U-OF-MIAMI-DME-POLL VOL2 NASA-CR-154624 vol 2

# DEPARTMENT OF MECHANICAL ENGINEERING

THREE-DIMENSIONAL THERMAL
POLLUTION MODELS
VOLUME II - RIGIDLID MODELS

BY

Samuel S. Lee and Subrata Sengupta

ENVIRONMENTAL
COLLECTION MAR 2 1 1986

## School of Engineering and Environmental Design

## UNIVERSITY OF MIAMI

Coral Gables, Florida 33124

ENVIRONMENTAL
RETROSPECTIVE COLLECTION ISSUES

NASA CR-154624

# THREE-DIMENSIONAL THERMAL POLLUTION MODELS VOLUME II - RIGIDLID MODELS

BY

Samuel S. Lee and Subrata Sengupta

Prepared for:
National Aeronautics and Space Administration
(NASA Contract NAS10-8926)

Department of Mechanical Engineering
School of Engineering and Environmental Design
University of Miami
Coral Gables, Florida

May, 1978

## (i) PREFACE

This volume is the second of a three volume set presenting description and program documentation of a mathematical model package for thermal pollution analysis and prediction. Two sets of programs one with the rigid-lid formulation and the other with the free-surface formulation were developed by the Thermal Pollution Group at the University of Miami. These models are three-dimensional and time dependent using the primitive equation approach. They have sufficient generalarity in programming procedure to allow application at sites with diverse topographical features. Both the formulations have near and far field versions. The near field simulating thermal plume areas and the far field version simulating the larger receiving ecosystems. The models simulate the velocity and temperature fields for given meterological and plant intake and discharge conditions.

The first volume summarizes the mathematical formulation, application experience and overall evaluation of the model package. The present volume, namely Volume II presents the rigid-lid programs. Three versions are presented. One for near field simulation. The second for far-field unstratified situations. The third is for stratified basin far-field simulations. Since these versions have many common subroutines, an unified listing is provided with main programs for three possible application conditions mentioned above. The programs are named NASUM I reflecting NASA funding and technical support and University of Miami effort. The third volume presents the

program documentation for the free-surface models.

These volumes are intended as user's manuals and as such presents specific instructions regarding data preparation for program execution and specific sample problems.

## ACKNOWLEDGEMENTS

The work reported in these three volumes has been the result of co-operation between several institutions and individuals. We wish to acknowledge our sincere gratitude to Mr. Phillip Claybourne, Mr. Reed Barnett and Mr. Roy Bland of NASA-KSC for their continued support during the course at this effort. We are also grateful to Mr. Roy Bland for his technical direction of remote sensing data acquisition and processing. The efforts of Mssers. John Pruitt, Jimmy Neff, Harold Reed, John Renou, Al Bradford and Herb Cribb, were invaluable to the remote sensing studies.

The support of Mr. Charles Dewey, Dr. Ben Sill, Bill McCabe, Jack Jenkins, Bob Edmonds, Ralph Roberts, Miles Majors, Ken Morris, Dave Wiseman, Barry Hurdt and John Gaertner was extremely helpful for the Belews Lake study.

The University of Miami Thermal Pollution Team, in addition to the authors, consisted of

Dr. Norman L. Wienberg

Dr. Homer Hiser

Dr. Josyula Venkata

Mr. Harvey Miller

Dr. Sudarshan Mathavan

Dr. C. F. Tsai

Mr. Ruey Lee

Dr. Cecil Carter

Mr. James Byrne

Various members contributed in different areas of expertise during the course of the study. Special gratitude is expressed to Mr. Harvey Miller for contribution to Volume I and III. Dr. Josyula Venkata was a prime contributor to Volume II.

# LIST OF SYMBOLS

The following list of symbols which are obtained from Volume I are presented here for convenience.

| | |
|---|---|
| $A_H$ | horizontal kinematic eddy viscosity |
| $A_V$ | vertical kinematic eddy viscosity |
| $A_Z$ | vertical eddy viscosity |
| $A_{ref}$ | reference kinematic eddy viscosity |
| $A_V^*$ | $A_V/A_{ref}$ |
| $B_H$ | horizontal diffusivity |
| $B_V$ | vertical diffusivity |
| Bref | reference diffusivity |
| $B_V^*$ | $B_V/B_{ref}$ |
| $B_z$ | vertical conductivity $\rho C_p B_V$ |
| $C_p$ | specific heat at constant pressure |
| Eu | Euler number |
| f | Coriolis parameter |
| Fr | Froude Number |
| g | acceleration due to gravity |
| h | depth at any location in the basin |
| H | reference depth |
| I | grid index in x-direction or $\alpha$ direction |
| J | grid index in y-direction or $\beta$ direction |
| K | grid index in z-direction or $\gamma$ direction |
| k | thermal conductivity |

$K_s$ surface heat transfer coefficient

L horizontal length scale

P pressure

$P_s$ surface pressure

Pr turbulent Prandtl number $(\frac{A_{ref}}{B_{ref}})$

Pe Peclet number

Q* heat sources or sinks

Re Reynolds number (turbulent)

Ri Richardson number

T temperature

$T_{air}$ air temperature

$T_{ref}$ reference temperature

$T_E$ equilibrium 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

$\alpha$ horizontal coordinate in stretched system

$\beta$ horizontal coordinate in stretched system

$\gamma$ vertical coordinate in stretched system

| | |
|---|---|
| $\mu$ | absolute vsicosity |
| $\rho$ | density |
| $\Phi$ | dissipation terms in energy equation |
| $\tau_{xz}$ | surface shear stress in x-direction |
| $\tau_{yz}$ | surface shear stress in y-direction |

Superscripts

| | |
|---|---|
| $(\text{——})$ | dimensional quantity |
| $(\sim)$ | dimensional mean quantity |
| $(')$ | diemnsional fluctuating quantity |
| $(\ )$ | dimensional quantity |
| $(\ )_{ref}$ | reference quantity |

# TABLE OF CONTENTS, VOLUME II

i Preface

ii List of Symbols for Equations
(obtained from Volume 1)

1 INTRODUCTION

2 NASUM-I PROGRAM DESCRIPTIONS

2.1 Near-Field
2.1.1 Description - Algorithm
2.1.2 Flow Chart (General)

2.2 Far-Field
2.2.1 Description-Algorithm
2.2.2 Flow Chart

3 LIST OF PROGRAM SYMBOLS WITH EXPLANATION IN ALPHABETICAL ORDER
(Near Field and Far Field Combined)

4 NASUM-I MAIN PROGRAMS

4.1 Near Field

4.2 Far Field
4.2.1 Unstratified Conditions
4.2.2 Stratified Conditions

5 INPUT DATA

5.1 Near Field
5.1.1 Data Required
5.1.2 Format of Input

5.2 Far Field Model
5.2.1 Data Required
5.2.2 Format of Input
5.2.2.1 For Unstratified Conditions
5.2.2.2 For Stratified Conditions

6 SAMPLE CASE FOR NEAR FIELD

6.1 Problem Statement

6.2 Choice of Programs

6.3 Calculation of Input Parameters

6.4 Sample Input

6.5 Program Execution Procedure (For constant depth)

6.6 Program Execution Procedure (For variable depth)

6.7 Sample Output

7 SAMPLE CASE FOR FAR-FIELD
(Unstratified Condtions)

7.1 Problem Statement

7.2 Choice of Programs

7.3 Calculation of Input Parameters

7.4 Sample Input

7.5 Program Execution Procedure

7.6 Sample Output

8 SAMPLE CASE FOR FAR-FIELD
(Stratified Conditions)

8.1 Problem Statement

8.2 Choice of Programs

8.3 Calculation of Input Parameters

8.4 Sample Input

8.5 Program Execution Procedure

8.6 Sample Output

9 DESCRIPTION OF COMPUTER PROGRAMS

9.1 Main Programs for the Near-Field and Far-Field

9.2 Subroutines for the Near-Field and Far-Field

10 REFERENCES

11 APENDICES

(a) Equilibrium Temperature and Surface Heat Transfer Coefficient

(b) Heat Transfer Mechanisms

## 1.0 INTRODUCTION

This volume presents the main programs and subroutines for NASUM-I as described in Volume I. The program symbols are given in alphabetical order for the convenience of the user, and several sample problems are presented to illustrate clearly what input data is required to execute NASUM-I. Note, that the governing equations and associated approximations and assumptions are given in Volume I.

As previously noted, NASUM-I consists of three main programs: Near-Field Rigid-Lid, Far-Field Rigid-Lid, Unstratified, and Far-Field Rigid-Lid, Stratified. The Near-Field is defined as the domain of interest in the immediate neighborhood of the thermal discharge with open boundaries which extend far enough from the discharge to justify using the Far-Field solutions as boundary conditions on the Near-Field. The Far-Field yields the general circulation and temperature distribution of the water body without the discharge.

The subroutines for NASUM-I are presented in alphabetical order with corresponding program descriptions illustrating to the user an outline of the contents of each subprogram. This allows the user to gain sufficient ease in running NASUM-I, and develop the capability of modifying some of the individual subroutines as desired by the user.

NASUM-I consists of three main programs which use many of the same subroutines, but differ mainly in the use of open boundary conditions for the case of the Near-Field Rigid-Lid Model application. The Far-Field Rigid-Lid Model has been

applied for cases of open boundaries by Sengupta et al (1976), but the nature of the open domain for the Near-Field requires using different main programs which in turn select different subroutines to perform the necessary calculations on these open boundaries.

The Far-Field Rigid-Lid Model programs have been separated into two cases, stratified flow and unstratified flow. This separation is necessary since for deep water bodies seasonal thermoclines are sustained, whereas for shallow water bodies turbulent mixing does not sustain stratification. Thus, for NASUM-I to be of a general nature this separation was essential.

Sample problems, showing sample input and sample output, have been included to provide the user with sufficient background and necessary details involved in executing NASUM-I. Note, that any program modifications should be made with great care, and these modifications should be validated by sample runs to assure the user of the effect these modifications have upon the numerical solutions.

## 2.0 NASUM-I PROGRAM DESCRIPTIONS

The Rigid-Lid Model program contains two parts. The first part is for Near-Field program. The Near-Field here means the region where the effects of the plume on the receiving waters are severe. The second part is for Far-Field. The Far-Field here means the total body of receiving water. The main difference between Rigid-Lid Near-Field and Far-Field lies in the boundary conditions. The Main programs for Near-Field and Far-Field call a different set of subroutines. There will be many subroutines which will be the same for both Near and Far Fields. This is the main reason for combining Near and Far Field programs. The Far-Field main programs had to be divided due to the Computer overflow. The computer used was UNIVAC 1106. In places where there are large computers, the Far-Field main programs can be combined.

### 2.1 Near Field

Near-Field, as said above, is the region where the effects of the discharge are felt most. There are four main programs which are explained in the later sections of this volume. There is no special reason for keeping four main programs and they can be combined. The authors found it convenient to work with four programs since modifications were easier to make during application at various sites.

#### 2.11 Description - Algorithm

The problem is set up as an initial value problem. The values of u, v, w, p, $\rho$, t are assumed known initially.

The values at successive time steps are obtained by using the finite difference equations and marching one step at a time. Fig. 2.1 shows the flow chart. The steps may be summarized as follows:

1. Using the values at time n, calculate the forcing function in the pressure equation.
2. Solve the pressure equation iteratively.
3. Calculate u and v from the u and v momentum equations respectively.
4. Calculate w from the continuity equation.
5. Calculate T from the energy equation.
6. Calculate $\rho$ from the equation of state.
7. Check for static stability and apply the infinite mixing criteria for unstable regions.

The values at (n+1) have thus been obtained. Repeat the above procedure for (n+2) using values at (n+1) and so on. The algorithm is essentially the same as that used by Sengupta and Lick (1974).

## 2.12 Flow Chart

Fig. (2.1) shows the general flow chart of the rigid lid programs for both near and far fields. The detailed flow charts with subroutines are given in section 9, where main programs and subroutines are described.

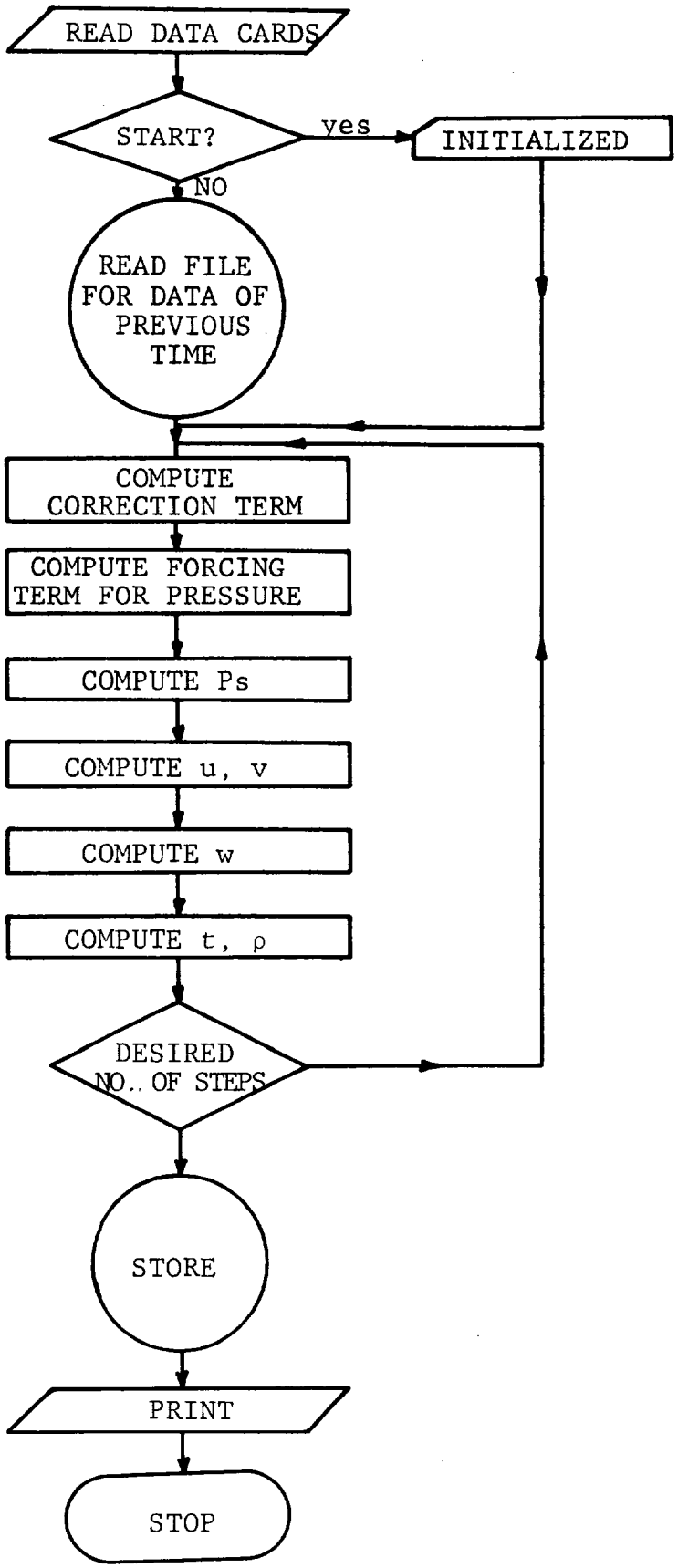



Fig. 2.1 Flow Chart (General)

## 2.2 Far Field

The Far-Field as said before is the total body of water into which the discharge is made. Far-Field programs are divided into two parts. The first part is for the case of an unstratified body of receiving water, and the second part is for a stratified body of receiving water. There are six main programs and these can be used in three different ways which are explained in section (4.2) of this volume.

### 2.2.1 Description-Algorithm

This is the same as for the Near-Field which is explained in section (2.1.1).

### 2.2.2 Flow Chart

The general Flow Chart for the Far-Field is similar to the one for the Near-Field which is in section (2.1.2). A detailed Flow Chart with subroutines is given in section (9.1.5) where the subroutines are described.

## 3.0 LIST OF PROGRAM SYMBOLS WITH EXPLANATION IN ALPHABETICAL ORDER (Near-Field and Far-Field Combined)

A

A : Constant in Equation of State, $\rho = A + BT + CT^2$

AA : Value of V at Plume Inlet (ie at I=9) for Near Field

ABR: $\frac{1}{\text{Rossby Number}}$

AH : $\frac{1}{Re}$

AI : Coefficient infront of inertia term

AKT: $\frac{Ks \quad Href}{B_z}$

AP : Coefficient infront of pressure term

ARBP: Arbitrary pressure

AV : $\frac{1}{\varepsilon^2 \, RE}$ where $\varepsilon = \frac{H}{L}$

A3 : Vertical eddy coefficient, normalized with reference viscosity

B

B : Constant in Equation of State, $\rho = A + BT + CT^2$

BB : Value of V at Plume inlet (ie at I=10) for near field

BZ : $\rho C\rho \quad B_v$

BV : Vertical eddy diffusion coefficient normalized with reference eddy diffusivity

C C : Constant in Equation of State, $\rho = A + BT + CT^2$

CC : Value for $\gamma$ constant

CWX: 0.0

CWY: 0.0

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: $\frac{\partial Ps}{\partial x}$ ; DPSY = $\frac{\partial Ps}{\partial y}$

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: $\frac{\partial (huv)}{\partial y}$ ; D1HUVX = $\frac{\partial (huv)}{\partial x}$ ; 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 = $\frac{\partial (vw)}{\partial z}$ ; D1VZ = $\frac{\partial u}{\partial z}$ ;

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 : Euler Number

EX : Residual error

F

FH : Forcing function

G

G : Dummy variable for v

H

H : Dummy variable for u

HI : Non Dimensional depth = $\frac{h}{\bar{H}}$

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 for x-axis, Main grid

ITN : Index for number of iterations

IW : Index for x-axis, half grid

IRUN: Index for number of runs
= 0 ,firstrun
= 1 ,from second time onwards

J

J : Index for y-axis, Main grid

JW ; Index for y-axis, half grid

JN : Maximum number of main grid points in y-direction

JWN : Maximum number of half grid points in y-direction,
JWN = JN-1

K  K : Index for z-axis

KN : Maximum number of main grid points in z-direction

L  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 to describe general location of a point in the half grid.

MAXIT: Maximum number of iterations

O  OMEGA: Relaxation factor

P  P : Non-dimensional pressure

PN : New pressure, non-dimensional

PINTH: Dummy variable for pressure

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 : Non dimensional density at main grid points

ROW : Non dimensional density at half grid points

RREF: Reference density

RW : Dimensional density at half grid points

T T : Non dimensional temperature at main grid points

TO : Initial temperature (dimensional)

TAMB: Ambient temperature (dimensional)

TAIR: Air temperature (dimensional)

TAI : Coefficient infront of convective terms in the energy equation = 1

TAH : $\frac{1}{Pe}$ where Pe = Re x Pr

TAV : $\frac{1}{Pe}\epsilon^2$ where pe = Re x Pr; $\epsilon = \frac{H}{L}$

TE : Equilibrium temperature (dimensional)

TTOT: Total time elapsed

TAUX: $\partial u/\partial \gamma$ (non-dimensional)

TAUY: $\partial v/\partial \gamma$ (non-dimensional)

TEM : Dimensional temperature at main grid points

TEMW: Dimensional temperature at half grid points

TREF: Reference temperature

TW : Non dimensional temperature at half grid points

TLL : Temperature at the discharge point (non-dimensional)

TMM : Temperature at the discharge point (non-dimensional)

U U : Velocity in x-direction (non-dimensional)

V V : Velocity in y-direction (non-dimensional)

VVIS: Vertical eddy viscosity (non-dimensional)

W W : Velocity in z-direction (non-dimensional)

WH : W at half grid points

WHLDT: Time derivative of WH at lid,

ie $\frac{\partial}{\partial t}$ (WH) at lid

or $\frac{\partial}{\partial t}$ (WH)/z=0

X XINT: Integral of x terms on the right hand side of Poisson's equation. (Eq 2.17, Vol. 1)

X : Horizontal coordinate across discharge for near field.

Y YINT: Integral of y terms on the right hand side of Poisson's equation. (Eq 2.17, Vol. 1)

Y : Horizontal coordinate along discharge for near field.

Z Z : Vertical coordinate

4.0 NASUM-I MAIN PROGRAMS : Near Field Main Programs are given for velocity only and for velocity and temperature

4.1 Near Field

There are four main programs for the near field. They are (1) AMAIN 1 (2) AMAIN 2, (3) TMAIN 1 and TMAIN 2

AMAIN programs are to be used when the velocity field is only required, neglecting the effects of density.

TMAIN programs are to be used when the velocity and temperature are both required.

AMAIN 1 and TMAIN 1 are to be used when the depth is constant.

AMAIN2 and TMAIN 2 are to be used when the depth is variable.

All the above main programs can be combined into a single program. But, the authors found it convenient to have four programs to work with.

4.2 Far Field

Far field main programs are provided for unstratified and stratified cooling reservoirs.

4.2.1 Far Field, unstratified

There are six main programs namely, TMAIN 4, TMAIN 4T, TMAIN 5, TMAIN 5T, TMAIN 5V and TMAIN 6. These programs can be used in three different ways. First, if the velocities and temperatures are to be simulated in a coupled fashion, then TMAIN 4, TMAIN 4T, TMAIN 5 and TMAIN 6 are to be used. Second, if the velocities alone are to be simulated, then TMAIN 4, TMAIN 4T, TMAIN 5V and TMAIN 6 are to be used. Third, if the temperatures alone are to be simulated, then TMAIN 4, TMAIN 4T, TMAIN 5T and TMAIN 6 are to be used.

TMAIN 4 and TMAIN 4T initialize the model. TMAIN 6 does the printing of results. Actual simulation is carried out by TMAIN 5, TMAIN 5V and TMAIN 5T. Thus, for successive simulation, TMAIN 5, TMAIN 5V and TMAIN 5T alone need to be executed. TMAIN 6 may be used if results of each run need to be printed.

4.2.2 Far Field Stratified

There are six main programs, namely TMAIN 4CB, TMAIN 4TB, TMAIN 5B, TMAIN 5TB, TMAIN 5VB and TMAIN 6B. These programs can be used in three different ways. First, if the velocities and temperatures are to be simulated in a coupled fashion, then TMAIN 4CB, TMAIN 4TB, TMAIN 5B and TMAIN 6B are to be used. Second, if the velocities alone are to be simulated, then TMAIN 4CB, TMAIN 4TB, TMAIN 5VB and TMAIN 6B are to be used. Third, if the temperatures alone are to be simulated, then TMAIN 4CB, TMAIN 4TB, TMAIN 5TB and TMAIN 6B are to be used.

TMAIN 4CB and TMAIN 4TB initialize the model. TMAIN 6B does the printing of the results. Actual simulation is carried out by TMAIN5B, TMAIN 5VB and TMAIN 5TB. Thus, for successive simulation, TMAIN 5B, TMAIN 5VB and TMAIN 5TB alone need to be executed. TMAIN 6B may be used if results of each run need to be printed.

5.0 Input Data:

The data that is to be given for the execution of the main programs is called Input Data. The data required is explained in section (5.1.1) for the Near field and in section (5.2.1) for the Far-Field. Sections (5.1.2)and 5.2.2) explain the format of input for the near-field and for the far-field respectively. The actual calculation of data for near field and far field are given in the sample problem section to follow.

5.1 Near Field: For near-field there are two classes of programs. They are AMAIN and TMAIN. AMAIN is used for obtaining velocity distribution only. TMAIN is used for velocity and temperature distributions. The data required for running the programs is described in the following section.

5.1.1 Data Required for Near Field: The data required for running TMAIN programs which will simulate velocity and temperature are described below. (The data needed for running AMAIN programs are not given separately as it can be obtained from the data for TMAIN programs). The Fortran symbols which appear in the main program as data are given in the brackets. For the meaning of algebraic symbols the reader is urged to look into the list of symbols.

| | |
|---|---|
| First time or continuation of run | (IRUN) |
| Number of cycles | (LN) |
| Total number of cycles/(LN) | (LLN) |
| Non-dimensional viscosity (vertical) | (VVIS) |
| 1/Rossby number = $\frac{1}{R_b}$ | (ABR) |

Coefficient infront of inertia term (AI)

1/Reynolds number = $\frac{1}{Re}$ (AH)

$\frac{1}{\varepsilon^2 Re}$ where $\varepsilon = \frac{H}{L}$ (AV)

Coefficient infront of pressure term = 1 (AP)

Convergence criterion (EPS)

Maximum number of iterations in the solution of Poisson Equation for surface pressure (MAXIT)

Relaxation factor (OMEGA)

Arbitrary Pressure for normalizing pressure solution after each iteration (ARBP)

Grid size in $\alpha$ and $\beta$ directions (Non dimensional) (DX,DY)

Grid size in $\gamma$ direction (Non-Dimensional) (DZ)

Coefficient infront of convective terms in the energy equation (TAI)

$\frac{1}{\text{Peclet Number}} = \frac{1}{Pe}$ where Pe = Re x Pr (TAH)

$\frac{1}{Pe\ \varepsilon^2}$ where $= \frac{H}{L}$ (TAV)

Constants in Equation of state (A,B,C)

(The equation of state is $\rho = A+BT+CT^2$ where A,B,C are constants, and T is temperature)

$\frac{Ks\ H}{Bz}$ (AKT)

where Ks is surface heat transfer coefficient

H is reference depth and

Bz is vertical conductivity.

Euler Number (EUL)

Temperature gradient at the vertical boundaries (CW)

Temperature gradient at the bottom (CB)

Value of v at discharge (for near field at I=9, J=1) (AA)

Value of v at discharge (for near field at I=10, J=1) (BB)

Depth of the basin in a constant depth case (CC)

Discharge temperature at Inlet (for near field at I=9, J=1) (TLL)

Discharge temperature at Inlet (for near field at I=10, J=1) (TMM)

Ambient temperature (dimensional) (TAMB)

$\partial u/\partial \gamma$ z=0 in x-direction (TAUX)

$\partial v/\partial \gamma$ z=0 in y-direction (TAUY)

Time step size (DT)

Also, depending on the size of the domain the parameter statement has to be changed. In the parameter statement the following parameters are to be changed.

Maximum number of nodes in $\alpha$-direction, full grid system (IN)

Maximum number of nodes in $\beta$-direction, full grid system (JN)

Maximum number of nodes in $\gamma$-direction, full grid system (KN)

Maximum number of nodes in $\alpha$-direction, half grid system (IWN)

Maximum number of nodes in $\beta$-direction, half grid system (JWN)

## 5.12 Format of Input For Near Field:

In the previous section (5.11) the data required to run the program is described. In this section, the format in which the data has to be given for the programs to execute will be listed. The calculation of the data is explained in the sample problem section (6.3)

| CARD NO. | VARIABLE | FORMAT |
|---|---|---|
| 1 | IRUN | I5 |
| 2 | LN | " |
| 3 | LLN | " |
| 4 | VVIS, ABR | FREE |
| 5 | AI, AH, AV, AP | " |
| 6 | EPS, MAXIT, OMEGA, ARBP | " |
| 7 | DX, DY, DZ | " |
| 8 | TAI, TAH, TAV | " |
| 9 | A, B, C | " |
| 10 | TO | " |
| 11 | AKT, EUL, CW, CB | " |
| 12 | AA, BB, CC | " |
| 13 | TLL, TMM | " |
| 14 | TAMB | " |
| 15 | TAUX, TAUY | " |
| 16 | DT | " |

## 5.2 Far-Field

Far-Field main programs are given for stratified and unstratified conditions of the receiving waters. There are six main programs (TMAIN 4, TMAIN 4T, TMAIN5, TAMIN 5T, TMAIN 5V and TMAIN 6) and these programs can be used in three different ways as described before in section (4.2). The data required for these programs is similar to the Near-Field data but it will be repeated again for convenience. Again Fortran symbols that appear in the main programs are shown in brackets.

For Algebraic symbols, the reader is urged to look into the list of symbols.

5.2.1 Data Required For Far-Field

Number of cycles for coupled velocity and temperature simulation (LN)

Number of cycles for uncoupled temperature simulation (LLN)

Non-dimensional viscosity (VVIS)

1/Rossby Number (ABR)

Coefficient infront of inertia-terms (AI)

$\frac{1}{\text{Reynolds Number}} = \frac{1}{Re}$ = (AH)

$\frac{1}{\varepsilon^2 \; Re}$ = (AV)

Where $\varepsilon = \frac{H}{L}$

Coefficient infront of pressure term (AP)

Convergence criterion (EPS)

Maximum Number of Iterations in the solution of Poisson Equation for Surface Pressure (MAXIT)

Relaxation factor (OMEGA)

Arbitrary Pressure for normalizing pressure solution after each rotation. (ARBP)

Grid sizes in $\alpha$ and $\beta$ directions,

(non-dimensional values) (DX, DY)

Grid size in $\gamma$-direction,

(non-dimensional value) (DZ)

Depth of the basin in a constant-depth case, zero otherwise (CC)

Wind shear in x-direction = $\tau_{xz}$ (TAUX)

Wind shear in y-direction = $\tau yz$ (TAUY)

Time step size, non-dimensional value (DT)

Coefficient infront of convective terms in the energy equation (TAI)

$\frac{1}{Pe}$ (TAH)

where Pe = Re x Pr

$\frac{1}{Pe\ \varepsilon^2}$ (TAV)

where $\varepsilon = \frac{H}{L}$ , Pe = Re x Pr

constants in Equation of state (A,B,C)

Reference temperature (dimensional) (TO)

$\frac{Ks\ H}{B_z}$ (AKT)

Where $K_s$ is surface heat transfer coefficient

H is reference depth

$B_z$ is vertical conductivity

Euler number (EUL)

Temperature gradient at the vertical boundaries (CW)

Temperature gradient at the bottom (CB)

Equilibrium temperature, dimensional (TAMB)

$\sigma''$, constant in the equation which is given below for vertical diffusivity in a thermally stratified lake (CONS)

$$A_v = AV_o \{1 + \sigma'' h\ Tref\ (-\gamma^2 \frac{\partial \tau}{\partial \gamma})\}^{-1}$$

Maximum Value of Vertical diffusivity in a thermally stratified lake, non-dimensional value (AVMX)

Minimum value of vertical diffusivity in a thermally stratified lake, non-dimensional value (AVMN)

Identification matrix in the full grid system (MAR(I,J))

Identification matrix in the half grid system (MRH(I,J))

Non-dimensional depth matrix (HI(I,J))

Number of Inlet-Nodes (NIN)

Number of Outlet Nodes (NOUT)

Location of Inlet and Outlet Nodes (I,J,K)

U-velocity at Inlet and Outlet nodes, non-dimensional (U(I,J,K))

V-velocity at Inlet and Outlet nodes, non-dimensional (V(I,J,K))

Temperature at Inlet nodes, non-dimensional
Also initial temperature if available (T(I,J,K))

Minimum surface temperature in a thermally stratified cooling lake (TSMN)

Maximum depth over which plume heat is accumulated in a thermally stratified cooling lake (DPMX)

A matrix of ambient vertical temperature distribution in a stratified cooling lake (AMINT(I,L))
The first column represents the depths and the second column represents dimensional temperatures.

Number of layers at which ambient temperature is specified. This is the number of rows in AMINT (NTL)

Number of columns in AMINT (N,L) (NTLV)

Depending on the domain size the following have to be changed in the parameter statement of the main programs

Maximum number of nodes in $\alpha$-direction, full grid system (IN)

Maximum number of nodes in $\beta$-direction, full grid system (JN)

Maximum number of nodes in $\gamma$-direction, full grid system, (KN)

Maximum number of nodes in $\alpha$-direction in half grid system (IWN)

Maximum of nodes in $\beta$-direction in half grid system (JWN)

## 5.22 Format of Input

In the previous section the data required is described for all the programs in general. In this section, the format in which the data has to be given for each program will be listed. The actual calculation of data will be explained in the sample problem section (7.3) and (8.3) for unstratified and stratified conditions of receiving waters respectively. Since there is more than one main program for each type of ambient condition , the data is given in the form of elements and each element is given a name. The element names and the main programs that goes with it are explained in the next section for the unstratified case and in section (5.2.2.2) for the stratified case.

5.2.2.1 Format of Input for Unstratified Condtions:

There are four elements for unstratified type conditions and they are (1) INDATA (2) INDATA 5 (3) INDATA 6 (4) ITPK1 and these go with the main programs (1) TMAIN4, (2) TMAIN5, TMAIN5T and TMAIN5V (3) TMAIN 6 and (4) TMAIN 4T respectively. The data that goes with these data elements is explained in the next sections. The main programs that goes with the elements is given in the brackets.

5.2.2.1.1 Data Element "INDATA" (For main program TMAIN 4)

| CARD NO | VARIABLE | FORMAT |
|---|---|---|
| 1 | LN, LLN | 16I5 |
| 2 | VVIS, ABR | Free |
| 3 | AI, AH, AV, AP | " |
| 4 | EPS, MAXIT, OMEGA, ARBP | " |
| 5 | DX, DY, DZ | " |
| 6 | CC | " |
| 7 | DT | " |
| 8 | TAI, TAH, TAV | " |
| 9 | A, B, C | " |
| 10 | TO | " |
| 11 | EUL, DW, EB | " |
| 12 | TAMB, AKT, TAUX, TAUY | " |
| 13 | MAR (1,1) MAR,(2,1) ,MAR (3,1)... | " |
| 14 | MAR (1,2) MAR,(2,2) ,MAR (3,2)... | " |
| | ------- | " |
| | ------- | " |
| | MRH (1,1), MRH (2,1), MRH (3,1) | " |
| | MRH (1,2), MRH (2,2), MRH (3,2) | " |

| VARIABLE | FORMAT |
|---|---|
| --- | Free |
| --- | " |
| HI (1,1) HI (1,2) HI (1,3) | " |
| HI (1,2) HI (2,2) HI (3,2) | " |
| --- | |
| --- | |

5.2.2.1.2 Data Element "INDATA 5" (For main programs TMAIN 5, TMAIN 5T and TMAIN 5V)

| CARD NO | VARIABLE | FORMAT |
|---|---|---|
| 1 | LN, LLN | 16I5 |
| 2 | VVIS, ABR | Free |
| 3 | AI, AH, AV, AP | " |
| 4 | EPS, MAXIT, OMEGA, ARBP | " |
| 5 | DX, DY, DZ | " |
| 6 | CC | " |
| 7 | DT | " |
| 8 | TAI, TAH, TAV | " |
| 9 | A,B,C | " |
| 10 | TO | " |
| 11 | EUL, DW, CB | " |
| 12 | TAMB, AKT, TAUX, TAUY | " |
| 13 | NIN, NOUT | " |
| 14 | I, J, K, U (I, J, K), V(I, J, K), T(I, J, K) | " |
| | --- | " |
| | (for inlet) | " |
| | --- | " |
| | I, J, K, U(I, J,K), V(I,J,K) | " |
| | (for inlet) | " |
| | --- | " |

5.2.2.1.3 Data Element INDATA 6 (For main program TMAIN6)

Same as first 12 lines of data element INDATA5.

5.2.2.2 Format of Input for Stratified Conditions:

There are four elements for stratified type conditions and they are (1), DATAML,(2) DATAML 5, (3) DATAML 6 and (4) ITLK1 and these data elements go with the main programs (1) TMAIN 4B, (2) TMAIN 5B, TMAIN 5TB and TMAIN 5VB, (3) TMAIN 6B and (4) TMAIN 4TB respectively. The data that goes with these data elements is explained in the next sections. The main programs that goes with the elements is given in the brackets.

5.2.2.2.1 Data Element "DATAML" (for main program TMAIN 4B)

| CARD NO | VARIABLE | FORMAT |
|---|---|---|
| 1 | LN, LLN | 16I5 |
| 2 | VVIS, ABR | Free |
| 3 | AI, AH, AV, AP | " |
| 4 | EPS, MAXIT, OMEGA, ARBP | " |
| 5 | DX, DY, DZ | " |
| 6 | CC | " |
| 7 | DT | " |
| 8 | TAI, TAH, TAV | " |
| 9 | A,B,C | " |
| 10 | TO | " |
| 11 | EUL, CW, CB | " |
| 12 | TAMB, AKT, TAUX, TAUY | " |
| 13 | CONS, AVMX, AVMN | " |
| 14 | AMINT (1,1), AMINT (1,2) | " |
| 15 | AMINT (2,1), AMINT (2,1) | " |
| 16 | AMINT (3,1), AMINT (3,2) | " |

| VARIABLE | FORMAT |
|---|---|
| --------- | Free |
| --------- | " |
| MAR (1,1), MAR (2,1) MAR (3,1)... | " |
| MAR (1,2), MAR (2,2) MAR (3,2)... | " |
| --------- | " |
| --------- | " |
| MRH (1,1), MRH (2,1), MRH (3,1)... | " |
| MRH (1,2), MRH (2,2), MRH (3,2)... | " |
| --------- | " |
| --------- | " |
| HI (1,1), HI (1,2), HI (1,3)... | " |
| HI (2,1), HI (2,2), HI (2,3)... | " |
| --------- | " |
| --------- | " |

5.2.2.2.2 Data Element "DATAML5" (For main programs TMAIN 5B, TMAIN 5TB and TMAIN 5VB)

| CARD NO | VARIABLE | FORMAT |
|---|---|---|
| 1 | LN, LLN | 1615 |
| 2 | VVIS, ABR | Free |
| 3 | AI, AH, AV, AP | " |
| 4 | EPS, MAXIT, OMEGA, ARBP | " |
| 5 | DX, DY, DZ | " |
| 6 | CC | " |
| 7 | DT | " |
| 8 | TAI, TAH, TAV | " |
| 9 | A,B,C | " |
| 10 | TO | " |

| CARD NO. | VARIABLE | FORMAT |
|---|---|---|
| 11 | EUL, CW, CB | Free |
| 12 | TAMB, AKT, TAUX, TAUY | " |
| 13 | CONS, AVMX, AVMN | " |
| 14 | NIN, NOUT | " |
| 15 | I, J, K, U (I,J,K), V (I,J,K), T(I,J,K) | " |
| | --- | " |
| | (for inlet) | " |
| | --- | " |
| | I, J, K, U (I, J, K), V (I, J, K) | " |
| | --- | " |
| | (for inlet) | " |
| | --- | " |

5.2.2.2.3 Data Element "DATAML6" (For main program TMAIN 6B)

Same as first 13 lines of DATAML 5

## 6.0 Sample Case, Near Field

6.1 Problem Statement: Florida Power and Light Company (FPL) has a fossil fuel power plant situated at Cutler Ridge site on the South Biscayne Bay. The discharge rate is $1.2 \times 10^7$ $cm^3/sec$ of water at temperature $35.9^oC$. It is required to find three-dimensional velocity and temperature distributions in the near-field (ie, in the region where the effects of the thermal discharge are noticeable) for the following environmental conditions.

Air temperature = $29.5^oC$

Initial temperature of Bay = $28.0^oC$

Wind speed = 1.2 m/sec, N-W

Average depth of bay = 1.2 m

Discharge width = 10 m

6.2 Choice of Programs: As velocity and temperature distributions are required, and, since depth is constant, the main program to be used is TMAIN 1. If the depth is variable TMAIN2 has to be used.

6.3 Calculation of Input Parameters: In this section, the specification of grid system, reference quantities and discharge velocity chosen will be presented first, followed by the actual calculation of the input data quantities as they appear in the main program.

6.3.1 Grid System: The Remote Sensing data and ground truth data was available for the Cutler Ridge site, (Fig.6.1) and it is used to determine the size of the domain. The data shows that the effects of dis-

charge would be in the region of 500 meters along the discharge axis (y-axis) and 425 meters across the axis of discharge. So, a domain of 500 m X 425 m is selected in the horizontal plane. A grid size of 25 m X 25 m is used. This would give 21 X 18 nodes in the horizontal plane. There are 5 nodes in the vertical direction. This gave a total of 21 X 18 X 5 nodes. The coordinate system and grid system are shown in Figs. (6.2 and 6.3). The MAR and MRH selection are explained in subroutine READ3.

6.3.2 Calculation of Discharge Velocity: The actual discharge width at the site is 10 meters. But, in the numerical model the discharge is 25 meters. So, the discharge velocity is calculated by balancing the mass flow as explained below.

For the numerical problem, with the above grid system, the discharge into the basin would be equal to ( v X 25 X 1.2 X $10^4$) X 2 $cm^3$/sec. The actual discharge volume = 1.2 X $10^7$ $cm^3$/sec.

ie (v X 25 X 1.2 X $10^4$) X 2 = 1.2 X $10^7$

ie v = 20 cm/sec

Velocity at discharge or Inlet velocity for the model is 20 cm/sec.

6.3.3 Reference Quantities: Reference Length = L = Maximum length of domain = 500 meters. Horizontal reference eddy viscosity is calculated using the following formula (Christodoulow et al, 1976)

$$Aref = 0.002\ L^{4/3}$$

Where L is the reference length of the domain in centimeters.

$$Aref = 0.002\ (500 \times 100)^{4/3} = 3553.3\ cm^2/sec$$

For Near Field problems about 2.8 times the calculated value was found suitable.

Aref = 10,000 $cm^2$/sec is used

Vertical reference eddy viscosity = 1 $cm^2$/sec

Horizontal reference eddy diffusivity = 10,000 $cm^2$/sec

Vertical reference eddy diffusivity = 1 $cm^2$/sec

Reference velocity = discharge velocity = 20 cm/sec

Reference temperature = 35.9°C

Reference depth = Average depth of bay = 1.2 meters

6.3.4 Calculation of Input data as it appears in the main program TMAIN:

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 1 | IRUN |

IRUN = 0, for the first time and

IRUN =1, for later runs

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 2 | LN |

LN can be any number depending on the number of cycles required and the total time the program has to be run. It is always advised to run the program for 10 or 15 cycles and check how the model is running.

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 3 | LLN |

LLN is Total Number of Cycles

LN

If LN is 15 and LLN is 2, then the model will give output after 15 and 30cycles.

| CARD NO. | FORTRAN STATEMENT |
|---|---|
| 4 | VVIS, ABR |

$$VVIS = \frac{Av}{Aref} = \frac{1}{10,000} = 0.0001$$

$$ABR = \frac{1}{\text{Rossby Number}} = \frac{1}{Rb} = \frac{fL}{Uref}$$

Where f is the coriolis function, L is maximum domain length and Uref is reference velocity, $f = 2\Omega \sin \phi$

Where $\Omega$ is angular velocity of earth, $\phi$ is the latitude angle.

Since Rb is large for near field ABR ≃ 0.0

| CARD NO. | FORTRAN STATEMENT |
|---|---|
| 5 | AI, AH, AV, AP |

AI is the coefficient infront of inertia term = 1.0

$$AH = \frac{1}{Re} = \frac{Aref}{Uref\ L} = \frac{10,000}{20 \times 500 \times 100} = \frac{1}{100} = 0.01$$

$$AV = \frac{1}{\varepsilon^2 Re} = \frac{L^2}{H^2\ Re} = \frac{500 \times 500}{1.2 \times 1.2 \times 100} = 1736.0$$

AP is coefficient infront of pressure term = 1.0

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 6 | EPS,MAXIT,OMEGA,ARBP |

EPS is the convergence factor and should be set equal to the convergence required. A value of EPS = 0.01 is a good value to start.

MAXIT: is the maximum number of iterations in the solution of Poisson equation for surface pressure. This depends upon the accuracy needed. A value of 50 is a good starting point.

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 7 | DX, DY, DZ |

$$DX = DY = \frac{\Delta X \text{or} \Delta Y}{L} = \frac{25}{500} = 0.05$$

$$DZ = \frac{1}{(KN-1)} = \frac{1}{4} = 0.25$$

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 8 | TAI,TAH,TAV |

TAI is the coefficient infront of convective terms in the energy equation = 1

TAH = AH If turbulent Prandtl number is equal to 1.

TAV = AV If turbulent Prandtl number is equal to 1.

$$TAV = \frac{1}{\varepsilon^2 Re} = 1736.0$$

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 9 | A,B,C |

A,B,C are constants in the equation of state and there values are

A = 1.000428

B = -0.000019

C = -0.0000046

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 10 | TO |

To is the reference temperature = 28.0°C

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 11 | AKT, EUL, CW, CB |

AKT is non-dimensional heat transfer coefficient = $\frac{KsH}{Bz}$

Where Ks is surface heat transfer coefficient, H is the reference depth and Bz = ρCp Bv

Where ρ is density, Cp specific heat at constant pressure, Bv is vertical diffusivity = 1 $cm^2/sec$

Bz = 62 $lbm/ft^3$ = 68.05 $slugs/m^2$

For Ks = 100 BTU/°F - $Ft^2$-day = 56.5 cal/°C - $m^2$ sec.

AKT = 0.0627

EUL = $\frac{g\ H}{Uref^2} = \frac{980 \times 1.2 \times 100}{20X20} = 294$

CW and CB are temperature gradient at the vertical boundaries and the bottom respectively and are equal to zero.

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 12 | AA, BB, CC |

AA, BB are non-dimensional input velocities. Since the discharge velocity is non dimensionalized with reference to discharge velocity AA and BB are equal 1.0 and 1.0.

CC is non dimensionalized depth and equal to 1.0 for a constant depth case.

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 13 | TLL, TMM |

TLL and TMM are the non-dimensional discharge temperatures.

$$TLL = TMM = \frac{35.9 - 28.0}{28.0} = 0.282$$

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 14 | TAMB |

TAMB is the air temperature = $29.5^{o}C$

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 15 | TAUX, TAUY |

TUAX, TAUY are the non-dimensional wind shear in x and y directions respectively. Wind shear is obtained from the Wilson (1960) curve which is given in Fig. (6.4). For a wind speed of 1.2 m/sec, the Wilson curve gives the Shear stress ($\tau^{w}$) equal to 0.049 dynes/$cm^2$.

$\tau_{xz}$ = Surface shear in x direction = $\tau_w \text{Cos } 45^{o} = 0.035$

$\tau_{yz}$ = Surface shear in y direction = $\tau_w \text{Sin } 45^{o} = 0.0345$

$$TAUX = \pm \frac{H}{Uref} \left(\frac{\tau xz}{Az}\right) = \frac{1.2 \times 100}{20} \times \frac{0.035}{1} = \pm 0.1768$$

$$TUAY = \pm \frac{H}{Uref} \left(\frac{\tau yz}{Az}\right) = \frac{1.2 \times 100}{20} \times \frac{0.035}{1} = \pm 0.1768$$

The direction +ve or -ve is decided as follows. TAUX (or TAUY) is +ve when the wind force is in the opposite direction of x (or Y) and vice versa. For NW wind, TAUX = +0.1768, TAUY = -0.1768

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 16 | DT |

In order to determine the time step (DT), the stability analysis has to be made which is done as follows.

Convective $\Delta t < \frac{\Delta x}{U} = \frac{25 \times 100}{20} = 125$ sec

Viscous $\Delta t < \frac{\Delta x}{2AH} = \frac{(25 \times 100)^2}{2 \times 10,000} = 312$ sec

$\Delta t < \frac{\Delta x^2}{2AV} = \frac{(0.3 \times 100)^2}{2} = 450$ sec

From above it can be seen that convective criterion is dominant.

ie $\Delta t < 125$ sec

About ¼ times this value is reasonable to use $\Delta t = \frac{1}{4}(125)$

$= 30$ sec (approx)

This value of $\Delta t$ is non-dimensionalized and used as the value for DT.

$DT = \frac{\Delta T}{t_{ref}}$

Where $t_{ref} = \frac{L}{Uref} = \frac{500 \times 100}{20} = 250$ sec

$DT = \frac{30}{2500} = 0.01$

## 6.4 Sample Input

| SYMBOLS | VALUES & FORMAT |
|---|---|
| * 1. IRUN | ␣ ␣ ␣ ␣ 0 |
| ** 2. LN | ␣ ␣ ␣ 5 0 |
| ***3. LLN | ␣ ␣ ␣ ␣ 1 |
| 4. VVIS, ABR | 0.0001, 0.0 |
| 5. AI, AH, AV, AP | 1.0, 0.01, 1736.1, 1.0 |
| 6. EPS, MAXIT, OMEGA, ARBP | 0.01, 50, 1.8, 1.0 |
| 7. DX, DY, DZ | 0.05, 0.05, 0.25 |
| 8. TAI, TAH TAV | 1.0, 0.01, 1736.1 |
| 9. A, B, C | 1.000428, -0.00002, -0.0000048 |
| 10. TO | 28.0 |
| 11. AKT, EUL, CW CB | 0.627, 294.0, 0.0, 0.0 |
| 12. AA, BB, CC | 1.0, 1.0, 1.0 |
| 13. TLL, TMM | 0.282, 0.282 |
| 14. TAMB | 29.5 |
| 15. TAUX, TAUY | 0.1768, -0.1768 |
| 16. DT | 0.01 |

NOTE: ␣ = blank space

* 0 for the first run and 1 for later runs

** number of cycles required

*** Prints after 50 cycles in this case. If it is equal to 2, it will print after 50 and 100 cycles.

## 6.5 Program Execution Procedure (For Constant Depth)

In order to execute the programs for the near field constant depth case, the following steps have to be followed.

(1) Input Parameters: The calculation of input parameters is explained in the sample problem section (6.3). The input parameters depend on the discharge conditions, ambient conditions and reference quantities chosen.

(2) First Run: In order to obtain three dimensional velocities and temperatures, the main program to be executed is TMAIN1. In the programs, there are two units, one is read unit designated as, unit 7 and the other is write unit designated as, unit 8. During the first run, there is no need for unit 7, and unit 8 has to be provided to store results. This store unit,'8', would be a magnetic tape.

(3) Continuation of a Run: For extending the results, the run has to be continued. The magnetic tape which was'unit 8' in the first run would now be read'unit 7' for reading the previous results. Another magnetic tape is to be provided as unit 8 for storing the extended run results. The above procedure can be repeated until the results are obtained for the desired time. It is to be noted that for the first run IRUN=0 and for continuation of a run IRUN=1.

The following are a set of control cards that were used on UNIVAC 1106 computer in order to run the near field programs for a constant depth case. The explanation of the control card is given in the brackets.

CARD 1

@ RUN

(Schedule a new run for initiation)

CARD 2

@ ASG,A SKM*DULL

(All parameters on @ ASG Control Statement are optional except file name.)

A-specifies that the file being assigned is currently catalogued SKM is the qualifier and DULL is file name)

CARD 3

@ PACK SKM*DULL

(Packs the non-deleted elements of a program file, by rewriting the file and eliminating the deleted elements).

CARD 4

@ PREP SKM*DULL

(Prepares an entry point table for program file, for use by the @ MAP process or in searching a LIB specified program file to satisfy undefined symbols)

CARD 5

@ ASG, T 8., 16N, BAY1

(T-specifies that the file to be assigned temporary and allows it to have a name the same as that of an unassigned catalogued file. BAY1 is the name of the tape.)

CARD 6

@ MAP

(Call the MAP process or (the collector) to collect a specified set of relocatable elements, and produce from this an executable program which is in an absolute element format)

CARD 7

IN SKIM*DULL. TMAIN1

(TMAIN1 is the main program which would be executed)

CARD 8

LIB SKM*DULL

(specifies file as a library to be searched)

CARD 9

@ XQT

(Initiates the execution of a program which is in an absolute element format)

CARD 10

Data cards as shown in section (6.4) of the sample problem

— — —

CARD 26

@ FIN

(Terminates a run)

## 6.6 Program Execution Procedure (for variable depth)

In order to execute the programs for the near field variable depth case, the following steps have to be followed.

(1) Input Parameters: The calculation of input parameters is explained in the sample problem section (6.3). The input parameters depend on the discharge conditions, ambient conditions and reference quantities chosen.

(2) The depths in the receiving basin at every grid point have to be obtained. If the depth contours of the receiving basin are available, the depths at all grid points can be obtained by interpolation. These depths have to be non-dimensionalized with respect to reference dpeth. The non-dimensionalized depths have to be read in before execution of the program. In the program, the subroutine HEIGH1 reads the depths, in the direction perpendicular to the discharge axis ie along I or x axis. The depths go as input cards in the format given by subroutine HEIGH1 during the first run after the input card which reads "TAUX, TAUY".

(3) First Run: In order to obtain three dimnsional velocities and temperatures, the main program to be executed is TMAIN 2. In the programs, there are two units. One is read unit designated as, unit 7 and the other is write unit designated as, unit 8. During the first run, there is no need for unit 7, and unit 8 has to be provided to store results. The unit '8', would be a magnetic tape.

(4) Continuation of a Run: For extending the results, the run has to be continued. The magnetic tape which was used as store unit (unit 8) in the first run would now be the read unit (unit 7) for reading the previous results. Another magnetic tape

has to be provided as unit 8 for storing the extended run results. The above procedure can be repeated until the results are obtained for the desired time period. It is to be noted that for the first run IRUN=0 and for continuation of a run IRUN=1.

The following are a set of control cards that were used on UNIVAC 1106 computer in order to run the near field programs. The explanation of the control cardsis given in the brackets.

CARD 1

@ RUN

(Schedule a new run for initiation)

CARD 2

@ ASG,A SKM*DULL

(All parameters on @ ASG Control Statement are optional except file name.

A-specifies that the file being assigned is currently catalogued SKM is the qualifier and DULL is file name.)

CARD 3

@ PACK SKM*DULL

(Packs the non-deleted elements of a program file, by rewriting the file and eliminating the deleted elements.)

CARD 4

@ PREP SKM*DULL

(Prepares an entry point table for program file, for use by the a) MAP processor in searching a LIB specified program file to satisfy undefined symbols).

CARD 5

@ ASG,T 8., 16N, BAY1

(T-specifies that the file to be assigned temporary and allows it to have a name the same as that of an unassigned catalogued file. BAY 1 is the name of the tape)

CARD 6

@ MAP

(Call the MAP processor (the collector) to collect a specified set of relocatable elements, anp produce from this an executable program which is in an absolute element format)

CARD 7

IN SKM * DULL TMAIN 1

(TMAIN 1 is the main program which would be executed)

CARD 8

LIB SKM*DULL

(Specifies file as a library to be searched)

CARD 9

@ XQT

(Initiates the execution of a program which is in an absolute element format)

CARD 10

Data cards as shown in section (6.4) of the sample problem.

CARD 26

@ ADD SKM*DULL HEIGH2

(This adds the depths which is stored in the file as an element under the name HEIGH2. This card should be taken out after the first run ie the input cards which read non-dimensionalized depths are included for IRUN=0 only)

CARD 27

@ FIN

(Terminates a run)

## 6.7 Sample Output

The output of the near field rigid lid program consists of the following.

1. Input parameters
2. MAR and MRH matrices
3. Number of iterations (ITN) in the Poissons Equation and residue (EX)
4. Surface Pressure
5. U and V at all grid points (non-dimensional) ie U and V at I=1 to IN, J=1 to JN and K=1 to KN
6. WH at all grid points (non-dimensional) ie WH at IN- 1 to IWN, JW= 1 to JWN and K= 1 to KN
7. Temperatures both non-dimensional and dimensional at all grid points.
   ie T at I=1 to IN, J=1 to JN and K= 1 to KN.

