(NASA-CR-165623-Vol-5) THERMAL POLLUTION MATHEMATICAL MODEL. VOLUME 5: USER'S MANUAL FOR THREE-DIMENSIONAL RIGID-LID MODEL Final Report (Miami Univ.) 154 p HC A08/MF A01 CSCL 13B G3/43

N81-23558

Unclas 23983

THERMAL POLLUTION MATHEMATICAL MODEL

(Volume Five of Seven Volumes)

USER'S MANUAL FOR THREE-DIMENSIONAL RIGID-LID MODEL

Volume V

by

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

Prepared for

National Aeronautics and Space Administration
and
Environmental Protection Agency
(NASA Contract NAS 10-9410)

August 1980

STANDARD TITLE PAGE

1. Report No. CR-165623	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Thermal Pollution Math Model Volume V, User's Manual for Three-Dimensional Rigid-Lid Model		5. Report Date
		6. Performing Organization Code TR 43-4
7. Author(s) Drs. Samuel Lee and Subrata Sengupta; Emmanuel V. Nwadike and Sumon K. Sinha		8. Performing Organization Report No.
9. Performing Organization Name and Address John F. Kennedy Space Center, Kennedy Space Center, Florida 32899, Project Manager: Roy A. Bland Code PT-FAP		10. Work Unit No.
		11. Contract or Grant No. NAS10-9410
		13. Type of Report and Period Covered Final
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration and Environmental Protection Agency, Research Triangle Park, N.C.		14. Sponsoring Agency Code

15. 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.

16. Key Words

Thermal Pollution
Math Models
Nuclear or Fossil Fuel Power Plants
Waste Heat

17. Bibliographic Control	18. Distribution
Announce in Star Category 43	Unlimited

19. Security Classif.(of this report)	20. Security Classif.(of this page)	21. No. of Pages	22. Price
None	None		

KSC FORM 16-272NS (REV. 7/78)

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).

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

Preface ii
Abstract iii
Figures v
Tables vi
Symbols vii
Acknowledgments viii

1. Introduction 1
2. Recommendations 2
3. Program Description and Flow Chart 3
   Description of program algorithm 3
   Flow chart 3
   Subroutine descriptions 3
4. List of Program Symbols Used in Main Program 6
   Description of main variables 6
   Marker matrices 11
5. Preparation of Runs 13
6. Input Data 14
7. Plotting Programs 15
   Description of plot programs 15
   Subroutines 15

References 16
Appendices 17

A. Example Case 18
   Introduction 18
   Problem statement 18
   Calculations of parameters and input data 20
   Sample input 22
   Lake Keowee execution deck 24
B. Fortran Source Program Listing 45
   List of main program and subroutines 46
   List of plot programs 47
   Sample run 120
   Sample plots 128

FIGURES

Number		Page
1	Flow chart (main program)	30
2	Coordinates and grid system	32
3	Map of Lake Keowee	37
4	Lake Keowee (region of interest)	38
5	MAR matrix	39
6	MRH matrix	40
7	Keowee hydro discharge, February 27, 1979	43
8	Jocassee-pumped storage station discharge, February 27, 1979	44

TABLES

Number		Page
1	Governing Equations	5
2	Subroutines for Calculations	26
3	Input Data to Main Program	33
4	Subroutines for Plots	36
5	Meteorological Data for Lake Keowee, February 27, 1979	41
6	Summary of Inflows and Outflows to Lake Keowee, February 27, 1979	42

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	Peclet 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
$(\bar{\ })$	Dimensional mean quantity
$(\ ')$	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

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

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

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

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

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

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

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

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

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

$\varepsilon = H/L$

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

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

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

$A_H^* = A_H/A_{ref}$ 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 $A_{ref} = 0.002\ L^{4/3}$

$= 38311.48\ cm^2 sec.$

For better agreement with data the value chosen is 60,000 cm^2/sec.

Reference depth = H = 16 m.

Reference vertical $A_v = 0.002 \times (H)^{4/3}$.

Eddy viscosity = 37.43 cm^2/sec.

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

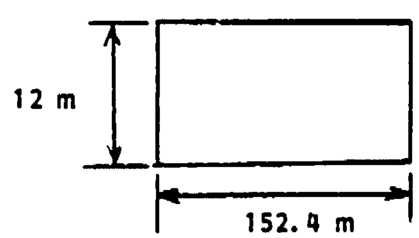
Reference temperature = T_{ref} = 10.0°C.

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:

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

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 $Q = (152.4 \times 12 \times V)\ m^3/sec$

where V = discharge velocity (m/sec)

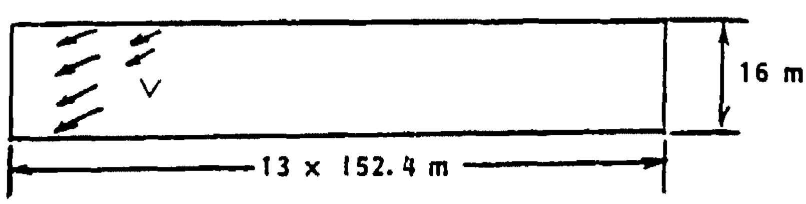
$$\therefore V = [Q/(152.4 \times 12)] \text{ m/sec} = \left(\frac{Q}{152.4 \times 12 \times 100}\right) \text{ cm/sec}$$

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	=	$t_{ref}/3600 = 2.6811111$
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)

<u>First Run</u>

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. $\frac{\partial u}{\partial x}$, $\frac{\partial u}{\partial \theta}$ are computed at interior, boundary or corner pts by scheme similar to the one used in DINERU.
3	DVVY	Computes D1HVVY.	Called by INTE. $\frac{\partial}{\partial \theta}$ (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. $\frac{\partial}{\partial \theta}$ (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.	

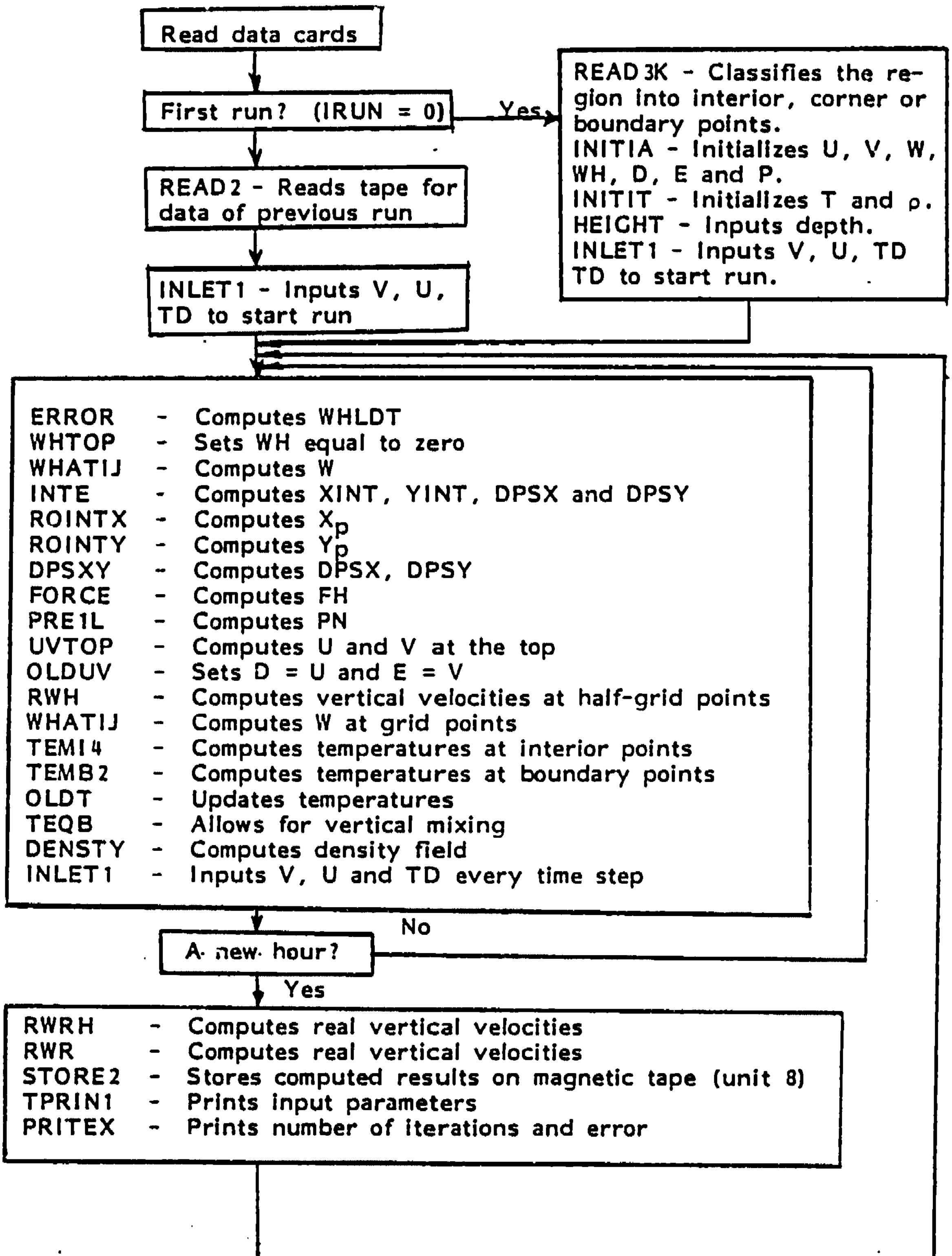


Figure 1. Flow chart (main program)

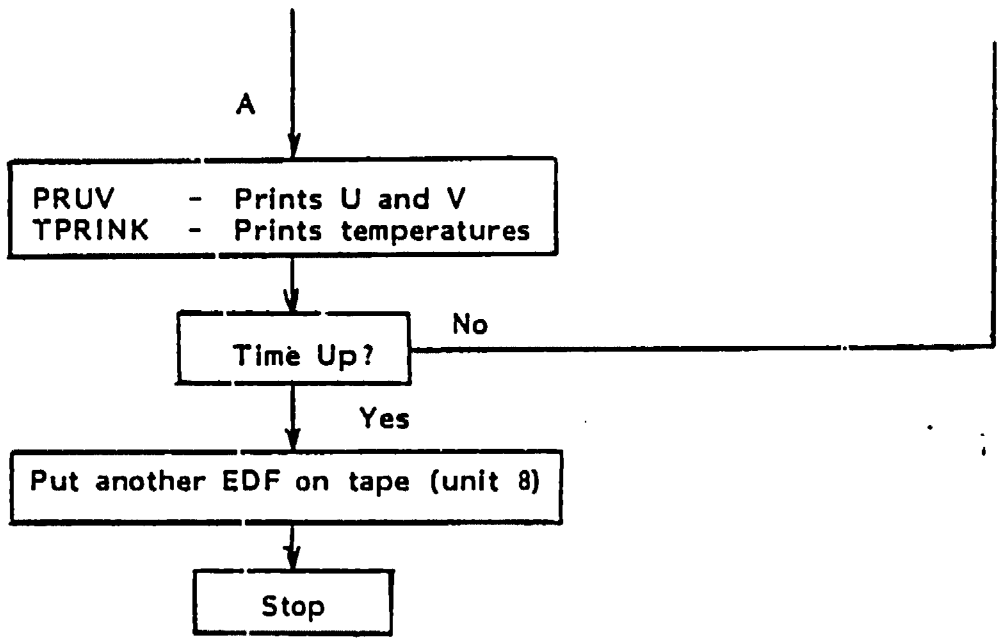


Figure 1 (Continued). Flow chart (main program)

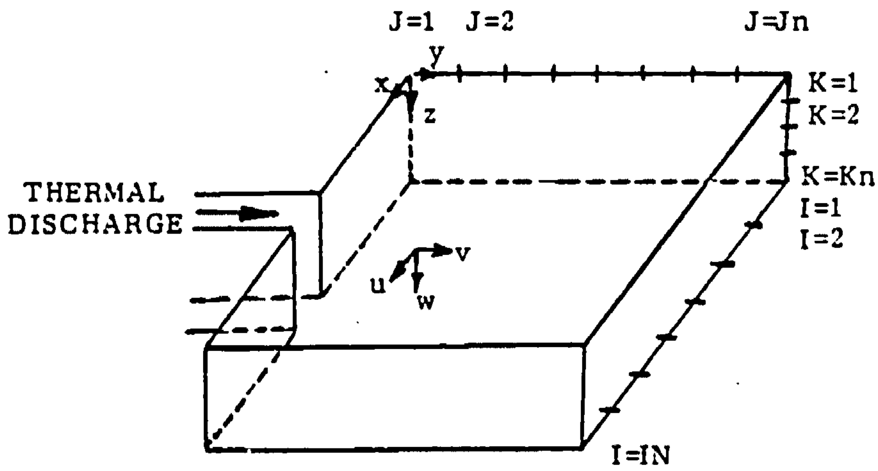


Figure 2. Coordinate and grid system

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

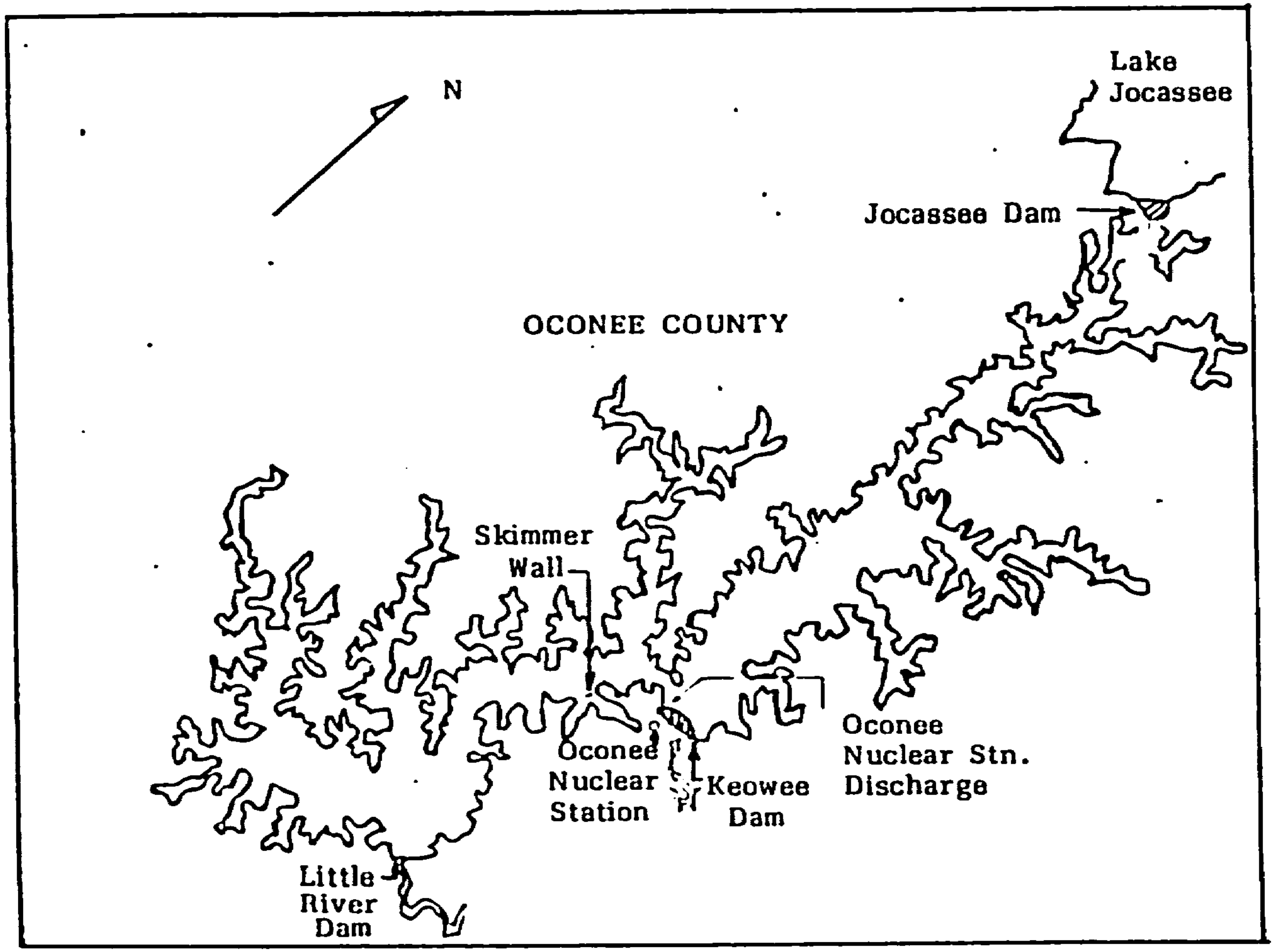


Figure 3. Lake Keowee

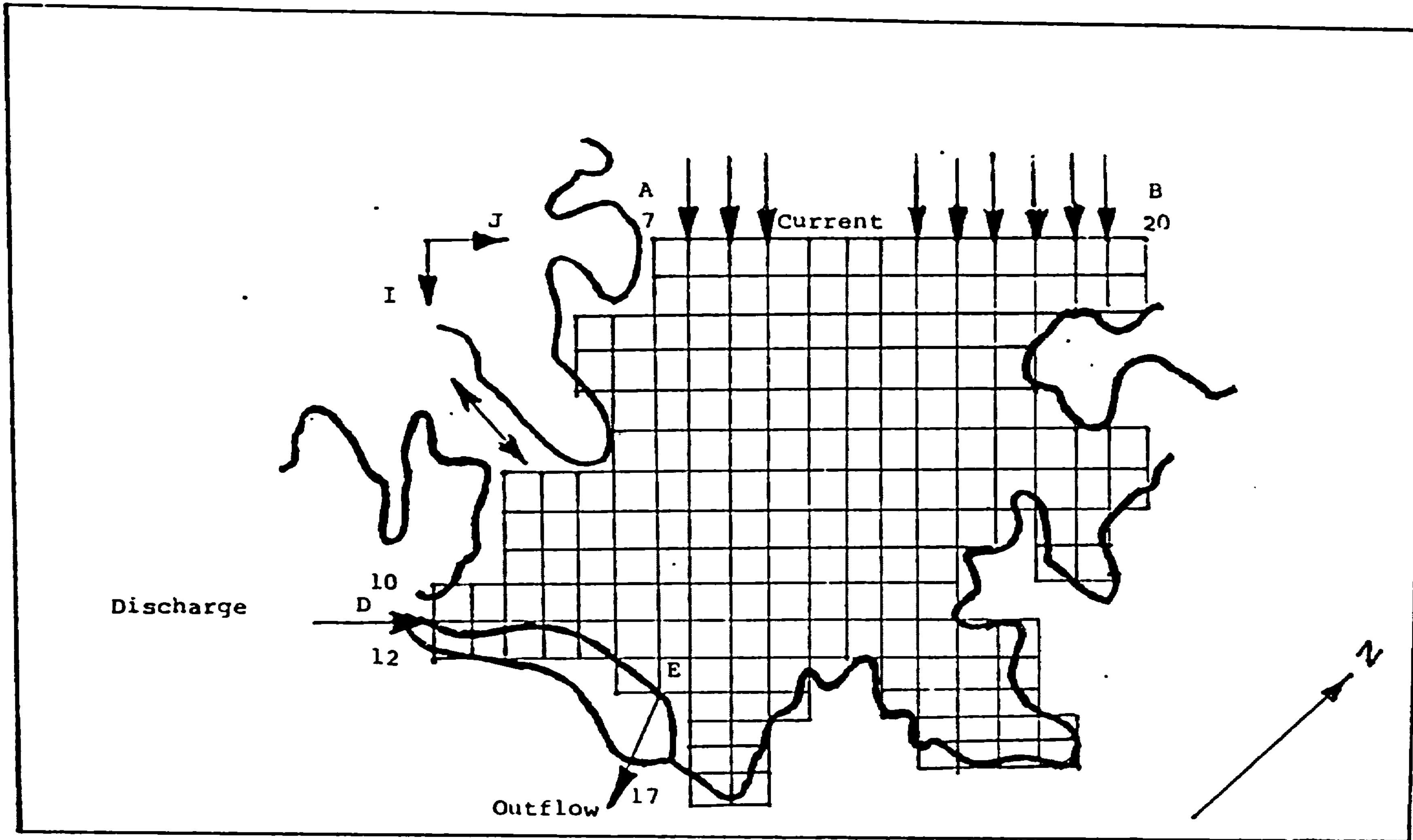


Figure 4. Lake Keowee (region of interest) showing inputs and outputs (for 3-D model)

