINUE
            RX=DZ*(RO(I+1,J,K)+RO(I+1,J,K-1)-RO(I-1,J,K)-RO(I-1,J,K-1))/(4*D
            GO TO 102
      19    CONTINUE
            RX=DZ*(3*(RO(I,J,K)+RO(I,J,K-1))+(RO(I-2,J,K)+RO(I-2,J,K-1))
           C-4*(RO(I-1,J,K)+RO(I-1,J,K-1)))/(4*DX)
            GO TO 102
      20    CONTINUE
            RX=DZ*(3*(RO(I,J,K)+RO(I,J,K-1))+(RO(I-2,J,K)+RO(I-2,J,K-1))
           C-4*(RO(I-1,J,K)+RO(I-1,J,K-1)))/(4*DX)
            GO TO 102
      102   CONTINUE
            RINTX(I,J,K)=RINTX(I,J,K-1)+RX*HI(I,J)+(RO(I,J,K)+RO(I,J,K-1))*D
           CHX(I,J)/2.0
      110   CONTINUE
      101   CONTINUE
      100   CONTINUE
            DO 200 I=1,IN
            DO 200 J=1,JN
            IF (MAR(I,J).EQ.0) GO TO 201
            DO 210 K=2,KN
            RINTX(I,J,K)=RINTX(I,J,K)-(K-1)*DZ*HX(I,J)*(RO(I,J,K)+RO(I,J,K-1
           C/2.0
      210   CONTINUE
      201   CONTINUE
      200   CONTINUE
            DO 300 I=1,IN
            DO 300 J=1,JN
            IF (MAR(I,J).EQ.0) GO TO 301
```

```
      DO 310 K=2,KN
      RSUMX=(RINTX(I,J,K)+RINTX(I,J,K-1))*(DZ/2)*AP*EUL*HI(I,J)
      XINT(I,J)=XINT(I,J)+RSUMX
  310 CONTINUE
  301 CONTINUE
  300 CONTINUE
      RETURN
      END
```

ASA*NASA(1).ROINTY FOR CREATED ON 5 MAY 80 AT 11:54:03

```
C*****************************************************************
C THIS SUBROUTINE COMPUTES YP IN THE POISSONS EQUATION
C*****************************************************************
C
      SUBROUTINE ROINTY(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,MAR,
     CRINTY,HY,YINT)
      DIMENSION RINTY(IN,JN,KN),RO(IN,JN,KN),YINT(IN,JN),HI(IN,JN),
     CHY(IN,JN),MAR(IN,JN)
      DO 100 I=1,IN
      DO 100 J=1,JN
      IF (MAR(I,J).EQ.0) GO TO 101
      RINTY(I,J,1)=0.0
      DO 110 K=2,KN
      IF (MAR(I,J).EQ.1) GO TO 11
      IF (MAR(I,J).EQ.2) GO TO 12
      IF (MAR(I,J).EQ.3) GO TO 13
      IF (MAR(I,J).EQ.4) GO TO 14
      IF (MAR(I,J).EQ.5) GO TO 15
      IF (MAR(I,J).EQ.6) GO TO 16
      IF (MAR(I,J).EQ.7) GO TO 17
      IF (MAR(I,J).EQ.8) GO TO 18
      IF (MAR(I,J).EQ.9) GO TO 19
      IF (MAR(I,J).EQ.10) GO TO 20
      RY=DZ*(RO(I,J+1,K)+RO(I,J+1,K-1)-RO(I,J-1,K)-RO(I,J-1,K-1))/(4*D
      GO TO 102
 11   CONTINUE
      RY=DZ*(3*(RO(I,J,K)+RO(I,J,K-1))+(RO(I,J-2,K)+RO(I,J-2,K-1))
     C-4*(RO(I,J-1,K)+RO(I,J-1,K-1)))/(4*DY)
      GO TO 102
 12   CONTINUE
      RY=DZ*(4*(RO(I,J+1,K)+RO(I,J+1,K-1))-3*(RO(I,J,K)+RO(I,J,K-1))
     C-(RO(I,J+2,K)+RO(I,J+2,K-1)))/(4*DY)
      GO TO 102
 13   CONTINUE
      RY=DZ*(RO(I,J+1,K)+RO(I,J+1,K-1)-RO(I,J-1,K)-RO(I,J-1,K-1))/(4*D
      GO TO 102
 14   CONTINUE
      RY=DZ*(RO(I,J+1,K)+RO(I,J+1,K-1)-RO(I,J-1,K)-RO(I,J-1,K-1))/(4*D
      GO TO 102
 15   CONTINUE
      RY=DZ*(3*(RO(I,J,K)+RO(I,J,K-1))+(RO(I,J-2,K)+RO(I,J-2,K-1))
     C-4*(RO(I,J-1,K)+RO(I,J-1,K-1)))/(4*DY)
      GO TO 102
 16   CONTINUE
      RY=DZ*(RO(I,J+1,K)+RO(I,J+1,K-1)-RO(I,J-1,K)-RO(I,J-1,K-1))/(4*D
      GO TO 102
 17   CONTINUE
      RY=DZ*(4*(RO(I,J+1,K)+RO(I,J+1,K-1))-3*(RO(I,J,K)+RO(I,J,K-1))
     C-(RO(I,J+2,K)+RO(I,J+2,K-1)))/(4*DY)
      GO TO 102
 18   CONTINUE
      RY=DZ*(RO(I,J+1,K)+RO(I,J+1,K-1)-RO(I,J-1,K)-RO(I,J-1,K-1))/(4*D
      GO TO 102
 19   CONTINUE
      RY=DZ*(4*(RO(I,J+1,K)+RO(I,J+1,K-1))-3*(RO(I,J,K)+RO(I,J,K-1))
     C-(RO(I,J+2,K)+RO(I,J+2,K-1)))/(4*DY)
      GO TO 102
 20   CONTINUE
      RY=DZ*(3*(RO(I,J,K)+RO(I,J,K-1))+(RO(I,J-2,K)+RO(I,J-2,K-1))
     C-4*(RO(I,J-1,K)+RO(I,J-1,K-1)))/(4*DY)
      GO TO 102
 102  CONTINUE
      RINTY(I,J,K)=RINTY(I,J,K-1)+RY*HI(I,J)+(RO(I,J,K)+RO(I,J,K-1))*D
     CHY(I,J)/2.0
 110  CONTINUE
 101  CONTINUE
 100  CONTINUE
      DO 200 I=1,IN
      DO 200 J=1,JN
      IF (MAR(I,J).EQ.0) GO TO 201
      DO 210 K=2,KN
      RINTY(I,J,K)=RINTY(I,J,K)-(K-1)*DZ*HY(I,J)*(RO(I,J,K)+RO(I,J,K-1
     C/2.0
 210  CONTINUE
 201  CONTINUE
 200  CONTINUE
      DO 300 I=1,IN
      DO 300 J=1,JN
```

```
      IF (MAR(I,J).EQ.0) GO TO 301
      DO 310 K=2,KN
      RSUMY=(RINTY(I,J,K)+RINTY(I,J,K-1))*(DZ/2)*AP*CUL*HI(I,J)
      YINT(I,J)=YINT(I,J)+RSUMY
  310 CONTINUE
  301 CONTINUE
  300 CONTINUE
      RETURN
      END
```

ASA*NASA(1).RWH FOR CREATED ON 5 MAY 80 AT 11:56:16

```
C**************************************************************
C   THIS SUBROUTINE COMPUTES VERTICAL VELOCITIES AT HALF
C   GRID POINTS.
C**************************************************************
C
      SUBROUTINE RWH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,DX,DY,DZ,
     CMRH)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),WH(IWN,JWN,KN),HI(IN,JN)
      DIMENSION MRH(IWN,JWN)
      KNM1=KN-1
      DO 10 IW=1,IWN
      DO 10 JW=1,JWN
      IF (MRH(IW,JW).EQ.0) GO TO 8
      DO 9 KD=1,KNM1
      K=KN-KD+1
      I=IW
      J=JW
      DIHUX=(HI(I+1,J+1)*(U(I+1,J+1,K)+U(I+1,J+1,K-1))+HI(I+1,J)*
     C(U(I+1,J,K)+U(I+1,J,K-1))-HI(I,J+1)*(U(I,J+1,K)+U(I,J+1,K-1))
     C-HI(I,J)*(U(I,J,K)+U(I,J,K-1)))/(4*DX)
      DIHVY=(HI(I+1,J+1)*(V(I+1,J+1,K)+V(I+1,J+1,K-1))+HI(I,J+1)*
     C(V(I,J+1,K)+V(I,J+1,K-1))-HI(I+1,J)*(V(I+1,J,K)+V(I+1,J,K-1))
     C-HI(I,J)*(V(I,J,K)+V(I,J,K-1)))/(4*DY)
      HH=(HI(I+1,J+1)+HI(I+1,J)+HI(I,J+1)+HI(I,J))/4.0
      WH(IW,JW,K-1)=WH(IW,JW,K)+(1.0/HH)*(DIHUX+DIHVY)*DZ
    9 CONTINUE
    8 CONTINUE
   10 CONTINUE
      RETURN
      END
```

```
ASA*NASA(1).RWR ELT CREATED ON 5 MAY 80 AT 11:58:57
C***********************************************************
C   THIS SUBROUTINE COMPUTES REAL VERTICAL VELOCITIES AT
C   INTEGRAL GRID POINTS.
C***********************************************************
C
      SUBROUTINE RWR(I,J,K,IN,JN,KN,U,V,W,WR,HI,HX,HY,DZ,MAR)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WR(IN,JN,KN),
     CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN)
      DO 10 I=1,IN
      DO 10 J=1,JN
      IF (MAR(I,J).LT.11) GO TO 8
      KNM1=KN-1
      DO 9 K=1,KNM1
      WR(I,J,K)=(K-1)*DZ*(U(I,J,K)*HX(I,J)+V(I,J,K)*HY(I,J))+HI(I,J)
     C*W(I,J,K)
    9 CONTINUE
    8 CONTINUE
   10 CONTINUE
      RETURN
      END
```

```
ASA*NASA(1).RWMH ELT CREATED ON 5 MAY 80 AT 12:01:08
      C*************************************************************
      C   THIS SUBROUTINE COMPUTES REAL VERTICAL VELOCITIES AT HALF
      C   GRID POINTS.
      C*************************************************************
      C
            SUBROUTINE RWMH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,HX,HY,
           CDX,DY,DZ,MRH,WRH)
            DIMENSION U(IN,JN,KN),V(IN,JN,KN),WH(IWN,JWN,KN),HI(IN,JN),
           CHX(IN,JN),HY(IN,JN),MRH(IWN,JWN)
            DIMENSION WRH(IWN,JWN,KN)
            KNM1=KN-1
            DO 10 IW=1,IWN
            DO  10 JW=1,JWN
            IF(MRH(IW,JW).EQ.0)GO TO 8
            HXAV=(HX(I+1,J)+HX(I+1,J+1)+HX(I,J)+HX(I,J+1))/4.
            HYAV=(HY(I+1,J)+HY(I+1,J+1)+HY(I,J)+HY(I,J+1))/4.
            HIAV=(HI(I+1,J)+HI(I+1,J+1)+HI(I,J)+HI(I,J+1))/4.
            DO 9 K=1,KNM1
            I=IW
            J=JW
            UAV=(U(I+1,J,K)+U(I+1,J+1,K)+U(I,J,K)+U(I,J+1,K))/4.
            VAV=(V(I+1,J,K)+V(I+1,J+1,K)+V(I,J,K) +V(I,J+1,K))/4.
            WRH(IW,JW,K)=(K-1)*DZ*(UAV*HXAV+VAV*HYAV)+HIAV*WH(IW,JW,K)
      9     CONTINUE
      8     CONTINUE
      10    CONTINUE
            RETURN
            END
```

ASA*NASA(1).STORE2 FOR CREATED ON 5 MAY 80 AT 12:02:30

```
C*****************************************************************
C        THIS PROGRAM STORES THE RELEVANT DATA INTO FILE 8
C*****************************************************************
      SUBROUTINE STORE2(U,V,WH,PINTH,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,
     CHX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,
     CTAI,TAH,TAV,AKT,CB,CW,A,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,TAMB
     CTTOT)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
     CWH(IWN,JWN,KN),PINTH(IWN,JWN)
      DIMENSION HX(IN,JN),HY(IN,JN),HI(IN,JN),MAR(IN,JN),MRH(IWN,JWN),
     CW(IN,JN,KN),WR(IN,JN,KN),WRH(IWN,JWN,KN)
      DIMENSION T(IN,JN,KN),RO(IN,JN,KN),TW(IWN,JWN,KN),ROW(IWN,JWN,KN
      WRITE (8) (((U(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((V(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((D(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((E(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((WH(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
     C(((W(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((WR(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((WRH(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
     C((PINTH(IW,JW),JW=1,JWN),IW=1,IWN)
     C,((HI(I,J),J=1,JN),I=1,IN),((HX(I,J),J=1,JN),I=1,IN),((HY(I,J),J
     C=1,JN),I=1,IN),((MAR(I,J),J=1,JN),I=1,IN),((MRH(IW,JW),JW=1,JWN),
     CIW=1,IWN),(((T(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((RO(I,J,K),K=1,KN),J=1,JN),I=1,IN),
     C(((TW(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
     C(((ROW(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
     CTAI,TAH,TAV,AKT,CB,CW,A,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,TAMB
     CAI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,TTOT
      END FILE 8
      RETURN
      END
```

NASA*NASA(1).TEMB2 FOR CREATED ON 5 MAY 80 AT 12:04:20

```
C*******************************************************************
C     THIS SUBROUTINE COMPUTES BOUNDARY TEMPERATURES
C*******************************************************************
C
      SUBROUTINE TEMB2(I,J,K,IN,JN,KN,TO,DX,DY,DZ,HAR,CB,HI,AKT,CW,
     CTAMB,HX,HY,T,TREF,TAV,TAI,TAH,B3,DT)
      DIMENSION T(IN,JN,KN),TO(IN,JN,KN),MAR(IN,JN),HX(IN,JN),HY(IN,J
     CHI(IN,JN)
      KNM1=KN-1
      DO 100 K=1,KN
      DO 100 I=1,IN
      DO 100 J=1,JN
      D1HTUX=0.0
      D1HTVY=0.0
      D1TWZ=0.0
      IF (MAR(I,J).EQ.0) GO TO 300
      IF (MAR(I,J).EQ.11) GO TO 300
      IF (MAR(I,J).EQ.1) GO TO 11
      IF (MAR(I,J).EQ.2) GO TO 12
      IF (MAR(I,J).EQ.3) GO TO 13
      IF (MAR(I,J).EQ.4) GO TO 14
      IF (MAR(I,J).EQ.5) GO TO 15
      IF (MAR(I,J).EQ.6) GO TO 16
      IF (MAR(I,J).EQ.7) GO TO 17
      IF (MAR(I,J).EQ.8) GO TO 18
      IF (MAR(I,J).EQ.9) GO TO 19
      IF (MAR(I,J).EQ.10) GO TO 20
   11 CONTINUE
      D1TX=(T(I+1,J,K)-T(I-1,J,K))/(2*DX)
      D2TX=(T(I+1,J,K)+T(I-1,J,K)-2*T(I,J,K))/(DX*DX)
      D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
      D1TY=0.0
      D2TY=2*(T(I,J-1,K)-T(I,J,K))/(DZ*DZ)
      IF (K.EQ.1) GO TO 110
      IF (K.EQ.KN) GO TO 120
      GO TO 200
   12 CONTINUE
      D1TX=(T(I+1,J,K)-T(I-1,J,K))/(2*DX)
      D2TX=(T(I+1,J,K)+T(I-1,J,K)-2*T(I,J,K))/(DX*DX)
      D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
      D1TY=0.0
      D2TY=2*(T(I,J+1,K)-T(I,J,K))/(DY*DY)
      IF (K.EQ.1) GO TO 110
      IF (K.EQ.KN) GO TO 120
      GO TO 200
   13 CONTINUE
      D1TX=0.0
      D2TX=2*(T(I+1,J,K)-T(I,J,K))/(DX*DX)
      D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
      D1TY=(T(I,J+1,K)-T(I,J-1,K))/(2*DY)
      D2TY=(T(I,J+1,K)+T(I,J-1,K)-2*T(I,J,K))/(DY*DY)
      IF (K.EQ.1) GO TO 110
      IF (K.EQ.KN) GO TO 120
      GO TO 200
   14 CONTINUE
      D1TX=0.0
      D2TX=2*(T(I-1,J,K)-T(I,J,K))/(DX*DX)
      D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
      D1TY=(T(I,J+1,K)-T(I,J-1,K))/(2*DY)
      D2TY=(T(I,J+1,K)+T(I,J-1,K)-2*T(I,J,K))/(DY*DY)
      IF (K.EQ.1) GO TO 110
      IF (K.EQ.KN) GO TO 120
      GO TO 200
   15 CONTINUE
      D1TX=0.0
      D2TX=2*(T(I+1,J,K)-T(I,J,K))/(DX*DX)
      D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
      D1TY=0.0
      D2TY=2*(T(I,J-1,K)-T(I,J,K))/(DZ*DZ)
      IF (K.EQ.1) GO TO 110
      IF (K.EQ.KN) GO TO 120
      GO TO 200
   17 CONTINUE
      D1TX=0.0
      D2TX=2*(T(I+1,J,K)-T(I,J,K))/(DX*DX)
      D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
      D1TY=0.0
      D2TY=2*(T(I,J+1,K)-T(I,J,K))/(DY*DY)
```

```
      IF (K.EQ.1) GO TO 110
      IF (K.EQ.KN) GO TO 120
      GO TO 200
   19 CONTINUE
      D1TX=0.0
      D2TX=2*(T(I-1,J,K)-T(I,J,K))/(DX*DX)
      D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
      D1TY=0.0
      D2TY=2*(T(I,J+1,K)-T(I,J,K))/(DY*DY)
      IF (K.EQ.1) GO TO 110
      IF (K.EQ.KN) GO TO 120
      GO TO 200
   20 CONTINUE
      D1TX=0.0
      D2TX=2*(T(I-1,J,K)-T(I,J,K))/(DX*DX)
      D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
      D1TY=0.0
      D2TY=2*(T(I,J-1,K)-T(I,J,K))/(DZ*DZ)
      IF (K.EQ.1) GO TO 110
      IF (K.EQ.KN) GO TO 120
      GO TO 200
  110 CONTINUE
      CT=AKT*((T(I,J,1)*TREF+TREF)-TAMB)/TREF
      CT=CT*HI(I,J)
      D2TZ=2*(T(I,J,2)-CT*DZ-T(I,J,1))/(DZ*DZ)
      GO TO 200
  120 CONTINUE
      D2TZ=2*(T(I,J,K-1)-T(I,J,K))/(DZ*DZ)
      GO TO 200
  200 CONTINUE
      TD(I,J,K)=(1.0/HI(I,J))*(-TAI*(D1HTUX+D1HTVY+HI(I,J)+D1TWZ)+TAM
     C*(D1TX*HX(I,J)+D2TX*HI(I,J)+D1TY*HY(I,J)+D2TY*HI(I,J))+(TAV/HI(I,
     C))*B3*D2TZ)*DT+T(I,J,K)
   16 CONTINUE
   18 CONTINUE
  300 CONTINUE
  100 CONTINUE
      RETURN
      END
```

```
AS1*NASA(1).TEMI4 FOR CREATED ON 5 MAY 80 AT 12:07:14
         C*****************************************************
         C   THIS SUBROUTINE COMPUTES TEMPERATURES AT INTERIOR
         C   GRID POINTS.
         C*****************************************************
         C
               SUBROUTINE TEMI4(I,J,K,IN,JN,KN,U,V,T,TD,DX,
              CCB,
              CDY,DZ,W,DT,TAI,TAH,TAV,B3,HI,HX,HY,MAR,AKT,TREF,TAMB)
               DIMENSION U(IN,JN,KN),V(IN,JN,KN),HI(IN,JN),HX(IN,JN),HY(IN,JN),
              CMAR(IN,JN),T(IN,JN,KN),TD(IN,JN,KN),W(IN,JN,KN)
               KNM1=KN-1
               DO 10 I=1,IN
               DO 10 J=1,JN
               IF(MAR(I,J).EQ.6) GO TO 100
               IF(MAR(I,J).EQ.8) GO TO 100
               IF (MAR(I,J).LT.11) GO TO 9
         100   CONTINUE
               DO 8 K=1,KN
               D1HTUX=(U(I+1,J,K)*T(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)*T(I-1,J,K)
              C*HI(I-1,J))/(2*DX)
               D1HTVY=(V(I,J+1,K)*T(I,J+1,K)*HI(I,J+1)-V(I,J-1,K)*T(I,J-1,K)*
              CHI(I,J-1))/(2*DY)
               D1TX=(T(I+1,J,K)-T(I-1,J,K))/(2*DX)
               D1TY=(T(I,J+1,K)-T(I,J-1,K))/(2*DY)
               D1TWZ=(T(I,J,K+1)*W(I,J,K+1)-T(I,J,K-1)*W(I,J,K-1))/(2*DZ)
               D2TX=(T(I+1,J,K)+T(I-1,J,K)-2*T(I,J,K))/(DX*DX)
               D2TY=(T(I,J+1,K)+T(I,J-1,K)-2*T(I,J,K))/(DY*DY)
               D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
               IF(MAR(I,J).EQ.11) GO TO 200
               D1HTUX=0.0
               D1HTVY=0.0
               D1TWZ=0.0
         200   CONTINUE
               IF (K.EQ.1) GO TO 24
               IF (K.EQ.KN) GO TO 20
               GO TO 21
         20    CONTINUE
               D1TWZ=0.0
               D2TZ=2*(T(I,J,K-1)-T(I,J,K)+CB*HI(I,J)*DZ)/(DZ*DZ)
               GO TO 21
         24    CONTINUE
               CT=AKT*((T(I,J,1)*TREF+TREF)-TAMB)/TREF
               CT=CT*HI(I,J)
               D1TWZ=0.0
               D2TZ=2*(T(I,J,2)-CT*DZ-T(I,J,1))/(DZ*DZ)
         21    CONTINUE
               TD(I,J,K)=(1.0/HI(I,J))*(-TAI*(D1HTUX+D1HTVY+HI(I,J)*D1TWZ)
              C+TAH*(D1TX*HX(I,J)+D2TX*HI(I,J)+D1TY*HY(I,J)+D2TY*HI(I,J))
              C+(TAV/HI(I,J))*B3*D2TZ)*DT+T(I,J,K)
         8     CONTINUE
         9     CONTINUE
         10    CONTINUE
               RETURN
               END
```