Selected portions of a sample unit are shown in the next few pages.

```
EXOT

 AI=     .1000000+01
 AH=     .9000000-02
 AV=     .1900000+04
 AP=     .1050000+01
 DX=     .4500000-01
 DY=     .4500000-01
 DZ=     .2500000+00
 DT=     .5000000-02
 DL2=     .2025000-02
 MAXIT=    25
 EPS=     .1000000-01
 OMEGA=      .1800000+01
 ARRP=     .1000000+01
 TAUX=     .1767800+00
 TAUY=    -.1767800+00
 TTOT=     .1539999+01

 MAR       5  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 10

 MAR       3 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  4

 MAR       3 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11  4

 MPH       3  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  7

 MPH      10  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  8

 MPH      10  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  8

 TAT=     .1000000+01
 TAH=     .9000000-02
 TAV=     .1900000+04
 CE=     .0000000
 CM=     .0000000
 AKT=     .1256000-01
 TACF=     .2800000+02
 OPSF=     .1025103+01
 SUL=     .2943000+03
 A=     .1029431+01
 B=    -.2000000-04
 C=    -.4800000-05
 TC=     .5057143-01
 TO=     .2800000+02
```

ITN= 2 EX= .9649403-02

IN= 1

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| .9712510+00 | .9796368+00 | .9891320+00 | .9968368+00 | .1000148+01 | .9994304+00 | .9956555+00 | .9902621+00 |
| .9852902+00 | .9816037+00 | .9795445+00 | .9793325+00 | .9804468+00 | .9825214+00 | .9850702+00 | .9878384+00 |
| .9903966+00 | .9929032+00 | .9951658+00 | .9973141+00 | .9992978+00 | | | |

IN= 2

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| .9762118+00 | .9959485+00 | .1007876+01 | .1017817+01 | .1022069+01 | .1019648+01 | .1013337+01 | .1004318+01 |
| .9959319+00 | .9891763+00 | .9845265+00 | .9822157+00 | .9817503+00 | .9826486+00 | .9845964+00 | .9870521+00 |
| .9897485+00 | .9923119+00 | .9947692+00 | .9970712+00 | .9992371+00 | | | |

K= 1 I= 1

U-VELOCITY

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| .0000000 | -.8432708-01 | -.1250604+00 | -.1589122+00 | -.1872824+00 | -.2078608+00 | -.2200564+00 | -.2246015+00 |
| -.2227570+00 | -.2160208+00 | -.2062216+00 | -.1953686+00 | -.1850861+00 | -.1764437+00 | -.1698313+00 | -.1651218+00 |
| -.1619874+00 | -.1600189+00 | -.1588789+00 | -.1583312+00 | -.1582259+00 | -.1580208+00 | | |

V-VELOCITY

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| .0000000 | .5099698-01 | .6221541-01 | .7094028-01 | .7949291-01 | .8791187-01 | .9603377-01 | .1036965+00 |
| .1108334+00 | .1174145+00 | .1233038+00 | .1284693+00 | .1328277+00 | .1364991+00 | .1394777+00 | .1418969+00 |
| .1438332+00 | .1453362+00 | .1464828+00 | .1472607+00 | .1477346+00 | .1476875+00 | | |

K= 1 I= 2

U-VELOCITY

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| .0000000 | -.8432708-01 | -.1250604+00 | -.1589122+00 | -.1872824+00 | -.2078608+00 | -.2200564+00 | -.2246015+00 |
| -.2227570+00 | -.2160208+00 | -.2062216+00 | -.1953686+00 | -.1850861+00 | -.1764437+00 | -.1698313+00 | -.1651218+00 |
| -.1619874+00 | -.1600189+00 | -.1588789+00 | -.1583312+00 | -.1582259+00 | -.1582259+00 | | |

V-VELOCITY

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| .0000000 | .5099698-01 | .6221541-01 | .7094028-01 | .7949291-01 | .8791187-01 | .9603377-01 | .1036965+00 |
| .1108334+00 | .1174145+00 | .1233038+00 | .1284693+00 | .1328277+00 | .1364991+00 | .1394777+00 | .1418969+00 |
| .1438332+00 | .1453362+00 | .1464828+00 | .1472607+00 | .1477346+00 | .1477346+00 | | |

K= 1 IW= 1
WH-VELOCITY

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| .4396481-01 | .1357717+00 | .1217223+00 | .1202851+00 | .1169994+00 | .1128212+00 | .1074473+00 | .1017101+00 |
| .9581072-01 | .8722552-01 | .7758281-01 | .6593132-01 | .5578171-01 | .4510475-01 | .3651339-01 | .2909730-01 |
| .2239221-01 | .1702783-01 | .1130871-01 | .6696155-02 | -.1791012-02 | | | |

K= 1 IW= 2
WH-VELOCITY

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| .6881407-01 | .3270424-01 | .4115359-01 | .3409834-01 | .2782663-01 | .2448114-01 | .2172996-01 | .1836613-01 |
| .1723544-01 | .1459773-01 | .1236063-01 | .8892525-02 | .6274632-02 | .3241206-02 | .1785427-02 | .7434740-03 |
| .4196624-03 | .5323560-03 | .2234573-03 | .5595127-03 | -.3580152-02 | | | |

K= 1 I= 1
TEMPERATURE

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| .3662746-01 | .3762813-01 | .4110203-01 | .4415308-01 | .4527573-01 | .4361053-01 | .3916583-01 | .3266881-01 |
| .2527270-01 | .1817149-01 | .1223506-01 | .7833605-02 | .4898430-02 | .3117855-02 | .2126274-02 | .1615495-02 |
| .1370770-02 | .1261267-02 | .1215466-02 | .1197704-02 | .1191853-02 | .1191674-02 | | |

DENSITY

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| -.2938491-03 | -.3020329-03 | -.3304221-03 | -.3554534-03 | -.3646694-03 | -.3509908-03 | -.3145777-03 | -.2616223-03 |
| -.2017042-03 | -.1445479-03 | -.9705813-04 | -.6202592-04 | -.3873894-04 | -.2463884-04 | -.1678930-04 | -.1274924-04 |
| -.1081492-04 | -.9957290-05 | -.9593865-05 | -.9448523-05 | -.9404914-05 | -.9404914-05 | | |

K= 1 I= 2
TEMPERATURE

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| .3662746-01 | .3762813-01 | .4110203-01 | .4415308-01 | .4527573-01 | .4361053-01 | .3916583-01 | .3266881-01 |
| .2527270-01 | .1817149-01 | .1223506-01 | .7833605-02 | .4898430-02 | .3117855-02 | .2126274-02 | .1615495-02 |
| .1370770-02 | .1261267-02 | .1215466-02 | .1197704-02 | .1191853-02 | .1191853-02 | | |

DENSITY

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| -.2938491-03 | -.3020329-03 | -.3304221-03 | -.3554534-03 | -.3646694-03 | -.3509908-03 | -.3145777-03 | -.2616223-03 |
| -.2017042-03 | -.1445479-03 | -.9705813-04 | -.6202592-04 | -.3873894-04 | -.2463884-04 | -.1678930-04 | -.1274924-04 |
| -.1081492-04 | -.9957290-05 | -.9593865-05 | -.9448523-05 | -.9404914-05 | -.9404914-05 | | |

TEMPERATURES FOR K= 2

29.0 29.1 29.1 29.2 29.2 29.2 29.1 28.9 28.7 28.5 28.3 28.2 28.1 28.1 28.1 28.0 28.0 28.0 28.0 28.0 28.0 28.0

29.0 29.1 29.1 29.2 29.2 29.2 29.1 28.9 28.7 28.5 28.3 28.2 28.1 28.1 28.1 28.0 28.0 28.0 28.0 28.0 28.0 28.0

29.3 29.4 29.5 29.7 29.7 29.7 29.5 29.2 29.0 28.7 28.5 28.3 28.2 28.1 28.1 28.0 28.0 28.0 28.0 28.0 28.0 28.0

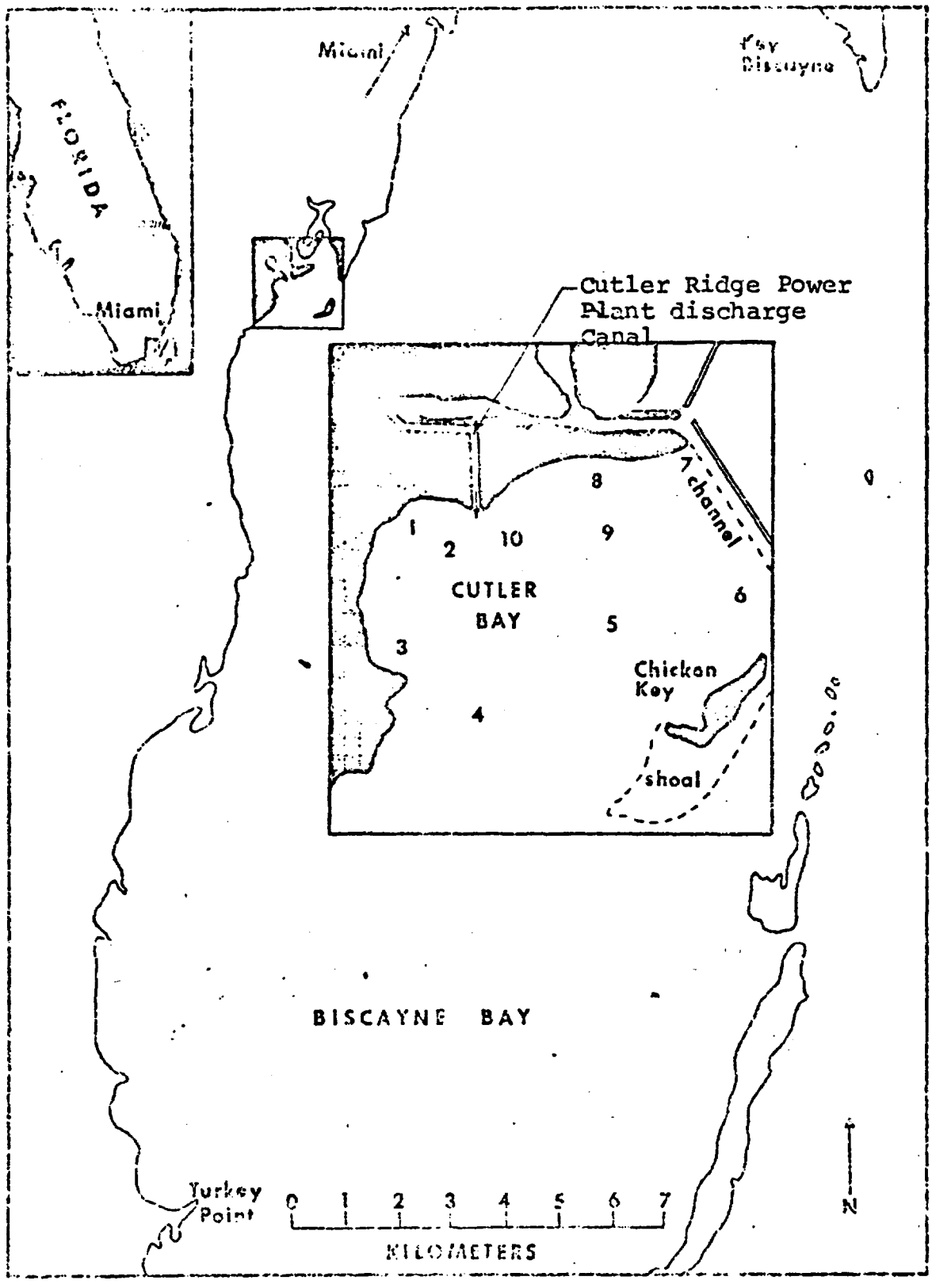



Fig. 6.1 Map showing Cutler Ridge Power Plant and discharge location.

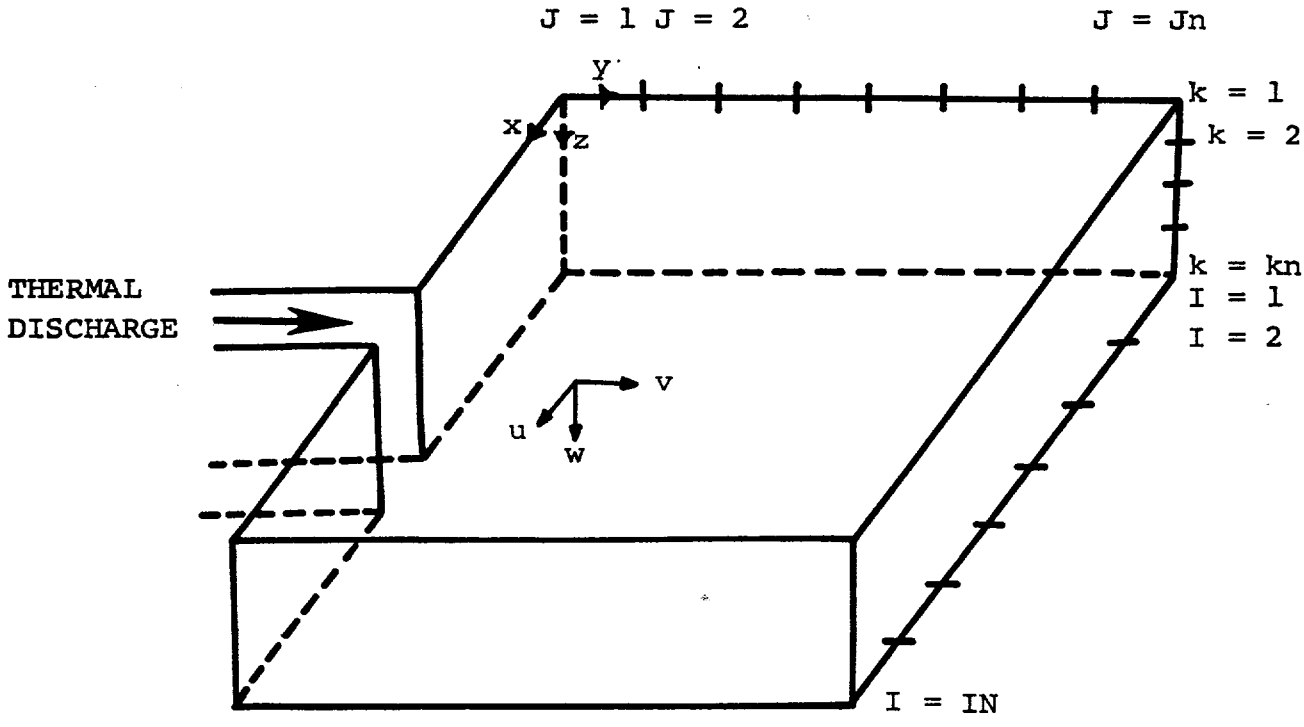



Fig. 6.2 Coordinate and Grid System

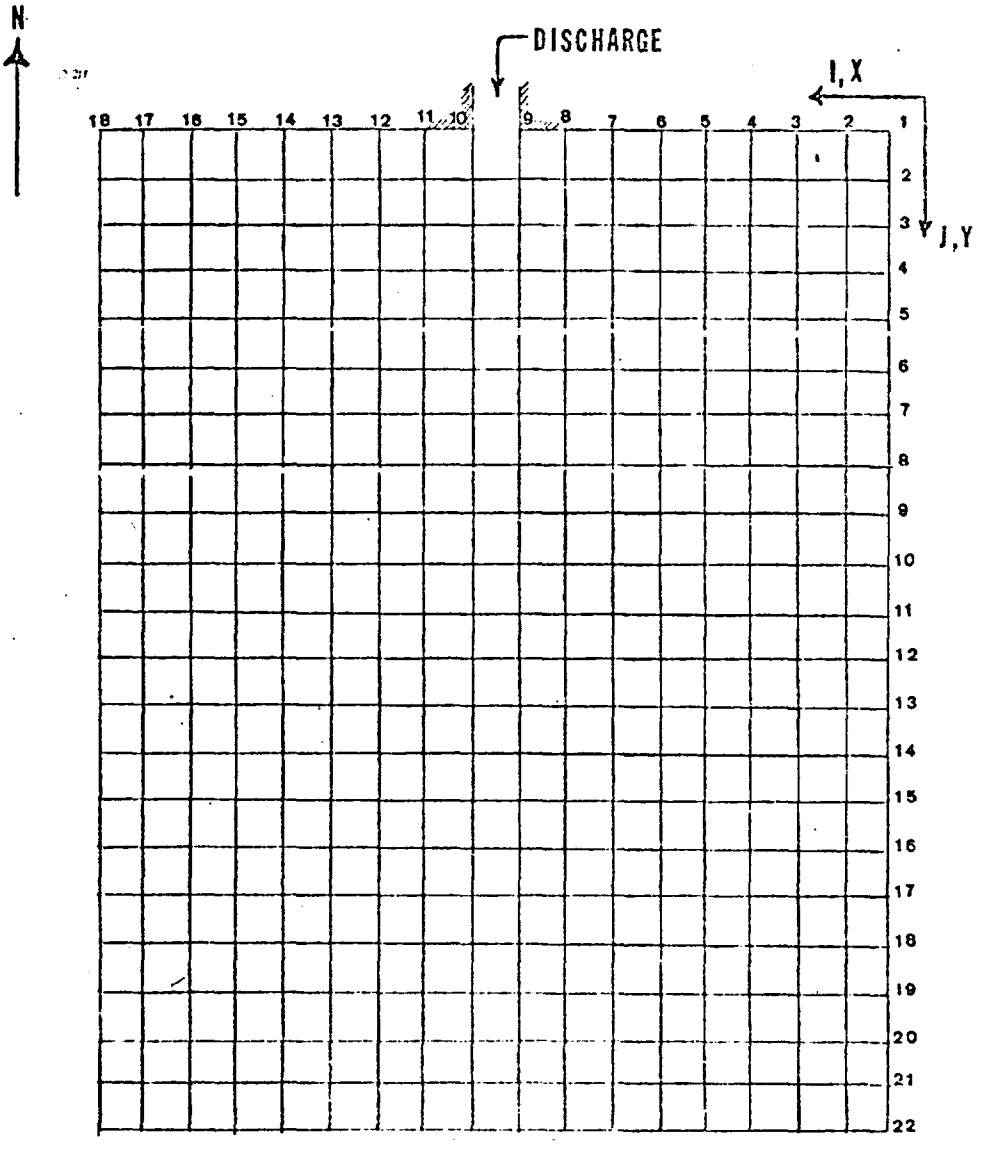



Fig. 6.3 . Grid System For Rigid Lid Near Field Model of Cutler Ridge Site

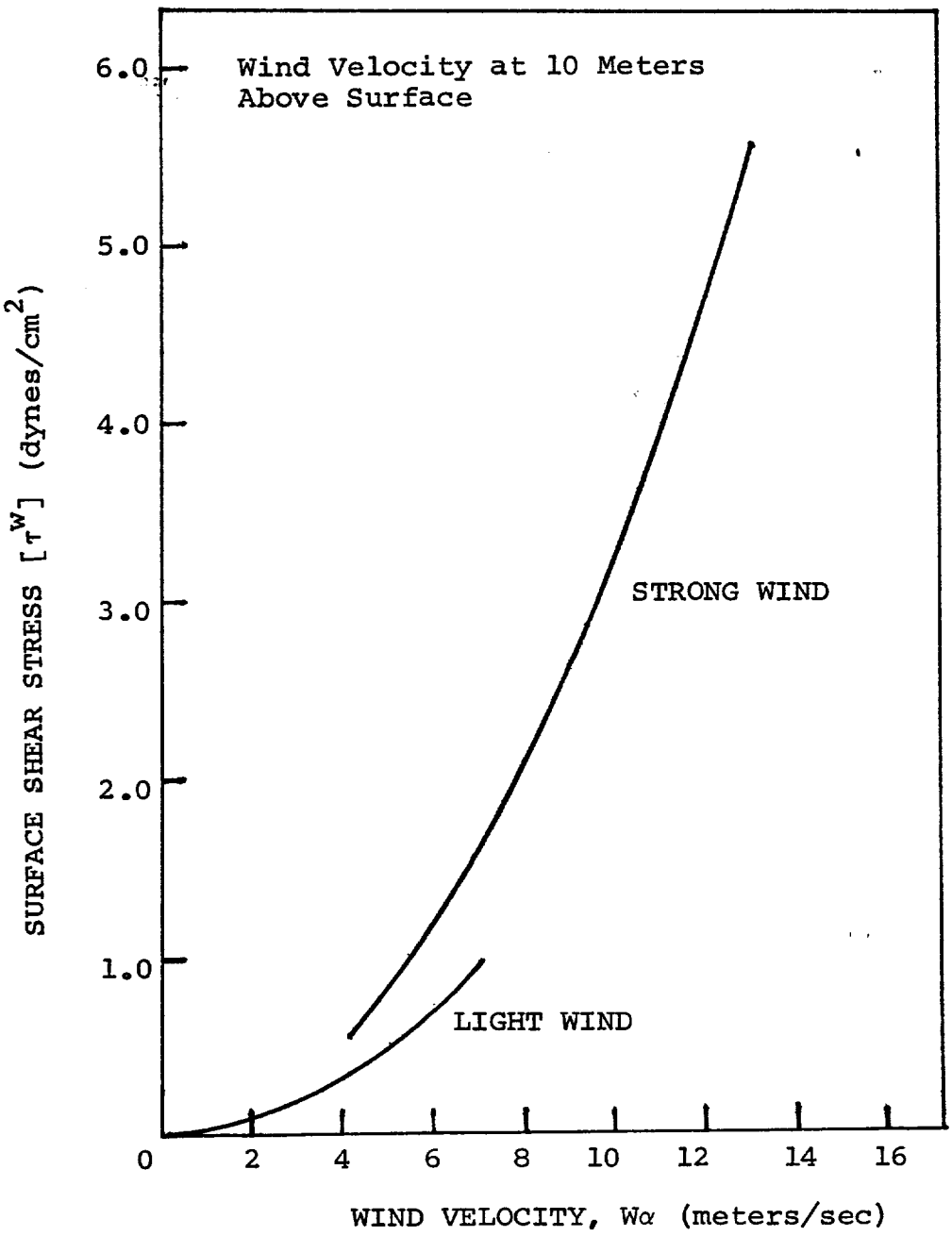



Figure 6.4 WIND SHEAR STRESS RELATION

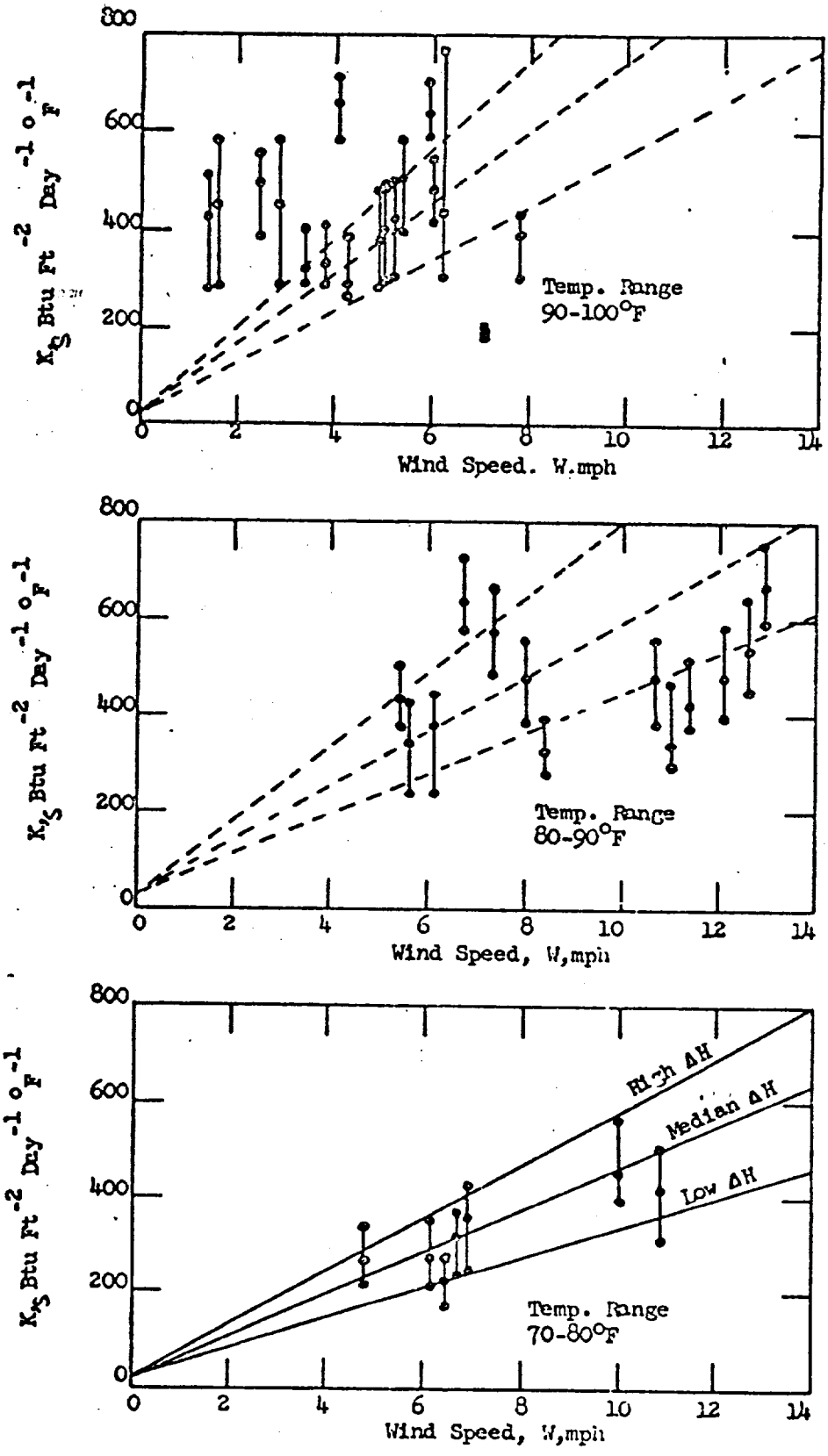



Fig: 6.5 Wind speed relationship

# SOME REPRESENTATIVE RESULTS FOR THE NEAR-FIELD SAMPLE CASE

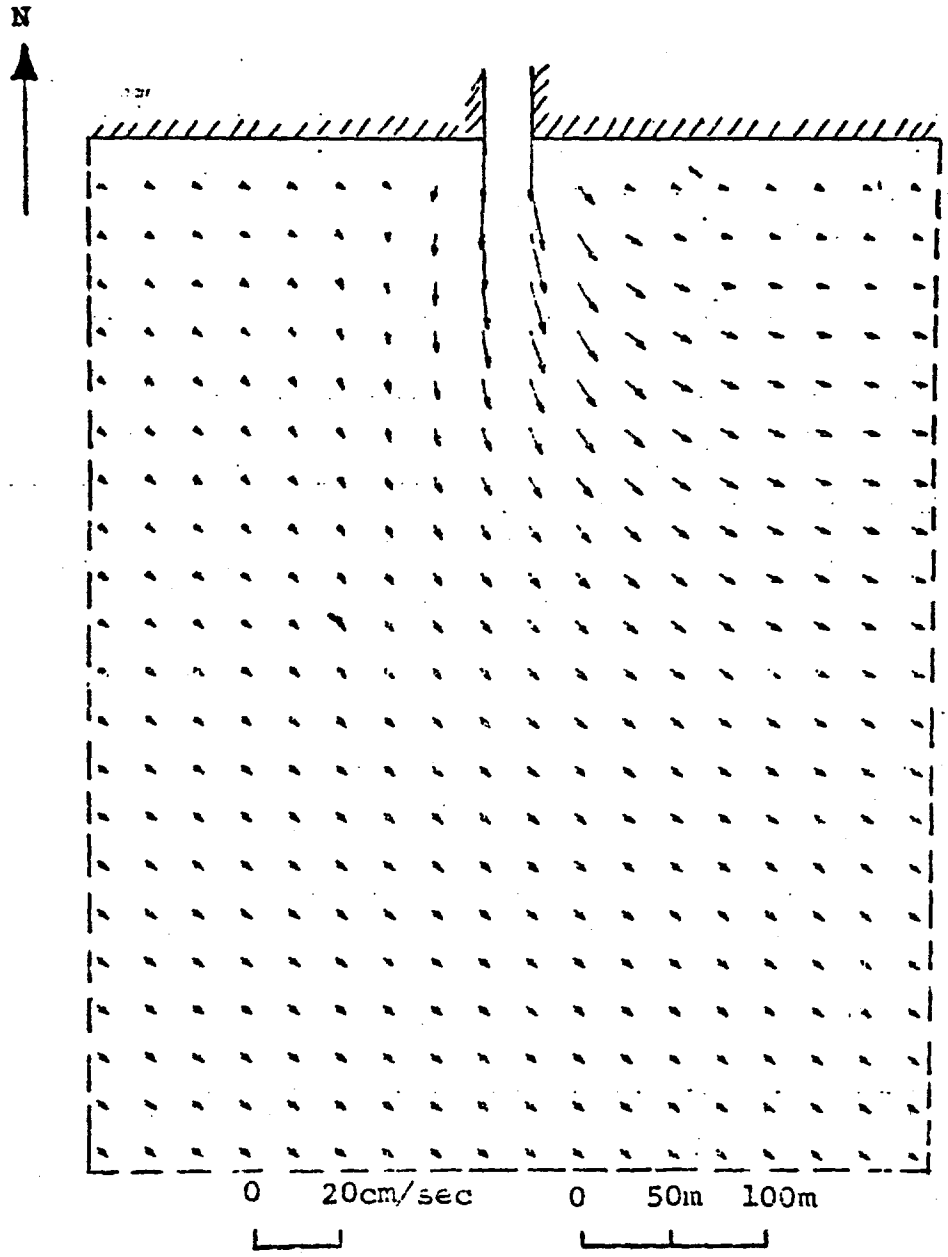



Fig. 6.6 Surface velocity distribution

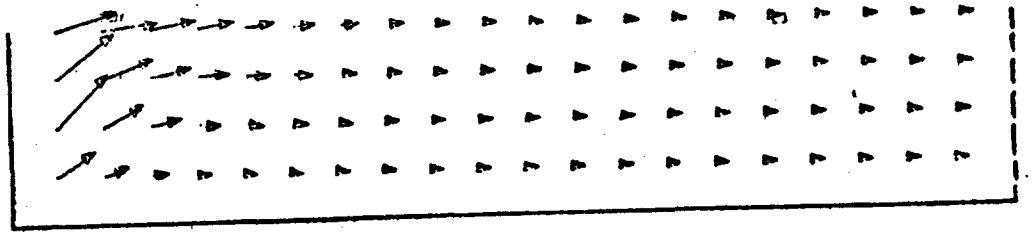


Fig. 6.7 Velocity distribution along the axis of discharge.

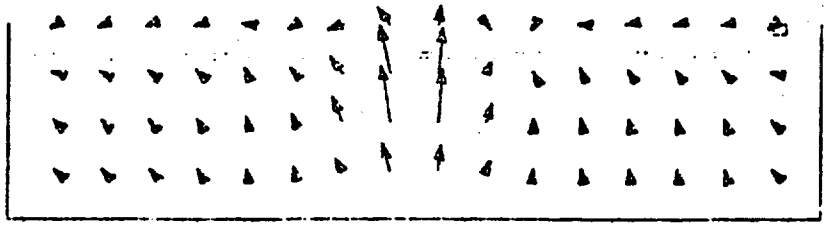
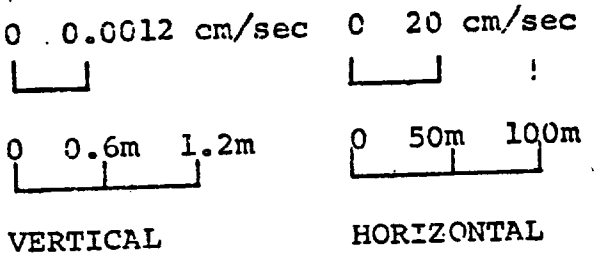


Fig. 6.8 Velocity distribution perpendicular to the axis of discharge.



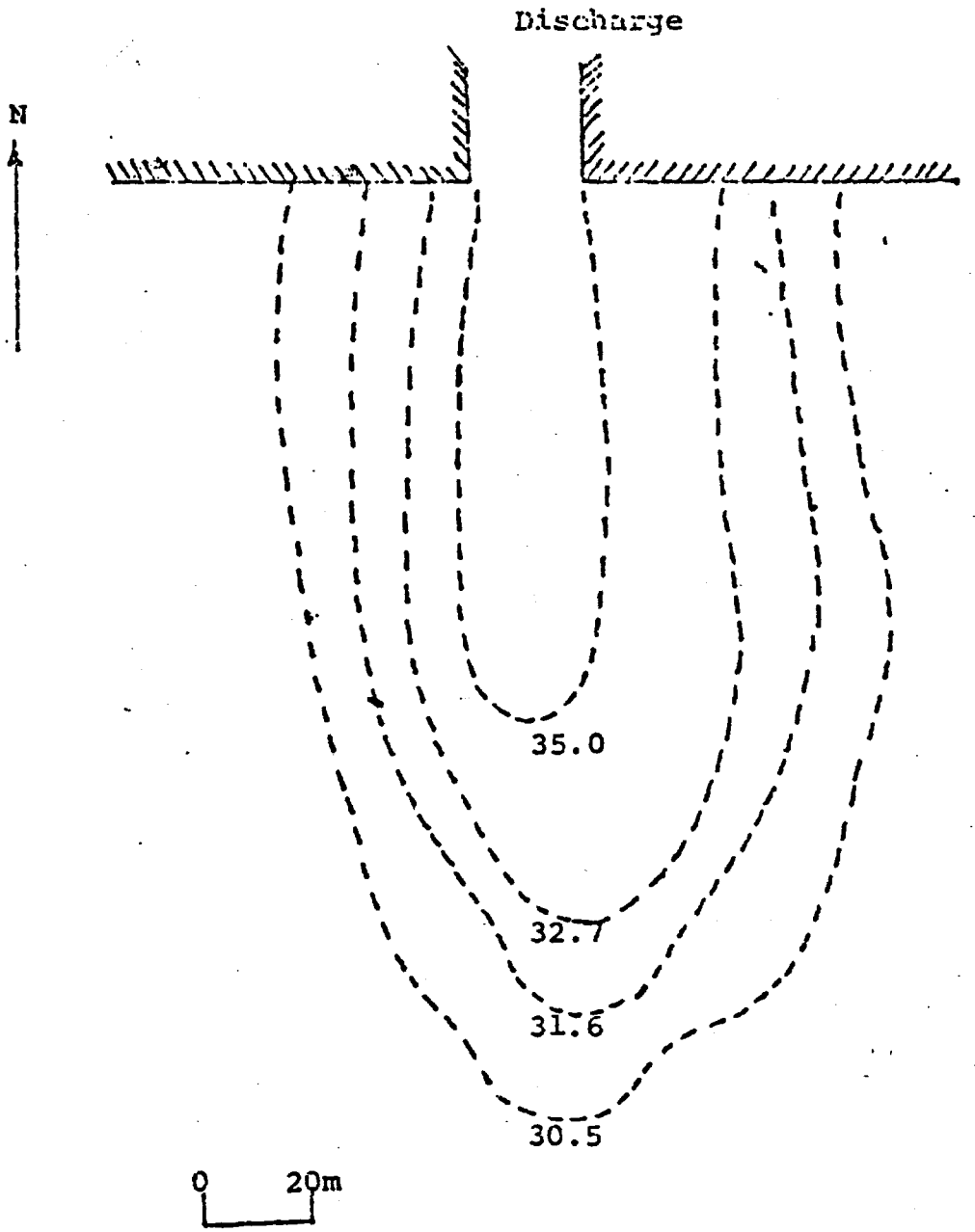
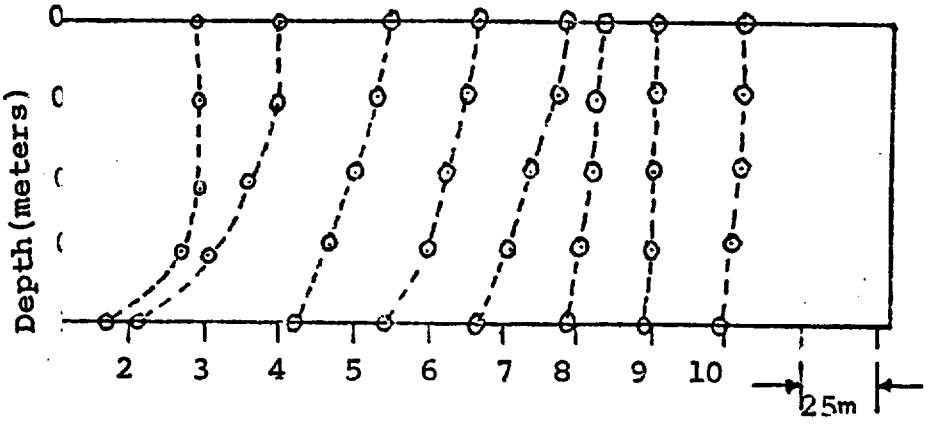



Fig. 6.9 Surface isotherms in the near field.




ance from discharge along Canal center line
eters.

Fig. 6.10 Vertical section isotherms along canal centerline for the near field.

## 7.0 Sample Case, Far Field (Unstratified Case, Pond)

7.1 Problem Statement: Duke Power Company of North Carolina operates a coal fired electric power plant at Lake Belews Creek in North Carolina. There are two power units and each has a capacity of 1143 MWe. Both these units use Belews Lake as a cooling reservoir. The lake, as shown in figure 7.1 consists of a pond and a main body of the lake. The pond and main body of lake are connected by a canal. The pond has a maximum depth of 13.72 meters (45') and is well mixed. The main body of lake has a maximum depth of 39.26 meters (130') and is thermally stratified. The discharge from the condensers is made into the pond. It is required to find three dimensional velocity and temperature deistributions in the pond and the main body of the lake for the following discharge and environmental conditions (these conditions are of August 26, 1976).

Discharge Volume = $228.9 \times 10^6$ kg/hr

Discharge temperature = $38.9^{\circ}C$

Air temperature = $25.0^{\circ}C$

Wind speed = 4.25 meters/sec $225^{\circ}N$

Total incident radiation = 43.19 calories/$cm^2$/hr

7.2 Choice of Programs for Unstratified Pond: Since the pond is unstratified, the main programs to be used are (1) TMAIN 4 (2) TMAIN5, TMAIN 5T, TMAIN 5V and (3) TMAIN 6. The data elements that go with these programs are (1) INDATA (2) INDATA 5 and (3) INDATA 6. If the initial conditions are known (ie Ground truth data and IR data), then the program TMAIN 4T can be used.

The data element that goes with it is IRPKI. TMAIN4T can be eliminated if the initial conditons are not known and one can still obtain velocity and temperature distributions using the model. The calculation of input parameters is explained in the next section.

7.3 Calculation of Input Parameters: In this section, the specification of the grid system and reference quantities chosen will be presented first followed by the actual calculation of the input data quantities as they appear in the main programs.

1) Grid System: A computational domain of 720 meters by 1680 meters in the horizontal plane as shown in Fig. 7-2 covers most of the significant portions of the Mixing Pond. The grid size chosen is 60 meters by 60 meters. This gives a 29 X13 mesh in the horizontal plane. The depth of the pond is 13.72 meters and is divided into five layers thus giving total mesh of 29 X 13 X 6 for the entire computational domain of the pond.

2) Reference Quantities: Reference quantities are used to non-dimensionalize the input parameters and they are given below.

Reference Velocity : 30cm/sec

Reference Temperature: $30^{o}C$

Reference Horizontal Length = Length of domain = 1680 meters

Reference Depth = Maximum depth of domain = 20 meters

Reference Horizontal eddy viscosity = 45,000 $cm^2/sec$

Reference Vertical eddy viscosity = 10 $cm^2/sec$

Pr = 1 is chosen

$\therefore$ Reference Horizontal eddy diffusivity = 45,000 $cm^2/sec$

Reference Vertical eddy diffusivity = 10 $cm^2/sec$

7.3.1 Calculation of Input data for data element "INDATA (for main program TMAIN4.)

The definition of the symbols are given in section (5.2.1).

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 1 | LN, LLN |

These are not calculated values and can be any number depending on the number of cycles required and the total time the program has to be run. It is always advised to run the program for 10 or 15 cycles and check how the model is running.

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 2 | VVIS, ABR |

$$VVIS = \frac{Av}{Aref} = \frac{1}{45,000}$$

$$ABR = \frac{1}{\text{Rossby Number}} = \frac{1}{RB} = \frac{f\ L}{Uref}$$

Where f is coriolis function, L is domain length and Uref reference velocity

$$f = 2\ \Omega\ \sin\ \phi$$

Where $\Omega$ is angular velocity of earth and $\phi$ is the latitude angle.

$$f = 2 \times \frac{2\pi}{24 \times 60 \times 60} \text{ Sin } \phi$$

The latitude of Lake Belews is $36^{o}$ - 16' - 15" North.

This gives $f = 0.8604 \times 10^{-4}$/sec

$$\text{Rossby Number} = \frac{Uref}{f\ L} = \frac{30}{0.8604 \times 10^{-4} \times 1680 \times 100} = 2.07159$$

$$\text{and } ABR = \frac{1}{RB} = \frac{1}{2.07159} = 0.482721$$

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 3 | AI, AH, AV, AP |

AI is the coefficient infront of inertia terms in the momentum equations (horizontal) = 1

$$AH = \frac{1}{Re}$$

$$\text{Where } Re = \frac{Uref \ L}{Aref} = \frac{30 \times 1680 \times 100}{45{,}000} = 112.2$$

$$AH = \frac{1}{112.2} = 0.008912$$

$$AV = \frac{1}{\varepsilon^2 Re}$$

$$\text{Where } \varepsilon = \frac{H}{L} = \frac{20}{1680} = 0.0118833$$

$$AV = \frac{1}{(0.0118833)^2 (112.2)} = 63.11$$

AP is the coefficient infront of pressure term and is equal to 1

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 4 | EPS, MAXIT, OMEGA, ARBP |

EPS is the convergence criterion and should be set equal to the convergence required. A value of EPS = 0.01 is a good value to start.

MAXIT is the maximum number of iterations in the solution of Poisson Equation for surface pressure. This depends upon the accuracy needed. A value of MAXIT = 50 is a good starting point.

OMEGA is relaxation factor and a value of 1.8 provides rapid convergence.

ARBP is arbitrary pressure and a value of 1.0 is used.

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 5 | DX, DY, DZ |

$DX = \frac{60}{1680} = 0.03565$

$DY = \frac{60}{1680} = 0.03565$

$DZ = \frac{1}{5} = 0.2$

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 6 | CC |

The value of CC = 1.0

(For constant depth CC = 1.0 and zero for variable depth)

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 7 | DT |

DT is the time step to be used. In order to obtain DT value, stability analysis has to be made in order to see what criterion to be used.

Convection Criterion

$$u \frac{\Delta t}{\Delta x} < 1$$

$$\text{or } \Delta t < \frac{\Delta x}{u} = \frac{60 \times 100}{30} = 200 \text{ sec.}$$

Viscous Criterion

$$\Delta t < \frac{\Delta x^2}{2AH} = \frac{60 \times 60 \times 10^4}{2 \times 45{,}000} = 0.04 \times 10^4 \text{ sec.}$$

$$\Delta t \quad \frac{\Delta z^2}{2A_v} = \frac{60 \times 60 \times 10^4}{2 \times 10} = 0.2 \times 10^4 \text{ sec.}$$

Diffusive Criterion

$$\Delta t < \frac{\Delta x^2}{2BH}$$

$$\Delta t = \frac{60 \times 60 \times 10^4}{2 \times 45{,}000}$$

$$= 400 \text{ Sec}$$

The lowest value here is non-dimensionalized with reference to time ($t_{ref}$)

$$t_{ref} = \frac{L}{Uref} = \frac{1680 \times 100}{30} = 5.6 \times 10^3 \text{ Sec} = 1.33 \text{ hrs.}$$

The non dimensional time step (DT) = $\frac{\Delta t}{tnet}$

$$DT = \frac{\Delta t}{tref} = \frac{200}{5.6 \times 10^3} = 0.035$$

About ¼ of this value is reasonable to use

$$\therefore DT = \tfrac{1}{4}(0.035) = 8.92 \times 10^{-3}$$

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 8 | TAI, TAH, TAV |

TAI is the coefficient infront of convective terms in the energy equation and is equal to 1.

If Pr = 1.0 then Aref = Bref, since Prnadtl number is defined as equal to the ratio of Aref over Bref, and

TAH = AH

TAV = AV

TAI = 1.0

TAH = 0.008912

TAV = 63.11

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 9 | A,B,C |

A,B,C are coefficients in the equation of state and they are constants. The values are

A = 1.000428

B = -0.000019

C = -0.0000046

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 10 | TO |

TO is the reference temperature. In this case the reference temperature is taken as $30^{o}C$.

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 11 | EUL, CW, CB |

$$EUL = \frac{gH}{U_{ref}^2} = \frac{980 \times 20 \times 100}{30 \times 30} = 2180$$

CW : Temperature gradient at the vertical boundaries and is equal to zero in this case.

CB : Temperature gradient at the bottom and is equal to zero in this case.

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 12 | TAMB, AKT, TAUX, TAUY |

TAMB is the equilibrium temperature and is equal to 89.1°F for this case (The details of calculation of equilibrium temperature are given Appendix (A).

$$AKT = \frac{KsH}{Bz}$$

Where AKT is non-dimensional surface heat transfer coefficient.

Ks is the surface heat transfer coefficient and its evaluation is given in Appendix A.

H is the reference depth and is equal to 20 m

$Bz = \rho CpBv$

Where $\rho$ is density, Cp specific heat at constant pressure, Bv is vertical diffusivity = 10 $cm^2/sec$

For the case considered substituting the above values the value of AKT comes out to be 0.349.

TAUX, TAUY are equal to $\frac{\partial u}{\partial \tau}$ and $\frac{\partial v}{\partial \tau}$ non-dimensional in x and y directions respectively.

Wind shear is obtained from the Wilson (1960) curve which is given in Fig. ( 7.5). For a wind speed of 4.25 meters/sec 225°N the Wilson curve gives the shear stress ($\tau w$) equal to 0.3575 dynes/cm$^2$.

$\tau_w x = \pm$ 0.3359 dynes/cm$^2$

$\tau_w y = \pm$ 0.1223 dynes/cm$^2$

The direction (+ or -) is decided as follows:

$\tau_w x$ (or $\tau_w y$) - ve or +ve when wind stress is in the direction of x (or y) or in the opposite direction of x (or y). After finding $\tau_w x$ and $\tau_w y$ they are non-dimensionalizedas shown below to obtain TAUX and TAUY.

$$TAUX = \frac{H}{U_{ref}} \frac{\tau wx}{A_z} = \frac{10 \times 100}{30} \quad \frac{0.3359}{10} = 2.24$$

$$TAUY = \frac{H}{U_{ref}} \frac{\tau wy}{A_z} = \frac{10 \times 100}{30} \quad \frac{0.1223}{10} = 0.82$$

| CARD NOS.. | FORTRAN QUANTITY |
|---|---|
| 13 | MAR(1,1);MAR(2,1):MAR(3,1) |

This is to be selected basing on the domain type and an example how to choose is given in Fig. (7.3).

| CARD NOS. | FORTRAN QUANTITY |
|---|---|
| | MRH (1,1); MRH(2,1): MRH(3,1) |

The way MRH is to be selected is similar to MAR. Fig. (7.4)

| CARD NOS. | FORTRAN QUANTITY |
|---|---|
| | HI(1,1): HI(1,2): HI(1,3) |

HI is the non-dimensional depth and changes from one domain to another This is given in subroutine HITEA.

7.3.2 Calculation of Input Data For Data Element "INDATA5"

(For main programs TMAIN5, TMAIN 5T and TMAIN5V)

The first 12 input data cards are same as for data element "INDATA" which are calculated in the previous section. Now, it will be shown how to calculate the rest of the data.

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 13 | NIN, NOUT |

NIN is the number of vertical inlet nodes and is equal to 5 in this case and are at I=2 and J=1. NOUT is the number of vertical outlet nodes and is equal to 5 in this case and are at I=11 and J=13. (See Fig.7.2)

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 14 | I,J,K,U(I,J,K),V(I,J,K),T(I,J,K) |

(For Inlet)

The discharge or inlet is at I=2, J=1. The discharge is in the y-direction so the velocity in the x direction (v-velocity) is zero and there is only u velocity. The u velocity is calculated as explained below.

The depth at the inlet point is 10.8 meters. There are six grid points in the vertical direction, one being at the surface and one being at the bottom. The velocity at the bottom is zero. The flow depth in the numerical grid system would give ½ of grid spacing in the vertical direction at the top and bottom. The flow depth associated with intermediate

grid points is one full grid spacing in the vertical direction. The flow width at the inlet is equal to grid spacing in the y direction. Assuming equal velocities at the upper five grid points and zero velocity at the bottom the following formula is used to compute the inlet velocity (U).

$$\text{Velocity } (\bar{U}) = \frac{\text{Rate of discharge}}{0.9 \text{ X depth X width}}$$

$$U = \frac{228.0 \text{ X } 10^6 \text{ kg}}{0.9 \text{ X } 10.8 \text{ X } 60 \text{ X } 10^4}$$

$$= 11.9 \text{ cm/sec}$$

Non-dimensionalizing with respect to reference velocity (30 cm/sec) the velocity $U(I,J,K) = \frac{11.9}{30} = 0.373$. The discharge temperature is $38.9^{o}C$. Non-dimensional temperature $T(I,J,K) = \frac{38.9 - 30.0}{30.0} = 0.2967$.

The input data up to card No. 18 would be the same as card 14 except K value will be increasing by 1

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 19 | I,J,K U(I,J,K), V(I,J,K) |

(For outlet)

The fluid leaves the mixing pond at location I=11 and J=13 (See Fig. 7.2) This location is called outlet. The outlet velocity is in the y direction and therefore it has only v-velocity. The depth at outlet is equal to depth at inlet. Thus, the velocities at inlet is taken to be equal to velocities at outlet. ie

U(I,J,K) = 0.373, V(I,J,K) =0

The rest of the cards (up to 23) would be similar to card 19 except K has to be increased by 1 for every card.

7.3.3 Calculation of Input Data For Data Element "INDATA6" (For Main Program TMAIN6)

Same as first 12 lines of "INDATA" or "INDATA5" which are calculated in the previous sections.

7.4 Sample Input (For Far-Field Unstratified Case)

In the previous section the calculation of input parameters for different programs are presented. In this section the calculated numerical values are summarized in order for each Main program.

(Continued)

## 7.4.1 Sample Input for TMAIN4 (INDATA)

```
   20   20
0.0002222,0.4827
1.0,0.008912,63.117,1.0
0.01,100,1.8,1.0
0.03565,0.03565,0.20
0.0
0.00107
1.0,0.008912,63.117
1.000428,-0.000019,-0.0000046
30.0
2160.0,0.0,0.0
31.7,0.349,0.396,0.328
7,2,9,0,0,0,0,0,7,2,9,0,0,0,0,0,7,2,2,2,9,0,7,2,9,0,0,0,0,
3,11,4,0,0,0,0,7,8,11,4,0,0,0,0,0,3,11,11,11,8,2,8,11,4,0,0,0,0,
3,11,4,0,0,7,2,8,11,11,8,2,2,2,9,7,8,11,11,11,11,11,11,11,4,0,0,0,0,
3,11,8,2,2,8,11,11,11,11,11,11,11,11,8,8,11,11,11,11,11,11,11,11,8,2,9,0,0,
3,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,4,0
0,
5,6,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,6,6,11,11,11,11,11,11,8,2,9,
0,5,6,11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,4,5,1,1,6,11,11,11,11,11,4,
0,0,5,1,6,11,6,1,6,11,11,11,11,11,11,11,11,11,4,0,0,7,8,11,11,11,11,11,4,
0,0,0,0,5,1,10,0,3,11,11,11,11,11,11,11,11,6,10,0,0,3,11,11,11,11,11,11,4,
0,0,0,0,0,0,0,0,3,11,11,11,11,11,11,11,11,4,0,0,0,3,11,6,6,11,11,11,4,
0,0,0,0,0,0,0,0,3,11,11,6,6,11,11,6,1,10,0,0,0,5,1,10,5,1,6,11,4,
0,0,0,0,0,0,0,0,5,6,11,4,5,1,1,10,0,0,0,0,0,0,0,0,0,5,1,10,
0,0,0,0,0,0,0,0,0,5,1,10,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
4,6,0,0,0,0,0,0,4,6,0,0,0,0,0,0,4,2,2,6,0,0,4,6,0,0,0,0,
10,8,0,0,0,0,0,0,4,9,8,0,0,0,0,0,0,10,9,9,9,2,2,9,8,0,0,0,0,
10,8,0,0,0,4,2,9,9,9,2,2,2,6,0,4,9,9,9,9,9,9,9,8,0,0,0,0,
10,9,2,2,2,9,9,9,9,9,9,9,9,9,2,9,9,9,9,9,9,9,9,9,2,6,0,0,
3,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,1,9,9,9,9,9,9,8,0,0,
0,3,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,8,0,3,1,1,9,9,9,9,2,6,
0,0,3,1,9,9,1,1,9,9,9,9,9,9,9,9,9,8,0,0,0,0,10,9,9,9,9,8,
0,0,0,0,3,7,0,0,10,9,9,9,9,9,9,9,9,7,0,0,0,4,9,9,9,9,9,8,
0,0,0,0,0,0,0,0,10,9,9,9,9,9,9,9,8,0,0,0,0,10,9,1,9,9,9,8,
0,0,0,0,0,0,0,0,10,9,9,9,1,9,9,9,9,1,7,0,0,0,0,3,7,0,3,1,9,8,
0,0,0,0,0,0,0,0,3,9,8,0,3,1,7,0,0,0,0,0,0,0,0,0,0,0,3,7,
0,0,0,0,0,0,0,0,0,3,7,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
1,0.27,0.27,0.35,0.35,0.35,0.35,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0
2,0.53,0.53,0.69,0.69,0.69,0.35,0.35,1.0,1.0,1.0,1.0,1.0,1.0,1.0
3,0.27,0.27,0.35,0.35,0.69,0.69,0.35,0.35,1.0,1.0,1.0,1.0,1.0
4,1.0,1.0,1.0,0.35,0.69,0.69,0.69,0.35,1.0,1.0,1.0,1.0,1.0
5,1.0,1.0,1.0,0.35,0.69,0.69,0.69,0.35,0.35,1.0,1.0,1.0,1.0
6,1.0,1.0,0.21,0.25,0.69,0.69,0.69,0.69,0.35,1.0,1.0,1.0,1.0
7,1.0,1.0,0.16,0.30,0.69,0.69,0.69,0.35,0.35,1.0,1.0,1.0,1.0
8,1.0,0.16,0.16,0.46,0.69,0.69,0.69,0.35,1.0,1.0,1.0,1.0,1.0
9,0.16,0.16,0.38,0.61,0.69,0.69,0.61,0.32,0.23,0.12,0.15,0.21,1.0
10,0.32,0.61,0.69,0.69,0.69,0.69,0.69,0.61,0.46,0.23,0.30,0.27,0.27
11,0.32,0.32,0.35,0.69,0.69,0.12,0.12,0.38,0.69,0.69,0.61,0.53,0.53
12,1.0,1.0,0.35,0.69,0.69,0.12,0.12,0.38,0.23,0.15,0.12,0.27,0.27
13,1.0,1.0,0.35,0.69,0.69,0.38,0.38,0.53,0.69,0.50,0.23,0.23,1.0
14,1.0,1.0,0.35,0.69,0.38,0.38,0.69,0.69,0.69,0.61,0.46,0.23,1.0
15,1.0,1.0,0.30,0.25,0.30,0.53,0.58,0.69,0.69,0.66,0.30,0.15,1.0
16,1.0,1.0,0.20,0.14,0.15,0.30,0.46,0.69,0.69,0.69,0.25,0.20,1.0
17,0.17,0.17,0.24,0.41,0.46,0.46,0.46,0.69,0.69,0.61,0.32,1.0,1.0
18,0.18,0.35,0.53,0.69,0.69,0.69,0.69,0.69,0.35,0.32,0.32,1.0,1.0
```

19,0.25,0.50,0.60,0.69,0.69,0.35,0.35,0.35,0.35,1.0,1.0,1.0,1.0
20,0.19,0.38,0.69,0.69,0.35,0.15,0.15,1.0,1.0,1.0,1.0,1.0,1.0
21,0.19,0.23,0.53,0.69,0.53,0.23,0.15,1.0,1.0,1.0,1.0,1.0,1.0
22,1.0,0.23,0.46,0.61,0.69,0.47,0.23,0.19,0.19,0.19,0.19,1.0,1.0
23,0.16,0.16,0.16,0.47,0.69,0.58,0.26,0.23,0.38,0.38,0.19,1.0,1.0
24,0.16,0.32,0.53,0.66,0.61,0.69,0.46,0.53,0.47,0.19,0.19,1.0,1.0
25,0.16,0.16,0.27,0.28,0.46,0.61,0.69,0.64,0.58,0.27,0.27,1.0,1.0
26,1.0,1.0,1.0,0.21,0.41,0.50,0.61,0.69,0.69,0.53,0.27,1.0,1.0
27,1.0,1.0,1.0,0.21,0.21,0.25,0.50,0.61,0.69,0.53,0.27,0.27,1.0
28,1.0,1.0,1.0,1.0,1.0,0.20,0.38,0.53,0.53,0.53,0.38,0.27,1.0
29,1.0,1.0,1.0,1.0,1.0,0.27,0.27,0.27,0.27,0.27,0.27,0.27,1.0

## 7.4.2 Sample Input for Main Programs TMAIN 5, TMAIN 5T and TMAIN 5V (IN DATA 5)

```
 1          20   20
 2       0.0002222,0.4827
 3       1.0,0.008912,63.117,1.0
 4       0.01,100,1.8,1.0
 5       0.03565,0.03565,0.20
 6       0.0
 7       0.00107
 8       1.0,0.008912,63.117
 9       1.000428,-0.0000019,-0.0000046
10       30.0
11       2180.0,0.0,0.0
12       31.7,0.349,0.896,0.328
13       5,5
14       2,1,1,0.0,0.373,0.2967
15       2,1,2,0.0,0.373,0.2967
16       2,1,3,0.0,0.373,0.2967
17       2,1,4,0.0,0.373,0.2967
18       2,1,5,0.0,0.373,0.2967
19       11,13,1,0.0,0.373
20       11,13,2,0.0,0.373
21       11,13,3,0.0,0.373
22       11,13,4,0.0,0.373
23       11,13,5,0.0,0.373
```

## 7.4.3 Sample Input for TMAIN 6 (IN DATA 6)

```
1          20   20
2       0.0002222,0.4827
3       1.0,0.008912,63.117,1.0
4       0.01,100,1.8,1.0
5       0.03565,0.03565,0.20
6       0.0
7       0.00187
8       1.0,0.008912,63.117
9       1.000428,-0.000019,-0.0000046
10      30.0
11      2180.0,0.0,0.0
12      31.7,0.349,0.896,0.328
```

## 7.5 PROGRAM EXECUTION PROCEDURE:

In order to execute the programs for the far field model, the following steps have to be followed.