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

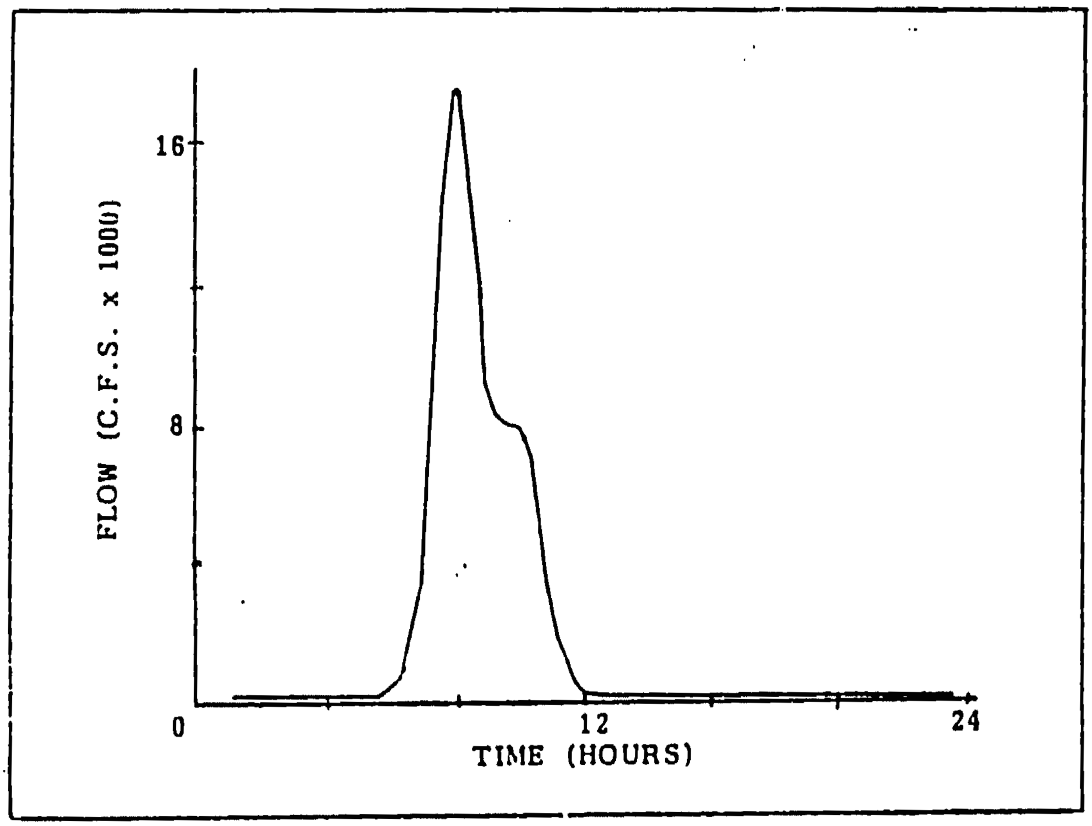


Figure 7. Keowee hydro discharge (February 27, 1979)

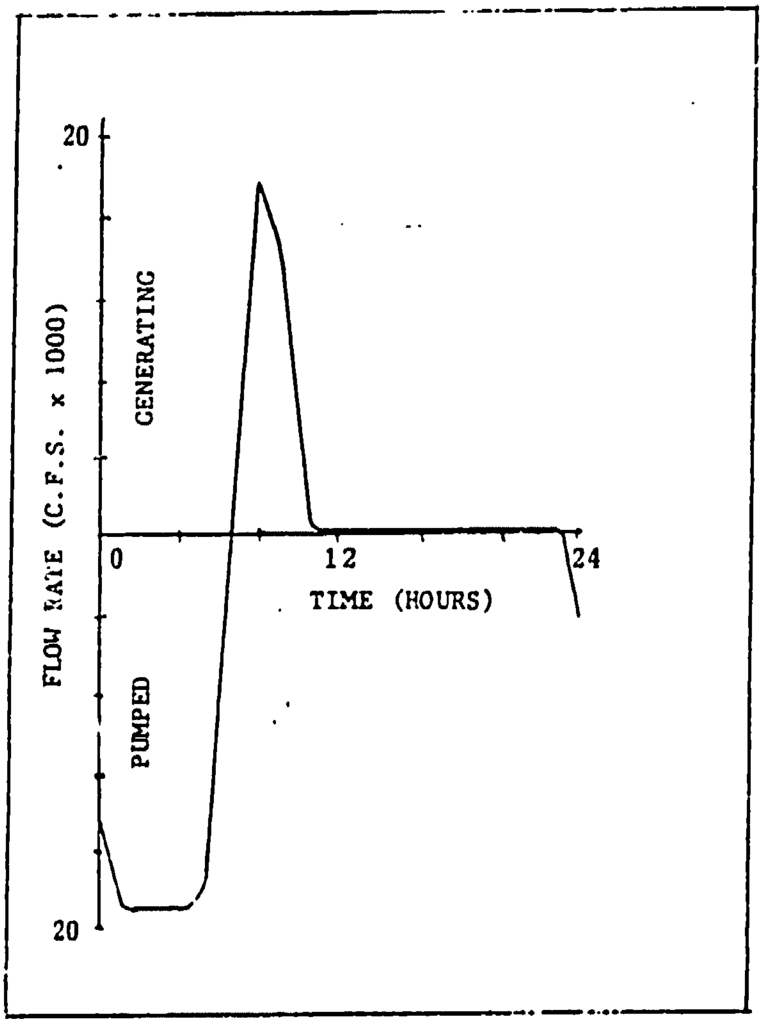


Figure 8. Jocassee-pumped storage station discharge data (February 27, 1979)

APPENDIX B

FORTRAN SOURCE PROGRAM LISTING

LIST OF MAIN PROGRAM AND SUBROUTINES

```
.ARNASA(11).THAINN FOR CREATED ON 5 MAY 83 AT 10:40:24
C*******************************************************************
C*  THIS IS THE MAIN PROGRAM FOR THE 3-D RIGID-LID MODEL           *
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),
     C WR(IN,JN,KN),WRH(IWN,JWN,KN),P(IWN,JWN),D(IN,JN,KN),E(IN,JN,KN),
     C WLOT(IWN,JWN),XINT(IN,JN),YINT(IN,JN),H(IN,JN,KN),G(IN,JN,KN),
     C HI(IN,JN),HA(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN),FHI(IWN,JWN)
     C OPSX(IN,JN),OPSY(IN,JN)
      DIMENSION A3(KN)
      DIMENSION T(IN,JN,KN),TH(IN,JN,KN),TU(IN,JN,KN),HU(IN,JN,KN),
     C RINTA(IN,JN,KN),RINTY(IN,JN,KN),WC(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 AISS
      INM1=IN-1
      READ 1,IRUN
      READ 1,LLN
      READ 1,ASTORE
    1 FORMAT()
      TTOT1=0.
      HTTOT=0.0
      READ 2, VVIS,ABP
      READ 2, AI,AM,AV,AP
      READ 2, EPS,MAXIT,OMEGA,ARBP
      READ 2, DX,DY,DZ
      READ 2, TAI,TAM,TAV
      READ 2, A,B,C
      READ 2, TO
      READ 2, CUL,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
      RHEF=A+B*TO+C*TO*TO
      IF(IRUN.GT.0) GO TO 3
      CALL READ1(I,J,IN,JN,IW,JW,IWN,JWN,MAR,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,MAR,MRH,TREF,
     CRREF,TW,ROW,TO)
      CALL HEIGHT(I,J,K,IN,JN,KN,HI,HA,HY,CC)
      CALL INLET(I,J,K,IN,JN,KN,U,V,H,G,T,TO,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,HA,HY,HI,
     CMAR,MRH,AI,AM,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAM,TAV,
     CCB,CW,A,B,C,CUL,T,TW,RO,ROW,TE,RREF,TREF,TO,TIME,TTOT)
      PRINT 333, TTOT,TAUX,TAUY
  333 FORMAT(4X,3F15.5)
      HTTOT=CTTOT*TTOT
      CALL INLET(I,J,K,IN,JN,KN,U,V,H,G,T,TO,AA,TLL,DT,HTTOT)
    4 CONTINUE
      DL2=DX*DX
      ISTOP=1
      IF(HTTOT.EQ.0.0) GO TO 666
      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,5(',',F12.6),2(',',I5))
   61 CONTINUE
  666 CONTINUE
      DO 5 LL=1,LLN
      READ 2,WS,TSU,TDEW,RADN,ISGNX,ISGNY
```

ORIGINAL PAGE IS
OF POOR QUALITY

PRECEDING PAGE BLANK NOT FILMED

```
      READ 2,ANGLE
      PRINT 151,N-5,TSU,TDEW,RADN,ANGLE,WTTOT,ISGNX,ISGNY
222   TSU=T(IN-1,JN-5,1)
      TSU=TREF*(1.+TSU)
      TTOT=TTOT+DT
      TTOT1=TTOT1+DT
      TM=(TSU+TDEW)/2.
COMMENT : THE NEXT 6 LINES ARE USED TO CALCULATE THE
C       : EQUILIBRIUM TEMPERATURE.
C
      F=19.2+0.46*WS**2
      BETAS=0.35+0.015*TM+0.0012*TM**2
      AISS=4.5+0.05*TSU+BETAS*F+0.47*F
      TAMB=TDEW+RADN/AISS
      TE=(TAMB-TREF)/TREF
C
COMMENT : THE SURFACE HEAT EXCHANGE COEFFICIENT IS
C       : NON-DIMENSIONALIZED.
C
      AHT=AISS*0.001/H
C
COMMENT : ANGLE (DEGREES) * ANGLE * 0.01745329 (RADIANS).
C
      TAUX=(0.15*(SIN(ANGLE*0.01745329))*ISGNX
      TAUY=(0.15*(COS(ANGLE*0.01745329))*ISGNY
      PRINT 161,TSU,TM,F,BETAS,TAMB,AISS,ISTOP,LL
      CALL ERROR(IN,JN,I,J,DT,DH,WHECT,KN,MAR)
      CALL WATOP(I,J,K,IN,JN,KN,WH,W,MAR)
      CALL WHAT(U,I,J,K,IN,JN,IN,JN,KN,IN,JN,W,WH,MAR)
      CALL INTE(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,MAR,XINT,YINT,AI,A,
     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 ROINT(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,
     CMAR,AI,HX,HI,XINT)
      CALL ROINTY(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,MAR,
     CRINTY,HY,YINT)
      CALL DPSX(I,J,IN,JN,I,J,IN,JN,DPSX,DPSY,P,DX,DY,MAR)
      CALL FORCE(I,J,IN,JN,XINT,YINT,WHECT,DX,DY,HI,HX,HY,MAR,
     CDPSX,DPSY,FM,AP,IN,JN,IN,JN,XINTX,XINTY,U,V,EUL,ABR,MAR,K)
      CALL PRECL(EPS,MAXIT,IN,JN,P,ITN,DPSX,DPSY,FM,DLZ,OMEGA,
     CMAR,I,J,K,I,J,DX,DY,E,IN,JN,ARBP)
      CALL UV(I,J,K,I,J,IN,JN,KN,IN,JN,U,V,D,E,M,U,DX,DY,DZ,
     CXINTX,XINTY,EUL,W,DT,AI,AP,AH,AV,A,HI,HX,HY,P,MAR)
      CALL UVTOP(H,D,TAUX,TAUY,I,J,K,D,IN,JN,KN,HI,MAR)
      CALL OLDUV(I,J,K,IN,JN,K,U,V,U2)
      CALL OLDW(I,J,K,IN,JN,W,W,W1)
      CALL WH(I,J,K,I,J,IN,JN,KN,I,J,U,V,WH,HI,DX,DY,DZ,MAR)
      CALL WHAT(U,I,J,K,I,J,IN,JN,KN,IN,JN,W,WH,MAR)
15    CONTINUE
      DO 20 I=1,IN
      DO 20 J=1,JN
      WO(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 TEMI(I,J,K,IN,JN,KN,U,V,W,T,TO,DX,
     CDY,
     CDY,DZ,DT,TAI,TAH,TAV,B3,HI,HX,HY,MAR,AHT,TREF,TAMB)
      CALL TEMB(I,J,K,IN,JN,KN,TO,DX,DY,DZ,MAR,DE,HI,AHT,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,TO,T)
      CALL TEQ(I,J,K,IN,JN,KN,T,MAR)
      CALL DENSTY(I,J,K,I,J,IN,JN,KN,IN,JN,A,B,C,MAW,MAH,I,T,
     CRO,ROW,RREF,TREF)
      DO 2010 I=9,11
      K=1
C     PRINT 9200,IC,I,K,(V(I,J,K),J=1,JN)
2010  CONTINUE
C9200  FORMAT(/' I=',I3,3X,'I=',I3,3X,'K=',/,3X' V-VELOCITY'/(15X,...
      DO 40 I=1,IN
      DO 40 J=1,JN
      W(I,J,1)=WO(I,J,1)
40    CONTINUE
      CALL TIME(I,J,K,IN,JN,KN,U,V,W,T,TO,XX,TCC,DT,WTTOT)
      TIME=TTOT1*CTTOT
```

```
      PRTT=1.
      IF(TIME.GE.PRTT) GO TO 444
      GO TO 222
  444 TTOT1=0.0
      CALL [illegible](I,J,K,IM,JM,IN,JN,KN,IMN,JMN,U,V,WH,HI,HX,HY,
     CDX,DY,DZ,[illegible])
      CALL [illegible](I,J,K,IN,JN,KN,U,[illegible],HI,HX,HY,DZ,MAR)
      IF(NSTORE.GT.0) GO TO 1000
      CALL STORE(I,[illegible])
     CHI,MAR,[illegible],AI,AM,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,[illegible],TAI,TAU,
     CTAV,AAT,CB,CW,A,B,C,EUL,T,TW,RO,ROW,TE,AREF,TREF,TC,TAMA,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 TPRINT(I,[illegible],TAM,TAV,CB,CW,AAT,TREF,AREF,EUL,A,B,C,TE,TW)
      CALL PRITE(I,[illegible])
      CALL PHUV(I,J,K,IN,JN,KN,U,V,UA,VA,MAN)
      CALL TPRINA(I,J,K,IN,JN,KN,T,RO,TREF,MAN,TACTUL)
      ISTOP=ISTOP+1
    6 CONTINUE
      END FILE 5
      END
```

```
ASA*NASA(1).CORINT FOR CREATED ON 5 MAY 90 AT 10:49:36
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.1) 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*AJ(1).DENSTY FOR CREATED ON 15 MAY 74 AT 11:30:19
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,
     *MAR,MRW,
     *T,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 13 I=1,IN
      DO 12 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
   13 CONTINUE
      DO 23 IW=1,IWN
      DO 22 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
   23 CONTINUE
      RETURN
      END
```

```
[illegible].DINERU FOR CREATED ON 5 MAY 80 AT 11:00:12
C*****************************************************************
C    THIS SUBROUTINE COMPUTES DIHUUX,DIHUVX WHICH ARE USED IN    *
C    THE POISSON EQUATION FOR PRESSURE.                          *
C*****************************************************************

      SUBROUTINE DINERU(I,J,K,IN,JN,KN,U,V,H,DX,DY,DIHUUX,DIHUVX,MAR)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),H(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)*H(I+1,J)-U(I-1,J,K)
     C*U(I-1,J,K)*H(I-1,J))/(2*DX)
      DIHUVX=(U(I+1,J,K)*V(I+1,J,K)*H(I+1,J)-U(I-1,J,K)
     C*V(I-1,J,K)*H(I-1,J))/(2*DX)
      GO TO 50
31    CONTINUE
      DIHUUX=(U(I+1,J,K)*U(I+1,J,K)*H(I+1,J)-U(I-1,J,K)
     C*U(I-1,J,K)*H(I-1,J))/(2*DX)
      DIHUVX=(U(I+1,J,K)*V(I+1,J,K)*H(I+1,J)-U(I-1,J,K)
     C*V(I-1,J,K)*H(I-1,J))/(2*DX)
      GO TO 50
32    CONTINUE
      DIHUUX=(U(I+1,J,K)*U(I+1,J,K)*H(I+1,J)-U(I-1,J,K)
     C*U(I-1,J,K)*H(I-1,J))/(2*DX)
      DIHUVX=(U(I+1,J,K)*V(I+1,J,K)*H(I+1,J)-U(I-1,J,K)
     C*V(I-1,J,K)*H(I-1,J))/(2*DX)
      GO TO 50
33    CONTINUE
      DIHUUX=(4*H(I+1,J)*U(I+1,J,K)*U(I+1,J,K)-3*H(I,J)*U(I,J,K)
     C*U(I,J,K)-H(I+2,J)*U(I+2,J,K)*U(I+2,J,K))/(2*DX)
      DIHUVX=(4*H(I+1,J)*U(I+1,J,K)*V(I+1,J,K)-3*H(I,J)*U(I,J,K)
     C*V(I,J,K)-H(I+2,J)*U(I+2,J,K)*V(I+2,J,K))/(2*DX)
      GO TO 50
34    CONTINUE
      DIHUUX=(3*H(I,J)*U(I,J,K)*U(I,J,K)-4*H(I-1,J)*U(I-1,J,K)
     C*U(I-1,J,K)+H(I-2,J)*U(I-2,J,K)*U(I-2,J,K))/(2*DX)
      DIHUVX=(3*H(I,J)*U(I,J,K)*V(I,J,K)-4*H(I-1,J)*U(I-1,J,K)
     C*V(I-1,J,K)+H(I-2,J)*U(I-2,J,K)*V(I-2,J,K))/(2*DX)
      GO TO 50
35    CONTINUE
      DIHUUX=(4*H(I+1,J)*U(I+1,J,K)*U(I+1,J,K)-3*H(I,J)*U(I,J,K)
     C*U(I,J,K)-H(I+2,J)*U(I+2,J,K)*U(I+2,J,K))/(2*DX)
      DIHUVX=(4*H(I+1,J)*U(I+1,J,K)*V(I+1,J,K)-3*H(I,J)*U(I,J,K)
     C*V(I,J,K)-H(I+2,J)*U(I+2,J,K)*V(I+2,J,K))/(2*DX)
      GO TO 50
36    CONTINUE
      DIHUUX=(U(I+1,J,K)*U(I+1,J,K)*H(I+1,J)-U(I-1,J,K)
     C*U(I-1,J,K)*H(I-1,J))/(2*DX)
      DIHUVX=(U(I+1,J,K)*V(I+1,J,K)*H(I+1,J)-U(I-1,J,K)
     C*V(I-1,J,K)*H(I-1,J))/(2*DX)
      GO TO 50
37    CONTINUE
      DIHUUX=(4*H(I+1,J)*U(I+1,J,K)*U(I+1,J,K)-3*H(I,J)*U(I,J,K)
     C*U(I,J,K)-H(I+2,J)*U(I+2,J,K)*U(I+2,J,K))/(2*DX)
      DIHUVX=(4*H(I+1,J)*U(I+1,J,K)*V(I+1,J,K)-3*H(I,J)*U(I,J,K)
     C*V(I,J,K)-H(I+2,J)*U(I+2,J,K)*V(I+2,J,K))/(2*DX)
      GO TO 50
38    CONTINUE
      DIHUUX=(U(I+1,J,K)*U(I+1,J,K)*H(I+1,J)-U(I-1,J,K)
     C*U(I-1,J,K)*H(I-1,J))/(2*DX)
      DIHUVX=(U(I+1,J,K)*V(I+1,J,K)*H(I+1,J)-U(I-1,J,K)
     C*V(I-1,J,K)*H(I-1,J))/(2*DX)
      GO TO 50
39    CONTINUE
      DIHUUX=(3*H(I,J)*U(I,J,K)*U(I,J,K)-4*H(I-1,J)*U(I-1,J,K)
     C*U(I-1,J,K)+H(I-2,J)*U(I-2,J,K)*U(I-2,J,K))/(2*DX)
      DIHUVX=(3*H(I,J)*U(I,J,K)*V(I,J,K)-4*H(I-1,J)*U(I-1,J,K)
     C*V(I-1,J,K)+H(I-2,J)*U(I-2,J,K)*V(I-2,J,K))/(2*DX)
      GO TO 50
```