```
SP*NASA(1).TEQB FOR CREATED ON 5 MAY 80 AT 12:09:16
      C*********************************************************************
      C   THIS SUBROUTINE ALLOWS FOR VERTICAL MIXING AT A
      C   PARTICULAR GRID POINT.
      C*********************************************************************
      C
            SUBROUTINE TEQB(I,J,K,IN,JN,KN,T,MAR)
            DIMENSION T(IN,JN,KN),MAR(IN,JN)
            KNM1=KN-1
            DO 10 I=1,IN
            DO 10 J=1,JN
            IF (MAR(I,J).EQ.0) GO TO 9
      110   CONTINUE
            MARK=0
            DO 8 K=1,KNM1
            TX=0.01573
            TT=T(I,J,K)+TX
            IF (K.EQ.1) GO TO 7
            IF (K.EQ.KNM1) GO TO 6
            IF (TT.GE.T(I,J,K+1)) GO TO 111
            MARK=1
            AVT=(TT+T(I,J,K+1))/2.0
            T(I,J,K)=AVT-TX
            T(I,J,K+1)=AVT
            GO TO 5
      7     CONTINUE
            IF (TT.GE.T(I,J,K+1)) GO TO 111
            MARK=1
            AVT=(TT+2*T(I,J,K+1))/3.0
            T(I,J,K)=AVT-TX
            T(I,J,K+1)=AVT
            GO TO 5
      6     CONTINUE
            IF (TT.GE.T(I,J,K+1)) GO TO 111
            MARK=1
            AVT=(2*TT+T(I,J,K+1))/3.0
            T(I,J,K)=AVT-TX
            T(I,J,K+1)=AVT
            GO TO 5
      5     CONTINUE
      111   CONTINUE
      8     CONTINUE
      9     CONTINUE
      10    CONTINUE
            RETURN
            END
```

ASA*NASA(1).TPRINK FOR CREATED ON 19 NOV 79 AT 11:11:47

```
      C*********************************************************************
      C        THIS PROGRAM PRINTS THE VALUES OF T,TW,RO,ROW
      C      FOR LAKE KEHOEE
      C*********************************************************************
            SUBROUTINE TPRINK(I,J,K,IN,JN,KN,T,RO,TREF,MAR,TACTUL)
            DIMENSION T(IN,JN,KN),RO(IN,JN,KN),MAR(IN,JN),TACTUL(IN,JN,KN)
            IF(KN.LE.6) GO TO 101
            DO 10 K=1,KN
            DO 10 I=1,IN
            PRINT 11,K,I,(T(I,J,K),J=1,JN)
       10   PRINT 12,(RO(I,J,K),J=1,JN)
       11   FORMAT(/' K=',I3,3X,'I=',I3/' TEMPERATURE'/(5X,8E15.7))
       12   FORMAT(' DENSITY'/(5X,8E15.7))
      101   CONTINUE
            DO 100 K=1,KN
            DO 100 J=1,JN
            DO 100 I=1,IN
            TACTUL(I,J,K)=T(I,J,K)
            TACTUL(I,J,K)=(1.*TACTUL(I,J,K))+TREF
            IF(MAR(I,J).EQ.0) TACTUL(I,J,K)=1000000.00
      100   CONTINUE
            DO 150 K=1,KN
            WRITE(6,105) K
            DO 140 I=1,IN
            WRITE(6,106) (TACTUL(I,J,K),J=1,JN)
      140   CONTINUE
      150   CONTINUE
      105   FORMAT('1',' TEMPERATURES FOR K=',I5)
      106   FORMAT(//,22F6.2)
            RETURN
            END
```

```
NASA(1).UVT FOR CREATED ON 5 MAY 80 AT 12:13:03
      C**************************************************
      C   THIS SUBROUTINE COPUTES U AND V FOR VARIABLE DENSITY
      C**************************************************
      C
            SUBROUTINE UVT(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,D,E,H,G,DX,DY,DZ,
           CRINTX,RINTY,EUL,
           CW,
           CDT,AI,AP,AH,AV,A3,HI,HX,HY,P,MAR)
            DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
           CH(IN,JN,KN),G(IN,JN,KN),HI(IN,JN),HX(IN,JN),HY(IN,JN),P(IWN,JWN),
           CMAR(IN,JN)
            DIMENSION W(IN,JN,KN)
            DIMENSION A3(KN)
            DIMENSION RINTX(IN,JN,KN),RINTY(IN,JN,KN)
            KNM1=KN-1
            A=DT*AH*(1/(DX*DX)+1/(DY*DY))
            DO 10 I=1,IN
            DO 10 J=1,JN
            IW=I
            JW=J
            IF (MAR(I,J).LT.11) GO TO 9
            DO 8 K=2,KNM1
            D1PX=(P(IW,JW)-P(IW-1,JW)+P(IW,JW-1)-P(IW-1,JW-1))/(2*DX)
           C+EUL*RINTX(I,J,K)
            D1PY=(P(IW,JW)-P(IW,JW-1)+P(IW-1,JW)-P(IW-1,JW-1))/(2*DY)
           C+EUL*RINTY(I,J,K)
            B=DT*AV*A3(K)/(DZ*DZ)
            C=(HI(I,J)+A*HI(I,J)+B/HI(I,J))
            D1HUUX=(U(I+1,J,K)*U(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
           C*U(I-1,J,K)*HI(I-1,J))/(2*DX)
            D1HUVY=(U(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-U(I,J-1,K)
           C*V(I,J-1,K)*HI(I,J-1))/(2*DY)
            D1HUVX=(U(I+1,J,K)*V(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
           C*V(I-1,J,K)*HI(I-1,J))/(2*DX)
            D1HVVY=(V(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-V(I,J-1,K)*
           CV(I,J-1,K)*HI(I,J-1))/(2*DY)
            D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
            D1UY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
            D1VX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
            D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
            D1UWZ=(U(I,J,K+1)*W(I,J,K+1)-U(I,J,K-1)*W(I,J,K-1))/(2*DZ)
            D1VWZ=(V(I,J,K+1)*W(I,J,K+1)-V(I,J,K-1)*W(I,J,K-1))/(2*DZ)
            DDUX=(U(I+1,J,K)+U(I-1,J,K)-D(I,J,K))/(DX*DX)
            DDUY=(U(I,J+1,K)+U(I,J-1,K)-D(I,J,K))/(DY*DY)
            DDUZ=(U(I,J,K+1)+U(I,J,K-1)-D(I,J,K))/(DZ*DZ)
            DDVX=(V(I+1,J,K)+V(I-1,J,K)-E(I,J,K))/(DX*DX)
            DDVY=(V(I,J+1,K)+V(I,J-1,K)-E(I,J,K))/(DY*DY)
            DDVZ=(V(I,J,K+1)+V(I,J,K-1)-E(I,J,K))/(DZ*DZ)
            H(I,J,K)=(DT/C)*(-AI*(D1HUUX+D1HUVY+HI(I,J)*D1UWZ)-HI(I,J)*AP
           C*D1PX+AH*HI(I,J)*(DDUX+DDUY)+AH*HX(I,J)*D1UX+AH*HY(I,J)*D1UY
           C+AV*A3(K)*DDUZ/HI(I,J))+HI(I,J)*U(I,J,K)/C
            G(I,J,K)=(DT/C)*(-AI*(D1HUVX+D1HVVY+HI(I,J)*D1VWZ)-HI(I,J)*AP
           C*D1PY+AH*HI(I,J)*(DDVX+DDVY)+AH*HX(I,J)*D1VX+AH*HY(I,J)*D1VY
           C+AV*A3(K)*DDVZ/HI(I,J))+HI(I,J)*V(I,J,K)/C
      8     CONTINUE
      9     CONTINUE
      10    CONTINUE
            RETURN
            END
```

```
NASA*NASA(1).UVTOP ELT CREATED ON 19 NOV 79 AT 11:09:19
      C          THIS PROGRAM CALCULATES U AND V VELOCITIES AT THE SURFACE US'
      C                      BOUNDARY CONDITIONS
      C*****************************************************************
            SUBROUTINE UVTOP(H,G,TAUX,TAUY,I,J,K,DZ,IN,JN,KN,HI,MAR)
          * DIMENSION HI(IN,JN),MAR(IN,JN),H(IN,JN,KN),G(IN,JN,KN)
            DO 800 I=1,IN
            DO 800 J=1,JN
            IF (MAR(I,J).LT.1)) GO TO 700
            K=1
            TX=TAUX*HI(I,J)
            TY=TAUY*HI(I,J)
            H(I,J,K)=(4*H(I,J,K+1)-H(I,J,K+2)-2*DZ*TX)/3.
            G(I,J,K)=(4*G(I,J,K+1)-G(I,J,K+2)-2*DZ*TY)/3.
      700   CONTINUE
      800   CONTINUE
            RETURN
            END
```

```
NASA(1).WHATIJ FOR CREATED ON 14 MAY 74 AT 15:50:10
      C************************************************************************
      C     THIS PROGRAM CALCULATES THE VALUE OF W AT I,J FROM VALUES OF WH AT
      C************************************************************************
            SUBROUTINE WHATIJ(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
            DIMENSION WH(IWN,JWN,KN),W(IN,JN,KN)
            DIMENSION MAR(IN,JN)
            DO 3550 I=1,IN
            DO 3550 J=1,JN
            IF (MAR(I,J).LT.11) GO TO 3540
            DO 3510 K=1,KN
            IW=I
            JW=J
            W(I,J,K)=(WH(IW,JW,K)+WH(IW,JW-1,K)+WH(IW-1,JW,K)+WH(IW-1,JW-1,K))
           C/4.
      3510  CONTINUE
      3540  CONTINUE
      3550  CONTINUE
            RETURN
            END
```

```
NASA*NASA(1).WHTOP FOR CREATED ON 14 MAY 74 AT 15:50:00
C*******************************************************************
C       THIS PROGRAM SETS THE VALUE OF WH EQUAL TO ZERO AT THE SURFACE
C*******************************************************************
        SUBROUTINE WHTOP(IW,JW,IWN,JWN,KN,WH,K,MRH)
        DIMENSION WH(IWN,JWN,KN)
        DIMENSION MRH(IWN,JWN)
        DO 3300 IW=1,IWN
        DO 3300 JW=1,JWN
        IF (MRH(IW,JW).EQ.0) GO TO 3000
        WH(IW,JW,1)=0
 3000   CONTINUE
 3300   CONTINUE
        RETURN
        END
```

LISTINGS OF PLOT PROGRAMS

```
RCON*SCRCW(1).PLOTT ELT CREATED ON 25 AUG 81 AT 12:50:43
C*************************************************************
C     THIS PROGRAM PLOTS THE SURFACE ISOTHERMS FOR THE REGION
C     OF INTEREST.
C     THE FOLLOWING VARIABLES ARE READ WITH AN OPEN FORMAT :-
C     P1 = DISCHARGE VELOCITY.
C     P2 = DISCHARGE TEMPERATURE.
C     P1C= RUN NUMBER.
C     P4 = WIND SPEED (MAXIMUM).
C     P5 = CURRENT.
C     CTTOT = USED TO DIMENSIONALIZE TIME TO HOURS.
C     P6 = TOTAL SIMULATED TIME (HOURS). ** THIS IS NOT READ **
C     ALL THE OTHER VARIABLES HAVE BEEN DESCRIBED IN THE
C     USERS MANUAL.
C
C*************************************************************
C
C
      PARAMETER IN=17,JN=20,IWN=16,JWN=19,KN=5,KNM1=4
      INTEGER RGRID
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
     CWM(IWN,JWN,KN),W(IN,JN,KN),WR(IN,JN,KN),WRH(IWN,JWN,KN),
     CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN)
      DIMENSION TW(IWN,JWN,KN),RO(IN,JN,KN),PINTH(IWN,JWN),ROW(IWN,JWN,
     CKN),T(IN,JN,KN)
      DIMENSION IBUF(1000)
      READ 51,P1,P2,P10,P4,P5,CTTOT,NTIME
51    FORMAT()
      CALL PLOTS(IBUF,1000,11)
      NPLT=0
      NTIME=3
      DO 10 INJ=1,NTIME
5     CALL READ2(U,V,WH,PINTH,I,J,K,IW,JW,IN,JN,KN,
     CIWN,JWN,D,E,HX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,
     CDY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT,
     CCB,CW,A,B,C,EUL,T,TW,RO,ROW,TC,RREF,TREF,TO,
     CTAMB,TTOT)
6     CONTINUE
      CALL PLOTS(IBUF,1000,11)
      CALL PLOT(0.0,3.0,-3)
1     FORMAT()
      K=1
      P8=K
      P6=CTTOT*TTOT
      N1=0.0
      N2=0.0
      START=10.5
      DO 333 JJ=1,JN
      DO 333 II=1,IN
      HX(II,JJ)=TREF*(1.+T(II,JJ,1))
      IF(MAR(II,JJ).EQ.0)GO TO 333
      IF(HX(13,15).GT.10.5)START=11.0
333   CONTINUE
      CALL PLOT(7.0,0.0,-3)
      CALL ECHKON(HX,IN,JN,1,IN,1,JN,4.75,4.0,0.04,START,0.5,RGRID,
     CIN,JN,START,19.0,0.0,0.0,0.0,0.0,N1,N2,0.07,1.0,NPLT)
C
C*************************************************************
C     THE NEXT 66 LINES ARE FOR DRAWING THE BOUNDARIES OF THE
C     DOMAIN. THESE LINES MUST BE CHANGED FOR ANY OTHER DOMAIN.
C*************************************************************
C
C
      CALL PLOT(-0.25,-0.25,-3)
      CALL FACTOR(0.25)
      CALL PLOT(-1.0,-1.0,3)
      CALL PLOT(12.0,1.0,3)
      CALL PLOT(12.0,6.0,2)
      CALL PLOT(13.0,6.0,2)
      CALL PLOT(13.0,8.0,2)
      CALL PLOT(17.0,8.0,2)
      CALL PLOT(17.0,10.0,2)
      CALL PLOT(14.0,10.0,2)
      CALL PLOT(14.0,11.0,2)
      CALL PLOT(12.0,11.0,2)
      CALL PLOT(12.0,13.0,2)
      CALL PLOT(14.0,13.0,2)
      CALL PLOT(14.0,14.0,2)
      CALL PLOT(16.0,14.0,2)
```

```
      CALL PLOT(16.0,18.0,2)
      CALL PLOT(14.0,18.0,2)
      CALL PLOT(14.0,17.0,2)
      CALL PLOT(11.0,17.0,2)
      CALL PLOT(11.0,16.0,2)
      CALL PLOT(9.0,16.0,2)
      CALL PLOT(9.0,16.0,2)
      CALL PLOT(8.0,16.0,2)
      CALL PLOT(8.0,17.0,2)
      CALL PLOT(10.0,17.0,2)
      CALL PLOT(10.0,19.0,2)
      CALL PLOT(8.0,19.0,2)
      CALL PLOT(8.0,20.0,2)
      CALL PLOT(6.0,20.0,2)
      CALL PLOT(6.0,17.0,2)
      CALL PLOT(3.0,17.0,2)
      CALL PLOT(3.0,20.0,2)
      CALL PLOT(1.0,20.0,2)
      CALL PLOT(1.0,17.0,2)
      CALL PLOT(1.0,17.0,3)
      CALL PLOT(1.0,15.0,3)
      CALL PLOT(1.0,15.0,2)
      CALL PLOT(1.0,13.0,2)
      CALL PLOT(1.0,13.0,3)
      CALL PLOT(1.0,11.0,3)
      CALL PLOT(1.0,11.0,2)
      CALL PLOT(1.0,9.0,2)
      CALL PLOT(1.0,9.0,3)
      CALL PLOT(1.0,7.0,3)
      CALL PLOT(1.0,7.0,2)
      CALL PLOT(3.0,7.0,2)
      CALL PLOT(3.0,5.0,2)
      CALL PLOT(5.0,5.0,2)
      CALL PLOT(5.0,6.0,2)
      CALL PLOT(7.0,6.0,2)
      CALL PLOT(7.0,3.0,2)
      CALL PLOT(10.0,3.0,2)
      CALL PLOT(10.0,1.0,2)
      CALL PLOT(10.0,1.0,3)
      CALL FACTOR(0.25)
      CALL PLOT(0.0,0.0,3)
      CALL FACTOR(1.0)
      CALL PLOT(0.0,-0.5,-3)
      CALL PLOT(-1.0,-1.0,-3)
      CALL PLOT(0.0,1.1,3)
      CALL PLOT(6.0,1.1,2)
      CALL PLOT(6.0,9.0,2)
      CALL PLOT(0.0,9.0,2)
      CALL PLOT(0.0,1.1,2)
      CALL PLOT(0.0,0.0,3)
C
C*****************************************************************
C   THE NEXT 25 LINES ARE FOR WRITING THE CAPTIONS OF THE
C   PLOTS. THE SPECIFIC USER MUST SCRUTINIZE THESE LINES
C   AND MAKE NECESSARY CHANGES.
C*****************************************************************
C
C
      CALL SYMBOL(0.0,0.6,0.14,23HFIG     ISOTHERMS AT K= ,0.0,23)
      CALL NUMBER(999.,999.,0.14,P8,0.0,0)
      CALL SYMBOL(0.0,0.3,0.14,36H           LAKE KEOWEE-(RIGID-LID MODEL),0.
     C0,36)
      IF(P6.GE.25.0)GO TO 22
      GO TO 23
22    CALL SYMBOL(1.0,0.0,0.14,28HSIMULATIONS FOR FEB. 26 1979,
     C0.0,28)
      GO TO 123
23    CALL  SYMBOL(1.0,0.0,0.14,28HSIMULATIONS FOR FEB. 27 1979,
     C0.0,28)
123   CONTINUE
      CALL SYMBOL(1.5,8.7,0.1,12HRUN NO: L00 ,0.0,12)
      CALL NUMBER(999.,999.,0.1,P10,0.0,0)
      CALL SYMBOL(1.5,8.5,0.1,33HDISCHARGE VELOCITY   :     CM/SEC,0.0,3
     C3)
      CALL NUMBER(3.8,8.5,0.1,P1,0.0,+2)
      CALL SYMBOL(1.5,8.3,0.1,29HDISCHARGE TEMPERATURE:     C,0.0,29)
      CALL SYMBOL(4.2,8.4,0.07,1HO,0.0,1)
      CALL NUMBER(3.8,8.3,0.1,P2,0.0,+1)
      CALL SYMBOL(1.5,8.1,0.1,32HWIND SPEED (MAX)     :     M/SEC,0.0,32
```

```
     C)
      CALL NUMBER(3.8,8.1,0.1,P4,0.0,.2)
      CALL SYMBOL(1.5,7.9,0.1,33HCURRENT(UUCASEL FLOW):      CM/SEC,0.0,3
     C3)
      CALL NUMBER(3.8,7.9,0.1,P5,0.0,+1)
      CALL SYMBOL(1.5,7.7,0.1,31HTOTAL SIMULATED TIME :      HRS,0.0,31)
      CALL NUMBER(3.8,7.7,0.1,P6,0.0,+2)
      CALL SYMBOL(1.5,7.2,0.1,20HLENGTH SCALE(METERS),0.0,20)
      CALL AXIS(4.1,7.2,1H ,+0,1.0,0.,0.,610.)
      CALL SYMBOL(1.5,6.7,0.1,22HVELOCITY SCALE(CM/SEC),0.0,22)
      CALL AXIS(4.1,6.7,1H ,+0,1.0,0.,0.,12.)
      CALL SYMBOL(0.2,6.7,0.21,2H N,45.0,2)
      CALL PLOT(8.C,-15.0,-3)
   10 CONTINUE
      END
@PRT,S S.PL.V
```