1) Input Parameters: The calculation of input parameters is explained in the sample problem section (7.3). The input parameters depend on the discharge conditions, ambient conditions and the reference quantities chosen.

2) First Run: In order to obtain three dimensional velocities and temperatures, the main programs that have to be executed are TMAIN 4, TMAIN 4T, TMAIN 5 and TMAIN 6. The flow chart is shown in Figure (7.6a). The main program TMAIN 4 initializes velocities and temperatures ie it sets, all velocities equal to zero and temperature equal to the reference temperature. TMAIN 4T reads IR data as the initial conditons for temperature. If IR data is not available, the main program TMAIN4T should not be executed. TMAIN5 does computations and TMAIN 6 prints the results. In the programs, there are two units. One is read unit, designated as unit 7 and the other is store unit, designated as unit 8. Two tapes have to be provided, one for the read unit (unit 7) and another for the store unit (unit 8).

3) Continuation of a Run: For extending the previous results for more time, the run has to be continued. Now the programs that need to be executed are TMAIN 5 and TMAIN 6. Two tapes have to be provided for the continuation of a run also.

The following are a set of a control cards that were used on UNIVAC 1106 computer in order to run the programs for the first time for a far-field unstratified receiving basin. The explanation for the control cards is given in

the brackets.

CARD 1

@ RUN

(Schedule a new run for initiation)

CARD 2

@ASG,A SKM*DULL

(All parameters on @ ASG Control Statement are optional except file name.

A-specifies that the file being assigned is currently catalogued. SKM is the qualifier and DULL is file name.)

CARD 3

@PACK SKM * DULL

(Packs the non-deleted elements of a program file, by rewriting the file and eliminating the deleted elements)

CARD 4

@ PREP SKM * DULL

(Prepares an entry point table for program file, for use by the @ MAP processor in searching a LIB specified program file to satisfy undefined symbols)

CARD 5

@ ASG,T 8., 16N, LAKE 1

(T-specifies that the file to be assigned temporary and allows it to have a name the same as that of an unassigned catalogued

file. LAKE 1 is name of the tape)

CARD 6

@ MAP

(Call the MAP processor (the collector) to collect a specified set of relocatable elements, and produce from this and executable program which is in an absolute element format)

CARD 7

IN SKM * DULL TMAIN 4

(TMAIN 1 is the main program which would be executed)

CARD 8

LIB SKM * DULL

(Specifies file as a library to be searched)

CARD 9

@ XQT

(Initiates the execution of a program which is in an absolute element format)

CARD 10

@ ADD SKM * DULL INDATA

(Note: INDATA is the data element that provides the input data for the TMAIN 4 main program. The calculation of INDATA is given in the sample problem)

CARD 11

@ ASG,T 7., 16N, LAKE 1

CARD 12

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

(Note: LAKE 2 is the name of the tape)

CARD 13

@ MAP

CARD 14

IN SKM * DULL TMAIN 4T

CARD 15

LIB SKM*DULL

CARD 16

@ XQT

CARD 17

@ ADD SKM & DULL ITPK1

(Note: ITPK1 is the data element that provides the input data for the TMAIN 4T main program. The calculation of ITPK1 is given in the sample problem)

CARD 18

@ ASG,T 7., 16N, LAKE 2

CARD 19

@ ASG,T 8., 16N, LAKE1

CARD 20

@ MAP

CARD 21

IN SKM *DULL TMAIN5

CARD 22

LIB SKM * DULL

CARD 23

@ XQT

CARD 24

@ ADD SKM * DULL INDATA5

(Note: INDATA 5 is the data element that provides the input data for the TMAIN 5 main program. The calculation of INDATA5 is given in the sample problem)

CARD 25

@ ASG,T 7., 16N. LAKE1

CARD 26

@ MAP

CARD 27

IN SKM * DULL TMAIN6

CARD 28

LIB SKM * DULL

CARD 29

@ XQT

CARD 30

@ ADD SKM * DULL INDATA 6

(Note: INDATA6 is the data element that provides the input data for the TMAIN 6 main program. The calculation of INDATA6 is given in the sample problem)

## 7.6 Sample Output

The sample output for far field is similar to the near field, which is given in section 6.6.

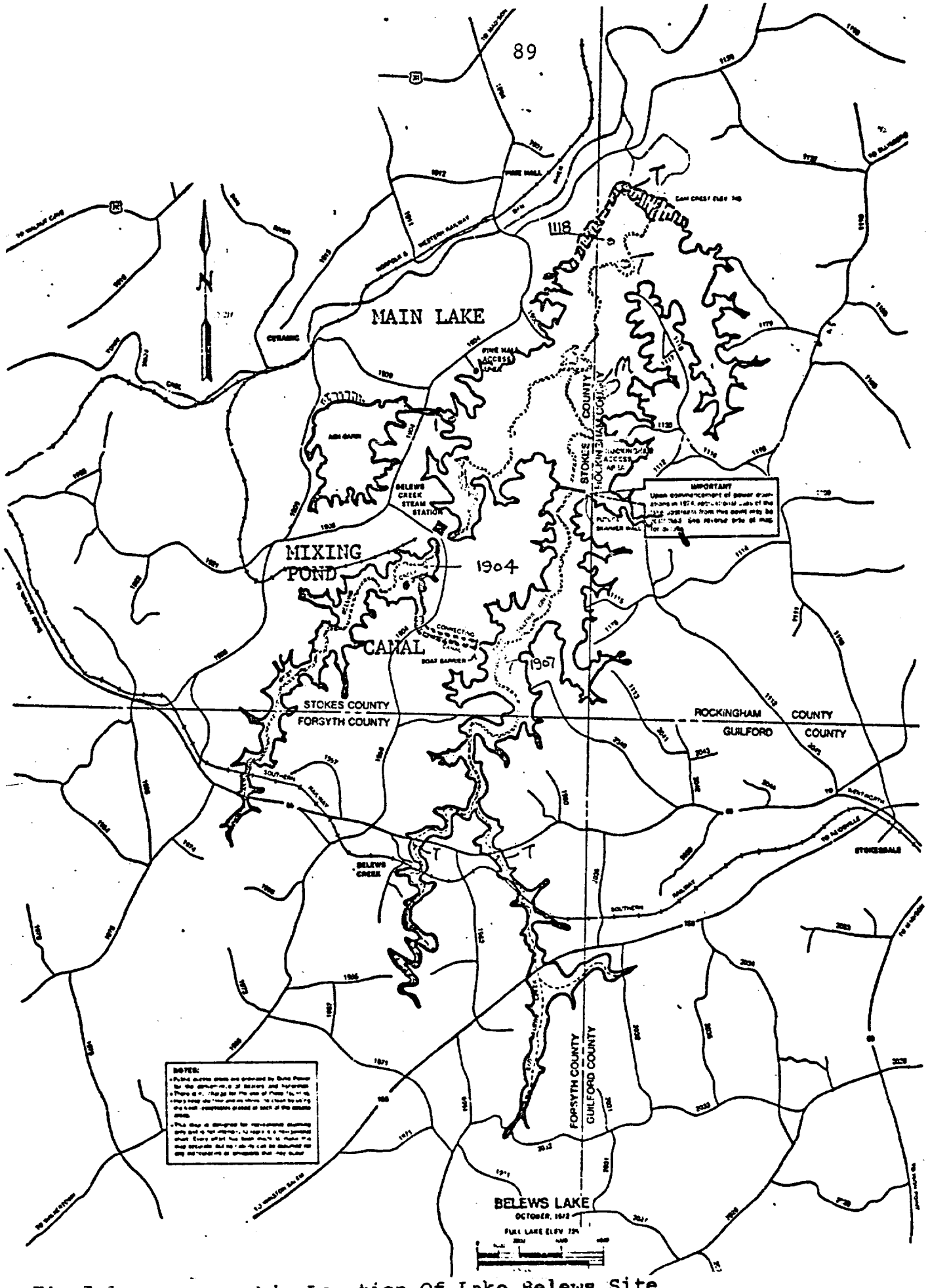



Fig 7.1 Geographic Location Of Lake Belews Site

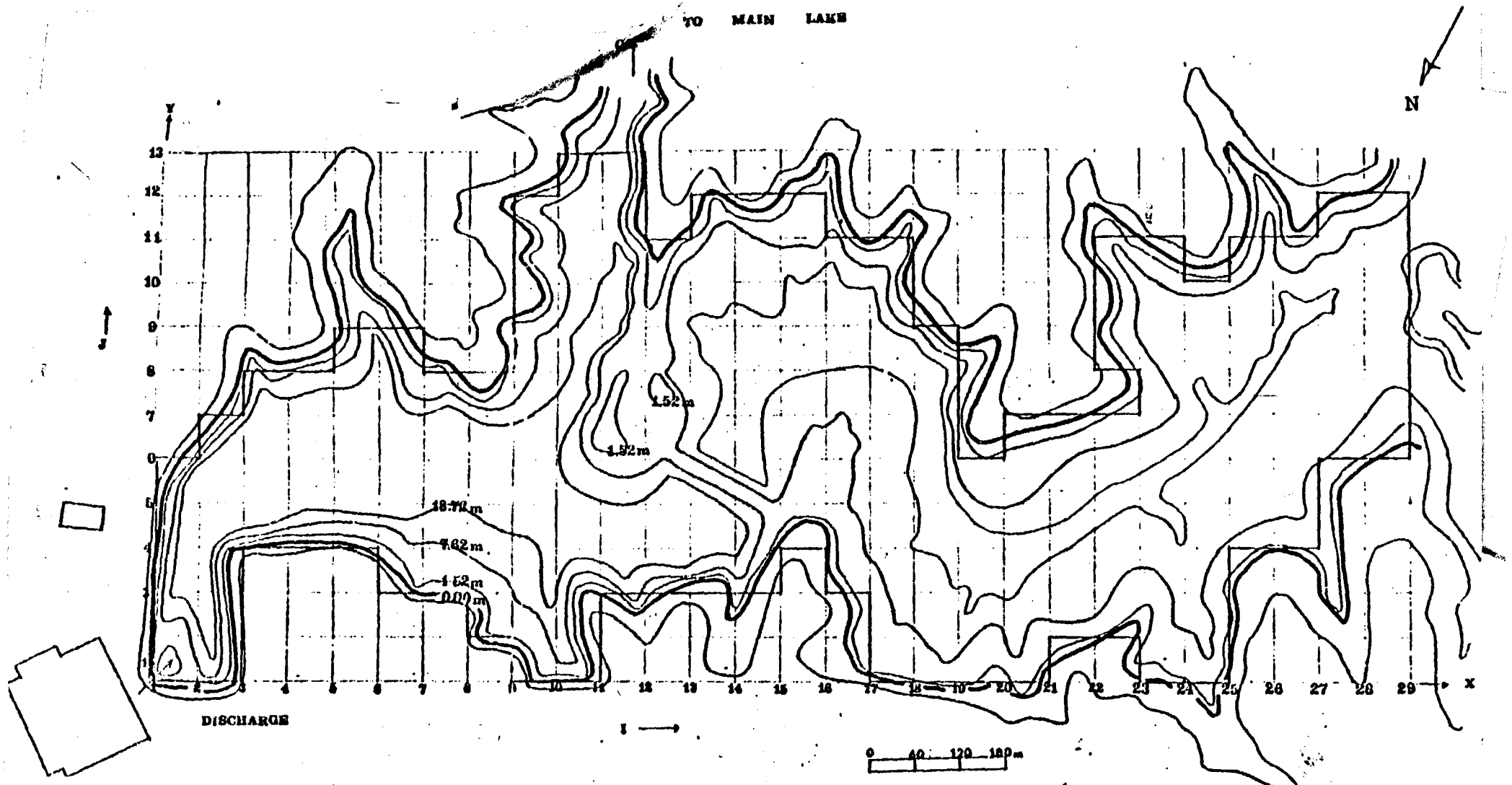



Fig. 7.2 Computational grid for mixing pond at Lake Belews site.

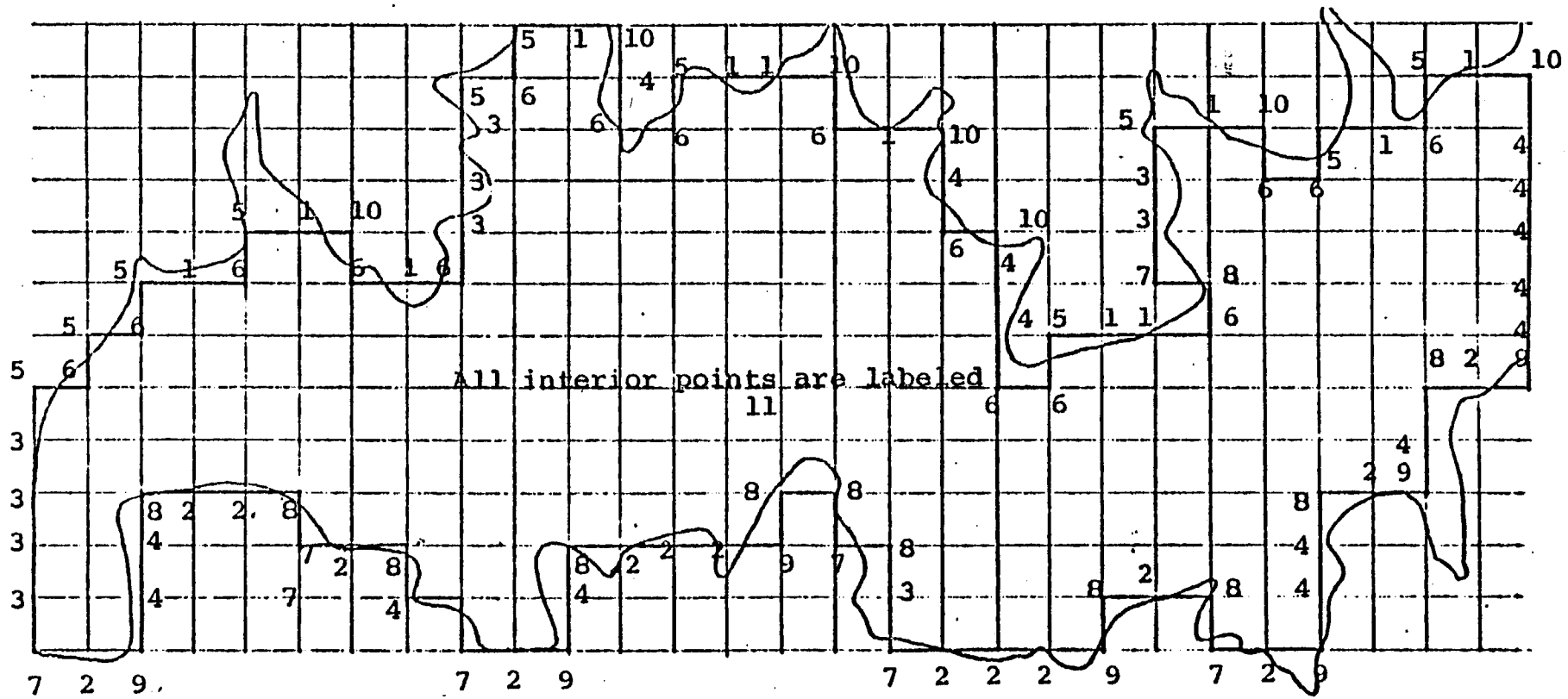



Fig. 7.3 Identifying numbers in the Main Grid System, the MAR matrix for the Mixing Pond

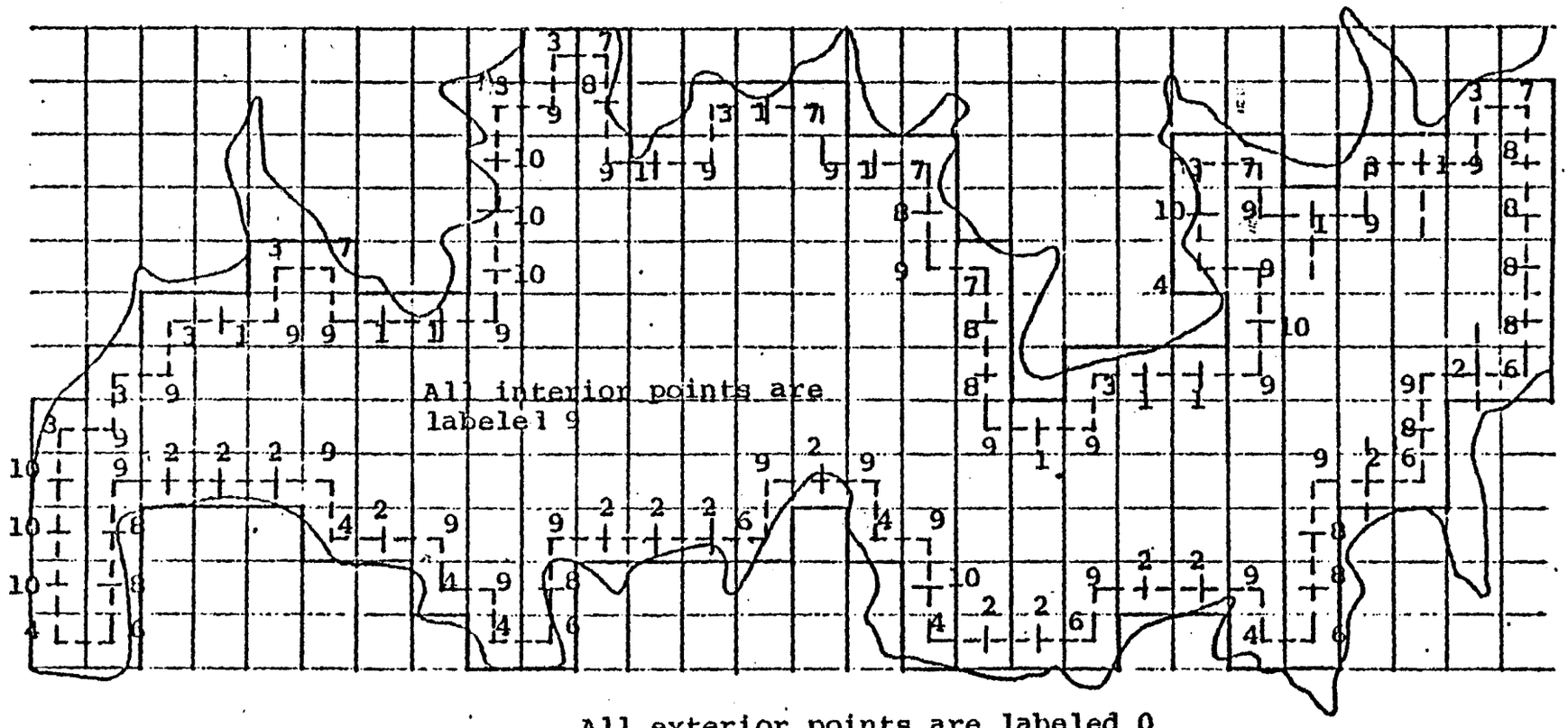



Fig.7.4 **Identifying numbers in the Half Grid System**
The MRH matrix for the Mixing Pond

(a) $\tau = 3.2 \times 10^{-6} v^{2}$, Ekman (1905)
(b) $\tau = 9.3 \times 10^{-6} v^{1.8}$, Hallstrom (1941)
(d) $\tau = 1.21 \times 10^{-6} v^{2} + 2.25 \times 10^{-6} (v-5.6)^{2}$, Van Dorn (1953)
(e) $\tau = 1.98 \times 10^{-6} v^{2}$, Wilson (1960)
(f) $\tau = 0.79 \times 10^{-6} v^{2}$, Wilson/2.5

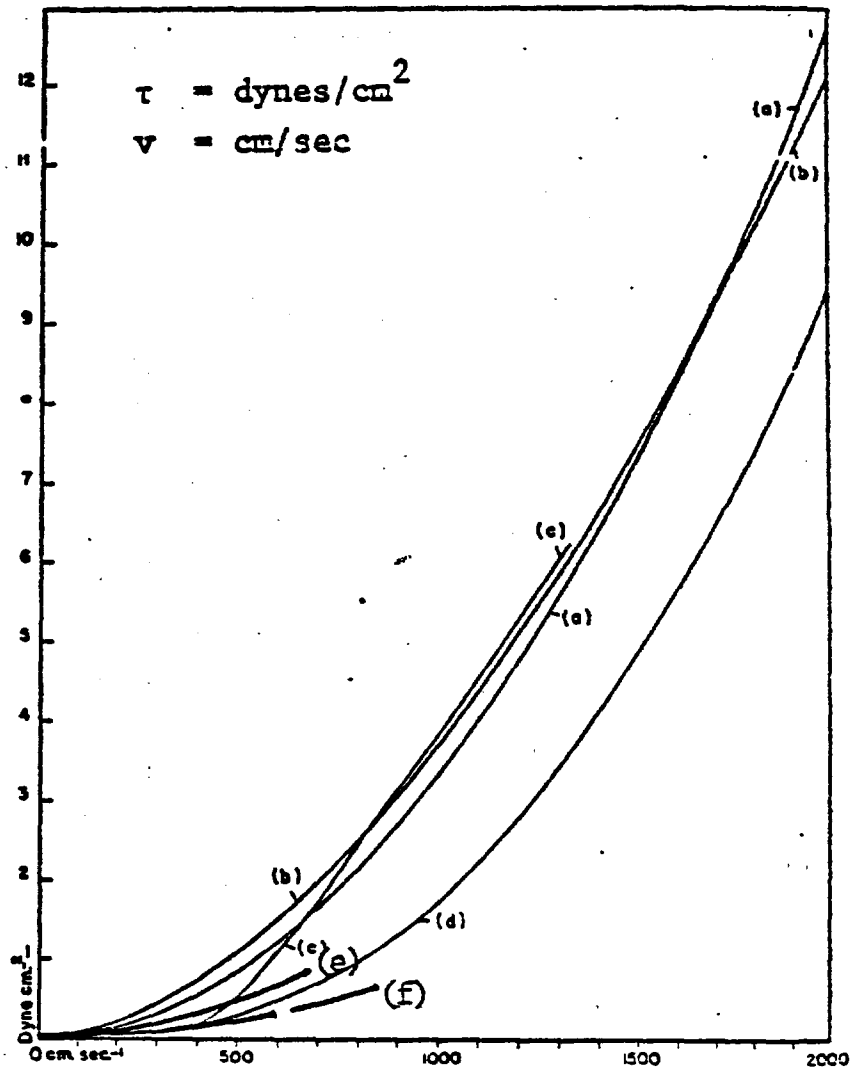



FIGURE 77. Relation of wind stress $\tau_s$ to wind velocity W, based on (a) equation 50, (b) equation 51, (c) Saur's computions after Munk and Anderson, (d) equation 56.

Fig. 7.5 Comparison of the used wind speed versus surface shear stress relationship with the various suggested relationships. (Reproduction from "A treatise on Limnology" by George Evelyn Hutchinson, 1957)

▲ A = Main Lake, horizontal

▲ B = Mixing Pond, horizontal

▲ C = Vertical, Main Lake and Mixing Pond

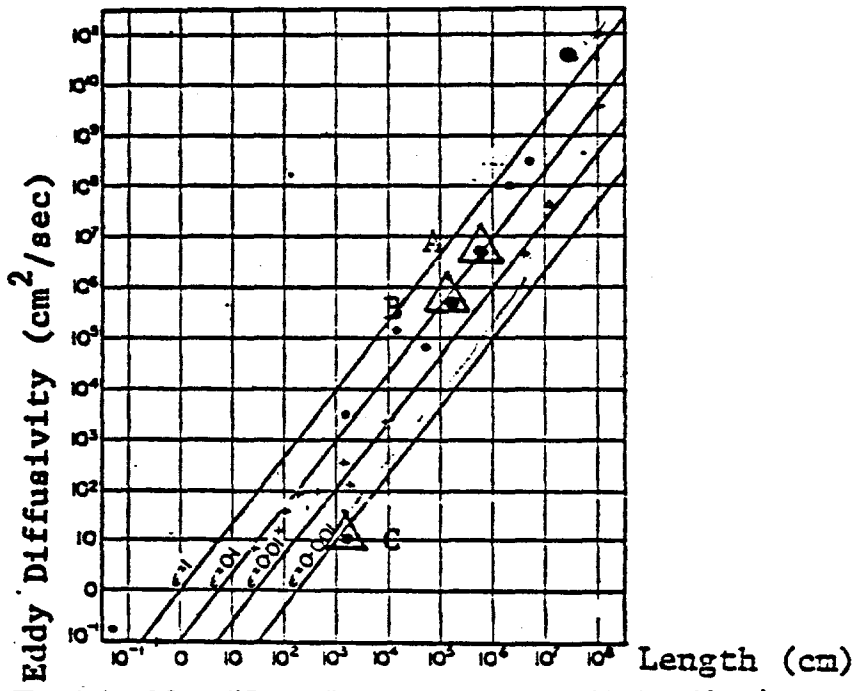



FIG. 167*a*. The relation $F(l) = el^{4/3}$ according to observations (logarithmic scale): points, values of Richardson from the atmosphere; crosses, values of Stommel (Blaimore, Bermuda and Woods Hole); triangles, values of Hanzawa.

Fig. 7.6 Comparison of the used turbulent eddy diffusivities with the observed values. (Reproduction from "Physical Oceanography" by Albert Defant, 1961)

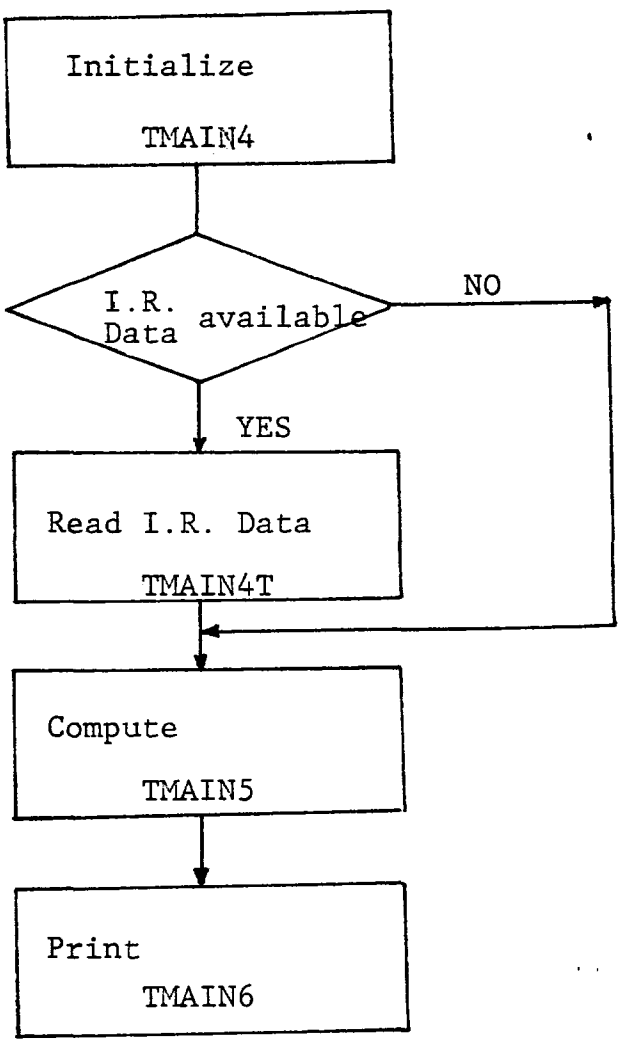



Fig. 7.6a Flow Chart for Program Execution

# SOME REPRESENTATIVE RESULTS FOR THE FAR FIELD MODEL
## (UNSTRATIFIED CASE)

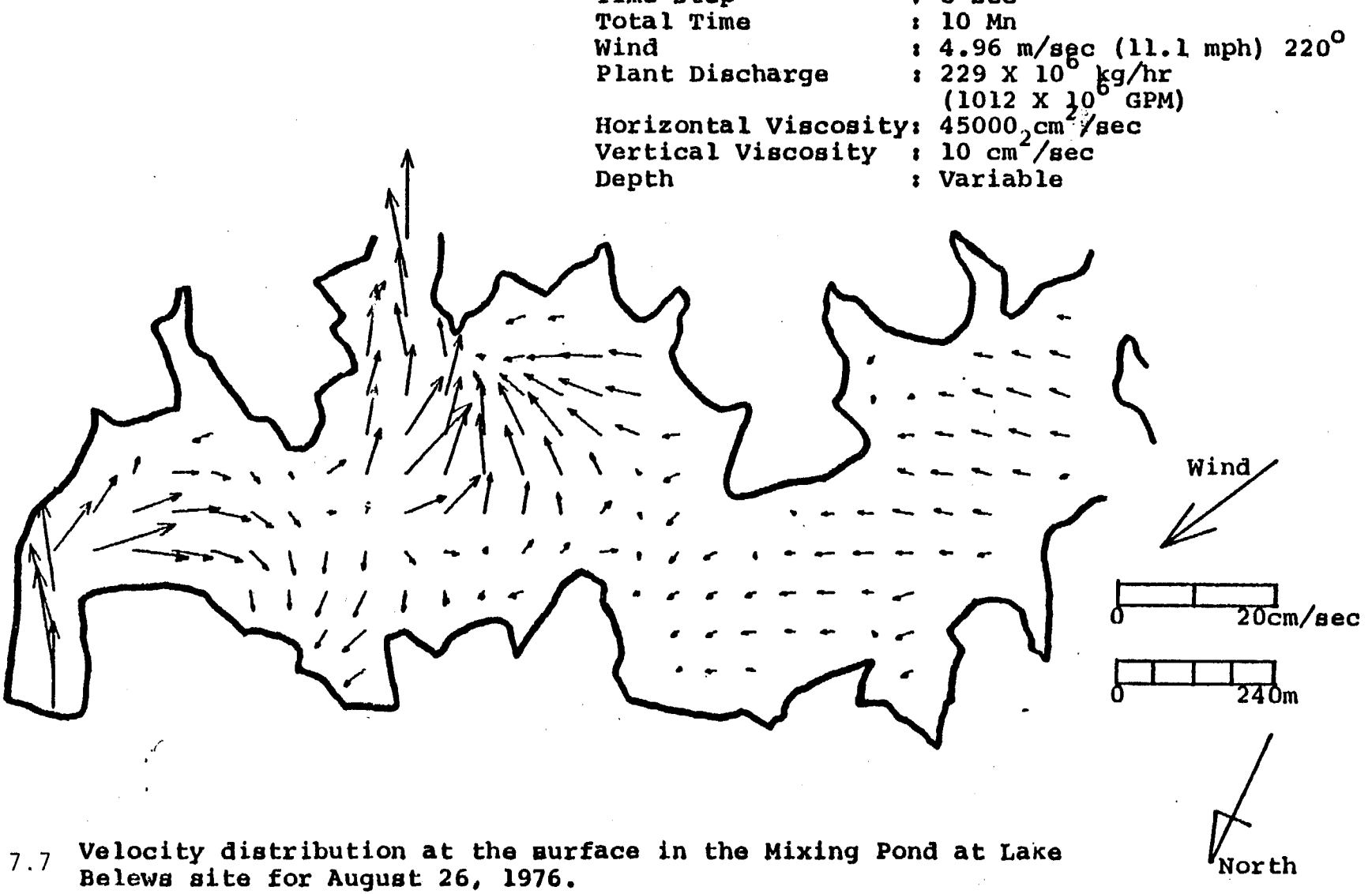



Fig. 7.7 **Velocity distribution at the surface in the Mixing Pond at Lake Belews site for August 26, 1976.**

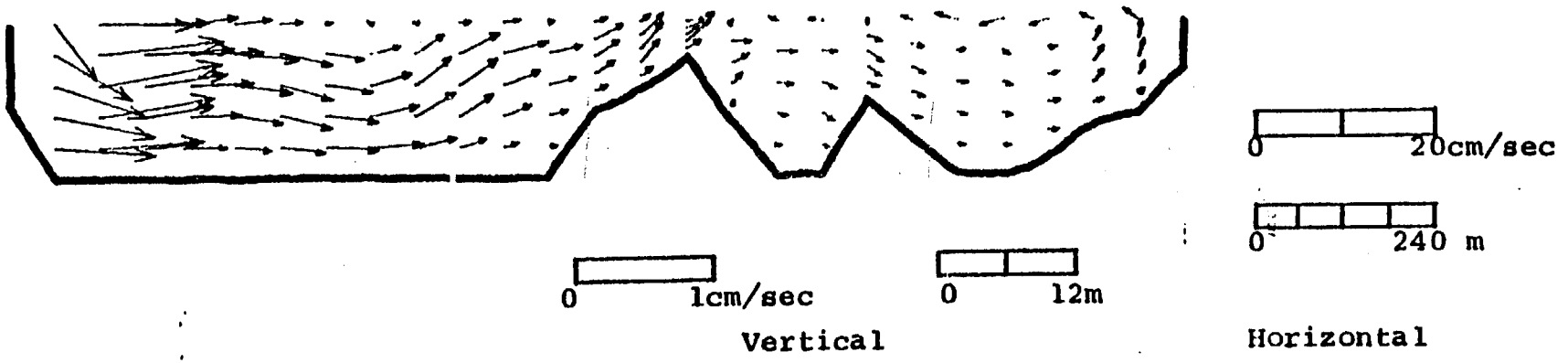



Fig. 7.8 Velocity distribution at vertical section along J=5 in the Mixing Pond at Lake Belews site for August 26, 1976.

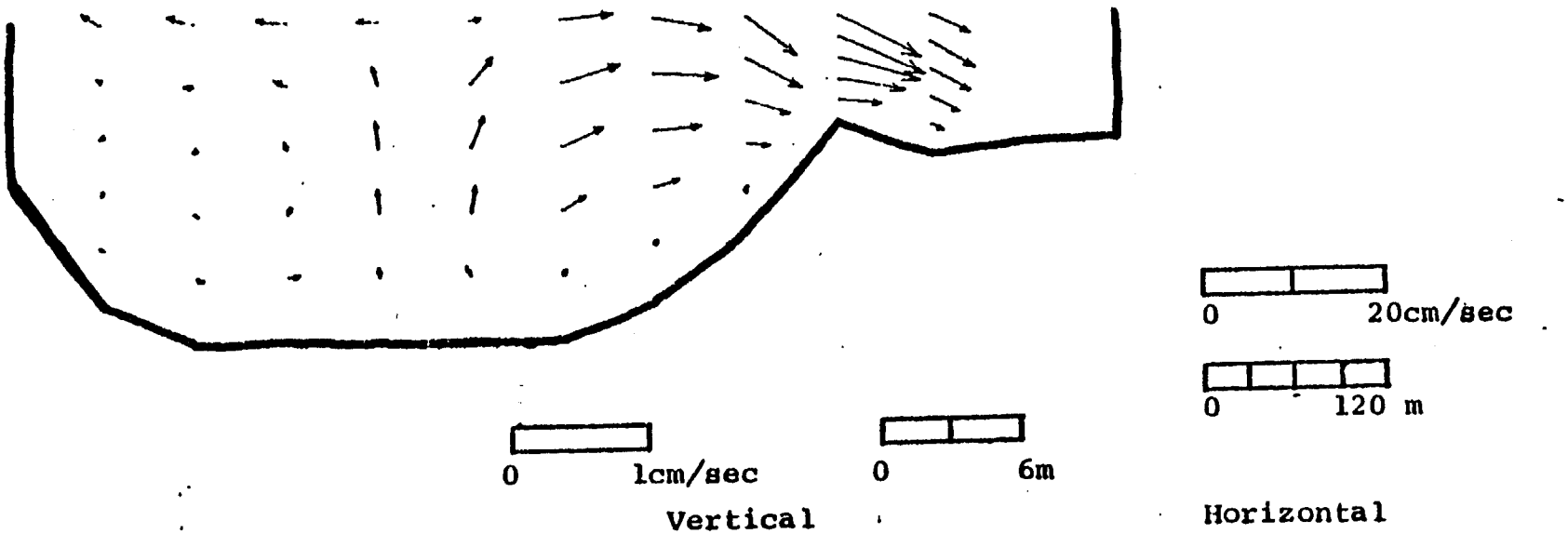



Fig. 7.9 Velocity distribution at vertical section along I=10 in the Mixing Pond at Lake Belews site for August 26, 1976.

Discharge Temp: 38.9 - 39.1°C
Air Temp- : 23.9 - 28.9°C
Wind : 3.7 - 5.0 m/sec
(8.3 - 1.2 mph) SSW

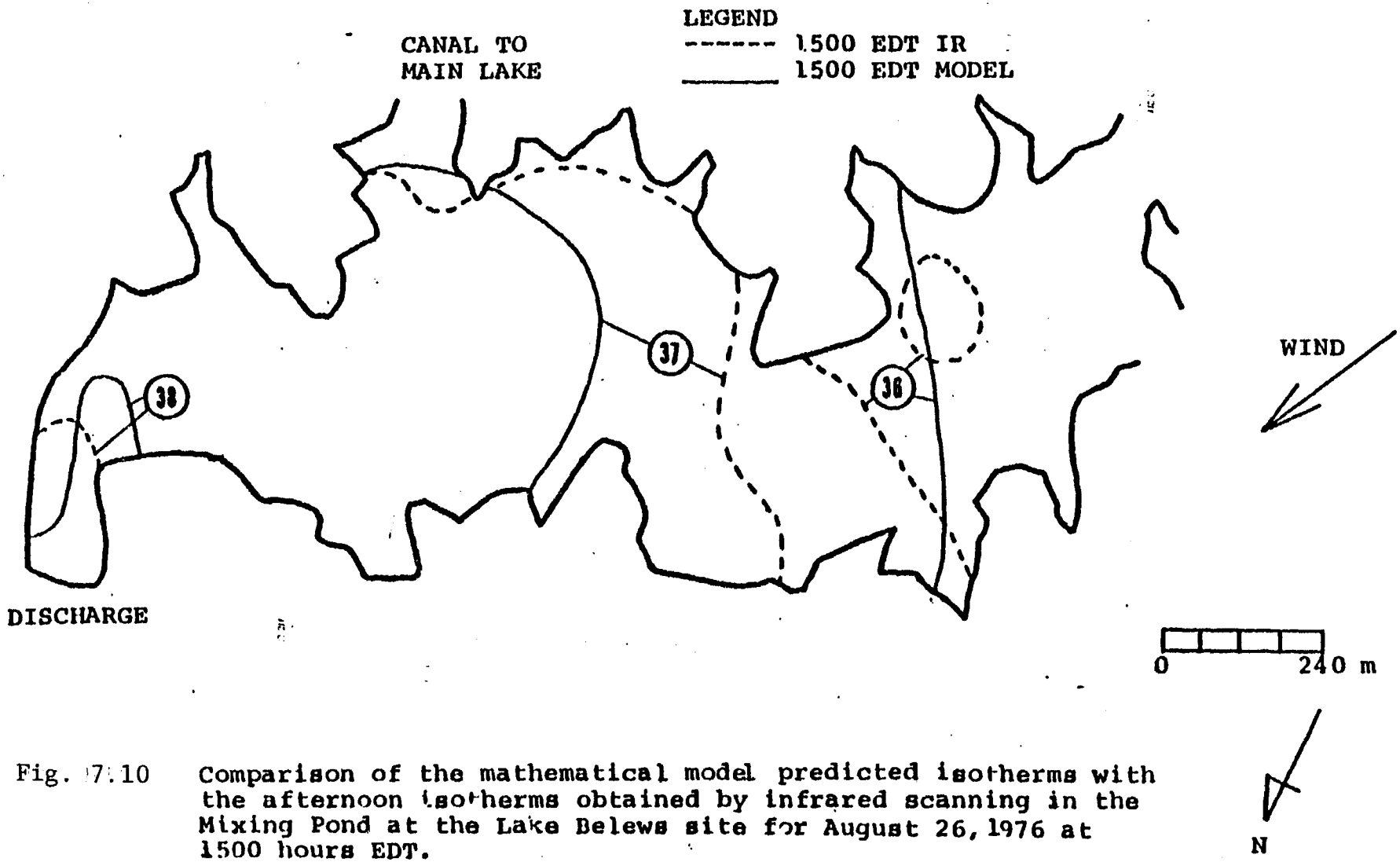



Fig. 7.10 Comparison of the mathematical model predicted isotherms with the afternoon isotherms obtained by infrared scanning in the Mixing Pond at the Lake Belews site for August 26, 1976 at 1500 hours EDT.

## 8.0 Sample Case, Far-Field (Stratified Case, Lake)

The sample case for far-field stratified and unstratified cases are very much similar. In this section the extra information that is needed for the stratified case will be shown

8.1 Problem Statement: This is the same as that for an unstratified pond case, except here it is required to find velocity and temperature distributions in a stratified lake.

8.2 Choice of Programs: Since this is a stratified case, the programs to be used are (1) TMAIN 4B, (2) TMAIN5B, TMAIN5TB, TMAIN5VB and (3) TMAIN 6B. The data elements that go with these main programs are (1) DATAML (2) DAML5 and (3) DATAML6. If the initial conditions are known (ie ground truth data and IR data), then the program TMAIN4TB can be used. The data element that goes with it is ITLK1. The authors in this case had initial conditions for August 26, 1977 (Mathavan, 1977) and, therefore, they used TMAIN 4TB and ITLK1.If the initial conditions are not known, zero velocity and and ambient temperature can be used as initial conditions. In this case, the program TMAIN 4TB and the data element ITLK1 that goes with it can be thrown away and yet velocity and temperature distributions can be obtained. The flow chart is shown in Fig.s (9.2 to 9.4)

8.3 Calculation of Input Parameters: In this section, the specification of the grid system, and the reference quantities chosen, will be presented first followed by the actual calculation of input parameters. Only those quantities that did

not appear in the unstratified section and are necessary for stratified case, will be presented.

(1) Grid System: The computational domain of 2880 meters by 6720 meters in the horizontal plane as shown in Fig. (8.1) covers most of the lake. A grid size of 240 meters by 240 meters gives a 23 X 13 mesh in the horizontal plane. The depth is divided into five layers, thus giving a 29 X 13 X6 mesh for the entire computational domain.

(2) Reference Quantities: Reference quantities are used to non-dimensionalize the input parameters and they are given below:

Reference velocity : 30 cm/sec

Reference temperature: $30^{o}C$

Reference horiztonal length: 6732 meters

Reference dpeth = Maximum depth = 40 meters

Reference Horizontal eddy viscosity = $4.5 \times 10^{6}$ $cm^2/sec$

Reference vertical eddy viscosity = 10 $cm^2/sec$

Pr = 1 is chosen

Reference horizontal eddy diffusivity = $4.5 \times 10^{6}$ $cm^2/sec$

Reference vertical eddy diffusivity = 10 $cm^2/sec$

8.3.1 Input Data for Data Element "DATAML" (For main program TMAIN 4B)

The calculation of the first 12 input data cards is similar to "INDATA" which was explained in section 7.3 of the previous section. Here the rest of the input data will be explained.

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 28 | MAR(1,1),MAR(2,1),MAR(3,1) |
| 35 | ------ |

This is to be selected based on the domain type and an example how to make the choice is given in Fig.(7.4). This data can be seen in the input data in section 8.4).

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 36 | MRH(1,1), MRH(2,1) |
| . | |
| . | |
| . | |
| 52 | ------ |

The procedure for dtermining MRH is similar to MAR and can be determined very easily as is explained in Fig. (7.5) This data can be seen in the input data in section (8.4).

CONTINUED

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 53 | HI(1,1), HI(1,2)... |
| . | HI(2,1), HI92,2)... |
| . | |
| . | ------ |
| . | ------ |
| 81 | ------ |

This set of input data cards represents the non-dimensional depth matrix. The depth data for Lake Belews site is obtained from geological survey maps and they are non-dimensionalized using a reference depth of 40 meters. This data can be seen in the input data in section (8.4)

8.3.2 Calculation of Input Data for Data Element "DATAML5"

(For main programs TMAIN5B, TMAIN5TB and TMAIN5VB)

The first 13 cards are the same as that of "DATAML" which are explained in the section 8.3.1. Now the rest of the input data will be explained.

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 14 | NIN,NOUT |

Number of Inlet nodes (NIN) in this case = 3

Number outlet nodes (NOUT) in this case = 3

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 15 | I,J,K V(I,J,K), V(I,J,K),T( I,J,K) |
| 16 | |
| 17 | ------ |

This set of input data cards are for the inlet into the lake. The discharge is introduced at three grid points identified by I=27, J=3 and K= 1,2 and 3. The discharge velocity is calculated uisng the following quantities.

Plant discharge = $228.9 \times 10^6$kg

Discharge depth = 8 meters

Discharge width = 240 meters

The discharge is in the x-direction (ie v-velocity only, u-velocity is zero)

The v-velocity is calculated as follows:

$$V_i = \frac{228 \times 10^6 \times 10^3}{.2 \times 3600 \times 2400 \times 800 \times 0.9926}$$

$$= 3.336 \text{ cm/sec}$$

This velocity is non-dimensionalized with reference velocity which is equal to 30 cm/sec.

$$v(I,J,K) = \frac{3.336}{30} = 0.111$$

The discharge temperature = $39.0^{\circ}C$

$$\text{Non-dimensional temperature} = \frac{39 - 30}{30} = 0.1833$$

T (I,J,K) = 0.1833

| CARD NO. | FORTRAN QUANTITY |
|---|---|
| 18 | I,J,K U(I,J,K), V(I,J,K) |
| 19 | ------ |
| 20 | ------ |

This set of cards are for outlet velocity and outlet velocity is in the y-direction. The outlet velocity is equal to the inlet velocity.

$U_o$ = 3.336 cm/sec

which after non-dimensionalization gives

$$U(I,J,K) = \frac{Uo}{Uref} = \frac{3.336}{30} = 0.111$$

$V(I,J,K) = U(I,J,K)$ [ie. $V_1(27,3,K) = U_o(19,2,K)$]

The outlet velocity is at I = 19, J=2, K = 1,2 and 3.

8.3.3 Calculation of Input Data for "DATAML6" (for main programs TMAIN6)

Same as first 13 lines of "DATAML5"

8.4 Sample Input (For Far Field Stratified Case)

In the previous section the calculation of Input parameters for different programs is presented. In this section the calculated numerical values are summarized in order for each main program.

## 8.4.1 Sample Input for Main Program TMAIN4B (DATAML)

```
   20   20
0.0002222,1.9309
1.0,0.002223,63.114,1.0
0.01,100,1.8,1.0
0.03565,0.03565,0.2
0.0
0.0002673
1.0,0.002228,63.114
1.000428,-0.000019,-0.0000046
30.0
4360.0,0.0,0.0
31.7,0.698,1.37,-1.32
250.0,0.000667,0.00000667
0.0000,29.4
0.0075,29.4
0.05,29.4
0.10,29.4
0.15,29.4
0.20,28.7
0.25,26.9
0.30,17.2
0.35,13.7
0.40,12.2
0.45,11.6
0.50,11.4
0.55,11.0
0.60,11.0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,7,2,2,2,9,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,7,2,2,2,8,11,11,11,4,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,7,2,2,8,11,11,11,11,6,1,10,0,0,0,0,0,7,2,2,9
0,0,0,0,0,0,0,0,0,3,11,11,11,11,11,6,10,0,0,0,0,0,0,7,2,8,11,11,4
0,0,0,0,0,0,0,0,7,8,11,11,11,11,6,10,0,0,0,0,0,0,7,8,11,11,11,11,4
0,0,0,0,0,0,0,0,3,11,11,11,6,1,10,0,0,0,0,0,0,0,3,11,11,6,1,1,10
0,0,0,0,0,0,7,2,8,11,11,6,10,7,2,2,2,2,2,2,2,8,11,6,10,0,0,0
0,0,0,7,2,2,8,11,11,11,11,8,2,8,11,11,11,11,11,11,11,11,11,4,0,0,0,0
0,7,2,8,11,11,11,11,11,11,11,11,11,11,11,11,6,1,1,1,1,1,1,10,0,0,0,0
7,8,11,11,11,11,11,11,11,11,11,11,11,11,11,11,4,0,0,0,0,0,0,0,0,0,0,0
3,11,11,11,11,11,11,11,11,11,11,6,1,1,1,1,10,0,0,0,0,0,0,0,0,0,0,0
3,11,11,11,6,1,1,6,11,11,6,10,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
5,1,1,1,10,0,0,5,1,1,10,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,4,2,2,6,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,4,2,2,9,9,1,7,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,4,2,9,9,9,9,7,0,0,0,0,0,0,0,0,4,2,6
0,0,0,0,0,0,0,0,10,9,9,9,9,7,0,0,0,0,0,0,0,0,4,2,9,9,8
0,0,0,0,0,0,0,0,4,9,9,9,1,7,0,0,0,0,0,0,0,0,4,9,9,1,1,7
0,0,0,0,0,0,0,0,10,9,9,7,0,0,0,0,0,0,0,0,0,0,10,9,7,0,0,0
0,0,0,0,0,0,4,2,9,9,0,0,0,4,2,2,2,2,2,2,2,9,8,0,0,0,0
0,0,0,4,2,2,9,9,9,9,9,2,2,9,9,9,1,1,1,1,1,1,7,0,0,0,0
0,4,2,9,9,9,9,9,9,9,9,9,9,9,8,0,0,0,0,0,0,0,0,0,0,0
4,9,9,9,9,9,9,9,9,9,9,1,1,1,1,7,0,0,0,0,0,0,0,0,0,0,0
10,9,9,9,1,1,1,9,9,9,7,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3,1,1,7,0,0,0,3,1,7,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1,0.,0.,0.,0.,0.,0.,0.,0.,0.,0.36,0.36,0.47,0.50
2,0.,0.,0.,0.,0.,0.,0.,0.,0.34,0.36,0.72,0.95,0.50
3,0.,0.,0.,0.,0.,0.,0.,0.,0.34,0.69,0.95,0.91,0.50
4,0.,0.,0.,0.,0.,0.,0.,0.42,0.40,0.80,0.95,0.91,0.50
```

```
5,0.,0.,0.,0.,0.,0.,0.42,0.84,0.95,0.72,0.46,0.50
6,0.,0.,0.,0.,0.,0.,0.36,0.72,0.95,0.53,0.27,0.
7,0.,0.,0.,0.,0.,0.26,0.32,0.65,0.95,0.42,0.21,0.
8,0.,0.,0.,0.,0.,0.19,0.34,0.42,0.42,0.30,0.19,0.19
9,0.,0.,0.,0.,0.28,0.28,0.27,0.34,0.42,0.42,0.30,0.27,0.19
10,0.,0.,0.19,0.19,0.28,0.57,0.76,0.80,0.88,0.50,0.30,0.27,0.19
11,0.,0.,0.19,0.27,0.50,0.65,0.46,0.46,0.88,0.50,0.30,0.19,0.19
12,0.,0.19,0.19,0.27,0.57,0.80,0.31,0.34,0.91,0.50,0.24,0.19,0.
13,0.,0.19,0.34,0.69,0.80,0.40,0.31,0.46,0.91,0.50,0.24,0.,0.
14,0.,0.30,0.61,0.50,0.57,0.28,0.33,0.37,0.91,0.50,0.24,0.,0.
15,0.24,0.24,0.50,0.34,0.23,0.28,0.28,0.57,0.88,0.50,0.24,0.,0.
16,0.24,0.50,0.50,0.19,0.21,0.,0.28,0.57,0.80,0.50,0.24,0.,0.
17,0.32,0.32,0.32,0.26,0.,0.,0.40,0.80,0.40,0.24,0.24,0.,0.
18,0.32,0.32,0.32,0.,0.,0.,0.36,0.72,0.36,0.,0.,0.,0.
19,0.32,0.32,0.32,0.,0.,0.,0.28,0.57,0.28,0.,0.,0.,0.
20,0.,0.,0.,0.,0.,0.,0.28,0.57,0.28,0.,0.,0.,0.
21,0.,0.,0.,0.,0.,0.,0.40,0.80,0.40,0.,0.,0.,0.
22,0.,0.,0.,0.,0.,0.,0.34,0.69,0.34,0.,0.,0.,0.
23,0.,0.,0.,0.,0.32,0.32,0.35,0.76,0.38,0.,0.,0.,0.
24,0.,0.,0.,0.32,0.32,0.65,0.65,0.69,0.34,0.,0.,0.,0.
25,0.,0.,0.,0.32,0.65,0.65,0.32,0.34,0.34,0.,0.,0.,0.
26,0.,0.,0.32,0.32,0.65,0.32,0.32,0.,0.,0.,0.,0.,0.
27,0.,0.,0.32,0.65,0.30,0.19,0.,0.,0.,0.,0.,0.,0.
28,0.,0.,0.32,0.65,0.30,0.19,0.,0.,0.,0.,0.,0.,0.
29,0.,0.,0.32,0.32,0.57,0.19,0.,0.,0.,0.,0.,0.,0.
```

## 8.4.2 Sample Input for Main Programs TMAIN5B, TMAIN5TB and TMAIN5VB (DATAML5)

```
 1          20   20
 2       0.0002222,1.9309
 3       1.0,0.002228,63.114,1.0
 4       0.01,100,1.8,1.0
 5       0.03565,0.03565,0.2
 6       0.0
 7       0.0002673
 8       1.0,0.002228,63.114
 9       1.000428,-0.0003019,-0.0000046
10       30.0
11       4360.0,0.0,0.0
12       31.7,0.698,1.37,-1.32
13       250.0,0.000667,0.00000667
14       3,3
15       27,3,1,0.0,0.111,0.1833
16       27,3,2,0.0,0.111,0.1833
17       27,3,3,0.0,0.111,0.1833
18       19,2,1,0.111,0.0
19       19,2,2,0.111,0.0
20       19,2,3,0.111,0.0
```

## 8.4.3 Sample Input for Main Program TMAIN6 (DATAML6)

| CARD NO. | FORMAT & VALUE |
|---|---|
| 1 | 20 20 |
| 2 | 0.0002222,1.9309 |
| 3 | 1.0,0.002228,63.114,1.0 |
| 4 | 0.01,100,1.8,1.0 |
| 5 | 0.03565,0.03565,0.2 |
| 6 | 0.0 |
| 7 | 0.0002673 |
| 8 | 1.0,0.002228,63.114 |
| 9 | 1.000428,-0.000019,-0.0000046 |
| 10 | 30.0 |
| 11 | 4360.0,0.0,0.0 |
| 12 | 31.7,0.698,1.37,-1.32 |
| 13 | 250.0,0.000667,0.00000667 |

## 8.5 Program Execution Procedure

The execution procedure for far field stratified case is similar to the unstratified case which is explained in section (7.5), except TMAIN 4T, TMAIN 5 and TMAIN 6 have to be replaced by TMAIN 5TB, TMAIN 5B and TMAIN 6B and the data elements ITPK1, INDATA 5 have to be changed to ITLK1, DATAML5.