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

```
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 TERMS OF POISSON'S EQUATION AT HALF GRID PO-  *
C  INTS                                                     *
C***********************************************************
      SUBROUTINE DPSXY(I,J,IN,JN,IW,JW,IN,JN,DPSX,DPSY,P,DX,DY,MARI
      DIMENSION P(IW,JW),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
```

```
******FILE 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,H,DY,DIHUVY,MAR)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),H(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)*H(I,J+1)-U(I,J-1,K)
     C*V(I,J-1,K)*H(I,J-1))/(2*DY)
      GO TO 50
31    CONTINUE
      DIHUVY=(3*H(I,J)*U(I,J,K)*V(I,J,K)-4*H(I,J-1)*U(I,J-1,K)
     C*V(I,J-1,K)+H(I,J-2)*U(I,J-2,K)*V(I,J-2,K))/(2*DY)
      GO TO 50
32    CONTINUE
      DIHUVY=(4*H(I,J+1)*U(I,J+1,K)*V(I,J+1,K)-3*H(I,J)*
     CU(I,J,K)*V(I,J,K)-H(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)*H(I,J+1)-U(I,J-1,K)
     C*V(I,J-1,K)*H(I,J-1))/(2*DY)
      GO TO 50
34    CONTINUE
      DIHUVY=(U(I,J+1,K)*V(I,J+1,K)*H(I,J+1)-U(I,J-1,K)
     C*V(I,J-1,K)*H(I,J-1))/(2*DY)
      GO TO 50
35    CONTINUE
      DIHUVY=(3*H(I,J)*U(I,J,K)*V(I,J,K)-4*H(I,J-1)*U(I,J-1,K)
     C*V(I,J-1,K)+H(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)*H(I,J+1)-U(I,J-1,K)
     C*V(I,J-1,K)*H(I,J-1))/(2*DY)
      GO TO 50
37    CONTINUE
      DIHUVY=(4*H(I,J+1)*U(I,J+1,K)*V(I,J+1,K)-3*H(I,J)*
     CU(I,J,K)*V(I,J,K)-H(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)*H(I,J+1)-U(I,J-1,K)
     C*V(I,J-1,K)*H(I,J-1))/(2*DY)
      GO TO 50
39    CONTINUE
      DIHUVY=(4*H(I,J+1)*U(I,J+1,K)*V(I,J+1,K)-3*H(I,J)*
     CU(I,J,K)*V(I,J,K)-H(I,J+2)*U(I,J+2,K)*V(I,J+2,K))/(2*DY)
      GO TO 50
40    CONTINUE
      DIHUVY=(3*H(I,J)*U(I,J,K)*V(I,J,K)-4*H(I,J-1)*U(I,J-1,K)
     C*V(I,J-1,K)+H(I,J-2)*U(I,J-2,K)*V(I,J-2,K))/(2*DY)
50    CONTINUE
      RETURN
      END
```

```
DVISU.FOR CREATED ON 5 MAY 80 AT 11:10:49
C*****************************************************************
C     THIS SUBROUTINE COMPUTES D1UX,D2UX & D1UY                  *
C*****************************************************************
      SUBROUTINE DVISU(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1UX,D2UX,D1UY,D2U
     *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
      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)
   60 CONTINUE
      RETURN
      END
```

```
DVISV.FOR CREATED ON 5 MAY 80 AT 11:13:50
C*****************************************************
C     THIS SUBROUTINE COMPUTES D1VY,D2VY,D1VX & D2VX      *
C*****************************************************
      SUBROUTINE DVISV(I,J,K,IN,JN,KN,U,V,H,DX,DY,D1VX,D2VX,D1VY,D2VY
     C,MAR)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),H(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
```

```
      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=(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
```

```
[illegible] DIHVVY FOR CREATED ON 6 MAY 80 AT 11:16:10
C     *****************************************************
C     THIS SUBROUTINE COMPUTES DIHVVY                     *
C     *****************************************************
      SUBROUTINE DVVY(I,J,K,IN,JN,KN,U,V,H,DY,DIHVVY,MAR)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),H(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)*H(I,J+1)-V(I,J-1,K)*
     CV(I,J-1,K)*H(I,J-1))/(2*DY)
      GO TO 50
31    CONTINUE
      DIHVVY=(3*H(I,J)*V(I,J,K)*V(I,J,K)+H(I,J-2)*V(I,J-2,K)
     C*V(I,J-2,K)-4*H(I,J-1)*V(I,J-1,K)*V(I,J-1,K))/(2*DY)
      GO TO 50
32    CONTINUE
      DIHVVY=(4*H(I,J+1)*V(I,J+1,K)*V(I,J+1,K)-3*H(I,J)*V(I,J,K)
     C*V(I,J,K)-H(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)*H(I,J+1)-V(I,J-1,K)*
     CV(I,J-1,K)*H(I,J-1))/(2*DY)
      GO TO 50
34    CONTINUE
      DIHVVY=(V(I,J+1,K)*V(I,J+1,K)*H(I,J+1)-V(I,J-1,K)*
     CV(I,J-1,K)*H(I,J-1))/(2*DY)
      GO TO 50
35    CONTINUE
      DIHVVY=(3*H(I,J)*V(I,J,K)*V(I,J,K)+H(I,J-2)*V(I,J-2,K)
     C*V(I,J-2,K)-4*H(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)*H(I,J+1)-V(I,J-1,K)*
     CV(I,J-1,K)*H(I,J-1))/(2*DY)
      GO TO 50
37    CONTINUE
      DIHVVY=(4*H(I,J+1)*V(I,J+1,K)*V(I,J+1,K)-3*H(I,J)*V(I,J,K)
     C*V(I,J,K)-H(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)*H(I,J+1)-V(I,J-1,K)*
     CV(I,J-1,K)*H(I,J-1))/(2*DY)
      GO TO 50
39    CONTINUE
      DIHVVY=(4*H(I,J+1)*V(I,J+1,K)*V(I,J+1,K)-3*H(I,J)*V(I,J,K)
     C*V(I,J,K)-H(I,J+2)*V(I,J+2,K)*V(I,J+2,K))/(2*DY)
      GO TO 50
40    CONTINUE
      DIHVVY=(3*H(I,J)*V(I,J,K)*V(I,J,K)+H(I,J-2)*V(I,J-2,K)
     C*V(I,J-2,K)-4*H(I,J-1)*V(I,J-1,K)*V(I,J-1,K))/(2*DY)
50    CONTINUE
      RETURN
      END
```

```
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,DIUWZ,
     CA3,TAUX,TAUY,DIVWZ,DIUZ,D2UZ,DIVZ,D2VZ,DIA3Z)
      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
      DIUZ=(U(I,J,K+1)-U(I,J,K-1))/(2*DZ)
      DIVZ=(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)
      DIA3Z=(A3(K+1)-A3(K-1))/(2.*DZ)
      DIUWZ=(U(I,J,K+1)*W(I,J,K+1)-U(I,J,K-1)*W(I,J,K-1))/(2.*DZ)
      DIVWZ=(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
      DIUZ=HI(I,J)*TAUX
      DIVZ=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)
      DIA3Z=(4.*A3(K+1)-3.*A3(K)-A3(K+2))/(2.*DZ)
      DIUWZ=(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)*
     CW(I,J,K+2))/(2.*DZ)
      DIVWZ=(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)*
     CW(I,J,K+2))/(2.*DZ)
      GO TO 63
   62 CONTINUE
      DIUZ=(3.*U(I,J,K)-4.*U(I,J,K-1)+U(I,J,K-2))/(2.*DZ)
      DIVZ=(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)
      DIA3Z=(3.*A3(K)-4.*A3(K-1)+A3(K-2))/(2*DZ)
      DIUWZ=(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)
      DIVWZ=(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
```

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

```
...FORCE.FOR CREATED ON 5 MAY 83 AT 11:19:29
      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,
           COPSX,OPSY,FW,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
           CN,JN),HY(IN,JN),OPSX(IN,JN),OPSY(IN,JN),FW(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.1) GO TO 90
            OPSX(I,J)=OPSX(I,J)-EUL*RINTX(I,J,K)+V(I,J,K)*ABR
            OPSY(I,J)=OPSY(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
            OPSXH=(OPSX(I,J)+OPSX(I+1,J)+OPSX(I,J+1)+OPSX(I+1,J+1))/4.0
            OPSYH=(OPSY(I,J)+OPSY(I+1,J)+OPSY(I,J+1)+OPSY(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
            FW(IW,JW)=(1./AP)*(-(1./HH)*(DXINT+DYINT)-WHLDT(IW,JW)-(AP/HH)*
           C(HXH*OPSXH+HYH*OPSYH))
      9     CONTINUE
      10    CONTINUE
            RETURN
            END
```

```
*****(FILE).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,8E14.7)
            RETURN
            END
```

```
[illegible],INITIA FOR CREATED ON 14 MAY 79 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,
       CARPP)
        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)=WBP
        DO 200 K=1,KN
        WH(IW,JW,K)=0
200     CONTINUE
        RETURN
        END
```

```
**SSYNSSAC(1).INITT FOR CREATED ON 15 MAY 79 AT 11:35:25
      C************************************************************************
      C       THIS PROGRAM INITIALIZES TEMP AND DENSITY
      C************************************************************************
            SUBROUTINE INITT(I,J,K,IN,JN,KN,IW,JW,IWN,JWN,A,B,C,T,RO,
           CMAR,MAW,
           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),MAW(IWN,JWN)
            TOO=(TO-TREF)/TREF
            R=A+B*TO+C*TO*TO
            ROO=(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)=TOO
            RO(I,J,K)=ROO
       11   CONTINUE
       12   CONTINUE
       10   CONTINUE
            DO 20 IW=1,IWN
            DO 20 JW=1,JWN
            IF (MAW(IW,JW).EQ.0) GO TO 22
            DO 21 K=1,KN
            TW(IW,JW,K)=TOO
            ROW(IW,JW,K)=ROO
       21   CONTINUE
       22   CONTINUE
       20   CONTINUE
            RETURN
            END
```

```
SA(1).INLET1 ELT CREATED ON 5 MAY 80 AT 11:34:12
C***********************************************************
C          THIS SUBROUTINE FOR INLET AND OUTLETS FOR
C                         DOMAIN
C***********************************************************
      SUBROUTINE INLET1(I,J,K,IN,JN,KN,U,V,W,G,T,TO,AA,TLL,DT,HTTOT)
      DIMENSION W(IN,JN,KN),G(IN,JN,KN),U(IN,JN,KN)
      DIMENSION V(IN,JN,KN),T(IN,JN,KN),TO(IN,JN,KN)
      TSOT=HTTOT
      TSOT1=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
      W(11,1,3)=0.0
      T(11,1,3)=TLL
      TO(11,1,3)=TLL
10    CONTINUE
      SF=0.00322579
      PI=3.141592653
      AJOCSE=0.5475
      BJOCSE=11.4525
      IF(TSOT.GE.0.0.AND.TSOT.LE.1.0)SV=SF*(-14.395-(18.75-14
     C.395)*TSOT)
      IF(TSOT.GE.1.0.AND.TSOT.LE.5.0)SV=SF*(-18.75))
      IF(TSOT.GE.5.0.AND.TSOT.LE.8.0)SV=SF*(((16.823+
     C18.75)/3.)*(TSOT-5.0)-18.75))
      IF(TSOT.GE.8.0.AND.TSOT.LE.11.0)SV=-SF*(((16.823-
     C0.1)/3.)*(TSOT-8.0)-16.823))
      IF(TSOT.GE.11.0.AND.TSOT.LE.23.0)SV=SF*0.1
      IF(TSOT.GE.23.0.AND.TSOT.LE.24.0)SV=-SF*(((4.5+0.1))
     C*TSOT-0.1))
C
COMMENT : STOP CHANGE.
C
      DO 20 K=1,KNM1
      DO 20 J=1,JNM1
      U(1,J,K)=SV
      W(1,J,K)=SV
      V(1,J,K)=0.0
      G(1,J,K)=0.0
      T(1,J,K)=T(2,J,K)
      TO(1,J,K)=TO(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)
      W(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(TSOT.GE.0.0.AND.TSOT.LE.6.0)SV1=0.048*SX
      IF(TSOT.GE.6.0.AND.TSOT.LE.9.0)SV1=SX*(((17.54-0.048)/3.)*
     C(TSOT-6.0)+0.048)
      IF(TSOT.GE.9.0.AND.TSOT.LE.12.0)SV1=SX*(((.048-17.54)/3.
     C)*(TSOT-9.0)+17.54)
      IF(TSOT.GE.12.0.AND.TSOT.LE.24.0)SV1=SX*0.048
45    DO 40 K=1,3
      U(13,7,K)=SV1
      W(13,7,K)=SV1
      T(13,7,K)=T(12,7,K)
      TO(13,7,K)=TO(12,7,K)
40    CONTINUE
      T(13,7,K)=T(12,7,K)
```

```
      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
```

```
*****INTE.FOR CREATED ON 5 MAY 80 AT 11:36:17
C***********************************************************
C   THIS SUBROUTINE COMPUTES XINT,YINT,OPSX & OPSY.
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,OPSX,OPSY,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 OPSX(IN,JN),OPSY(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 DIVHUU(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*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
      D1UT=(U(I,J,K)-D(I,J,K))/DT
      D1VT=(V(I,J,K)-E(I,J,K))/DT
      Q=2.0/DZ
      OPSX(I,J)=(1./AP)*(1./HI(I,J))*(-XSUM*Q-HI(I,J)*D1UT)
      OPSY(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
```

```
[illegible] SYM CREATED ON 19 NOV 77 AT 09:57:00
0.03178,0.79
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
1.00043,-0.000017,-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
0,0,0,0,0,0,2,11,11,11,11,11,11,11,11,11,11,11,11,1
0,0,0,0,7,3,8,11,11,11,11,11,11,11,11,11,6,4,4,10
0,0,0,0,2,11,11,11,11,11,11,11,11,11,11,11,1,0,0,0
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
0,0,7,3,3,3,11,11,11,11,11,11,11,11,11,11,11,11,11,1
0,0,2,11,11,11,11,11,11,11,11,11,11,11,11,6,8,11,6,10
0,0,2,11,11,11,11,11,11,11,11,11,11,11,1,6,10,2,11,1,0
7,3,8,11,11,11,11,11,11,11,11,11,11,11,1,0,3,4,10,0
2,11,11,11,11,11,11,11,11,11,11,11,11,11,1,6,3,5,0,0,0
9,4,4,4,4,5,11,11,11,11,6,4,8,11,11,11,1,0,0,0
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
0,0,0,0,0,2,9,9,9,9,9,9,9,9,1,0,0,0
0,0,0,0,0,2,9,9,9,9,9,9,9,9,3,10,10,3
0,0,4,10,10,9,9,9,9,9,9,9,9,9,9,8,3,9,7
0,0,2,9,9,9,9,9,9,9,9,9,9,7,0,2,1,0
0,0,2,9,9,9,9,9,9,9,9,9,9,1,0,0,6,7,0
4,10,9,9,9,9,9,9,9,9,9,9,1,0,0,0,0,0
6,6,8,8,8,9,9,9,9,9,8,8,9,9,10,3,7,0,0
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,-3.33,-2.79,0.0,-1,1
15.0
1.073,-0.72,-1.67,0.0,-1,1
75.0
2.335,-1.61,-1.61,0.0,-1,1
50.0
1.565,-2.22,-2.29,0.0,-1,1
15.0
2.055,-1.33,-1.39,0.0,-1,1
50.0
1.783,-2.17,-2.22,0.0,-1,1
35.0
2.012,-2.72,-2.78,20.94,1,1
35.0
2.250,-1.67,-2.79,195.39,-1,-1
60.0
4.625,0.01,-3.33,369.85,1,-1
5.0
1.785,3.06,-2.22,544.91,-1,-1
75.0
1.607,5.83,-2.22,655.96,1,-1
15.0
1.768,8.83,-1.33,725.75,1,1
40.0
3.127,11.06,-2.78,746.69,1,1
80.0
2.593,12.28,-5.0,704.41,1,1
70.0
1.520,13.39,-5.56,579.20,-1,1
30.0
1.237,13.89,-5.56,393.81,1,1
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
```

```
2.058,11.72,-4.44,0.0,-1,-1
30.0
1.162,9.72,-2.78,0.0,-1,-1
35.0
2.772,8.33,5.28,0.0,1,-1
55.0
3.661,7.78,5.56,0.0,1,-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
```

```
[illegible].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
```

```
.*SAHNASAI(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
```

```
1SA*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 15
      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,J
      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,
     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+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
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
```

```
*S3*.3(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 PRITEX(ITN,EX)
              PRINT 5500,ITN,EX
      5500    FORMAT(/' ITN=',I4,5X,' EX=',E15.7)
              RETURN
              END
```

```
NISAI(1).PROV SYM CREATED ON 5 MAY 80 AT 11:49:16
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(//,12F6.2)
      RETURN
      END
```

```
$SANASA(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,WM,PINTH,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,
       C-H,HT,HI,MAR,MRM,AI,AM,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRM,
       CTAI,TAM,TAV,AKT,CP,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),
       CWM(IWN,JWN,KN),PINTH(IWN,JWN)
        DIMENSION HY(IN,JN),HT(IN,JN),HI(IN,JN),MAR(IN,JN),MRM(IWN,JWN),
       CW(IN,JN,KN),WR(IN,JN,KN),WRM(IWN,JWN,KN)
        DIMENSION T(IN,JN,KN),RO(IN,JN,KN),TW(IWN,JWN,KN),ROW(IWN,JWN,K
        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(((WM(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(((WRM(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),((HA(I,J),J=1,JN),I=1,IN),((HY(I,J),
       CJ=1,JN),I=1,IN),((MAR(I,J),J=1,JN),I=1,IN),((MRM(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,TAM,TAV,AKT,CP,CW,A,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,TAM
       CAI,AM,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,TTOT
    1   CONTINUE
        RETURN
        END
```

```
*JAWS1SA(1).READ3W ELT CREATED ON 5 MAY 80 AT 11:50:20
C*****************************************************************
C  THIS SUBROUTINE READS AND PRINTS THE MAR & MAR MATRICES
C*****************************************************************
C
      SUBROUTINE READ3W(I,J,IN,JN,IW,JW,IWN,JWN,MAR,MRW)
      DIMENSION MAR(IN,JN),MRW(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,(MRW(IW,JW),JW=1,JWN)
      PRINT 3,IW,(MRW(IW,JW),JW=1,JWN)
400   CONTINUE
2     FORMAT()
3     FORMAT(/15X,30I4)
      RETURN
      END
```

```
*SANASAI(1).ROINTX FOR CREATED ON 5 MAY 83 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,HY,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*
      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*
      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*
      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*
      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,N
      RSUMK=(RINTX(I,J,K)*WINT(I,J)**(1)*(Z/2)*AP*EUL-H(I,J)) [illegible]
      RINT(I,J)=RINT(I,J)+RSUMK
  310 CONTINUE
  301 CONTINUE
  300 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE RINTY FOR    CREATED ON 5 MAY 80 AT 11:54:03
C*************************************************************
C THIS SUBROUTINE COMPUTES YF 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
      AY=DZ*(RO(I,J+1,K)+RO(I,J+1,K-1)-RO(I,J-1,K)-RO(I,J-1,K-1))/(4*
      GO TO 102
   11 CONTINUE
      AY=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
      AY=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
      AY=DZ*(RO(I,J+1,K)+RO(I,J+1,K-1)-RO(I,J-1,K)-RO(I,J-1,K-1))/(4*
      GO TO 102
   14 CONTINUE
      AY=DZ*(RO(I,J+1,K)+RO(I,J+1,K-1)-RO(I,J-1,K)-RO(I,J-1,K-1))/(4*
      GO TO 102
   15 CONTINUE
      AY=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
      AY=DZ*(RO(I,J+1,K)+RO(I,J+1,K-1)-RO(I,J-1,K)-RO(I,J-1,K-1))/(4*
      GO TO 102
   17 CONTINUE
      AY=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
      AY=DZ*(RO(I,J+1,K)+RO(I,J+1,K-1)-RO(I,J-1,K)-RO(I,J-1,K-1))/(4*
      GO TO 102
   19 CONTINUE
      AY=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
      AY=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)+AY*HI(I,J)*(RO(I,J,K)+RO(I,J,K-1))*
     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 (MARI(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*COL*H(I,J)
      YINT(I,J)=YINT(I,J)+RSUMY
  310 CONTINUE
  301 CONTINUE
  300 CONTINUE
      RETURN
      END
```