```
*COW*SLMON(1).PLUV ELT CREATED ON 7 MAY 80 AT 18:39:52
      C*****************************************************************
      C   THIS PROGRAM PLOTS THE U - V VELOCITIES FOR THE REGION
      C   OF INTEREST.
      C   THE FOLLOWING VARIABLES ARE READ WITH AN OPEN FORMAT :-
      C   P1 = DISCHARGE VELOCITY.
      C   P2 = DISCHARGE TEMPERATURE.
      C   P10= RUN NUMBER.
      C   P4 = WIND SPEED (MAXIMUM).
      C   P5 = CURRENT.
      C   CTTOT = USED TO DIMENSIONALIZE TIME TO HOURS.
      C   NTIME = THE NUMBER OF HOURS TO BE PLOTTED.
      C   P6 = TOTAL SIMULATED TIME (HOURS). ** THIS IS NOT READ **
      C   ALL THE OTHER VARIABLES HAVE BEEN DESCRIBED IN THE
      C   USERS MANUAL.
      C
      C*****************************************************************
      C
      C
      C        PLOTS U AND V ON CONSTANT DEPTH SECTIONS (FEBRUARY 1979 MISSION)
            PARAMETER IN=17,JN=20,IWN=16,JWN=19,KN=5,KNM1=4
            DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
           CWH(IWN,JWN,KN),W(IN,JN,KN),WR(IN,JN,KN),WRH(IWN,JWN,KN),
           CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN)
            DIMENSION TW(IWN,JWN,KN),RO(IN,JN,KN),PINTH(IWN,JWN),ROW(IWN,JWN,
           CKN),T(IN,JN,KN)
            DIMENSION IBUF(1000)
            READ 51,P1,P2,P10,P4,P5,CTTOT,NTIME
   51       FORMAT()
            USCALE=10.0
            VSCALE=10.0
            NTIME=3
             ARMIN=0.04
             ARMAX=0.15
             DO 11 INEW=1,NTIME
            CALL READ2(U,V,WH,PINTH,I,J,K,IW,JW,IN,JN,KN,
           CIWN,JWN,D,E,HX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,
           CDY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT,
           CCB,CA,A,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,
           CTAMB,TTOT)
    6       CONTINUE
            DO 99 I=1,IN
            DO 99 J=1,JN
            HI(I,J)=1.0
   99       CONTINUE
            CALL PLOTS(IBUF,1000,11)
            CALL PLOT(0.0,2.0,-3)
    1       FORMAT ()
            DO 10 K=1,KN
            P5=K
            P6=CTTOT*TTOT
            CALL FACTOR(0.25)
            IF(K.GT.1) GO TO 20
            DO 30 I=1,IN
            DO 30 J=1,JN
            IF (MAR(I,J).EQ.0) GO TO 35
            AI=(I-1)*1.0
            AJ=(J-1)*1.0
            AAI=AI+U(I,J,K)*USCALE
            AAJ=AJ+V(I,J,K)*VSCALE
            YW=0.2*SQRT((AAI-AI)**2+(AAJ-AJ)**2)
            YW=AMAX1(ARMIN/0.25,AMIN1(YW,ARMAX/0.25))
            CALL AROHD(AI,AJ,AAI,AAJ,YW,0.0,12)
   35       CONTINUE
   30       CONTINUE
            GO TO 100
   20       CONTINUE
            DEPTH=(1.0/KNM1)*(K-1)
            DO 40 I=1,IN
            DO 40 J=1,JN
            IF(HI(I,J).GT.DEPTH) GO TO 45
            GO TO 50
   45       CONTINUE
            DDZ=HI(I,J)/KNM1
            LD1=(DEPTH/HI(I,J))*KNM1
            IF(LD1.EQ.0) GO TO 55
            LD2=LD1+1
            LD3=LD1+2
            DIFF=(DEPTH-LD1*DDZ)
```

```
C         COEFFS OF SECOND DEGREE FIT
          U1=U(I,J,LD1)
          U2=U(I,J,LD2)
          U3=U(I,J,LD3)
          V1=V(I,J,LD1)
          V2=V(I,J,LD2)
          V3=V(I,J,LD3)
          AU=(U3-2*U2+U1)/(2*DDZ*DDZ)
          BU=(4*U2-3*U1-U3)/(2*DDZ)
          CU=U1
          AV=(V3-2*V2+V1)/(2*DDZ*DDZ)
          BV=(4*V2-3*V1-V3)/(2*DDZ)
          CV=V1
          AZ=DDZ*DIFF
          UDEPTH=AU*AZ*AZ+BU*AZ+CU
          VDEPTH=AV*AZ*AZ+BV*AZ+CV
          GO TO 60
65        CONTINUE
          AZ=DEPTH
          AU=(U(I,J,3)-2*U(I,J,2)+U(I,J,1))/(2*DDZ*DDZ)
          BU=(4*U(I,J,2)-3*U(I,J,1)-U(I,J,3))/(2*DDZ)
          CU=U(I,J,1)
          UDEPTH=AU*AZ*AZ+BU*AZ+CU
          AV=(V(I,J,3)-2*V(I,J,2)+V(I,J,1))/(2*DDZ*DDZ)
          BV=(4*V(I,J,2)-3*V(I,J,1)-V(I,J,3))/(2*DDZ)
          CV=V(I,J,1)
          VDEPTH=AV*AZ*AZ+BV*AZ+CV
60        CONTINUE
          AI=(I-1)*1.0
          AJ=(J-1)*1.0
          AAI=AI+UDEPTH*USCALE
          AAJ=AJ+VDEPTH*VSCALE
          YW=0.2*SQRT((AAI-AI)**2+(AAJ-AJ)**2)
          YW=AMAX1(ARMIN/0.25,AMIN1(YW,ARMAX/0.25))
          CALL AROHD(AI,AJ,AAI,AAJ,YW,0.0,12)
50        CONTINUE
40        CONTINUE
100       CONTINUE
          CALL PLOT(-1.0,-1.0,-3)
C
C*************************************************************
C   THE NEXT 49 LINES ARE FOR DRAWING THE BOUNDARIES OF THE
C   DOMAIN. THESE LINES MUST BE CHANGED FOR ANY OTHER DOMAIN.
C*************************************************************
C
C
          CALL PLOT(12.0,1.0,3)
          CALL PLOT(12.0,6.0,2)
          CALL PLOT(13.0,6.0,2)
          CALL PLOT(13.0,8.0,2)
          CALL PLOT(17.0,8.0,2)
          CALL PLOT(17.0,10.0,2)
          CALL PLOT(14.0,10.0,2)
          CALL PLOT(14.0,11.0,2)
          CALL PLOT(12.0,11.0,2)
          CALL PLOT(12.0,13.0,2)
          CALL PLOT(14.0,13.0,2)
          CALL PLOT(14.0,14.0,2)
          CALL PLOT(16.0,14.0,2)
          CALL PLOT(16.0,18.0,2)
          CALL PLOT(14.0,18.0,2)
          CALL PLOT(14.0,17.0,2)
          CALL PLOT(11.0,17.0,2)
          CALL PLOT(11.0,15.0,2)
          CALL PLOT(9.0,16.0,2)
          CALL PLOT(5.0,16.0,2)
          CALL PLOT(8.0,16.0,2)
          CALL PLOT(8.0,17.0,2)
          CALL PLOT(10.0,17.0,2)
          CALL PLOT(10.0,19.0,2)
          CALL PLOT(8.0,19.0,2)
          CALL PLOT(8.0,20.0,2)
          CALL PLOT(6.0,20.0,2)
          CALL PLOT(6.0,17.0,2)
          CALL PLOT(3.0,17.0,2)
          CALL PLOT(3.0,20.0,2)
          CALL PLOT(1.0,20.0,2)
          CALL PLOT(1.0,7.0,2)
          CALL PLOT(3.0,7.0,2)
```

```
      CALL PLOT(3.0,5.0,2)
      CALL PLOT(5.0,5.0,2)
      CALL PLOT(5.0,6.0,2)
      CALL PLOT(7.0,6.0,2)
      CALL PLOT(7.0,3.0,2)
      CALL PLOT(10.0,3.0,2)
      CALL PLOT(10.0,1.0,2)
      CALL PLOT(10.0,1.0,3)
      CALL FACTOR(1.0)
      CALL PLOT(-1.0,-1.0,-3)
      CALL PLOT(0.0,1.1,3)
      CALL PLOT(6.0,1.1,2)
      CALL PLOT(6.0,9.0,2)
      CALL PLOT(0.0,9.0,2)
      CALL PLOT(0.0,1.1,2)
      CALL PLOT(0.0,0.0,3)
C
C*********************************************************
C   THE NEXT 25 LINES ARE FOR WRITING THE CAPTIONS OF THE
C   PLOTS. THE SPECIFIC USER MUST SCRUTINIZE THESE LINES
C   AND MAKE NECESSARY CHANGES.
C*********************************************************
C
C
      CALL SYMBOL(0.0,0.6,0.14,24HFIG     VELOCITIES AT K= ,0.0,24)
      CALL NUMBER(999.,999.,0.14,P8,0.0,0)
      CALL SYMBOL(0.0,0.3,0.14,36H            LAKE KEOWEE-(RIGID-LID MODEL),0.
     C0,36)
      IF(P6.GE.25.0)GO TO 22
      GO TO 23
22    CALL SYMBOL(1.0,0.0,0.14,28HSIMULATIONS FOR FEB. 28 1979,
     C0.0,28)
      GO TO 123
23    CALL  SYMBOL(1.0,0.0,0.14,28HSIMULATIONS FOR FEB. 27 1979,
     C0.0,28)
123   CONTINUE
      CALL SYMBOL(1.5,8.7,0.1,12HRUN NO: L00 ,0.0,12)
      CALL NUMBER(999.,999.,0.1,P10,0.0,0)
      CALL SYMBOL(1.5,8.5,0.1,33HDISCHARGE VELOCITY    :     CM/SEC,0.0,3
     C3)
      CALL NUMBER(3.8,8.5,0.1,P1,0.0,+2)
      CALL SYMBOL(1.5,8.3,0.1,29HDISCHARGE TEMPERATURE:      C,0.0,29)
      CALL SYMBOL(4.2,8.4,0.07,1HO,0.0,1)
      CALL NUMBER(3.8,8.3,0.1,P2,0.0,+1)
      CALL SYMBOL(1.5,8.1,0.1,32HWIND SPEED (MAX)      :     M/SEC,0.0,32
     C)
      CALL NUMBER(3.8,8.1,0.1,P4,0.0,+2)
      CALL SYMBOL(1.5,7.9,0.1,33HCURRENT(JOCASSE FLOW):     CM/SEC,0.0,3
     C3)
      CALL NUMBER(3.8,7.9,0.1,P5,0.0,+1)
      CALL SYMBOL(1.5,7.7,0.1,31HTOTAL SIMULATED TIME :      HRS,0.0,31)
      CALL NUMBER(3.8,7.7,0.1,P6,0.0,+2)
      CALL SYMBOL(1.5,7.2,0.1,20HLENGTH SCALE(METERS),0.0,20)
      CALL AXIS(4.1,7.2,1H ,+0,1.0,0.,L.,610.)
      CALL SYMBOL(1.5,6.7,0.1,22HVELOCITY SCALE(CM/SEC),0.0,22)
      CALL AXIS(4.1,6.7,1H ,+0,1.0,0.,0.,12.)
      CALL SYMBOL(0.2,0.7,0.21,2H N,45.0,2)
      CALL PLOT(10.0,1.25,-3)
10    CONTINUE
      CALL PLOT(0.0,-3.0,-3)
11    CONTINUE
      CALL PLOT(10.0,-2.0,-3)
      END
@PRT,S S.PLUM
```

```
ΓC=*SUMON(1).PLUW ELT CREATED ON 7 MAY 80 AT 19:00:29
      C*************************************************************
      C   THIS PROGRAM PLOTS THE V - W VELOCITIES FOR THE REGION
      C   OF INTEREST.
      C   THE FOLLOWING VARIABLES ARE READ WITH AN OPEN FORMAT :-
      C   P1 = DISCHARGE VELOCITY.
      C   P2 = DISCHARGE TEMPERATURE.
      C   P10= RUN NUMBER.
      C   P4 = WIND SPEED (MAXIMUM).
      C   P5 = CURRENT.
      C   CTTOT = USED TO DIMENSIONALIZE TIME TO HOURS.
      C   NTIME = TOTAL NUMBER OF HOURS SIMULATED
      C   P6 = TOTAL SIMULATED TIME (HOURS). ** THIS IS NOT READ **
      C   ALL THE OTHER VARIABLES HAVE BEEN DESCRIBED IN THE
      C   USERS MANUAL.
      C
      C*************************************************************
      C
      C
            PARAMETER IN=17,JN=20,KN=5,IWN=16,JWN=19,KNM1=4
            DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
           CWH(IWN,JWN,KN),W(IN,JN,KN),WR(IN,JN,KN),WRH(IWN,JWN,KN),
           CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN)
            DIMENSION TW(IWN,JWN,KN),RO(IN,JN,KN),PINTH(IWN,JWN),ROW(IWN,JWN,
           CKN),T(IN,JN,KN),IBCD(6)
            DIMENSION IBUF(1000)
            USCALE=10.0
            VSCALE=10.0
            WSCALE=20.0
            HBYL=0.33
            NTIME=3
            DO 11 INEW=1,NTIME
            CALL READ2(U,V,WH,PINTH,I,J,K,IW,JW,IN,JN,KN,
           CIWN,JWN,D,E,HX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,
           CDY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT,CB,CW,
           CA,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,
           CTAMB,TTOT)
      11    CONTINUE
            CALL PLOTS(IBUF,1000,11)
            READ 51,P1,P2,P10,P4,P5,CTTOT,NTIME
      51    FORMAT()
            ARMIN=0.04
            ARMAX=0.15
      1     FORMAT ()
            DO 10 I=7,13
            CALL FACTOR(0.25)
            CALL PLOT(0.0,16.0,-3)
            DO 20 J=1,JN
            P8=I
            P6=CTTOT*TTOT
            IF (MAR(I,J).LT.11) GO TO 20
            AJ=(J-1)*1.0
            DO 15 K=1,KNM1
            AK=-(K-1)*HI(I,J)
            AAJ=AJ+V(I,J,K)*VSCALE
            AAK=AK-W(I,J,K)*WSCALE*HBYL
            YW=0.2*SQRT((AAJ-AJ)**2+(AAK-AK)**2)
            YW=AMAX1(ARMIN/0.25,AMIN1(YW,ARMAX/0.25))
            CALL AROHD(AJ,AK,AAJ,AAK,YW,0.0,12)
      15    CONTINUE
      20    CONTINUE
      C     DRAWS BOTTOM SURFACE
            NN=0
            DO 30 J=1,JN
            IF (MAR(I,J).EQ.0) GO TO 32
            NN=NN+1
            IF (NN.GT.1) GO TO 33
            AAJ=(J-1)*1.0
            AAK=-HI(I,J)*KNM1
            CALL PLOT(AAJ,0.0,3)
            CALL PLOT(AAJ,AAK,2)
            GO TO 32
      33    CONTINUE
            AAJ=(J-1)*1.0
            AAK=-HI(I,J)*KNM1
            CALL PLOT(AAJ,AAK,2)
            JD=J
            AJD=(JD-1)*1.0
      32    CONTINUE
```

```
30      CONTINUE
        CALL PLOT(AJD,0.0,3)
        AAK=-HI(I,JD)+KNMI
        CALL PLOT(AJD,AAK,2)
C
C*********************************************************************
C   THE NEXT 07 LINES ARE FOR DRAWING THE BOUNDARIES OF THE
C   DOMAIN. THESE LINES MUST BE CHANGED FOR ANY OTHER DOMAIN.
C*********************************************************************
C
C
        CALL FACTOR(1.0)
        CALL PLOT(-0.5,-2.0,-3)
        CALL PLOT(6.0,0.0,2)
        CALL PLOT(6.0,8.0,2)
        CALL PLOT(0.0,8.0,2)
        CALL PLOT(0.0,0.0,2)
        CALL PLOT(0.0,-2.0,-3)
C
C*********************************************************************
C   THE NEXT 25 LINES ARE FOR WRITING THE CAPTIONS OF THE
C   PLOTS. THE SPECIFIC USER MUST SCRUTINIZE THESE LINES
C   AND MAKE NECESSARY CHANGES.
C*********************************************************************
C
C
        CALL SYMBOL(0.0,1.5,0.14,24HFIG     VELOCITIES AT I= ,0.0,24)
        CALL NUMBER(3.4,1.5,0.14,P8,0.0,0)
        CALL SYMBOL(0.0,1.2,0.14,36H        LAKE KEOWEE-(RIGID-LID MODEL),0.
       C0,36)
        IF(P6.GE.25.0)GO TO 22
        GO TO 23
22      CALL SYMBOL(1.0,0.9,0.14,28HSIMULATIONS FOR FEB. 26 1979,
       C0.0,28)
        GO TO 123
23      CALL  SYMBOL(1.0,0.9,0.14,28HSIMULATIONS FOR FEB. 27 1979,
       C0.0,28)
123     CONTINUE
        CALL SYMBOL(1.5,9.5,0.1,12HRUN NO: L00 ,0.0,12)
        CALL NUMBER(2.6,9.5,0.1,P10,0.0,0)
        CALL SYMBOL(1.5,9.3,0.1,33HDISCHARGE VELOCITY   :     CM/SEC,0.0,3
       C3)
        CALL NUMBER(3.8,9.3,0.1,P1,0.0,+2)
        CALL SYMBOL(1.5,9.1,0.1,29HDISCHARGE TEMPERATURE:      C,0.0,29)
        CALL SYMBOL(4.2,9.2,0.07,1HO,0.0,1)
        CALL NUMBER(3.8,9.1,0.1,P2,0.0,+1)
        CALL SYMBOL(1.5,8.9,0.1,32HWIND SPEED (MAX)     :     M/SEC,0.0,32
       C)
        CALL NUMBER(3.8,8.9,0.1,P4,0.0,+2)
        CALL SYMBOL(1.5,8.7,0.1,33HCURRENT(JOCASSE FLOW):     CM/SEC,0.0,3
       C3)
        CALL NUMBER(3.8,8.7,0.1,P5,0.0,+1)
        CALL SYMBOL(1.5,8.5,0.1,31HTOTAL SIMULATED TIME :     HRS,0.0,31)
        CALL NUMBER(3.8,8.5,0.1,P6,0.0,+2)
        CALL SYMBOL(1.5,8.0,0.1,19HSCALES (HORIZONTAL),0.0,19)
        CALL SYMBOL(1.5,7.5,0.1,20HLENGTH SCALE(METERS),0.0,20)
        CALL AXIS(4.1,7.5,1H ,+0,1.0,0.,0.,610.)
        CALL SYMBOL(1.5,7.0,0.1,22HVELOCITY SCALE(CM/SEC),0.0,22)
        CALL AXIS(4.1,7.0,1H ,+0,1.0,0.,0.,12.)
        CALL SYMBOL(1.5,6.5,0.1,17HSCALES (VERTICAL),0.0,17)
        CALL SYMBOL(1.5,6.0,0.1,20HLENGTH SCALE(METERS),0.0,20)
        CALL AXIS(4.1,6.0,1H ,+0,1.0,0.,0.,20.)
        CALL SYMBOL(1.5,5.5,0.1,22HVELOCITY SCALE(CM/SEC),0.0,22)
        CALL AXIS(4.1,5.5,1H ,+0,1.0,0.,0.,6.)
        CALL PLOT(10.0,0.0,-3)
10      CONTINUE
11      CONTINUE
        END
@PRT,S S.PLV.
```