## 8.6 Sample Output

The sample output for far field stratified is similar to the near field which is given in section (6.6).

# SOME REPRESENTATIVE RESULTS FOR FAR-FIELD MODEL (STRATIFIED CASE)

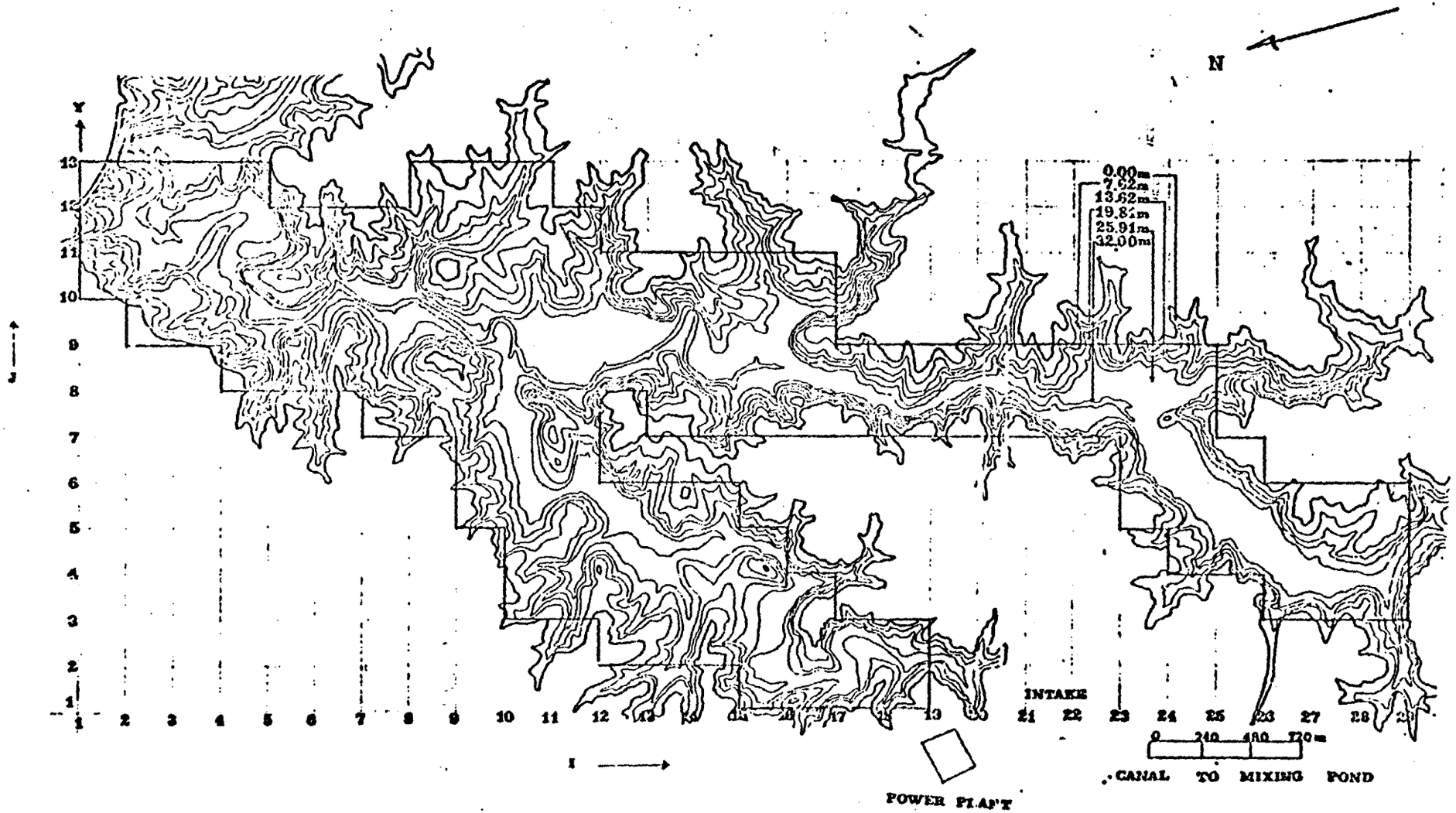
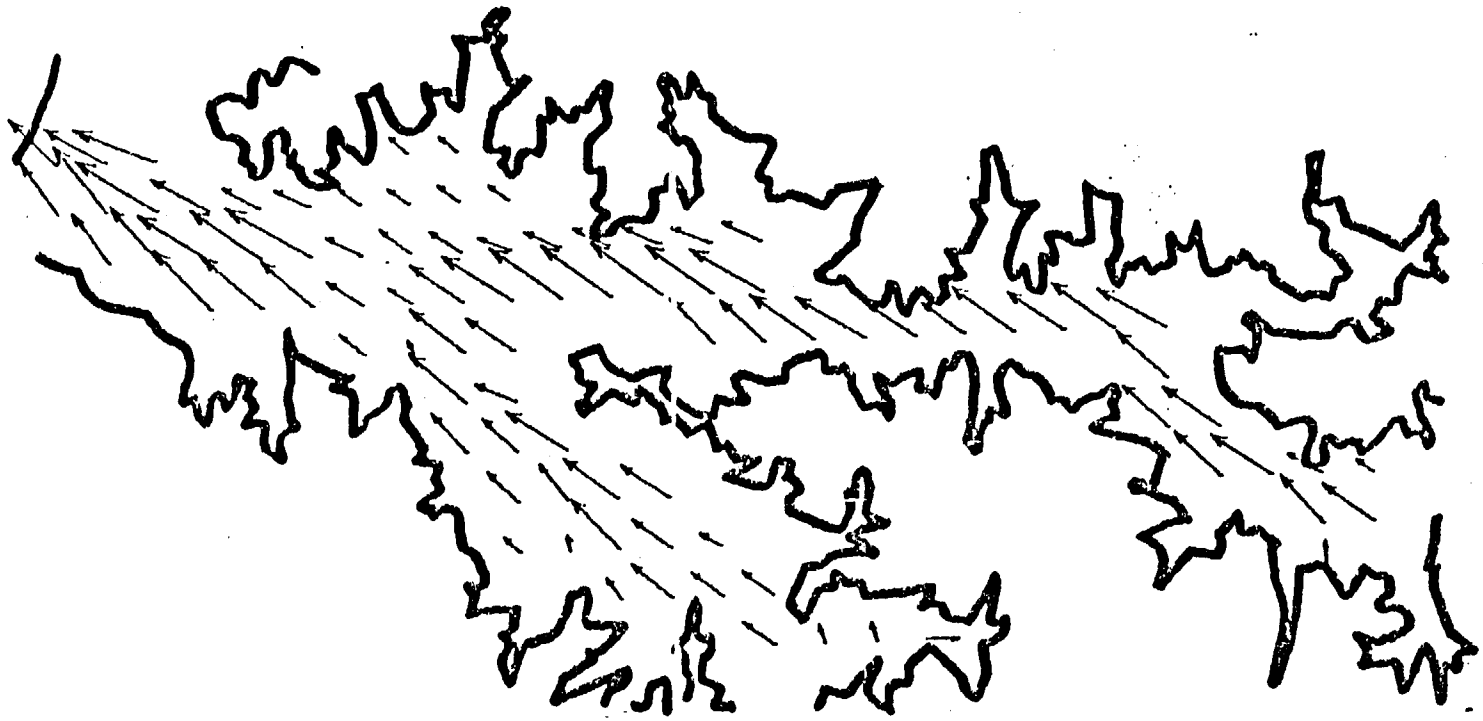



Fig. 8.1 **COMPUTATIONAL GRID FOR MAIN LAKE AT BELLEWS LAKE SITE**

| | |
|---|---|
| Time Step | : 1 sec |
| Total Time | : 75 sec |
| Wind | : 4.96m/sec (11.1 mph) 220° |
| Plant Discharge | : 229 X $10^6$ kg/hr (1012 X $10^3$ GPM) |
| Horizontal Viscosity | : 4.5 X $10^6$ $cm^2$/sec |
| Vertical Viscosity | : 10-0.1 $cm^2$/sec |
| Depth | : Variable |



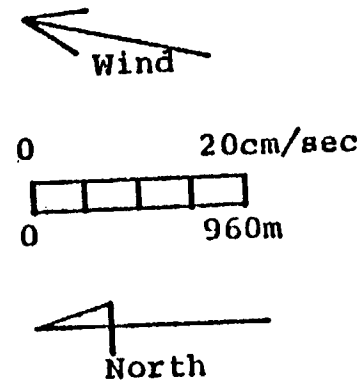



Fig. 8.2 Velocity distribution at the surface in the Main Lake at Lake Belews site for August 26, 1976.

| | |
|---|---|
| Time Step | : 1 sec |
| Total Time | : 75 sec |
| Wind | : 4.96 m/sec (11.1 mph) 220° |
| Plant Discharge | : 229 X $10^6$ kg/hr ($1012 \times 10^6$ $cm^2$/sec) |
| Horizontal Viscosity | : 4.5 X $10^6$ $cm^2$/sec |
| Vertical Viscosity | : 10-0.1 $cm^2$/sec |
| Depth | : Variable |

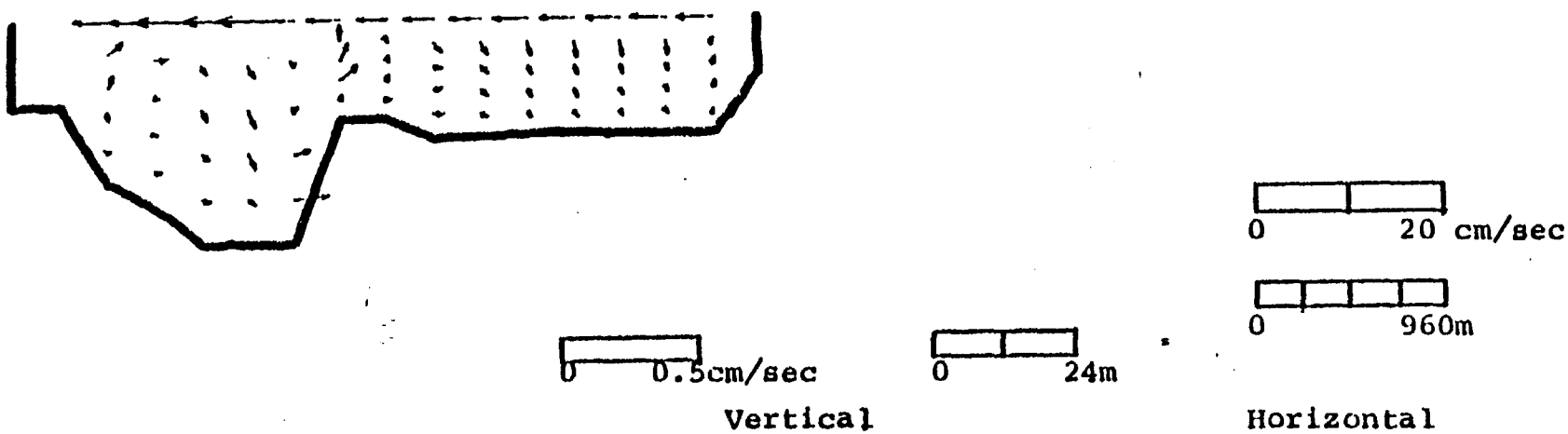



Fig. 8.3 Velocity distribution at vertical section along J=10 in the Main Lake at Lake Belews site for August 26, 1976.

Time Step : 1 sec
Total Time: 75 sec
Wind : 4.96 m/sec (11.1 mph) 220°
Plant Discharge : 229 X $10^6$ kg/hr
(1012 X $10^3$ GPM)
Horizontal Viscosity: 4.5 X $10^6$ $cm^2$/sec
Vertical Viscosity: 10-0.1 $cm^2$/sec
Depth : Variable

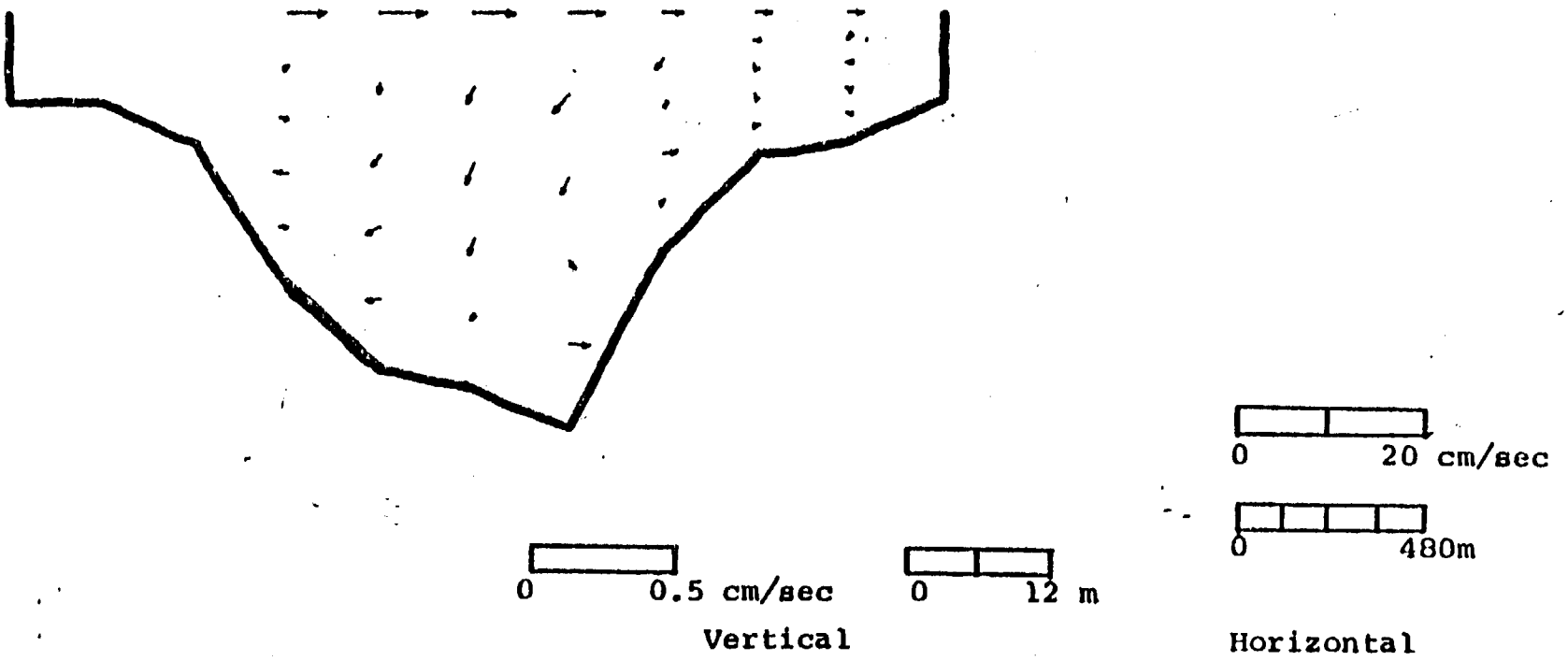



Fig. 8.9 Velocity distribution at vertical section along I=10 in the Main Lake at Lake Belews site for August 26, 1976.

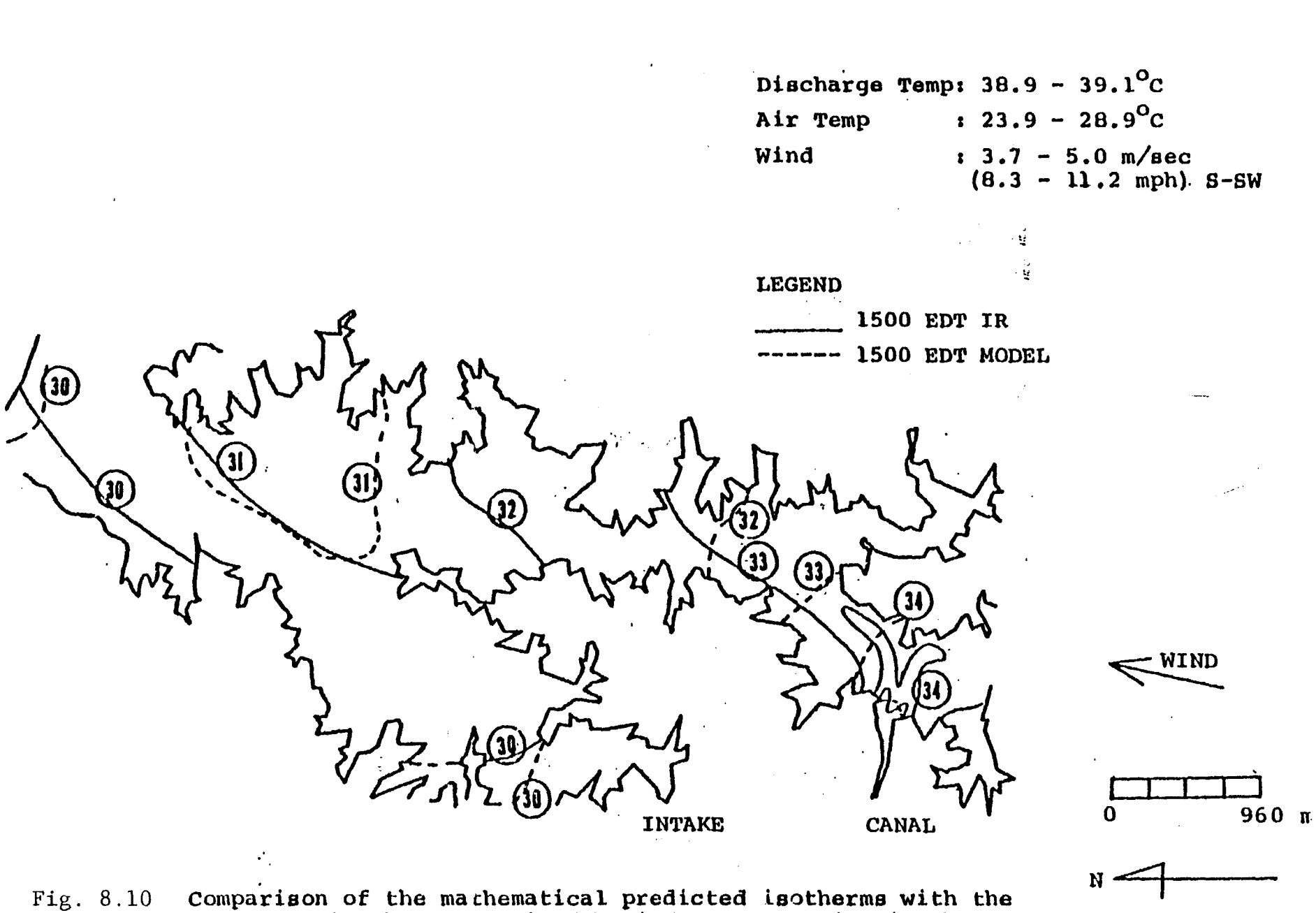



Fig. 8.10 Comparison of the mathematical predicted isotherms with the afternoon isotherms obtained by infrared scanning in the Main Lake at Lake Belews site for August 26, 1976 at 1500 hours EDT.

# REFERENCES


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

2. Venkata, J., "A Numerical Investigation of Thermal Plumes", Ph.D. thesis submitted to the University of Miami, August, 1977.

3. Mathavan, S.K., "Experimental and Numerical Study of Current and Temperature Fields in Lake Belews. An Artificial Cooling Pond", Ph.D. thesis submitted to the University of Miami, August, 1977.

4. Lee, S.S. and Sengupta, S., "Three Dimensional Thermal Pollution Models", Phase III, prepared for National Aeronautics and Space Administration under contract NASA 10-8926, December, 1976.

5. Harleman, D.R.F., and Stolzenback, K.D., "Engineering and Enviromental Aspects of Heat Disposal from Power Generation." Dept. of Civil Engineering, M.I.T., June, 1975.

6. Parker, F.L. and Krenkel, P.A., Engineering Aspects of Thermal Pollution (1968).

7. Sengupta, S., and Lick, W., "A Numerical Model for Wind Driven Circulation and Temperature Field in Lakes and Ponds", FTAS/TR-74-99, Case Western Reserve University (1974)

8. Documentation of 3 Dimensional Mathematical Models for Thermal Pollution Studies. Vols. I, and III.

# APPENDIX A

## THE EQUILIBRIUM TEMPERATURE AND THE SURFACE HEAT TRANSFER COEFFICIENT

The net heat transfer through a water surface is composed of radiation penetrating the water surface from above, radiation out of the water surface, evaporation, and conduction transfer. These are indicated schematically in the following figure.

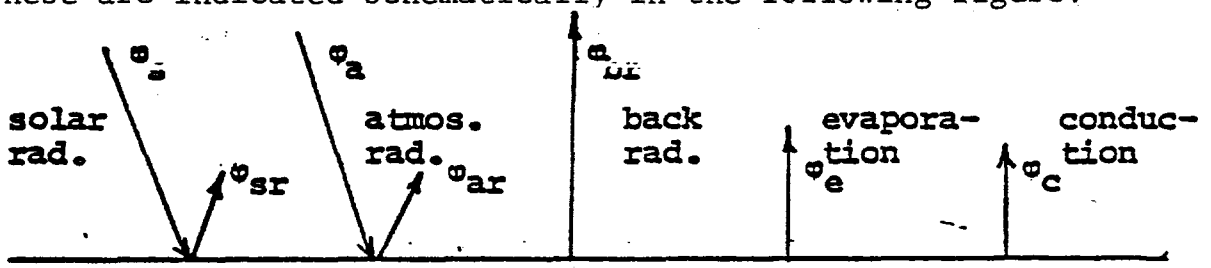



Fig. 1 Heat Transfer Mechanisms at the Water Surface

The following heat balance results,

$$\varphi_n = \underbrace{\varphi_s - \varphi_{sr}}_{\varphi_{sn}} + \underbrace{\varphi_a - \varphi_{ar}}_{\varphi_{an}} - \varphi_{br} - \varphi_e - \varphi_c \quad \text{.....(A-1a)}$$

where $\varphi_n$ = net heat input = $\varphi_{sn} + \varphi_{an} - \varphi_{br} - \varphi_e - \varphi_c$ ...(A-1b)

Now, equation (A-1) may be rewritten as,

$$\varphi_n = \varphi_r - \varphi_L \quad \text{.....(A-2)}$$

Where $\varphi_r$ = net absorbed radiation = $\varphi_{sn} + \varphi_{an}$

and $\varphi_L = \varphi_{br} + \varphi_e + \varphi_c$

A.1 Equilibrium Temperature Calculation, Te (See Appendix B)

Under equilibrium conditions equation (A-2) yields,

$$\varphi_n = 0 = \varphi_r - \varphi_L$$

so that

$$\varphi_r = \varphi_L \quad \text{.....} \quad \text{(A-3)}$$

Then by using the approximate formulae in Harelman et al (1975) we obtain by setting $T_s = T_e$,

$$0.94\,\varphi_{sc}(1-0.65C^2) + 1.16 \times 10^{-13}(T_a^*)^6(1+0.17C^2) = 0.97\sigma(T_e^*)^4 + F(W)\,[(e_s-e_a) + C_b(T_e-T_a)] \quad \text{(A-4)}$$

Where $\varphi_{sc}$ = clear sky solar radiation

C = cloudiness ratio

$T_a$ = air temperature ($^oC$ or $^oF$)

$T_e$ = equilibrium temperature ($^oC$ or $^oF$)

$T^*$ = absolute temperature ($^oK$ or $^oR$)

F(W) = windspeed function (BTU/ft$^2$/day, mm Hg)

$e_s$ = saturated vapor pressure at water surface temperature (mm Hg)

$e_a$ = saturated vapor pressure at air temperature (mm Hg)

$\sigma$ = Stefan-Boltxmann constant $\simeq$ 4.1 X $10^{-8}$ BTU/ft$^2$, day, $^oR^4$

$C_b$ = Bowen constant = 0.255 mm Hg/$^oF$ (see Appendix B)

W = windspeed (mph)

For natural water surface,

$$F(W) = 17W \quad \text{(A-5a)}$$

and, for an artificially heated surface,

$$F(W) = 22.4\,(T_e - T_a)^{1/3} + 17W \quad \text{(A-5b)}$$

Thus, equation (A-4) becomes,

$$0.94\,\varphi_{sc}(1-0.65C^2) + 1.16 \times 10^{-13}(T_a^*)^6(1+0.17C^2) = 0.97\sigma(T_e^*)^4 + 17W\,[(e_s-e_a) + 0.255(T_e-T_a)] \quad \text{(A-6)}$$

Location - Miami (latitude 26°N)

Date - December 20

From CRC (1970),

$$@T_a = 25^{\circ}C \rightarrow e_a \cong 0.43 \text{ psia}$$

$$@T_e \cong 27^{\circ}C \text{ as guess} \cong e_s = e_e \cong 0.51 \text{ psia}$$

From Harleman et al (1975),

$$\phi_{sc} \cong 425 \text{ Langleys/day} = 1560 \text{ BTU/ft}^2\text{/day} \ldots \text{using}$$

100% sunshine curve at 26°N, Dec. 20

Note: 1 Langley/min. - 220.62 BTU/ft$^2$/day ... using

100% sunshine curve at 26°N, December 20

Note: 1 Langley/min. = 220.62 BTU/ft$^2$, hr. = 1 calorie/cm$^2$,min.

Then using equation (C-6) with C=0

$$0.94\ (1560)\ (1) + 1.16 \times 10^{-13} (5.37 \times 10^2)^6 (1) = 4250$$

$$4 \times 10^{-8} (5.406 \times 10^2)^4 + 170 [\ (.255)\ (2) + (.08)\ (51.7)]$$

4206 close enough!

$$\therefore\ T_e \cong 27^{\circ}C$$

(where 1 psia = 51.7 mm Hg)

Then from equation (A-8),

$$K = 3.88 \times 4.1 \times 10^{-8} (5.406 \times 10^2)^3 + 170\,(.255 + 0.0251 \times 51.7)$$

$$K \cong 290 \text{ BTU/ft}^2, {}^{\circ}\text{F, day}$$

where $\left.\frac{\partial e_s}{\partial T}\right]_{T=T_{av}} \cong \frac{e_e - e_a}{T_e - T_a} = .0251$

or,

$$0.94\,\varphi_{sc}(1-0.65C^2) + 1.16 \times 10^{-13}(T_a^*)^6(1+0.17C^2)$$

$$= 0.97\varphi(T_e^*)^4 + [\,22.4(T_e-T_a)^{1/3} + 14W]$$

$$[(e_s-e_a) + 0.255(T_e-T_a)\,]\text{----------(A-7)}$$

Therefore, for known $\varphi_{sc}$, $e_s$, $e_a$, $T_a$ and W $T_e$ can be determined by trial and error methods.

A.2 Surface Heat Transfer Coefficient (K)

From Harleman et al (1975) the surface heat transfer coefficient K, can be determined as follows

$$K = \frac{\partial \varphi_L}{\partial T_{av}} = \frac{\partial \varphi_n}{\partial T_{av}}, \text{ since } \varphi_r \neq \varphi_r(T_s) \text{ and } \therefore \frac{\partial \varphi_r}{\partial T_{av}} = 0.$$

where $T_{av} = (T_s + T_e)/2$

Thus,

$$K = 3.88\sigma(T_{av}^*)^3 + F(W)_{Tav}\left[\left(\frac{\partial e_s}{\partial T}\right)_{T=T_{av}} + C_b\right]$$

$$+ [(e_{av} - e_a) + C_b(T_{av} - T_a)]\,\frac{\partial F(W)}{\partial T_{av}}\text{----------(A-8)}$$

Where

$$\frac{\partial F(W)}{\partial T_{av}} = \begin{cases} 0 & \text{for natural water surface} \\ 1/3(22.4)(T_{av}-T_a)^{-2/3} & \text{for artificially heated water surface} \end{cases}$$

A.3 Nummerical Example

Consider natural water surface

C = 0

$T_a$ = 25°C

W = 10 mph

A.4 Discussion

The quilibrium surface temperature, $T_e$, for a natural water surface, can be greater than the atmospheric temperature, $T_a$, whereby $T_s$ increases from values below $T_a$ up to $T_e$ as equilibrium is reached. As can be seen in the figure below $T_e$ can be greater or smaller than $T_a$ depending on the time of the day. Simply $T_e > T_a$ during the hours of sunshine and $T_e < T_a$ at night when the water surface is cooling.

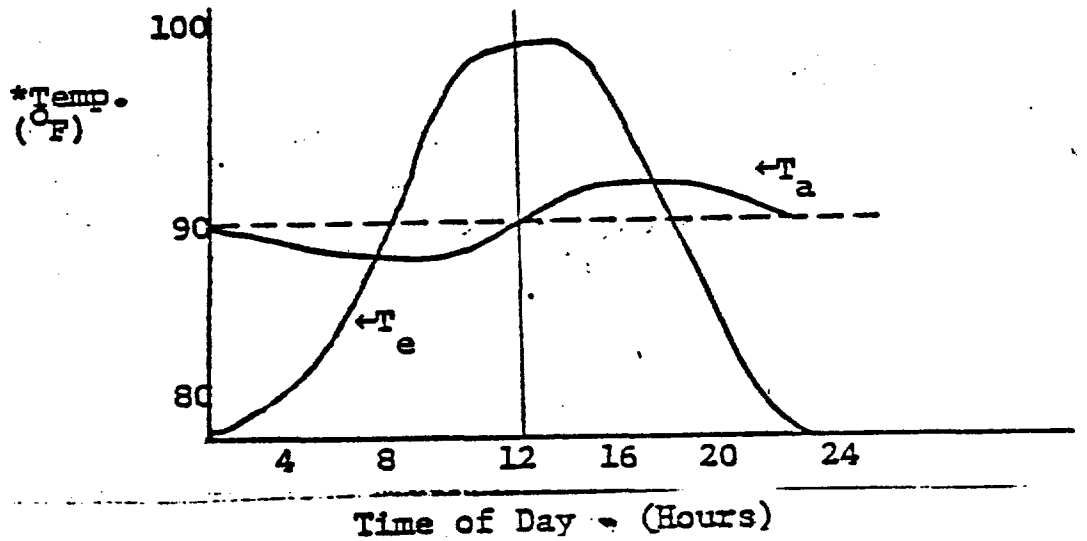



Fig. ( 2 ) Daily variation of air temperature and computed equilibrium temperature.

* This plot is taken from Parker (1968) and has no relation to the numerical example given in this paper. However, the numerical example considered 100% possible hours of sunshine

# APPENDIX B

## HEAT TRANSFER MECHANISMS

The analysis in this section is taken from Harleman et al. (1975) and is summarized.

### B.1 Solar Radiation (short wave)

The incident solar radiation impinging on the water surface may be expressed as

$$\varphi_s = \varphi_{sc}(1-0.65C^2)$$

Where $\varphi_{sc}$ = clear sky solar radiation obtained using the 100% possible sunshine curve (given in Appendix A)

C = fraction of sky covered by clouds.

The reflected solar radiation is typically 6% of incident solar radiation, hence the net solar radiation absorbed by the water surface is,

$$\varphi_{sn} = \varphi_s - \varphi_{sr} \simeq 0.94\varphi_{sc}(1-0.65C^2)$$

### B.2 Atmospheric Radiation (long wave)

The basic equation for the incident atmospheric radiation, is given as

$$\varphi_a = \epsilon\sigma T_a^{*4}$$

Where $\epsilon$ = average emmitance of the atmosphere

$\sigma$ = Stefan-Boltzmann constant

$T_a^*$ = air temperature (absolute)

However, good agreement with experimental data has indicated that $\epsilon$ is a function of $T_a$, and specifically, $T_a^{*6}$ dependence gives best results for atmospheric radiation at low temperatures,

as well as providing a good fit at high temperatures. Clear sky incident atmospheric radiation, $\varphi_{ac}$, may be expressed as,

$$\varphi_{ac} = 1.2 \times 10^{-13}(T_a^*)^6$$

and, then incident atmospheric radiation including cloudiness may be expressed as,

$$\varphi_a = \varphi_{ac}(1 + 0.17C^2)$$

A figure of 3% is usually accepted as reflectance of a water surface to longwave radiation. Thus the net atmospheric radiation absorbed by the surface is

$$\varphi_{an} = \varphi_a - \varphi_{ar} = 0.97\varphi_a$$

and, therefore, we have

$$\varphi_{an} = 1.16 \times 10^{-13}(T_a^*)^6(1 + 0.17C^2)$$

B.3 Longwave Radiation from the Water Surface, $\varphi_{br}$

Harleman et al (1975) note that the emmissivity of a water surface is independent of temperature and salt or colloidal concentrations, and gives a value of 0.97. Thus we obtain,

$$\varphi_{br} = 0.97\sigma(T_s^*)^4$$

Where $T_s$ = water surface temperature.

B.4 Evaporative Heat Flux, $\varphi_e$

Evaporation from a water surface occurs as a result of both forced (wind-driven ) convection and free (bouyancy driven) convection. The evaporation from a water surface is usually

written (mass/area/time) as

$$E = \rho F(W_z)(e_s - e_z)$$

Where, E = mass vlux (mass/area/time)

$\rho_w$ = density of water

$W_z$ = windspeed at height z above surface

F(Wz) = windspeed function for mass flux including both free and forced convection effects (length/time/pressure)

$e_s$ = saturated vapor pressure at $T_s$

$e_z$ = vapor pressure at height z above surface

Then writing the above equation in heat units, the evaporative heat flux $\varphi_e$, is given by,

$$\varphi_e = F(W_z)(e_s - e_z)$$

Where $F(W_z)$ = windspeed function for heat flux (energy/area/time/pressure)

Now, dropping the z subscript (and assuming W measured "z" above the surface $\sim$ W at the surface) we may express F(W) for a natural water surface and for an artificially heated water surface as

$F(W) = 17W...$ natural water surface

and

$F(W) = 22.4\ (T_s - T_a)^{1/3} + 14W...$ artifically heated surface.

B.5 Conduction Heat Flux, $\varphi_c$

Bowen (1926) has suggested that conduction can be directly

related to evaporative fluxes by assuming that eddy diffusivities of heat and mass are identical. Thus,

$$\varphi_c = R_b \varphi_e$$

Where $R_b = C_b \left| \frac{T_s - T_a}{e_s - e_a} \right|$ = Bowen Ratio

and $C_b$ = Bowen constant = 0.255 mm Hg/$^{o}$F

and, therefore the conduction heat flux, $\varphi_c$, may be expressed as,

$$\varphi_c = C_b F(W) (T_s - T_a)$$

# COMPUTER PROGRAMS

## 9.0 DESCRIPTION OF COMPUTER PROGRAMS

For the readers convenience, the computer programs and their description are included in this section. There are two sets of main programs, one set for the near field and the second for the far-field. These main programs are described along with the flow charts in subsection (9.1.1) Subsection (9.2) describes subroutines for both the near field and far field.

### 9.1 Main Programs for the Near-Field and Far-Field

There are four main programs for the near-field. They are AMAIN1, AMAIN2, TMAIN1 and TMAIN2. AMAIN programs are for velocity only, and TMAIN programs are for velocity and temperature. AMAIN1 and TAMIN1 are to be used for constant depth and AMAIN2 and TMAIN2 for variable depth conditions.

### 9.1.1 AMAIN 1 (Main Program for Near-Field)

The main program reads in the data, initializes the necessary quantities and coordinates the subroutines and calculates velocities only for a near field problem. The parameter statement defines the size of the computational domain. The subroutine READ3 reads MAR and MRH matrices. The subroutine 'INITIA' sets the velocities in the receiving basin equal to zero. If there is current, the subroutine 'CURNT' has to be called after the subroutine 'INITIA', which would set the velocities everywhere in the domain equal to the CURNT velocity. Then it follows a set of subroutines to calculate the velocity field for the entire domain for the given discharge and ambient conditions.

```
*DULL(1).AMAIN1
  1       C       CALCULATES U,V,W, FOR VARIABLE DEPTH LAKE WITH STRETCHG
  2       C       ALGORITHM
  3              PARAMETER IN=21,JN=21,KN=5,IWN=20,JWN=20
  4              DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IWN,JWN,KN),
  5             CWR(IN,JN,KN),WRH(IWN,JWN,KN),P(IWN,JWN),D(IN,JN,KN),E(IN,JN,KN),
  6             CWHLDT(IWN,JWN),XINT(IN,JN),YINT(IN,JN),H(IN,JN,KN),G(IN,JN,KN),
  7             CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN),FH(IWN,JWN),
  8             CDPSX(IN,JN),DPSY(IN,JN)
  9              DIMENSION A3(KN)
 10              INM1=IN-1
 11              READ 1, IRUN
 12              READ 1, LN
 13       1      FORMAT (I5)
 14              READ 2, VVIS,ABR
 15              A3(1)=VVIS
 16              A3(2)=VVIS
 17              A3(3)=VVIS
 18              A3(4)=VVIS
 19              A3(5)=VVIS
 20              READ 2, AI,AH,AV,AP
 21              READ 2, EPS,MAXIT,OMEGA,ARBP
 22              READ 2, DX,DY,DZ
 23              READ 2, AA,BB,CC
 24              DL2=(DX*DX*DY*DY)/(DX*DX+DY*DY)
 25       2      FORMAT ()
 26              IF(IRUN.GT.0) GO TO 3
 27              CALL INITIA(IN,JN,KN,IWN,JWN,U,V,W,WH,D,E,P,I,J,K,IW,JW,ARBP)
 28              CALL READ3(I,J,IN,JN,IW,JW,IWN,JWN,MAR,MRH)
 29              CALL HEIGHT(I,J,K,IN,JN,KN,HI,HX,HY,CC)
 30              CALL INLET(I,J,K,IN,JN,KN,G,V,AA,BB)
 31              TTOT=0.0
 32              GO TO 4
 33            3 CONTINUE
 34              REWIND 7
 35              READ (7) (((U(I,J,K),K=1,KN),J=1,JN),I=1,IN),
 36             C(((V(I,J,K),K=1,KN),J=1,JN),I=1,IN),
 37             C(((D(I,J,K),K=1,KN),J=1,JN),I=1,IN),
 38             C(((E(I,J,K),K=1,KN),J=1,JN),I=1,IN),
 39             C(((WH(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
 40             C(((W(I,J,K),K=1,KN),J=1,JN),I=1,IN),
 41             C(((WR(I,J,K),K=1,KN),J=1,JN),I=1,IN),
 42             C(((WRH(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
 43             C((P(IW,JW),JW=1,JWN),IW=1,IWN)
 44             C,((HI(I,J),J=1,JN),I=1,IN),
 45             C((HX(I,J),J=1,JN),I=1,IN),
 46             C((HY(I,J),J=1,JN),I=1,IN),
 47             C((MAR(I,J),J=1,JN),I=1,IN),
 48             C((MRH(IW,JW),JW=1,JWN),IW=1,IWN),
 49             CAI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,TTOT
 50              REWIND 7
 51              CALL INLET(I,J,K,IN,JN,KN,G,V,AA,BB)
 52       4      CONTINUE
 53              READ 2, DT
 54              DO 5 L=1,LN
 55              TTOT=TTOT+DT
 56              CALL ERROR(IWN,JWN,IW,JW,DT,WH,WHLDT,KN,MRH)
```

```
      CALL WHTOP(IW,JW,IWN,JWN,KN,WH,K,MRH)
      CALL WHATIJ(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
      CALL INTE(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,MAR,XINT,YINT,A3,AI,
     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 DPSXY(I,J,IN,JN,IW,JW,IWN,JWN,DPSX,DPSY,P,DX,DY,MAR)
      CALL FORCE(I,J,IW,JW,XINT,YINT,WHLDT,DX,DY,HI,HX,HY,MRH,
     C DPSX,DPSY,FH,AP,IN,JN,IWN,JWN)
      CALL PRE2(EPS,MAXIT,IN,JN,P,ITN,DPSX,DPSY,FH,DL2,OMEGA,
     CMRH,I,J,K,IW,JW,DX,DY,EX,IWN,JWN,ARBP)
      CALL UV(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,D,E,H,G,DX,DY,DZ,
     CW,DT,AI,AP,AH,AV,A3,HI,HX,HY,P,MAR)
      CALL UANVC(I,J,K,IN,JN,KN,ABR,DT,U,V,H,G,HI,MAR)
      CALL UVTOP(H,G,TAUX,TAUY,I,J,K,DZ,IN,JN,KN,HI,MAR)
      CALL OUTVEL(I,J,K,IN,JN,KN,H,G)
      CALL OLDUV(I,J,K,IN,JN,KN,U,V,D,E)
      CALL OLDUV(I,J,K,IN,JN,KN,H,G,U,V)
      CALL RWH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,DX,DY,DZ,MRH)
5     CONTINUE
      CALL WHATIJ(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
      CALL RWRH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,HX,HY,
     CDX,DY,DZ,MRH,WRH)
      CALL RWR(I,J,K,IN,JN,KN,U,V,W,WR,HI,HX,HY,DZ,MAR)
      CALL STORE(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,
     CHX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TTOT)
      CALL PRPARA(AI,AH,AV,AP,DX,DY,DZ,DT,DL2,MAXIT,EPS,OMEGA,
     CARBP,TAUX,TAUY,TTOT,MAR,MRH,IN,JN,IWN,JWN)
      CALL PREROR(IWN,IW,WHLDT,JW,JWN,IN,JN)
      CALL PRINTE(I,J,IW,JW,XINT,YINT,IN,JN)
      CALL PRSORC(IW,JW,IWN,JWN,FH)
      CALL PDPSXY(I,J,IN,JN,DPSX,DPSY)
      CALL PRITEX(ITN,EX)
      CALL PRPINT(IW,JW,IWN,JWN,P)
      CALL PRUV(I,J,K,IN,JN,KN,U,V)
      CALL PRWH(IW,JW,K,IWN,JWN,KN,WRH)
      END
```

### 9.1.2 AMAIN2 (Main Program for Near-Field)

This is same as AMAIN1, except, this main program is to be used when the depth is variable. The subroutine GRADS computes slopes of the bottom in x and y directions respectively.

```
DULL(1).AMAIN2
 1       C       CALCULATES U,V,W, FOR VARIABLE DEPTH LAKE WITH STRETCHING
 2       C       ALGORITHM
 3              PARAMETER IN=18,JN=21,KN=5,IWN=17,JWN=20
 4              DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IWN,JWN,KN),
 5             CWR(IN,JN,KN),WRH(IWN,JWN,KN),P(IWN,JWN),D(IN,JN,KN),E(IN,JN,KN),
 6             CWHLDT(IWN,JWN),XINT(IN,JN),YINT(IN,JN),H(IN,JN,KN),G(IN,JN,KN),
 7             CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN),F4(IWN,JWN),
 8             CDPSX(IN,JN),DPSY(IN,JN),JX(JN)
 9              DIMENSION A3(KN)
10              INM1=IN-1
11              READ 1, IRUN
12              READ 1,LN,LLN
13       1      FORMAT (16I5)
14              READ 2, VVIS,ABR
15              A3(1)=VVIS
16              A3(2)=VVIS
17              A3(3)=VVIS
18              A3(4)=VVIS
19              A3(5)=VVIS
20              READ 2, AI,AH,AV,AP
21              READ 2, EPS,MAXIT,OMEGA,ARBP
22              READ 2, DX,DY,DZ
23              READ 2,AA,BB,CC
24              READ 2,TAUX,TAUY
25              DL2=DX*DX
26              READ 2,DT
27       2      FORMAT ()
28              IF(IRUN.GT.0) GO TO 3
29              TTOT=0.0
30              CALL INITIA(IN,JN,KN,IWN,JWN,U,V,W,WH,D,E,P,I,J,K,IW,JW,ARBP)
31              CALL READ3(I,J,IN,JN,IW,JW,IWN,JWN,MAR,MRH)
32              CALL HEIGH1(I,J,K,IN,JN,KN,HI,HX,HY,CC,JX)
33              CALL GRADS1(IN,JN,KN,IWN,JWN,HI,HX,HY,MAR,MRH,DX,DY)
34              GO TO 4
35            3 CONTINUE
36              REWIND 7
37              READ (7) (((U(I,J,K),K=1,KN),J=1,JN),I=1,IN),
38             C(((V(I,J,K),K=1,KN),J=1,JN),I=1,IN),
39             C(((D(I,J,K),K=1,KN),J=1,JN),I=1,IN),
40             C(((E(I,J,K),K=1,KN),J=1,JN),I=1,IN),
41             C(((WH(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
42             C(((W(I,J,K),K=1,KN),J=1,JN),I=1,IN),
43             C(((WR(I,J,K),K=1,KN),J=1,JN),I=1,IN),
44             C(((WRH(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
45             C((P(IW,JW),JW=1,JWN),IW=1,IWN)
46             C,((HI(I,J),J=1,JN),I=1,IN),
47             C((HX(I,J),J=1,JN),I=1,IN),
48             C((HY(I,J),J=1,JN),I=1,IN),
49             C((MAR(I,J),J=1,JN),I=1,IN),
50             C((MRH(IW,JW),JW=1,JWN),IW=1,IWN),
51             CAI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,TTOT
52              REWIND 7
53       4      CONTINUE
54              DO 6 LL=1,LLN
55              DO 5 L=1,LN
56              TTOT=TTOT+DT
```

```
57              CALL INLET (I,J,K,IN,JN,KN,V,G,AA,BB)
58              CALL ERROR(IWN,JWN,IW,JW,DT,WH,WHLDT,KN,MRH)
59              CALL WHTOP(IW,JW,IWN,JWN,KN,WH,K,MRH)
60              CALL WHATIJ(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
61              CALL INTE(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,MAR,XINT,YINT,A3,AI,
62             CAH,AV,TAUX,TAUY,DX,DY,DZ,D,E,DT,DPSX,DPSY,AP)
63              CALL CORINT(I,J,K,IN,JN,KN,ABR,U,V,XINT,YINT,DZ,HI,MAR)
64              CALL DPSXY(I,J,IN,JN,IW,JW,IWN,JWN,DPSX,DPSY,P,DX,DY,MAR)
65              CALL FORCE(I,J,IW,JW,XINT,YINT,WHLDT,DX,DY,HI,HX,HY,MRH,
66             CDPSX,DPSY,FH,AP,IN,JN,IWN,JWN,RINTX,RINTY,U,W,EUL,ABR,MAR,KN)
67              CALL PRE2(EPS,MAXIT,IN,JN,P,ITN,DPSX,DPSY,FH,DL2,OMEGA,
68             CMRH,I,J,K,IW,JW,DX,DY,EX,IWN,JWN,ARBP)
69              CALL UV(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,D,E,H,G,DX,DY,DZ,
70             CW,DT,AI,AP,AH,AV,A3,HI,HX,HY,P,MAR)
71              CALL UANVC(I,J,K,IN,JN,KN,AER,DT,U,V,H,G,HI,MAR)
72              CALL UVTOP(H,G,TAUX,TAUY,I,J,K,DZ,IN,JN,KN,HI,MAR)
73               CALL OUTVEL(I,J,K,IN,JN,KN,H,G)
74              CALL OLDUV(I,J,K,IN,JN,KN,U,V,D,E)
75              CALL OLDUV(I,J,K,IN,JN,KN,H,G,U,V)
76              CALL RWH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,DX,DY,DZ,MRH)
77        5     CONTINUE
78              CALL WHATIJ(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
79              CALL RWRH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,HX,HY,
80             CDX,DY,DZ,MRH,WRH)
81              CALL RWR(I,J,K,IN,JN,KN,U,V,W,WR,HI,HX,HY,DZ,MAR)
82              CALL STORE(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,
83             CHX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TTOT)
84              CALL PRPARA(AI,AH,AV,AP,DX,DY,DZ,DT,DL2,MAXIT,EPS,OMEGA,
85             CARBP,TAUX,TAUY,TTOT,MAR,MRH,IN,JN,IWN,JWN)
86              CALL PRUV(I,J,K,IN,JN,KN,U,V)
87        6     CONTINUE
88              CALL PREROR(IWN,IW,WHLDT,JW,JWN,IN,JN)
89              CALL PRINTE(I,J,IW,JW,XINT,YINT,IN,JN)
90              CALL PRSOPC(IW,JW,IWN,JWN,FH)
91              CALL PDPSXY(I,J,IN,JN,DPSX,DPSY)
92              CALL PRITEX(ITN,EX)
93              CALL PRPINT(IW,JW,IWN,JWN,P)
94              CALL PRUV(I,J,K,IN,JN,KN,U,V)
95              CALL PRWH(IW,JW,K,IWN,JWN,KN,WRH)
96              END
```

9.1.3 TMAIN1 (Main Program for Near Field)

This is a main program and is used for obtaining velocity and temperature distribution in the near field of a constant depth basin. The parameter statement defines the size of the computational domain. The subroutine READ 3 reads MAR and MRH matrices. The subroutine INITIA initializes the velocity and pressure field. The subroutine INITIT initializes the temperature field. It sets temperature in the whole domain equal to the reference temperature. The Flow Chart which the main program follows is shown in Fig. ( 9.1). The subroutines are also given in the Flow Chart.