ISA(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 KO=1,KNM1
      K=KN-KO+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
```

```
*SA*N*SA(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
```

```
*SA*NASA(1).RWRH 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 RWRH(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
```

```
*SA*N3SA(1).STORE2 FOR CREATED ON 5 MAY 80 AT 12:32:30
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,AM,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,
     CTAI,TAM,TAV,AKT,CB,CW,A,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,TAMA
     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
     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,TAM,TAV,AKT,CB,CW,A,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,TAMA
     CAI,AM,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,TTOT
      END FILE 8
      RETURN
      END
```

```
*SAMNASA(1).TEMB2 FOR CREATED ON 5 MAY 80 AT 12:04:20
      C*************************************************************
      C   THIS SUBROUTINE COMPUTES BOUNDARY TEMPERATURES
      C*************************************************************
            SUBROUTINE TEMB2(I,J,K,IN,JN,KN,TO,DX,DY,DZ,MAR,CB,HI,AKT,CW,
           *TAMB,HX,HY,T,TREF,TAV,TAI,TAM,D2,DT)
            DIMENSION T(IN,JN,KN),TO(IN,JN,KN),MAR(IN,JN),HX(IN,JN),HY(IN,J
           CH2(IN,JN)
            KNM1=KN-1
            DO 100 K=1,KN
            DO 100 I=1,IN
            DO 100 J=1,JN
            DIMTUX=0.0
            DIMTVY=0.0
            DITWZ=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*H(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/H(I,J))*(-TA*(D1HTUX+D1HTY)*H(I,J)*D1T-[illegible])+TA*
     C*(D1TX*HX(I,J)+D2TX*H(I,J)+D1TY*HY(I,J)+D2TY*H(I,J))+(TA/H(I,
     C))*B3*D2TZ)*DT+T(I,J,K)
16    CONTINUE
15    CONTINUE
300   CONTINUE
100   CONTINUE
      RETURN
      END
```

*S1*NEW(1).TEMIN FOR CREATED ON 5 MAY 80 AT 12:07:14

```
C*************************************************************
C   THIS SUBROUTINE COMPUTES TEMPERATURES AT INTERIOR
C   GRID POINTS.
C*************************************************************
C
      SUBROUTINE TEMIN(I,J,K,IN,JN,KN,U,V,T,TD,DX,
     CCB,
     CDY,DZ,W,DT,TAI,TAH,TAV,AJ,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 9 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))*AJ*D2TZ)*DT+T(I,J,K)
8     CONTINUE
9     CONTINUE
10    CONTINUE
      RETURN
      END
```


C-2

```
**MAS(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
```

```
TSAMSSATL1.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 KEWOEE
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'/15X,8E15.7)
12    FORMAT(' DENSITY'/15X,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
```

```
*ACAIII.UVI FOR CREATED ON 5 MAY 80 AT 12:13:03
C******************************************************************
C   THIS SUBROUTINE COMPUTES U AND V FOR VARIABLE DENSITY
C******************************************************************
C
      SUBROUTINE UV(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,D,E,H,G,DX,DY,DZ,
     CRINTX,RINTY,EUL,
     C
     CDT,AT,AP,AH,AV,A3,HT,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),HT(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.1) GO TO 9
      DO 8 K=2,KNM1
      DIPX=(P(IW,JW)-P(IW-1,JW)+P(IW,JW-1)-P(IW-1,JW-1))/(2*DX)
     C+EUL*RINTX(I,J,K)
      DIPY=(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=(HT(I,J)+A*HT(I,J)+B/HT(I,J))
      DIHUUX=(U(I+1,J,K)*U(I+1,J,K)*HT(I+1,J)-U(I-1,J,K)
     C*U(I-1,J,K)*HT(I-1,J))/(2*DX)
      DIHUVY=(U(I,J+1,K)*V(I,J+1,K)*HT(I,J+1)-U(I,J-1,K)
     C*V(I,J-1,K)*HT(I,J-1))/(2*DY)
      DIHUVX=(U(I+1,J,K)*V(I+1,J,K)*HT(I+1,J)-U(I-1,J,K)
     C*V(I-1,J,K)*HT(I-1,J))/(2*DX)
      DIHVVY=(V(I,J+1,K)*V(I,J+1,K)*HT(I,J+1)-V(I,J-1,K)*
     CV(I,J-1,K)*HT(I,J-1))/(2*DY)
      DIUX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
      DIUY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
      DIVX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
      DIVY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
      DIUWZ=(U(I,J,K+1)*W(I,J,K+1)-U(I,J,K-1)*W(I,J,K-1))/(2*DZ)
      DIVWZ=(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)*(-AT*(DIHUUX+DIHUVY+HT(I,J)*DIUWZ)-HT(I,J)*AP
     C*DIPX+AH*HT(I,J)*(DDUX+DDUY)+AH*HX(I,J)*DIUX+AH*HY(I,J)*DIUY
     C+AV*A3(K)*DDUZ/HT(I,J))+HT(I,J)*U(I,J,K)/C
      G(I,J,K)=(DT/C)*(-AT*(DIHUVX+DIHVVY+HT(I,J)*DIVWZ)-HT(I,J)*AP
     C*DIPY+AH*HT(I,J)*(DDVX+DDVY)+AH*HX(I,J)*DIVX+AH*HY(I,J)*DIVY
     C+AV*A3(K)*DDVZ/HT(I,J))+HT(I,J)*V(I,J,K)/C
    8 CONTINUE
    9 CONTINUE
   10 CONTINUE
      RETURN
      END
```

```
*ISA*NICAIII).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 700 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
```

```
WHATIJ.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
```

```
*ASA*ASA(II).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,MRM)
            DIMENSION WH(IWN,JWN,KN)
            DIMENSION MRM(IWN,JWN)
            DO 3300 IW=1,IWN
            DO 3300 JW=1,JWN
            IF (MRM(IW,JW).EQ.0) GO TO 3000
            WH(IW,JW,1)=0
       3000 CONTINUE
       3300 CONTINUE
            RETURN
            END
```

LISTINGS OF PLOT PROGRAMS

```
[illegible] PROGRAM PLOTT [illegible]
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     P10 : 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=23,IWN=16,JWN=19,KN=5,KNM1=4
      INTEGER PGRID
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
     CW(IWN,JWN,KN),Q(IN,JN,KN),R(IN,JN,KN),WRM(IWN,JWN,KN),
     CH(IN,JN),HX(IN,JN),HY(IN,JN),HAR(IN,JN),HRH(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,W,PINTH,I,J,K,IW,JW,IN,JN,KN,
     CIWN,JWN,D,F,HX,HY,H,HAR,HRH,II,AH,AV,AP,DX,
     CDY,DZ,DT,TAUX,TAUY,WUR,WRM,TAI,TAM,TAV,ANT,
     CCB,CW,A,B,C,EUL,T,TW,RO,ROW,TE,PREF,TREF,TC,
     CTIME,TTOT)
6     CONTINUE
      CALL PLOTS(IBUF,1000,11)
      CALL PLOT(0.0,3.0,-3)
1     FORMAT()
      A=1
      P5=A
      P6=CTTOT*TTOT
      N1=0.5
      N2=0.3
      START=10.5
      DO 333 JJ=1,JN
      DO 333 II=1,IN
      H(II,JJ)=TREF*(1.+T(II,JJ,1))
      IF(HAR(II,JJ).EQ.0)GO TO 333
      IF(H(11,15).GT.10.5)START=11.0
333   CONTINUE
      CALL PLOT(7.0,0.0,-3)
      CALL ECHO3N(H,IN,JN,1,IN,1,JN,4.75,8.0,3.04,START,0.5,PGRID,
     CIN,JN,START,19.0,0.0,0.0,0.0,0.0,N1,N2,0.07,1.0,NPLT)
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(12.0,1.0,2)
      CALL PLOT(15.0,6.0,2)
      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(14.0,18.0,2)
      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(8.0,19.0,2)
      CALL PLOT(10.0,19.0,2)
      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(1.0,17.0,2)
      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(8.0,5.0,2)
      CALL PLOT(8.0,6.0,2)
      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(-1.3,-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 WRITTING THE CAPTIONS OF THE
C   PLOTS. THE SPECIFIC USER MUST SCRUTINIZE THESE LINES
C   AND MAKE NECESSARY CHANGES.
C*******************************************************
C
      CALL SYMBOL(0.0,0.0,0.0,0.14,23HFIG    ISOTHERMS AT X= ,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(F0.GE.25.0)GO TO 22
      GO TO 23
22    CALL SYMBOL(1.0,0.0,0.0,0.14,26HSIMULATIONS FOR FEB. 26 1979,
     C0.0,26)
      GO TO 123
23    CALL  SYMBOL(1.0,0.0,0.0,0.14,26HSIMULATIONS FOR FEB. 27 1979,
     C0.0,26)
123   CONTINUE
      CALL SYMBOL(1.5,8.7,0.1,12HRUN NO: L00 ,0.0,12)
      CALL NUMBER(999.,999.,0.1,T10,0.0,-1)
      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(4.3,8.4,0.07,1HO,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
```

```
      [illegible]
      CALL SYMBOL(1.5,7.[illegible],0.1,33,CURRENT [illegible]SCALE [illegible]     CM/SEC,[illegible]
     C3)
      CALL NUMBER(3.3,7.9,0.1,PS,0.0,-1)
      CALL SYMBOL(1.5,7.7,0.1,31HTOTAL SIMULATED TIME :      HRS,0.0,31)
      CALL NUMBER(3.3,7.7,0.1,P6,0.0,-1)
      CALL SYMBOL(1.5,7.2,0.1,20HLENGTH SCALE(METERS),0.0,20)
      CALL AXIS(4.1,7.2,1H ,-1,1.0,0.0,0.,617.)
      CALL SYMBOL(1.5,6.7,0.1,22HVELOCITY SCALE(CM/SEC),0.0,22)
      CALL AXIS(4.1,6.7,1H ,-1,1.0,0.0,0.,12.)
      CALL SYMBOL(0.2,6.7,0.21,1H N,45.0,2)
      CALL PLOT(8.0,-15.0,-3)
   10 CONTINUE
      END
 @PRT,S [illegible]
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
.PLUV ... CREATED ON ... 13.31.3.
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     CT(3) = 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     PLOTS U AND V ON CONSTANT DEPTH SECTIONS (FEBRUARY 1979 MISSION)
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),C(IN,JN,KN),
     CWH(IWN,JWN,KN),W(IN,JN,KN),R(IN,JN,KN),WRH(IWN,JWN,KN),
     CH(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),WRH(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=0.34
      ARMAX=0.15
      DO 11 INE=1,NTIME
      CALL READ2(U,V,WH,PINTH,I,J,K,IW,JW,IN,JN,KN,
     CIWN,JWN,D,E,HX,HY,H,MAR,ARM,AI,AH,AV,AP,DI,
     CDY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAM,TAV,AKT,
     CCP,C,A,B,C,EUL,T,TW,RO,ROW,TE,AREF,TREF,TC,
     CTA,3,IIOT)
 6    CONTINUE
      DO 99 I=1,IN
      DO 99 J=1,JN
      H(I,J)=1.0
 99   CONTINUE
      CALL PLOTS(IBUF,1000,11)
      CALL PLOT(0.0,2.0,-3)
 1    FORMAT ()
      DO 1. K=1,KN
      P=K
      P6=CTTOT*TTOT
      CALL FACTOR(0.25)
      IF(K.GT.1) GO TO 23
      DO 33 I=1,IN
      DO 33 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
      YM=0.2*SQRT((AAI-AI)**2+(AAJ-AJ)**2)
      YM=AMAX1(ARMIN/0.25,AMIN1(YM,ARMAX/0.25))
      CALL ARROW(AI,AJ,AAI,AAJ,YM,0.0,12)
 35   CONTINUE
 33   CONTINUE
      GO TO 130
 23   CONTINUE
      DEPTH=(1.0/KNM1)*(K-1)
      DO 40 I=1,IN
      DO 40 J=1,JN
      IF(H(I,J).GT.DEPTH) GO TO 45
      GO TO 40
 45   CONTINUE
      DZ=H(I,J)/KNM1
      LC1=(DEPTH/H(I,J))*KNM1
      IF(LC1.EQ.0) GO TO 55
      LC2=LC1+1
      LC3=LC1+2
      DIFF=(DEPTH-LC1*DZ)
```

```
C       SECTION OF SECOND ORDER FIT
        U1=U(I,J,LO1)
        U2=U(I,J,LO2)
        U3=U(I,J,LO3)
        V1=V(I,J,LO1)
        V2=V(I,J,LO2)
        V3=V(I,J,LO3)
        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
55      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   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(16.0,19.0,2)
        CALL PLOT(14.0,19.0,2)
        CALL PLOT(14.0,17.0,2)
        CALL PLOT(11.0,17.0,2)
        CALL PLOT(11.0,15.0,2)
        CALL PLOT(9.0,15.0,2)
        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(5.0,19.0,2)
        CALL PLOT(5.0,20.0,2)
        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(3.0,4.0,2)
      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(1.5,4.7,0.1,12HRUN NO: 100 ,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 ,0,1.0,0.0,0.,610.)
      CALL SYMBOL(1.5,6.7,0.1,22HVELOCITY SCALE(CM/SEC),0.0,22)
      CALL AXIS(4.1,6.7,1H ,0,1.0,0.0,0.,12.)
      CALL SYMBOL(0.2,6.7,0.21,2H 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 .PLUO
```

```
PROGRAM PLOW
C*********************************************************************
C   THIS PROGRAM PLOTS THE U - 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,INN=16,JNN=19,KNM1=4
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
     CWH(INN,JNN,KN),W(IN,JN,KN),WA(IN,JN,KN),WAH(INN,JNN,KN),
     CH(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(INN,JNN)
      DIMENSION TW(INN,JNN,KN),RO(IN,JN,KN),PINTH(INN,JNN),ROW(INN,JNN,
     CKN),T(IN,JN,KN),IBCU(6)
      DIMENSION IBUF(1000)
      USCALE=10.0
      VSCALE=10.0
      WSCALE=20.0
      HBYL=0.33
      NTIME=1
      DO 11 INEW=1,NTIME
      CALL READ2(U,V,WH,PINTH,I,J,K,IW,JW,IN,JN,KN,
     CINN,JNN,D,E,HX,HY,HI,MAR,MRH,II,IH,AV,AP,DX,
     CDY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAM,TAV,AKT,CB,CW,
     CA,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,
     CTIME,TTOT)
    6 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(3.0,16.0,-3)
      DO 20 J=1,JN
      P3=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
      YA=0.5*SQRT((AAJ-AJ)**2+(AAK-AK)**2)
      YA=AMAX1(ARMIN/0.25,AMIN1(YA,ARMAX/0.25))
      CALL ARROW(AJ,AK,AAJ,AAK,YA,0.0,12)
   15 CONTINUE
   20 CONTINUE
C     DRAW 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 30
   33 CONTINUE
      AAJ=(J-1)*1.0
      AAK=-HI(I,J)*KNM1
      CALL PLOT(AAJ,AAK,2)
      JC=J
      AJC=(JC-1)*1.0
   30 CONTINUE
```

```
1         CONTINUE
          CALL PLOT(AJO,0.0,3)
          AAA=-H(II,JJ)*AYYI
          CALL PLOT(AJO,AAA,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,-1)
          CALL SYMBOL(0.0,1.2,0.14,36H        LAKE KEOWEE-(RIGID-LID MODEL),0.
     C0,36)
          IF(P6.GE.25.)GO TO 22
          GO TO 21
22        CALL SYMBOL(1.0,0.9,0.14,28HSIMULATIONS FOR FEB. 26 1979,
     C0.0,28)
          GO TO 121
21        CALL  SYMBOL(1.0,0.9,0.14,28HSIMULATIONS FOR FEB. 27 1979,
     C0.0,28)
121       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,+1)
          CALL SYMBOL(1.5,9.1,0.1,29HDISCHARGE TEMPERATURE:      C,0.0,29)
          CALL SYMBOL(4.3,9.1,0.07,1HO,0.0,1)
          CALL NUMBER(3.4,9.1,0.1,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(13.0,0.0,-3)
          CONTINUE
11        CONTINUE
          C...
```

```
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     P13= 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
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 MX(IN,JN),MY(IN,JN),HT(IN,JN),MAR(IN,JN),MRH(IWN,JWN),
     CW(IN,JN,KN),Q(IN,JN,KN),WH(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
      WSTL=0.33
      NTIME=3
      DO 11 IME=1,NTIME
      CALL READ2(U,V,WH,PINTH,I,J,K,IW,JW,IN,JN,KN,
     CIWN,JWN,D,E,MX,MY,HT,MAR,MRH,AT,AH,AV,AP,DX,
     CDY,DZ,DT,TAUX,TAUY,WWR,WPH,TAI,TAN,TAV,AKT,CB,CW,
     CA,B,C,CUL,T,TW,RO,ROW,TE,PREF,TREF,TO,
     CTIME,TTOT)
    8 CONTINUE
      CALL PLOTS(IBUF,1000,11)
      READ 51,P1,P2,P13,P4,P5,CTTOT,NTIME
   51 FORMAT()
      ARMIN=0.04
      ARMAX=3.15
    1 FORMAT ()
      DO 1  J=1,3
      CALL FACTOR(0.25)
      CALL PLOT(0.0,16.0,-3)
      DO 20 I=1,IN
      P3=J
      P6=CTTOT*TTOT
      IF (MAR(I,J).LT.11) GO TO 20
      AI=(I-1)*1.0
      DO 30 K=1,KNM1
      AA=-(K-1)*HT(I,J)
      AAI=AI+U(I,J,K)*USCALE
      W(I,J,K)=1.0
      AAK=AA-W(I,J,K)*WSCALE*WSTL
      YW=0.5*ASQRT((AAI-AI)**2+(AAK-AA)**2)
      YW=AMAX1(AMIN1/0.25,AMIN1(YW,ARMAX/0.25))
      CALL ARROW(AI,AA,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=-HT(I,J)*KNM1
      CALL PLOT(AAI,AAK,2)
      GO TO 35
   33 CONTINUE
      AAI=(I-1)*1.0
      AAK=-HT(I,J)*KNM1
      CALL PLOT(AAI,AAK,2)
      IC=1
```