```
RECW*LUNOM(1).PLVWELT CREATED ON 22 AUG 80 AT 16:19:19
      C**********************************************************
      C   THIS PROGRAM PLOTS THE V - W VELOCITIES FOR THE REGION
      C   OF INTEREST.
      C   THE FOLLOWING VARIABLES ARE READ WITH AN OPEN FORMAT :-
      C   P1 = DISCHARGE VELOCITY.
      C   P2 = DISCHARGE TEMPERATURE.
      C   P10= RUN NUMBER.
      C   P4 = WIND SPEED (MAXIMUM).
      C   P5 = CURRENT.
      C   CTTOT = USED TO DIMENSIONALIZE TIME TO HOURS.
      C   NTIME = TOTAL NUMBER OF HOURS SIMULATED.
      C   P6 = TOTAL SIMULATED TIME (HOURS). ** THIS IS NOT READ **
      C   ALL THE OTHER VARIABLES HAVE BEEN DESCRIBED IN THE
      C   USERS MANUAL.
      C
      C**********************************************************
      C
      C
            PARAMETER  IN=17,JN=20,IWN=16,JWN=19,KN=5,KNM1=4
             DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
           CWH(IWN,JWN,KN),PINTH(IWN,JWN)
             DIMENSION HX(IN,JN),HY(IN,JN),HT(IN,JN),MAR(IN,JN),MRH(IWN,JWN),
           CW(IN,JN,KN),WR(IN,JN,KN),WRH(IN,JN,KN)
             DIMENSION T(IN,JN,KN),RO(IN,JN,KN),TW(IWN,JWN,
           CKN),ROW(IWN,JWN,KN)
             DIMENSION IBUF(1000)
             USCALE=10.0
             VSCALE=10.0
             WSCALE=20.0
             HBYL=0.33
             NTIME=3
             DO 11 INEW=1,NTIME
             CALL READ2(U,V,WH,PINTH,I,J,K,IW,JW,IN,JN,KN,
           CIWN,JWN,D,E,HX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,
           CDY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT,CB,CW,
           CA,B,C,CUL,T,TW,RO,ROW,TE,RREF,TREF,TO,
           CTAMB,TTOT)
      0      CONTINUE
             CALL PLOTS(IBUF,1000,11)
             READ 51,P1,P2,P10,P4,P5,CTTOT,NTIME
      51     FORMAT()
             ARMIN=0.08
             ARMAX=0.15
      1      FORMAT ()
             DO 10 J=1,8
             CALL FACTOR(0.25)
             CALL PLOT(0.0,16.0,-3)
             DO 20 I=1,IN
             P8=J
             P6=CTTOT*TTOT
             IF (MAR(I,J).LT.11) GO TO 20
             AI=(I-1)*1.0
             DO 30 K=1,KNM1
             AK=-(K-1)*HI(I,J)
             AAI=AI+U(I,J,K)*USCALE
             W(I,1,K)=0.0
             AAK=AK-W(I,J,K)*WSCALE*HBYL
             YW=0.2*SQRT((AAI-AI)**2+(AAK-AK)**2)
             YW=AMAX1(ARMIN/0.25,AMIN1(YW,ARMAX/0.25))
             CALL AROHD(AI,AK,AAI,AAK,YW,0.0,12)
      30     CONTINUE
      20     CONTINUE
      C      DRAWS BOTTOM SURFACE
             NN=0
             DO 35 I=1,IN
             IF(MAR(I,J).EQ.0) GO TO 40
             NN=NN+1
             IF (NN.GT.1) GO TO 33
             AAI=(I-1)*1.0
             CALL PLOT(AAI,0.0,3)
              AAK=-HI(I,J)*KNM1
             CALL PLOT(AAI,AAK,2)
             GO TO 40
      33     CONTINUE
             AAI=(I-1)*1.0
             AAK=-HI(I,J)*KNM1
             CALL PLOT(AAI,AAK,2)
             ID=I
```

```
            AID=(ID-1)*1.0
   40       CONTINUE
   35       CONTINUE
            CALL PLOT(AID,0.0,3)
            AAK=-H1(I,J)*KWM1
            CALL PLOT(AID,AAK,2)
      C
      C***********************************************************
      C   THE NEXT 07 LINES ARE FOR DRAWING THE BOUNDARIES OF THE
      C   DOMAIN. THESE LINES MUST BE CHANGED FOR ANY OTHER DOMAIN.
      C***********************************************************
      C
      C
            CALL FACTOR(1.0)
            CALL PLOT(-0.5,-2.0,-3)
            CALL PLOT(6.0,0.0,2)
            CALL PLOT(6.0,8.0,2)
            CALL PLOT(0.0,8.0,2)
            CALL PLOT(0.0,0.0,2)
            CALL PLOT(0.0,-2.0,-3)
      C
      C***********************************************************
      C   THE NEXT 25 LINES ARE FOR WRITING THE CAPTIONS OF THE
      C   PLOTS. THE SPECIFIC USER MUST SCRUTINIZE THESE LINES
      C   AND MAKE NECESSARY CHANGES.
      C***********************************************************
      C
      C
            CALL SYMBOL(0.0,1.5,0.14,24HFIG      VELOCITIES AT J= ,0.0,24)
            CALL NUMBER(3.4,1.5,0.14,P8,0.0,0)
            CALL SYMBOL(0.0,1.2,0.14,36H         LAKE KEOWEE-(RIGID-LID MODEL),0.
           C0,36)
            IF(P6.GE.25.0)GO TO 22
            GO TO 23
   22       CALL SYMBOL(1.0,0.9,0.14,26HSIMULATIONS FOR FEB. 26 1979,
           C0.0,28)
            GO TO 123
   23       CALL  SYMBOL(1.0,0.9,0.14,28HSIMULATIONS FOR FEB. 27 1979,
           C0.0,28)
  123       CONTINUE
            CALL SYMBOL(1.5,9.5,0.1,12HRUN NO: L00 ,0.0,12)
            CALL NUMBER(2.6,9.5,0.1,P10,0.0,0)
            CALL SYMBOL(1.5,9.3,0.1,33HDISCHARGE VELOCITY   :     CM/SEC,0.0,3
           C3)
            CALL NUMBER(3.8,9.3,0.1,P1,0.0,+2)
            CALL SYMBOL(1.5,9.1,0.1,29HDISCHARGE TEMPERATURE:      C,0.0,29)
            CALL SYMBOL(4.2,9.2,0.07,1HO,0.0,1)
            CALL NUMBER(3.8,9.1,0.1,P2,0.0,+1)
            CALL SYMBOL(1.5,8.9,0.1,32HWIND SPEED (MAX)     :     M/SEC,0.0,32
           C)
            CALL NUMBER(3.8,8.9,0.1,P4,0.0,+2)
            CALL SYMBOL(1.5,8.7,0.1,33HCURRENT JOCASSEE FLOW:     CM/SEC,0.0,3
           C3)
            CALL NUMBER(3.8,8.7,0.1,P5,0.0,+1)
            CALL SYMBOL(1.5,8.5,0.1,31HTOTAL SIMULATED TIME :     HRS,0.0,31)
            CALL NUMBER(3.8,8.5,0.1,P6,0.0,+2)
            CALL SYMBOL(1.5,8.0,0.1,19HSCALES (HORIZONTAL),0.0,19)
            CALL SYMBOL(1.5,7.5,0.1,20HLENGTH SCALE(METERS),0.0,20)
            CALL AXIS(4.1,7.5,1H ,+0,1.0,0.,0.,610.)
            CALL SYMBOL(1.5,7.0,0.1,22HVELOCITY SCALE(CM/SEC),0.0,22)
            CALL AXIS(4.1,7.0,1H ,+0,1.0,0.,0.,12.)
            CALL SYMBOL(1.5,6.5,0.1,17HSCALES (VERTICAL),0.0,17)
            CALL SYMBOL(1.5,6.0,0.1,20HLENGTH SCALE(METERS),0.0,20)
            CALL AXIS(4.1,6.0,1H ,+0,1.0,0.,0.,20.)
            CALL SYMBOL(1.5,5.5,0.1,22HVELOCITY SCALE(CM/SEC),0.0,22)
            CALL AXIS(4.1,5.5,1H ,+0,1.0,0.,0.,6.)
            CALL PLOT(10.0,0.0,-3)
   10       CONTINUE
   11       CONTINUE
            END
@PRT,S S.ECHKON
```

```
..EC..SUMON(1).ECHKON ELT CREATED ON 24 JAN 80 AT 09.02.53.
      SUBROUTINE ECHKON
C
C          THIS IS ENTRY SUBROUTINE FOR NHC CONTOURING PROGRAM
C                 [CALCOMP OR MILGO TYPE PLOTTER]
C
C
C     THE COMPLETE PACKAGE CONSISTS OF 3 SUBROUTINES, ECHKON, CONLIN, AND END
C     ALL 3 ARE CATALOGUED TOGETHER IN THE UM 360/65 UNDER MODULE NAME ECHKON
C        AND DECKS ARE NOT NEEDED.
C
C     ANY RECTANGULAR GRIDDED SCALAR FIELD CAN BE CONTOURED ON MILGO
C     OR CALCOMP TYPE PLOTTER BY SETTING UP PROPER CALLING ARGUMENTS AND
C        PROCEDURES AS INDICATED BELOW AND THEN CALLING ECHKON.
C
C           ------CALLING STATEMENT IS AS FOLLOWS------
C
C     CALL ECHKON[HH,IN1,IN2,NEX1,NEX2,NEY1,NEY2,HI,WID,PLTINC,SAMCON,
C    2CONINT,RGRID,IN3,IN4,ZLIT,ZBIG,ANORTH,ASOUTH,AEAST,AWEST,NDASHD,
C    3NDASHU,XLABEL,SMOOTH,IRECCY]
C
C              ---DESCRIPTION OF CALLING ARGUMENTS---
C
C     HH IS ARRAY CONTAINING GRID DATA TO BE CONTOURED. ITS DIMENSIONS
C        ARE IN1 AND IN2. DIMENSION HH[IN1,IN2]. POINT 1,1 IS LOWER LEFT
C        CORNER OF GRID. IN1 IS DIMENSION IN X DIRECTION AND IN2 IS
C        DIMENSION IN Y DIRECTION.
C    [X INCREASES FROM WEST TO EAST AND Y INCREASES FROM SOUTH TO NORTH]
C
C     NEX1, NEX2, NEY1, AND NEY2 DETERMINE THE PORTION OF HH GRID TO
C        BE USED. NEX1 AND NEX2 ARE THE FIRST[LEFTMOST] AND LAST[RIGHTMOST]
C        COLUMNS TO BE USED. NEY1 AND NEY2 ARE THE FIRST[BOTTOM] AND LAST[TOP
C        ROWS TO BE USED.  [THUS ANY SECTION OF HH CAN BE USED]
C           FOR FULL GRID---
C                           NEX1 > 1
C                           NEX2 > IN1
C                           NEY1 > 1
C                           NEY2 > IN2
C
C     HI IS HEIGHT IN INCHES OF CONTOUR MAP BETWEEN LIMITS NEY1 AND NEY2
C     WID IS WIDTH IN INCHES OF CONTOUR MAP BETWEEN LIMITS NEX1 AND NEX2
C
C     PLTINC IS STRAIGHT LINE PLOT INCREMENT IN INCHES TO BE USED
C        ALONG CONTOUR. GOOD VALUE IS .04, BUT CAN BE VARIED UP OR DOWN.
C      SINCE LARGER VALUES CAUSE PROGRAM TO RUN A LITTLE FASTER, IDEAL VALUE
C       IS LARGEST THAT WILL STILL GIVE SMOOTH LOOKING CURVES.
C       --DO SOME EXPERIMENTING WITH IT. START WITH .03 OR .04 AND INCREASE.
C
C     SAMCON IS ANY SAMPLE CONTOUR VALUE. IT IS USED AS A STARTING POINT
C        FOR COUNTING UP AND DOWN TO GET OTHER CONTOUR VALUES.
C
C     CONINT IS CONTOUR INTERVAL TO BE USED.
C
C     RGRID IS AN INTEGER*2 STORAGE ARRAY USED INTERNALLY IN PROGRAM
C        AND NEED NOT BE INITIALIZED. IT IS INCLUDED AS ARGUMENT IN ORDER
C        TO TAKE ADVANTAGE OF VARIABLE DIMENSIONS. DECLARE AS INTEGER*2
C        BEFORE CALLING.
C
C     IN3 AND IN4 ARE X AND Y DIMENSIONS OF RGRID.  DIMENSION RGRID[IN3,IN4]
C        IN3 MUST BE AT LEAST AS LARGE AS NEX2-NEX1<1
C        IN4 MUST BE AT LEAST AS LARGE AS NEY2-NEY1<1
C         [THUS RGRID MUST BE AS LARGE AS PORTION OF DATA ARRAY HH BEING USED
C
C     ZLIT AND ZBIG ARE LOWER AND UPPER CONTOUR CHECK LIMITS. NO CONTOUR
C        WILL BE DRAWN BELOW VALUE OF ZLIT OR ABOVE VALUE OF ZBIG.
C         [USEFUL TO PREVENT DRAWING FOR ANY COMPLETELY WILD DATA]
C
C     ANORTH, ASOUTH, AEAST, AND AWEST CAN BE USED TO ELIMINATE ANY
C        NUMBER OF INCHES FROM ANY SIDE OF FINAL DRAWING.
C
C        FOR FULL DRAWING WITH HEIGHT > HI AND WIDTH > WID.
C          INITIALIZE ALL 4 OF ABOVE ARGUMENTS TO ZERO.
C
C        FOR EACH OF THE ABOVE WITH POSITIVE VALUE, THIS MANY INCHES
C         WILL BE ELIMINATED ON SIDE TO WHICH IT APPLIES.
C          THIS ALLOWS US TO FIT ANY RECTANGULAR GRID TO ANY MERCATOR
C           OR OTHER MAP LIMITS WITHOUT ACTUALLY ADJUSTING THE GRID.
C
C     NDASHD AND NDASHU CONTROL TYPE OF CONTOURS [SOLID OR DASHED LINES]
```

```
C        IF EITHER OR BOTH ARE ZERO OR LESS, CONTOURS ARE SOLID LINES.
C
C       IF BOTH ARE POSITIVE, CONTOURS WILL BE DASHED AS FOLLOWS----
C
C         PEN DOWN SECTION LENGTH > NDASHD*PLTINC    [PLTINC IS INCREMENT LENGT
C           PEN UP SECTION LENGTH > NDASHU*PLTINC
C             [THUS LENGTH OF DASHES AND SKIPS IS FULLY VARIABLE]
C
C         XLABEL CONTROLS LABELING OF CONTOURS. LINES ARE LABELED
C              ONLY IF XLABEL GREATER THAN ZERO. VALUE OF XLABEL
C               IS HEIGHT IN INCHES OF LABEL NUMBERS. LINES ARE LABELED
C               WITH NEAREST WHOLE NUMBER VALUE OF CONTOUR. IF SPECIAL
C                LABELING TO INCLUDE ONLY PART OF NUMBER OR TO INCLUDE
C                DECIMALS IS DESIRED, SUBROUTINE ENDER MUST BE CHANGED.
C
C         SMOOTH IS A CONTROL FOR VARYING CONTOUR SMOOTHING.
C             INITIALIZE SMOOTH TO SOME VALUE BETWEEN 0.25 AND 7.5
C          [ANY VALUE OUTSIDE THIS RANGE IS SET INTERNALLY TO 1.0]
C              LARGER VALUES GIVE SMOOTHER CHART WITH LESS DETAIL, WHILE
C                SMALLER VALUES GIVE LESS SMOOTHING AND MORE DETAIL.
C         [NORMAL VALUE FOR MOST RUNS SHUD BE ABOUT 1.5]
C         ANYTHING LESS THAN ABOUT 0.40 OR LARGER THAN ABOUT 3. IS
C          PROBABLY NO GOOD. BEGIN WITH 1.5 AND EXPERIMENT UP OR DOWN
C         TO DETERMINE MOST DESIRABLE VALUE FOR YOUR NEEDS.
C        [INPUT GRID DATA VALUES ARE NOT ALTERED IN THIS SMOOTHING]
C
C         .IRECCY IS PLOT TAPE RECORD COUNTER. INITIALIZE TO NUMBER
C              OF PLOT RECORDS WRITTEN BEFORE FIRST CALL TO CONTOUR SUBROUTINE.
C
C       ALL OF THE ABOVE ARGUMENTS EXCEPT ARRAY RGRID MUST BE DEFINED.
C         ARGUMENTS ARE NOT ALTERED WITHIN PROGRAM, AND RETURN INTACT.
C
C          PLOTTER BUFFER SPACE MUST BE SET UP AND CALL TO PLOTS
C            MADE BEFORE FIRST CALL TO THIS SUBROUTINE.
C
C            PLOT TAPE MUST BE CLOSED OUT AFTER FINAL CALL.
C
C          ANY NUMBER OF SUCCESSIVE CALLS CAN BE MADE TO CONTOUR
C          SUBROUTINE ECHKON. EACH MAP BECOMES A SEPARATE PLOT RECORD.
C          NO INTERNAL MAP SPACING IS PROVIDED, WITH PEN RETURNING TO
C          ORIGINAL ORIGIN[LOWER LEFT CORNER] AT COMPLETION OF MAP.
C       [THUS IT IS SIMPLE TO PUT MORE THAN ONE SET OF CONTOURS ON SAME MAP]
C          ANY SPECIAL MARKINGS OR LABELS THAT ARE DESIRED MUST BE DONE
C           OUTSIDE THIS SUBROUTINE. THIS SUBROUTINE DRAWS CONTOURS ONLY
C          WITH INCOMING ORIGIN BEING LOWER LEFT CORNER OF CONTOUR CHART.
C
C------------------------------------------------------------------------
C------------------------------------------------------------------------
C
      SUBROUTINE ECHKON(HH,IN1,IN2,NEX1,NEX2,NEY1,NEY2,HI,WID,PLTINC,
     2SAMCON,CONINT,RGRID,IN3,IN4,ZLIT,ZBIG,ANORTH,ASOUTH,AEAST,AWEST,
     3NDASHD,NDASHU,XLABEL,SMOOTH,IRECCY)
C
C          SEE ABOVE COMMENTS FOR DESCRIPTION AND USE OF ABOVE ARGUMENTS
C
      COMMON /STRCON/SMHI,SMWI,X,Y,XGRID,YGRID,CUTOF,SOHI,SOWI,THAX,XPP,
     2YPP,CGIG,U,V,NXUX,JOOO,NUVX,NUVY,YORTH,SOUTH,EAST,WEST,CLIT,CBIG,
     3LCLX,LCLY,INCROS,JINC,CLOSIT,PVAL,PVOL,NENTER,HINUM,NMX1,NMY1,
     4NMX11,NMY11,MOSINC,VALLIN,HINC,MAXCRD,WHAT,LDASH1,LDASH2,DASHER,
     5DOLABS,OUTS
      COMMON /OENDEQ/HIXEN,WODE,HOGH,XXLAST,YYLAST
      LOGICAL DASHER,DOLABS,OUTS
      DIMENSION HH(IN1,IN2),RGRID(IN3,IN4),MEN(4)
      INTEGER RGRID
      DATA IMAP/0/
C
C     NHC SUBROUTINE FOR CONTOURING SCALAR FIELD ON CALCOMP OR MILGO TYPE PLO'
C
      I=IMAP+1
      WRITE(6,10)I
   10 FORMAT(///,2X,11HCONTOUR MAP,I4,1X,26HINCOMING ARGUMENTS FOLLOW.)
      WRITE(6,12)IN1,IN2,NEX1,NEX2,NEY1,NEY2,HI,WID,PLTINC,SAMCON,CONINT
     2,IN3,IN4
   12 FORMAT(/,8X,63HIN1 IN2 NEX1 NEX2 NEY1 NEY2 HI WID PLTINC SAMCON CO
     2NINT IN3 IN4,//,2X,6I5,2F10.4,3F12.3,2I10)
      WRITE(6,14)ZLIT,ZBIG,ANORTH,ASOUTH,AEAST,AWEST,NDASHD,NDASHU,XLABE
     2L,SMOOTH,IRECCY
   14 FORMAT(/,5X,70HZLIT ZBIG ANORTH ASOUTH AEAST AWEST NDASHD NDASHU X
     2LABEL SMOOTH IRECCY,//,2X,2F12.3,4F10.4,2I6,2F10.3,I6,/)
```