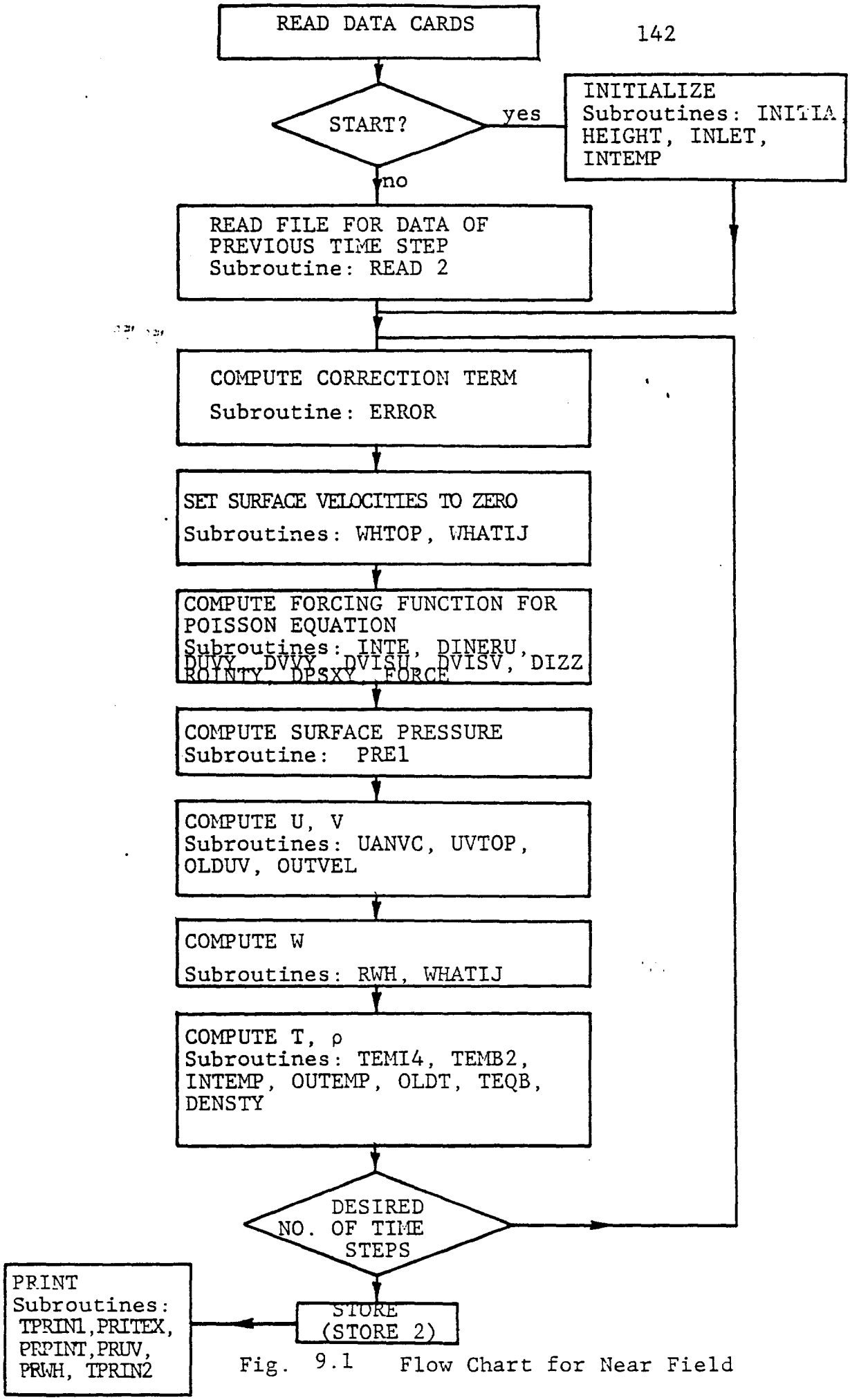



Fig. 9.1 Flow Chart for Near Field

```
*DOC.THAIN1
      PARAMETER IN=18,JN=22,KN=5,IWN=17,JWN=21
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IWN,JWN,KN),
     CWR(IN,JN,KN),WRH(IWN,JWN,KN),P(IWN,JWN),D(IN,JN,KN),E(IN,JN,KN),
     CWHLDT(IWN,JWN),XINT(IN,JN),YINT(IN,JN),H(IN,JN,KN),G(IN,JN,KN),
     CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN),FH(IWN,JWN),
     CDPSX(IN,JN),DPSY(IN,JN)
      DIMENSION A3(KN)
      DIMENSION T(IN,JN,KN),TP(IN,JN,KN),TD(IN,JN,KN),RO(IN,JN,KN),
     CRINTX(IN,JN,KN),RINTY(IN,JN,KN),WD(IN,JN,KN)
      DIMENSION TW(IWN,JWN,KN),ROW(IWN,JWN,KN)
      INM1=IN-1
      READ 1, IRUN
      READ 1, LN
      READ 1, LLN
    1 FORMAT (I5)
      READ 2, VVIS,ABR
      A3(1)=VVIS
      A3(2)=VVIS
      A3(3)=VVIS
      A3(4)=VVIS
      A3(5)=VVIS
      B3=VVIS
      READ 2, AI,AH,AV,AP
      READ 2, EPS,MAXIT,OMEGA,ARBP
      READ 2, DX,DY,DZ
      READ 2, TAI,TAH,TAV
      READ 2, A,B,C
      READ 2, T0
      READ 2, AKT,EUL,CW,CB
      READ 2,AA,BB,CC
      READ 2,TLL,TMM
    2 FORMAT ()
      DL2=DX*DX
      TREF=T0
      RREF=A+B*T0+C*T0*T0
      IF(IRUN.GT.0) GO TO 3
      CALL READ3(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,P0,MAR,MRH,TREF,
     CRREF,TW,ROW,TC)
      CALL HEIGHT(I,J,K,IN,JN,KN,HI,HX,HY,CC)
      CALL INLET(I,J,K,IN,JN,KN,V,G,AA,BB)
      CALL INTEMP(I,J,K,IN,JN,KN,T,TD,TLL,TMM)
      GO TO 4
    3 CONTINUE
      CALL READ2(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,HI,
     CHAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT
     C,CB,CW,A,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,T0,TAMB,TTOT)
      CALL INLET(I,J,K,IN,JN,KN,V,G,AA,BB)
      CALL INTEMP(I,J,K,IN,JN,KN,T,TD,TLL,TMM)
    4 CONTINUE
      READ 2, TAMB
      TE=(TAMB-TREF)/TREF
      READ 2,TAUX,TAUY
      READ 2, DT
```

```
      DO 6 LL=1,LLN
      DO 5 L=1,LN
      TTOT=TTOT+DT
      CALL ERROR(IWN,JWN,IW,JW,DT,WH,WHLDT,KN,MRH)
      CALL WHTOP(IW,JW,IWN,JWN,KN,WH,K,MRH)
      CALL WHATIJ(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
      CALL INTE(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,MAR,XINT,YINT,A3,AI,
     CAH,AV,TAUX,TAUY,DX,DY,DZ,D,E,DT,DPSX,DPSY,AP)
      CALL ROINTX(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,
     CMAR,RINTX,HX,XINT)
      CALL ROINTY(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,MAR,
     CRINTY,HY,YINT)
      CALL DPSXY(I,J,IN,JN,IW,JW,IWN,JWN,DPSX,DPSY,P,DX,DY,MAR)
      CALL FORCE(I,J,IW,JW,XINT,YINT,WHLDT,DX,DY,HI,HX,HY,MRH,
     CDPSX,DPSY,FH,AP,IN,JN,IWN,JWN)
      CALL PRE2(EPS,MAXIT,IN,JN,P,ITN,DPSX,DPSY,FH,DL2,OMEGA,
     CMRH,I,J,K,IW,JW,DX,DY,EX,IWN,JWN,ARBP)
      CALL UVT(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,D,E,H,G,DX,DY,DZ,
     CRINTX,RINTY,EUL,W,DT,AI,AP,AH,AV,A3,HI,HX,HY,P,MAR)
      CALL UVTOP(H,G,TAUX,TAUY,I,J,K,DZ,IN,JN,KN,HI,MAR)
      CALL OUTVEL(I,J,K,IN,JN,KN,H,G)
      CALL OLDUV(I,J,K,IN,JN,KN,U,V,D,E)
      CALL OLDUV(I,J,K,IN,JN,KN,H,G,U,V)
      CALL RWH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,DX,DY,DZ,MRH)
      CALL WHATIJ(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
      DO 20 I=1,IN
      DO 20 J=1,JN
      WD(I,J,1)=W(I,J,1)
20    CONTINUE
      DO 30 I=1,IN
      DO 30 J=1,JN
      W(I,J,1)=0.0
30    CONTINUE
      CALL TEMI4(I,J,K,IN,JN,KN,U,V,T,TD,DX,
     CCB,
     CDY,DZ,W,DT,TAI,TAH,TAV,B3,HI,HX,HY,MAR,AKT,TREF,TAMB)
      CALL TEMB2(I,J,K,IN,JN,KN,TD,DX,DY,DZ,MAR,CB,HI,AKT,CW,TAMB,
     CHX,HY,T,TREF,TAV,TAI,TAH,B3,DT)
      CALL INTEMP(I,J,K,IN,JN,KN,T,TD,TLL,TMM)
      CALL OUTEMP(I,J,K,IN,JN,KN,TD)
      CALL OLDT(I,J,K,IN,JN,KN,T,TP)
      CALL OLDT(I,J,K,IN,JN,KN,TD,T)
      CALL TEQB(I,J,K,IN,JN,KN,T,MAR)
      CALL DENSTY(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,A,B,C,MAR,MRH,T,TW,
     CRO,ROW,RREF,TREF)
      DO 40 I=1,IN
      DO 40 J=1,JN
      W(I,J,1)=WD(I,J,1)
40    CONTINUE
5     CONTINUE
      CALL RWRH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,HX,HY,
     CDX,DY,DZ,MRH,WRH)
      CALL RWR(I,J,K,IN,JN,KN,U,V,W,WR,HI,HX,HY,DZ,MAR)
      CALL STORE2(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,
     CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,
     CTAV,AKT,CB,CW,A,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,TAMB,TTOT)
      CALL PRPARA(AI,AH,AV,AP,DX,DY,DZ,DT,DL2,MAXIT,EPS,OMEGA,
```

```
     CARBP,TAUX,TAUY,TTOT,MAR,MRH,IN,JN,IWN,JWN)
      CALL TPRIN1(TAI,TAH,TAV,CB,CW,AKT,TREF,RREF,EUL,A,B,C,TE,TO)
      CALL PRITEX(ITN,EX)
      CALL PRPINT(IW,JW,IWN,JWN,P)
      CALL PRUV(I,J,K,IN,JN,KN,U,V)
      CALL PRWH(IW,JW,K,IWN,JWN,KN,WRH)
      CALL TPRIN2(I,J,K,IN,JN,KN,T,RO,TREF)
6     CONTINUE
      END
```

9.1.4 TMAIN2 (Main Program for Near Field)

This is the same as TMAIN 1, except, this program is to be used when the depth is variable. The surbourtine GRADS1 computes slopes of the botton in x and y directions respectively.

```
N*DOC.=TMAIN2
   1              PARAMETER IN=18,JN=20,KN=5,IWN=17,JWN=19
   2              DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IWN,JWN,KN),
   3             CWR(IN,JN,KN),WRH(IWN,JWN,KN),P(IWN,JWN),D(IN,JN,KN),E(IN,JN,KN),
   4             CWHLDT(IWN,JWN),XINT(IN,JN),YINT(IN,JN),H(IN,JN,KN),G(IN,JN,KN),
   5             CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN),FH(IWN,JWN),
   6             CDPSX(IN,JN),DPSY(IN,JN)
   7              DIMENSION A3(KN)
   8              DIMENSION T(IN,JN,KN),TP(IN,JN,KN),TD(IN,JN,KN),RO(IN,JN,KN),
   9             CRINTX(IN,JN,KN),RINTY(IN,JN,KN),WD(IN,JN,KN)
  10              DIMENSION TW(IWN,JWN,KN),ROW(IWN,JWN,KN),JX(JN),II(IN)
  11              INM1=IN-1
  12              READ 1, IRUN
  13              READ 1, LN
  14              READ 1, LLN
  15        1     FORMAT (I5)
  16              READ 2, VVIS,ABR
  17              A3(1)=VVIS
  18              A3(2)=VVIS
  19              A3(3)=VVIS
  20              A3(4)=VVIS
  21              A3(5)=VVIS
  22              B3=VVIS
  23              READ 2, AI,AH,AV,AP
  24              READ 2, EPS,MAXIT,OMEGA,ARBP
  25              READ 2, DX,DY,DZ
  26              READ 2, TAI,TAH,TAV
  27              READ 2, A,B,C
  28              READ 2, TO
  29              READ 2, AKT,EUL,CW,CB
  30              READ 2,AA,BB,CC
  31              READ 2,TLL,TMM
  32        2     FORMAT ()
  33              DL2=DX*DX
  34              TREF=TO
  35              RREF=A+B*TO+C*TO*TO
  36              IF(IRUN.GT.0) GO TO 3
  37              CALL READ3(I,J,IN,JN,IW,JW,IWN,JWN,MAR,MRH)
  38              CALL INITIA(IN,JN,KN,IWN,JWN,U,V,W,WH,D,E,
  39             CP,I,J,K,IW,JW,ARBP)
  40              CALL INITIT(I,J,K,IN,JN,KN,IW,JW,IWN,JWN,A,B,C,T,RO,MAR,MRH,TREF,
  41             CRREF,TW,ROW,TO)
  42              CALL IRDATA(I,J,K,IN,JN,KN,T,TREF,II)
  43              CALL HEIGH1(I,J,K,IN,JN,KN,HI,HX,HY,CC,JX)
  44              CALL GRADS1(IN,JN,KN,IWN,JWN,HI,HX,HY,MAR,MRH,DX,DY)
  45              CALL INLET(I,J,K,IN,JN,KN,V,G,AA,BB)
  46              CALL INTEMP(I,J,K,IN,JN,KN,T,TD,TLL,TMM)
  47              CALL CURNT(I,J,K,IN,JN,KN,U,V,D,E,H,G)
  48              GO TO 4
  49        3     CONTINUE
  50              CALL READ2(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,HI,
  51             CMAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT
  52             C,CB,CW,A,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,TAMB,TTOT)
  53              CALL INLET(I,J,K,IN,JN,KN,V,G,AA,BB)
  54              CALL INTEMP(I,J,K,IN,JN,KN,T,TD,TLL,TMM)
  55        4     CONTINUE
  56              READ 2, TAMB
```

```
      TE=(TAMB-TREF)/TREF
      READ 2,TAUX,TAUY
      READ 2, DT
      DO 6 LL=1,LLN
      DO 5 L=1,LN
      TTOT=TTOT+DT
      CALL ERROR(IWN,JWN,IW,JW,DT,WH,WHLDT,KN,MRH)
      CALL WHTOP(IW,JW,IWN,JWN,KN,WH,K,MRH)
      CALL WHATIJ(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
      CALL INTE(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,MAR,XINT,YINT,A3,AI,
     CAH,AV,TAUX,TAUY,DX,DY,DZ,D,E,DT,DPSX,DPSY,AP)
      CALL CORINT(I,J,K,IN,JN,KN,ABR,U,V,XINT,YINT,DZ,HI,MAR)
      CALL ROINTX(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,
     CMAR,RINTX,HX,XINT)
      CALL ROINTY(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,MAR,
     CRINTY,HY,YINT)
      CALL DPSXY(I,J,IN,JN,IW,JW,IWN,JWN,DPSX,DPSY,P,DX,DY,MAR)
      CALL FORCE(I,J,IW,JW,XINT,YINT,WHLDT,DX,DY,HI,HX,HY,MRH,
     CDPSX,DPSY,FH,AP,IN,JN,IWN,JWN,RINTX,RINTY,U,V,EUL,ABR,MAR,KN)
      CALL PRE2(EPS,MAXIT,IN,JN,P,ITN,DPSX,DPSY,FH,DL2,OMEGA,
     CMRH,I,J,K,IW,JW,DX,DY,EX,IWN,JWN,ARBP)
      CALL UVT(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,D,E,H,G,DX,DY,DZ,
     CRINTX,RINTY,EUL,W,DT,AI,AP,AH,AV,A3,HI,HX,HY,P,MAR)
      CALL UVTOP(H,G,TAUX,TAUY,I,J,K,DZ,IN,JN,KN,HI,MAR)
      CALL OTVELS(I,J,K,IN,JN,KN,H,G)
      CALL OLDUV(I,J,K,IN,JN,KN,U,V,D,E)
      CALL OLDUV(I,J,K,IN,JN,KN,H,G,U,V)
      CALL RWH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,DX,DY,DZ,MRH)
      CALL WHATIJ(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
15    CONTINUE
      DO 20 I=1,IN
      DO 20 J=1,JN
      WD(I,J,1)=W(I,J,1)
20    CONTINUE
      DO 30 I=1,IN
      DO 30 J=1,JN
      W(I,J,1)=0.0
30    CONTINUE
      CALL TEMI4(I,J,K,IN,JN,KN,U,V,T,TD,CX,
     CCB,
     CDY,DZ,W,DT,TAI,TAH,TAV,B3,HI,HX,HY,MAR,AKT,TREF,TAMB)
      CALL TEMB2(I,J,K,IN,JN,KN,TD,DX,DY,DZ,MAR,CB,HI,AKT,CW,TAMB,
     CHX,HY,T,TREF,TAV,TAI,TAH,B3,DT)
      CALL INTEMP(I,J,K,IN,JN,KN,T,TD,TLL,TMM)
      CALL OUTEMP(I,J,K,IN,JN,KN,TD)
      CALL OLDT(I,J,K,IN,JN,KN,T,TP)
      CALL OLDT(I,J,K,IN,JN,KN,TD,T)
      CALL TEQB(I,J,K,IN,JN,KN,T,MAR)
      CALL DENSTY(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,A,B,C,MAR,MRH,T,TW,
     CRO,ROW,RREF,TREF)
      DO 40 I=1,IN
      DO 40 J=1,JN
      W(I,J,1)=WD(I,J,1)
40    CONTINUE
5     CONTINUE
      CALL RWRH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,HX,HY,
     CDX,DY,DZ,MRH,WRH)
```

```
      CALL RWR(I,J,K,IN,JN,KN,U,V,W,WR,HI,HX,HY,DZ,MAR)
      CALL STORE2(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,
     CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,
     CTAV,AKT,CB,CW,A,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,TAMB,TTOT)
      CALL PRPARA(AI,AH,AV,AP,DX,DY,DZ,DT,DLZ,MAXIT,EPS,OMEGA,
     CARBP,TAUX,TAUY,TTOT,MAR,MRH,IN,JN,IWN,JWN)
      CALL INTEMP(I,J,K,IN,JN,KN,T,TO,TLL,TMM)
      CALL CALM5(I,J,K,IN,JN,KN,U,V,D,E,H,G)
      CALL TPRIN1(TAI,TAH,TAV,CB,CW,AKT,TREF,RREF,EUL,A,B,C,TE,TO)
      CALL PRITEX(ITN,EX)
      CALL PRPINT(IW,JW,IWN,JWN,P)
      CALL PRUV(I,J,K,IN,JN,KN,U,V)
      CALL PRWH(IW,JW,K,IWN,JWN,KN,WRH)
      CALL TPRIN2(I,J,K,IN,JN,KN,T,RO,TREF)
6     CONTINUE
      END
```

# MAIN PROGRAMS FOR FAR-FIELD

9.1.5 TMAIN4 (Main Program for Far-Field)

This is a main program. This program is used for initializing velocity and temperature fields for shallow unstratified basins with constant vertical viscosity. The program also fills in the depth matrix. The program fills in the MAR and MRH matrices which define the relative location of computational points in the full and the half grid systems, respectively. The PARAMETER statement defines the size of the computational grid system. The program uses the data element INDATA. First, twelve lines of the data are read in by TMAIN4. The subroutine READ3 reads the MAR and MRH matrices in that sequence. The INITIA subroutine initializes the velocity and pressure fields. The velocity field is set equal to zero and the pressure field is set equal to unity everywhere. The subroutine INITIT initializes the temperature field equal to the reference temperature everywhere in the domain of interest. The subroutine HITEA reads in the depth matrix. The subroutine GRADS computes slopes of the bottom of the basin in x and y directions. The print statements are included in subroutines READ3A and GRADS. The subroutine READ3A prints out MAR and MRH matrices. The subroutine GRADS prints out the depth matrix and the two matrices of x and y slopes. The program stores initialized and computed physical quantities on Unit 8. Element RTM4 is used to provide computer commands necessary to execute this program on UNIVAC-1106 computer.

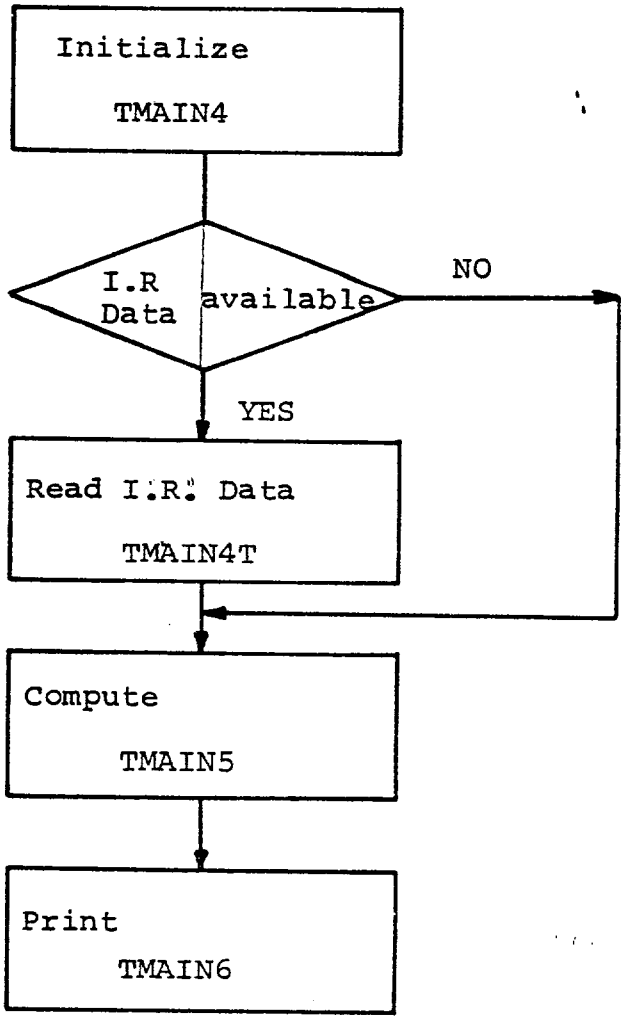
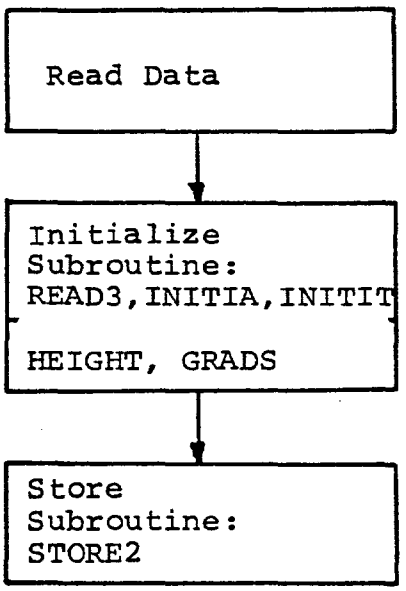



Fig. 9.2 Flow Chart

TMAIN4




TMAIN5

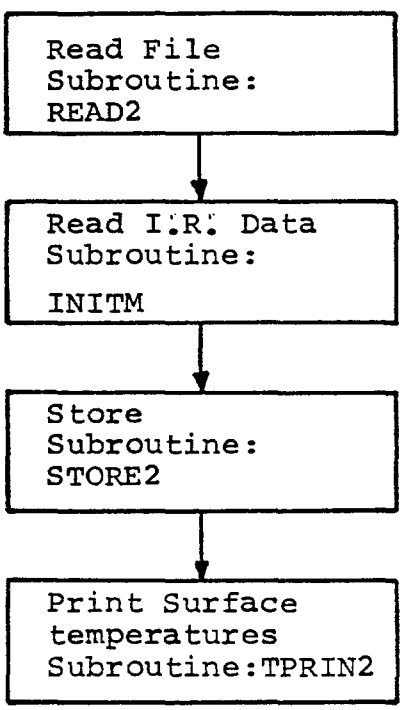



Fig. 9.3 Flow Chart

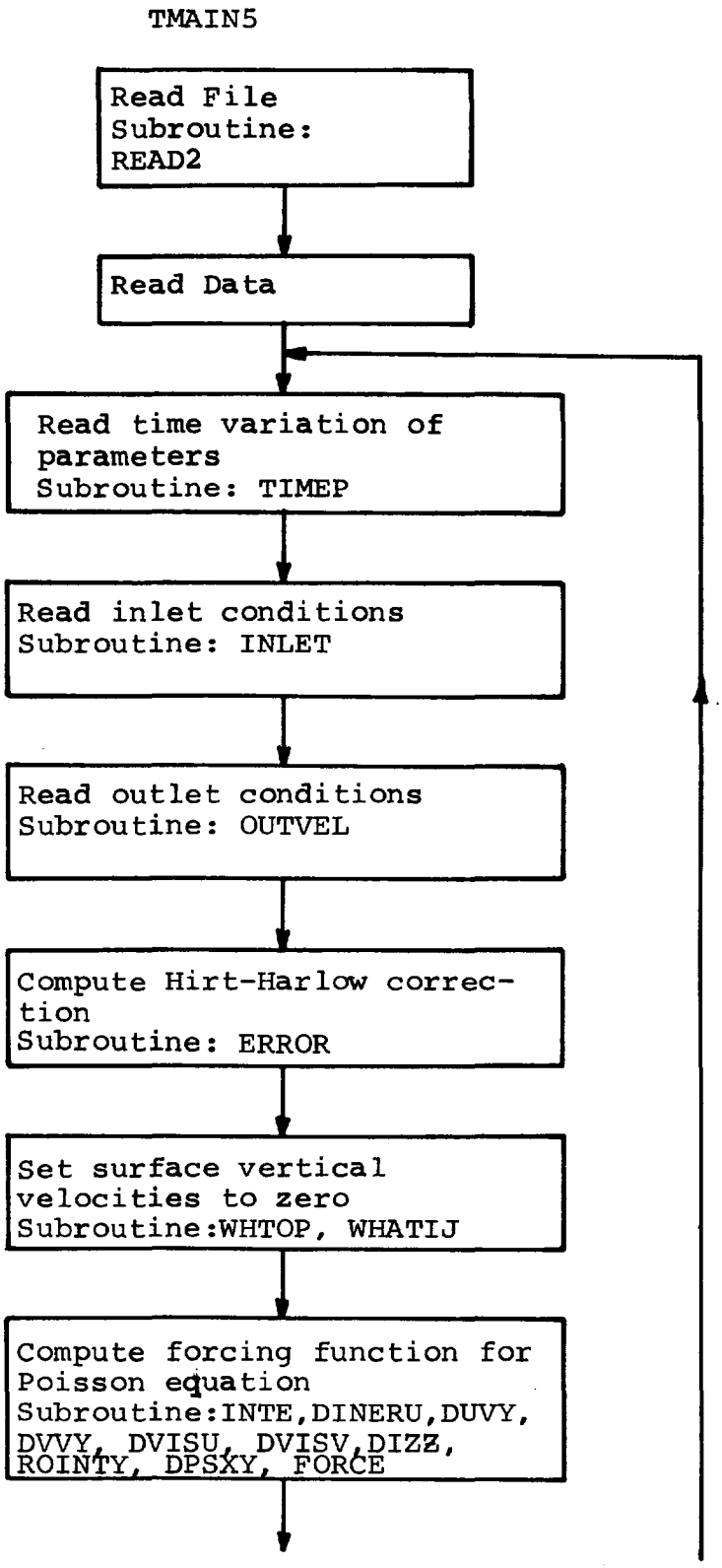



Fig. 9.4 Flow Chart

(Continued on next page)

TMAIN5 continued

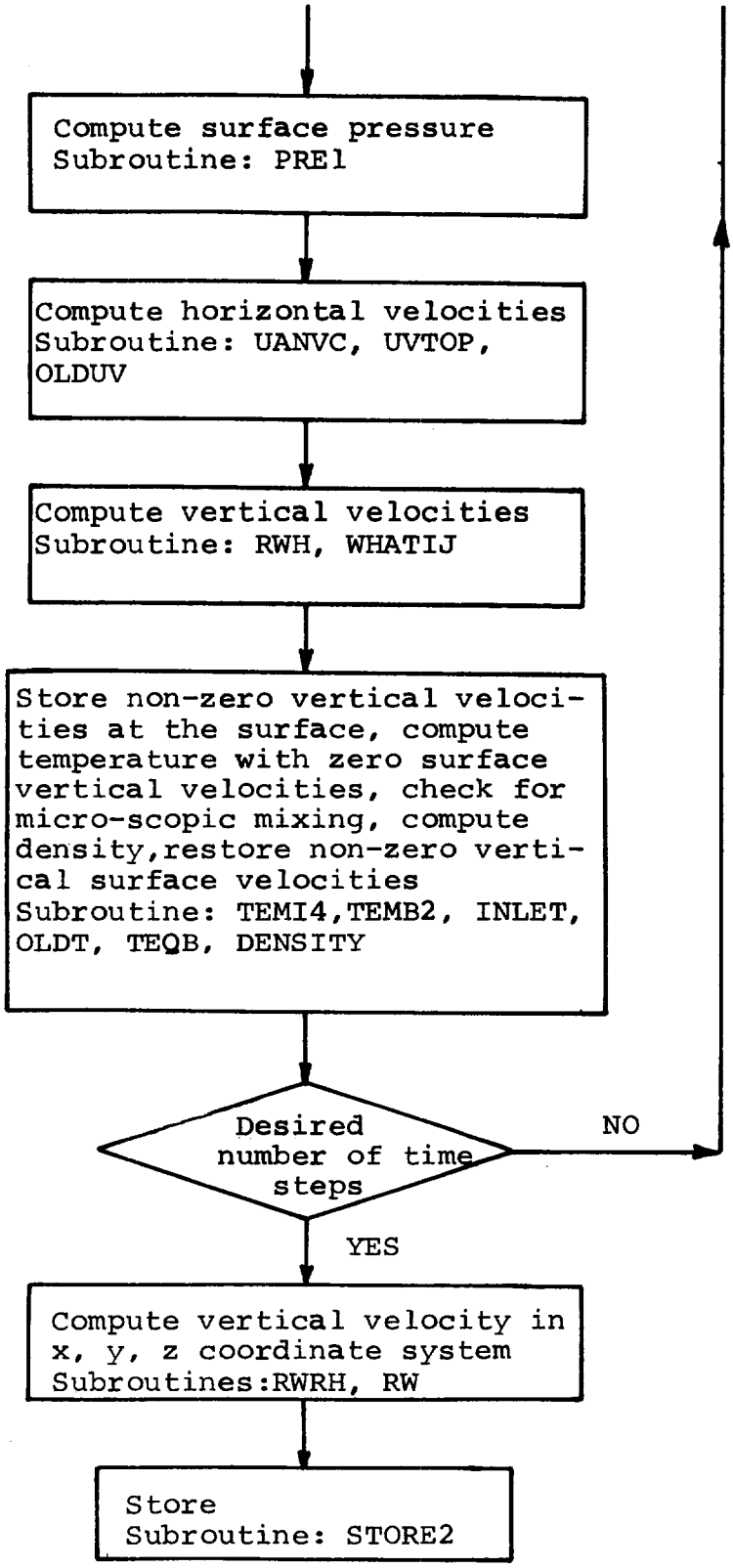



Fig. 3.5 Flow Chart (contd)

```
H*TPFS.TMAIN4
  1              PARAMETER IN=29,JN=13,KN=6,IWN=28,JWN=12
  2              DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IWN,JWN,KN),
  3             CWR(IN,JN,KN),WRH(IWN,JWN,KN),P(IWN,JWN),D(IN,JN,KN),E(IN,JN,KN),
  4             CWHLDT(IWN,JWN),XINT(IN,JN),YINT(IN,JN),H(IN,JN,KN),G(IN,JN,KN),
  5             CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN),FH(IWN,JWN),
  6             CDPSX(IN,JN),DPSY(IN,JN)
  7              DIMENSION A3(KN)
  8              DIMENSION T(IN,JN,KN),TD(IN,JN,KN),RO(IN,JN,KN),
  9             CRINTX(IN,JN,KN),RINTY(IN,JN,KN),WD(IN,JN,KN)
 10              INM1=IN-1
 11              READ 1, IRUN
 12              READ 1, LN,LLN
 13        1     FORMAT (16I5)
 14              READ 1,NDTGR
 15              READ 2, VVIS,ABR
 16              A3(1)=VVIS
 17              A3(2)=VVIS
 18              A3(3)=VVIS
 19              A3(4)=VVIS
 20              A3(5)=VVIS
 21              A3(6)=VVIS
 22              B3=VVIS
 23              READ 2, AI,AH,AV,AP
 24              READ 2, EPS,MAXIT,OMEGA,ARBP
 25              READ 2, DX,DY,DZ
 26              READ 2,CC
 27              READ 2,TAUXM,TAUYM
 28              READ 2,DT
 29              READ 2, TAI,TAH,TAV
 30              READ 2, A,B,C
 31              READ 2, TO
 32              READ 2, AKT,EUL,CW,CB
 33              READ 2,TAMB,TINM
 34              READ 2,IIN,JIN,IOUT,JOUT,UINM,VINM
 35        2     FORMAT ()
 36              DL2=DX*DX
 37              TREF=TO
 38              RREF=A+B*TO+C*TC*TO
 39              IF (IRUN.EQ.0) GO TO 15
 40              IF (IRUN.EQ.1) GO TO 16
 41              IF (IRUN.EQ.2) GO TO 3
 42        15    CONTINUE
 43              GO TO 1000
 44              REWIND 9
 45              READ (9) ((MAR(I,J),I=1,IN),J=1,JN),
 46             C((MRH(IW,JW),IW=1,IWN),JW=1,JWN),
 47             C((HI(I,J),I=1,IN),J=1,JN),
 48             C((HX(I,J),I=1,IN),J=1,JN),
 49             C((HY(I,J),I=1,IN),J=1,JN)
 50              REWIND 9
 51        1000  CONTINUE
 52              CALL READ3(I,J,IN,JN,IW,JW,IWN,JWN,MAR,MRH)
 53              CALL INITIA(IN,JN,KN,IWN,JWN,U,V,W,WH,D,E,P,I,J,K,IW,JW,ARBP)
 54              CALL INITIT(I,J,K,IN,JN,KN,IW,JW,IWN,JWN,A,B,C,T,RO,MAR,MRH,TREF,
 55             CRREF,TO)
 56              CALL HEIGHT(I,J,K,IN,JN,KN,HI,HX,HY,CC)
```

```
57              CALL GRADS(IN,JN,KN,IWN,JWN,HI,HX,HY,MAR,MRH,DX,DY)
58              TTOT=0.0
59              ITN=0
60              EX=0.
61              GO TO 4
62        16    CONTINUE
63        3     CONTINUE
64        4     CONTINUE
65              CALL STORE2(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,
66             CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,
67             CTAV,AKT,CB,CW,A,B,C,EUL,T,RO,TE,RREF,TREF,TO,TAMB,TTOT,
68             CITN,EX)
69              CALL TPRIN9(I,J,K,IN,JN,KN,T,RO,TREF,MAR)
70        6     CONTINUE
71              END
```

9.1.6 TMAIN4T (Main Program for Far Field)

This is a main program. This program is used for shallow unstratified basins with constant vertical viscosity. This program is used to update the initial temperature field from a constant value everywhere as defined by TMAIN4 to a better defined temperature field. The subroutine READ2A reads in the values of physical quantities stored by TMAIN4. The read-in unit is numbered 7. The subroutine INITM reads in the surface temperatures defined as data element ITPK1. Such temperatures are obtained from infrared scanning of the water surface. The temperatures below the surface are computed from a specified temperature gradient in the vertical direction. Computations for temperatures below the surface are also made in subroutine INITM. The subroutine STOR2A writes out the updated values on unit #8. The subroutine TPRIN2 prints out the updated temperature field. The program also sets the time count to zero. The element RTM4T provides with the computer commands necessary to execute program TMAIN4T on UNIVAC-1106 computer.

```
1*TPFS.TMAIN4T
  1              PARAMETER IN=29,JN=13,KN=6,IWN=28,JWN=12
  2              PARAMETER NTL=14,NTLV=2
  3              DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IWN,JWN,KN),
  4             CWR(IN,JN,KN),WRH(IWN,JWN,KN),P(IWN,JWN),D(IN,JN,KN),E(IN,JN,KN),
  5             CWHLDT(IWN,JWN),XINT(IN,JN),YINT(IN,JN),H(IN,JN,KN),G(IN,JN,KN),
  6             CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN),FH(IWN,JWN),
  7             CDPSX(IN,JN),DPSY(IN,JN)
  8              DIMENSION A3(KN),UV(JN),THETA(JN),AMINT(NTL,NTLV)
  9              DIMENSION T(IN,JN,KN),TD(IN,JN,KN),RO(IN,JN,KN),
 10             CRINTX(IN,JN,KN),RINTY(IN,JN,KN),WD(IN,JN,KN)
 11        3     CONTINUE
 12              CALL READ2(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,HI,
 13             CMAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT
 14             C,CB,CW,A,B,C,EUL,T,RO,TE,RREF,TREF,TO,TAMB,TTOT,ITN,EX)
 15        4     CONTINUE
 16              READ 2,((AMINT(N,L),L=1,NTLV),N=1,NTL)
 17        2     FORMAT ( )
 18              CALL INITIT(I,J,K,IN,JN,KN,IW,JW,IWN,JWN,A,B,C,T,RO,MAR,MRH,TREF,
 19             CRREF,TO,AMINT,HI,NTL,NTLV)
 20              CALL INITM(I,J,K,IN,JN,KN,IW,JW,IWN,JWN,A,B,C,T,RO,MAR,MRH,TREF,
 21             CRREF,TO,HI)
 22              TTOT=0.000
 23              CALL STORE2(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,
 24             CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,
 25             CTAV,AKT,CB,CW,A,B,C,EUL,T,RO,TE,RREF,TREF,TO,TAMB,TTOT,
 26             CITN,EX)
 27              CALL TPRIN9(I,J,K,IN,JN,KN,T,RO,TREF,MAR)
 28        6     CONTINUE
 29              END
```

9.1.7 TMAIN5 (Main Program for Far Field)

This is a main program. This is the program which performs most of the computations. Solutions are propagated in time for both the velocity field and the temperature field. Values stored by TMAIN4T are read in by subroutine READ2A from Unit #7. First twelve lines of data element INDATA (defined as data element INDATA5) are used to provide with the values for basic parameters. The subroutine INLETAimposes the inlet and outlet velocities and temperatures onto the computational domain. INLETA subroutine reads lines 13 through the last line of data element INDATA5. The subroutine ERROR computes the contribution of nonzero surface vertical velocities to the rigid lid pressure. The subroutines WHTOP and WHATIJ set the surface vertical velocities to zero. The subroutines INTE partially evaluates the forcing function in the Poisson's equation for rigid-lid pressure. The subroutine CORINT adds the contribution of Coriolis force to the values computed by INTE. The subroutines ROINTX and ROINTY add the contribution of bouyancy to the terms in the forcing function. The subroutine DPSXY computes the pressure gradients at the surface along the boundary for use in the solution of Poisson's equation for rigid lid pressure. The subroutine FORCE combines all terms of the forcing function. The subroutine PRE1 computes the rigid lid pressure field by iteration. The subroutine UVT uses the $\alpha$ and $\beta$ momentum equations to evaluate u and v at grid points below the surface. The subroutine UANVC add the contribution of Coriolis force to u and v. The subroutine UVTOP computes the surface velocities from the velocities below the surface by

utilizing the specified velocity gradient due to wind. The subroutine OLDUV updates the values of u and v. The subroutines RWH and WHATIJ compute the vertical velocity, the continuity equation is used in the process. The nonzero surface vertical velocities are then saved as WD. The surface vertical velocities W are set equal to zero before going into the energy equation. The subroutine TEMI4 computes temperatures at the interior points. The subroutine TEMB2 computes temperatures at the boundary points. The subroutine OLDT updates the values of temperature. The subroutine TEQB checks for thermal instability and mixes the water to create stable temperature field. The subroutine DENSTY computes the new density field from the new temperature field using the equation of state. The nonzero vertical surface velocities are now restored. The subroutines RWRH and RWR compute the vertical velocities in the x-y-z coordinates from the values in $\alpha, \beta, \gamma$ coordinates. The subroutine STOR2A stores the updated values on unit #8. The element RTM5 provides, the computer commands, necessary to execute program TMAIN5 on UNIVAC-1106 computer.

```
H*TPFS.TMAINS
  1              PARAMETER IN=29,JN=13,KN=6,IWN=28,JWN=12
  2              DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IWN,JWN,KN),
  3             CWR(IN,JN,KN),WRH(IWN,JWN,KN),P(IWN,JWN),D(IN,JN,KN),E(IN,JN,KN),
  4             CWHLDT(IWN,JWN),XINT(IN,JN),YINT(IN,JN),H(IN,JN,KN),G(IN,JN,KN),
  5             CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN),FH(IWN,JWN),
  6             CDPSX(IN,JN),DPSY(IN,JN)
  7              DIMENSION A3(KN)
  8              DIMENSION T(IN,JN,KN),TD(IN,JN,KN),RO(IN,JN,KN),
  9             CRINTX(IN,JN,KN),RINTY(IN,JN,KN),WD(IN,JN,KN)
 10              CALL READ2(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,HI,
 11             CMAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT
 12             C,CB,CW,A,B,C,EUL,T,RO,TE,RREF,TREF,TO,TAMB,TTOT,ITN,EX)
 13              INM1=IN-1
 14              READ 1, IRUN
 15              READ 1, LN,LLN
 16        1     FORMAT (16I5)
 17              READ 1,NDTGR
 18              READ 2, VVIS,ABR
 19              A3(1)=VVIS
 20              A3(2)=VVIS
 21              A3(3)=VVIS
 22              A3(4)=VVIS
 23              A3(5)=VVIS
 24              A3(6)=VVIS
 25              B3=VVIS
 26              READ 2, AI,AH,AV,AP
 27              READ 2, EPS,MAXIT,OMEGA,ARBP
 28              READ 2, DX,DY,DZ
 29              READ 2,CC
 30              READ 2,TAUXM,TAUYM
 31              READ 2,DT
 32              READ 2, TAI,TAH,TAV
 33              READ 2, A,B,C
 34              READ 2, TO
 35              READ 2, AKT,EUL,CW,CB
 36              READ 2,TAMB,TINM
 37              READ 2,IIN,JIN,IOUT,JOUT,UINM,VINM
 38        2     FORMAT ()
 39              DL2=DX*DX
 40              TREF=TO
 41              RREF=A+B*TO+C*TO*TO
 42        4     CONTINUE
 43              TE=(TAMB-TREF)/TREF
 44              DO 5 L=1,LN
 45              TTOT=TTOT+DT
 46              CALL TIMEP(TAMB,AKT,UINM,VINM,TINM,TAUXM,TAUYM,TTOT,DT)
 47              CALL INLET(I,J,K,IN,JN,KN,U,V,H,G,TTOT,DT,
 48             CNDTGR,TAUXM,TAUYM,TAUX,TAUY,TINM,T,IIN,JIN,UINM,VINM)
 49              CALL OUTVEL(I,J,K,IN,JN,KN,U,V,H,G,MAR,IOUT,JOUT,IIN,JIN)
 50              CALL ERROR(IWN,JWN,IW,JW,DT,WH,WHLDT,KN,MRH)
 51              CALL WHTOP(IW,JW,IWN,JWN,KN,WH,K,MRH)
 52              CALL WHATIJ(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
 53              CALL INTE(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,MAR,XINT,YINT,A3,AI,
 54             CAH,AV,TAUX,TAUY,DX,DY,DZ,D,E,DT,DPSX,DPSY,AP,T,TREF)
 55              CALL CORINT(I,J,K,IN,JN,KN,ABR,U,V,XINT,YINT,DZ,HI,MAR)
 56              CALL ROINTX(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,
```

```
     CMAR,RINTX,HX,XINT)
      CALL ROINTY(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,MAR,
     CRINTY,HY,YINT)
      CALL DPSXY(I,J,IN,JN,IW,JW,IWN,JWN,DPSX,DPSY,P,DX,DY,MAR)
      CALL FORCE(I,J,IW,JW,XINT,YINT,WHLDT,DX,DY,HI,HX,HY,MRH,
     CDPSX,DPSY,FH,AP,IN,JN,IWN,JWN,RINTX,RINTY,U,V,EUL,ABR,MAR,KN)
      CALL PRE1(EPS,MAXIT,IN,JN,P,ITN,DPSX,DPSY,FH,DL2,OMEGA,
     CMRH,I,J,K,IW,JW,DX,DY,EX,IWN,JWN,ARBP)
      CALL UVT(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,D,E,H,G,DX,DY,DZ,
     CRINTX,RINTY,EUL,W,DT,AI,AP,AH,AV,A3,HI,HX,HY,P,MAR,T,TREF)
      CALL UANVC(I,J,K,IN,JN,KN,ABR,DT,U,V,H,G,HI,MAR)
      CALL UVTOP(H,G,TAUX,TAUY,I,J,K,DZ,IN,JN,KN,HI,MAR)
      CALL INLET(I,J,K,IN,JN,KN,U,V,H,G,TTOT,DT,
     CNOTGR,TAUXH,TAUYH,TAUX,TAUY,TINH,T,IIN,JIN,UINH,VINH)
      CALL OUTVEL(I,J,K,IN,JN,KN,U,V,H,G,MAR,IOUT,JOUT,IIN,JIN)
      CALL OLDUV(I,J,K,IN,JN,KN,U,V,D,E)
      CALL OLDUV(I,J,K,IN,JN,KN,H,G,U,V)
      CALL RWH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,DX,DY,DZ,MRH)
      CALL WHATIJ(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
      DO 20 I=1,IN
      DO 20 J=1,JN
      WD(I,J,1)=W(I,J,1)
   20 CONTINUE
      DO 30 I=1,IN
      DO 30 J=1,JN
      W(I,J,1)=0.0
   30 CONTINUE
      CALL TEMI4(I,J,K,IN,JN,KN,U,V,T,TD,DX,CB,DY,DZ,W,DT,TAI,TAH,TAV,
     CB3,HI,HX,HY,MAR,AKT,TREF,TAMB)
      CALL TEMB2(I,J,K,IN,JN,KN,TD,DX,DY,DZ,MAR,CB,HI,AKT,CW,TAMB,HX,
     CHY,T,TREF,TAV,TAI,TAH,B3,DT)
      CALL OLDT(I,J,K,IN,JN,KN,TD,T)
      CALL TSTAB(I,J,K,IN,JN,KN,T,MAR)
      CALL DENSTY(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,A,B,C,MAR,MRH,T,
     CRO,RREF,TREF)
      DO 40 I=1,IN
      DO 40 J=1,JN
      W(I,J,1)=WD(I,J,1)
   40 CONTINUE
    5 CONTINUE
      CALL RWRH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,HX,HY,
     CDX,DY,DZ,MRH,WRH)
      CALL RWR(I,J,K,IN,JN,KN,U,V,W,WR,HI,HX,HY,DZ,MAR)
      CALL STORE2(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,
     CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,
     CTAV,AKT,CB,CW,A,B,C,EUL,T,RO,TE,RREF,TREF,TO,TAMB,TTOT,
     CITN,EX)
      END
!T,S
```

### 9.1.8 TMAIN5T (Main Program for Far Field)

This program simulates temperatures only for unstratified shallow basins with constant vertical eddy viscosity. Steps involved in its execution are similar to those involved in the execution of TMAIN5. This program uses data element INDATA5.

```
$H*TPFS.TMAIN5T
  1              PARAMETER IN=29,JN=13,KN=6,IWN=28,JWN=12
  2              DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IWN,JWN,KN),
  3             CWR(IN,JN,KN),WRH(IWN,JWN,KN),P(IWN,JWN),D(IN,JN,KN),E(IN,JN,KN),
  4             CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN)
  5              DIMENSION G(IN,JN,KN),H(IN,JN,KN)
  6              DIMENSION A3(KN)
  7              DIMENSION T(IN,JN,KN),TD(IN,JN,KN),RO(IN,JN,KN),WD(IN,JN,KN)
  8              CALL READ2(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,HI,
  9             CMAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT
 10            -C,CB,CW,A,B,C,EUL,T,PO,TE,RREF,TREF,TO,TAMB,TTOT,ITN,EX)
 11              INM1=IN-1
 12              READ 1, IRUN
 13              READ 1, LN,LLN
 14        1     FORMAT (16I5)
 15              READ 1,NDTGR
 16              READ 2, VVIS,ABR
 17              A3(1)=VVIS
 18              A3(2)=VVIS
 19              A3(3)=VVIS
 20              A3(4)=VVIS
 21              A3(5)=VVIS
 22              A3(6)=VVIS
 23              B3=VVIS
 24              READ 2, AI,AH,AV,AP
 25              READ 2, EPS,MAXIT,OMEGA,ARBP
 26              READ 2, DX,DY,DZ
 27              READ 2,CC
 28              READ 2,TAUXM,TAUYM
 29              READ 2,DT
 30              READ 2, TAI,TAH,TAV
 31              READ 2, A,B,C
 32              READ 2, TO
 33              READ 2, AKT,EUL,CW,CB
 34              READ 2,TAMB,TINM
 35              READ 2,IIN,JIN,IOUT,JOUT,UINM,VINM
 36        2     FORMAT ()
 37              DL2=DX*DX
 38              TREF=TO
 39              RREF=A+B*TO+C*TO*TO
 40        4     CONTINUE
 41              DO 20 I=1,IN
 42              DO 20 J=1,JN
 43              WD(I,J,1)=W(I,J,1)
 44        2C    CONTINUE
 45              DO 30 I=1,IN
 46              DO 30 J=1,JN
 47              W(I,J,1)=0.0
 48        3C    CONTINUE
 49              DO 6 LL=1,LLN
 50              TTOT=TTOT+DT
 51              CALL TIMEP(TAMB,AKT,UINM,VINM,TINM,TAUXM,TAUYM,TTOT,DT)
 52              CALL INLET(I,J,K,IN,JN,KN,U,V,H,G,TTOT,DT,
 53             CNDTGR,TAUXM,TAUYM,TAUX,TAUY,TINM,T,IIN,JIN,UINM,VINM)
 54              CALL OUTVEL(I,J,K,IN,JN,KN,U,V,H,G,MAR,IOUT,JOUT,IIN,JIN)
 55              CALL TEMI4(I,J,K,IN,JN,KN,U,V,T,TD,DX,CB,DY,DZ,W,DT,TAI,TAH,TAV,
 56             CB3,HI,HX,HY,MAR,AKT,TREF,TAMB)
```

```
57              CALL TEMB2(I,J,K,IN,JN,KN,TD,DX,DY,DZ,MAR,CB,HI,AKT,CW,TAMB,HX,
58             CHY,T,TREF,TAV,TAI,TAH,B3,DT)
59              CALL OLDT(I,J,K,IN,JN,KN,TD,T)
60              CALL TSTAB(I,J,K,IN,JN,KN,T,MAR)
61        6      CONTINUE
62              CALL DENSTY(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,A,B,C,MAP,MRH,T,
63             CR0,RREF,TREF)
64              DO 40 I=1,IN
65              DO 40 J=1,JN
66              W(I,J,1)=WD(I,J,1)
67        40    CONTINUE
68              CALL STORE2(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,
69             CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,
70             CTAV,AKT,CB,CW,A,B,C,EUL,T,R0,TE,RREF,TREF,TO,TAMB,TTOT,
71             CITN,EX)
72              END
```

9.1.9 TMAIN5V (Main Program for Far Field)

This program simulates velocities only for unstratified shallow basins with constant vertical eddy viscosity. Steps involved in its execution are similar to those involved in the execution of TMAIN5. This program uses data element INDATA5.

```
H*TPFS.TMAIN5V
  1              PARAMETER IN=29,JN=13,KN=6,IWN=28,JWN=12
  2              DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IWN,JWN,KN),
  3             CWR(IN,JN,KN),WRH(IWN,JWN,KN),P(IWN,JWN),D(IN,JN,KN),E(IN,JN,KN),
  4             CWHLDT(IWN,JWN),XINT(IN,JN),YINT(IN,JN),H(IN,JN,KN),G(IN,JN,KN),
  5             CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN),FH(IWN,JWN),
  6             CDPSX(IN,JN),DPSY(IN,JN)
  7              DIMENSION A3(KN)
  8              DIMENSION T(IN,JN,KN),RO(IN,JN,KN),
  9             CRINTX(IN,JN,KN),RINTY(IN,JN,KN)
 10              CALL READ2(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,HI,
 11             CMAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT
 12             C,CB,CW,A,B,C,EUL,T,PO,TE,RREF,TREF,TO,TAMB,TTOT,ITN,EX)
 13              INM1=IN-1
 14              READ 1, IRUN
 15              READ 1, LN,LLN
 16        1     FORMAT (16I5)
 17              READ 1,NDTGR
 18              READ 2, VVIS,ABR
 19              A3(1)=VVIS
 20              A3(2)=VVIS
 21              A3(3)=VVIS
 22              A3(4)=VVIS
 23              A3(5)=VVIS
 24              A3(6)=VVIS
 25              B3=VVIS
 26              READ 2, AI,AH,AV,AP
 27              READ 2, EPS,MAXIT,OMEGA,ARBP
 28              READ 2, DX,DY,DZ
 29              READ 2,CC
 30              READ 2,TAUXM,TAUYM
 31              READ 2,DT
 32              READ 2, TAI,TAH,TAV
 33              READ 2, A,B,C
 34              READ 2, TO
 35              READ 2, AKT,EUL,CW,CB
 36              READ 2,TAMB,TINM
 37              READ 2,IIN,JIN,IOUT,JOUT,UINM,VINM
 38        2     FORMAT ()
 39              DL2=DX*DX
 40              TREF=TO
 41              RREF=A+B*TO+C*TO*TO
 42        4     CONTINUE
 43              TE=(TAMB-TREF)/TREF
 44              DO 5 L=1,LN
 45              TTOT=TTOT+DT
 46              CALL TIMEP(TAMB,AKT,UINM,VINM,TINM,TAUXM,TAUYM,TTOT,DT)
 47              CALL INLET(I,J,K,IN,JN,KN,U,V,H,G,TTOT,DT,
 48             CNDTGR,TAUXM,TAUYM,TAUX,TAUY,TINM,T,IIN,JIN,UINM,VINM)
 49              CALL OUTVEL(I,J,K,IN,JN,KN,U,V,H,G,MAR,IOUT,JOUT,IIN,JIN)
 50              CALL ERRCR(IWN,JWN,IW,JW,DT,WH,WHLDT,KN,MRH)
 51              CALL WHTOP(IW,JW,IWN,JWN,KN,WH,K,MRH)
 52              CALL WHATIJ(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
 53              CALL INTE(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,MAR,XINT,YINT,A3,AI,
 54             CAH,AV,TAUX,TAUY,DX,DY,DZ,D,E,DT,DPSX,DPSY,AP,T,TREF)
 55              CALL CORINT(I,J,K,IN,JN,KN,ABR,U,V,XINT,YINT,DZ,HI,MAR)
 56              CALL ROINTX(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,
```

```
57              CMAR,RINTX,HX,XINT)
58               CALL ROINTY(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,MAR,
59              CRINTY,HY,YINT)
60               CALL DPSXY(I,J,IN,JN,IW,JW,IWN,JWN,DPSX,DPSY,P,DX,DY,MAR)
61               CALL FORCE(I,J,IW,JW,XINT,YINT,WHLDT,DX,DY,HI,HX,HY,MRH,
62              CDPSX,DPSY,FH,AP,IN,JN,IWN,JWN,RINTX,RINTY,U,V,EUL,ABR,MAR,KN)
63               CALL PRE1(EPS,MAXIT,IN,JN,P,ITN,DPSX,DPSY,FH,DL2,OMEGA,
64              CMRH,IVJ,K,IW,JW,DX,DY,EX,IWN,JWN,ARBP)
65               CALL UVT(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,D,E,H,G,DX,DY,DZ,
66              CRINTX,RINTY,EUL,W,DT,AI,AP,AH,AV,A3,HI,HX,HY,P,MAR,T,TREF)
67               CALL UANVC(I,J,K,IN,JN,KN,ABR,DT,U,V,H,G,HI,MAR)
68               CALL UVTOP(H,G,TAUX,TAUY,I,J,K,DZ,IN,JN,KN,HI,MAR)
69               CALL INLET(I,J,K,IN,JN,KN,U,V,H,G,TTOT,DT,
70              CNDTGR,TAUXM,TAUYM,TAUX,TAUY,TINM,T,IIN,JIN,UINM,VINM)
71               CALL OUTVEL(I,J,K,IN,JN,KN,U,V,H,G,MAR,IOUT,JOUT,IIN,JIN)
72               CALL OLDUV(I,J,K,IN,JN,KN,U,V,D,E)
73               CALL OLDUV(I,J,K,IN,JN,KN,H,G,U,V)
74               CALL RWH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,DX,DY,DZ,MRH)
75               CALL WHATIJ(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
76          5    CONTINUE
77               CALL RWRH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,HX,HY,
78              CDX,DY,DZ,MRH,WRH)
79               CALL RWR(I,J,K,IN,JN,KN,U,V,W,WR,HI,HX,HY,DZ,MAR)
80               CALL STORE2(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,
81              CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,
82              CTAV,AKT,CB,CW,A,B,C,EUL,T,RO,TE,RREF,TREF,TO,TAMB,TTOT,
83              CITN,EX)
84               END
```

9.1.10 TMAIN6 (Main Program for Far Field)

This is a main program. This program prints out values stored on Unit #8. Subroutines PRPARA, TPRIN1 and PRITEX print out the basic parameters. The subroutine PRPINT prints out the rigid lid pressure field. The subroutine PRUV prints out the horizontal velocity field. The subroutine PRWH prints out the vertical velocity field. The subroutine TPRIN2 prints out the temperature field. The element RTM6 provides with the computer commands necessary to execute TMAIN6 on UNIVAC -1106 computer.

```
H*TPFS.TMAIN6
  1              PARAMETER IN=29,JN=13,KN=6,IWN=28,JWN=12
  2              DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IWN,JWN,KN),
  3             CWR(IN,JN,KN),WRH(IWN,JWN,KN),P(IWN,JWN),D(IN,JN,KN),E(IN,JN,KN),
  4             CWHLDT(IWN,JWN),XINT(IN,JN),YINT(IN,JN),H(IN,JN,KN),G(IN,JN,KN),
  5             CHI(IN,JN),HX(IN,JN),HY(IN,JN),HAR(IN,JN),HRH(IWN,JWN),FH(IWN,JWN),
  6             CDPSX(IN,JN),DPSY(IN,JN)
  7              DIMENSION A3(KN),UV(JN),THETA(JN)
  8              DIMENSION T(IN,JN,KN),TD(IN,JN,KN),RO(IN,JN,KN),
  9             CRINTX(IN,JN,KN),RINTY(IN,JN,KN),WD(IN,JN,KN)
 10              DIMENSION TW(IWN,JWN,KN),ROW(IWN,JWN,KN)
 11              READ 1,IRUN
 12        1     FORMAT(16I5)
 13              IF (IRUN.EQ.0) GO TO 3
 14              IF (IRUN.EQ.1) GO TO 16
 15              IF (IRUN.EQ.2) GO TO 3
 16        15    CONTINUE
 17              GO TO 4
 18        16    CONTINUE
 19              CALL READ1(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,
 20             CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TTOT)
 21              GO TO 4
 22        3     CONTINUE
 23              CALL READ2(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,HI,
 24             CMAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT
 25             C,CB,CW,A,B,C,EUL,T,RO,TE,RREF,TREF,TO,TAMB,TTOT,ITN,EX)
 26        4     CONTINUE
 27              CALL PRPARA(AI,AH,AV,AP,DX,DY,DZ,DT,DL2,MAXIT,EPS,OMEGA,
 28             CARBP,TAUX,TAUY,TTOT,MAR,MRH,IN,JN,IWN,JWN)
 29              CALL TPRIN1(TAI,TAH,TAV,CB,CW,AKT,TREF,RREF,EUL,A,B,C,TE,TO)
 30              CALL PRITEX(ITN,EX)
 31              CALL PRPINT(IW,JW,IWN,JWN,P)
 32              CALL PRUV(I,J,K,IN,JN,KN,U,V,UV,THETA,MAR)
 33              CALL PRWH(IW,JW,K,IWN,JWN,KN,WRH)
 34              CALL TPRIN2(I,J,K,IN,JN,KN,T,RO,TREF,MAR)
 35        6     CONTINUE
 36              END
```