```
      [illegible]
      CONTINUE
 35   CONTINUE
      CALL PLOT(AID,0.0,3)
      AAA=-H(I[illegible])
      CALL PLOT(AID,AAA,2)
C
C**********************************************************************
C   THE NEXT 37 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,-3.4,-1)
      CALL PLOT(6.0,C.0,2)
      CALL PLOT(6.0,3.0,2)
      CALL PLOT(G.0,8.0,2)
      CALL PLOT(G.0,0.0,2)
      CALL PLOT(G.0,-2.C,-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(3.5,1.5,0.14,P4,0.0,-1)
      CALL SYMBOL(0.0,1.2,0.14,36H         LAKE KEOWEE-(RIGID-LID MODEL),0.
     C0,36)
      IF(PE.GE.25.)GO TO 22
      GO TO 23
 22   CALL SYMBOL(1.0,0.9,0.14,28HSIMULATIONS FOR FEB. 28 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: LOG ,0.0,12)
      CALL NUMBER(2.6,9.5,0.1,P10,0.0,-1)
      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,-1)
      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,P3,0.0,-1)
      CALL SYMBOL(1.5,9.9,0.1,32HWIND SPEED (MAX)    :    M/SEC,0.0,32
     C)
      CALL NUMBER(3.8,8.9,0.1,P4,0.0,-1)
      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,-1)
      CALL SYMBOL(1.5,[illegible],0.1,19HSCALES (HORIZONTAL),0.0,19)
      CALL SYMBOL(1.5,[illegible],0.1,20HLENGTH SCALE(METERS),0.0,20)
      CALL AXIS([illegible])
      CALL SYMBOL(1.5,7.0,0.1,22HVELOCITY SCALE(CM/SEC),0.0,22)
      CALL AXIS([illegible])
      CALL SYMBOL(1.5,[illegible],0.1,17HSCALES (VERTICAL),0.0,17)
      CALL SYMBOL(1.5,[illegible],0.1,20HLENGTH SCALE(METERS),0.0,20)
      CALL AXIS([illegible])
      CALL SYMBOL(1.5,[illegible],0.1,22HVELOCITY SCALE(CM/SEC),0.0,22)
      CALL AXIS([illegible])
      CALL PLOT(10.0,0.0,-3)
 1[illegible] CONTINUE
 11   CONTINUE
      END
```

```
      SUBROUTINE ECHRON
C
C         THIS IS ENTRY SUBROUTINE FOR NMC CONTOURING PROGRAM
C                (CALCOMP OR MILGO TYPE PLOTTER)
C
C
C      THE COMPLETE PACKAGE CONSISTS OF 3 SUBROUTINES, ECHRON, CONLIN, AND ENC
C      ALL 3 ARE CATALOGUED TOGETHER IN THE UM 360/65 UNDER MODULE NAME ECHRON
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 ECHRON.
C
C            ------CALLING STATEMENT IS AS FOLLOWS------
C
C      CALL ECHRON(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)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
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       IRECCT 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 ECHAON, 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 ECHAON(MM,IN1,IN2,NEX1,NEX2,NEY1,NEY2,HI,WIO,PLTINC,
     1SAMCON,CONINT,RGRID,IN3,IN4,ZLIT,ZBIG,ANORTH,ASOUTH,AEAST,AWEST,
     3NDASHD,NDASHU,XLABEL,SMOOTH,IRECCT)
C
C        SEE ABOVE COMMENTS FOR DESCRIPTION AND USE OF ABOVE ARGUMENTS
C
      COMMON /STRCON/SMHI,SMWI,X,Y,XGRID,YGRID,CUTOF,SQHI,SQWI,THAX,XPP,
     1YPP,CBIG,U,V,NBUX,JODD,NUYX,NUYY,TORTH,SOUTH,EAST,WEST,CLIT,CBIG,
     2LCLX,LCLY,INCROS,JINC,CLOSIT,PVAL,PVOL,NENTER,MINUM,NMXI,NMYI,
     3NMXII,NMYII,MOSINC,VALLIN,MINC,MAXCRO,WHAT,LDASH1,LDASH2,DASHER,
     4DOLABS,OUTS
      COMMON /BENDEQ/MIXLN,JODE,MOOM,XLAST,YLAST
      LOGICAL DASHER,DOLABS,OUTS
      DIMENSION MM(IN1,IN2),RGRID(IN3,IN4),MENIN)
      INTEGER RGRID
      DATA IMAP/0/
C
C     THE SUBROUTINE FOR CONTOURING SCALAR FIELD ON CALCOMP OR MILGO TYPE PLO
C
      IMAP=IMAP+1
      WRITE(6,13)IMAP
   13 FORMAT(1H1,5X,11HCONTOUR MAP,I3,14,29HINCOMING ARGUMENTS FOLLOW)
      WRITE(6,15)IN1,IN2,NEX1,NEX2,NEY1,NEY2,HI,WIO,PLTINC,SAMCON,CONINT
     1,IN3,IN4
   15 FORMAT(1X,95H IN1 IN2 NEX1 NEX2 NEY1 NEY2 HI WIO PLTINC SAMCON CO
     1NINT IN3 IN4,//,2X,6I5,2F13.4,3F12.3,2I10)
      WRITE(6,17)ZLIT,ZBIG,ANORTH,ASOUTH,AEAST,AWEST,NDASHD,NDASHU,XLABE
     1L,SMOOTH,IRECCT
   17 FORMAT(1X,75H ZLIT ZBIG ANORTH ASOUTH AEAST AWEST NDASHD NDASHU X
     1LABEL SMOOTH IRECCT,//,2X,2F12.3,4F12.4,2I6,2F10.3,I6,/)
```

```
      NX=SMOOTH
      NMAP=99
      CONINC=CONINI
      IF(CONINC.NE.0.)GO TO 5
      WRITE(6,2)I
    2 FORMAT(/,2X,3HMAP,I3,14H ZERO INTERVAL)
      GO TO 120
    5 IPECT=IPECCY+1
      LDASH1=NDASHD
      LDASH2=NDASHU
      DASHER=.FALSE.
      IF(LDASH2.GT.0.AND..LDASH1.GT.0)DASHER=.TRUE.
      MINUM=LABEL
      DOLABS=.FALSE.
      IF(MINUM.GT.0.)DOLABS=.TRUE.
      IF(CONINC.LT.0.)CONINC=-CONINC
      PVOL=.COS*CONINC
      MOST=0
      VALLIN=-9898981989
      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 6
      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
    6 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
      MIACN=MINUM
      WDCE=EAST-WEST
      HCGM=YORTH-SOUTH
      XXLAST=99.
      YYLAST=99.
      DINC=PLTINC
      CLIT=DINC/1.99
      CBIG=DINC/2.
      TMAX=9.00(YORTH-SOUTH+EAST-WEST)
      XGRID=WID/FLOAT(NUVX-1)
      YGRID=HI/FLOAT(NUVY-1)
      MINC=XGRID
      IF(YGRID.LT.XGRID)MINC=YGRID
      X=SMOOTH
      IF(X.LT..25.OR.X.GT.7.)SIX=1.0
      MINC=X*MINC
      CUTOF=SQRT(XGRID*XGRID+YGRID*YGRID)+.01
      CLOSIT=.04
C     CLOSIT IS VALUE FOR CLOSED CONTOUR CHECK
      NEX1=NEX1
      NEY1=NEY1
      NMX1=NEX1-1
      NMY1=NEY1-1
      NEX4=NEX1-1
      NEY4=NEY1-1
C
C        NEXT DETERMINE MAX AND MIN VALUES IN SCALAR FIELD
      ZMAX=HH(NEX1,NEY1)
      ZMIN=ZMAX
      DO 13 I=NEX1,NEX2
      DO 13 J=NEY1,NEY2
      IF(HH(I,J).GT.ZMAX)ZMAX=HH(I,J)
      IF(HH(I,J).LT.ZMIN)ZMIN=HH(I,J)
   13 CONTINUE
      IF(ZMAX.GT.BIG)ZMAX=BIG
      IF(ZMIN.LT.ZLIT)ZMIN=ZLIT
C        NEXT DETERMINE BOTTOM STARTING VALUE FOR CONTOUR LOOP
      PVAL=SAMSIN
   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
```

```
    1 XPP=0.
      YPP=0.
      N1=NUMX-1
      N2=NUMY-1
C
C     CONTOUR LOOP STARTS HERE 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
   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 D2=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=NEX+I
      JJ=NEY+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=YGRIDC*(FLOAT(J-1)+(PVAL-HEN(1))/(HEN(2)-HEN(1)))
      X=XGRIDC*FLOAT(I-1)
      GO TO 45
  342 X=XGRIDC*(FLOAT(I-1)+(PVAL-HEN(2))/(HEN(3)-HEN(2)))
      Y=YGRIDC*FLOAT(J)
      GO TO 45
  344 Y=YGRIDC*(FLOAT(J-1)+(PVAL-HEN(4))/(HEN(3)-HEN(4)))
      X=XGRIDC*FLOAT(I)
      GO TO 45
  346 X=XGRIDC*(FLOAT(I-1)+(PVAL-HEN(1))/(HEN(4)-HEN(1)))
      Y=YGRIDC*FLOAT(J-1)
   45 D=(X-XPP)*(X-XPP)+(Y-YPP)*(Y-YPP)
      IF(D.GE.D2)GO TO 100
      D2=D
```

```
      [illegible]
      LCLX=I
      LCLY=J
      XIT=X
      YIT=Y
  1.. CONTINUE
      IF(G2.GE.9999......) [illegible]
      IF(RGRID(LCLX,LCLY).EQ.[illegible])
      X=XIT
      Y=YIT
C
      WRITE(6,121)PVAL,DOLABS,OUTS
  121 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
  125 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
      IRECCT=IRECCT+1
      WRITE(6,115)IMAP,IRECI,IRECCT
  115 FORMAT(/,10X,11HCONTOUR MAP,I3,24H BEGINS WITH PLOT RECORD,I3,14H
     2 AND 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 [illegible]
```

```
      SUBROUTINE CONLIN(...)
      COMMON /STACON/SMAI,S...,X,Y,AGRID,IG,IL,C...,SUMI,S...,IMA,AMP,
     1 ...P,CBIG,J...,...,...,NUVY,NORTH,SOUTH,EAST,WEST,CLIT,CBIG,
     2 LCLX,LCLY,INCROS,SINC,CLOSIT,PVAL,PVOL,NEYTER,MINUM,MMAI,SMYI,
     3 ...,SMYI,MOSINC,VALLIM,MINC,MAICRD,WHAT,LDASH1,LDASH2,DASHER,
     4 ...,OUTS
      DIMENSION MHI(NI,1,2),CIO(N,2),XXPLOT(275,2),MAXI(N),LEXE(N),
     1 CORD(N,2),HIPPS(NCD)
      INTEGER AGRID(NX,NY)
      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 MINUM
      DASHIX=DASHER
      LABLIT=0
      IF(DOLABS)LABLIT=2
      INCS=.FALSE.
      YMAX=-9.
      XMAX=-9.
      NENST=NENTER
      ISPLOT=2
      MHARD=LDASH1
      MSOFT=LDASH2
      NUGG=2
      X=X
      Y=Y
      XBIG=XX
      YBIG=YY
      LXX=LCLX
      LYY=LCLY
      IPEN=2
      XO=0.
      YO=0.
      TOT=0
      HYPTOT=0.
      NCORD=0
      CLOS=.FALSE.
      GO TO 400
C
C     END SETUP. BEGIN LOOP THAT PICKS EACH STRAIGHT LINE SEGMENTED TRAVERSE
C
  250 IF(NCORD.LT.400)GO TO 252
      WRITE(6,251)NCORD,PVAL
  251 FORMAT(/,2X,I5,14H SQUARES ...,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)=XXSC
      CORD(NCORD,2)=YYSC
      XO=XC+XXSC
      YO=YC+YYSC
      IF(NEXET.EQ.1)LZX=LZX-1
      IF(NEXET.EQ.2)LZX=LZX+1
      IF(NEXET.EQ.3)LZY=LZY+1
      IF(NEXET.EQ.4)LZY=LZY-1
      IF(LZX.LT.1.OR.LZX.GE.NX)GO TO 730
      IF(LZY.LT.1.OR.LZY.GE.NY)GO TO 730
      IF(LZX.EQ.LCLX.AND.LZY.EQ.LCLY.AND.SQRT(XO*XO+YO*YO).LE.CLOSIT.AND
     1 .OUTS.AND.NCORD.GT.3)GO TO 721
      GO TO 723
  721 CLOS=.TRUE.
      IF(LABLIT.EQ.0)LABLIT=1
      GO TO 730
      ...
      IF(NEXET.GT.2)NENTER=NEXET-2
      ...
      ...
  400 ...
      X...=XO+AGRID*FLOAT(LZX-1)
      IF(X...LT...)...
      IF(X...GT...)X...=...
      Y...=YO+AGRID*FLOAT(LZY-1)
```

```
      IF(IOUT.EQ.1)WRITE(3,1) [illegible]
      I=LZX+NMX11
      J=LZY+NMY11
C
C     START EXIT POINT LOOP
C
      MAX(1)=HH(I,J)
      MAX(2)=HH(I,J+1)
      MAX(3)=HH(I+1,J+1)
      MAX(4)=HH(I+1,J)
      DO 401 II=1,4
      IF(ABS(MAX(II)-PVAL).GE.[illegible]) GO TO 401
      IF(MAX(II).GE.PVAL)MAX(II)=PVAL+FVGL
      IF(MAX(II).LT.PVAL)MAX(II)=PVAL-FVGL
  401 CONTINUE
      NEXIT=0
      DO 435 III=1,4
      CIDE(III,1)=-1.
      CIDE(III,2)=-1.
      II=III
      I2=III+1
      IF(III.EQ.4)I2=1
C
C     STATEMENT BELOW SELECTS SIDES THAT HAVE EXIT POINTS
C
      IF(MAX(I1).LT.PVAL.AND.MAX(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 DY2=((PVAL-MAX(1))/(MAX(2)-MAX(1)))*YGRID
      DX2=0.
      GO TO 430
  424 DX2=((PVAL-MAX(2))/(MAX(3)-MAX(2)))*XGRID
      DY2=YGRID
      GO TO 430
  426 DY2=((PVAL-MAX(4))/(MAX(3)-MAX(4)))*YGRID
      DX2=XGRID
      GO TO 430
  428 DX2=((PVAL-MAX(1))/(MAX(4)-MAX(1)))*XGRID
      DY2=0.
  430 CIDE(III,1)=DX2
      CIDE(III,2)=DY2
  435 CONTINUE
C     UNLESS WE HAVE NULL POINT SQUARE, NUMOUT SHOULD BE 1 WITH OUT AT DX2, DY2
      IF(NUMOUT.NE.1)GO TO 432
      NEXIT=NN1
      GO TO 445
  432 IF(NUMOUT.EQ.2)GO TO 438
      WRITE(6,433)LZX,LZY,NUMOUT,PVAL,[illegible]
  433 FORMAT(/,[illegible],12HNO EXIT OUT,[illegible])
      GO TO 500
C
C     SECTION THAT DETERMINES PROPER PATH THRU GRID SQUARE
C     CONTAINING HYPERBOLIC CONFIGURATION. (2 ENTRY AND 2 EXIT SIDES)
C
  438 IF(IRGRID(LZX,LZY).GT.1)GO TO 442
      XIO=CIDE(NN1,1)-DX1
      YIO=CIDE(NN1,2)-DY1
      D1=SQRT(XIO*XIO+YIO*YIO)
      XIO=CIDE(NN2,1)-DX1
      YIO=CIDE(NN2,2)-DY1
      D2=SQRT(XIO*XIO+YIO*YIO)
      IF(D2.LT.D1)GO TO 440
  439 DX2=CIDE(NN1,1)
      DY2=CIDE(NN1,2)
      NEXIT=NN1
      GO TO 445
  440 DX2=CIDE(NN2,1)
      DY2=CIDE(NN2,2)
      NEXIT=NN2
  441 [illegible]
      GO TO 445
  442 [illegible](LZX,LZY)/10
      [illegible]GRID(LZX,LZY)-I3*I1
      [illegible]
      IF([illegible].AND.I2.GT.2.AND.I1.NE.I2.AND.NENTER.GT.[illegible])GO TO 41[illegible]
```

```
[illegible]
416 FORMAT(1/,2X,12HBAD COL EXIT,2X,[illegible])
    GO TO 500
417 LEXE(1)=0
    LEXE(2)=0
    LEXE(3)=0
    LEXE(4)=0
    LEXE(I)=1
    LEXE(I-1)=1
    DO 418 I=1,4
    IF(LEXE(I).EQ.0.AND.[illegible].NENTER.AND.CIDE(I,1).GT.(-.5).AND.CIDE
   1(I,2).GT.(-.5))GO TO 419
418 CONTINUE
    GO TO 415
419 NEXET=I
    OX2=CIDE(I,1)
    OY2=CIDE(I,2)
C                        END HYPERBOLIC GRID SQUARE SECTION
445 XIND=OX2*XGRID*FLOAT(L2X-1)
    YIND=OY2*YGRID*FLOAT(L2Y-1)
    IF(NGRID(L2X,L2Y).EQ.MINGRID(L2X,L2Y))=1
    XXSQ=XIND-XBIG
    YYSQ=YIND-YBIG
    HYPE=SQRT(XXSQ*XXSQ+YYSQ*YYSQ)
    IF(HYPE.GE..0001.AND.HYPE.LE.CUTOF)GO TO 346
    WRITE(6,397)L2X,L2Y,NENTER,NEXET,XBIG,YBIG,XIND,YIND,OX1,OY1,OX2,
   2OY2,XXSQ,YYSQ,HYPE,NENST,LCLX,LCLY,NCORD,XX,YY
397 FORMAT(1/,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 500
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)DASHIT=.FALSE.
    HI=0.
    XEND=XX
    YEND=YY
    DO 743 I=1,NCORD
    IF(HI+HIPPS(I).GE.HANC)GO TO 742
    HI=HI+HIPPS(I)
    XEND=XEND+CORD(I,1)
743 YEND=YEND+CORD(I,2)
    GO TO 745
742 R=(HANC-HI)/HIPPS(I)
    XEND=XEND+R*CORD(I,1)
    YEND=YEND+R*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(1/,2X,14HHYPE TOO SMALL,2X,4F12.6)
    GO TO 500
750 CANG=XXSQ/HYPE
    SANG=YYSQ/HYPE
    HYH=HYPE
    IF(CLOS)GO TO 751
    COSRAC=CANG
    SINRAC=SANG
    GO TO 759
751 XE=XX
    YE=YY
```

```
      DO 755 J=1,NCORD
      J=NCORD-J+1
      IF(H+HIPPS(J).GE.HANC)GO TO 755
      H=H+HIPPS(J)
      X=XB-CORD(J,1)
   52 Y=YB-CORD(J,2)
      GO TO 757
  755 X=(HANC-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 300 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 754 I=1,NCORD
      IF(H+HIPPS(I).GE.ZANC)GO TO 766
      H=H+HIPPS(I)
      XIND=XIND+CORD(I,1)
  754 YIND=YIND+CORD(I,2)
      GO TO 763
  766 X=(ZANC-H)/HIPPS(I)
      XIND=XIND+X*CORD(I,1)
      YIND=YIND+X*CORD(I,2)
  763 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
      EXIT=ATAN2(SOUT,COUT)
  200 HYP=HYPE
      IF(HYP.GT.HINC)GO TO 449
      IF(XEND.LT.WEST.OR.XEND.GT.EAST.OR.YEND.LT.SOUTH.OR.YEND.GT.NORTH)
     1GO TO 790
      IF(INCS)GO TO 446
      INCS=.TRUE.
      CALL PLOT(XBEG-WEST,YBEG-SOUTH,3)
  446 I=IPEN
      IF(GASH(I).AND.I.EQ.2)I=IUPLOT
      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     ... CALL ... IN ... TO FIT ... AT ENDS OF SEGMENT
C     ... INTERPOLATION ... TO FIT BETWEEN 2 END DIRECTIONS.
C     ...STRAIGHT LINE INTERPOLATION I. ...
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
  447 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=-CSIG
      HYPMAX=HYP-CLIT
      H25=.25*HYP
  450 TYP=TYP+SINC
      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     NCAT 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.
      NONC=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(DASHIT.AND.IPER.EQ.2)GO TO 615
      CALL PLOT(XUX,YUY,IPER)
      GO TO 609
  615 CALL PLOT(XUX,YUY,IOPLOT)
```