```
      HAACRD=0
      WHAT=-99.
      CONINC=CONINT
      IF(CONINC.NE.0.)GO TO 3
      WRITE(6,2)I
    2 FORMAT(/,2X,3HMAP,I3,14H ZERO INTERVAL)
      GO TO 120
    3 IRECI=IRECCY+1
      LDASH1=NDASHO
      LDASH2=NDASHU
      DASHER=.FALSE.
      IF(LDASH2.GT.0.AND.LDASH1.GT.0)DASHER=.TRUE.
      HINUM=XLABEL
      DOLABS=.FALSE.
      IF(HINUM.GT.0.)DOLABS=.TRUE.
      IF(CONINC.LT.0.)CONINC=-CONINC
      PVOL=.005*CONINC
      MOSINC=0
      VALLIN=-989898.989
      NUVX=NEX2-NEX1+1
      NUVY=NEY2-NEY1+1
      IF(NUVX.GT.3.AND.NUVX.LE.IN3.AND.NUVY.GT.3.AND.NUVY.LE.IN4)GO TO 8
      WRITE(6,7)NEX1,NEX2,NUVX,NEY1,NEY2,NUVY
    7 FORMAT(/,10X,23HBAD ARRAY LIMITS. SKIP./10X,3I10/10X,3I10)
      GO TO 120
C        SKIP IF NUVX OR NUVY LESS THAN 4
    8 YORTH=HI-ANORTH
      SOUTH=ASOUTH
      EAST=WID-AEAST
      WEST=AWEST
      IF(WEST.LT.0.)WEST=0.
      IF(EAST.GT.WID)EAST=WID
      IF(SOUTH.LT.0.)SOUTH=0.
      IF(YORTH.GT.HI)YORTH=HI
      HIXEN=HINUM
      WODE=EAST-WEST
      HOGH=YORTH-SOUTH
      XXLAST=99.
      YYLAST=99.
      QINC=PLTINC
      CLIT=QINC/1.99
      CBIG=QINC/2.
      TMAX=4.0*(YORTH-SOUTH+EAST-WEST)
      XGRID=WID/FLOAT(NUVX-1)
      YGRID=HI/FLOAT(NUVY-1)
      HINC=XGRID
      IF(YGRID.LT.XGRID)HINC=YGRID
      X=SMOOTH
      IF(X.LT..25.OR.X.GT.7.5)X=1.0
      HINC=X*HINC
      CUTOF=SQRT(XGRID*XGRID+YGRID*YGRID)+.01
      CLOSIT=.04
C     CLOSIT IS VALUE FOR CLOSED CONTOUR CHECK
      NMX1=NEX1
      NMY1=NEY1
      NMX11=NMX1-1
      NMY11=NMY1-1
      NEX4=NEX1-1
      NEY4=NEY1-1
C
C        NEXT DETERMINE MAX AND MIN VALUES IN SCALAR FIELD
      ZMAX=HH(NEX1,NEY1)
      ZMIN=ZMAX
      DO 30 I=NEX1,NEX2
      DO 30 J=NEY1,NEY2
      IF(HH(I,J).GT.ZMAX)ZMAX=HH(I,J)
      IF(HH(I,J).LT.ZMIN)ZMIN=HH(I,J)
   30 CONTINUE
      IF(ZMAX.GT.ZBIG)ZMAX=ZBIG
      IF(ZMIN.LT.ZLIT)ZMIN=ZLIT
C        NEXT DETERMINE BOTTOM STARTING VALUE FOR CONTOUR LOOP
      PVAL=SAMCON
   32 IF(PVAL.GT.ZMIN)GO TO 34
      PVAL=PVAL+CONINC
      GO TO 32
   34 IF(PVAL-CONINC.LT.ZMIN)GO TO 35
      PVAL=PVAL-CONINC
      GO TO 34
```

```
   35 XPP=0.
      YPP=0.
      N1=NUVX-1
      N2=NUVY-1
C
C     CONTOUR LOOP STARTS BELOW AT STATEMENT 36
C      THIS LOOP DETERMINES WHERE TO START A NEW CONTOUR, THEN CALLS
C        SUBROUTINE CONLIN TO DRAW EACH CONTOUR.  EXIT IS MADE WHEN
C      ALL CONTOURS COMPLETED.
C
C     THERE ARE 2 SCANS FOR EACH CONTOUR VALUE. FIRST WITH VARIABLE OUTS AS
C      FALSE SELECTS ONLY CONTOURS ENTERING GRID FROM OUTSIDE EDGES.
C       SECOND SCAN WITH OUTS TRUE SELECTS REMAINING INNER CONTOURS.
C      STARTING POINT CLOSEST TO PLOT PEN POSITION IS SELECTED IN EACH CASE.
C
C
   36 IF(PVAL.GE.ZMAX)GO TO 110
      OUTS=.FALSE.
      DO 37 I=1,N1
      DO 37 J=1,N2
   37 RGRID(I,J)=0
   38 DZ=999999.
      DO 100 I=1,N1
      DO 100 J=1,N2
      IF(OUTS)GO TO 600
      IF(I.EQ.1.OR.J.EQ.1.OR.I.EQ.N1.OR.J.EQ.N2)GO TO 600
      GO TO 100
  600 IF(RGRID(I,J).EQ.1)GO TO 100
      IF(RGRID(I,J).GT.1.AND.OUTS)GO TO 100
      II=KEX4+I
      JJ=KEY4+J
      HEN(1)=HH(II,JJ)
      HEN(2)=HH(II,JJ+1)
      HEN(3)=HH(II+1,JJ+1)
      HEN(4)=HH(II+1,JJ)
      DO 400 K=1,4
      IF(ABS(HEN(K)-PVAL).GE.PVOL)GO TO 400
      IF(HEN(K).GE.PVAL)HEN(K)=PVAL+PVOL
      IF(HEN(K).LT.PVAL)HEN(K)=PVAL-PVOL
  400 CONTINUE
      IF(OUTS)GO TO 250
      NENN=1
      IF(I.EQ.1.AND.HEN(1).GT.PVAL.AND.HEN(2).LT.PVAL)GO TO 601
      NENN=3
      IF(I.EQ.N1.AND.HEN(3).GT.PVAL.AND.HEN(4).LT.PVAL)GO TO 601
      NENN=4
      IF(J.EQ.1.AND.HEN(4).GT.PVAL.AND.HEN(1).LT.PVAL)GO TO 601
      NENN=2
      IF(J.EQ.N2.AND.HEN(2).GT.PVAL.AND.HEN(3).LT.PVAL)GO TO 601
      GO TO 602
  250 DO 410 K=1,4
      I1=K
      I2=K+1
      IF(K.EQ.4)I2=1
      IF(HEN(I1).GT.PVAL.AND.HEN(I2).LT.PVAL)GO TO 408
  410 CONTINUE
      GO TO 602
  408 NENN=K
  601 IF(RGRID(I,J).EQ.0.OR.OUTS)GO TO 640
      I1=RGRID(I,J)/10
      I2=RGRID(I,J)-10*I1
      IF(I1.EQ.NENN.OR.I2.EQ.NENN)GO TO 100
  640 GO TO(340,342,344,346),NENN
  602 IF(RGRID(I,J).EQ.0)RGRID(I,J)=1
      GO TO 100
  340 Y=YGRID*(FLOAT(J-1)+(PVAL-HEN(1))/(HEN(2)-HEN(1)))
      X=XGRID*FLOAT(I-1)
      GO TO 45
  342 X=XGRID*(FLOAT(I-1)+(PVAL-HEN(2))/(HEN(3)-HEN(2)))
      Y=YGRID*FLOAT(J)
      GO TO 45
  344 Y=YGRID*(FLOAT(J-1)+(PVAL-HEN(4))/(HEN(3)-HEN(4)))
      X=XGRID*FLOAT(I)
      GO TO 45
  346 X=XGRID*(FLOAT(I-1)+(PVAL-HEN(1))/(HEN(4)-HEN(1)))
      Y=YGRID*FLOAT(J-1)
   45 D=(X-XPP)*(X-XPP)+(Y-YPP)*(Y-YPP)
      IF(D.GE.DZ)GO TO 100
      DZ=D
```

```
      NENTER=NLNN
      LCLX=I
      LCLY=J
      XTT=X
      YTT=Y
  100 CONTINUE
      IF(DZ.GE.999990.)GO TO 105
      IF(RGRID(LCLX,LCLY).EQ.0)RGRID(LCLX,LCLY)=1
      X=XTT
      Y=YTT
C
      WRITE(6,101)PVAL,DDLABS,OUTS
  101 FORMAT(1X,E13.5,2L13)
C          NEXT CALL SUBROUTINE CONLIN TO ACTUALLY DRAW CONTOUR WITH VALUE PV
C
      CALL CONLIN(HH,IN1,IN2,RGRID,IN3,IN4)
C
C      NOW GO BACK TO INNER LOOP TO SEE IF THERE ARE OTHER PVAL CONTOURS
C      TO BE DRAWN.
C
      GO TO 38
  105 IF(OUTS)GO TO 612
      OUTS=.TRUE.
      GO TO 38
  612 PVAL=PVAL+CONINC
C          INCREMENT CONTOUR AND GO TO TOP OF LOOP FOR NEXT CONTOUR
      GO TO 36
  110 CALL PLOT(0.,0.,-3)
      IMAP=IMAP+1
      IRECCY=IRECCY+1
      WRITE(6,115)IMAP,IREC1,IRECCY
  115 FORMAT(/,10X,11HCONTOUR MAP,I3,24H BEGINS WITH PLOT RECORD,I3,14H
     2AND ENDS WITH,I3)
      WRITE(6,116)MOSINC,VALLIN,MAXCRD,WHAT
  116 FORMAT(12X,21HMOST LINE INCREMENTS ,I5,12H ON CONTOUR ,F10.2,/,12X
     2,12HMOST SQUARES,I4,12H ON CONTOUR ,F10.2,/)
  120 RETURN
      END
@PRT,S S.CONLIN
```

```
*C ---SUMDN(1).CONLIN ELT CREATED ON 2 JUL 79 AT 21:11.24
      SUBROUTINE CONLIN(HH,IN1,IN2,QGRID,IN3,IN4)
      COMMON /STRCON/SNWI,SAWI,X,Y,XGRID,YGRID,CUTUF,SONI,SOWI,IMAX,XPP,
     2YPP,CGIG,J,V,NXUX,JDOO,NUVX,NUVY,YORTH,SOUTH,EAST,WEST,CLIT,CBIG,
     3LCLX,LCLY,INCROS,QINC,CLOSIT,PVAL,PVOL,NENTER,HINUM,NMX1,NMY1,
     4NMX11,NMY11,MOSINC,VALLIN,MINC,MAXCRD,WHAT,LDASH1,LDASH2,DASHER,
     5DOLABS,OUTS
      DIMENSION HH(IN1,IN2),CIDE(4,2),XXPLOT(275,2),HAX(4),LEXE(4),
     2CORD(400,2),HIPPS(400)
      INTEGER QGRID(IN3,IN4)
      LOGICAL INCS,DOLABS,DASHER,CLOS,OUTS,DASHIX
C
C           THIS SUBROUTINE IS CALLED TO DRAW EACH INDIVIDUAL CONTOUR
C
C        IF DOLABS ENTERS AS TRUE, LABEL CONTOURS WITH HEIGHT HINUM
      DASHIX=DASHER
      LABLIT=9
      IF(DOLABS)LABLIT=0
      INCS=.FALSE.
      YMAX=-9.
      XMAX=-9.
      NENST=NENTER
      IDPLOT=2
      NHARD=LDASH1
      NSOFT=LDASH2
      NUGG=0
      XX=X
      YY=Y
      XBIG=XX
      YBIG=YY
      L2X=LCLX
      L2Y=LCLY
      IPER=2
      XD=0.
      YD=0.
      TOT=0.
      HYPTOT=0.
      NCORD=0
      CLOS=.FALSE.
      GO TO 400
C
C     END SETUP. BEGIN LOOP THAT PICKS EXACT STRAIGHT LINE SEGMENTED TRAVERSE
C
  250 IF(NCORD.LT.400)GO TO 252
      WRITE(6,251)NCORD,PVAL
  251 FORMAT(/,2X,I5,19H SQUARES LINE ,F10.5,2X,7HSHUTOFF)
C     SHUTOFF MESSAGE HERE INDICATES THAT THIS CONTOUR CROSSES MORE
C        THAN 400 GRID SQUARES. ARRAYS CORD AND HIPPS ARE TOO SMALL. CONTOUR
C       WILL BE CUT OFF AT SQUARE 400. CURE IS TO ENLARGE ARRAYS
C           AND ASSOCIATED CUTOFF CHECK STATEMENT ABOVE.
      GO TO 730
  252 NCORD=NCORD+1
      HYPTOT=HYPTOT+HYPE
      HIPPS(NCORD)=HYPE
      CORD(NCORD,1)=XXSQ
      CORD(NCORD,2)=YYSQ
      XD=XD+XXSQ
      YD=YD+YYSQ
      IF(NEXET.EQ.1)LZX=LZX-1
      IF(NEXET.EQ.3)LZX=LZX+1
      IF(NEXET.EQ.2)LZY=LZY+1
      IF(NEXET.EQ.4)LZY=LZY-1
      IF(LZX.LT.1.OR.LZX.GE.NUVX)GO TO 730
      IF(LZY.LT.1.OR.LZY.GE.NUVY)GO TO 730
      IF(LZX.EQ.LCLX.AND.LZY.EQ.LCLY.AND.SQRT(XD*XD+YD*YD).LE.CLOSIT.AND
     2.OUTS.AND.NCORD.GT.3)GO TO 701
      GO TO 700
  701 CLOS=.TRUE.
      IF(LABLIT.EQ.0)LABLIT=-1
      GO TO 730
  700 NENTER=NEXET+2
      IF(NEXET.GT.2)NENTER=NEXET-2
      XBIG=XIND
      YBIG=YIND
  400 NUMOUT=0
      DX1=XBIG-XGRID*FLOAT(LZX-1)
      IF(DX1.LT.0.)DX1=0.
      IF(DX1.GT.XGRID)DX1=XGRID
      DY1=YBIG-YGRID*FLOAT(LZY-1)
```

```
      IF(OY1.GT.0.)OY1=0.
      IF(OY1.GT.YGRID)OY1=YGRID
      I=LZX+NMX11
      J=LZY+NMY11
C
C        START EXIT POINT LOOP
C
      HAX(1)=HH(I,J)
      HAX(2)=HH(I,J+1)
      HAX(3)=HH(I+1,J+1)
      HAX(4)=HH(I+1,J)
      DO 401 II=1,4
      IF(ABS(HAX(II)-PVAL).GE.PVOL)GO TO 401
      IF(HAX(II).GE.PVAL)HAX(II)=PVAL+PVOL
      IF(HAX(II).LT.PVAL)HAX(II)=PVAL-PVOL
    . CONTINUE
      NEXET=0
      DO 435 III=1,4
      CIDE(III,1)=-1.
      CIDE(III,2)=-1.
      I1=III
      I2=III+1
      IF(III.EQ.4)I2=1
C
C        STATEMENT BELOW SELECTS SIDES THAT HAVE EXIT POINTS
C
      IF(HAX(I1).LT.PVAL.AND.HAX(I2).GT.PVAL)GO TO 420
      GO TO 435
  420 NUMOUT=NUMOUT+1
      IF(NUMOUT.EQ.1)NN1=III
      IF(NUMOUT.EQ.2)NN2=III
      GO TO(422,424,426,428),III
  422 OY2=((PVAL-HAX(1))/(HAX(2)-HAX(1)))*YGRID
      OX2=0.
      GO TO 430
  424 OX2=((PVAL-HAX(2))/(HAX(3)-HAX(2)))*XGRID
      OY2=YGRID
      GO TO 430
  426 OY2=((PVAL-HAX(4))/(HAX(3)-HAX(4)))*YGRID
      OX2=XGRID
      GO TO 430
  428 OX2=((PVAL-HAX(1))/(HAX(4)-HAX(1)))*XGRID
      OY2=0.
  430 CIDE(III,1)=OX2
      CIDE(III,2)=OY2
  435 CONTINUE
C     UNLESS WE HAVE NULL POINT SQUARE, NUMOUT SHUD BE 1 WITH OUT AT OX2, OY2
      IF(NUMOUT.NE.1)GO TO 432
      NEXET=NN1
      GO TO 495
  432 IF(NUMOUT.EQ.2)GO TO 438
  431 WRITE(6,436)LZX,LZY,NUMOUT,PVAL,XBIG,YBIG
  436 FORMAT(/,2X,10HNO WAY OUT,6X,3I10,3F10.2,/)
      GO TO 500
C
C     BEGIN SECTION THAT DETERMINES PROPER PATH THRU GRID SQUARE
C      CONTAINING HYPERBOLIC CONFIGURATION. (2 ENTRY AND 2 EXIT SIDES)
C
  436 IF(RGRID(LZX,LZY).GT.1)GO TO 442
      XID=CIDE(NN1,1)-OX1
      YID=CIDE(NN1,2)-OY1
      DAA=SQRT(XID*XID+YID*YID)
      XID=CIDE(NN2,1)-OX1
      YID=CIDE(NN2,2)-OY1
      DBB=SQRT(XID*XID+YID*YID)
      IF(DBB.LT.DAA)GO TO 440
  439 OX2=CIDE(NN1,1)
      OY2=CIDE(NN1,2)
      NEXET=NN1
      GO TO 414
  440 OX2=CIDE(NN2,1)
      OY2=CIDE(NN2,2)
      NEXET=NN2
  414 RGRID(LZX,LZY)=10*NENTER+NEXET
      GO TO 495
  442 I1=RGRID(LZX,LZY)/10
      I2=RGRID(LZX,LZY)-10*I1
      RGRID(LZX,LZY)=1
      IF(I1.GT.0.AND.I2.GT.0.AND.I1.NE.I2.AND.NENTER.GT.0)GO TO 417
```

```
  415 WRITE(6,416)(2X,L2Y,I1,I2,CENTER
  416 FORMAT(/,2X,12HBAD COL EXIT,2X,5I12)
      GO TO 500
  417 LEXE(1)=0
      LEXE(2)=0
      LEXE(3)=0
      LEXE(4)=0
      LEXE(I1)=1
      LEXE(I2)=1
      DO 418 II=1,4
      IF(LEXE(II).EQ.0.AND.II.NE.NENTER.AND.CIDE(II,1).GT.(-.5).AND.CIDE
     2(II,2).GT.(-.5))GO TO 419
  418 CONTINUE
      GO TO 415
  419 NEXET=II
      OX2=CIDE(II,1)
      OY2=CIDE(II,2)
C                     END HYPERBOLIC GRID SQUARE SECTION
  445 XIND=OX2+XGRID*FLOAT(LZX-1)
      YIND=OY2+YGRID*FLOAT(LZY-1)
      IF(RGRID(LZX,LZY).EQ.0)RGRID(LZX,LZY)=1
      XXSQ=XIND-XBIG
      YYSQ=YIND-YBIG
      HYPE=SQRT(XXSQ*XXSQ+YYSQ*YYSQ)
      IF(HYPE.GE..0001.AND.HYPE.LE.CUTOF)GO TO 396
      WRITE(6,397)LZX,LZY,NENTER,NEXET,XBIG,YBIG,XIND,YIND,OX1,OY1,OX2,
     2OY2,XXSQ,YYSQ,HYPE,NENST,LCLX,LCLY,NCORD,XX,YY
  397 FORMAT(/,2X,4HHERE,4I10,/,2X,11F10.5,//,2X,4I10,5X,2F12.6,//)
      GO TO 500
  396 GO TO 250
C
C         LINE SEGMENTED CONTOUR TRAVERSE NOW COMPLETE. NEXT, DIVIDE THIS
C           TRAVERSE INTO NRINC EQUAL SEGMENTS. THIS NUMBER IS FUNCTION NOT
C             ONLY OF LENGTH OF TRAVERSE, BUT OF INCOMING ARGUMENT SMOOTH, WHICH
C               CONTROLS DEGREE OF SMOOTHING DESIRED.
C
  730 IF(NCORD.LE.MAXCRD)GO TO 732
      MAXCRD=NCORD
      WHAT=PVAL
  732 NRINC=HYPTOT/HINC+1
      HANC=HYPTOT/FLOAT(NRINC)
      IF(.NOT.CLOS.AND.NCORD.GT.1.AND.NRINC.GT.1)GO TO 734
      IF(CLOS.AND.NCORD.GT.3.AND.NRINC.GT.2)GO TO 734
      GO TO 502
C
C         NEXT, SET UP ENTRY AND EXIT SLOPE DATA FOR FIRST SEGMENT BEFORE
C           ENTERING MAIN CURVILINEAR INTERPOLATE AND PLOT LOOP.
C
  734 XBEG=XX
      YBEG=YY
      IF(NRINC*HANC.LT..75)DASHIX=.FALSE.
      H=0.
      XEND=XX
      YEND=YY
      DO 740 I=1,NCORD
      IF(H+HIPPS(I).GE.HANC)GO TO 742
      H=H+HIPPS(I)
      XEND=XEND+CORD(I,1)
  740 YEND=YEND+CORD(I,2)
      GO TO 745
  742 X=(HANC-H)/HIPPS(I)
      XEND=XEND+X*CORD(I,1)
      YEND=YEND+X*CORD(I,2)
  745 XXSQ=XEND-XBEG
      YYSQ=YEND-YBEG
      HYPE=SQRT(XXSQ*XXSQ+YYSQ*YYSQ)
      IF(HYPE.GE..0001)GO TO 750
  747 WRITE(6,748)XBEG,YBEG,PVAL,HYPE
  748 FORMAT(/,2X,14HHYPE TOO SMALL,2X,4F12.6)
      GO TO 500
  750 CANG=XXSQ/HYPE
      SANG=YYSQ/HYPE
      HYPM=HYPE
      IF(CLOS)GO TO 751
      COSBAC=CANG
      SINBAC=SANG
      GO TO 759
  751 XE=XX
      YE=YY
```