9.1.11 TMAIN4CB (Main Program for Far Field)

This program is similar to TMAIN4. This program is used to initialize velocity, temperature and pressure fields for a deep stratified cooling lake. This program reads first 27 lines of data element DATAML. Element RTM4B provides control statements to execute this program on UNIVAC-1106.

```
I*DULL(1).TMAIN4CB
  1              PARAMETER IN=29,JN=13,KN=6,IWN=28,JWN=12
  2              PARAMETER NTL=14,NTLV=2
  3              DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IWN,JWN,KN),
  4             CWR(IN,JN,KN),WRH(IWN,JWN,KN),P(IWN,JWN),D(IN,JN,KN),E(IN,JN,KN),
  5            *CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN)
  6              DIMENSION A3(KN),AMINT(NTL,NTLV)
  7              DIMENSION T(IN,JN,KN),RO(IN,JN,KN)
  8              INM1=IN-1
  9              READ 1, LN,LLN
 10        1     FORMAT (16I5)
 11              READ 2, VVIS,ABR
 12              A3(1)=VVIS
 13              A3(2)=VVIS
 14              A3(3)=VVIS
 15              A3(4)=VVIS
 16              A3(5)=VVIS
 17              A3(6)=VVIS
 18              B3=VVIS
 19              READ 2, AI,AH,AV,AP
 20              READ 2, EPS,MAXIT,OMEGA,ARBP
 21              READ 2, DX,DY,DZ
 22              READ 2,CC
 23              READ 2,DT
 24              READ 2, TAI,TAH,TAV
 25              READ 2, A,B,C
 26              READ 2, TO
 27              READ 2, EUL,CW,CB
 28              READ 2,TAMB,AKT,TAUX,TAUY
 29              READ 2,CONS,AVMX,AVMN
 30              READ 2,((AMINT(N,L),L=1,NTLV),N=1,NTL)
 31        2     FORMAT ()
 32              DL2=DX*DX
 33              TREF=TO
 34              RREF=A+B*TO+C*TO*TO
 35              CALL READ3A(I,J,IN,JN,IW,JW,IWN,JWN,MAR,MRH)
 36              CALL HITEA(I,J,K,IN,JN,KN,HI,HX,HY,CC)
 37              CALL INITIA(IN,JN,KN,IWN,JWN,U,V,W,WH,D,E,P,I,J,K,IW,JW,ARBP)
 38              CALL INITB(I,J,K,IN,JN,KN,IW,JW,IWN,JWN,A,B,C,T,RO,MAR,MRH,TREF,
 39             CRREF,TO,AMINT,HI,NTL,NTLV)
 40              CALL GRADS(IN,JN,KN,IWN,JWN,HI,HX,HY,MAR,MRH,DX,DY)
 41              TTOT=0.0
 42              ITN=0
 43              EX=0.
 44              GO TO 4
 45        16    CONTINUE
 46        3     CONTINUE
 47        4     CONTINUE
 48              CALL STOR23(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,
 49             CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,
 50             CTAV,AKT,CB,CW,A,B,C,EUL,T,RO,TE,RREF,TREF,TO,TAMB,TTOT,
 51             CITN,EX)
 52              CALL TPRIN9(I,J,K,IN,JN,KN,T,RO,TREF,MAR)
 53        6     CONTINUE
 54              END
```

9.1.12 TMAIN4TB (Main Program for Far Field)

This program is similar to TMAIN4T. This program specifies measured initial temperatures for a deep stratified cooling lake. This program reads data element ITLK1. Element RTM4TB provides control statements to execute this program on UNIVAC-1106.

```
1*DULL(1).TMAIN4TB
  1              PARAMETER IN=29,JN=13,KN=6,IWN=28,JWN=12
  2              PARAMETER NTL=14,NTLV=2
  3              DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IWN,JWN,KN),
  4             CWR(IN,JN,KN),WRH(IWN,JWN,KN),P(IWN,JWN),D(IN,JN,KN),E(IN,JN,KN),
  5             CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN)
  6              DIMENSION AMINT(NTL,NTLV)
  7        C     DIMENSION T(IN,JN,KN),RO(IN,JN,KN)
  8        3     CONTINUE
  9              CALL READ2B(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,HI,
 10             CMAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT
 11             C,CB,CW,A,B,C,EUL,T,RO,TE,RREF,TREF,TO,TAMB,TTOT,ITN,EX)
 12        4     CONTINUE
 13              READ 2,((AMINT(N,L),L=1,NTLV),N=1,NTL)
 14        2     FORMAT ()
 15              CALL INITB(I,J,K,IN,JN,KN,IW,JW,IWN,JWN,A,B,C,T,RO,MAR,MRH,TREF,
 16             CRREF,TO,AMINT,HI,NTL,NTLV)
 17              CALL INITMB(I,J,K,IN,JN,KN,IW,JW,IWN,JWN,A,B,C,T,RO,MAR,MRH,TREF,
 18             CRREF,TO,HI)
 19              TTOT=0.000
 20              CALL STOR2B(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,
 21             CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,
 22             CTAV,AKT,CB,CW,A,B,C,EUL,T,RO,TE,RREF,TREF,TO,TAMB,TTOT,
 23             CITN,EX)
 24              CALL TPRIN9(I,J,K,IN,JN,KN,T,RO,TREF,MAR)
 25        6     CONTINUE
 26              END
```

9.1.13 TMAIN5B (Main Program for Far Field)

This program is similar to TMAIN5. This program is used to simulate velocities and temperatures in a coupled fashion for a deep stratified cooling lake. This program reads data element DATAML5. Element RTM5B provides control statements to execute this program on UNIVAC-1106.

```
*DULL(1).TMAIN5B
   1              PARAMETER IN=29,JN=13,KN=6,IWN=28,JWN=12
   2              DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IWN,JWN,KN),
   3             CWR(IN,JN,KN),WRH(IWN,JWN,KN),P(IWN,JWN),D(IN,JN,KN),E(IN,JN,KN),
   4             CWHLDT(IWN,JWN),XINT(IN,JN),YINT(IN,JN),H(IN,JN,KN),G(IN,JN,KN),
   5             CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN),FH(IWN,JWN),
   6            *CDPSX(IN,JN),DPSY(IN,JN)
   7              DIMENSION A3(KN)
   8              DIMENSION T(IN,JN,KN),TD(IN,JN,KN),RO(IN,JN,KN),
   9             CRINTX(IN,JN,KN),RINTY(IN,JN,KN),WD(IN,JN,KN)
  10              CALL READ2B(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,HI,
  11             CMAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT
  12             C,CB,CW,A,B,C,EUL,T,RO,TE,RREF,TREF,TO,TAMB,TTOT,ITN,EX)
  13              INM1=IN-1
  14              READ 1, LN,LLN
  15        1     FORMAT (16I5)
  16              READ 2, VVIS,ABP
  17              A3(1)=VVIS
  18              A3(2)=VVIS
  19              A3(3)=VVIS
  20              A3(4)=VVIS
  21              A3(5)=VVIS
  22              A3(6)=VVIS
  23              B3=VVIS
  24              READ 2, AI,AH,AV,AP
  25              READ 2, EPS,MAXIT,OMEGA,ARBP
  26              READ 2, DX,DY,DZ
  27              READ 2,CC
  28              READ 2,DT
  29              READ 2, TAI,TAH,TAV
  30              READ 2, A,B,C
  31              READ 2, TO
  32              READ 2, EUL,CW,CB
  33              READ 2,TAMB,AKT,TAUX,TAUY
  34              READ 2,CONS,AVMX,AVMN
  35        2     FORMAT ()
  36              DL2=DX*DX
  37              TREF=TO
  38              RREF=A+B*TO+C*TO*TO
  39        4     CONTINUE
  40              TE=(TAMB-TREF)/TREF
  41              CALL INLET3(I,J,K,IN,JN,KN,U,V,H,G,T)
  42              DO 5 L=1,LN
  43              CALL ERROR(IWN,JWN,IW,JW,DT,WH,WHLDT,KN,MRH)
  44              CALL WHTOP(IW,JW,IWN,JWN,KN,WH,K,MRH)
  45              CALL WHATIJ(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
  46              CALL INTE8(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,MAR,XINT,YINT,A3,AI,
  47             CAH,AV,TAUX,TAUY,DX,DY,DZ,D,E,DT,DPSX,DPSY,AP,T,TREF
  48             C,CONS,AVMX,AVMN)
  49              CALL CORINT(I,J,K,IN,JN,KN,ABR,U,V,XINT,YINT,DZ,HI,MAR)
  50              CALL ROINTX(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,
  51             CMAR,RINTX,HX,XINT)
  52              CALL ROINTY(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,MAR,
  53             CRINTY,HY,YINT)
  54              CALL DPSXY(I,J,IN,JN,IW,JW,IWN,JWN,DPSX,DPSY,P,DX,DY,MAR)
  55              CALL FORCE(I,J,IW,JW,XINT,YINT,WHLDT,DX,DY,HI,HX,HY,MRH,
  56             CDPSX,DPSY,FH,AP,IN,JN,IWN,JWN,RINTX,RINTY,U,V,EUL,ABR,MAR,KN)
```

```
57              CALL PRE1(EPS,MAXIT,IN,JN,P,ITN,DPSX,DPSY,FH,DL2,OMEGA,
58             CMRH,I,J,K,IW,JW,DX,DY,EX,IWN,JWN,ARUP)
59              CALL UVTB(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,D,E,H,G,DX,DY,DZ,
60             CRINTX,RINTY,EUL,W,DT,AI,AP,AH,AV,A3,HI,HX,HY,P,MAR,T,TREF
61             C,CONS,AVMX,AVMN)
62      C       CALL UANVC(I,J,K,IN,JN,KN,ABR,DT,U,V,H,G,HI,MAR)
63              CALL UVTOP(H,G,TAUX,TAUY,I,J,K,DZ,IN,JN,KN,HI,MAR)
64              CALL OLDUV(I,J,K,IN,JN,KN,U,V,D,E)
65              CALL OLDUV(I,J,K,IN,JN,KN,H,G,U,V)
66              CALL RWH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,DX,DY,DZ,MRH)
67              CALL WHATIJ(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
69              DO 20 I=1,IN
69              DO 20 J=1,JN
70              WD(I,J,1)=W(I,J,1)
71      20      CONTINUE
72              DO 30 I=1,IN
73              DO 30 J=1,JN
74              W(I,J,1)=0.0
75      30      CONTINUE
76              CALL TEMI4B(I,J,K,IN,JN,KN,U,V,T,TD,DX,CB,DY,DZ,W,DT,TAI,TAH,TAV,
77             CB3,HI,HX,HY,MAR,AKT,TREF,TAMB,A3,CONS,AVMX,AVMN)
78              CALL TEMB2B(I,J,K,IN,JN,KN,TD,DX,DY,DZ,MAR,CB,HI,AKT,CW,TAMB,HX,
79             CHY,T,TREF,TAV,TAI,TAH,B3,DT,A3,CONS,AVMX,AVMN)
80              CALL OLDT(I,J,K,IN,JN,KN,TD,T)
81              CALL TSTAB(I,J,K,IN,JN,KN,T,MAR)
82              CALL DENSTB(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,A,B,C,MAR,MRH,T,
83             CRO,RREF,TREF)
84              DO 40 I=1,IN
85              DO 40 J=1,JN
86              W(I,J,1)=WD(I,J,1)
87      40      CONTINUE
88      5       CONTINUE
89              CALL RWRH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,HX,HY,
90             CDX,DY,DZ,MRH,WRH)
91              CALL RWR(I,J,K,IN,JN,KN,U,V,W,WR,HI,HX,HY,DZ,MAR)
92              CALL STOR2B(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,
93             CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,
94             CTAV,AKT,CB,CW,A,B,C,EUL,T,RO,TE,RREF,TREF,TO,TAMB,TTOT,
95             CITN,EX)
96              END
```

9.1.14 TMAIN5VB (Main Program for Far Field)

This program is similar to TMAIN5V. This program is used to simulate velocities only for a deep stratified cooling lake. This program uses data element DATAML5. Element RTM5VB provides control statements to execute this program on UNIVAC-1106.

```
1*DULL(1).TMAINSVB
      1              PARAMETER IN=29,JN=13,KN=6,IWN=28,JWN=12
      2              DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IWN,JWN,KN),
      3             CWR(IN,JN,KN),WRH(IWN,JWN,KN),P(IWN,JWN),D(IN,JN,KN),E(IN,JN,KN),
      4             CWHLDT(IWN,JWN),XINT(IN,JN),YINT(IN,JN),H(IN,JN,KN),G(IN,JN,KN),
      5             CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN),FH(IWN,JWN),
      6            *DPSX(IN,JN),DPSY(IN,JN)
      7              DIMENSION AZ(KN)
      8              DIMENSION T(IN,JN,KN),RO(IN,JN,KN),
      9             CRINTX(IN,JN,KN),RINTY(IN,JN,KN)
     10              CALL READ3B(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,HI,
     11             CMAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT
     12             C,CB,CW,A,B,C,EUL,T,RO,TE,RREF,TREF,TO,TAMB,TTOT,ITN,EX)
     13              INM1=IN-1
     14              READ 1, LN,LLN
     15         1    FORMAT (16I5)
     16              READ 2, VVIS,ABP
     17              A3(1)=VVIS
     18              A3(2)=VVIS
     19              A3(3)=VVIS
     20              A3(4)=VVIS
     21              A3(5)=VVIS
     22              A3(6)=VVIS
     23              B3=VVIS
     24              READ 2, AI,AH,AV,AP
     25              READ 2, EPS,MAXIT,OMEGA,ARBP
     26              READ 2, DX,DY,DZ
     27              READ 2,CC
     28              READ 2,DT
     29              READ 2, TAI,TAH,TAV
     30              READ 2, A,B,C
     31              READ 2, TO
     32              READ 2, EUL,CW,CB
     33              READ 2,TAMB,AKT,TAUX,TAUY
     34              READ 2,CONS,AVMX,AVMN
     35         2    FORMAT ()
     36              DL2=DX*DX
     37              TREF=TO
     38              RREF=A+B*TO+C*TO*TO
     39         4    CONTINUE
     40              TE=(TAMB-TREF)/TREF
     41              CALL INLET3(I,J,K,IN,JN,KN,U,V,H,G,T)
     42              DO 5 L=1,LN
     43              TTOT=TTOT+DT
     44              CALL ERROR(IWN,JWN,IW,JW,DT,WH,WHLDT,KN,MRH)
     45              CALL WHTOP(IW,JW,IWN,JWN,KN,WH,K,MRH)
     46              CALL WHATIJ(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WP,MAR)
     47              CALL INTEB(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,MAR,XINT,YINT,AZ,AI,
     48             CAH,AV,TAUX,TAUY,DX,DY,DZ,D,E,DT,DPSX,DPSY,AP,T,TREF
     49             C,CONS,AVMX,AVMN)
     50              CALL CORINT(I,J,K,IN,JN,KN,ABR,U,V,XINT,YINT,DZ,HI,MAR)
     51              CALL ROINTX(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,
     52             CMAR,RINTX,HX,XINT)
     53              CALL ROINTY(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,MAR,
     54             CRINTY,HY,YINT)
     55              CALL DPSXY(I,J,IN,JN,IW,JW,IWN,JWN,DPSX,DPSY,P,DX,DY,MAR)
     56              CALL FORCE(I,J,IW,JW,XINT,YINT,WHLDT,DX,DY,HI,HX,HY,MRH,
```

```
     57              CDPSX,DPSY,FH,AP,IN,JN,IWN,JWN,RINTX,RINTY,U,V,EUL,ABR,MAR,KN)
     58               CALL PRE1(EPS,MAXIT,IN,JN,P,ITN,DPSX,DPSY,FH,DL2,OMEGA,
     59              CMRH,I,J,K,IW,JW,DX,DY,EX,IWN,JWN,ARSP)
     60               CALL UVTB(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,D,E,H,G,DX,DY,DZ,
     61              CRINTX,RINTY,EUL,W,DT,AI,AP,AH,AV,A3,HI,HX,HY,P,MAR,T,TREF
     62              C,CONS,AVMX,AVMN)
     63               CALL UANVC(I,J,K,IN,JN,KN,ABR,DT,U,V,H,G,HI,MAR)
     64               CALL UVTOP(H,G,TAUX,TAUY,I,J,K,DZ,IN,JN,KN,HI,MAR)
     65               CALL OLDUV(I,J,K,IN,JN,KN,U,V,D,E)
     66               CALL OLDUV(I,J,K,IN,JN,KN,H,G,U,V)
     67               CALL RWH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,DX,DY,DZ,MRH)
     68               CALL WHATIJ(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
     69          5    CONTINUE
     70               CALL RWRH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,HX,HY,
     71              CDX,DY,DZ,MRH,WRH)
     72               CALL RWR(I,J,K,IN,JN,KN,U,V,W,WR,HI,HX,HY,DZ,MAR)
     73               CALL STOR2B(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,
     74              CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,
     75              CTAV,AKT,CB,CW,A,B,C,EUL,T,RG,TE,RREF,TREF,TO,TAMB,TTOT,
     76              CITN,EX)
     77               END
```

### 9.1.15 TMAIN5TB (Main Program for Far Field)

This program is similar to TMAIN5T. This program is used to simulate temperatures only for a deep stratified cooling lake. This program uses data element DATAML5. Element RTM5B provides with the control statements needed to execute this program on UNIVAC-1106.

```
M*DULL(1).TMAINSTB
   1              PARAMETER IN=29,JN=13,KN=6,IWN=28,JWN=12
   2              DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IWN,JWN,KN),
   3             CWR(IN,JN,KN),WRH(IWN,JWN,KN),P(IWN,JWN),D(IN,JN,KN),E(IN,JN,KN),
   4             CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN)
   5              DIMENSION G(IN,JN,KN),H(IN,JN,KN)
   6              DIMENSION A3(KN)
   7              DIMENSION T(IN,JN,KN),TD(IN,JN,KN),RO(IN,JN,KN),WD(IN,JN,KN)
   8              CALL READ2B(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,HI,
   9             CMAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT
  10             C,CB,CW,A,B,C,EUL,T,RO,TE,RREF,TREF,TO,TAMB,TTOT,ITN,EX)
  11              INM1=IN-1
  12              READ 1, LN,LLN
  13         1    FORMAT (16I5)
  14              READ 2, VVIS,ABR
  15              A3(1)=VVIS
  16              A3(2)=VVIS
  17              A3(3)=VVIS
  18              A3(4)=VVIS
  19              A3(5)=VVIS
  20              A3(6)=VVIS
  21              B3=VVIS
  22              READ 2, AI,AH,AV,AP
  23              READ 2, EPS,MAXIT,OMEGA,ARBP
  24              READ 2, DX,DY,DZ
  25              READ 2,CC
  26              READ 2,DT
  27              READ 2, TAI,TAH,TAV
  28              READ 2, A,B,C
  29              READ 2, TO
  30              READ 2, EUL,CW,CB
  31              READ 2,TAMB,AKT,TAUX,TAUY
  32              READ 2,CONS,AVMX,AVMN
  33         2    FORMAT ()
  34              DL2=DX*DX
  35              TREF=TO
  36              RREF=A+B*TO+C*TO*TO
  37         4    CONTINUE
  38              DO 20 I=1,IN
  39              DO 20 J=1,JN
  40              WD(I,J,1)=W(I,J,1)
  41        20    CONTINUE
  42              DO 30 I=1,IN
  43              DO 30 J=1,JN
  44              W(I,J,1)=0.0
  45        30    CONTINUE
  46              CALL INLETB(I,J,K,IN,JN,KN,U,V,H,G,T)
  47              DO 6 LL=1,LLN
  48              TTOT=TTOT+DT
  49              CALL TEMI4B(I,J,K,IN,JN,KN,U,V,T,TD,DX,CB,DY,DZ,W,DT,TAI,TAH,TAV,
  50             CB3,HI,HX,HY,MAR,AKT,TREF,TAMB,A3,CONS,AVMX,AVMN)
  51              CALL TEMP2B(I,J,K,IN,JN,KN,TD,DX,DY,DZ,MAR,CB,HI,AKT,CW,TAMB,HX,
  52             CHY,T,TREF,TAV,TAI,TAH,B3,DT,A3,CONS,AVMX,AVMN)
  53              CALL OLDT(I,J,K,IN,JN,KN,TD,T)
  54              CALL TSTAB(I,J,K,IN,JN,KN,T,MAR)
  55         6     CONTINUE
  56              CALL DENSTB(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,A,B,C,MAR,MRH,T,
```

```
     57              CRO,RREF,TREF)
     58               DO 40 I=1,IN
     59               DO 40 J=1,JN
     60               W(I,J,1)=WD(I,J,1)
     61         40    CONTINUE
     62               CALL STOR2B(U,V,WH,P,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,
     63              CHI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,
     64              CTAV,AKT,CB,CW,A,B,C,EUL,T,RO,TE,RREF,TREF,TO,TAMB,TTOT,
     65              CITN,EX)
     66               END
```

9.1.16 TMAIN6B (Main Program for Far Field)

This program is similar to TMAIN6. This program prints out the results for the simulation of velocity and/or temperature in a deep stratified cooling pond. Element RTM6B provides with the control statements necessary to execute this program on UNIVAC-1106.

```
M*DULL(1).TMAIN6B
    1              PARAMETER IN=29,JN=13,KN=6,IWN=28,JWN=12
    2              DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WH(IWN,JWN,KN),
    3             CWR(IN,JN,KN),WRH(IWN,JWN,KN),P(IWN,JWN),D(IN,JN,KN),E(IN,JN,KN),
    4             CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN)
    5              DIMENSION A3(KN),UV(JN),THETA(JN)
    6              DIMENSION T(IN,JN,KN),RO(IN,JN,KN)
    7              INM1=IN-1
    8              READ 1, LN,LLN
    9        1     FORMAT (16I5)
   10              READ 2, VVIS,ABR
   11              A3(1)=VVIS
   12              A3(2)=VVIS
   13              A3(3)=VVIS
   14              A3(4)=VVIS
   15              A3(5)=VVIS
   16              A3(6)=VVIS
   17              B3=VVIS
   18              READ 2, AI,AH,AV,AP
   19              READ 2, EPS,MAXIT,OMEGA,ARBP
   20              READ 2, DX,DY,DZ
   21              READ 2,CC
   22              READ 2,DT
   23              READ 2, TAI,TAH,TAV
   24              READ 2, A,B,C
   25              READ 2, TC
   26              READ 2, EUL,CW,CB
   27              READ 2,TAMB,AKT,TAUX,TAUY
   28              READ 2,CONS,AVMX,AVMN
   29        2     FORMAT ()
   30              DL2=DX*DX
   31              CALL READ2B(U,V,WH,P,I,J,K,IN,JW,IN,JN,KN,IWN,JWN,D,E,HX,HY,HI,
   32             CMAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT
   33             C,CB,CW,A,B,C,EUL,T,RO,TE,RREF,TREF,TO,TAMB,TTOT,ITN,EX)
   34        4     CONTINUE
   35              CALL PRPARA(AI,AH,AV,AP,DX,DY,DZ,DT,DL2,MAXIT,EPS,OMEGA,
   36             CARBP,TAUX,TAUY,TTOT,MAR,MRH,IN,JN,IWN,JWN)
   37              CALL TPRIN1(TAI,TAH,TAV,CB,CW,AKT,TREF,RREF,EUL,A,B,C,TE,TO)
   38              CALL PRITEX(ITN,EX)
   39              CALL PRPINT(IW,JW,IWN,JWN,P)
   40              CALL PRUVA(I,J,K,IN,JN,KN,U,V,UV,THETA,MAR)
   41              CALL PRWH(IW,JW,K,IWN,JWN,KN,WRH)
   42              CALL TPRINA(I,J,K,IN,JN,KN,T,RO,TREF,MAR)
   43        6     CONTINUE
   44              END
```

# SUBROUTINES FOR NEAR AND FAR-FIELD

## 9.2 Subroutines for the Near Field and Far Field

The subroutines for the Near field and Far field are given in alphabetical order in this section. These subroutines are called by the main programs. The Fortran symbol explanation can be seen in section 3 where they are explained in alphabetical order.

### 9.2.1 CORINT

This program adds integral of Coriolis component to XINT and YINT. XINT and YINT are calculated in the subroutine INTE. XINT and YINT are integral of x and y components on the right hand side of Poisson's equation (Eq 2.17, Vol.1)

```
      SUBROUTINE CORINT(I,J,K,IN,JN,KN,ABR,U,V,XINT,YINT,DZ,HI,MAR)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),XINT(IN,JN),YINT(IN,JN),HI(IN,
     CJN),MAR(IN,JN)
      DO 10 I=1,IN
      DO 10 J=1,JN
      IF (MAR(I,J).LT.11) GO TO 9
      DO 8 K=2,KN
      XINT(I,J)=XINT(I,J)-ABR*HI(I,J)*(V(I,J,K-1)+V(I,J,K))*DZ/2
      YINT(I,J)=YINT(I,J)+ABR*HI(I,J)*(U(I,J,K-1)+U(I,J,K))*DZ/2
    8 CONTINUE
    9 CONTINUE
   10 CONTINUE
      RETURN
      END
```

### 9.2.2 CURNT

This program puts current into the model. This program must be changed depending on the direction and magnitude of the current.

(This subroutine is not used in the sample case, as the current is not considered)

```
*DOC.CURNTS
  1              SUBROUTINE CURNTS(I,J,K,IN,JN,KN,U,V,D,E,H,G)
  2              DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),
  3             CE(IN,JN,KN),H(IN,JN,KN),G(IN,JN,KN)
  4              KNM1=KN-1
  5              DO 10 I=1,IN
  6              DO 10 K=1,KNM1
  7              DO 10 J=2,JN
  8              U(I,J,K)=-0.1
  9              V(I,J,K)=0.0
 10              D(I,J,K)=-0.1
 11              E(I,J,K)=0.0
 12              H(I,J,K)=-0.1
 13              G(I,J,K)=0.0
 14        10    CONTINUE
 15              RETURN
 16              END
```

### 9.2.3 CWXY

This program computes horizontal temperature gradients at the vertical walls in $\alpha$ and $\beta$ directions from the heat flux in x and y directions.

```
#*DOC.CWXY
  1             SUBROUTINE CWXY(CWX,CWY,CW,I,J,K,IN,JN,KN,HI,HX,HY,T,DZ)
  2             DIMENSION T(IN,JN,KN),HI(IN,JN),HX(IN,JN),HY(IN,JN)
  3             IF (K.EQ.1) GO TO 99
  4             IF (K.EQ.KN) GO TO 101
  5             D1TZ=(T(I,J,K+1)-T(I,J,K-1))/(2*DZ)
  6             GO TO 206
  7       99    CONTINUE
  8             D1TZ=(4*T(I,J,K+1)-3*T(I,J,K)-T(I,J,K+2))/(2*DZ)
  9             GO TO 206
 10       101   CONTINUE
 11             D1TZ=(3*T(I,J,K)+T(I,J,K-2)-4*T(I,J,K-1))/(2*DZ)
 12       206   CONTINUE
 13             CWX=CW+(K-1)*DZ*HX(I,J)*D1TZ/HI(I,J)
 14             CWY=CW+(K-1)*DZ*HY(I,J)*D1TZ/HI(I,J)
 15             RETURN
 16             END
```

9.2.4 DENSTA

This subroutine is used for the far field unstratified cooling reservoir. The subroutine is similar to the subroutine DENSTY. The only difference being that the matrix ROW (densities at the half grid points) is eliminated to save computer core space.

```
SKM*DULL(1).DENSTA
      1       C*************************************************************
      2       C         THE FOLLOWING PROGRAM CALCULATES THE DENSITY FIELD FROM
      3       C            THE TEMPERATURE FIELD
      4       C*************************************************************
      5             SUBROUTINE DENSTA(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,A,B,C,
      6            CMAR,MRH,
      7            CT,TW,RO,RREF,TREF)
      8             DIMENSION RO(IN,JN,KN),T(IN,JN,KN)
      9             DIMENSION TW(IWN,JWN,KN)
     10             DIMENSION MAR(IN,JN),MRH(IWN,JWN)
     11             DO 10 I=1,IN
     12             DO 10 J=1,JN
     13             IF (MAR(I,J).EQ.0) GO TO 12
     14             DO 11 K=1,KN
     15             TEM=T(I,J,K)*TREF+TREF
     16             R=A+B*TEM+C*TEM*TEM
     17             RO(I,J,K)=(R-RREF)/RREF
     18       11    CONTINUE
     19       12    CONTINUE
     20       10    CONTINUE
     21             RETURN
     22             END
```

### 9.2.5 DENSTB

This subroutine is used for the far field stratified cooling lake. The subroutine is similar to DENSTY, the only difference being that the matrices TW (temperatures at the half grid points) and ROW (densities at the half grid points) are eliminated to save computer core space.

```
SKM*DULL(1).DENSTB
      1       C*************************************************************
      2       C        THE FOLLOWING PROGRAM CALCULATES THE DENSITY FIELD FROM
      3       C           THE TEMPERATURE FIELD
      4       C*************************************************************
      5             SUBROUTINE DENSTB(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,A,B,C,
      6            CMAR,MRH,
      7            CT,RO,RREF,TREF)
      8             DIMENSION RO(IN,JN,KN),T(IN,JN,KN)
      9             DIMENSION MAR(IN,JN),MRH(IWN,JWN)
     10             DO 10 I=1,IN
     11             DO 10 J=1,JN
     12             IF (MAR(I,J).EQ.0) GO TO 12
     13             DO 11 K=1,KN
     14             TEM=T(I,J,K)*TREF+TREF
     15             R=A+B*TEM+C*TEM*TEM
     16             RO(I,J,K)=(R-RREF)/RREF
     17       11    CONTINUE
     18       12    CONTINUE
     19       10    CONTINUE
     20             RETURN
     21             END
```

9.2.6 DENSTY

This program uses the equation of state and computes density field from the temperature field

Eq of State

$$\rho = A + B(T) + C(T)^2$$

In the program

$$\rho = A + B\ (Tem) + C(TEM)^2$$

Where A,B,C are constants and there values are A = 1029431, B = -0.00002 and C = -0.0000048

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

### 9.2.7 DINERU

This subroutine computes DIHUUX, DIHUVX. This program is called in INTE. The results are used in Poisson equation for pressure.

```
*DOC.DINERU
  1              SUBROUTINE DINERU(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1HUUX,D1HUVX,MAR)
  2              DIMENSION U(IN,JN,KN),V(IN,JN,KN),HI(IN,JN),MAR(IN,JN)
  3              IF(MAR(I,J).EQ.0) GO TO 50
  4              IF(MAR(I,J).EQ.1) GO TO 31
  5              IF(MAR(I,J).EQ.2) GO TO 32
  6              IF(MAR(I,J).EQ.3) GO TO 33
  7              IF(MAR(I,J).EQ.4) GO TO 34
  8              IF(MAR(I,J).EQ.5) GO TO 35
  9              IF(MAR(I,J).EQ.6) GO TO 36
 10              IF(MAR(I,J).EQ.7) GO TO 37
 11              IF(MAR(I,J).EQ.8) GO TO 38
 12              IF(MAR(I,J).EQ.9) GO TO 39
 13              IF(MAR(I,J).EQ.10)GO TO 40
 14              D1HUUX=(U(I+1,J,K)*U(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
 15             C*U(I-1,J,K)*HI(I-1,J))/(2*DX)
 16              D1HUVX=(U(I+1,J,K)*V(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
 17             C*V(I-1,J,K)*HI(I-1,J))/(2*DX)
 18              GO TO 50
 19       31     CONTINUE
 20              D1HUUX=(U(I+1,J,K)*U(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
 21             C*U(I-1,J,K)*HI(I-1,J))/(2*DX)
 22              D1HUVX=(U(I+1,J,K)*V(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
 23             C*V(I-1,J,K)*HI(I-1,J))/(2*DX)
 24              GO TO 50
 25       32     CONTINUE
 26              D1HUUX=(U(I+1,J,K)*U(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
 27             C*U(I-1,J,K)*HI(I-1,J))/(2*DX)
 28              D1HUVX=(U(I+1,J,K)*V(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
 29             C*V(I-1,J,K)*HI(I-1,J))/(2*DX)
 30              GO TO 50
 31       33     CONTINUE
 32              D1HUUX=(4*HI(I+1,J)*U(I+1,J,K)*U(I+1,J,K)-3*HI(I,J)*U(I,J,K)
 33             C*U(I,J,K)-HI(I+2,J)*U(I+2,J,K)*U(I+2,J,K))/(2*DX)
 34              D1HUVX=(4*HI(I+1,J)*U(I+1,J,K)*V(I+1,J,K)-3*HI(I,J)*U(I,J,K)
 35             C*V(I,J,K)-HI(I+2,J)*U(I+2,J,K)*V(I+2,J,K))/(2*DX)
 36              GO TO 50
 37       34     CONTINUE
 38              D1HUUX=(3*HI(I,J)*U(I,J,K)*U(I,J,K)-4*HI(I-1,J)*U(I-1,J,K)
 39             C*U(I-1,J,K)+HI(I-2,J)*U(I-2,J,K)*U(I-2,J,K))/(2*DX)
 40              D1HUVX=(3*HI(I,J)*U(I,J,K)*V(I,J,K)-4*HI(I-1,J)*U(I-1,J,K)
 41             C*V(I-1,J,K)+HI(I-2,J)*U(I-2,J,K)*V(I-2,J,K))/(2*DX)
 42              GO TO 50
 43       35     CONTINUE
 44              D1HUUX=(4*HI(I+1,J)*U(I+1,J,K)*U(I+1,J,K)-3*HI(I,J)*U(I,J,K)
 45             C*U(I,J,K)-HI(I+2,J)*U(I+2,J,K)*U(I+2,J,K))/(2*DX)
 46              D1HUVX=(4*HI(I+1,J)*U(I+1,J,K)*V(I+1,J,K)-3*HI(I,J)*U(I,J,K)
 47             C*V(I,J,K)-HI(I+2,J)*U(I+2,J,K)*V(I+2,J,K))/(2*DX)
 48              GO TO 50
 49       36     CONTINUE
 50              D1HUUX=(U(I+1,J,K)*U(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
 51             C*U(I-1,J,K)*HI(I-1,J))/(2*DX)
 52              D1HUVX=(U(I+1,J,K)*V(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
 53             C*V(I-1,J,K)*HI(I-1,J))/(2*DX)
 54              GO TO 50
 55       37     CONTINUE
 56              D1HUUX=(4*HI(I+1,J)*U(I+1,J,K)*U(I+1,J,K)-3*HI(I,J)*U(I,J,K)
```

```
     C*U(I,J,K)-HI(I+2,J)*U(I+2,J,K)*U(I+2,J,K))/(2*DX)
      D1HUVX=(4*HI(I+1,J)*U(I+1,J,K)*V(I+1,J,K)-3*HI(I,J)*U(I,J,K)
     C*V(I,J,K)-HI(I+2,J)*U(I+2,J,K)*V(I+2,J,K))/(2*DX)
      GO TO 50
38    CONTINUE
      D1HUUX=(U(I+1,J,K)*U(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
     C*U(I-1,J,K)*HI(I-1,J))/(2*DX)
      D1HUVX=(U(I+1,J,K)*V(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
     C*V(I-1,J,K)*HI(I-1,J))/(2*DX)
      GO TO 50
39    CONTINUE
      D1HUUX=(3*HI(I,J)*U(I,J,K)*U(I,J,K)-4*HI(I-1,J)*U(I-1,J,K)
     C*U(I-1,J,K)+HI(I-2,J)*U(I-2,J,K)*U(I-2,J,K))/(2*DX)
      D1HUVX=(3*HI(I,J)*U(I,J,K)*V(I,J,K)-4*HI(I-1,J)*U(I-1,J,K)
     C*V(I-1,J,K)+HI(I-2,J)*U(I-2,J,K)*V(I-2,J,K))/(2*DX)
      GO TO 50
40    CONTINUE
      D1HUUX=(3*HI(I,J)*U(I,J,K)*U(I,J,K)-4*HI(I-1,J)*U(I-1,J,K)
     C*U(I-1,J,K)+HI(I-2,J)*U(I-2,J,K)*U(I-2,J,K))/(2*DX)
      D1HUVX=(3*HI(I,J)*U(I,J,K)*V(I,J,K)-4*HI(I-1,J)*U(I-1,J,K)
     C*V(I-1,J,K)+HI(I-2,J)*U(I-2,J,K)*V(I-2,J,K))/(2*DX)
50    CONTINUE
      RETURN
      END
```

9.2.8 DUVY

This program computes DIHUVY. The program is called in by INTE, $\frac{\partial}{\partial \beta}$ (huv) is computed for interior, boundary and corner points by the scheme similar to the one used in DINERU.

```
I*DOC.DUVY
  1              SUBROUTINE DUVY(I,J,K,IN,JN,KN,U,V,HI,DY,D1HUVY,MAR)
  2              DIMENSION U(IN,JN,KN),V(IN,JN,KN),HI(IN,JN),MAR(IN,JN)
  3              IF(MAR(I,J).EQ.0) GO TO 50
  4              IF(MAR(I,J).EQ.1) GO TO 31
  5              IF(MAR(I,J).EQ.2) GO TO 32
  6              IF(MAR(I,J).EQ.3) GO TO 33
  7              IF(MAR(I,J).EQ.4) GO TO 34
  8              IF(MAR(I,J).EQ.5) GO TO 35
  9              IF(MAR(I,J).EQ.6) GO TO 36
 10              IF(MAR(I,J).EQ.7) GO TO 37
 11              IF(MAR(I,J).EQ.8) GO TO 38
 12              IF(MAR(I,J).EQ.9) GO TO 39
 13              IF(MAR(I,J).EQ.10)GO TO 40
 14              D1HUVY=(U(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-U(I,J-1,K)
 15             C*V(I,J-1,K)*HI(I,J-1))/(2*DY)
 16              GO TO 50
 17        31    CONTINUE
 18              D1HUVY=(3*HI(I,J)*U(I,J,K)*V(I,J,K)-4*HI(I,J-1)*U(I,J-1,K)
 19             C*V(I,J-1,K)+HI(I,J-2)*U(I,J-2,K)*V(I,J-2,K))/(2*DY)
 20              GO TO 50
 21        32    CONTINUE
 22              D1HUVY=(4*HI(I,J+1)*U(I,J+1,K)*V(I,J+1,K)-3*HI(I,J)*
 23             CU(I,J,K)*V(I,J,K)-HI(I,J+2)*U(I,J+2,K)*V(I,J+2,K))/(2*DY)
 24              GO TO 50
 25        33    CONTINUE
 26              D1HUVY=(U(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-U(I,J-1,K)
 27             C*V(I,J-1,K)*HI(I,J-1))/(2*DY)
 28              GO TO 50
 29        34    CONTINUE
 30              D1HUVY=(U(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-U(I,J-1,K)
 31             C*V(I,J-1,K)*HI(I,J-1))/(2*DY)
 32              GO TO 50
 33        35    CONTINUE
 34              D1HUVY=(3*HI(I,J)*U(I,J,K)*V(I,J,K)-4*HI(I,J-1)*U(I,J-1,K)
 35             C*V(I,J-1,K)+HI(I,J-2)*U(I,J-2,K)*V(I,J-2,K))/(2*DY)
 36              GO TO 50
 37        36    CONTINUE
 38              D1HUVY=(U(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-U(I,J-1,K)
 39             C*V(I,J-1,K)*HI(I,J-1))/(2*DY)
 40              GO TO 50
 41        37    CONTINUE
 42              D1HUVY=(4*HI(I,J+1)*U(I,J+1,K)*V(I,J+1,K)-3*HI(I,J)*
 43             CU(I,J,K)*V(I,J,K)-HI(I,J+2)*U(I,J+2,K)*V(I,J+2,K))/(2*DY)
 44              GO TO 50
 45        38    CONTINUE
 46              D1HUVY=(U(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-U(I,J-1,K)
 47             C*V(I,J-1,K)*HI(I,J-1))/(2*DY)
 48              GO TO 50
 49        39    CONTINUE
 50              D1HUVY=(4*HI(I,J+1)*U(I,J+1,K)*V(I,J+1,K)-3*HI(I,J)*
 51             CU(I,J,K)*V(I,J,K)-HI(I,J+2)*U(I,J+2,K)*V(I,J+2,K))/(2*DY)
 52              GO TO 50
 53        40    CONTINUE
 54              D1HUVY=(3*HI(I,J)*U(I,J,K)*V(I,J,K)-4*HI(I,J-1)*U(I,J-1,K)
 55             C*V(I,J-1,K)+HI(I,J-2)*U(I,J-2,K)*V(I,J-2,K))/(2*DY)
 56        50    CONTINUE
```

```
57              RETURN
58              END
```

9.2.9 DVVY

This program computes DIHVVY. This program is called by INTE. $\frac{\partial}{\partial \beta}$ (hvv) is computed for interior, boundary or corner by the scheme similar to the one used in the subroutine DINERU.

```
*DOC.DVVY
  1              SUBROUTINE DVVY(I,J,K,IN,JN,KN,U,V,HI,DY,D1HVVY,MAR)
  2              DIMENSION U(IN,JN,KN),V(IN,JN,KN),HI(IN,JN),MAR(IN,JN)
  3              IF(MAR(I,J).EQ.0) GO TO 50
  4              IF(MAR(I,J).EQ.1) GO TO 31
  5              IF(MAR(I,J).EQ.2) GO TO 32
  6              IF(MAR(I,J).EQ.3) GO TO 33
  7              IF(MAR(I,J).EQ.4) GO TO 34
  8              IF(MAR(I,J).EQ.5) GO TO 35
  9              IF(MAR(I,J).EQ.6) GO TO 36
 10              IF(MAR(I,J).EQ.7) GO TO 37
 11              IF(MAR(I,J).EQ.8) GO TO 38
 12              IF(MAR(I,J).EQ.9) GO TO 39
 13              IF(MAR(I,J).EQ.10)GO TO 40
 14              D1HVVY=(V(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-V(I,J-1,K)*
 15             CV(I,J-1,K)*HI(I,J-1))/(2*DY)
 16              GO TO 50
 17        31    CONTINUE
 18              D1HVVY=(3*HI(I,J)*V(I,J,K)*V(I,J,K)+HI(I,J-2)*V(I,J-2,K)
 19             C*V(I,J-2,K)-4*HI(I,J-1)*V(I,J-1,K)*V(I,J-1,K))/(2*DY)
 20              GO TO 50
 21        32    CONTINUE
 22              D1HVVY=(4*HI(I,J+1)*V(I,J+1,K)*V(I,J+1,K)-3*HI(I,J)*V(I,J,K)
 23             C*V(I,J,K)-HI(I,J+2)*V(I,J+2,K)*V(I,J+2,K))/(2*DY)
 24              GO TO 50
 25        33    CONTINUE
 26              D1HVVY=(V(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-V(I,J-1,K)*
 27             CV(I,J-1,K)*HI(I,J-1))/(2*DY)
 28              GO TO 50
 29        34    CONTINUE
 30              D1HVVY=(V(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-V(I,J-1,K)*
 31             CV(I,J-1,K)*HI(I,J-1))/(2*DY)
 32              GO TO 50
 33        35    CONTINUE
 34              D1HVVY=(3*HI(I,J)*V(I,J,K)*V(I,J,K)+HI(I,J-2)*V(I,J-2,K)
 35             C*V(I,J-2,K)-4*HI(I,J-1)*V(I,J-1,K)*V(I,J-1,K))/(2*DY)
 36              GO TO 50
 37        36    CONTINUE
 38              D1HVVY=(V(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-V(I,J-1,K)*
 39             CV(I,J-1,K)*HI(I,J-1))/(2*DY)
 40              GO TO 50
 41        37    CONTINUE
 42              D1HVVY=(4*HI(I,J+1)*V(I,J+1,K)*V(I,J+1,K)-3*HI(I,J)*V(I,J,K)
 43             C*V(I,J,K)-HI(I,J+2)*V(I,J+2,K)*V(I,J+2,K))/(2*DY)
 44              GO TO 50
 45        38    CONTINUE
 46              D1HVVY=(V(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-V(I,J-1,K)*
 47             CV(I,J-1,K)*HI(I,J-1))/(2*DY)
 48              GO TO 50
 49        39    CONTINUE
 50              D1HVVY=(4*HI(I,J+1)*V(I,J+1,K)*V(I,J+1,K)-3*HI(I,J)*V(I,J,K)
 51             C*V(I,J,K)-HI(I,J+2)*V(I,J+2,K)*V(I,J+2,K))/(2*DY)
 52              GO TO 50
 53        40    CONTINUE
 54              D1HVVY=(3*HI(I,J)*V(I,J,K)*V(I,J,K)+HI(I,J-2)*V(I,J-2,K)
 55             C*V(I,J-2,K)-4*HI(I,J-1)*V(I,J-1,K)*V(I,J-1,K))/(2*DY)
 56        50    CONTINUE
```

```
57              RETURN
58              END
```

### 9.2.10 DIVSU

This subroutine computes DIUX, D2UX, DIUY. This program is called by INTE. $\frac{\partial u}{\partial \alpha}$ and $\frac{\partial u}{\partial \beta}$ are computed for interior, boundary or corner points by a scheme similar to the one used in DINERU.

```
/*DOC.DVISU
  1              SUBROUTINE DVISU(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1UX,D2UX,D1UY,D2UY,
  2             CHAR)
  3              DIMENSION U(IN,JN,KN),V(IN,JN,KN),HI(IN,JN),MAR(IN,JN)
  4              IF(MAR(I,J).EQ.0) GO TO 50
  5              IF(MAR(I,J).EQ.1) GO TO 31
  6              IF(MAR(I,J).EQ.2) GO TO 32
  7              IF(MAR(I,J).EQ.3) GO TO 33
  8              IF(MAR(I,J).EQ.4) GO TO 34
  9              IF(MAR(I,J).EQ.5) GO TO 35
 10              IF(MAR(I,J).EQ.6) GO TO 36
 11              IF(MAR(I,J).EQ.7) GO TO 37
 12              IF(MAR(I,J).EQ.8) GO TO 38
 13              IF(MAR(I,J).EQ.9) GO TO 39
 14              IF(MAR(I,J).EQ.10)GO TO 40
 15              D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
 16              D1UY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
 17              D2UX=(U(I+1,J,K)-2*U(I,J,K)+U(I-1,J,K))/(DX*DX)
 18              D2UY=(U(I,J+1,K)-2*U(I,J,K)+U(I,J-1,K))/(DY*DY)
 19              GO TO 50
 20        31    CONTINUE
 21              D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
 22              D2UX=(U(I+1,J,K)-2*U(I,J,K)+U(I-1,J,K))/(DX*DX)
 23              D1UY=(3*U(I,J,K)+U(I,J-2,K)-4*U(I,J-1,K))/(2*DY)
 24              D2UY=(U(I,J,K)+U(I,J-2,K)-2*U(I,J-1,K))/(DY*DY)
 25              GO TO 50
 26        32    CONTINUE
 27              D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
 28              D2UX=(U(I+1,J,K)-2*U(I,J,K)+U(I-1,J,K))/(DX*DX)
 29              D1UY=(4*U(I,J+1,K)-3*U(I,J,K)-U(I,J+2,K))/(2*DY)
 30              D2UY=(U(I,J+2,K)+U(I,J,K)-2*U(I,J+1,K))/(DY*DY)
 31              GO TO 50
 32        33    CONTINUE
 33              D1UY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
 34              D2UY=(U(I,J+1,K)-2*U(I,J,K)+U(I,J-1,K))/(DY*DY)
 35              D1UX=(4*U(I+1,J,K)-3*U(I,J,K)-U(I+2,J,K))/(2*DX)
 36              D2UX=(U(I+2,J,K)-2*U(I+1,J,K)+U(I,J,K))/(DX*DX)
 37              GO TO 50
 38        34    CONTINUE
 39              D1UY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
 40              D2UY=(U(I,J+1,K)-2*U(I,J,K)+U(I,J-1,K))/(DY*DY)
 41              D1UX=(3*U(I,J,K)-4*U(I-1,J,K)+U(I-2,J,K))/(2*DX)
 42              D2UX=(U(I,J,K)-2*U(I-1,J,K)+U(I-2,J,K))/(DX*DX)
 43              GO TO 50
 44        35    CONTINUE
 45              D1UY=(3*U(I,J,K)+U(I,J-2,K)-4*U(I,J-1,K))/(2*DY)
 46              D2UY=(U(I,J,K)+U(I,J-2,K)-2*U(I,J-1,K))/(DY*DY)
 47              D1UX=(4*U(I+1,J,K)-3*U(I,J,K)-U(I+2,J,K))/(2*DX)
 48              D2UX=(U(I+2,J,K)-2*U(I+1,J,K)+U(I,J,K))/(DX*DX)
 49              GO TO 50
 50        36    CONTINUE
 51              D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
 52              D1UY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
 53              D2UX=(U(I+1,J,K)-2*U(I,J,K)+U(I-1,J,K))/(DX*DX)
 54              D2UY=(U(I,J+1,K)-2*U(I,J,K)+U(I,J-1,K))/(DY*DY)
 55              GO TO 50
 56        37    CONTINUE
```

```
57              D1UY=(4*U(I,J+1,K)-3*U(I,J,K)-U(I,J+2,K))/(2*DY)
58              D2UY=(U(I,J+2,K)+U(I,J,K)-2*U(I,J+1,K))/(DY*DY)
59              D1UX=(4*U(I+1,J,K)-3*U(I,J,K)-U(I+2,J,K))/(2*DX)
60              D2UX=(U(I+2,J,K)-2*U(I+1,J,K)+U(I,J,K))/(DX*DX)
61              GO TO 50
62        38    CONTINUE
63              D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
64              D1UY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
65              D2UX=(U(I+1,J,K)-2*U(I,J,K)+U(I-1,J,K))/(DX*DX)
66              D2UY=(U(I,J+1,K)-2*U(I,J,K)+U(I,J-1,K))/(DY*DY)
67              GO TO 50
68        39    CONTINUE
69              D1UY=(4*U(I,J+1,K)-3*U(I,J,K)-U(I,J+2,K))/(2*DY)
70              D2UY=(U(I,J+2,K)+U(I,J,K)-2*U(I,J+1,K))/(DY*DY)
71              D1UX=(3*U(I,J,K)-4*U(I-1,J,K)+U(I-2,J,K))/(2*DX)
72              D2UX=(U(I,J,K)-2*U(I-1,J,K)+U(I-2,J,K))/(DX*DX)
73              GO TO 50
74        40    CONTINUE
75              D1UY=(3*U(I,J,K)+U(I,J-2,K)-4*U(I,J-1,K))/(2*DY)
76              D2UY=(U(I,J,K)+U(I,J-2,K)-2*U(I,J-1,K))/(DY*DY)
77              D1UX=(3*U(I,J,K)-4*U(I-1,J,K)+U(I-2,J,K))/(2*DX)
78              D2UX=(U(I,J,K)-2*U(I-1,J,K)+U(I-2,J,K))/(DX*DX)
79        50    CONTINUE
80              RETURN
81              END
```

9.2.11 DVISV

This program computes DIVY, D2VY, DIVX AND D2VX. This program is called by subroutine INTE. Schemes used are similar to the one used in DVISU.

```
EN*DOC.DVISV
    1              SUBROUTINE DVISV(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1VX,D2VX,D1VY,D2VY,
    2             CMAR)
    3              DIMENSION U(IN,JN,KN),V(IN,JN,KN),HI(IN,JN),MAR(IN,JN)
    4              IF(MAR(I,J).EQ.0) GO TO 50
    5              IF(MAR(I,J).EQ.1) GO TO 31
    6              IF(MAR(I,J).EQ.2) GO TO 32
    7              IF(MAR(I,J).EQ.3) GO TO 33
    8              IF(MAR(I,J).EQ.4) GO TO 34
    9              IF(MAR(I,J).EQ.5) GO TO 35
   10              IF(MAR(I,J).EQ.6) GO TO 36
   11              IF(MAR(I,J).EQ.7) GO TO 37
   12              IF(MAR(I,J).EQ.8) GO TO 38
   13              IF(MAR(I,J).EQ.9) GO TO 39
   14              IF(MAR(I,J).EQ.10)GO TO 40
   15              D1VX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
   16              D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
   17              D2VX=(V(I+1,J,K)-2*V(I,J,K)+V(I-1,J,K))/(DX*DX)
   18              D2VY=(V(I,J+1,K)-2*V(I,J,K)+V(I,J-1,K))/(DY*DY)
   19              GO TO 50
   20        31    CONTINUE
   21              D1VX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
   22              D2VX=(V(I+1,J,K)-2*V(I,J,K)+V(I-1,J,K))/(DX*DX)
   23              D1VY=(3*V(I,J,K)-4*V(I,J-1,K)+V(I,J-2,K))/(2*DY)
   24              D2VY=(V(I,J,K)+V(I,J-2,K)-2*V(I,J-1,K))/(DY*DY)
   25              GO TO 50
   26        32    CONTINUE
   27              D1VX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
   28              D2VX=(V(I+1,J,K)-2*V(I,J,K)+V(I-1,J,K))/(DX*DX)
   29              D1VY=(4*V(I,J+1,K)-3*V(I,J,K)-V(I,J+2,K))/(2*DY)
   30              D2VY=(V(I,J+2,K)+V(I,J,K)-2*V(I,J+1,K))/(DY*DY)
   31              GO TO 50
   32        33    CONTINUE
   33              D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
   34              D2VY=(V(I,J+1,K)-2*V(I,J,K)+V(I,J-1,K))/(DY*DY)
   35              D1VX=(4*V(I+1,J,K)-3*V(I,J,K)-V(I+2,J,K))/(2*DX)
   36              D2VX=(V(I+2,J,K)-2*V(I+1,J,K)+V(I,J,K))/(DX*DX)
   37              GO TO 50
   38        34    CONTINUE
   39              D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
   40              D2VY=(V(I,J+1,K)-2*V(I,J,K)+V(I,J-1,K))/(DY*DY)
   41              D1VX=(3*V(I,J,K)-4*V(I-1,J,K)+V(I-2,J,K))/(2*DX)
   42              D2VX=(V(I,J,K)-2*V(I-1,J,K)+V(I-2,J,K))/(DX*DX)
   43              GO TO 50
   44        35    CONTINUE
   45              D1VY=(3*V(I,J,K)-4*V(I,J-1,K)+V(I,J-2,K))/(2*DY)
   46              D2VY=(V(I,J,K)+V(I,J-2,K)-2*V(I,J-1,K))/(DY*DY)
   47              D1VX=(4*V(I+1,J,K)-3*V(I,J,K)-V(I+2,J,K))/(2*DX)
   48              D2VX=(V(I+2,J,K)-2*V(I+1,J,K)+V(I,J,K))/(DX*DX)
   49              GO TO 50
   50        36    CONTINUE
   51              D1VX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
   52              D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
   53              D2VX=(V(I+1,J,K)-2*V(I,J,K)+V(I-1,J,K))/(DX*DX)
   54              D2VY=(V(I,J+1,K)-2*V(I,J,K)+V(I,J-1,K))/(DY*DY)
   55              GO TO 50
   56        37    CONTINUE
```

```
57              D1VY=(4*V(I,J+1,K)-3*V(I,J,K)-V(I,J+2,K))/(2*DY)
58              D2VY=(V(I,J+2,K)+V(I,J,K)-2*V(I,J+1,K))/(DY*DY)
59              D1VX=(4*V(I+1,J,K)-3*V(I,J,K)-V(I+2,J,K))/(2*DX)
60              D2VX=(V(I+2,J,K)-2*V(I+1,J,K)+V(I,J,K))/(DX*DX)
61              GO TO 50
62        38    CONTINUE
63              D1VX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
64              D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
65              D2VX=(V(I+1,J,K)-2*V(I,J,K)+V(I-1,J,K))/(DX*DX)
66              D2VY=(V(I,J+1,K)-2*V(I,J,K)+V(I,J-1,K))/(DY*DY)
67              GO TO 50
68        39    CONTINUE
69              D1VY=(4*V(I,J+1,K)-3*V(I,J,K)-V(I,J+2,K))/(2*DY)
70              D2VY=(V(I,J+2,K)+V(I,J,K)-2*V(I,J+1,K))/(DY*DY)
71              D1VX=(3*V(I,J,K)-4*V(I-1,J,K)+V(I-2,J,K))/(2*DX)
72              D2VX=(V(I,J,K)-2*V(I-1,J,K)+V(I-2,J,K))/(DX*DX)
73              GO TO 50
74        40    CONTINUE
75              D1VY=(3*V(I,J,K)-4*V(I,J-1,K)+V(I,J-2,K))/(2*DY)
76              D2VY=(V(I,J,K)+V(I,J-2,K)-2*V(I,J-1,K))/(DY*DY)
77              D1VX=(3*V(I,J,K)-4*V(I-1,J,K)+V(I-2,J,K))/(2*DX)
78              D2VX=(V(I,J,K)-2*V(I-1,J,K)+V(I-2,J,K))/(DX*DX)
79        50    CONTINUE
80              RETURN
81              END
```

9.2.12 D12Z

This subroutine computes D1UZ, DZUZ, D1VZ, DZVZ, D1A3Z, D1UVZ. This subroutine is called by INTE. At points on the surface, KN=1, forward difference scheme is employed.

```
N*DOC.D12Z
      C ***********************************************************************
      C       THIS PROGRAM CALCULATES THE Z DERIVATIVES
      C ***********************************************************************
            SUBROUTINE D12Z(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,DX,DY,DZ,D1UWZ,
           CA3,TAUX,TAUY,
           CD1VWZ,D1UZ,D2UZ,D1VZ,D2VZ,D1A3Z)
            DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),HI(IN,JN),
           CHX(IN,JN),HY(IN,JN)
            DIMENSION A3(KN)
            IF (K.EQ.1) GO TO 61
            IF (K.EQ.KN) GO TO 62
            D1UZ=(U(I,J,K+1)-U(I,J,K-1))/(2*DZ)
            D1VZ=(V(I,J,K+1)-V(I,J,K-1))/(2*DZ)
            D2UZ=(U(I,J,K+1)-2*U(I,J,K)+U(I,J,K-1))/(DZ*DZ)
            D2VZ=(V(I,J,K+1)-2*V(I,J,K)+V(I,J,K-1))/(DZ*DZ)
            D1A3Z=(A3(K+1)-A3(K-1))/(2*DZ)
            D1UWZ=(U(I,J,K+1)*W(I,J,K+1)-U(I,J,K-1)*W(I,J,K-1))/(2*DZ)
            D1VWZ=(V(I,J,K+1)*W(I,J,K+1)-V(I,J,K-1)*W(I,J,K-1))/(2*DZ)
            GO TO 63
      61    CONTINUE
            D1UZ=HI(I,J)*TAUX
            D1VZ=HI(I,J)*TAUY
            D2UZ=2*(U(I,J,K+1)-U(I,J,K))/(DZ*DZ)-2*(TAUX*HI(I,J)/DZ)
            D2VZ=2*(V(I,J,K+1)-V(I,J,K))/(DZ*DZ)-2*(TAUY*HI(I,J)/DZ)
            D1A3Z=(4*A3(K+1)-3*A3(K)-A3(K+2))/(2*DZ)
            D1UWZ=(4*U(I,J,K+1)*W(I,J,K+1)-3*U(I,J,K)*W(I,J,K)-U(I,J,K+2)*W(I
           C,J,K+2))/(2*DZ)
            D1VWZ=(4*V(I,J,K+1)*W(I,J,K+1)-3*V(I,J,K)*W(I,J,K)
           C-V(I,J,K+2)*W(I,J,K+2))/(2*DZ)
            GO TO 63
      62    CONTINUE
            D1UZ=(3*U(I,J,K)-4*U(I,J,K-1)+U(I,J,K-2))/(2*DZ)
            D1VZ=(3*V(I,J,K)-4*V(I,J,K-1)+V(I,J,K-2))/(2*DZ)
            D2UZ=(U(I,J,K-2)+U(I,J,K)-2*U(I,J,K-1))/(DZ*DZ)
            D2VZ=(V(I,J,K-2)+V(I,J,K)-2*V(I,J,K-1))/(DZ*DZ)
            D1A3Z=(3*A3(K)-4*A3(K-1)+A3(K-2))/(2*DZ)
            D1UWZ=(3*U(I,J,K)*W(I,J,K)-4*U(I,J,K-1)*W(I,J,K-1)
           C+U(I,J,K-2)*W(I,J,K-2))/(2*DZ)
            D1VWZ=(3*V(I,J,K)*W(I,J,K)+4*V(I,J,K-1)*W(I,J,K-1)
           C+V(I,J,K-2)*W(I,J,K-2))/(2*DZ)
      63    CONTINUE
            RETURN
            END
```

9.2.13 ERROR: Calculates "HIRT and HARLOW" correction term at half grid points and at the surface. The last term in Poisson's equation is "HIRT and HARLOW" correction term. This is evaluated by a backward difference in time with present time set equal to zero. This is necessary because Poisson's equation is usually solved by the iterative technique usually leading to errors. If they are not corrected, continuity equation will not be satisfied leading to accumulation or loss of fluid from the system.