```
      IF(...) GO TO ...
      ...=NHARD-1
      IF(NHARD.GT.0)GO TO 609
      ICPLOT=1
      NHARD=LDASH1
      GO TO 609
  ... NSOFT=NSOFT-1
      IF(NSOFT.GT.0)GO TO 609
      ICPLOT=1
      NSOFT=LDASH2
  609 IPEN=2
      NUNC=NUNC+1
      IF(.NOT.DOLABS.OR.YUT.LE....)GO TO 610
      Y...=YUT
      X...=XUT
      GO TO 610
  ... IPEN=3
      IF(LABLIT.NE.1)GO TO 610
      CALL ENDER(X-WEST,Y-SOUTH,PVAL,1)
      LABLIT=...
  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(6,602)PVAL,TMAX
  602 FORMAT(/,2X,16HREACHED TMAX ON ,2F12.4)
      GO TO 500
  790 IF(LUP...EQ.NDINC)GO TO 800
      HYP=HYPF2N
      SENT=SOUT
      CENT=COUT
      ENTIR=CXET
      SANG=SINF2N
      CANG=COSFOR
      XBEG=XEND
      YBEG=YEND
      XEND=XIND
      YEND=YIND
  ... CONTINUE
C
C        END MAIN CURVILINEAR INTERPOLATE AND PLOT LOOP
C
  500 IF(.NOT.CLOS.OR..NOT.DOLABS.OR.YMAX.LT...)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(XUT,YUT,PVAL,1)
      IF(NUGG.LE.MOSINC)GO TO 502
      MOSINC=NUGG
      VALLIN=PVAL
  502 RETURN
      END
```

ENDER

```
      SUBROUTINE ENDER(X,Y,I)
      COMMON /SCALE3/HONUM,MODE,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(I.EQ.2)GO TO 14
      D(1)=ABS(Y-HOGH)
      D(2)=ABS(X-MODE)
      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.[illegible]	-.[illegible]	[illegible]	.[illegible]	15.[illegible]	.[illegible]	-1.	1
.	10.000000.	3.610000.	[illegible].745549.	.419780.	-2.7[illegible]	[illegible]	0.	1
.	9.991421.	3.605710.	10.745549.	.419637.	-2.7[illegible]	10.014379.	0.	1
.	9.984[illegible].	3.6[illegible]2026.	10.745549.	.4196[illegible].	-2.78[illegible]	10.012075.	0.	1
.	9.977635.	3.598818.	10.745549.	.4195[illegible].	-2.780000.	10.011939.	0.	1
.	9.971972.	3.595596.	10.745549.	.419457.	-2.780000.	10.010936.	0.	1
.	9.965914.	3.593457.	10.745549.	.419397.	-2.780000.	10.010041.	0.	1
.	9.962347.	3.591175.	10.745549.	.4194[illegible].	-2.780000.	10.009234.	0.	1
.	9.958187.	3.589094.	10.745549.	.41[illegible]294.	-2.750000.	10.008499.	0.	1
.	9.954354.	3.587179.	10.745549.	.41[illegible]249.	-2.7[illegible]	10.007829.	0.	1
.	9.953426.	3.585424.	10.745519.	.419237.	-2.780000.	10.007193.	0.	1
.	9.947492.	3.583746.	10.745519.	.419168.	-2.780000.	10.006606.	0.	1
.	9.949374.	3.582187.	10.745549.	.419131.	-2.780000.	10.006055.	0.	1

TOTAL TIME THUS FAR = 1.0HPS

TAUX= -.339954 TAUY= .146753

TAI= .1000000+001
TAM= .6907031-002
TAV= .2262147+003
CB= .3000000
CW= .3000000
AKT= .1911157-001
TREF= .1000000+003

RREF= .6447740+001
EUL= .1742222+004
A= .1003420+001
B= -.1500000-004
C= -.4600000-005
TC= -.1278000+001
TD= .1500000+002

IIG= 6 EX= .7981277-001

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	-.72	-.67	-.56	-.52	-.46	-.44	-.41	-.39	-.32	-.23	.14	-.37	.00
.00	.00	.00	.00	.00	.00	.00	-.59	-.72	-.64	-.56	-.51	-.46	-.49	-.53	-.49	.00	.00	.00	.00
.00	.00	.00	.00	.00	.22	-.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	-.46	-.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	.23	.00
.00	.00	.00	-.30	-.76	-.60	-.63	-.55	-.44	-.37	-.31	-.32	-.26	-.16	.14	.00	.00	.15	.00	.00
.00	.00	.00	-1.26	-1.15	-.72	-.47	-.37	-.32	-.25	-.21	-.21	-.20	-.12	.00	.00	.00	.15	.00	.00
.00	.00	.00	-2.41	-1.45	-.53	-.25	-.13	-.14	-.15	-.12	-.04	-.10	-.05	.00	.00	.00	.00	.00	.00
.00	-1.13	-2.24	-1.65	-.72	-.04	.15	.13	.02	-.03	.31	.06	.02	-.13	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.29	.22	.14	.10	[illegible]	.06	.20	-.03	.24	.29	.00	.00	.00	.00
.00	[illegible]	.00	.02	.01	.00	[illegible]	.03	.14	.14	.00	.00	.00	.37	.20	.27	.25	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.02	.22	.20	.04	.01	.00	.00	.17	.23	.03	.00	.07	.00
.00	.00	.00	.00	.00	.07	.00	.73	.11	.00	.01	.06	.00	.00	.17	.21	.19	.00	.00	.00
.00	[illegible]	.06	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.14	[illegible]	[illegible]	[illegible]	[illegible]	.00	.00	.07	.00	.00	.00	.00
[illegible]	[illegible]	.02	[illegible]	[illegible]	[illegible]	[illegible]	.00	[illegible]	[illegible]	[illegible]	.06	[illegible]	.06	.00	.00	.05	.00	.00	.00

U-VELOCITY FOR K= [illegible]

.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	-.83	-.74	-.72	-.67	-.64	-.62	-.55	-.25	-.08	-.29	.00
.00	.00	.00	.00	.00	.00	.00	-.43	-.93	-.85	-.76	-.73	-.70	-.71	-.75	-.71	.09	.00	.00	.00
.00	.00	.00	.00	.00	.02	-.25	-.74	-.88	-.94	-.77	-.73	-.71	-.73	-.77	-.67	.03	.03	.00	.00
.00	.00	.00	.00	.00	.00	-.45	-.73	-.81	-.79	-.74	-.70	-.65	-.70	-.72	-.58	.03	.00	.00	.00
.00	.00	.00	.00	.00	.03	-.66	-.83	-.79	-.73	-.66	-.64	-.62	-.60	-.62	-.63	.03	.00	.03	.00
.00	.00	.00	.00	-.62	.03	-.93	-.85	-.75	-.67	-.62	-.56	-.54	-.47	-.36	-.29	.22	.22	.12	.00
.00	.00	.00	-.55	-.98	-.81	-.84	-.75	-.66	-.59	-.55	-.52	-.48	-.38	-.31	.03	.03	-.06	.03	.03
.00	.00	.00	-1.42	-1.34	-.93	-.71	-.58	-.52	-.46	-.45	-.43	-.42	-.33	.00	.00	.03	-.07	.03	.00
.00	.00	.01	-2.44	-1.66	-.74	-.42	-.35	-.36	-.37	-.34	-.31	-.33	-.31	.00	.00	.03	.00	.00	.00
.00	-1.35	-2.42	-2.03	-.93	-.26	-.06	-.22	-.23	-.26	-.21	-.16	-.20	-.32	.00	.00	.03	.00	.00	.00
.00	.00	.00	.00	.00	.03	.05	.03	-.29	-.12	.00	.00	.00	-.24	.32	.07	.03	.03	.03	.03
.00	.00	.00	.00	.00	.03	.04	.02	-.08	-.04	.03	.00	.02	-.12	-.02	.05	.03	.02	.00	.00
.00	.00	.01	.02	.03	.02	.02	.09	-.01	.03	.00	.01	.00	.05	-.05	.00	.03	.02	.00	.00
.00	.00	.00	.03	.05	.02	.02	.03	-.12	.04	.02	.02	.03	.03	-.05	-.01	-.02	.00	.09	.00
.00	.03	.01	.03	.01	.03	.00	.03	-.07	.06	.04	.04	.05	.02	.03	.03	.03	.03	.03	.00
.00	.00	.00	.00	.01	.03	.00	.02	.03	.01	.06	.00	.00	.04	.00	.09	.03	.02	.00	.03

U-VELOCITY FOR K=

.00	.00	.00	.00	.00	.00	.00	-1.39	-1.39	-1.39	-1.39	-1.39	-1.35	-1.36	-1.39	-1.35	-1.39	-1.35	-1.39	.00
.00	.00	.00	.00	.00	.00	.00	-.96	-.95	-.97	-.81	-.76	-.74	-.71	-.69	-.62	-.31	-.13	-.33	.00
.00	.00	.00	.00	.00	.00	.00	-.94	-.94	-.90	-.83	-.76	-.76	-.77	-.81	-.75	.00	.00	.00	.00
.00	.00	.00	.00	.00	-.02	-.29	-.75	-.92	-.86	-.61	-.76	-.77	-.79	-.82	-.79	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	-.52	-.72	-.95	-.84	-.75	-.75	-.74	-.75	-.77	-.62	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	-.71	-.85	-.84	-.79	-.74	-.76	-.66	-.66	-.67	-.64	.00	.00	.00	.00
.00	.00	.00	.00	-.64	.00	-.97	-.91	-.81	-.73	-.65	-.64	-.60	-.54	-.42	-.34	-.04	.16	.07	.00
.00	.00	.00	-.57	-.96	-.83	-.89	-.92	-.72	-.65	-.61	-.58	-.55	-.44	-.13	.00	.00	-.11	.00	.00
.00	.00	.00	-1.77	-1.57	-.97	-.75	-.54	-.59	-.55	-.52	-.50	-.48	-.39	.00	.00	.00	-.11	.00	.00
.00	.00	.00	-1.71	-1.61	-.76	-.46	-.42	-.42	-.44	-.41	-.36	-.39	-.36	.00	.00	.00	.00	.00	.00
.00	-1.50	-2.24	-1.44	-.87	-.40	-.14	-.19	-.28	-.32	-.27	-.22	-.26	-.37	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.01	-.31	-.15	-.13	.00	.00	.00	-.33	-.34	.01	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.04	.00	-.39	-.34	.00	.00	.00	-.16	-.13	-.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 [illegible] 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	.01	.00	.00	.00	.00	-.66	-.76	-.72	-.64	-.61	-.60	-.61	-.65	-.62	.00	.00	.00	.00
.00	.00	.01	.02	.00	-.02	-.23	-.61	-.72	[illegible]	-.64	-.61	-.60	-.62	-.65	-.57	.00	.00	.00	.00
.00	.01	.00	.00	.00	.00	-.35	-.61	-.67	-.66	-.62	-.59	-.56	-.59	-.61	-.49	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	-.58	-.68	-.65	-.62	-.55	-.55	-.53	-.52	-.53	-.47	.00	.00	.00	.00
.00	.00	.04	.00	-.44	.00	-.73	-.72	-.64	-.59	-.54	-.51	-.48	-.43	-.33	-.28	-.03	.13	.05	.00
.00	.00	.00	-.41	-.75	-.65	-.71	-.65	-.57	-.52	[illegible]	-.46	-.44	-.36	-.10	.00	.03	-.11	.00	.00
.00	.03	.00	-.96	-1.01	-.75	-.55	-.51	-.46	-.44	-.41	-.35	-.39	-.31	.00	.00	.00	-.10	.00	.00
.00	.00	.06	-2.17	-1.24	-.59	-.37	-.33	-.35	-.35	[illegible]	-.32	-.31	-.26	.00	.00	.00	.00	.00	.00
.00	-1.32	-1.71	-1.55	-.63	-.21	[illegible]	-.15	-.22	[illegible]	-.21	-.17	-.21	-.13	.00	.06	.03	.00	.00	.00
.00	.00	.04	.00	.06	.03	-.01	-.05	-.12	-.15	.03	.04	.03	-.25	-.34	.01	.03	.00	.00	.00
.00	.00	.05	.03	[illegible]	.03	[illegible]	.00	-.08	-.07	.06	.01	.00	-.15	-.29	-.02	.00	.00	.00	.00
.00	[illegible]	[illegible]	.07	.02	.03	[illegible]	.00	[illegible]	.03	[illegible]	.02	.02	.02	-.12	-.06	.03	.00	.00	.00
.07	.07	.00	.12	[illegible]	.02	[illegible]	.02	-.17	[illegible]	[illegible]	.06	.06	.03	-.79	-.77	-.06	.00	.00	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.00	-.04	[illegible]	[illegible]	[illegible]	.31	[illegible]	.00	[illegible]	[illegible]	[illegible]	[illegible]	.06
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.00	.00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.00	[illegible]	.03	[illegible]	[illegible]	[illegible]

U-VELOCITY FOR [illegible]

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.00	[illegible]	[illegible]	.00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	.00	[illegible]	[illegible]	[illegible]	[illegible]	.00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.00	.00
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.00	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.00
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.00
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

V-VELOCITY FO[illegible]

.00	.00	.00	.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	-.55	-.61	-.58	-.46	.00	.00	.00	.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	-.66	-.64	-.70	-.75	-.75	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	-.52	-.42	-.49	-.45	-.5[illegible]	-.56	-.63	-.73	-.87	-.97	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	[illegible]	[illegible]	-.32	-.39	-.46	-.53	-.60	-.73	-.93	-1.26	-1.45	-1.04	-.63	.00
.00	.00	[illegible]	-.53	-.21	.04	.36	-.35	-.27	-.31	-.42	-.50	-.56	-.60	-.62	[illegible]	[illegible]	-.82	.00	.00
[illegible]	[illegible]	[illegible]	-.15	.33	.43	.26	.04	-.15	-.24	-.34	-.49	-.57	-.61	.02	.00	.00	-.65	.00	.00
.00	.00	.00	1.73	1.36	[illegible]	[illegible]	.07	-.14	-.27	-.35	-.51	-.63	-.67	.00	.00	.00	.00	.00	.00
.00	[illegible]	[illegible]	[illegible]	2.19	.83	.26	-.3[illegible]	-.22	-.34	-.43	-.56	-.75	-.81	.00	.03	.00	.00	.30	.00
.00	.00	.00	.00	.00	.03	-.31	-.33	-.41	-.51	[illegible]	.00	.06	-.83	-.91	[illegible]	.03	.00	.00	.00
.00	[illegible]	[illegible]	[illegible]	[illegible]	.00	[illegible]	.03	-.63	-.64	[illegible]	[illegible]	.00	-.67	-.75	-.79	.03	.00	.03	.00
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.06	[illegible]	[illegible]	-.72	.09	[illegible]	[illegible]	.00	.00	-.67	-.66	.00	.00	.30	.00
.00	[illegible]	[illegible]	.03	[illegible]	[illegible]	[illegible]	.33	-.75	.00	[illegible]	.06	.00	.00	-.52	-.61	-.56	.00	.03	.33
[illegible]	[illegible]	[illegible]	[illegible]	.00	.00	[illegible]	[illegible]	-.74	[illegible]	[illegible]	[illegible]	.00	.00	[illegible]	[illegible]	.03	.00	.00	.00
[illegible]	[illegible]	[illegible]	[illegible]	.00	[illegible]	[illegible]	[illegible]	.03	.00	[illegible]	[illegible]	.00	[illegible]	.00	[illegible]	.00	.00	.0[illegible]	.00

ORIGINAL PAGE IS
OF POOR QUALITY

V-VELOCITY FOR [illegible]

.00	.00	.00	.00	.00	.00	.05	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.05	.05	.05	.03	.05	-.39	-.03	.04	.07	.11	.13	.19	.30	.50	.73	.79	.87	.00
.00	.00	.00	.03	.03	.03	.00	.22	.17	.17	.17	.18	.26	.24	.33	.36	.33	.09	.00	.00
.00	.05	.00	.03	.05	.25	.45	.43	.34	.28	.24	.22	.21	.20	.20	.18	.33	.03	.30	.00
.00	.03	[illegible]	.03	.02	.03	.31	.41	.40	.35	.33	.25	.21	.15	.10	.07	.33	.00	.00	.00
.00	.02	.05	.00	.02	.00	.31	.42	.44	.40	.35	.25	.22	.12	-.32	-.14	.09	.00	.00	.00
.00	.33	.30	.02	.03	.05	.52	.57	.53	.46	.39	.32	.25	.15	-.06	-.43	-.63	-.22	.16	.00
.00	[illegible]	[illegible]	.25	.50	.66	.45	.76	.67	.52	.43	.35	.26	.24	.19	.03	.33	.01	.32	.33
.00	.03	.07	.63	1.13	1.25	1.17	.86	.72	.56	.46	.35	.27	.22	.33	.06	.03	.14	.03	.03
.00	.30	.30	2.27	2.09	1.64	1.22	.92	.71	.57	.47	.33	.21	.16	.36	.00	.03	.00	.09	.06
.00	7.52	7.52	[illegible]	2.75	1.42	1.09	.41	.62	.52	.34	.25	.06	.31	.33	.08	.33	.03	.32	.00
.00	.30	.30	.53	.22	.73	.57	.52	.43	.32	.22	.05	.00	-.31	-.39	.16	.33	.02	.33	.33
.04	.02	.76	.35	[illegible]	.22	.05	.03	.19	.17	[illegible]	.02	.05	.14	.03	.12	.05	.00	.20	.20
.01	.06	[illegible]	.22	.07	[illegible]	[illegible]	.72	.25	[illegible]	[illegible]	[illegible]	.07	.05	.15	.16	.00	.00	.30	.00
.03	.32	[illegible]	.22	.72	.02	.05	.03	.05	.03	[illegible]	.02	.00	.04	.18	.20	.24	.00	.00	.03
.10	[illegible]	.21	[illegible]	[illegible]	.73	[illegible]	.72	.04	.02	[illegible]	[illegible]	.76	.33	.33	.03	.73	.03	.32	.20
.05	[illegible]	[illegible]	[illegible]	.01	.02	.00	.72	.30	[illegible]	[illegible]	[illegible]	.00	.35	.33	.70	.03	.00	.03	.06

V-VELOCITY FOR K=

.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.76	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.12	.23	.31	.36	.36	.40	.45	.57	.76	.95	1.01	1.04	.00
.00	.00	.00	.00	.00	.00	.00	.47	.47	.48	.49	.50	.51	.55	.60	.61	.00	.00	.00	.00
.00	.00	.00	.00	.00	.38	.66	.72	.65	.59	.56	.54	.53	.52	.51	.43	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.54	.70	.70	.66	.51	.57	.53	.48	.41	.33	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.55	.72	.75	.71	.66	.61	.54	.45	.35	.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	.61	1.09	1.17	1.05	.97	.92	.74	.66	.60	.54	.41	.00	.00	.19	.00	.00
.00	.00	.00	.65	1.32	1.51	1.42	1.19	1.00	.97	.77	.67	.58	.47	.00	.00	.00	.29	.00	.00
.00	.00	.00	1.69	2.05	1.94	1.95	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	1.09	.92	.79	.64	.56	.33	.26	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.69	.75	.77	.55	.30	.32	.30	.23	.15	.35	.00	.00	.00	.00
.00	.00	.05	.00	.00	.00	.00	.00	.41	.36	.00	.00	.00	.34	.35	.36	.00	.00	.00	.00
.00	.00	[illegible]	.00	.00	.00	[illegible]	.00	.28	.33	.00	.00	.06	.00	.36	.39	.00	.00	.00	.00
.00	.19	.00	.79	.09	.00	.00	.02	.27	.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	[illegible]	.00	.00	.00	.00	.00	.00	.00	.00	.72	.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	.67	.99	.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	.50	.74	.58	.61	.56	.52	.55	.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	.56	.54	.49	.42	.33	.15	.00	.00	.00	.00
.00	.00	.00	.00	.90	.00	.65	.73	.71	.66	.61	.56	.51	.44	.26	-.06	-.32	.04	.30	.00
.00	.00	.00	.33	.62	.51	.96	.89	.79	.71	.64	.56	.53	.46	.38	.00	.00	.20	.00	.00
.00	.00	.00	.39	1.07	1.24	1.15	.93	.64	.74	.66	.52	.51	.42	.00	.00	.00	.27	.00	.00
.00	.00	.00	.97	1.51	1.48	1.23	1.00	.85	.74	.65	.55	.45	.36	.00	.00	.00	.00	.00	.00
.00	-2.43	-.15	1.11	1.43	1.30	1.04	.91	.77	.67	.56	.44	.30	.24	.00	.00	.00	.00	.00	.00
.00	.00	.00	.50	.00	.00	.58	.04	.60	.48	.02	.00	.00	.22	.15	.32	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.76	.32	.06	.06	.00	.32	.33	.32	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.25	.00	[illegible]	.00	.00	.30	.35	.35	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.22	.00	.00	.00	.00	.00	.34	.37	.35	.00	.00	.00
.00	.01	[illegible]	[illegible]	[illegible]	[illegible]	.00	.00	.20	.00	.00	.00	.00	.00	.00	.00	.07	.00	.00	.00
.00	.00	.00	.00	.00	.00	[illegible]	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00