```
      H=0.
      DO 755 I=1,NCORD
      J=NCORD-I+1
      IF(H+HIPPS(J).GE.HANG)GO TO 755
      H=H+HIPPS(J)
      XB=XB-CORD(J,1)
   63 YB=YB-CORD(J,2)
      GO TO 757
  755 X=(HANG-H)/HIPPS(J)
      XB=XB-X*CORD(J,1)
      YB=YB-X*CORD(J,2)
  757 XXSQ=XBEG-XB
      YYSQ=YBEG-YB
      HYPE=SQRT(XXSQ*XXSQ+YYSQ*YYSQ)
      IF(HYPE.LT..0001)GO TO 747
      SINBAC=YYSQ/HYPE
      COSBAC=XXSQ/HYPE
  759 A=.5*ATAN2(SANG*COSBAC-CANG*SINBAC,CANG*COSBAC+SANG*SINBAC)
      SA=SIN(A)
      CA=COS(A)
      SENT=SINBAC*CA+COSBAC*SA
      CENT=COSBAC*CA-SINBAC*SA
      ENTIR=ATAN2(SENT,CENT)
      SSENT=SENT
      CCENT=CENT
C
C         ENTER MAIN CURVILINEAR INTERPOLATE AND PLOT LOOP
C
      DO 800 LUPE=1,NRINC
      IF(LUPE.NE.NRINC)GO TO 762
      IF(CLOS)GO TO 760
      SOUT=SANG
      COUT=CANG
      GO TO 200
  760 SOUT=SSENT
      COUT=CCENT
      GO TO 200
  762 XIND=XX
      YIND=YY
      ZANC=HANC*FLOAT(LUPE+1)
      H=0.
      DO 764 I=1,NCORD
      IF(H+HIPPS(I).GE.ZANC)GO TO 766
      H=H+HIPPS(I)
      XIND=XIND+CORD(I,1)
  764 YIND=YIND+CORD(I,2)
      GO TO 768
  766 X=(ZANC-H)/HIPPS(I)
      XIND=XIND+X*CORD(I,1)
      YIND=YIND+X*CORD(I,2)
  768 XXSQ=XIND-XEND
      YYSQ=YIND-YEND
      HYPE=SQRT(XXSQ*XXSQ+YYSQ*YYSQ)
      IF(HYPE.LT..0001)GO TO 747
      SINFOR=YYSQ/HYPE
      COSFOR=XXSQ/HYPE
      HYPFOR=HYPE
      A=.5*ATAN2(SINFOR*CANG-COSFOR*SANG,COSFOR*CANG+SINFOR*SANG)
      SA=SIN(A)
      CA=COS(A)
      SOUT=SANG*CA+CANG*SA
      COUT=CANG*CA-SANG*SA
      EXCT=ATAN2(SOUT,COUT)
  200 HYP=HYPE
      IF(HYP.GT.QINC)GO TO 449
      IF(XEND.LT.WEST.OR.XEND.GT.EAST.OR.YEND.LT.SOUTH.OR.YEND.GT.NORTH)
     2GO TO 790
      IF(INCS)GO TO 446
      INCS=.TRUE.
      CALL PLOT(XBEG-WEST,YBEG-SOUTH,3)
  446 I=IPER
      IF(DASHIX.AND.I.EQ.2)I=IDPLOT
C     CALL PLOT(XEND-WEST,YEND-SOUTH,I)
      NUGG=NUGG+1
      TOT=TOT+HYP
      GO TO 790
C
C     BEGIN SNAKE INTERPOLATION FOR SEGMENT
C
```

```
C     ENTIR AND EXIT ARE TRUE ENTRY AND EXIT DIRECTIONS AT ENDS OF SEGMENT
C        SNAKELIKE INTERPOLATION IS MADE FOR FIT BETWEEN 2 END DIRECTIONS.
C     SENT AND CENT ARE SINE AND COSINE OF TRUE ENTRY ANGLE
C     SOUT AND COUT ARE SINE AND COSINE OF TRUE EXIT ANGLE
C
C     SANG AND CANG ARE SINE AND COSINE OF STRAIGHT LINE ENTRY/EXIT CONNECTOR
C
C     BEGIN SECTION THAT INTERPOLATES AND PLOTS THRU SEGMENT
C
  449 S=SENT*CANG-CENT*SANG
      C=CENT*CANG+SENT*SANG
      C1=(2.*ATAN2(S,C))/HYP
      SING=SANG*C-CANG*S
      CING=CANG*C+SANG*S
      S=SOUT*CING-COUT*SING
      C=COUT*CING+SOUT*SING
      C2=(2.*ATAN2(S,C))/HYP
      TX=XBEG
      TY=YBEG
      NINC=0
      TYP=-CBIG
      HYPMAX=HYP-CLIT
      H25=.25*HYP
  450 TYP=TYP+QINC
      D1=TYP
      IF(D1.GT.H25)D1=H25
      D2=TYP-H25
      IF(D2.LT.0.)D2=0.
      SING=ENTIR-C1*TYP+C2*(D2-D1)
      TX=TX+QINC*COS(SING)
      TY=TY+QINC*SIN(SING)
C
C        END SNAKE INTERPOLATION SECTION---TRY AND FIGURE IT OUT AND GO NUTS-
C     NEXT STORE POINTS THRU THIS SEGMENT FOR FINAL ADJUST AND PLOT
C
      IF(NINC.LT.275)GO TO 453
      WRITE(6,454)PVAL,XBEG,YBEG,XEND,YEND
  454 FORMAT(/,2X,12HNINC SHUTOFF,2X,5F12.3)
C     IF SHUTOFF MESSAGE RECEIVED HERE, ARRAY XXPLOT IS TOO SMALL.
C     FOR INFREQUENT MESSAGES, DONT WORRY ABOUT IT, SINCE LACK OF
C        CLOSURE IS ADJUSTED OUT.  IF MESSAGE PERSISTS, EITHER INCREASE
C          PLOT INCREMENT LENGTH OR SIZE OF XXPLOT.
      GO TO 455
  453 NINC=NINC+1
      XXPLOT(NINC,1)=TX
      XXPLOT(NINC,2)=TY
      IF(TYP.LE.HYPMAX)GO TO 450
C
C      ADJUST FOR CLOSURE ERROR, THEN PLOT CURVE ALONG THIS SEGMENT.
C
  455 XER=(XEND-XXPLOT(NINC,1))/FLOAT(NINC)
      YER=(YEND-XXPLOT(NINC,2))/FLOAT(NINC)
      U=0.
      V=0.
      NUNC=0
C
C       BEGIN SEGMENT PLOT LOOP----DASHED OR SOLID CURVES-----
C       SUBROUTINE ENDER IS CALLED TO LABEL LINES
C
      DO 610 I=1,NINC
      U=U+XER
      V=V+YER
      X=XXPLOT(I,1)+U
      Y=XXPLOT(I,2)+V
      IF(X.LT.WEST.OR.X.GT.EAST.OR.Y.LT.SOUTH.OR.Y.GT.YORTH)GO TO 608
      IF(INCS)GO TO 603
      INCS=.TRUE.
      IF(IPER.EQ.2)CALL PLOT(XBEG-WEST,YBEG-SOUTH,3)
  603 XPP=X
      YPP=Y
      XUX=XPP-WEST
      YUY=YPP-SOUTH
      IF(LABLIT.GT.0)GO TO 604
      IF(LABLIT.EQ.0)CALL ENDER(XUX,YUY,PVAL,1)
      LABLIT=1
  604 IF(DASHIX.AND.IPER.EQ.2)GO TO 615
      CALL PLOT(XUX,YUY,IPER)
      GO TO 609
  615 CALL PLOT(XUX,YUY,IDPLOT)
```

```
      IF(IDPLED.EQ.1)GO TO 630
      NHARD=NHARD-1
      IF(NHARD.GT.0)GO TO 609
      IOPLOT=3
      NHARD=LDASH1
      GO TO 609
  530 NSOFT=NSOFT-1
      IF(NSOFT.GT.0)GO TO 609
      IOPLOT=2
      NSOFT=LDASH2
  609 IPER=2
      NUNC=NUNC+1
      IF(.NOT.DOLABS.OR.YUY.LE.YMAX)GO TO 610
      YMAX=YUY
      XMAX=XUX
      GO TO 610
  608 IPER=3
      IF(LABLIT.NE.1)GO TO 610
      CALL ENDER(X-WEST,Y-SOUTH,PVAL,1)
      LABLIT=0
  610 CONTINUE
C
C        END ADJUST AND PLOT SECTION
C
      NUGG=NUGG+NUNC
      TOT=TOT+DINC*FLOAT(NUNC)
      IF(TOT.LT.TMAX)GO TO 790
      WRITE(6,462)PVAL,TMAX
  462 FORMAT(/,2X,16HREACHED TMAX ON ,2F12.4)
      GO TO 500
  790 IF(LUPE.EQ.NRINC)GO TO 800
      HYPM=HYPFOR
      SENT=SOUT
      CENT=COUT
      ENTIR=EXET
      SANG=SINFOR
      CANG=COSFOR
      XBEG=XEND
      YBEG=YEND
      XEND=XIND
      YEND=YIND
  800 CONTINUE
C
C        END MAIN CURVILINEAR INTERPOLATE AND PLOT LOOP
C
  500 IF(.NOT.CLOS.OR..NOT.DOLABS.OR.YMAX.LT..01)GO TO 501
      XPP=XMAX+WEST
      YPP=YMAX+SOUTH
      CALL ENDER(XMAX,YMAX,PVAL,2)
  501 IF(.NOT.CLOS.AND.LABLIT.EQ.1)CALL ENDER(XUX,YUY,PVAL,1)
      IF(NUGG.LE.MOSINC)GO TO 502
      MOSINC=NUGG
      VALLIN=PVAL
  502 RETURN
      END
.PRT,S S.ENDER
```

ENDER CREATED ON 2 JUL 79 AT 21:11:42

```
      SUBROUTINE ENDER(X,Y,PVAL,ICO)
      COMMON /QENDEQ/HONUM,WODE,HOGH,XLAS,YLAS
      DIMENSION D(3)
C
C        THIS SUBROUTINE IS CALLED TO LABEL CONTOURS
C
      DM=SQRT((X-XLAS)*(X-XLAS)+(Y-YLAS)*(Y-YLAS))
      IF(DM.LT.2.*HONUM.OR.HONUM.LE.0.)GO TO 25
      JJJ=0
      IF(ICO.EQ.2)GO TO 14
      D(1)=ABS(Y-HOGH)
      D(2)=ABS(X-WODE)
      D(3)=ABS(Y)
      K=1
      DM=ABS(X)
      DO 10 I=1,3
      IF(D(I).GE.DM)GO TO 10
      DM=D(I)
      K=I+1
   10 CONTINUE
      GO TO(12,14,16,18),K
   12 YAD=-HONUM/2.
      I=1
      GO TO 100
   14 YAD=.02
      I=2
      GO TO 100
   16 XAD=.02
      YAD=-HONUM/2.
      GO TO 20
   18 YAD=-HONUM-.02
      I=2
      GO TO 100
   20 XAD=X+XAD
      YAD=Y+YAD
      JJJ=JJJ+1
      IF(JJJ.EQ.2)GO TO 333
      GO TO 334
  333 CALL NUMBER(XAD,YAD,HONUM,PVAL,270.,2)
      JJJ=0
  334 CONTINUE
      CALL PLOT(X,Y,3)
      XLAS=X
      YLAS=Y
   25 RETURN
  100 XAD=-.75*HONUM
      IF(PVAL.GE.9.5.OR.PVAL.LE.(-.5))XAD=XAD-HONUM
      IF(PVAL.GE.99.5.OR.PVAL.LE.(-9.5))XAD=XAD-HONUM
      IF(PVAL.GE.999.5.OR.PVAL.LE.(-99.5))XAD=XAD-HONUM
      IF(I.EQ.2)XAD=.5*XAD
      GO TO 20
      END
```

SAMPLE RUN

```
,   1.833000,   -.330000,   -2.780000,   .000000,   15.000000,     .000000,  -1,  1
,  10.000000,   3.610000,   10.745549,   .419789,   -2.780000,   10.015399,   0,  1
,   9.991421,   3.605710,   10.745549,   .419637,   -2.780000,   10.014379,   0,  1
,   9.984053,   3.602026,   10.745549,   .419600,   -2.780000,   10.013075,   0,  1
,   9.977635,   3.598818,   10.745549,   .419524,   -2.780000,   10.011939,   0,  1
,   9.971972,   3.595986,   10.745549,   .419457,   -2.780000,   10.010936,   0,  1
,   9.965914,   3.593457,   10.745549,   .419397,   -2.780000,   10.010041,   0,  1
,   9.962349,   3.591175,   10.745549,   .419343,   -2.780000,   10.009234,   0,  1
,   9.958187,   3.589094,   10.745549,   .419294,   -2.780000,   10.008498,   0,  1
,   9.954359,   3.587179,   10.745549,   .419249,   -2.730000,   10.007820,   0,  1
,   9.950808,   3.585404,   10.745549,   .419207,   -2.730000,   10.007193,   0,  1
,   9.947492,   3.583746,   10.745549,   .419168,   -2.780000,   10.006606,   0,  1
,   9.944374,   3.582187,   10.745549,   .419131,   -2.780000,   10.006355,   0,  1
TOTAL TIME THUS FAR =  1.0HRS

TAUX=   -.039858     TAUY=    .146753

TAI=    .1000000+001
TAN=    .6907031-002
TAV=    .2262187+003
CB=    .0000000
CW=    .0000000
AKT=    .1911157-001
TREF=    .1000000+002



RPEF=    .9997780+000
EUL=    .1742220+004
A=    .1000428+001
B=   -.1900000-004
C=   -.4600000-005
TC=   -.1278000+001
TD=    .1000000+002