In the program

$$\text{WHLDT (IW, JW)} = \frac{-\text{WH(IW,JW,1)}}{\text{DT}}$$

WH at previous time step is set at zero,

WH at present time step is nonzero.

```
N*DOC.ERROR
      C*****************************************************************************
      C     THIS PROGRAM CALCULATES THE HIRT AND HARLOW CORRECTION TERM AT THE
      C                                   SURFACE
      C*****************************************************************************
            SUBROUTINE ERROR(IWN,JWN,IW,JW,DT,WH,WHLDT,KN,MRH)
            DIMENSION WHLDT(IWN,JWN),WH(IWN,JWN,KN)
            DIMENSION MRH(IWN,JWN)
      C WHLDT IS THE TIME DERIVATIVE OF W AT HALF GRID POINTS AT LID
            DO 3100 IW=1,IWN
            DO 3100 JW=1,JWN
            IF (MRH(IW,JW).EQ.0) GO TO 3000
            WHLDT(IW,JW)=-WH(IW,JW,1)/DT
      3000  CONTINUE
      3100  CONTINUE
            RETURN
            END
```

9.2.14 FORCE: This program computes R.H.S. of Poisson's equation at half grid points.

$$FH = \frac{1}{h}\frac{\partial}{\partial\alpha}(-Ax_1 + Ax_2 + C_x - Xp)$$

$$+ \frac{1}{h}\frac{\partial}{\partial\beta}(-Ay_1 - Av2 + Cy - YP)$$

$$- \frac{1}{h}\left(\frac{\partial h}{\partial\alpha}\frac{\partial Ps}{\partial\alpha} + \frac{\partial h}{\partial\beta}\frac{\partial Ps}{\partial\beta}\right) - \frac{\partial\Omega}{\partial t}\Big|_{z=0}$$

$$= \frac{1}{h}\frac{\partial}{\partial\alpha}(XINT) + \frac{1}{h}\frac{\partial}{\partial\beta}(YINT)$$

$$- \frac{1}{h}\left[\frac{\partial h}{\partial\alpha}\left(DPSX + \frac{1}{Rb}v - Bx\right) + \frac{\partial h}{\partial\beta}\left(DPSY - \frac{1}{Rb}u - By\right)\right]$$

$$- WHLDT$$

This program takes $\frac{1}{Rb}$ v, $\frac{1}{Rb}$ u, Bx and By equal to zero

$\frac{\partial Ps}{\partial\alpha}$ = DPSX

and $\frac{\partial Ps}{\partial\alpha}$ at half grid points is average of four surrounding main grid points.

$\frac{\partial}{\partial\alpha}$ (XINT) , $\frac{\partial}{\partial\beta}$ (YINT) at half grid points is average of four surrounding main grid points.

```
!*DOC.FORCE
      SUBROUTINE FORCE(I,J,IW,JW,XINT,YINT,WHLDT,DX,DY,HI,HX,HY,
     CMRH,
     CDPSX,DPSY,FH,AP,IN,JN,IWN,JWN,RINTX,RINTY,U,V,EUL,ABR,MAR,KN)
      DIMENSION XINT(IN,JN),YINT(IN,JN),WHLDT(IWN,JWN),HI(IN,JN),HX(IN,
     CJN),HY(IN,JN),DPSX(IN,JN),DPSY(IN,JN),FH(IWN,JWN)
      DIMENSION MRH(IWN,JWN)
      DIMENSION RINTX(IN,JN,KN),RINTY(IN,JN,KN),U(IN,JN,KN),V(IN,JN,KN)
     C,MAR(IN,JN)
      K=1
      DO 90 I=1,IN
      DO 90 J=1,JN
      IF(MAR(I,J).LT.11) GO TO 90
      DPSX(I,J)=DPSX(I,J)-EUL*RINTX(I,J,K)+V(I,J,K)*ABR
      DPSY(I,J)=DPSY(I,J)-EUL*RINTY(I,J,K)-U(I,J,K)*ABR
   90 CONTINUE
      DO 10 IW=1,IWN
      DO 10 JW=1,JWN
      I=IW
      J=JW
      IF (MRH(IW,JW).EQ.0) GO TO 9
      DPSXH=(DPSX(I,J)+DPSX(I+1,J)+DPSX(I,J+1)+DPSX(I+1,J+1))/4.0
      DPSYH=(DPSY(I,J)+DPSY(I+1,J)+DPSY(I,J+1)+DPSY(I+1,J+1))/4.0
      HXH=(HX(I,J)+HX(I+1,J)+HX(I,J+1)+HX(I+1,J+1))/4.0
      HYH=(HY(I,J)+HY(I+1,J)+HY(I,J+1)+HY(I+1,J+1))/4.0
      DXINT=(XINT(I+1,J)+XINT(I+1,J+1)-XINT(I,J)-XINT(I,J+1))/(2*DX)
      DYINT=(YINT(I,J+1)+YINT(I+1,J+1)-YINT(I,J)-YINT(I+1,J))/(2*DY)
      HH=(HI(I,J)+HI(I+1,J)+HI(I,J+1)+HI(I+1,J+1))/4.0
      FH(IW,JW)=(1./AP)*(-(1./HH)*(DXINT+DYINT)-WHLDT(IW,JW)-(AP/HH)*
     C(HXH*DPSXH+HYH*DPSYH))
    9 CONTINUE
   10 CONTINUE
      RETURN
      END
```

9.2.15 GRADS

This program compute slopes of the bottom using non-dimensionalized and unstretched depths and $\alpha$ and $\beta$ coordinates.

```
M*DULL(1).GRADS
      SUBROUTINE GRADS(IN,JN,KN,IWN,JWN,HI,HX,HY,MAR,MRH,DX,DY)
      DIMENSION MAR(IN,JN),HI(IN,JN),HX(IN,JN),HY(IN,JN)
      DIMENSION MRH(IWN,JWN)
C      TO CALCULATE HXAND HY
      DO 50 I=1,IN
      DO 50 J=1,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
      HX(I,J)=(HI(I+1,J)-HI(I-1,J))/(2*DX)
      HY(I,J)=(HI(I,J+1)-HI(I,J-1))/(2*DY)
      GO TO 50
31    CONTINUE
      HX(I,J)=(HI(I+1,J)-HI(I-1,J))/(2*DX)
      HY(I,J)=(3*HI(I,J)+HI(I,J-2)-4*HI(I,J-1))/(2*DY)
      GO TO 50
32    CONTINUE
      HX(I,J)=(HI(I+1,J)-HI(I-1,J))/(2*DX)
      HY(I,J)=(4*HI(I,J+1)-3*HI(I,J)-HI(I,J+2))/(2*DY)
      GO TO 50
33    CONTINUE
      HX(I,J)=(4*HI(I+1,J)-3*HI(I,J)-HI(I+2,J))/(2*DX)
      HY(I,J)=(HI(I,J+1)-HI(I,J-1))/(2*DY)
      GO TO 50
34    CONTINUE
      HX(I,J)=(3*HI(I,J)+HI(I-2,J)-4*HI(I-1,J))/(2*DX)
      HY(I,J)=(HI(I,J+1)-HI(I,J-1))/(2*DY)
      GO TO 50
35    CONTINUE
      HX(I,J)=(4*HI(I+1,J)-3*HI(I,J)-HI(I+2,J))/(2*DX)
      HY(I,J)=(3*HI(I,J)+HI(I,J-2)-4*HI(I,J-1))/(2*DY)
      GO TO 50
36    CONTINUE
      HX(I,J)=(HI(I+1,J)-HI(I-1,J))/(2*DX)
      HY(I,J)=(HI(I,J+1)-HI(I,J-1))/(2*DY)
      GO TO 50
37    CONTINUE
      HX(I,J)=(4*HI(I+1,J)-3*HI(I,J)-HI(I+2,J))/(2*DX)
      HY(I,J)=(4*HI(I,J+1)-3*HI(I,J)-HI(I,J+2))/(2*DY)
      GO TO 50
38    CONTINUE
      HX(I,J)=(HI(I+1,J)-HI(I-1,J))/(2*DX)
      HY(I,J)=(HI(I,J+1)-HI(I,J-1))/(2*DY)
      GO TO 50
39    CONTINUE
      HX(I,J)=(3*HI(I,J)+HI(I-2,J)-4*HI(I-1,J))/(2*DX)
      HY(I,J)=(4*HI(I,J+1)-3*HI(I,J)-HI(I,J+2))/(2*DY)
      GO TO 50
```

```
      40    CONTINUE
            HX(I,J)=(3*HI(I,J)+HI(I-2,J)-4*HI(I-1,J))/(2*DX)
            HY(I,J)=(3*HI(I,J)+HI(I,J-2)-4*HI(I,J-1))/(2*DY)
      50    CONTINUE
            WRITE (9) ((MAR(I,J),I=1,IN),J=1,JN),
           C((MRH(IW,JW),IW=1,IWN),JW=1,JWN),
           C((HI(I,J),I=1,IN),J=1,JN),
            DO 60 I=1,IN
            PRINT 61,I,(MAR(I,J),J=1,JN)
      61    FORMAT(/' I=',I3/,' MARKER'/(5X,9I3))
      60    CONTINUE
            DO 62 IW=1,IWN
            PRINT 63,IW,(MRH(IW,JW),JW=1,JWN)
      63    FORMAT(/' IW=',I3/,' MIDMARKER'/(5X,8I3))
      62    CONTINUE
            DO 70 I=1,IN
            PRINT 71,I,(HI(I,J),J=1,JN)
      71    FORMAT(/' I=',I3/,' DEPTH'/(5X,9E14.7))
            PRINT 72,I,(HX(I,J),J=1,JN)
      72    FORMAT(/' I=',I3/,' XGRAD'/(5X,9E14.7))
            PRINT 73,I,(HY(I,J),J=1,JN)
      73    FORMAT(' I=',I3/,' YGRAD'/(5X,9E14.7))
      70    CONTINUE
            END
```

9.2.16 HEIGHT

This program inputs depth of the basin into the model. This subroutine is for constant depth model.

HI (I,J) = CC = A constant and non-dimensionalized depth =1.0

HX (I,J) = 0.0 x derivative

HY (I,J) = 0.0 y derivative

```
N*DOC.HEIGHT
   1        C     THIS PROGRAM PUTS CONSTANT DEPTH FOR CC=1.0 IN THE DATA
   2              SUBROUTINE HEIGHT(I,J,K,IN,JN,KN,HI,HX,HY,CC)
   3              DIMENSION HI(IN,JN),HX(IN,JN),HY(IN,JN)
   4              DO 100 I=1,IN
   5              DO 100 J=1,JN
   6              HI(I,J)=CC
   7              HX(I,J)=0.0
   8              HY(I,J)=0.0
   9        100   CONTINUE
  10              RETURN
  11              END
```

### 9.2.17 HEIGH1

This subroutine reads the depths for the near field model in the format given.

```
*DOC.HEIGH1
  1       C     THIS PROGRAM READS DEPTHS FROM THE DATA
  2             SUBROUTINE HEIGH1(I,J,K,IN,JN,KN,HI,HX,HY,CC,JX)
  3             DIMENSION HI(IN,JN),HX(IN,JN),HY(IN,JN),JX(JN)
  4             DO 100 I=1,IN
  5             DO 100 J=1,JN
  6             HI(I,J)=CC
  7             HX(I,J)=0.0
  8             HY(I,J)=0.0
  9       100   CONTINUE
 10             DO 200 JJ=1,JN
 11             READ 2,J,(HI(I,J),I=1,IN)
 12             PRINT 3,J,(HI(I,J),I=1,IN)
 13             JX(JJ)=J
 14       200   CONTINUE
 15       3     FORMAT(5X,I5,18F6.2)
 16       2     FORMAT()
 17             RETURN
 18             END
```

9.2.18 HITEA

This subroutine is used by the far field stratified and unstratified models. The subroutine sets the depth everywhere equal to 1 if cc is nonzero, otherwise the depth matrix HI is read in from the data element DATAML when the main program TMAIN4 or TMAIN4CB is executed.

```
*DULL(1).HITEA
   1              SUBROUTINE HITEA(I,J,K,IN,JN,KN,HI,HX,HY,CC)
   2              DIMENSION HI(IN,JN),HX(IN,JN),HY(IN,JN)
   3              IF(CC.LT.0.0001) GO TO 150
   4              DO 100 I=1,IN
   5              DO 100 J=1,JN
   6              HI(I,J)=CC
   7              HX(I,J)=0.0
   8              HY(I,J)=0.0
   9        100   CONTINUE
  10              GO TO 250
  11        150   CONTINUE
  12              DO 200 II=1,IN
  13              READ 2,I,(HI(I,J),J=1,JN)
  14              PRINT 3,I,(HI(I,J),J=1,JN)
  15        200   CONTINUE
  16        250   CONTINUE
  17        3     FORMAT(5X,I5,13F6.2)
  18        2     FORMAT()
  19              RETURN
  20              END
```

9.2.19 INITB

This procedure is used for initializing the temperature field for a far field stratified model. The subroutine sets the temperature everywhere equal to the ambient temperature profile defined by the matrix AMINT (NTL, NTLV). The first column in this matrix is the depths and the second column are the corresponding temperatures.

```
H*DULL(1).INITB
    1       C*************************************************************************
    2       C       THIS PROGRAM INITIALIZES TEMP AND DENSITY
    3       C*************************************************************************
    4             SUBROUTINE INITB(I,J,K,IN,JN,KN,IW,JW,IWN,JWN,A,B,C,T,RO,
    5            CMAR,MRH,
    6            CTREF,RREF,
    7           *    CTO,AMINT,HI,NTL,NTLV)
    8             DIMENSION T(IN,JN,KN),RO(IN,JN,KN)
    9             DIMENSION MAR(IN,JN),MRH(IWN,JWN),AMINT(NTL,NTLV),HI(IN,JN)
   10             TOD=(TO-TREF)/TREF
   11             R=A+B*TO+C*TO*TO
   12             ROD=(R-RREF)/RREF
   13             DO 10 I=1,IN
   14             DO 10 J=1,JN
   15             IF (MAR(I,J).EQ.0) GO TO 12
   16             DO 11 K=1,KN
   17             HIK=(K-1)*HI(I,J)/(KN-1)
   18             NTLM=NTL-1
   19             DO 100 N=1,NTLM
   20             IF(HIK.GE.AMINT(N,1).AND.HIK.LT.AMINT(N+1,1))
   21            CT(I,J,K)=AMINT(N,2)
   22      100    CONTINUE
   23             IF(HIK.GE.AMINT(NTL,1))
   24            CT(I,J,K)=AMINT(NTL,2)
   25             RO(I,J,K)=A+B*T(I,J,K)+C*T(I,J,K)**2
   26             T(I,J,K)=(T(I,J,K)-TREF)/TREF
   27             RO(I,J,K)=(RO(I,J,K)-RREF)/RREF
   28      11     CONTINUE
   29      12     CONTINUE
   30      10     CONTINUE
   31             RETURN
   32             END
```

9.2.20 INITIA

This program initializes the values of u,v, w. $\dot{W}H$, D, E and PINTH. This program sets u, v, w, D, E. and $wH$ equal to zero. PINTH is set equal to ARBP.

DOC.INITIA

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

9.2.21 INITIT

This program sets initial temperature field. It sets the temperature field to the reference temperature at all points.

```
*DOC,INITIT
  1       C*****************************************************************************
  2       C       THIS PROGRAM INITIALIZES TEMP AND DENSITY
  3       C*****************************************************************************
  4             SUBROUTINE INITIT(I,J,K,IN,JN,KN,IW,JW,IWN,JWN,A,B,C,T,RO,
  5            CMAR,MRH,
  6            CTREF,RREF,
  7            CTW,ROW,TO)
  8             DIMENSION T(IN,JN,KN),RO(IN,JN,KN),TW(IWN,JWN,KN),ROW(IWN,JWN,KN)
  9             DIMENSION MAR(IN,JN),MRH(IWN,JWN)
 10             TOD=(TO-TREF)/TREF
 11             R=A+B*TO+C*TO*TO
 12             ROD=(R-RREF)/RREF
 13             DO 10 I=1,IN
 14             DO 10 J=1,JN
 15             IF (MAR(I,J).EQ.0) GO TO 12
 16             DO 11 K=1,KN
 17             T(I,J,K)=TOD
 18             RO(I,J,K)=ROD
 19       11    CONTINUE
 20       12    CONTINUE
 21       10    CONTINUE
 22             DO 20 IW=1,IWN
 23             DO 20 JW=1,JWN
 24             IF (MRH(IW,JW).EQ.0) GO TO 22
 25             DO 21 K=1,KN
 26             TW(IW,JW,K)=TOD
 27             ROW(IW,JW,K)=ROD
 28       21    CONTINUE
 29       22    CONTINUE
 30       20    CONTINUE
 31             RETURN
 32             END
```

9.2.22 INITM

This subroutine is used by the far field unstratified model. This subroutine is used by the main program TMAIN4T which sets up the temperature field equal to a measured initial value. This subroutine reads in the surface temperature matrix stored as data element ITPK1. Temperature below the surface are computed by assuming a temperature drop of $1^{o}C$ over the reference depth, a condition which may be changed if desired by making changes in line #21 of this subroutine.

```
*DULL(1).INITM
  1       C*************************************************************************
  2       C       THIS PROGRAM INITIALIZES TEMP AND DENSITY  FOR MORNING
  3       C*************************************************************************
  4             SUBROUTINE INITM(I,J,K,IN,JN,KN,IW,JW,IWN,JWN,A,B,C,T,RO,
  5            CMAR,MRH,
  6            CTREF,RREF,
  7            CTW,ROW,TO,HI)
  8             DIMENSION T(IN,JN,KN),RO(IN,JN,KN),TW(IWN,JWN,KN),ROW(IWN,JWN,KN)
  9             DIMENSION MAR(IN,JN),MRH(IWN,JWN),HI(IN,JN)
 10             TOD=(TO-TREF)/TREF
 11             R=A+B*TO+C*TO*TO
 12             ROD=(R-RREF)/RREF
 13             DO 900 II=1,IN
 14             READ 2,I,(T(I,J,1),J=1,JN)
 15       900   CONTINUE
 16       2     FORMAT ()
 17             DO 1001 K=2,KN
 18             DO 1001 J=1,JN
 19             DO 1001 I=1,IN
 20             DP=HI(I,J)*FLOAT(K-1)/FLOAT(KN-1)
 21             T(I,J,K)=T(I,J,1)-DP*1.0
 22       1001  CONTINUE
 23             DO 1002 K=1,KN
 24             DO 1002 J=1,JN
 25             DO 1002 I=1,IN
 26             RO(I,J,K)=A+B*T(I,J,K)+C*T(I,J,K)**2
 27             T(I,J,K)=(T(I,J,K)-TREF)/TREF
 28             RO(I,J,K)=(RO(I,J,K)-RREF)/RREF
 29       1002  CONTINUE
 30       11    CONTINUE
 31       12    CONTINUE
 32       10    CONTINUE
 33             DO 20 IW=1,IWN
 34             DO 20 JW=1,JWN
 35             IF (MRH(IW,JW).EQ.0) GO TO 22
 36             DO 21 K=1,KN
 37             TW(IW,JW,K)=T(IW,JW,K)
 38             ROW(IW,JW,K)=RO(IW,JW,K)
 39       21    CONTINUE
 40       22    CONTINUE
 41       20    CONTINUE
 42             RETURN
 43             END
```

9.2.23 INITMB

This subroutine updates the temperature field for a far field stratified model from the ambient field to the measured temperature field including the thermal plume. The subroutine reads in TSMN (the minimum surface temperature), DPMX (maximum depth to which the effect of thermal plume is extended) and the surface temperature matrix. Temperatures below the surface are computed by assuming a linear accumulation of plume heat from a maximum value at the surface to a zero at DPMX, Thus:

Temperature at depth d (<DPMX) = Surface Temperature

$$\text{Temp} - \frac{d}{DPMX} \text{ (Surface temperature - TSMN)}$$

Temperature at depth d (>DPMX) = Ambient temperature

```
*DULL(1).INITMB
  1       C*************************************************************************
  2       C       THIS PROGRAM INITIALIZES TEMP AND DENSITY  FOR MORNING
  3       C*************************************************************************
  4             SUBROUTINE INITMB(I,J,K,IN,JN,KN,IW,JW,INN,JWN,A,B,C,T,RO,
  5            CMAR,MRH,
  6            CTREF,RREF,
  7            CTO,HI)
  8             DIMENSION T(IN,JN,KN),RO(IN,JN,KN)
  9             DIMENSION MAR(IN,JN),MRH(IWN,JWN),HI(IN,JN)
 10             TOD=(TO-TREF)/TREF
 11             R=A+B*TO+C*TO*TO
 12             ROD=(R-RREF)/RREF
 13             READ 2,TSMN,DPMX
 14             DO 900 II=1,IN
 15             READ 2,I,(T(I,J,1),J=1,JN)
 16       900   CONTINUE
 17       2     FORMAT ()
 18             DO 1001 K=2,KN
 19             DO 1001 J=1,JN
 20             DO 1001 I=1,IN
 21             T(I,J,K)=(1.0+T(I,J,K))*TREF
 22             DP=HI(I,J)*FLOAT(K-1)/FLOAT(KN-1)
 23             IF(DP.LT.DPMX) T(I,J,K)=T(I,J,K)+((DPMX-DP)/DPMX)*(T(I,J,1)-TSMN)
 24       1001  CONTINUE
 25             DO 1002 K=1,KN
 26             DO 1002 J=1,JN
 27             DO 1002 I=1,IN
 28             RO(I,J,K)=A+B*T(I,J,K)+C*T(I,J,K)**2
 29             T(I,J,K)=(T(I,J,K)-TREF)/TREF
 30             RO(I,J,K)=(RO(I,J,K)-RREF)/RREF
 31       1002  CONTINUE
 32       11    CONTINUE
 33       12    CONTINUE
 34       10    CONTINUE
 35             RETURN
 36             END
```

9.2.24 INLET

This program inputs the velocities u and v at plume discharge into the model. It defines the inlet v-velocity by two constants AA and BB.

AA and BB are non-dimensionalized numbers. Non dimensionalized with respect to discharge velocity.

Therefore AA = 1.0

BB = 1.0

```
V*DOC.INLET
   1              SUBROUTINE INLET(I,J,K,IN,JN,KN,V,G,AA,BB)
   2              DIMENSION G(IN,JN,KN),V(IN,JN,KN)
   3              INM1=IN-1
   4              JNM1=JN-1
   5              KNM1=KN-1
   6              DO 100 K=1,KNM1
   7              V(9,1,K)=AA
   8              G(9,1,K)=AA
   9              V(10,1,K)=BB
  10              G(10,1,K)=BB
  11        100   CONTINUE
  12              RETURN
  13              END
```

### 9.2.25 INLETA

This subroutine reads in the number of inlet and outlet points, u, v and T at inlet points and u and v at outlet points for the far field unstratified model. This subroutine is called in by the main programs TMAIN5, TMAIN5T and TMAIN5V. The subroutine reads in data from element INDATA5, the lines following the first twelve lines.

```
M*DULL(1).INLETA
    1               SUBROUTINE INLETA(I,J,K,IN,JN,KN,U,V,H,G,T)
    2               DIMENSION H(IN,JN,KN),G(IN,JN,KN),U(IN,JN,KN)
    3               DIMENSION V(IN,JN,KN),T(IN,JN,KN)
    4               READ 2,NIN,NOUT
    5         2     FORMAT()
    6               DO 20 NN=1,NIN
    7               READ 2,I,J,K,U(I,J,K),V(I,J,K),T(I,J,K)
    8               H(I,J,K)=U(I,J,K)
    9               G(I,J,K)=V(I,J,K)
   10        20     CONTINUE
   11               DO 30 NN=1,NOUT
   12               READ 2,I,J,K,U(I,J,K),V(I,J,K)
   13               H(I,J,K)=U(I,J,K)
   14               G(I,J,K)=V(I,J,K)
   15        30     CONTINUE
   16               RETURN
   17               END
```

### 9.2.26 INLETB

This subroutine is used for reading in the information at open boundaries for the far field stratified model. The subroutine is the same as INLETA. The subroutine reads in the data from element DATAML5, the lines following the first 13 lines. The subroutine is called in by the main program TMAIN5B, TMAIN5TB and TMAIN5VB.

```
M*DULL(1).INLETB
   1              SUBROUTINE INLETB(I,J,K,IN,JN,KN,U,V,H,G,T)
   2              DIMENSION H(IN,JN,KN),G(IN,JN,KN),U(IN,JN,KN)
   3              DIMENSION V(IN,JN,KN),T(IN,JN,KN)
   4              READ 2,NIN,NOUT
   5         2,   FORMAT()
   6              DO 20 NN=1,NIN
   7              READ 2,I,J,K,U(I,J,K),V(I,J,K),T(I,J,K)
   8              H(I,J,K)=U(I,J,K)
   9              G(I,J,K)=V(I,J,K)
  10         2C   CONTINUE
  11              DO 3C NM=1,NOUT
  12              READ 2,I,J,K,U(I,J,K),V(I,J,K)
  13              H(I,J,K)=U(I,J,K)
  14              G(I,J,K)=V(I,J,K)
  15         3C   CONTINUE
  16              RETURN
  17              END
```

9.2.27 INTE

This subroutine computes XINT, YINT, DPSX and DPSY. This subroutine uses x-momentum equation to compute $\frac{\partial Ps}{\partial x}$.

$$\frac{\partial Ps}{\partial \alpha} = DPSX + \frac{1}{R_b} v - B_x$$

$$\frac{\partial Ps}{\partial \beta} = DPXY - \frac{1}{R_b} u - By$$

```
*DOC.INTE
      SUBROUTINE INTE(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,MAR,XINT,YINT,A3
     C,AI,AH,AV,TAUX,TAUY
     C,DX,DY,DZ,D,E,DT,DPSX,DPSY,AP)
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),MAR(IN,JN),HI(IN,JN)
      DIMENSION HX(IN,JN),HY(IN,JN)
      DIMENSION A3(KN)
      DIMENSION XINT(IN,JN),YINT(IN,JN)
      DIMENSION DPSX(IN,JN),DPSY(IN,JN)
      DIMENSION D(IN,JN,KN),E(IN,JN,KN)
      DO 200 I=1,IN
      DO 200 J=1,JN
      IF(MAR(I,J).EQ.0) GO TO 200
      YINT(I,J)=0.0
      XINT(I,J)=0.0
      DO 190 K=1,KN
      CALL DINERU(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1HUUX,D1HUVX,MAR)
      CALL DUVY(I,J,K,IN,JN,KN,U,V,HI,DY,D1HUVY,MAR)
      CALL DVVY(I,J,K,IN,JN,KN,U,V,HI,DY,D1HVVY,MAR)
      CALL DVISU(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1UX,D2UX,D1UY,D2UY,MAR)
      CALL DVISV(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1VX,D2VX,D1VY,D2VY,MAR)
      CALL D12Z(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
      DPSX(I,J)=(1./AP)*(1./HI(I,J))*(-XSUM*Q-HI(I,J)*D1UT)
      DPSY(I,J)=(1./AP)*(1./HI(I,J))*(-YSUM*Q-HI(I,J)*D1VT)
      GO TO 1100
 1010 CONTINUE
      XSUM=(AI*(D1HUUX+D1HUVY+HI(I,J)*D1UWZ)
     C-AH*(D2UX*HI(I,J)+D2UY*HI(I,J))
     C-AH*(D1UX*HX(I,J)+D1UY*HY(I,J))
     C-AV*(1.0/HI(I,J))*(A3(K)*D2UZ+D1A3Z*D1UZ))*DZ/2.0
      YSUM=(AI*(D1HUVX+D1HVVY+HI(I,J)*D1VWZ)
     C-AH*(D2VX*HI(I,J)+D2VY*HI(I,J))
     C-AH*(D1VX*HX(I,J)+D1VY*HY(I,J))
```

```
57              C-AV*(1.0/HI(I,J))*(A3(K)*D2VZ+D1A3Z*D1VZ))*DZ/2.0
58        110C  CONTINUE
59              XINT(I,J)=XSUM+XINT(I,J)
60              YINT(I,J)=YSUM+YINT(I,J)
61        190   CONTINUE
62        200   CONTINUE
63              RETURN
64              END
```

### 9.2.28 INTEB

This subroutine is used by the far field stratified model. This subroutine is similar to the subroutine INTE with the difference that it calls the subroutine VERTDF which computes the vertical viscosity and its derivative for every grid point.

```
(M*DULL(1).INTEB
     1              SUBROUTINE INTEB(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,MAR,XINT,YINT,A3
     2             C,AI,AH,AV,TAUX,TAUY
     3             C,DX,DY,DZ,D,E,DT,DPSX,DPSY,AP,T,TREF,CONS,AVMX,AVMN)
     4              DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),MAR(IN,JN),HI(IN,JN)
     5              DIMENSION HX(IN,JN),HY(IN,JN),T(IN,JN,KN)
     6              DIMENSION A3(KN)
     7              DIMENSION XINT(IN,JN),YINT(IN,JN)
     8              DIMENSION DPSX(IN,JN),DPSY(IN,JN)
     9              DIMENSION D(IN,JN,KN),E(IN,JN,KN)
    10              DO 200 I=1,IN
    11              DO 200 J=1,JN
    12              IF(MAR(I,J).EQ.0) GO TO 200
    13              YINT(I,J)=0.0
    14              XINT(I,J)=0.0
    15              DO 190 K=1,KN
    16              CALL DINEFU(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1HUUX,D1HUVX,MAR)
    17              CALL DUVY(I,J,K,IN,JN,KN,U,V,HI,DY,D1HUVY,MAR)
    18              CALL DVVY(I,J,K,IN,JN,KN,U,V,HI,DY,D1HVVY,MAR)
    19              CALL DVISU(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1UX,D2UX,D1UY,D2UY,MAR)
    20              CALL DVISV(I,J,K,IN,JN,KN,U,V,HI,DX,DY,D1VX,D2VX,D1VY,D2VY,MAR)
    21              CALL D12Z(I,J,K,IN,JN,KN,U,V,W,HI,HX,HY,DX,DY,DZ,D1UWZ,A3,
    22             CTAUX,TAUY,D1VWZ,D1UZ,D2UZ,D1VZ,D2VZ,D1A3Z)
    23              CALL VERTDF(I,J,K,IN,JN,KN,HI,AB3,D1A3Z,D1B3Z,DZ,T,A3,TREF
    24             C,CONS,AVMX,AVMN)
    25              A3(K)=AB3
    26              IF (K.EQ.1) GO TO 1000
    27              IF (K.EQ.KN) GO TO 1010
    28              XSUM=(AI*(D1HUUX+D1HUVY+HI(I,J)*D1UWZ)
    29             C-AH*(D2UX*HI(I,J)+D2UY*HI(I,J))
    30             C-AH*(D1UX*HX(I,J)+D1UY*HY(I,J))
    31             C-AV*(1.0/HI(I,J))*(A3(K)*D2UZ+D1A3Z*D1UZ))*DZ
    32              YSUM=(AI*(D1HUVX+D1HVVY+HI(I,J)*D1VWZ)
    33             C-AH*(D2VX*HI(I,J)+D2VY*HI(I,J))
    34             C-AH*(D1VX*HX(I,J)+D1VY*HY(I,J))
    35             C-AV*(1.0/HI(I,J))*(A3(K)*D2VZ+D1A3Z*D1VZ))*DZ
    36              GO TO 1100
    37        1000  CONTINUE
    38              XSUM=(AI*(D1HUUX+D1HUVY+HI(I,J)*D1UWZ)
    39             C-AH*(D2UX*HI(I,J)+D2UY*HI(I,J))
    40             C-AH*(D1UX*HX(I,J)+D1UY*HY(I,J))
    41             C-AV*(1.0/HI(I,J))*(A3(K)*D2UZ+D1A3Z*D1UZ))*DZ/2.0
    42              YSUM=(AI*(D1HUVX+D1HVVY+HI(I,J)*D1VWZ)
    43             C-AH*(D2VX*HI(I,J)+D2VY*HI(I,J))
    44             C-AH*(D1VX*HX(I,J)+D1VY*HY(I,J))
    45             C-AV*(1.0/HI(I,J))*(A3(K)*D2VZ+D1A3Z*D1VZ))*DZ/2.0
    46              D1UT=(U(I,J,K)-D(I,J,K))/DT
    47              D1VT=(V(I,J,K)-E(I,J,K))/DT
    48              Q=2.0/DZ
    49              DPSX(I,J)=(1./AP)*(1./HI(I,J))*(-XSUM*Q-HI(I,J)*D1UT)
    50              DPSY(I,J)=(1./AP)*(1./HI(I,J))*(-YSUM*Q-HI(I,J)*D1VT)
    51              GO TO 1100
    52        1010  CONTINUE
    53              XSUM=(AI*(D1HUUX+D1HUVY+HI(I,J)*D1UWZ)
    54             C-AH*(D2UX*HI(I,J)+D2UY*HI(I,J))
    55             C-AH*(D1UX*HX(I,J)+D1UY*HY(I,J))
    56             C-AV*(1.0/HI(I,J))*(A3(K)*D2UZ+D1A3Z*D1UZ))*DZ/2.0
```

```
57                YSUM=(AI*(D1HUVX*D1HVVY*HI(I,J)*D1VWZ)
58               C-AH*(D2VX*HI(I,J)+D2VY*HI(I,J))
59               C-AH*(D1VX*HX(I,J)+D1VY*HY(I,J))
60               C-AV*(1.0/HI(I,J))*(A3(K)*D2VZ+D1A3Z*D1VZ))*DZ/2.0
61          1100  CONTINUE
62                XINT(I,J)=XSUM+XINT(I,J)
63                YINT(I,J)=YSUM+YINT(I,J)
64          190   CONTINUE
65          200   CONTINUE
66                RETURN
67                END
```

9.2.29 INTEMP

This program inputs discharge temperature into the model at plume discharge.

TLL and TMM are non-dimensionalized temperatures. Non dimensionalized with respect to reference temperature as shown below

$$TLL = TMM = \frac{T - Tref}{Tref}$$

```
*DOC.INTEMP
  1                SUBROUTINE INTEMP(I,J,K,IN,JN,KN,T,TD,TLL,TMM)
  2                DIMENSION T(IN,JN,KN),TD(IN,JN,KN)
  3                INM1=IN-1
  4                JNM1=JN-1
  5                KNM1=KN-1
  6                DO 100 K=1,KN
  7                T(9,1,K)=TLL
  8                TD(9,1,K)=TLL
  9                T(10,1,K)=TMM
 10                TD(10,1,K)=TMM
 11          100   CONTINUE
 12                RETURN
 13                END
```

9.2.30 OLDT

This program 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.

```
)OC.OLDT
 1              SUBROUTINE OLDT(I,J,K,IN,JN,KN,T,TP)
 2              DIMENSION T(IN,JN,KN),TP(IN,JN,KN)
 3              DO 10 I=1,IN
 4              DO 10 J=1,JN
 5              DO 10 K=1,KN
 6              TP(I,J,K)=T(I,J,K)
 7        10    CONTINUE
 8              RETURN
 9              END
```

9.2.31 OLDUV

This program 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.

```
DOC.OLDUV
 1      C**********************************************************************
 2      C     THIS PROGRAM SETS THE VALUES OF D AND E EQUAL TO U AND V RESPECTIVELY
 3      C         IN ORDER TO RETAIN VALUES OF U AND V AT ONE TIME STEP LAG
 4      C**********************************************************************
 5            SUBROUTINE OLDUV(I,J,K,IN,JN,KN,U,V,D,E)
 6            DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN)
 7            DO 831 K=1,KN
 8            DO 831 I=1,IN
 9            DO 831 J=1,JN
10            D(I,J,K)=U(I,J,K)
11            E(I,J,K)=V(I,J,K)
12      831   CONTINUE
13            RETURN
14            END
```

9.2.32 OUTEMP

This subroutine sets the boundary conditions for temperature at the outlets. It sets near field outlet temperatures equal to those at the adjacent grid.

Eg: $T_{IN-1} = T_{IN}$

```
)OC.OUTEMP
 1              SUBROUTINE OUTEMP(I,J,K,IN,JN,KN,TD)
 2              DIMENSION TD(IN,JN,KN)
 3              INM1=IN-1
 4              JNM1=JN-1
 5              KNM1=KN-1
 6              DO 200 K=1,KNM1
 7              DO 200 J=1,JN
 8              TD(1,J,K)=TD(2,J,K)
 9       200    CONTINUE
10              DO 300 K=1,KNM1
11              DO 300 J=1,JN
12              TD(IN,J,K)=TD(INM1,J,K)
13       300    CONTINUE
14              DO 400 K=1,KNM1
15              DO 400 I=2,INM1
16              TD(I,JN,K)=TD(I,JNM1,K)
17       400    CONTINUE
18              RETURN
19              END
```

9.2.33 OUTVEL

This program sets the boundary conditions at the outlets for velocity. In other words, it sets near field outlet velocities equal to those at the adjacent grid for U and V at the boundaries where there is no current. This implies that gradients normal to open boundary are equal to zero.

Eg: $U_{IN-1} = U_{IN}$

DOC.OUTVEL

```
 1              SUBROUTINE OUTVEL(I,J,K,IN,JN,KN,H,G)
 2              DIMENSION H(IN,JN,KN),G(IN,JN,KN)
 3              INM1=IN-1
 4              JNM1=JN-1
 5              KNM1=KN-1
 6              DO 200 K=1,KNM1
 7              DO 200 J=1,JN
 8              H(1,J,K)=H(2,J,K)
 9              G(1,J,K)=G(2,J,K)
10       200 - CONTINUE
11              DO 300 K=1,KNM1
12              DO 300 J=1,JN
13              H(IN,J,K)=H(INM1,J,K)
14              G(IN,J,K)=G(INM1,J,K)
15       300    CONTINUE
16              DO 400 K=1,KNM1
17              DO 400 I=2,INM1
18              H(I,JN,K)=H(I,JNM1,K)
19              G(I,JN,K)=G(I,JNM1,K)
20       400    CONTINUE
21              RETURN
22              END
```

### 9.2.34 PDPSXY

This program prints $\frac{\partial ps}{\partial \alpha}$ and $\frac{\partial ps}{\partial \beta}$ at main grid points.

```
I*DOC.PDPSXY
  1             SUBROUTINE PDPSXY(I,J,IN,JN,DPSX,DPSY)
  2             DIMENSION DPSX(IN,JN),DPSY(IN,JN)
  3             DO 10 I=1,IN
  4             PRINT 1,I,(DPSX(I,J),J=1,JN)
  5             PRINT 2,(DPSY(I,J),J=1,JN)
  6       1     FORMAT(/' I=',I3/' DPSX'/(5X,8E15.7))
  7       2     FORMAT(' DPSY'/(5X,8E15.7))
  8       10    CONTINUE
  9             RETURN
 10             END
```

### 9.2.35 POTUV

This program plots surface velocities for the near field.

```
*DOC.POTUV
      C       PLOTS U AND V ON CONSTANT DEPTH SECTIONS
              PARAMETER IN=18,JN=21,IWN=17,JWN=20,KN=5,KNM1=4
             DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
            CWH(IWN,JWN,KN),W(IN,JN,KN),WR(IN,JN,KN),WRH(IWN,JWN,KN),
            CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN)
             DIMENSION TW(IWN,JWN,KN),RO(IN,JN,KN),PINTH(IWN,JWN),ROW(IWN,JWN,
            CKN),T(IN,JN,KN)
             DIMENSION IBUF(1000)
             READ 1, IRUN
             READ 1,USCALE,VSCALE
              ARMIN=0.04
              ARMAX=0.15
             IF(IRUN. EQ.0) GO TO 4
             IF(IRUN. EQ.1) GO TO 5
      4       CALL READ1(U,V,WH,PINTH,I,J,K,IW,JW,IN,JN,KN,
            CIWN,JWN,D,E,HX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,
            CDY,DZ,DT,TAUX,YAUY,W,WR,WRH,TTOT)
             GO TO 6
      5       CALL READ2(U,V,WH,PINTH,I,J,K,IW,JW,IN,JN,KN,
            CIWN,JWN,D,E,HX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,
            CDY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT,
            CCB,CW,A,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,
            CTAMB,TTOT)
      6       CONTINUE
              CALL PLOTS(IBUF,1000,11)
              CALL FACTOR(0.25)
      1       FORMAT ()
              DO 10 K=1,KN
              IF(K.GT.1) GO TO 20
              DO 30 I=1,IN
              DO 30 J=1,JN
              IF (MAR(I,J).EQ.0) GO TO 35
              AI=(I-1)*1.0
              AJ=(J-1)*1.0
              AAI=AI+U(I,J,K)*USCALE
              AAJ=AJ+V(I,J,K)*VSCALE
              YW=0.2*SQRT((AAI-AI)**2+(AAJ-AJ)**2)
              YW=AMAX1(ARMIN/0.25,AMIN1(YW,ARMAX/0.25))
              CALL AROHD(AI,AJ,AAI,AAJ,YW,0.0,12)
      35      CONTINUE
      30      CONTINUE
              GO TO 100
      20      CONTINUE
              DEPTH=(1.0/KNM1)*(K-1)
              DO 40 I=1,IN
              DO 40 J=1,JN
              IF(HI(I,J).GT.DEPTH) GO TO 45
              GO TO 50
      45      CONTINUE
      50      DDZ=HI(I,J)/KNM1
              LD1=(DEPTH/HI(I,J))*KNM1
              IF(LD1.EQ.0) GO TO 55
              LD2=LD1+1
              LD3=LD1+2
              DIFF=(DEPTH-LD1*DDZ)
      C         COEFFS OF SECOND DEGREE FIT
```

```
      U1=U(I,J,LD1)
      U2=U(I,J,LD2)
      U3=U(I,J,LD3)
      V1=V(I,J,LD1)
      V2=V(I,J,LD2)
      V3=V(I,J,LD3)
      AU=(U3-2*U2+U1)/(2*DDZ*DDZ)
      BU=(4*U2-3*U1-U3)/(2*DDZ)
      CU=U1
      AV=(V3-2*V2+V1)/(2*DDZ*DDZ)
      BV=(4*V2-3*V1-V3)/(2*DDZ)
      CV=V1
      AZ=DDZ*DIFF
      UDEPTH=AU*AZ*AZ+BU*AZ+CU
      VDEPTH=AV*AZ*AZ+BV*AZ+CV
      GO TO 60
   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
      A=.2*USCALE
      B=16.0
      C=B+A
      CALL PLOT(B,0.0,3)
      CALL PLOT(C,0.0,2)
      CALL PLOT(C,-0.2,2)
      CALL PLOT(B,-0.2,2)
      CALL PLOT(B,0.0,2)
      CALL PLOT(50.0,0.0,-3)
   10 CONTINUE
      END
```

### 9.2.36 POTUW

This program plots the vertical section velocities perpendicular to the discharge for the near-field.

```
*DOC.POTUW
      PARAMETER IN=18,JN=21,KN=5,IWN=17,JWN=20,KNM1=4
      DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
     CWH(IWN,JWN,KN),W(IN,JN,KN),WR(IN,JN,KN),WRH(IWN,JWN,KN),
     CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),MRH(IWN,JWN)
      DIMENSION IW(IWN,JWN,KN),RO(IN,JN,KN),PINTH(IWN,JWN),ROW(IWN,JWN,
     CKN),T(IN,JN,KN)
      DIMENSION IBUF(1000)
      READ 1, IRUN
      IF(IRUN.EQ.0) GO TO 4
      IF(IRUN.EQ.1) GO TO 5
    4 CALL READ1(U,V,WH,PINTH,I,J,K,IW,JW,IN,JN,KN,
     CIWN,JWN,D,E,HX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,
     CDY,DZ,DT,TAUX,YAUY,W,WR,WRH,TTOT)
      GO TO 6
    5 CALL READ2(U,V,WH,PINTH,I,J,K,IW,JW,IN,JN,KN,
     CIWN,JWN,D,E,HX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,
     CDY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT,CB,CW,
     CA,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,
     CTAMB,TTOT)
    6 CONTINUE
      CALL PLOTS(IBUF,1000,11)
      CALL FACTOR(0.25)
      READ 1, USCALE,VSCALE,WSCALE,HBYL
      ARMIN=0.04
      ARMAX=0.15
    1 FORMAT ()
      DO 10 I=1,IN
      DO 20 J=1,JN
      IF (MAR(I,J).LT.11) GO TO 20
      AJ=(J-1)*1.0
      DO 15 K=1,KNM1
      AK=-(K-1)*HI(I,J)
      AAJ=AJ+V(I,J,K)*VSCALE
      AAK=AK-W(I,J,K)*WSCALE*HBYL
      YW=0.2*SQRT((AAJ-AJ)**2+(AAK-AK)**2)
      YW=AMAX1(ARMIN/0.25,AMIN1(YW,ARMAX/0.25))
      CALL AROHD(AJ,AK,AAJ,AAK,YW,0.0,12)
   15 CONTINUE
   20 CONTINUE
    C DRAWS BOTTOM SURFACE
      NN=0
      DO 30 J=1,JN
      IF (MAR(I,J).EQ.0) GO TO 32
      NN=NN+1
      IF (NN.GT.1) GO TO 33
      AAJ=(J-1)*1.0
      AAK=-HI(I,J)*KNM1
      CALL PLOT(AAJ,0.0,3)
      CALL PLOT(AAJ,AAK,2)
      GO TO 32
   33 CONTINUE
      AAJ=(J-1)*1.0
      AAK=-HI(I,J)*KNM1
      CALL PLOT(AAJ,AAK,2)
      JD=J
      AJD=(JD-1)*1.0
```

```
57       32     CONTINUE
58       30     CONTINUE
59              CALL PLOT(AJD,0.0,3)
60              AAK=-HI(I,JD)*KNM1
61              CALL PLOT(AJD,AAK,2)
62              A=0.2*USCALE
63              B=16.0
64              C=B+A
65              CALL PLOT(B,0.0,3)
66              CALL PLOT(C,0.0,2)
67              CALL PLOT(C,-0.2,2)
68              CALL PLOT(B,-0.2,2)
69              CALL PLOT(B,0.0,2)
70              CALL PLOT(30.0,0.0,-3)
71       10     CONTINUE
72              END
```

9.2.37 POTVW

This program plots the vertical section velocities along canal centerline for near-field.

```
I*DOC.POTVL
  1       C     PLOTS VELOCITIES IN J SECTIONS
  2             PARAMETER  IN=18,JN=21,IWN=17,JWN=20,KN=5,KNM1=4
  3             DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
  4            CWH(IWN,JWN,KN),PINTH(IWN,JWN)
  5             DIMENSION HX(IN,JN),HY(IN,JN),HI(IN,JN),MAR(IN,JN),MRH(IWN,JWN),
  6            CW(IN,JN,KN),WR(IN,JN,KN),WRH(IN,JN,KN)
  7             DIMENSION T(IN,JN,KN),RO(IN,JN,KN),TW(IWN,JWN,
  8            CKN),ROW(IWN,JWN,KN)
  9             DIMENSION IBUF(1000)
 10             READ 1, IRUN
 11             IF(IRUN.EQ.0) GO TO 4
 12             IF(IRUN.EQ.1) GO TO 5
 13       4     CALL READ1(U,V,WH,PINTH,I,J,K,IW,JW,IN,JN,KN,
 14            CIWN,JWN,D,E,HX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,
 15            CDY,DZ,DT,TAUX,TAUY,W,WR,WRH,TTOT)
 16             GO TO 6
 17       5     CALL READ2(U,V,WH,PINTH,I,J,K,IW,JW,IN,JN,KN,
 18            CIWN,JWN,D,E,HX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,
 19            CDY,DZ,DT,TAUX,TAUY,W,WR,WRH,TAI,TAH,TAV,AKT,CB,CW,
 20            CA,B,C,EUL,T,TW,PO,ROW,TE,RREF,TREF,TO,
 21            CTAMB,TTOT)
 22       6     CONTINUE
 23             CALL PLOTS(IBUF,1000,11)
 24             CALL FACTOR(0.25)
 25             READ 1, USCALE,VSCALE,WSCALE,HBYL
 26             ARMIN=0.04
 27             ARMAX=0.15
 28       1     FORMAT ()
 29             DO 10 J=1,JN
 30             DO 20 I=1,IN
 31             IF (MAR(I,J).LT.11) GO TO 20
 32             AI=(I-1)*1.0
 33             DO 30 K=1,KNM1
 34             AK=-(K-1)*HI(I,J)
 35             AAI=AI+U(I,J,K)*USCALE
 36             W(I,1,K)=0.0
 37             AAK=AK-W(I,J,K)*WSCALE*HBYL
 38             YW=0.2*SQRT((AAI-AI)**2+(AAK-AK)**2)
 39             YW=AMAX1(ARMIN/0.25,AMIN1(YW,ARMAX/0.25))
 40             CALL AROHD(AI,AK,AAI,AAK,YW,0.0,12)
 41       30    CONTINUE
 42       20    CONTINUE
 43       C     DRAWS BOTTOM SURFACE
 44             NN=0
 45             DO 35 I=1,IN
 46             IF(MAR(I,J).EQ.0) GO TO 40
 47             NN=NN+1
 48             IF (NN.GT.1) GO TO 33
 49             AAI=(I-1)*1.0
 50             CALL PLOT(AAI,0.0,3)
 51              AAK=-HI(I,J)*KNM1
 52             CALL PLOT(AAI,AAK,2)
 53             GO TO 40
 54       33    CONTINUE
 55             AAI=(I-1)*1.0
 56             AAK=-HI(I,J)*KNM1
```

```
57              CALL PLOT(AAI,AAK,2)
58              ID=I
59              AID=(ID-1)*1.0
60        40    CONTINUE
61        35    CONTINUE
62              CALL PLOT(AID,0.0,3)
63              AAK=-HI(I,J)*KNM1
64              CALL PLOT(AID,AAK,2)
65              A=.2*USCALE
66              B=16.0
67              C=B*A
68              CALL PLOT(B,0.0,3)
69              CALL PLOT(C,0.0,2)
70              CALL PLOT(C,-.2,2)
71              CALL PLOT(B,-.2,2)
72              CALL PLOT(B,0.0,2)
73              CALL PLOT(50.0,0.0,-3)
74        10    CONTINUE
75              END
```