V-VELOCITY FOR K= [illegible]

.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.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 [illegible] 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	9.93	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
********	********	10.32	9.97	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	********
12.16	11.79	17.55	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	[illegible]	11.11	[illegible]	13.32	9.95	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	9.94	********	********	********
12.16	11.81	13.5[illegible]	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 43

```
*********************************** 9.98  9.96  9.96  9.96  9.94  9.92  9.98  9.98  9.98  9.98  9.95  9.96  9.98  9.98
*********************************** 9.93  9.93  9.92  9.98  9.90  9.96  9.98  9.96  9.95  9.92  9.98  9.96  9.98  9.98
*********************** 9.95  9.96  9.98  9.95  9.93  9.98  9.92  9.98  9.98  9.96  9.93  9.98  9.93  9.96  9.95  9.98
*********************** 9.98  9.92  9.92  9.98  9.98  9.95  9.98  9.92  9.98  9.96  9.92  9.96  9.95*******************
*********************** 9.98  9.98  9.96  9.98  9.98  9.98  9.98  9.96  9.98  9.98  9.92  9.98  9.99*******************
***************************** 9.99  9.96  9.95  9.94  9.98  9.93  9.96  9.92  9.98  9.93  9.96  9.98  9.96  9.98  9.96
*********** 9.30  9.53  9.95  9.95  9.92  9.93  9.93  9.99  9.96  9.96  9.92  9.96  9.99  9.98  9.93  9.98  9.96  9.98
*********** 9.95  9.99  9.96  9.98  9.96  9.95  9.99  9.98  9.93  9.96  9.98  9.95  9.98  9.98  9.95  9.96  9.99  9.96
*********** 10.02 10.22 9.99  9.92  9.96  9.92  9.95  9.93  9.95  9.96  9.98  9.96  9.92  9.98  9.93  9.92  9.95******
13.51 11.99 10.57 10.19 10.31 9.99  9.95  9.95  9.99  9.98  9.93  9.96  9.92  9.98  9.98******  9.95  9.96  9.95******
10.15 12.85 11.24 10.42 10.27 9.99  9.96  9.92  9.95  9.95  9.94  9.96  9.90  9.95  9.35  9.96  9.93*******************
13.51 11.97 10.77 10.21 10.33 9.99  9.96  9.93  9.99  9.96  9.74  9.92  9.92  9.98  9.96  9.98  9.98*******************
***************************** 9.92  9.96  9.93  9.92  9.92  9.93*******  9.92  9.98  9.99  9.96  9.93*******************
*********************************** 9.96  9.96  9.96  9.93*******  9.96  9.90  9.92  9.90  9.92  9.96*************
*********************************** 9.99  9.90  9.98*****************  9.96  9.93  9.92  9.93  9.92*************
*********************************** 9.93  9.90  9.90*****************  9.95  9.95  9.98  9.95  9.96*************
*********************************** 9.98  9.90  9.94*************************************************************
```

TEMPERATURES FOR AT :

```
********************************** 13.00 13.33 10.03 13.00 10.03 10.00 10.03 10.03 13.03 13.03 13.33 13.00 10.00 13.00
********************************** 13.02 12.33 12.03 13.01 10.30 10.00 10.00 10.02 13.30 13.03 10.33 10.00 10.00 12.03
********************** 13.03 12.03 13.05 13.33 10.03 10.06 10.03 16.00 10.06 16.05 13.30 10.00 10.33 10.00 13.00 10.00
********************** 13.30 12.03 13.03 13.32 15.03 13.03 10.23 10.00 10.02 10.03 13.33 13.03 10.33*****************
********************** 15.00 10.03 13.01 13.02 10.00 13.03 17.02 10.06 13.05 10.00 13.00 10.03 10.00*****************
**************************** 13.03 13.06 10.03 10.32 13.03 10.02 10.06 10.00 10.03 10.30 10.03 10.03 10.03 12.03 10.00
*********** 12.30 10.00 13.00 10.30 13.00 10.03 12.03 13.03 10.03 10.03 13.00 10.03 13.00 10.03 10.33 13.00 10.00 13.00
*********** 12.33 13.03 13.32 13.03 13.00 13.33 15.03 13.33 12.32 10.30 13.30 13.03 12.33 13.00 10.03 13.03 13.30 10.03
*********** 17.25 13.03 13.32 10.30 13.02 10.02 10.05 13.05 13.32 13.00 13.02 10.03 13.00 10.03 10.03 13.00 10.33******
11.64 11.53 12.52 12.12 13.02 10.02 13.07 13.03 10.03 13.00 10.03 10.00 10.00 10.00 10.00****** 10.33 10.03 10.00******
10.40 13.37 11.06 12.29 13.35 10.03 13.00 10.33 10.03 13.03 15.03 10.03 13.00 10.00 10.00 10.03 10.03*****************
12.67 11.75 12.69 12.14 13.33 13.03 13.06 13.33 10.33 13.33 13.36 13.03 13.03 13.03 13.30 13.32 10.33*****************
**************************** 13.30 13.35 13.32 16.23 13.04 13.0******* 15.00 10.03 13.03 13.00 10.03*****************
****************************************** 13.35 10.00 10.30 12.03******* 10.00 10.00 10.32 10.03 13.33 13.00************
****************************************** 13.33 10.03 10.03****************** 10.03 10.03 10.00 10.03 10.00************
****************************************** 13.33 13.33 13.33****************** 13.33 13.32 13.33 13.33 13.03************
****************************************** 13.32 13.03 15.33**********************************************************
```

TEMPERATURES FOR [illegible]

```
****************************************** 11.03 10.33 10.33 12.33 10.33 13.03 10.30 13.03 13.30 13.03 10.33 13.03 10.33 10.00
****************************************** 11.03 10.33 10.03 13.00 13.02 10.30 10.00 10.00 13.03 10.00 10.53 10.00 10.30 10.00
****************************** 13.35 13.23 13.05 10.32 10.03 13.30 10.03 10.00 10.00 13.03 10.30 10.00 10.03 13.00 10.00 10.00
****************************** 13.30 13.30 11.33 13.32 10.03 13.03 13.03 10.00 10.00 10.00 13.30 10.00 10.33******************
****************************** 13.33 13.03 13.02 10.02 13.03 10.30 10.34 10.00 10.00 10.33 13.33 10.30 13.03******************
************************************ 10.50 11.00 10.02 10.03 13.06 13.30 13.00 10.03 10.03 13.03 10.02 10.33 13.03 10.00 10.03
****************** 10.35 13.03 13.30 13.05 13.05 10.05 10.03 13.33 10.03 10.00 10.03 10.03 13.03 10.00 10.33 13.00 10.33 10.00
****************** 17.32 13.03 13.33 13.03 11.32 16.35 10.37 17.32 13.33 10.02 10.30 13.33 13.33 13.30 10.73 13.05 10.33 10.03
****************** 10.34 17.71 12.02 13.72 11.00 13.35 10.33 13.33 17.32 10.02 10.32 10.03 13.30 10.33 10.33 13.02 10.33******
11.49 13.82 13.72 17.55 13.21 13.07 13.02 17.32 13.33 17.00 17.32 10.05 10.00 10.00 13.00****** 10.03 10.00 10.00******
12.14 11.04 13.33 12.09 13.02 12.50 13.02 17.33 10.03 13.00 13.02 13.72 17.06 13.30 13.00 10.00 10.00******************
11.45 12.53 17.24 12.67 27.31 13.03 13.00 13.05 10.05 10.30 13.33 10.02 13.03 13.32 13.02 13.03 10.33******************
************************************ 17.50 13.33 13.32 10.73 17.23 17.33****** 17.00 13.73 13.22 13.03 13.33******************
************************************************ 10.32 10.35 13.34 10.20****** 17.00 13.70 10.03 13.73 13.03 13.03************
************************************************ 13.02 10.03 13.30****************** 13.35 10.33 10.03 10.33 10.00************
************************************************ 17.33 17.22 17.33****************** 13.35 10.33 10.03 10.33 10.00************
************************************************ 13.32 13.33 13.30************************************************************
```

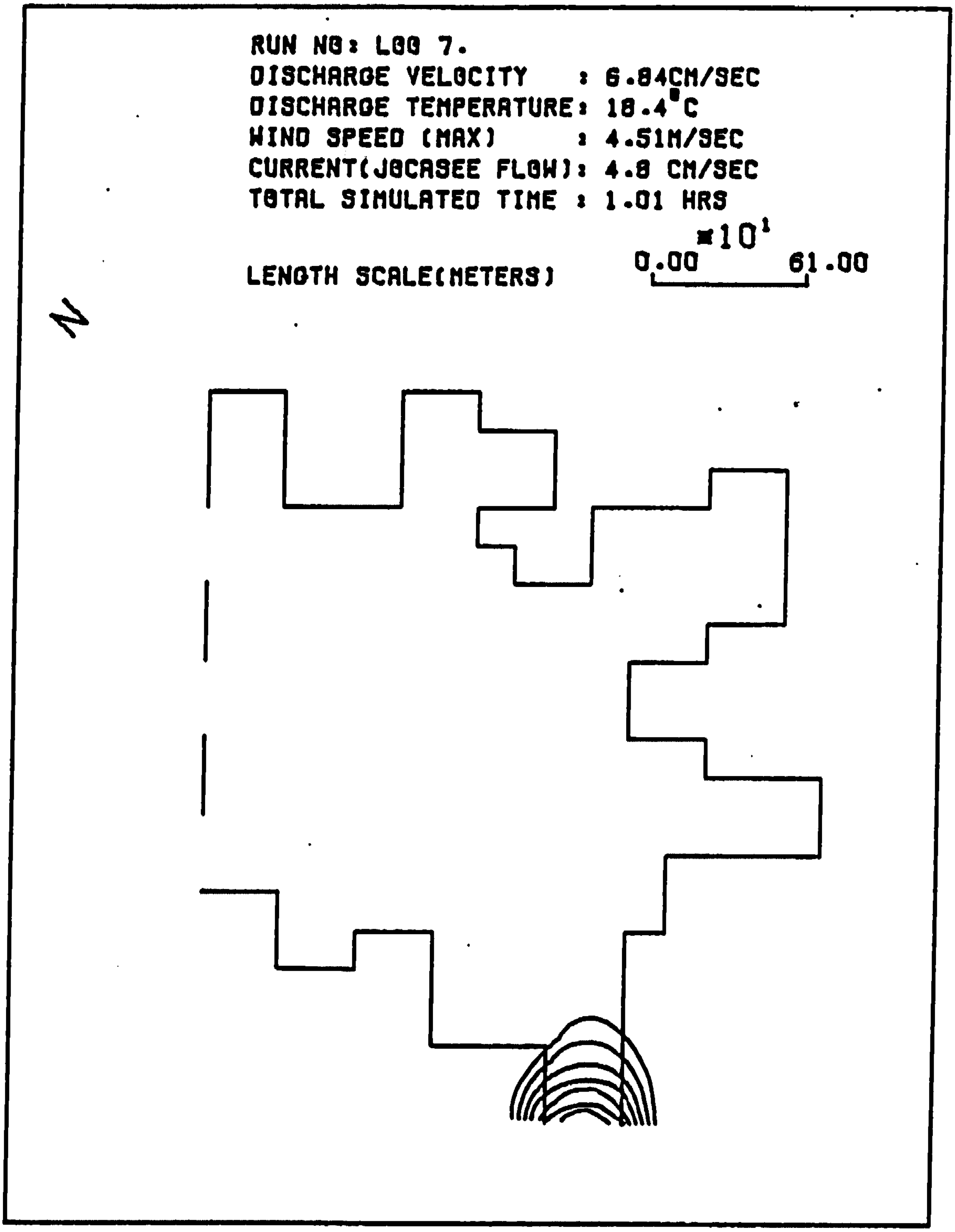
TEMPERATURES FOR [illegible] 5

```
********************************** 11.00 13.33 13.33 13.33 13.33 10.00 13.33 13.33 13.33 13.33 13.33 13.03 13.33 10.33
********************************** 11.00 13.33 10.33 13.30 13.03 10.00 13.30 13.33 13.33 13.33 10.33 13.03 13.33 13.30
********************** 13.00 13.03 13.00 13.33 10.03 13.00 10.00 10.00 10.00 10.03 13.33 10.33 10.03 10.00 10.00 13.33
********************** 13.00 10.00 11.00 10.33 10.03 13.30 13.03 13.00 10.00 10.03 13.30 10.00 10.03*****************
********************** 13.30 13.03 11.00 13.33 13.03 13.33 13.33 10.00 13.33 13.03 13.33 13.03 13.33*****************
**************************** 13.03 11.03 10.03 10.03 13.00 13.33 10.03 10.30 10.03 13.33 13.03 10.33 13.03 10.33 10.33
*********** 13.33 10.03 13.33 13.00 11.03 13.03 10.03 13.00 13.03 10.00 13.03 10.00 13.33 10.03 10.33 13.03 10.30 10.00
*********** 13.30 1.00 10.33 13.00 13.00 10.03 10.03 13.00 13.03 13.00 13.00 13.03 13.00 10.03 10.03 10.03 13.03 10.00
*********** 13.33 13.03 13.00 13.03 13.03 13.33 13.03 13.03 13.33 13.00 13.00 13.03 13.33 13.00 10.03 10.00 13.30******
13.74 13.42 13.11 13.02 13.03 10.00 11.00 13.33 13.03 13.30 13.33 10.00 13.30 10.00 13.33****** 13.33 13.00 10.30******
13.53 13.52 13.14 13.04 13.31 10.03 13.33 13.33 10.03 13.30 13.03 13.03 13.00 13.03 13.03 10.03 10.33*****************
13.74 13.43 13.14 13.03 13.01 13.03 11.03 13.33 13.03 13.00 13.33 10.30 10.30 10.03 13.05 10.33 13.33*****************
**************************** 13.33 11.00 13.33 13.33 13.30 13.3******* 10.30 13.03 13.33 13.33 13.33*****************
************************************** 13.03 13.33 13.30 13.33******* 13.00 13.03 13.33 10.33 13.33 13.03************
************************************** 13.03 10.03 13.03****************** 13.00 13.33 13.03 10.33 13.03************
************************************** 13.03 13.03 13.03****************** 13.33 10.35 13.33 10.33 13.03************
************************************** 13.73 10.07 13.3*******************************************************************
*   1.373300.    -.733033.    -1.473316.    .333330.    35.333336.   1.307245.   -1.   1
*   5.961627.    5.157716.    9.729011.    .4[illegible]1.   -.4703[illegible]   .665710.    1.   [illegible]
*   .929634.     5.196011.    9.72[illegible]1.   .4[illegible]   -1.6703[illegible]   5.677757.   1.   2
*   [illegible]  [illegible]  [illegible]   .4[illegible]   -1.6703[illegible]   .662[illegible]   1.   2
*   [illegible]  [illegible]  [illegible]   .4[illegible]   -1.6703[illegible]   5.662[illegible]   1.   [illegible]
*   [illegible]  [illegible]  [illegible]   .4[illegible]   -1.5703[illegible]   5.6617[illegible]   1.   [illegible]
*   [illegible]  [illegible]  [illegible]   .4[illegible]   -1.6703[illegible]   .6614[illegible]   1.   [illegible]
*   [illegible]  [illegible]  [illegible]   .4[illegible]   -1.6703[illegible]   5.661185.   1.   [illegible]
*   [illegible]  [illegible]  1.7796[illegible]   .4[illegible]   -1.5703[illegible]   5.660[illegible]   1.   [illegible]
*   [illegible]  [illegible]  9.7796[illegible]   .4[illegible]   -1.5703[illegible]   5.6635[illegible]   1.   [illegible]
*   [illegible]  [illegible]  9.7796[illegible]   .4[illegible]   -1.6703[illegible]   5.66[illegible]   1.   [illegible]
```

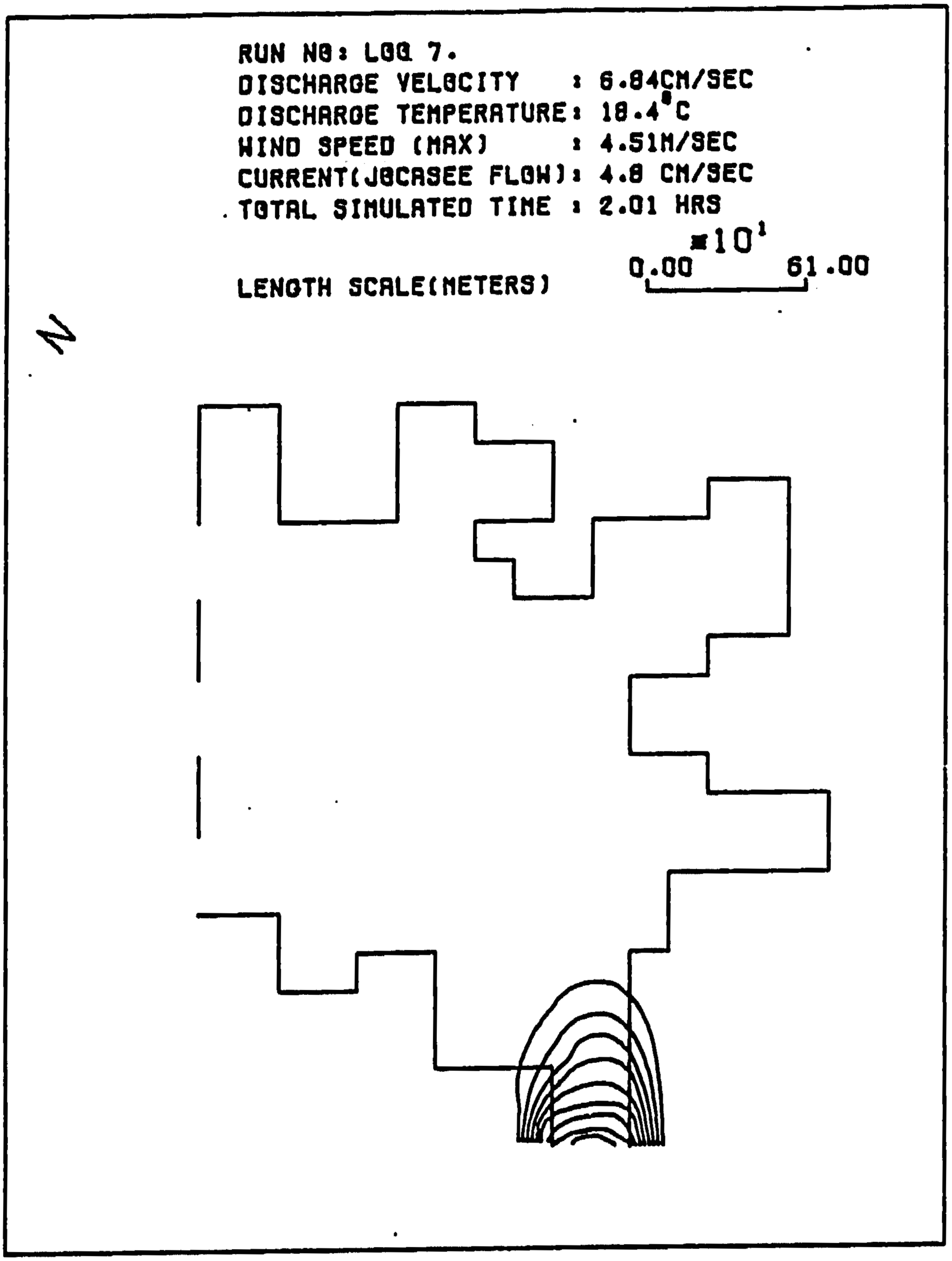
```
B=   -.1203203-03[illegible]
C=   -.1673239-03[illegible]
TE=   -.1167307+001
TC=    .1200706+002

ITN=   4       ER=    .5304957-003
```