ITN=   6        EX=    .7931227-003
```

U-VELOCITY FOR K= 1

.00	.00	.00	.00	.00	.00	.00	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	.00
.00	.00	.00	.00	.00	.00	.00	-.70	-.67	-.58	-.52	-.48	-.44	-.41	-.39	-.32	-.03	.14	-.67	.00
.00	.00	.00	.00	.00	.00	.00	-.59	-.72	-.64	-.56	-.51	-.48	-.49	-.53	-.49	.00	.00	.00	.00
.00	.00	.00	.00	.00	.23	-.03	-.53	-.67	-.62	-.56	-.51	-.49	-.52	-.55	-.46	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	-.23	-.52	-.60	-.57	-.52	-.48	-.46	-.48	-.50	-.37	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	-.45	-.59	-.57	-.52	-.47	-.43	-.40	-.38	-.40	-.39	.00	.00	.00	.00
.00	.00	.00	.00	-.41	.00	-.62	-.54	-.54	-.46	-.40	-.36	-.32	-.25	-.14	-.07	.24	.44	.33	.00
.00	.00	.00	-.39	-.78	-.60	-.63	-.55	-.44	-.37	-.35	-.30	-.26	-.16	.14	.00	.00	.15	.00	.00
.00	.00	.00	-1.26	-1.15	-.72	-.48	-.37	-.30	-.26	-.25	-.21	-.20	-.12	.00	.00	.00	.15	.00	.00
.00	.00	.00	-2.81	-1.48	-.53	-.20	-.13	-.14	-.15	-.12	-.09	-.10	-.09	.00	.00	.00	.00	.00	.00
.00	-1.13	-2.24	-1.85	-.72	-.04	.15	.10	.02	-.03	.01	.06	.02	-.10	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.28	.22	.14	.10	.00	.00	.00	-.03	.24	.29	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.08	.00	.18	.18	.00	.00	.00	.09	.20	.27	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.20	.00	.00	.00	.00	.00	.17	.23	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.11	.00	.00	.00	.00	.00	.17	.21	.19	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.02	.14	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00

U-VELOCITY FOR K= ?

.00	.00	.00	.00	.00	.00	.00	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	.00
.00	.00	.00	.00	.00	.00	.00	-.91	-.89	-.80	-.74	-.70	-.67	-.64	-.62	-.55	-.25	-.08	-.29	.00
.00	.00	.00	.00	.00	.00	.00	-.80	-.93	-.85	-.78	-.73	-.70	-.71	-.75	-.71	.00	.00	.00	.00
.00	.00	.00	.00	.00	.02	-.25	-.74	-.88	-.84	-.77	-.73	-.71	-.73	-.77	-.67	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	-.45	-.73	-.81	-.79	-.74	-.70	-.68	-.70	-.72	-.58	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	-.66	-.80	-.79	-.73	-.68	-.64	-.62	-.60	-.62	-.60	.00	.00	.00	.00
.00	.00	.00	.00	-.62	.00	-.83	-.85	-.75	-.67	-.62	-.56	-.54	-.47	-.36	-.29	.22	.22	.12	.00
.00	.00	.00	-.56	-.98	-.81	-.84	-.75	-.66	-.59	-.55	-.52	-.48	-.38	-.08	.00	.00	-.06	.00	.00
.00	.00	.00	-1.42	-1.34	-.93	-.70	-.58	-.52	-.46	-.45	-.43	-.42	-.33	.00	.00	.00	-.07	.00	.00
.00	.00	.00	-2.94	-1.66	-.74	-.42	-.35	-.36	-.37	-.34	-.31	-.33	-.31	.00	.00	.00	.00	.00	.00
.00	-1.35	-2.40	-2.00	-.93	-.26	-.08	-.12	-.20	-.26	-.21	-.16	-.20	-.32	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.06	.00	-.08	-.12	.00	.00	.00	-.24	.02	.07	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.08	.00	-.04	-.04	.00	.00	.00	-.12	-.02	.05	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	-.01	.00	.00	.00	.00	.00	-.05	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.05	.00	-.10	.00	.00	.00	.00	.00	-.05	-.01	-.02	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	-.37	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00

U-VELOCITY FOR K= 3

.00	.00	.00	.00	.00	.00	.00	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	.00
.00	.00	.00	.00	.00	.00	.00	-.96	-.95	-.97	-.81	-.78	-.74	-.71	-.69	-.62	-.31	-.13	-.33	.00
.00	.00	.00	.00	.00	.00	.00	-.89	-.98	-.90	-.83	-.78	-.76	-.77	-.81	-.75	.00	.00	.00	.00
.00	.00	.00	.00	.00	-.02	-.29	-.78	-.92	-.88	-.82	-.78	-.77	-.79	-.82	-.70	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	-.50	-.78	-.86	-.84	-.79	-.75	-.74	-.75	-.77	-.62	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	-.71	-.85	-.84	-.79	-.74	-.70	-.68	-.66	-.67	-.64	.00	.00	.00	.00
.00	.00	.00	.00	-.64	.00	-.87	-.91	-.81	-.73	-.68	-.64	-.60	-.54	-.42	-.34	-.04	.16	.07	.00
.00	.00	.00	-.57	-.98	-.83	-.89	-.82	-.72	-.65	-.61	-.68	-.65	-.44	-.13	.00	.00	-.11	.00	.00
.00	.00	.00	-1.32	-1.32	-.97	-.75	-.54	-.59	-.55	-.52	-.50	-.48	-.39	.00	.00	.00	-.11	.00	.00
.00	.00	.00	-2.73	-1.61	-.78	-.48	-.42	-.43	-.44	-.41	-.38	-.39	-.36	.00	.00	.00	.00	.00	.00
.00	-1.50	-2.29	-1.84	-.87	-.30	-.14	-.19	-.28	-.32	-.27	-.22	-.26	-.37	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.01	-.06	-.15	-.13	.00	.00	.00	-.30	-.04	.01	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.08	.00	-.09	-.09	.00	.00	.00	-.18	-.10	-.02	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	-.06	.00	.00	.00	.00	.00	-.11	-.06	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	-.15	.00	.00	.00	.00	.00	-.10	-.07	-.07	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	-.11	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00

U-VELOCITY FOR K= 4

.00	.00	.00	.00	.00	.00	.00	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	-1.39	.00
.00	.00	.00	.00	.00	.00	.00	-.79	-.75	-.69	-.65	-.63	-.60	-.58	-.57	-.52	-.27	-.12	-.29	.00
.00	.00	.00	.00	.00	.00	.00	-.66	-.76	-.70	-.64	-.61	-.60	-.61	-.65	-.62	.00	.00	.00	.00
.00	.00	.00	.00	.00	-.02	-.23	-.61	-.72	-.69	-.64	-.61	-.60	-.62	-.65	-.57	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	-.39	-.61	-.67	-.66	-.62	-.59	-.58	-.59	-.61	-.49	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	-.58	-.68	-.65	-.62	-.58	-.55	-.53	-.62	-.63	-.52	.00	.00	.00	.00
.00	.00	.00	.00	-.49	.00	-.72	-.72	-.64	-.58	-.54	-.51	-.48	-.43	-.33	-.28	-.03	.13	.05	.00
.00	.00	.00	-.41	-.75	-.65	-.71	-.65	-.57	-.52	-.48	-.46	-.44	-.36	-.10	.00	.00	-.11	.00	.00
.00	.00	.00	-.95	-1.31	-.75	-.59	-.51	-.46	-.44	-.41	-.39	-.39	-.31	.00	.00	.00	-.10	.00	.00
.00	.00	.00	-2.17	-1.24	-.59	-.37	-.33	-.34	-.35	-.32	-.30	-.31	-.28	.00	.00	.00	.00	.00	.00
.00	-1.32	-1.71	-1.35	-.63	-.21	-.10	-.15	-.22	-.23	-.22	-.17	-.21	-.30	.00	.00	.00	.06	.00	.00
.00	.00	.00	.00	.00	.00	-.01	-.05	-.12	-.15	.00	.00	.00	-.25	-.04	.01	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	-.08	-.07	.00	.00	.00	-.15	-.09	-.02	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	-.05	.00	.00	.00	.00	.05	-.10	[illegible]	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	-.12	.00	.00	.00	.00	.00	-.09	-.27	-.06	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	-.09	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00

U-VELOCITY FOR X=

.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00

V-VELOCITY FOR K = 1

.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	-.91	-.86	-.79	-.74	-.72	-.70	-.65	-.53	-.33	-.39	-.03	.07	.00
.00	.00	.00	.00	.00	.00	.00	-.61	-.67	-.68	-.67	-.67	-.65	-.61	-.58	-.46	.08	.00	.33	.00
.00	.00	.00	.00	.00	-.56	-.36	-.40	-.51	-.57	-.61	-.63	-.64	-.65	-.65	-.64	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	-.52	-.43	-.45	-.50	-.55	-.60	-.64	-.70	-.75	-.75	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	-.52	-.42	-.40	-.45	-.50	-.56	-.63	-.73	-.87	-.97	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	-.31	-.27	-.32	-.39	-.46	-.53	-.60	-.73	-.93	-1.26	-1.45	-1.04	-.63	.00
.00	.00	.00	-.53	-.21	.04	.06	-.08	-.22	-.33	-.42	-.50	-.56	-.60	-.62	.00	.02	-.83	.00	.00
.00	.00	.00	-.15	.33	.43	.26	.04	-.15	-.28	-.39	-.49	-.57	-.61	.02	.00	.03	-.65	.00	.00
.00	.00	.00	1.73	1.36	.83	.38	.07	-.14	-.27	-.38	-.51	-.63	-.67	.00	.00	.03	.00	.00	.00
.00	7.80	-8.34	5.06	2.19	.83	.26	-.03	-.22	-.34	-.43	-.58	-.75	-.81	.00	.00	.06	.00	.00	.00
.00	.00	.00	.00	.00	.00	-.31	-.30	-.48	-.51	.00	.00	.00	-.83	-.91	-.65	.08	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	-.63	-.64	.00	.00	.00	-.67	-.75	-.70	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	-.72	.00	.00	.00	.00	.00	-.67	-.66	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	-.75	.00	.00	.00	.00	.00	-.62	-.61	-.56	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	-.74	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00

V-VELOCITY FOR K= 2

.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	-.09	-.03	.04	.09	.11	.13	.19	.30	.50	.73	.79	.87	.00
.00	.00	.00	.00	.00	.00	.00	.22	.17	.17	.17	.18	.20	.24	.30	.36	.00	.00	.00	.00
.00	.00	.00	.00	.00	.23	.45	.43	.34	.28	.24	.22	.21	.20	.20	.18	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.37	.41	.40	.35	.30	.25	.21	.15	.10	.07	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.31	.42	.44	.40	.35	.25	.22	.12	-.02	-.14	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.52	.57	.53	.46	.39	.32	.25	.15	-.06	-.43	-.63	-.22	.16	.00
.00	.00	.00	.26	.60	.86	.90	.76	.63	.52	.43	.36	.28	.24	.19	.00	.00	.01	.80	.00
.00	.00	.00	.60	1.13	1.25	1.10	.88	.70	.56	.46	.35	.27	.22	.00	.00	.00	.14	.00	.00
.00	.00	.00	2.27	2.09	1.64	1.22	.92	.71	.57	.47	.33	.21	.16	.00	.00	.00	.00	.00	.00
.00	7.33	7.68	5.08	2.75	1.60	1.08	.81	.62	.50	.39	.25	.08	.01	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.60	.52	.43	.32	.00	.00	.08	-.01	-.09	.16	.00	.08	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.19	.17	.00	.00	.00	.14	.02	.12	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.09	.00	.00	.00	.00	.00	.15	.16	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.05	.00	.00	.00	.00	.00	.18	.20	.24	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.05	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00

V-VELOCITY FOR K= 3

.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.12	.23	.31	.36	.38	.40	.45	.57	.76	.95	1.01	1.04	.00
.00	.00	.00	.00	.00	.00	.00	.47	.47	.48	.49	.50	.51	.55	.68	.61	.00	.00	.00	.00
.00	.00	.00	.00	.00	.38	.68	.72	.65	.59	.56	.54	.53	.52	.51	.43	.08	.00	.00	.00
.00	.00	.00	.00	.00	.00	.54	.70	.70	.66	.51	.57	.53	.48	.41	.83	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.55	.72	.75	.71	.68	.61	.54	.45	.30	.13	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.76	.86	.83	.77	.70	.64	.57	.47	.25	-.16	-.42	-.01	.31	.00
.00	.00	.00	.42	.81	1.09	1.17	1.06	.93	.83	.74	.66	.60	.54	.41	.00	.08	.19	.00	.00
.00	.00	.00	.45	1.32	1.51	1.38	1.18	1.00	.87	.77	.67	.58	.47	.00	.00	.00	.29	.00	.00
.00	.00	.00	1.69	2.35	1.84	1.49	1.21	1.01	.87	.77	.63	.50	.40	.00	.00	.00	.00	.00	.00
6.84	3.69	3.47	2.91	2.22	1.62	1.33	1.39	.92	.79	.64	.50	.33	.26	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.69	.75	.70	.55	.00	.00	.00	.23	.15	.35	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.41	.36	.00	.00	.00	.34	.35	.36	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.28	.00	.00	.00	.00	.50	.38	.39	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.23	.00	.00	.00	.00	.00	.37	.42	.39	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.21	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00

V-VELOCITY FOR K = 4

.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	-.12	.22	.30	.34	.36	.38	.42	.51	.67	.83	.87	.93	.00
.00	.00	.00	.00	.00	.00	.00	.41	.42	.43	.45	.46	.47	.50	.54	.54	.00	.00	.00	.00
.00	.00	.00	.00	.00	.34	.58	.61	.56	.52	.50	.49	.48	.48	.46	.39	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.47	.59	.63	.57	.54	.51	.48	.44	.38	.31	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.47	.61	.64	.62	.58	.54	.49	.42	.30	.15	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.65	.73	.71	.66	.61	.56	.51	.44	.26	-.08	-.32	.04	.00	.00
.00	.00	.00	.33	.66	.91	.98	.89	.79	.71	.64	.58	.53	.48	.38	.00	.00	.26	.00	.00
.00	.00	.00	.39	1.03	1.24	1.15	.98	.84	.74	.66	.58	.51	.42	-.00	.00	.00	.27	.00	.00
.00	.00	.00	.90	1.51	1.48	1.23	1.00	.85	.74	.55	.55	.45	.36	.00	.00	.00	.00	.00	.00
.00	-2.43	-.15	1.11	1.43	1.50	1.09	.91	.77	.67	.56	.44	.30	.24	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.58	.64	.60	.48	.00	.00	.00	.22	.15	[illegible]	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.36	.32	.00	.00	.00	.32	.33	.32	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.25	.00	.00	.00	.00	.00	.35	.35	.06	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.22	.00	.00	.00	.00	.00	.34	.37	.35	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.20	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00

V-VELOCITY FOR K= 5

.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00

TEMPERATURES FOR K=   1

****	****	****	****	****	****	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.95	9.94	9.94	9.94
****	****	****	****	****	****	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.95	9.94	9.94	9.94
****	****	****	****	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94
****	****	****	****	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	****	****	****
****	****	****	****	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	****	****	****
****	****	****	****	****	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94
****	****	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94
****	****	9.95	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94
****	****	10.33	9.97	9.96	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	****
13.16	11.78	10.50	10.09	9.97	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	****	9.94	9.94	9.94	****
14.25	12.32	11.11	10.35	10.02	9.95	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	****	****	****
15.16	11.81	10.69	10.15	9.96	9.95	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	****	****	****
****	****	****	****	****	9.94	9.94	9.94	9.94	9.94	9.94	****	9.94	9.94	9.94	9.94	9.94	****	****	****
****	****	****	****	****	****	****	9.94	9.94	9.94	9.94	****	9.94	9.94	9.94	9.94	9.94	9.94	****	****
****	****	****	****	****	****	****	9.94	9.94	9.94	****	****	****	9.94	9.94	9.94	9.94	9.94	****	****
****	****	****	****	****	****	****	9.94	9.94	9.94	****	****	****	9.94	9.94	9.94	9.94	9.94	****	****
****	****	****	****	****	****	****	9.94	9.94	9.94	****	****	****	****	****	****	****	****	****	****

TEMPERATURES FOR K= 2

						9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98
						9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98
				9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98
				9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98			
				9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98			
					9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98
		9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98
		9.99	9.99	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98
		10.08	10.02	9.99	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	
13.51	11.94	10.57	10.14	10.01	9.99	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98		9.98	9.98	9.98	
15.19	12.86	11.24	10.40	10.07	9.99	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98			
13.51	11.97	10.77	10.21	10.03	9.99	9.98	9.93	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98	9.98			
					9.98	9.98	9.98	9.98	9.98	9.98		9.98	9.98	9.98	9.98	9.98			
							9.98	9.98	9.98	9.98		9.98	9.98	9.98	9.98	9.98	9.98		
							9.98	9.98	9.98				9.98	9.98	9.98	9.98	9.98		
							9.98	9.98	9.98				9.98	9.98	9.98	9.98	9.98		
							9.98	9.98	9.98										

TEMPERATURES FOR K= 1

*****	*****	*****	*****	*****	*****	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00
*****	*****	*****	*****	*****	*****	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00
*****	*****	*****	*****	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00
*****	*****	*****	*****	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	*****	*****	*****
*****	*****	*****	*****	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	*****	*****	*****
*****	*****	*****	*****	*****	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00
*****	*****	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00
*****	*****	10.01	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00
*****	*****	10.08	10.03	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	*****
13.66	11.92	10.52	10.12	10.02	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	*****	10.00	10.00	10.00	*****
18.40	13.02	11.06	10.29	10.05	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	*****	*****	*****
13.67	11.95	10.69	10.18	10.03	10.00	11.00	10.00	10.08	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.08	*****	*****	*****
*****	*****	*****	*****	*****	10.00	10.00	10.00	10.00	10.00	10.00	*****	10.00	10.00	10.00	10.00	10.00	*****	*****	*****
*****	*****	*****	*****	*****	*****	*****	10.00	10.00	10.00	10.00	*****	10.00	10.00	10.00	10.00	10.00	10.00	*****	*****
*****	*****	*****	*****	*****	*****	*****	10.00	10.00	10.00	*****	*****	*****	10.00	10.00	10.00	10.00	10.00	*****	*****
*****	*****	*****	*****	*****	*****	*****	10.00	10.08	10.00	*****	*****	*****	10.00	10.00	10.00	10.80	10.03	*****	*****
*****	*****	*****	*****	*****	*****	*****	10.00	10.00	10.00	*****	*****	*****	*****	*****	*****	*****	*****	*****	*****

TEMPERATURES FOR K= 4

```
****** ****** ****** ****** ****** ****** 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00
****** ****** ****** ****** ****** ****** 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00
****** ****** ****** ****** 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00
****** ****** ****** ****** 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 ****** ****** ******
****** ****** ****** ****** 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 ****** ****** ******
****** ****** ****** ****** ****** 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00
****** ****** 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00
****** ****** 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00
****** ****** 10.04 10.01 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 ******
11.48 10.82 10.22 10.05 10.01 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 ****** 10.00 10.00 10.00 ******
12.14 11.04 10.33 10.09 10.02 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 ****** ****** ******
11.49 10.83 10.28 10.07 10.01 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 ****** ****** ******
****** ****** ****** ****** ****** 10.00 10.00 10.00 10.00 10.00 10.00 ****** 10.00 10.00 10.00 10.00 10.00 ****** ****** ******
****** ****** ****** ****** ****** ****** ****** 10.00 10.00 10.00 10.00 ****** 10.00 10.00 10.00 10.00 10.00 10.00 ****** ******
****** ****** ****** ****** ****** ****** ****** 10.00 10.00 10.00 ****** ****** ****** 10.00 10.00 10.00 10.00 10.00 ****** ******
****** ****** ****** ****** ****** ****** ****** 10.00 10.00 10.00 ****** ****** ****** 10.00 10.00 10.00 10.00 10.00 ****** ******
****** ****** ****** ****** ****** ****** ****** 10.00 10.00 10.00 ****** ****** ****** ****** ****** ****** ****** ****** ****** ******
```

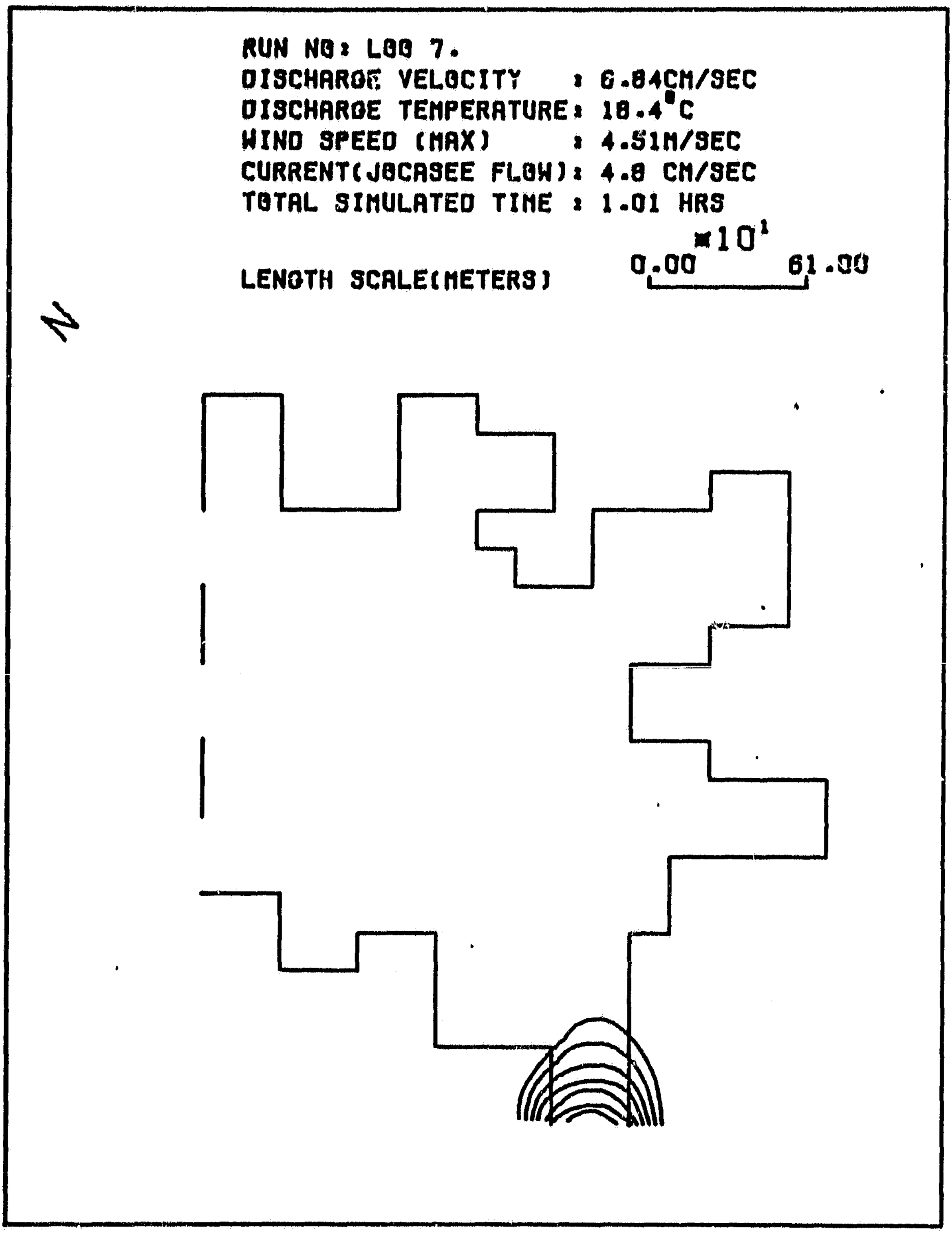
TEMPERATURES FOR K= 5

```
************************************ 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00
************************************ 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00
************************ 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00
************************ 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00******************
************************ 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00******************
****************************** 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00
************ 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00
************ 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00
************ 10.02 10.01 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00******
10.74 10.42 10.11 10.02 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00****** 10.00 10.00 10.00******
10.92 10.50 10.16 10.04 10.01 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00******************
10.74 10.43 10.14 10.03 10.01 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00******************
****************************** 10.00 10.00 10.00 10.00 10.00 10.00****** 10.00 10.00 10.00 10.00 10.00******************
****************************************** 10.00 10.00 10.00 10.00****** 10.00 10.00 10.00 10.00 10.00 10.00************
****************************************** 10.00 10.00 10.00****************** 10.00 10.00 10.00 10.00 10.00************
****************************************** 10.00 10.00 10.00****************** 10.00 10.00 10.00 10.00 10.00************
****************************************** 10.00 10.00 10.00******************************************************************
*    1.073003,    -.720000,   -1.670000,     .000000,   75.000000,    1.007945,  -1,  1
*    9.941427,    4.135714,    9.729611,     .432561,   -1.670000,    9.663010,   1,  2
*    9.939636,    4.134813,    9.729611,     .432538,   -1.670000,    9.662703,   1,  2
*    9.937851,    4.133925,    9.729611,     .432516,   -1.670000,    9.662398,   1,  2
*    9.936058,    4.133034,    9.729611,     .432494,   -1.670000,    9.662092,   1,  2
*    9.934295,    4.122147,    9.729611,     .432472,   -1.670000,    9.661789,   1,  2
*    9.932534,    4.131267,    9.729611,     .432450,   -1.670000,    9.661487,   1,  2
*    9.930791,    4.170395,    9.729611,     .432428,   -1.670000,    9.661188,   1,  2
*    9.929070,    4.129535,    9.729611,     .432407,   -1.670000,    9.660894,   1,  2
*    9.927373,    4.128687,    9.729611,     .432386,   -1.670000,    9.660604,   1,  2
*    9.925705,    4.127852,    9.729611,     .432365,   -1.670000,    9.660318,   1,  2
                                                        -1.670000     9.660033,   1,  2
```

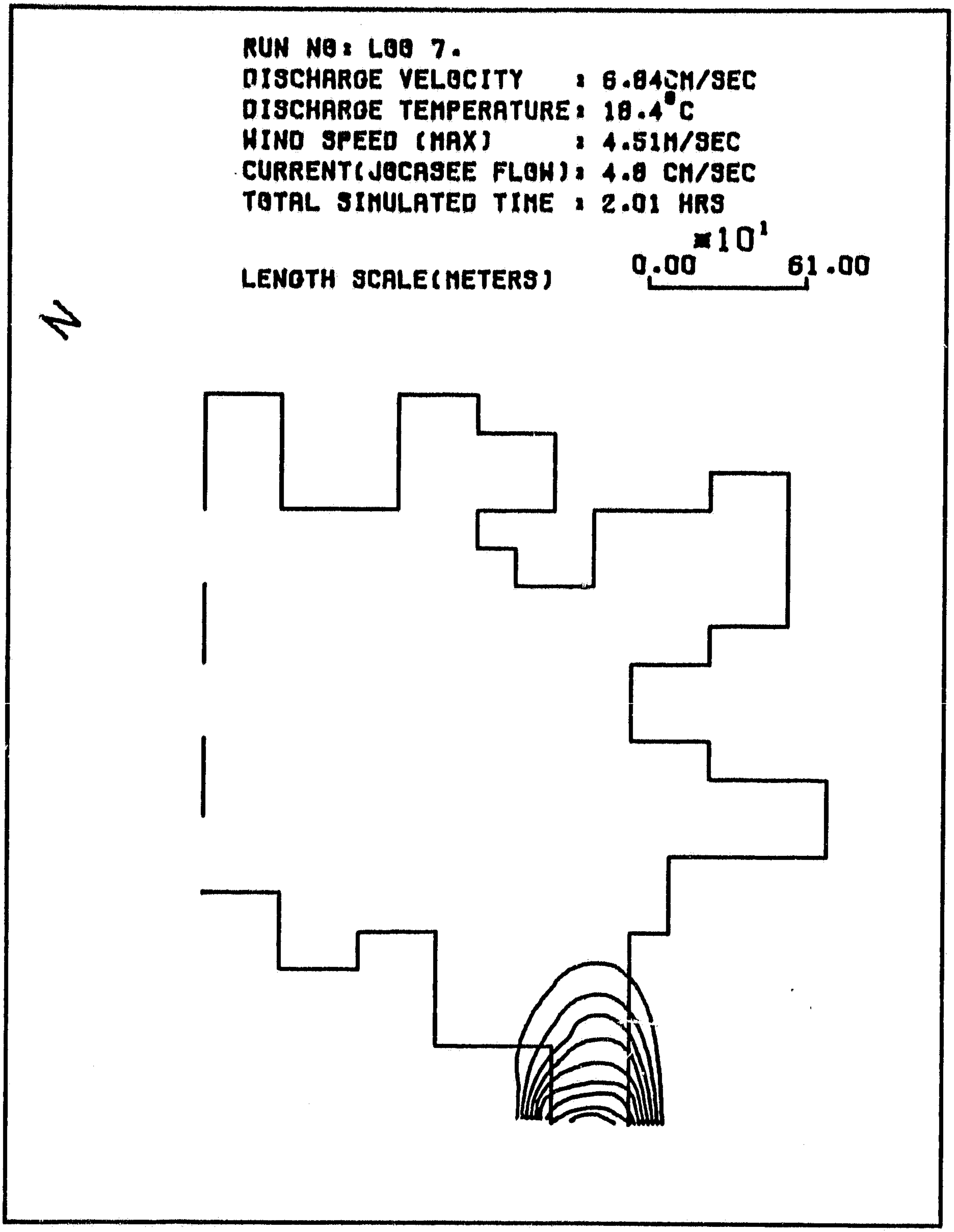
```
B=   -.1900000-004
C=   -.4600000-005
TE=   -.1167000+001
TC=    .1000000+002

Ilm=   4      Ex=    .9894957-003
```

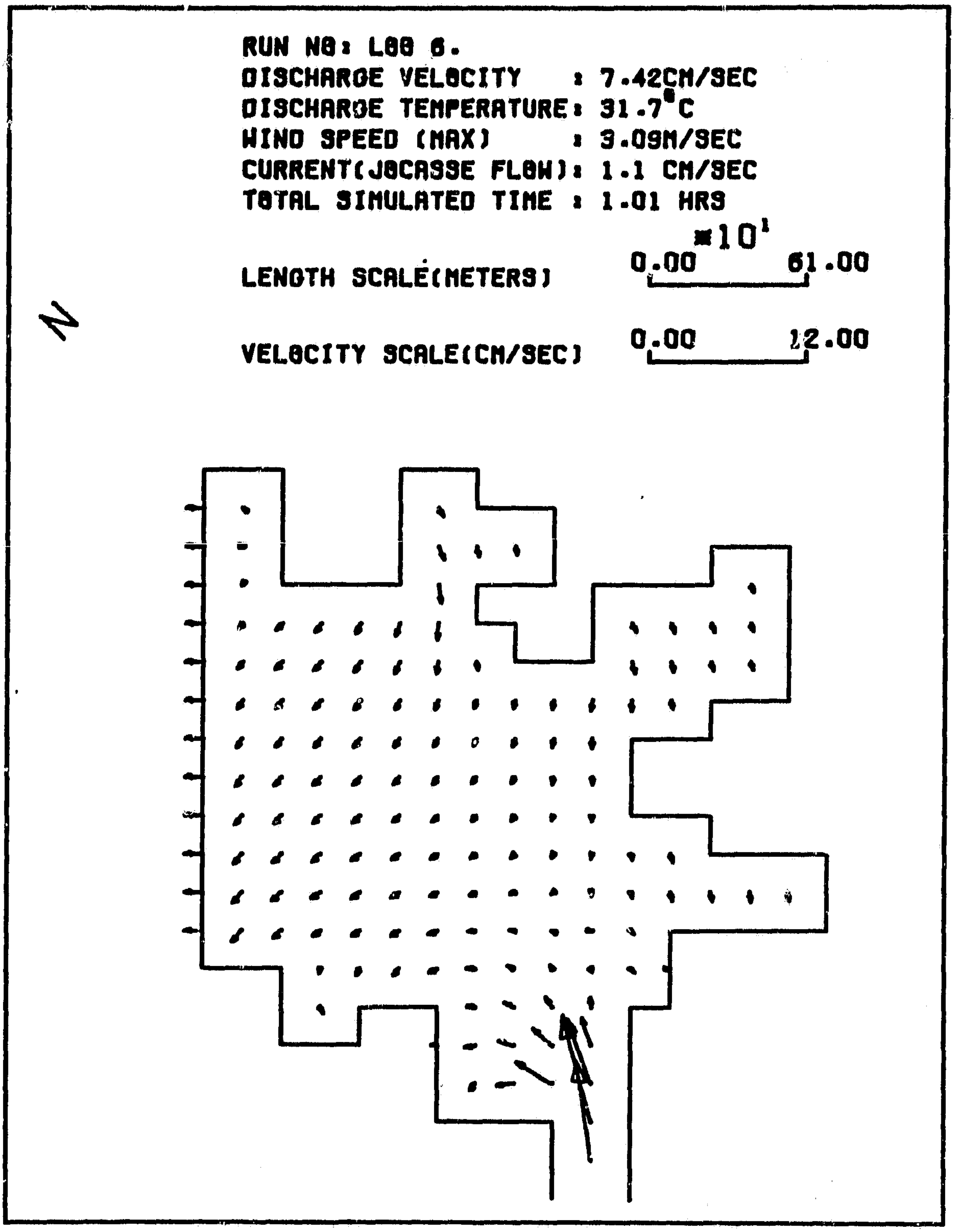
SAMPLE PLOTS



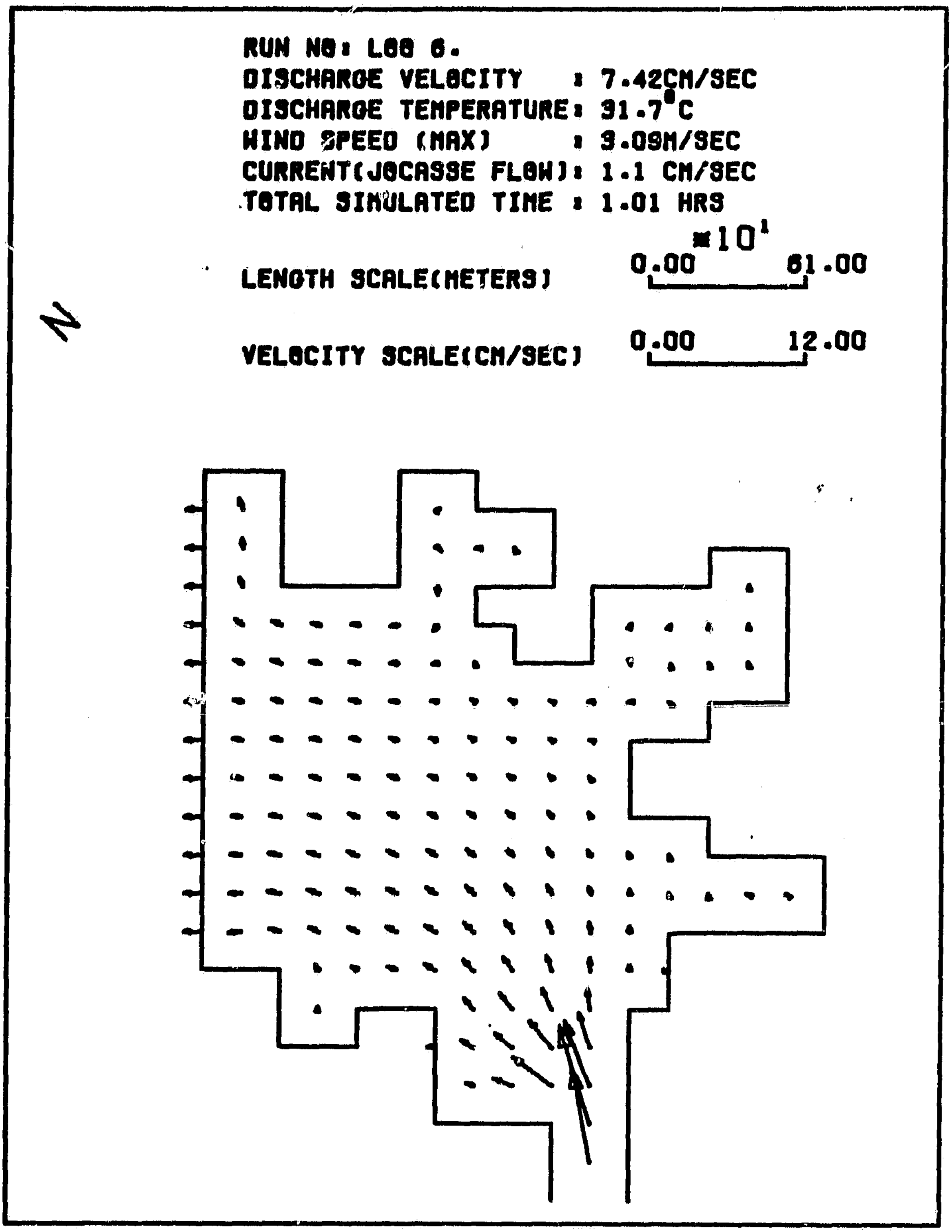
ISOTHERMS AT K= 1.
LAKE KEOWEE-(RIGID-LID MODEL)
SIMULATIONS FOR FEB. 27 1979



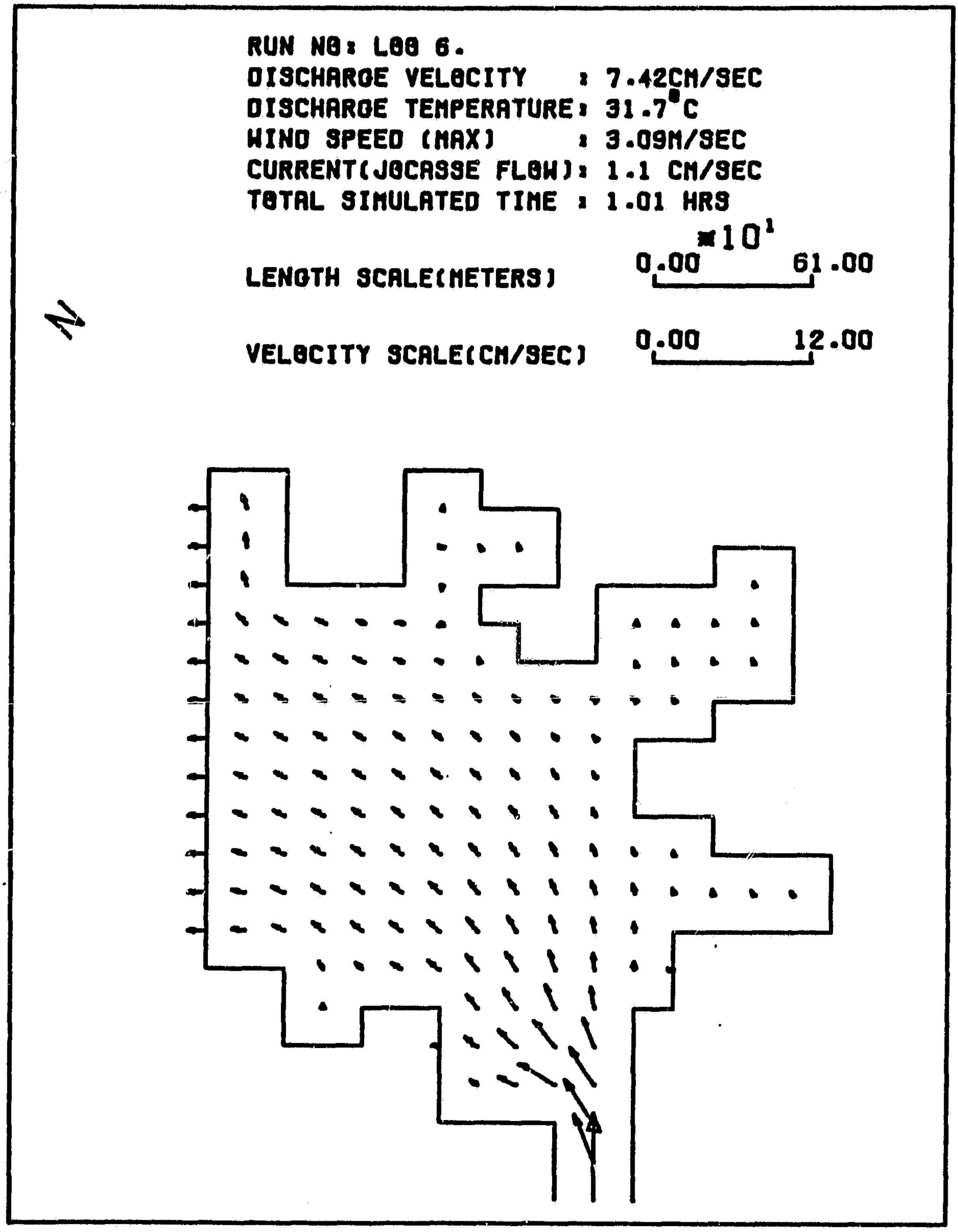
ISOTHERMS AT K= 1.
LAKE KEOWEE-(RIGID-LID MODEL)
SIMULATIONS FOR FEB. 27 1979



VELOCITIES AT K= 1.
LAKE KEOWEE-(RIGID-LID MODEL)
SIMULATIONS FOR FEB. 27 1979



VELOCITIES AT K= 2.
LAKE KEOWEE-(RIGID-LID MODEL)
SIMULATIONS FOR FEB. 27 1979

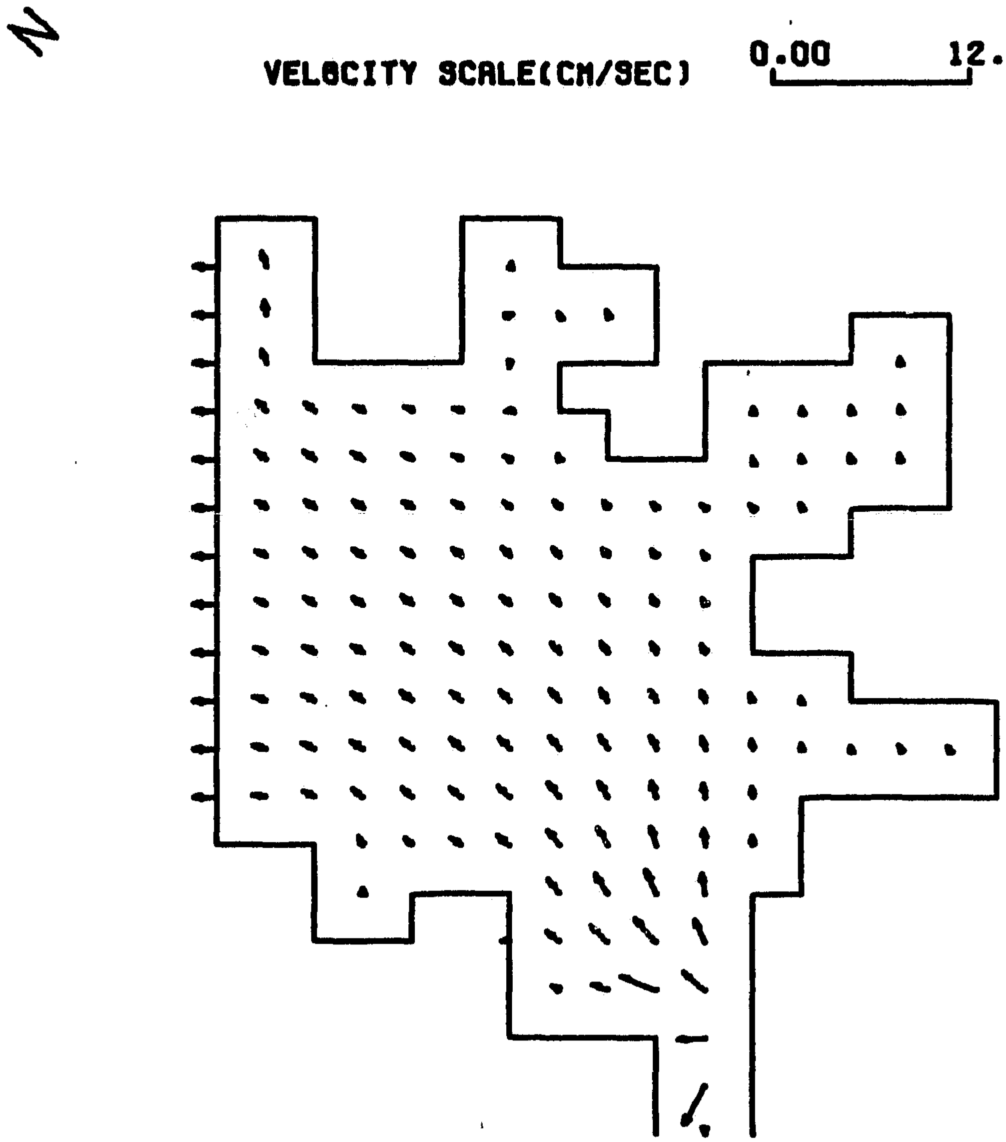


VELOCITIES AT K= 3.
LAKE KEOWEE-(RIGID-LID MODEL)
SIMULATIONS FOR FEB. 27 1979

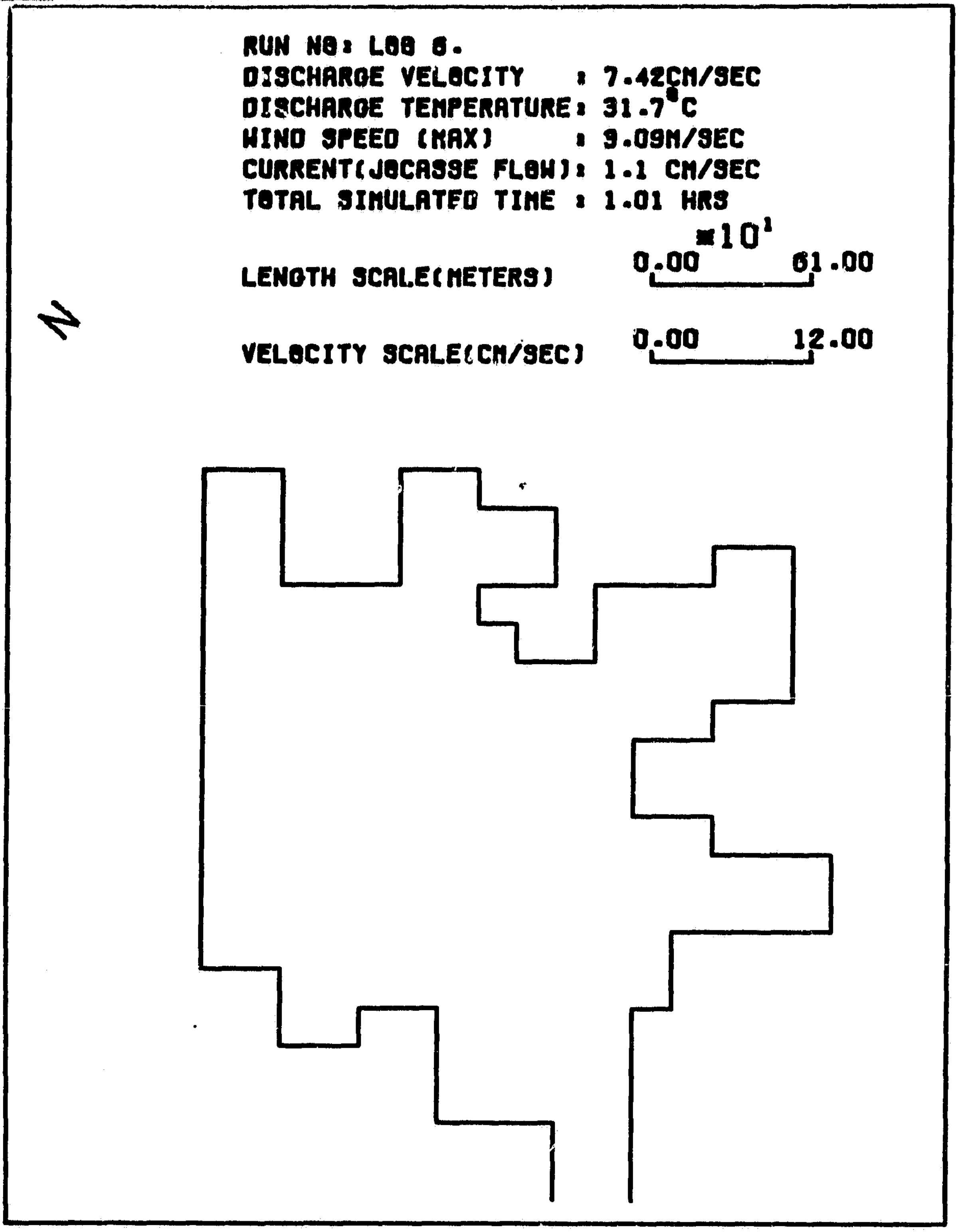
RUN NO: L88 6.
DISCHARGE VELOCITY : 7.42CM/SEC
DISCHARGE TEMPERATURE: 31.7°C
WIND SPEED (MAX) : 3.09M/SEC
CURRENT(JOCASSE FLOW): 1.1 CM/SEC
TOTAL SIMULATED TIME : 1.01 HRS

LENGTH SCALE(METERS) $\times 10^1$ 0.00 61.00

VELOCITY SCALE(CM/SEC) 0.00 12.00



VELOCITIES AT K= 4.
LAKE KEOWEE-(RIGID-LID MODEL)
SIMULATIONS FOR FEB. 27 1979



VELOCITIES AT K= 5.
LAKE KEOWEE-(RIGID-LID MODEL)
SIMULATIONS FOR FEB. 27 1979