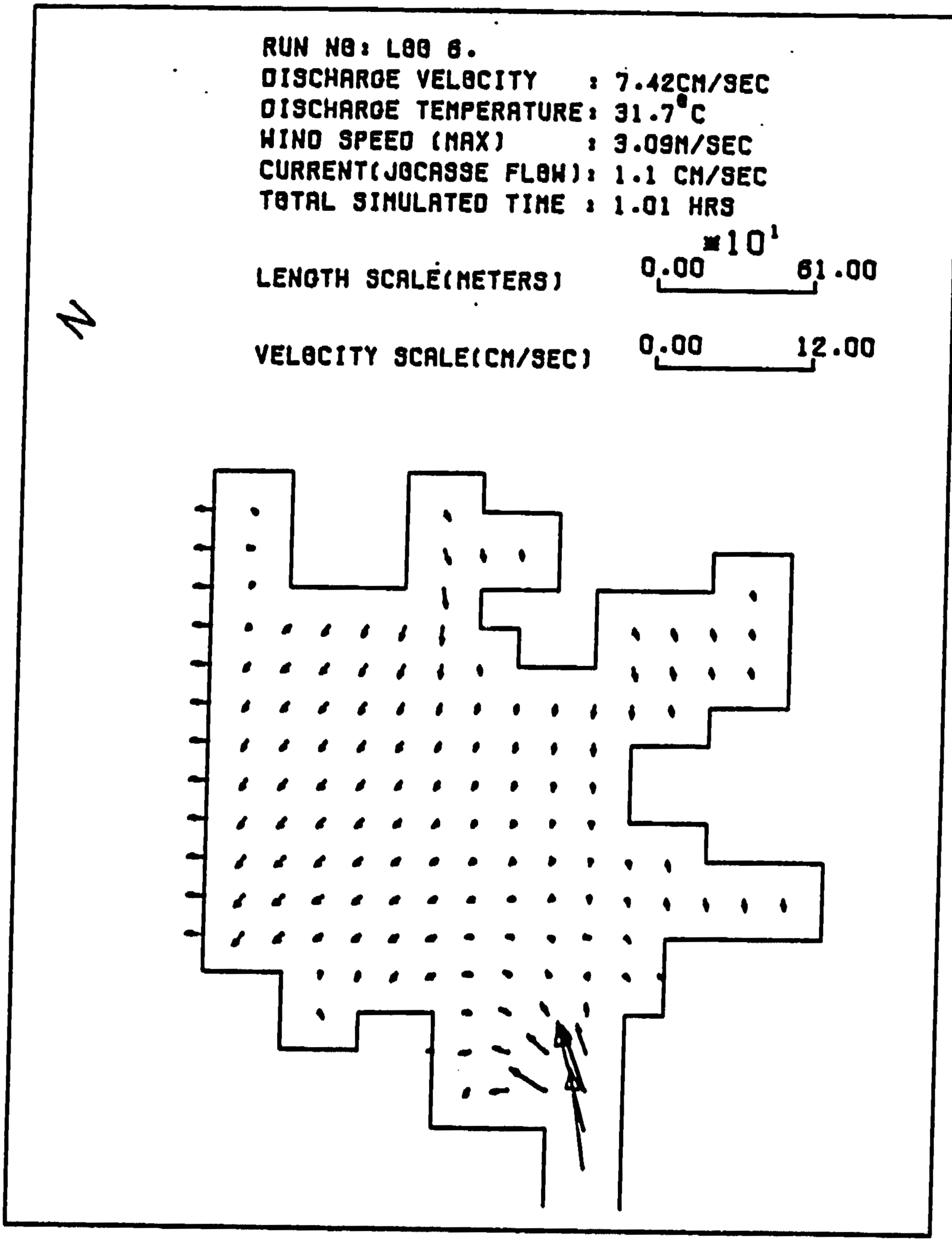
SAMPLE PLOTS



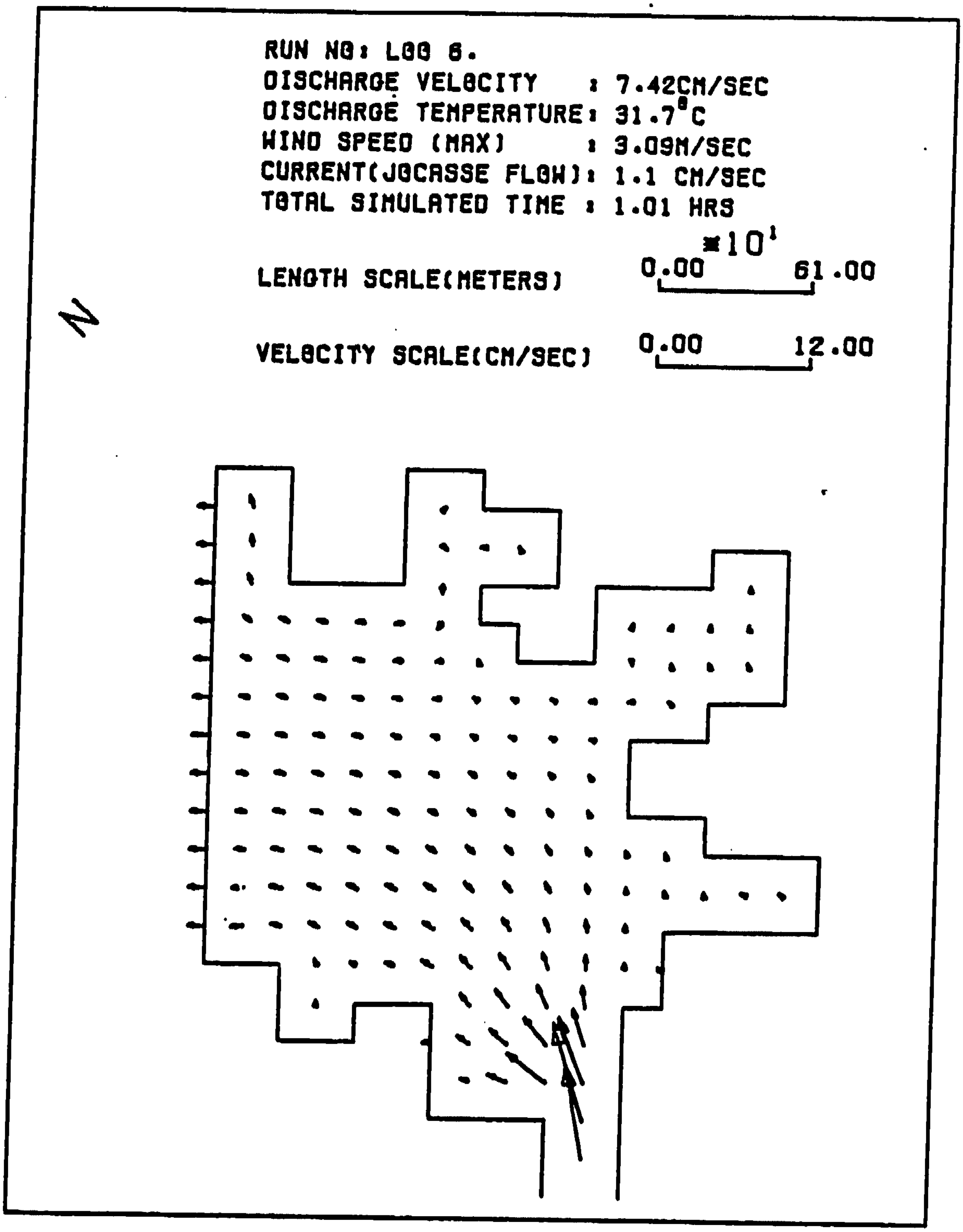
ISOTHERMS AT K= 1.
LAKE KEOWEE-(RIGID-LID MODEL)
SIMULATIONS FOR FEB. 27 1979



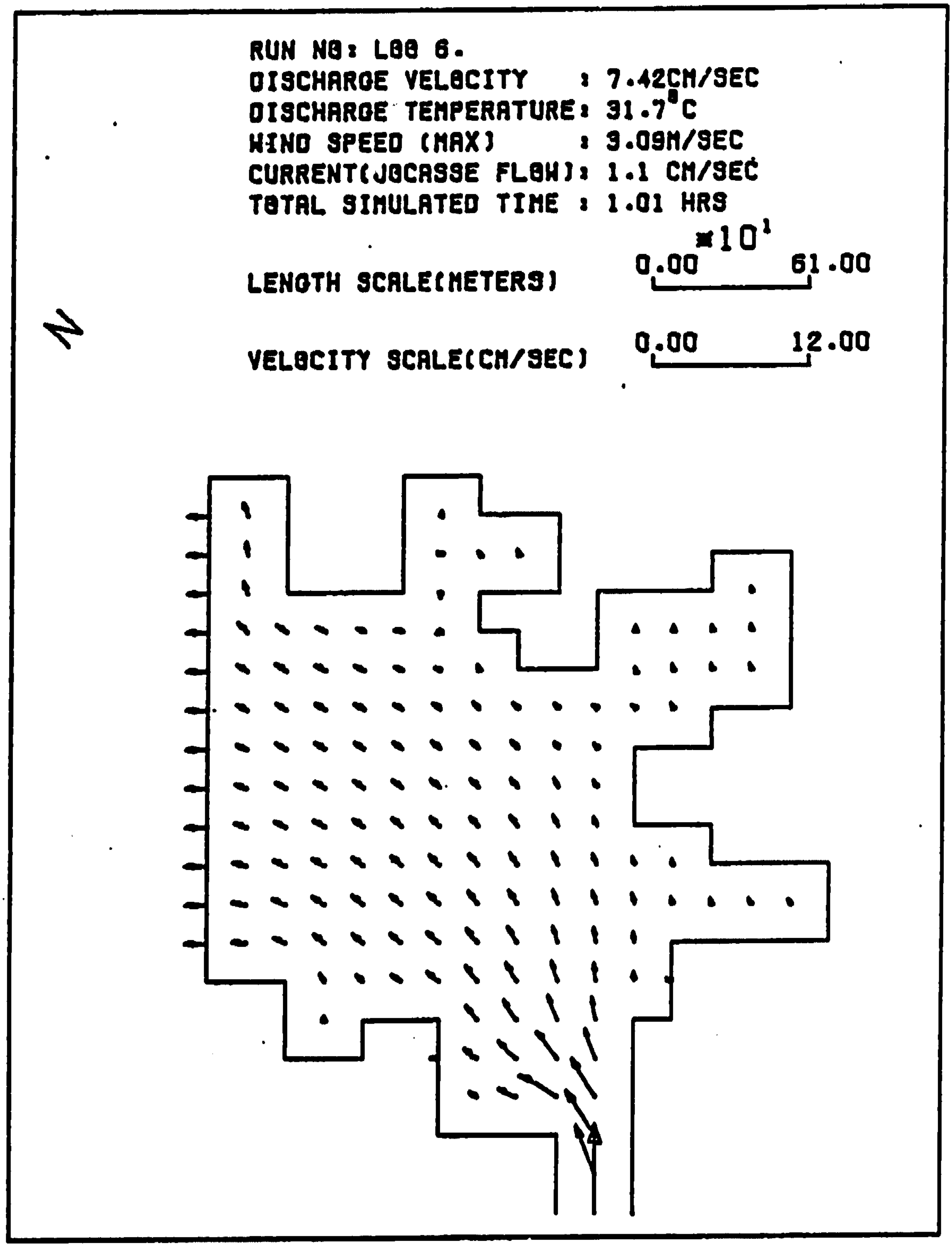
ISOTHERMS AT K= 1.
LAKE KEOWEE-(RIGID-LID MODEL)
SIMULATIONS FOR FEB. 27 1979



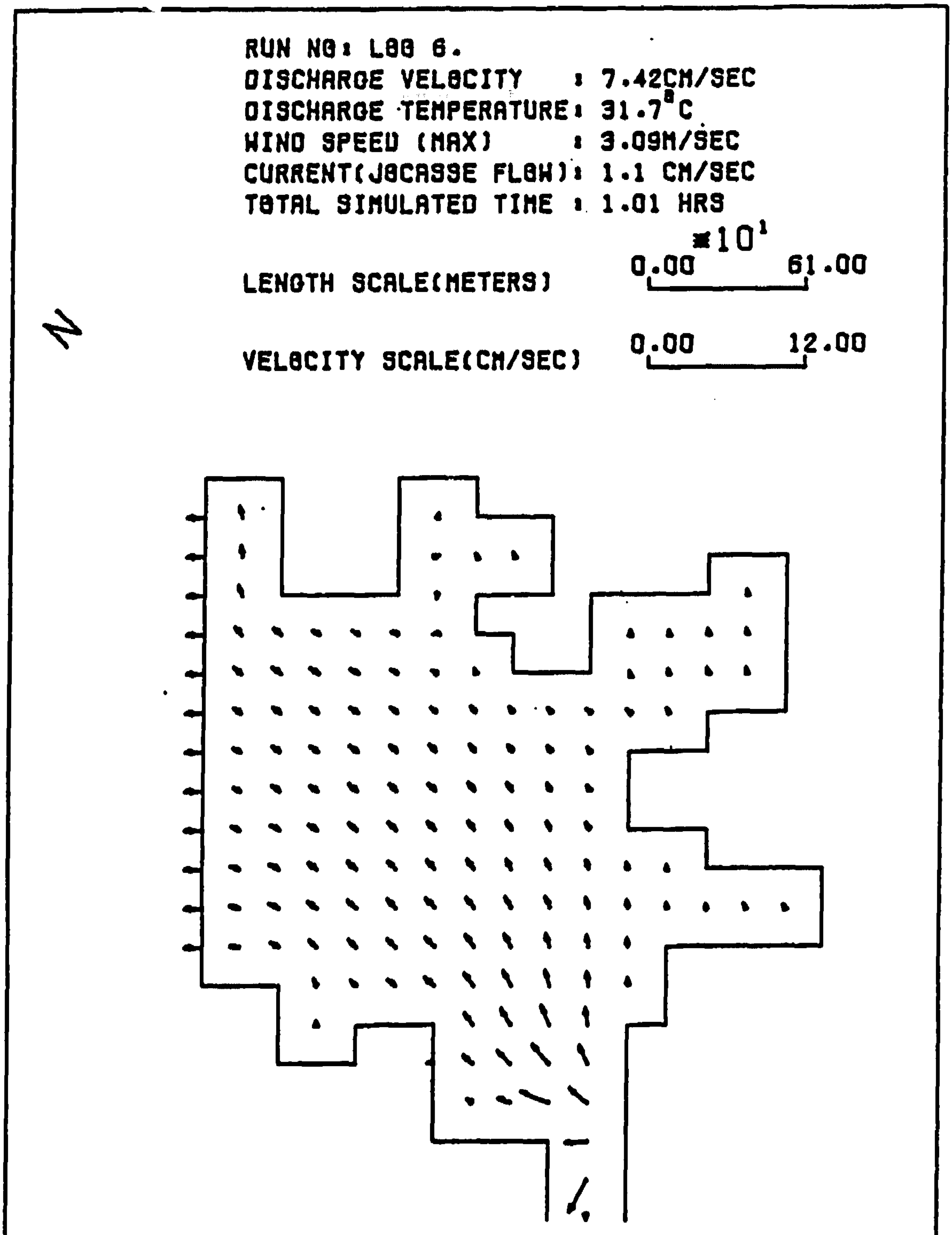
VELOCITIES AT K= 1.
LAKE KEOWEE-(RIGID-LID MODEL)
SIMULATIONS FOR FEB. 27 1979



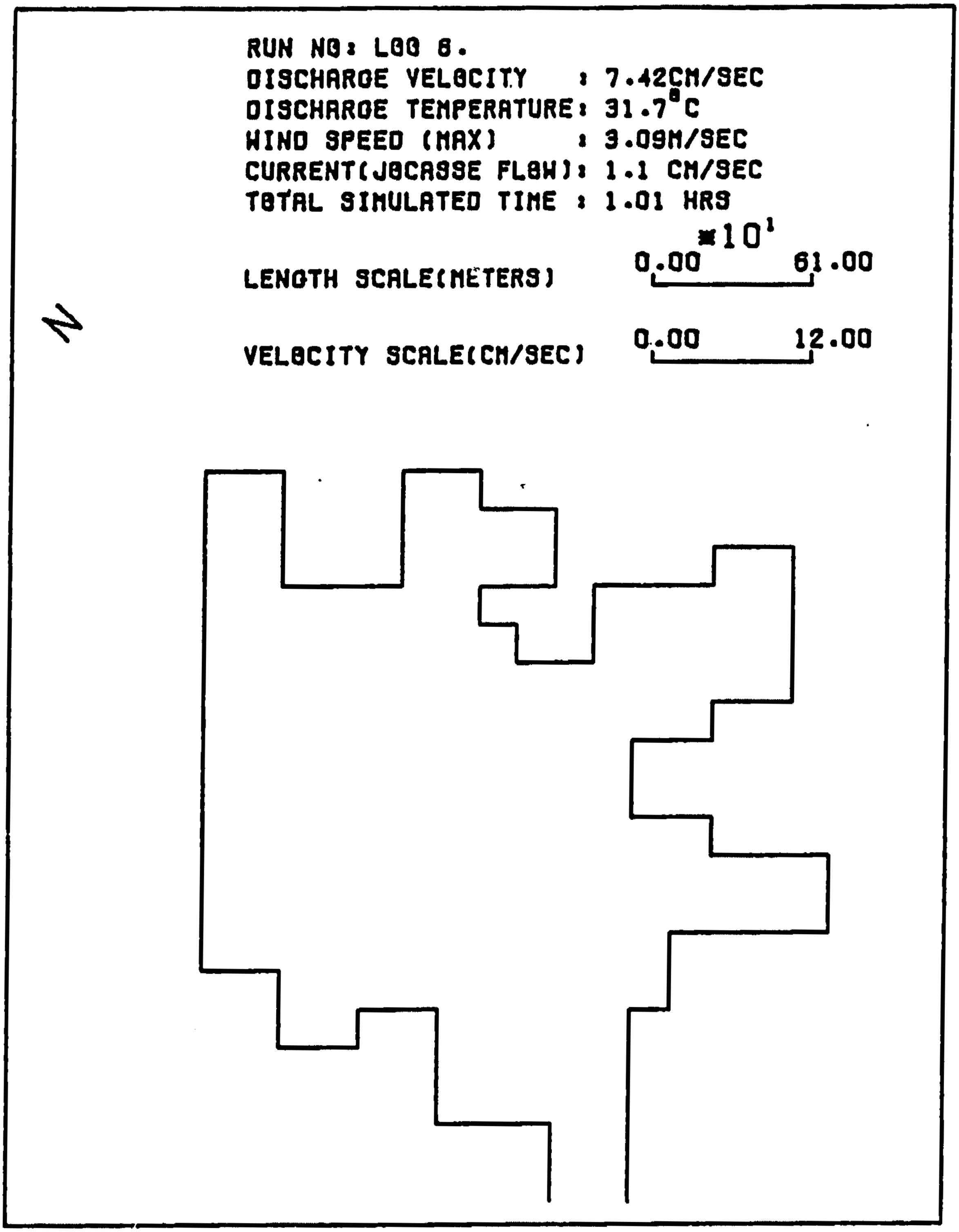
VELOCITIES AT K= 2.
LAKE KEOWEE-(RIGID-LID MODEL)
SIMULATIONS FOR FEB. 27 1979



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



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