9.2.38 PRPARA

This subroutine prints values computed after one time step. Quantities printed are AI, AH, AV, AP, DX, DY, DZ, DT, DL2, MAXIT, EPS, OMEGA, ARBP, TAUX, TAUY and TTOT.

```
)OC,PRPARA
 1              SUBROUTINE PRPARA(AI,AH,AV,AP,DX,DY,DZ,DT,DL2,MAXIT,EPS,
 2             COMEGA,ARBP,TAUX,TAUY,TTOT,MAR,MRH,IN,JN,IWN,JWN)
 3              DIMENSION MAR(IN,JN),MRH(IWN,JWN)
 4              PRINT 1, AI,AH,AV,AP,DX,DY,DZ,DT,DL2,MAXIT,EPS,OMEGA,
 5             CARBP,TAUX,TAUY,TTOT
 6       1      FORMAT (/' AI=',E15.7,/' AH=',E15.7,/' AV=',E15.7,/' AP=',E15.7,
 7             C/' DX=',E15.7,/' DY=',E15.7,/' DZ=',E15.7,/' DT=',E15.7,/' DL2=',
 8             CE15.7,/' MAXIT=',I5,/' EPS=',E15.7,/' OMEGA=',E15.7,/' ARBP=',
 9             CE15.7,/' TAUX=',E15.7,/' TAUY=',E15.7,/' TTOT=',E15.7/)
10              DO400J=1,JN
11              JJ=JN+1-J
12              PRINT 700,(MAR(I,JJ),I=1,IN)
13       400    CONTINUE
14              DO600JW=1,JWN
15              JJW=JWN+1-JW
16              PRINT 800,(MRH(IW,JJW),IW=1,IWN)
17       600    CONTINUE
18       700    FORMAT(/,'  MAR ',(3X,29I3,/))
19       800    FORMAT(/,'  MRH ',(3X,28I3,/))
20              RETURN
21              END
```

9.2.39 PRE1

This program computes pressure for far field from Poisson equation. An iterative scheme is employed. Values are computed at half grid points.

```
:M*DULL(1),PRE1
    1              SUBROUTINE PRE1(EPS,MAXIT,IN,JN,P,ITN,DPSX,DPSY,FH,DL2,OMEGA,
    2             CMRH,I,J,K,IW,JW,DX,DY,EX,IWN,JWN,ARBP)
    3              DIMENSION P(IWN,JWN),FH(IWN,JWN),DPSX(IN,JN),DPSY(IN,JN)
    4              DIMENSION MRH(IWN,JWN)
    5              ITN=0
    6        1     EX=0.0
    7              ITN=ITN+1
    8              DO 10 IWD=1,IWN
    9              DO 10 JW=1,JWN
   10              IW=(IWN+1)-IWD
   11              I=IW
   12              J=JW
   13              IF (MRH(IW,JW).EQ.0) GO TO 57
   14              IF (MRH(IW,JW).EQ.1) GO TO 11
   15              IF (MRH(IW,JW).EQ.2) GO TO 12
   16              IF (MRH(IW,JW).EQ.3) GO TO 13
   17              IF (MRH(IW,JW).EQ.4) GO TO 14
   18              IF (MRH(IW,JW).EQ.5) GO TO 18
   19              IF (MRH(IW,JW).EQ.6) GO TO 16
   20              IF (MRH(IW,JW).EQ.7) GO TO 17
   21              IF (MRH(IW,JW).EQ.8) GO TO 18
   22              IF (MRH(IW,JW).EQ.10) GO TO 19
   23              PN=.25*(P(IW-1,JW)+P(IW+1,JW)+P(IW,JW-1)+P(IW,JW+1)-DL2*FH(IW,JW))
   24              GO TO 50
   25        11    CONTINUE
   26              PN=.25*(P(IW-1,JW)+P(IW+1,JW)+P(IW,JW-1)+P(IW,JW)+(DPSY(I,J+1)
   27             C+DPSY(I+1,J+1))*DY/2.-DL2*FH(IW,JW))
   28              GO TO 50
   29        12    CONTINUE
   30              PN=.25*(P(IW-1,JW)+P(IW+1,JW)+P(IW,JW+1)+P(IW,JW)
   31             C-(DPSY(I,J)+DPSY(I+1,J))*DY/2-DL2*FH(IW,JW))
   32              GO TO 50
   33        13    CONTINUE
   34              PN=.25*(P(IW+1,JW)+P(IW,JW-1)+P(IW,JW)-(DPSX(I,J)+DPSX(I,J+1))
   35             C*DX/2+P(IW,JW)+(DPSY(I,J+1)+DPSY(I+1,J+1))*DY/2-DL2*FH(IW,JW))
   36              GO TO 50
   37        14    CONTINUE
   38              PN=.25*(P(IW+1,JW)+P(IW,JW+1)+P(IW,JW)-(DPSX(I,J)+DPSX(I,J+1))*
   39             CDX/2+P(IW,JW)-(DPSY(I,J)+DPSY(I+1,J))*DY/2-DL2*FH(IW,JW))
   40              GO TO 50
   41        15    CONTINUE
   42              PN=ARBP
   43              GO TO 50
   44        16    CONTINUE
   45              PN=.25*(P(IW,JW+1)+P(IW-1,JW)+P(IW,JW)+(DPSX(I+1,J+1)+DPSX(I+1,J))
   46             C*DX/2+P(IW,JW)-(DPSY(I,J)+DPSY(I+1,J))*DY/2-DL2*FH(IW,JW))
   47              GO TO 50
   48        17    CONTINUE
   49              PN=.25*(P(IW-1,JW)+P(IW,JW-1)+P(IW,JW)+(DPSX(I+1,J)+DPSX(I+1,J+1))
   50             C*DX/2+P(IW,JW)+(DPSY(I,J+1)+DPSY(I+1,J+1))*DY/2-DL2*FH(IW,JW))
   51              GO TO 50
   52        18    CONTINUE
   53              PN=.25*(P(IW,JW+1)+P(IW-1,JW)+P(IW,JW-1)+P(IW,JW)+(DPSX(I+1,J)
   54             C+DPSX(I+1,J+1))*DX/2-DL2*FH(IW,JW))
   55              GO TO 50
   56        19    CONTINUE
```

```
      PN=0.25*(P(IW,JW-1)+P(IW,JW+1)+P(IW+1,JW)+P(IW,JW)
     C-(DPSX(I,J)+DPSX(I,J+1))*DX/2-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
      HT=P(1,6)-ARBP
      DO 30 IW=1,IWN
      DO 30 JW=1,JWN
      IF (MRH(IW,JW).EQ.0) GO TO 31
      P(IW,JW)=P(IW,JW)-HT
   31 CONTINUE
   30 CONTINUE
      IF(EX.LT.EPS) GO TO 20
      IF(ITN.LT.MAXIT) GO TO 1
   20 CONTINUE
      RETURN
      END
```

9.2.40 PRE2

This program computes surface pressure for near field from Poisson equation. An iterative scheme is employed. Values are computed at half grid points.

```
*DOC.PRE2
      SUBROUTINE PRE2(EPS,MAXIT,IN,JN,P,ITN,DPSX,DPSY,FH,DL2,OMEGA,
     1MRH,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
      ITN=ITN+1
      DD=ARBP
      DO 10 IWD=1,IWN
      DO 10 JW=1,JWN
      IW=(IWN+1)-IWD
      I=IW
      J=JW
      IF (MRH(IW,JW).EQ.11) GO TO 2
      IF (MRH(IW,JW).EQ.1) GO TO 11
      IF (MRH(IW,JW).EQ.2) GO TO 12
      IF (MRH(IW,JW).EQ.3) GO TO 13
      IF (MRH(IW,JW).EQ.4) GO TO 14
      IF (MRH(IW,JW).EQ.5) GO TO 18
      IF (MRH(IW,JW).EQ.6) GO TO 16
      IF (MRH(IW,JW).EQ.7) GO TO 17
      IF (MRH(IW,JW).EQ.8) GO TO 18
      IF (MRH(IW,JW).EQ.9) GO TO 19
      IF (MRH(IW,JW).EQ.10) GO TO 20
    2 CONTINUE
      PN=.25*(P(IW-1,JW)+P(IW+1,JW)+P(IW,JW-1)+P(IW,JW+1)-DL2*FH(IW,JW))
      GO TO 50
   11 CONTINUE
      PN=.25*(P(IW-1,JW)+P(IW+1,JW)+P(IW,JW-1)+DD-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)+2*DD-DL2*FH(IW,JW))
      GO TO 50
   14 CONTINUE
      PN=.25*(P(IW,JW+1)+P(IW+1,JW)+2*DD-DL2*FH(IW,JW))
      GO TO 50
   16 CONTINUE
       PN=.25*(P(IW,JW+1)+P(IW-1,JW)+2*DD-DL2*FH(IW,JW))
      GO TO 50
   17 CONTINUE
      PN=.25*(P(IW,JW-1)+P(IW-1,JW)+2*DD-DL2*FH(IW,JW))
      GO TO 50
   18 CONTINUE
      PN=.25*(P(IW-1,JW)+P(IW,JW-1)+P(IW,JW+1)+DD-DL2*FH(IW,JW))
      GO TO 50
   19 CONTINUE
      PN=.25*(P(IW-1,JW)+P(IW+1,JW)+P(IW,JW-1)+P(IW,JW+1)-DL2*FH(IW,JW))
      GO TO 50
   20 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)
```

```
57              IF(ABS(PNEW).LT.(10.**-16.)) GO TO 51
58              DIFF=ABS((PNEW-P(IW,JW))/PNEW)
59              IF (DIFF.LT.EX) GO TO 51
60              EX=DIFF
61        51    P(IW,JW)=PNEW
62        10    CONTINUE
63              IF(EX.LT.EPS) GO TO 32
64              IF(ITN.LT.MAXIT) GO TO 1
65        32    CONTINUE
66              RETURN
67              END
```

9.2.41 PREROR

This subroutine prints Hirt and Harlow correction term, (WHLDT)

```
*DOC.PREROR
  1       C**********************************************************************
  2       C      THIS PROGRAM PRINTS THE HIRT AND HARLOW CORRECTION TERM
  3       C**********************************************************************
  4             SUBROUTINE PREROR(IWN,IW,WHLDT,JW,JWN,IN,JN)
  5             DIMENSION WHLDT(IWN,JWN)
  6             DO 7500 IW=1,IWN
  7       7500  PRINT*7510,IW,(WHLDT(IW,JW),JW=1,JWN)
  8       7510  FORMAT(/' IW=',I3,/' WHLDT'/(5X,8E15.7))
  9             RETURN
 10             END
```

9.2.42 PRINTE

This subroutine prints XINT and YINT used in computing surface pressure.

```
*DOC.PRINTE
  1       C*************************************************************************
  2       C     THIS PROGRAM PRINTS THE INTEGRATED VALUES OF COMBINED INERTIA AND
  3       C          AND VISCOUS TERMS IN U AND V MOMENTUM EQUATIONS
  4       C*************************************************************************
  5             SUBROUTINE PRINTE(I,J,IW,JW,XINT,YINT,IN,JN)
  6             DIMENSION XINT(IN,JN),YINT(IN,JN)
  7             DO 8000 I=1,IN
  8             PRINT 8100,I,(XINT(I,J),J=1,JN)
  9       8000  PRINT 8200,(YINT(I,J),J=1,JN)
 10       8100  FORMAT(/' I=',I3,/' XINT'/(5X,8E15.7))
 11       8200  FORMAT(' YINT'/(5X,8E15.7))
 12             RETURN
 13             END
```

### 9.2.43 PREPINT

This subroutine prints PINTH which is surface pressure.

```
*DOC.PRPINT
  1       C**********************************************************************
  2       C     THIS PROGRAM PRINTS THE VALUE OF PINTH OVER THE SURFACE
  3       C**********************************************************************
  4             SUBROUTINE PRPINT(IW,JW,IWN,JWN,PINTH)
  5             DIMENSION PINTH(IWN,JWN)
  6       5200  DO 5300 IW=1,IWN
  7             PRINT 5400,IW,(PINTH(IW,JW),JW=1,JWN)
  8       5400  FORMAT(/' IW=',I3,/(5X,8E15.7))
  9       5300  CONTINUE
 10             RETURN
 11             END
```

### 9.2.44 PRITEX

This subroutine prints the number of iterations and final residual error in solving Poisson equation.

```
*DOC.PRITEX
  1       C*************************************************************************
  2       C      THIS PROGRAM PRINTS OUT THE VALUES OF NUMBER OF ITERATIONS ANDFINAL
  3       C                           RESIDUAL ERROR IN SOLVING POISSON
  4       C*************************************************************************
  5             SUBROUTINE PRITEX(ITN,EX)
  6             PRINT 5500,ITN,EX
  7       5500  FORMAT(/' ITN=',I4,5X,' EX=',E15.7)
  8             RETURN
  9             END
```

9.2.45 PRSORC

This subroutine prints source term (Right hand side) in Poissons equation for surface pressure. (Eq. 2.17 Vol.1)

```
*DOC.PRSORC
  1       C*************************************************************************
  2       C     THIS PROGRAM PRINTS THE VALUE OF THE SOURCE TERM IN POISSON EQUATION
  3       C*************************************************************************
  4             SUBROUTINE PRSORC(IW,JW,IWN,JWN,F)
  5             DIMENSION F(IWN,JWN)
  6             DO 6000 IW=1,IWN
  7       6000  PRINT 6100,IW,(F(IW,JW),JW=1,JWN)
  8       6100  FORMAT(/' IW=',I3/' SOURCE TERM'/(5X,8E15.7))
  9             RETURN
 10             END
```

9.2.46 PRUV

This subroutine prints the values of U and V at all main grid points.

```
I*DOC.PRUV
  1       C*************************************************************************
  2       C     THIS PROGRAM PRINTS THE VULUES OF U AND V AT GRID POINTS IN THE DOMAIN
  3       C*************************************************************************
  4             SUBROUTINE PRUV(I,J,K,IN,JN,KN,U,V)
  5             DIMENSION U(IN,JN,KN),V(IN,JN,KN)
  6             DO 9100 K=1,KN
  7             DO 9100 I=1,IN
  8             PRINT 9000,K,I,(U(I,J,K),J=1,JN)
  9       9100  PRINT 9200,(V(I,J,K),J=1,JN)
 10       9000  FORMAT(/' K=',I3,3X,'I=',I3/' U-VELOCITY'/(5X,8E15.7))
 11       9200  FORMAT(' V-VELOCITY'/(5X,8E15.7))
 12             RETURN
 13             END
```

9.2.47 PRWH

This subroutine prints the values of the vertical velocities (WH) at all half grid points.

```
*DOC.PRWH
  1       C*************************************************************************
  2       C      THIS PROGRAM PRINTS THE VALUS OF WH IN THE DOMAIN
  3       C*************************************************************************
  4             SUBROUTINE PRWH(IW,JW,K,IWN,JWN,KN,WH)
  5             DIMENSION WH(IWN,JWN,KN)
  6             DO 7000 K=1,KN
  7             DO 7000 IW=1,IWN
  8       7000  PRINT 7100,K,IW,(WH(IW,JW,K),JW=1,JWN)
  9       7100  FORMAT(/' K=',I3,3X,' IW=',I3,/' WH-VELOCITY'/(5X,8E15.7))
 10             RETURN
 11             END
```

### 9.2.48 READ2

This subroutine reads in input papameters and physical quantities stored on file designated by Unit 7. Store 2 and Read 2 correspond to each other.

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

### 9.2.49 READ2A

This is a subroutine used to read information stored by the subroutine STOR2A, for a far field unstratified model. It differs primarily from READ2 in that it does not include the matrix ROW (the densities at half grid points).

```
KM*DULL(1).READ2A
      C*****************************************************************
      C       THIS PROGRAM READS TAPE FOR DATA I FOR THE VARIABLE DENSITY CASE
      C*****************************************************************
            SUBROUTINE READ2A(U,V,WH,PINTH,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,
           CHX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,
           CTAI,TAH,TAV,AKT,CB,CW,A,B,C,EUL,T,TW,RO,TE,RREF,TREF,TO,TAMB,
           CTTOT,ITN,EX)
            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)
            GO TO 100
            REWIND 7
      100   CONTINUE
            READ (7)  (((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(((TW(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
           CTAI,TAH,TAV,AKT,CB,CW,A,B,C,EUL,T,TW,RO,TW,TE,RREF,TREF,TO,TAMB,
           CAI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,ITN,EX
            GO TO 200
            REWIND 7
      200   CONTINUE
            RETURN
            END
```

9.2.50 READ2B

This subroutine reads information stored by the subroutine STOR2B for a far field stratified model. It differs from READ2 in the fact that the matrices TW (temperatures at the half grid points) and ROW (densities at the half grid points) are eliminated.

```
<M*DULL(1).READ2B
    1       C*************************************************************************
    2       C       THIS PROGRAM READS TAPE FOR DATA I FOR THE VARIABLE DENSITY CASE
    3       C*************************************************************************
    4             SUBROUTINE READ2B(U,V,WH,PINTH,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,
    5            CHX,HY,HI,MAR,MRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,
    6            CTAI,TAH,TAV,AKT,CB,CW,A,B,C,EUL,T,RO,TE,RREF,TREF,TO,TAMB,
    7       *    CTTOT,ITN,EX)
    8             DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
    9            CWH(IWN,JWN,KN),PINTH(IWN,JWN)
   10             DIMENSION HX(IN,JN),HY(IN,JN),HI(IN,JN),MAR(IN,JN),MRH(IWN,JWN),
   11            CW(IN,JN,KN),WR(IN,JN,KN),WRH(IWN,JWN,KN)
   12             DIMENSION T(IN,JN,KN),RO(IN,JN,KN)
   13             GO TO 100
   14             REWIND 7
   15       100   CONTINUE
   16             READ (7)  (((U(I,J,K),K=1,KN),J=1,JN),I=1,IN),
   17            C(((V(I,J,K),K=1,KN),J=1,JN),I=1,IN),
   18            C(((D(I,J,K),K=1,KN),J=1,JN),I=1,IN),
   19            C(((E(I,J,K),K=1,KN),J=1,JN),I=1,IN),
   20            C(((WH(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
   21            C(((W(I,J,K),K=1,KN),J=1,JN),I=1,IN),
   22            C(((WR(I,J,K),K=1,KN),J=1,JN),I=1,IN),
   23            C(((WRH(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
   24            C((PINTH(IW,JW),JW=1,JWN),IW=1,IWN)
   25            C,((HI(I,J),J=1,JN),I=1,IN),((HX(I,J),J=1,JN),I=1,IN),((HY(I,J),J=
   26            C1,JN),I=1,IN),((MAR(I,J),J=1,JN),I=1,IN),((MRH(IW,JW),JW=1,JWN),
   27            CIW=1,IWN),(((T(I,J,K),K=1,KN),J=1,JN),I=1,IN),
   28            C(((RO(I,J,K),K=1,KN),J=1,JN),I=1,IN),
   29            CTAI,TAH,TAV,AKT,CB,CW,A,B,C,EUL,TE,RREF,TREF,TO,TAMB,
   30            CAI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,ITN,EX
   31             GO TO 200
   32             REWIND 7
   33       200   CONTINUE
   34             RETURN
   35             END
```

9.2.51 READ3

This program classifies into interior, corner or boundary points as shown in Figs. (9.5 and 9.6).

Matrix MAR classifies points on the main grid.

MAR = 0, Point 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, Far corner on y-axis

MAR = 7, Corner at Origin

MAR = 9, Far corner on x-axis

MAR = 10, Corner at the far x-boundary and far y-boundary

Matrix MRH classifies point on the half grid.

MRH = 1, Corner at the 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 the near x and y boundaries

MRH = 6, Far corner on x-axis

MRH = 7, Corner at the far x and y bounds.

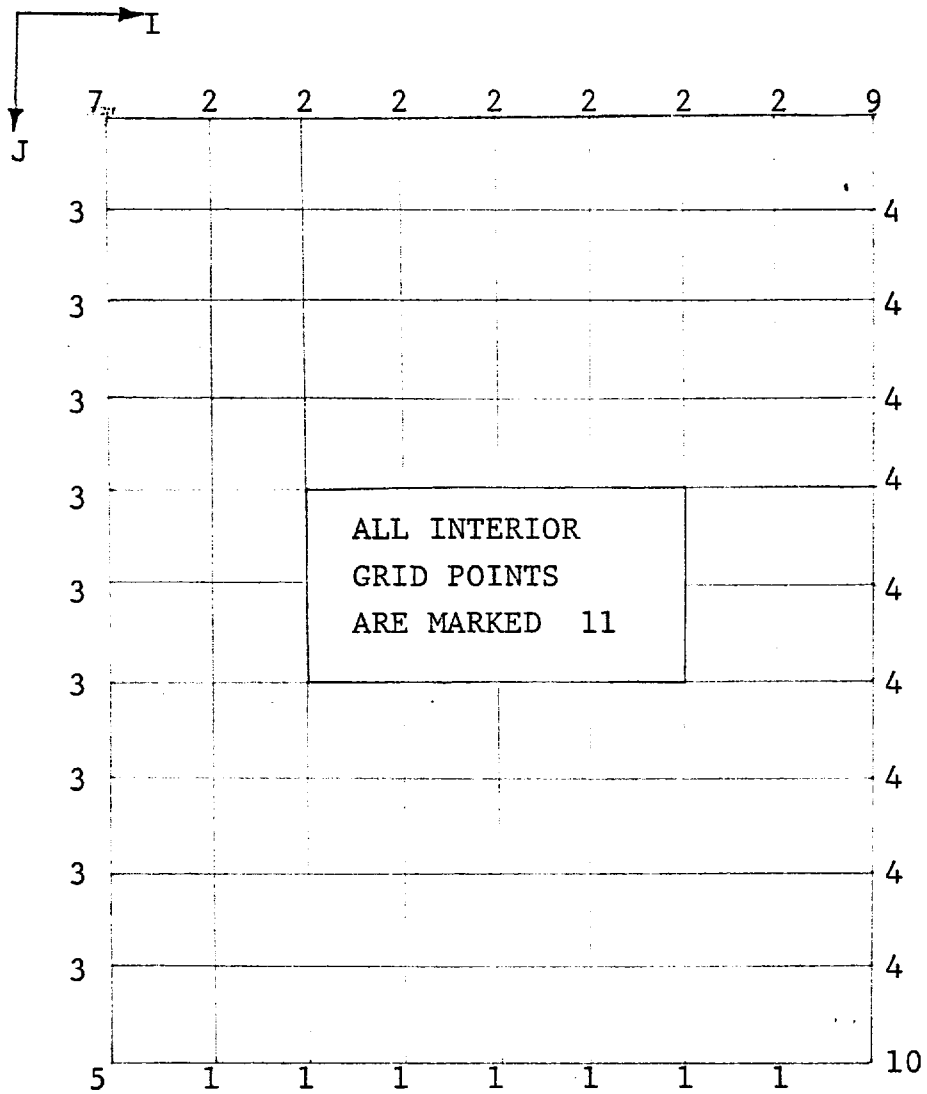



Fig: 9.5 Representation of Identifying numbers in the main grid system for near-field.

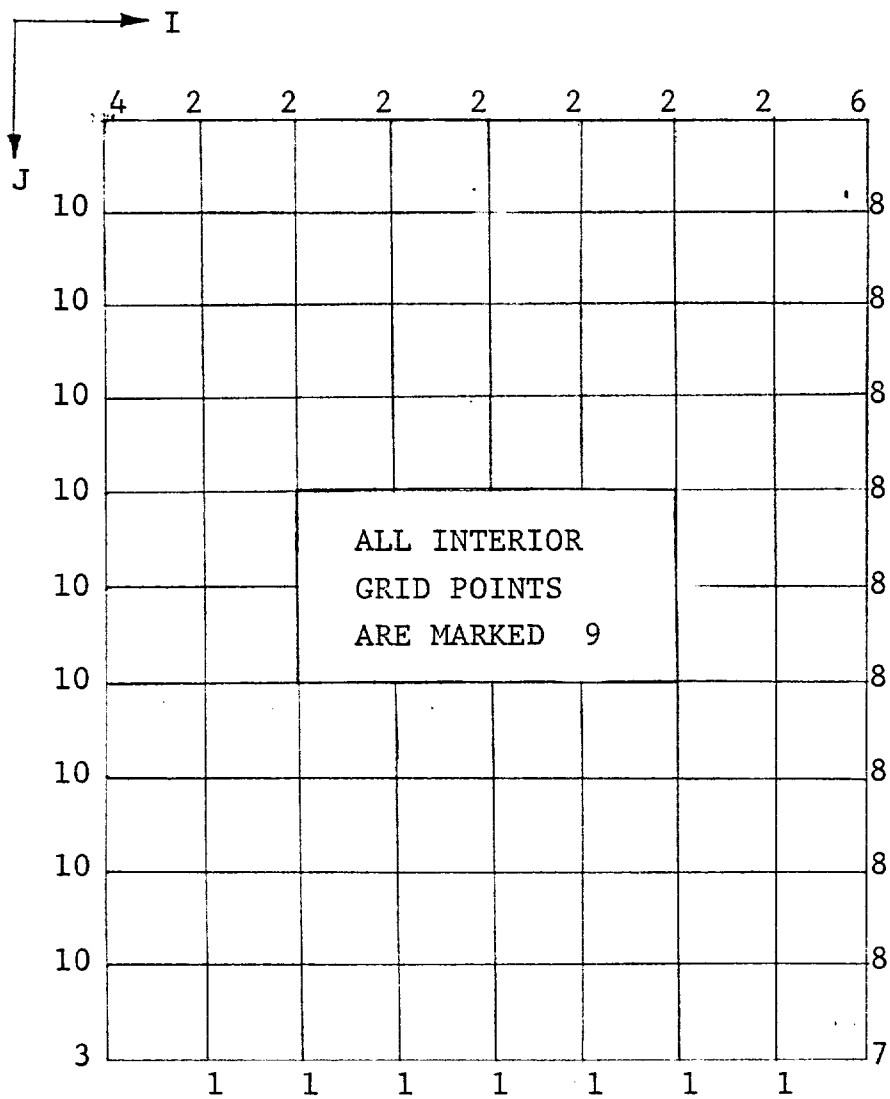



Fig:9.6 Representation of Identifying numbers in the half grid system for near-field.

```
I*DOC.READ3
  1              SUBROUTINE READ3(I,J,IN,JN,IW,JW,IWN,JWN,MAR,MRH)
  2              DIMENSION MAR(IN,JN),MRH(IWN,JWN)
  3              INM1=IN-1
  4              JNM1=JN-1
  5              IWN1=IWN-1
  6              JWN1=JWN-1
  7              DO 100 I=2,INM1
  8              DO 100 J=2,JNM1
  9              MAR(I,J)=11
 10        100  CONTINUE
 11              DO 200 IW=2,IWN1
 12              DO 200 JW=2,JWN1
 13              MRH(IW,JW)=9
 14        200  CONTINUE
 15              DO 300 I=2,INM1
 16              MAR(I,JN)=1
 17              MAR(I,1)=2
 18        300  CONTINUE
 19              DO 400 J=2,JNM1
 20              MAR(1,J)=3
 21              MAR(IN,J)=4
 22        400  CONTINUE
 23              DO 500 IW=2,IWN1
 24              MRH(IW,JWN)=1
 25              MRH(IW,1)=2
 26        500  CONTINUE
 27              DO 600 JW=2,JWN1
 28              MRH(1,JW)=10
 29              MRH(IWN,JW)=8
 30        600  CONTINUE
 31              MAR(1,1)=7
 32              MAR(1,JN)=5
 33              MAR(IN,JN)=10
 34        40
 35              MAR(IN,1)=9
 36              MRH(1,1)=4
 37              MRH(1,JWN)=3
 38              MRH(IWN,JWN)=7
 39              MRH(IWN,1)=6
 40              RETURN
 41              END
```

### 9.2.52 ROINTX

This subroutine computes $x_p$ in the Poisson's equation (Eq. 2.17, Vol.1) $x_p$ is then added to 'XINT'.

```
*DOC.ROINTX
  1              SUBROUTINE ROINTX(I,J,K,IN,JN,KN,DX,DY,DZ,RO,AP,EUL,HI,
  2             CHAR,RINTX,HX,XINT)
  3              DIMENSION RINTX(IN,JN,KN),RO(IN,JN,KN),XINT(IN,JN),HI(IN,JN),
  4             CHAR(IN,JN),HX(IN,JN)
  5              DO 100 I=1,IN
  6              DO 100 J=1,JN
  7              IF (CHAR(I,J).EQ.0) GO TO 101
  8              RINTX(I,J,1)=0.0
  9              DO 110 K=2,KN
 10              IF (CHAR(I,J).EQ.1) GO TO 11
 11              IF (CHAR(I,J).EQ.2) GO TO 12
 12              IF (CHAR(I,J).EQ.3) GO TO 13
 13              IF (CHAR(I,J).EQ.4) GO TO 14
 14              IF (CHAR(I,J).EQ.5) GO TO 15
 15              IF (CHAR(I,J).EQ.6) GO TO 16
 16              IF (CHAR(I,J).EQ.7) GO TO 17
 17              IF (CHAR(I,J).EQ.8) GO TO 18
 18              IF (CHAR(I,J).EQ.9) GO TO 19
 19              IF (CHAR(I,J).EQ.10) GO TO 20
 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*DX)
 21              GO TO 102
 22        11    CONTINUE
 23              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*DX)
 24              GO TO 102
 25        12    CONTINUE
 26              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*DX)
 27              GO TO 102
 28        13    CONTINUE
 29              RX=DZ*(4*(RO(I+1,J,K)+RO(I+1,J,K-1))-3*(RO(I,J,K)+RO(I,J,K-1))
 30             C-(RO(I+2,J,K)+RO(I+2,J,K-1)))/(4*DX)
 31        14    CONTINUE
 32              RX=DZ*(3*(RO(I,J,K)+RO(I,J,K-1))+(RO(I-2,J,K)+RO(I-2,J,K-1))
 33             C-4*(RO(I-1,J,K)+RO(I-1,J,K-1)))/(4*DX)
 34              GO TO 102
 35        15    CONTINUE
 36              RX=DZ*(4*(RO(I+1,J,K)+RO(I+1,J,K-1))-3*(RO(I,J,K)+RO(I,J,K-1))
 37             C-(RO(I+2,J,K)+RO(I+2,J,K-1)))/(4*DX)
 38              GO TO 102
 39        16    CONTINUE
 40              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*DX)
 41              GO TO 102
 42        17    CONTINUE
 43              RX=DZ*(4*(RO(I+1,J,K)+RO(I+1,J,K-1))-3*(RO(I,J,K)+RO(I,J,K-1))
 44             C-(RO(I+2,J,K)+RO(I+2,J,K-1)))/(4*DX)
 45              GO TO 102
 46        18    CONTINUE
 47              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*DX)
 48              GO TO 102
 49        19    CONTINUE
 50              RX=DZ*(3*(RO(I,J,K)+RO(I,J,K-1))+(RO(I-2,J,K)+RO(I-2,J,K-1))
 51             C-4*(RO(I-1,J,K)+RO(I-1,J,K-1)))/(4*DX)
 52              GO TO 102
 53        20    CONTINUE
 54              RX=DZ*(3*(RO(I,J,K)+RO(I,J,K-1))+(RO(I-2,J,K)+RO(I-2,J,K-1))
 55             C-4*(RO(I-1,J,K)+RO(I-1,J,K-1)))/(4*DX)
 56              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))*DZ*
           CHX(I,J)/2.0
      110   CONTINUE
      101   CONTINUE
      100   CONTINUE
            DO 200 I=1,IN
            DO 200 J=1,JN
            IF (MAR(I,J).EQ.0) GO TO 201
            DO 210 K=2,KN
            RINTX(I,J,K)=RINTX(I,J,K)-(K-1)*DZ*HX(I,J)*(RO(I,J,K)+RO(I,J,K-1))
           C/2.0
      210   CONTINUE
      201   CONTINUE
      200   CONTINUE
            DO 300 I=1,IN
            DO 300 J=1,JN
            IF (MAR(I,J).EQ.0) GO TO 301
            DO 310 K=2,KN
            RSUMX=(RINTX(I,J,K)+RINTX(I,J,K-1))*(DZ/2)*AP*EUL*HI(I,J)
            XINT(I,J)=XINT(I,J)+RSUMX
      310   CONTINUE
      301   CONTINUE
      300   CONTINUE
            RETURN
            END
```

### 9.2.53 ROINTY

This subroutine computes $y_p$ in the Poisson's equation. $y_p$ is then added to 'YINT'.

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

```
57              GO TO 102
58        102   CONTINUE
59              RINTY(I,J,K)=RINTY(I,J,K-1)+RY*HI(I,J)+(RO(I,J,K)+RO(I,J,K-1))*DZ*
60             CHY(I,J)/2.0
61        110   CONTINUE
62        101   CONTINUE
63        100   CONTINUE
64              DO 200 I=1,IN
65              DO 200 J=1,JN
66              IF (MAR(I,J).EQ.0) GO TO 201
67              DO 210 K=2,KN
68              RINTY(I,J,K)=RINTY(I,J,K)-(K-1)*DZ*HY(I,J)*(RO(I,J,K)+RO(I,J,K-1))
69             C/2.0
70        210   CONTINUE
71        201   CONTINUE
72        200   CONTINUE
73              DO 300 I=1,IN
74              DO 300 J=1,JN
75              IF (MAR(I,J).EQ.0) GO TO 301
76              DO 310 K=2,KN
77              RSUMY=(RINTY(I,J,K)+RINTY(I,J,K-1))*(DZ/2)*AP*EUL*HI(I,J)
78              YINT(I,J)=YINT(I,J)+RSUMY
79        310   CONTINUE
80        301   CONTINUE
81        300   CONTINUE
82              RETURN
83              END
```

9.2.54 RWH

This subroutine uses continuity equation to compute vertical velocities at half grid points.

```
>DOC.RWH
  1              SUBROUTINE RWH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,DX,DY,DZ,
  2             CMRH)
  3              DIMENSION U(IN,JN,KN),V(IN,JN,KN),WH(IWN,JWN,KN),HI(IN,JN)
  4              DIMENSION MRH(IWN,JWN)
  5              KNM1=KN-1
  6              DO 10 IW=1,IWN
  7              DO 10 JW=1,JWN
  8              IF (MRH(IW,JW).EQ.0) GO TO 8
  9              DO 9 KD=1,KNM1
 10              K=KN-KD+1
 11              I=IW
 12              J=JW
 13              D1HUX=(HI(I+1,J+1)*(U(I+1,J+1,K)+U(I+1,J+1,K-1))+HI(I+1,J)*
 14             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))
 15             C-HI(I,J)*(U(I,J,K)+U(I,J,K-1)))/(4*DX)
 16              D1HVY=(HI(I+1,J+1)*(V(I+1,J+1,K)+V(I+1,J+1,K-1))+HI(I,J+1)*
 17             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))
 18             C-HI(I,J)*(V(I,J,K)+V(I,J,K-1)))/(4*DY)
 19              HH=(HI(I+1,J+1)+HI(I+1,J)+HI(I,J+1)+HI(I,J))/4.0
 20              WH(IW,JW,K-1)=WH(IW,JW,K)+(1.0/HH)*(D1HUX+D1HVY)*DZ
 21        9     CONTINUE
 22        8     CONTINUE
 23        10    CONTINUE
 24              RETURN
 25              END
```

9.2.55 RWR

Computes real vertical velocities from modified vertical velocities used in equations at integral grid points.

```
*DOC.RWR
  1              SUBROUTINE RWR(I,J,K,IN,JN,KN,U,V,W,WR,HI,HX,HY,DZ,MAR)
  2              DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),WR(IN,JN,KN),
  3             CHI(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN)
  4              DO 10 I=1,IN
  5              DO 10 J=1,JN
  6              IF (MAR(I,J).LT.11) GO TO 8
  7              KNM1=KN-1
  8              DO 9 K=1,KNM1
  9              WR(I,J,K)=(K-1)*DZ*(U(I,J,K)*HX(I,J)+V(I,J,K)*HY(I,J))+HI(I,J)
 10             C*W(I,J,K)
 11        9     CONTINUE
 12        8     CONTINUE
 13        10    CONTINUE
 14              RETURN
 15              END
```

### 9.2.56 RWRH

Computes real vertical velocities at half grid points.

```
*DOC.RWRH
  1              SUBROUTINE RWRH(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,WH,HI,HX,HY,
  2             CDX,DY,DZ,MRH,WRH)
  3              DIMENSION U(IN,JN,KN),V(IN,JN,KN),WH(IWN,JWN,KN),HI(IN,JN),
  4             CHX(IN,JN),HY(IN,JN),MRH(IWN,JWN)
  5              DIMENSION WRH(IWN,JWN,KN)
  6              KNM1=KN-1
  7              DO 10 IW=1,IWN
  8              DO 10 JW=1,JWN
  9              IF (MRH(IW,JW).EQ.0) GO TO 8
 10              HXAV=(HX(I+1,J)+HX(I+1,J+1)+HX(I,J)+HX(I,J+1))/4.
 11              HYAV=(HY(I+1,J)+HY(I+1,J+1)+HY(I,J)+HY(I,J+1))/4.
 12              HIAV=(HI(I+1,J)+HI(I+1,J+1)+HI(I,J)+HI(I,J+1))/4.
 13              DO 9 K=1,KNM1
 14              I=IW
 15              J=JW
 16              UAV=(U(I+1,J,K)+U(I+1,J+1,K)+U(I,J,K)+U(I,J+1,K))/4.
 17              VAV=(V(I+1,J,K)+V(I+1,J+1,K)+V(I,J,K)+V(I,J+1,K))/4.
 18              WRH(IW,JW,K)=(K-1)*DZ*(UAV*HXAV+VAV*HYAV)+HIAV*WH(IW,JW,K)
 19        9     CONTINUE
 20        8     CONTINUE
 21        10    CONTINUE
 22              RETURN
 23              END
```

9.2.57 STORE2

This subroutine stores values of input parameters and physical quantities on a file designated as Unit 8.

```
*DOC .STORE2
  1       C*************************************************************************
  2       C        THIS PROGRAM STORES THE RELEVANT DATA INTO FILE 8
  3       C*************************************************************************
  4             SUBROUTINE STORE2(U,V,WH,PINTH,I,J,K,IW,JW,IN,JN,KN,IWN,JWN,D,E,
  5            CHX,HY,HI,HAR,HRH,AI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,W,WR,WRH,
  6            CTAI,TAH,TAV,AKT,CB,CW,A,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,TAMB,
  7            CTTOT)
  8             DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
  9            CWH(IWN,JWN,KN),PINTH(IWN,JWN)
 10             DIMENSION HX(IN,JN),HY(IN,JN),HI(IN,JN),HAR(IN,JN),HRH(IWN,JWN),
 11            CW(IN,JN,KN),WR(IN,JN,KN),WRH(IWN,JWN,KN)
 12             DIMENSION T(IN,JN,KN),RO(IN,JN,KN),TW(IWN,JWN,KN),ROW(IWN,JWN,KN)
 13             REWIND 8
 14             WRITE (8) (((U(I,J,K),K=1,KN),J=1,JN),I=1,IN),
 15            C(((V(I,J,K),K=1,KN),J=1,JN),I=1,IN),
 16            C(((D(I,J,K),K=1,KN),J=1,JN),I=1,IN),
 17            C(((E(I,J,K),K=1,KN),J=1,JN),I=1,IN),
 18            C(((WH(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
 19            C(((W(I,J,K),K=1,KN),J=1,JN),I=1,IN),
 20            C(((WR(I,J,K),K=1,KN),J=1,JN),I=1,IN),
 21            C(((WRH(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
 22            C((PINTH(IW,JW),JW=1,JWN),IW=1,IWN)
 23            C,((HI(I,J),J=1,JN),I=1,IN),((HX(I,J),J=1,JN),I=1,IN),((HY(I,J),J=
 24            C1,JN),I=1,IN),((HAR(I,J),J=1,JN),I=1,IN),((HRH(IW,JW),JW=1,JWN),
 25            CIW=1,IWN),(((T(I,J,K),K=1,KN),J=1,JN),I=1,IN),
 26            C(((RO(I,J,K),K=1,KN),J=1,JN),I=1,IN),
 27            C(((TW(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
 28            C(((ROW(IW,JW,K),K=1,KN),JW=1,JWN),IW=1,IWN),
 29            CTAI,TAH,TAV,AKT,CB,CW,A,B,C,EUL,T,TW,RO,ROW,TE,RREF,TREF,TO,TAMB,
 30            CAI,AH,AV,AP,DX,DY,DZ,DT,TAUX,TAUY,TTOT
 31             REWIND 8
 32             RETURN
 33             END
```

9.2.58 TEMB2

This program makes computations for temperatures at the boundary points in the domain of interest. Centraldifference schemes are used for computating derivative of temperature with respect to $\alpha$ and $\beta$. Heat flux specified at the vertical walls is used to compute the derivatives normal to the vertical walls. Heat flux boundary condition is used in a manner explained in TEMI4.

```
N*DOC.TEMB2
   1              SUBROUTINE TEMB2(I,J,K,IN,JN,KN,TD,DX,DY,DZ,MAR,CB,HI,AKT,CW,
   2             CTAMB,HX,HY,T,TREF,TAV,TAI,TAH,B3,DT)
   3              DIMENSION T(IN,JN,KN),TD(IN,JN,KN),MAR(IN,JN),HX(IN,JN),HY(IN,JN),
   4             CHI(IN,JN)
   5              KNM1=KN-1
   6              DO 100 K=1,KN
   7              DO 100 I=1,IN
   8              DO 100 J=1,JN
   9              D1HTUX=0.0
  10              D1HTVY=0.0
  11              D1TWZ=0.0
  12              IF (MAR(I,J).EQ.0) GO TO 300
  13              IF (MAR(I,J).EQ.11) GO TO 300
  14              IF (MAR(I,J).EQ.1) GO TO 11
  15              IF (MAR(I,J).EQ.2) GO TO 12
  16              IF (MAR(I,J).EQ.3) GO TO 13
  17              IF (MAR(I,J).EQ.4) GO TO 14
  18              IF (MAR(I,J).EQ.5) GO TO 15
  19              IF (MAR(I,J).EQ.6) GO TO 16
  20              IF (MAR(I,J).EQ.7) GO TO 17
  21              IF (MAR(I,J).EQ.8) GO TO 18
  22              IF (MAR(I,J).EQ.9) GO TO 19
  23              IF (MAR(I,J).EQ.10) GO TO 20
  24        11    CONTINUE
  25              D1TX=(T(I+1,J,K)-T(I-1,J,K))/(2*DX)
  26              D2TX=(T(I+1,J,K)+T(I-1,J,K)-2*T(I,J,K))/(DX*DX)
  27              D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
  28              D1TY=0.0
  29              D2TY=2*(T(I,J-1,K)-T(I,J,K))/(DY*DY)
  30              IF (K.EQ.1) GO TO 110
  31              IF (K.EQ.KN) GO TO 120
  32              GO TO 200
  33        12    CONTINUE
  34              D1TX=(T(I+1,J,K)-T(I-1,J,K))/(2*DX)
  35              D2TX=(T(I+1,J,K)+T(I-1,J,K)-2*T(I,J,K))/(DX*DX)
  36              D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
  37              D1TY=0.0
  38              D2TY=2*(T(I,J+1,K)-T(I,J,K))/(DY*DY)
  39              IF (K.EQ.1) GO TO 110
  40              IF (K.EQ.KN) GO TO 120
  41              GO TO 200
  42        13    CONTINUE
  43              D1TX=0.0
  44              D2TX=2*(T(I+1,J,K)-T(I,J,K))/(DX*DX)
  45              D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
  46              D1TY=(T(I,J+1,K)-T(I,J-1,K))/(2*DY)
  47              D2TY=(T(I,J+1,K)+T(I,J-1,K)-2*T(I,J,K))/(DY*DY)
  48              IF(K.EQ.1) GO TO 110
  49              IF(K.EQ.KN) GO TO 120
  50              GO TO 200
  51        14    CONTINUE
  52              D1TX=0.0
  53              D2TX=2*(T(I-1,J,K)-T(I,J,K))/(DX*DX)
  54              D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
  55              D1TY=(T(I,J+1,K)-T(I,J-1,K))/(2*DY)
  56              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))/(DY*DY)
      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))/(DY*DY)
      IF (K.EQ.1) GO TO 110
      IF (K.EQ.KN) GO TO 120
      GO TO 200
110   CONTINUE
      CT=AKT*((T(I,J,1)*TREF+TREF)-TAMB)/TREF
      CT=CT*HI(I,J)
      D2TZ=2*(T(I,J,2)-CT*DZ-T(I,J,1))/(DZ*DZ)
      GO TO 200
120   CONTINUE
      D2TZ=2*(T(I,J,K-1)-T(I,J,K))/(DZ*DZ)
      GO TO 200
200   CONTINUE
      TD(I,J,K)=(1.0/HI(I,J))*(-TAI*(D1HTUX+D1HTVY+HI(I,J)*D1TWZ)+TAH
     C*(D1TX*HX(I,J)+D2TX*HI(I,J)+D1TY*HY(I,J)+D2TY*HI(I,J))+(TAV/HI(I,J
     C))*B3*D2TZ)*DT+T(I,J,K)
16    CONTINUE
18    CONTINUE
300   CONTINUE
100   CONTINUE
      RETURN
      END
```

9.2.59 TEMB2A

This subroutine is used by the far field stratified model. This subroutine is similar to TEMB2 with the difference that this subroutine calls the subroutine VERTDF which supplies the value and the derivative of vertical diffusivity for every grid point.

```
SKM*DULL(1).TEMB2A
      1              SUBROUTINE TEMB2(I,J,K,IN,JN,KN,TD,DX,DY,DZ,MAR,CE,HI,AKT,CW,
      2             CTAMB,HX,HY,T,TREF,TAV,TAI,TAH,BS,DT)
      3              DIMENSION T(IN,JN,KN),TD(IN,JN,KN),MAR(IN,JN),HX(IN,JN),HY(IN,JN),
      4             CHI(IN,JN)
      5              KNM1=KN-1
      6              DO 100 K=1,KN
      7              DO 100 I=1,IN
      8              DO 100 J=1,JN
      9              D1HTUX=0.0
     10              D1HTVY=0.0
     11              D1TWZ=0.0
     12              IF (MAR(I,J).EQ.0) GO TO 300
     13              IF (MAR(I,J).EQ.11) GO TO 300
     14              IF (MAR(I,J).EQ.1) GO TO 11
     15              IF (MAR(I,J).EQ.2) GO TO 12
     16              IF (MAR(I,J).EQ.3) GO TO 13
     17              IF (MAR(I,J).EQ.4) GO TO 14
     18              IF (MAR(I,J).EQ.5) GO TO 15
     19              IF (MAR(I,J).EQ.6) GO TO 16
     20              IF (MAR(I,J).EQ.7) GO TO 17
     21              IF (MAR(I,J).EQ.8) GO TO 18
     22              IF (MAR(I,J).EQ.9) GO TO 19
     23              IF (MAR(I,J).EQ.10) GO TO 20
     24        11    CONTINUE
     25              D1TX=(T(I+1,J,K)-T(I-1,J,K))/(2*DX)
     26              D2TX=(T(I+1,J,K)+T(I-1,J,K)-2*T(I,J,K))/(DX*DX)
     27              D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
     28              D1TY=0.0
     29              D2TY=2*(T(I,J-1,K)-T(I,J,K))/(DZ*DZ)
     30              IF (K.EQ.1) GO TO 110
     31              IF (K.EQ.KN) GO TO 120
     32              GO TO 200
     33        12    CONTINUE
     34              D1TX=(T(I+1,J,K)-T(I-1,J,K))/(2*DX)
     35              D2TX=(T(I+1,J,K)+T(I-1,J,K)-2*T(I,J,K))/(DX*DX)
     36              D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
     37              D1TY=0.0
     38              D2TY=2*(T(I,J+1,K)-T(I,J,K))/(DY*DY)
     39              IF (K.EQ.1) GO TO 110
     40              IF (K.EQ.KN) GO TO 120
     41              GO TO 200
     42        13    CONTINUE
     43              D1TX=0.0
     44              D2TX=2*(T(I+1,J,K)-T(I,J,K))/(DX*DX)
     45              D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
     46              D1TY=(T(I,J+1,K)-T(I,J-1,K))/(2*DY)
     47              D2TY=(T(I,J+1,K)+T(I,J-1,K)-2*T(I,J,K))/(DY*DY)
     48              IF (K.EQ.1) GO TO 110
     49              IF (K.EQ.KN) GO TO 120
     50              GO TO 200
     51        14    CONTINUE
     52              D1TX=0.0
     53              D2TX=2*(T(I-1,J,K)-T(I,J,K))/(DX*DX)
     54              D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
     55              D1TY=(T(I,J+1,K)-T(I,J-1,K))/(2*DY)
     56              D2TY=(T(I,J+1,K)+T(I,J-1,K)-2*T(I,J,K))/(DY*DY)
```

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

9.2.60 TEMI4

This program computes temperatures at the interior points of the domain of interest. Computations are made at the full grid points. Central-difference schemes are used to write the first and second derivatives with respect to $\alpha$ and $\beta$. Single sided difference schemes are used for derivatives with respect to $\gamma$ at top surface and at bottom. Corner points designated by MAR= 6 and 8 are treated as interior points. Heat flux through the surface is used in specifying the first and second derivaties of temperature with respect to $\gamma$. Heat flux through the bottom is used in specifying the first and second derivatives of temperature with respect to $\gamma$. Energy equation is used to compute temperatures.

```
I*DOC.TEMI4
  1              SUBROUTINE TEMI4(I,J,K,IN,JN,KN,U,V,T,TD,DX,
  2             CCB,
  3             COY,DZ,W,DT,TAI,TAH,TAV,B3,HI,HX,HY,MAR,AKT,TREF,TAMB)
  4              DIMENSION U(IN,JN,KN),V(IN,JN,KN),HI(IN,JN),HX(IN,JN),HY(IN,JN),
  5             CMAR(IN,JN),T(IN,JN,KN),TD(IN,JN,KN),W(IN,JN,KN)
  6              KNM1=KN-1
  7              DO 10 I=1,IN
  8              DO 10 J=1,JN
  9              IF(MAR(I,J).EQ.6) GO TO 100
 10              IF(MAR(I,J).EQ.8) GO TO 100
 11              IF (MAR(I,J).LT.11) GO TO 9
 12        100   CONTINUE
 13              DO 8 K=1,KN
 14              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)
 15             C*HI(I-1,J))/(2*DX)
 16              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)*
 17             CHI(I,J-1))/(2*DY)
 18              D1TX=(T(I+1,J,K)-T(I-1,J,K))/(2*DX)
 19              D1TY=(T(I,J+1,K)-T(I,J-1,K))/(2*DY)
 20              D1TWZ=(T(I,J,K+1)*W(I,J,K+1)-T(I,J,K-1)*W(I,J,K-1))/(2*DZ)
 21              D2TX=(T(I+1,J,K)+T(I-1,J,K)-2*T(I,J,K))/(DX*DX)
 22              D2TY=(T(I,J+1,K)+T(I,J-1,K)-2*T(I,J,K))/(DY*DY)
 23              D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2*T(I,J,K))/(DZ*DZ)
 24              IF (K.EQ.1) GO TO 24
 25              IF (K.EQ.KN) GO TO 20
 26              GO TO 21
 27        20    CONTINUE
 28              D1TWZ=0.0
 29              D2TZ=2*(T(I,J,K-1)-T(I,J,K)+CB*HI(I,J)*DZ)/(DZ*DZ)
 30              GO TO 21
 31        24    CONTINUE
 32              CT=AKT*((T(I,J,1)*TREF+TREF)-TAMB)/TREF
 33              CT=CT*HI(I,J)
 34              D1TWZ=0.0
 35              D2TZ=2*(T(I,J,2)-CT*DZ-T(I,J,1))/(DZ*DZ)
 36        21    CONTINUE
 37              TD(I,J,K)=(1.0/HI(I,J))*(-TAI*(D1HTUX+D1HTVY+HI(I,J)*D1TWZ)
 38             C+TAH*(D1TX*HX(I,J)+D2TX*HI(I,J)+D1TY*HY(I,J)+D2TY*HI(I,J))
 39             C+(TAV/HI(I,J))*B3*D2TZ)*DT+T(I,J,K)
 40        8     CONTINUE
 41        9     CONTINUE
 42        10    CONTINUE
 43              RETURN
 44              END
```

9.2.61 TEMI4B

This subroutine is used by the far field stratified model. The subroutine is similar to TEMI4 with the difference that it calls for subroutine VERTDF which supplies the value of vertical diffusivity and its derivative.

```
*DULL(1).TEMI48
      SUBROUTINE TEMI48(I,J,K,IN,JN,KN,U,V,T,TD,DX,
     CCB,
     CDY,DZ,W,DT,TAI,TAH,TAV,B3,HI,HX,HY,MAR,AKT,TREF,TAMB,A3
     C,CONS,AVMX,AVMN)
      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),A3(KN)
C     KNM1=KN-1
      DO 10 I=1,IN
      DO 10 J=1,JN
      IF(MAR(I,J).EQ.6) GO TO 100
      IF(MAR(I,J).EQ.8) GO TO 100
      IF (MAR(I,J).LT.11) GO TO 9
  100 CONTINUE
      DO 8 K=1,KN
      CALL VERTDF(I,J,K,IN,JN,KN,HI,AB3,D1A3Z,D1B3Z,DZ,T,A3,TREF
     C,CONS,AVMX,AVMN)
      B3=AB3
      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)
      D1TZ=(T(I,J,K+1)-T(I,J,K-1))/(2.*DZ)
      IF(MAR(I,J).EQ.11) GO TO 200
      D1TWZ=0.0
  200 CONTINUE
      IF (K.EQ.1) GO TO 24
      IF (K.EQ.KN) GO TO 20
      GO TO 21
   20 CONTINUE
      GO TO 30
      D1TWZ=0.0
      D2TZ=2*(T(I,J,K-1)-T(I,J,K)+CB*HI(I,J)*DZ)/(DZ*DZ)
      D1TZ=0.0
      GO TO 21
   24 CONTINUE
      CT=AKT*((T(I,J,1)*TREF+TREF)-TAMB)/TREF
      CT=CT*HI(I,J)
      D1TZ=CT
      D1TWZ=(4.*T(I,J,K+1)*W(I,J,K+1)-3.*T(I,J,K)*W(I,J,K)-T(I,J,K+2)
     C*W(I,J,K+2))/(2.*DZ)
      D2TZ=2*(T(I,J,2)-CT*DZ-T(I,J,1))/(DZ*DZ)
   21 CONTINUE
      TD(I,J,K)=(1.0/HI(I,J))*(-TAI*(D1HTUX+D1HTVY+HI(I,J)*D1TWZ)
     C+TAH*(D1TX*HX(I,J)+D2TX*HI(I,J)+D1TY*HY(I,J)+D2TY*HI(I,J))
     C+(TAV/HI(I,J))*(B3*D2TZ+D1TZ*D1B3Z))*DT+T(I,J,K)
      GO TO 8
   30 CONTINUE
      TD(I,J,KN)=T(I,J,KN)
    8 CONTINUE
    9 CONTINUE
```

```
57        10    CONTINUE
58              RETURN
59              END
```

9.2.62 TEQB

This program allows for vertical mixing at a particular grid point, if the temperature of the fluid at the grid point just above it is less and the difference of temperatures is more than a specified maximum, then the temperatures at the two points are averaged.

```
*DOC.TEQB
  1             SUBROUTINE TEQB(I,J,K,IN,JN,KN,T,MAR)
  2             DIMENSION T(IN,JN,KN),MAR(IN,JN)
  3             KNM1=KN-1
  4             DO 10 I=1,IN
  5             DO 10 J=1,JN
  6             IF (MAR(I,J).EQ.0) GO TO 9
  7       110   CONTINUE
  8             MARK=0
  9             DO 8 K=1,KNM1
 10             TX=0.002
 11             TT=T(I,J,K)+TX
 12             IF (K.EQ.1) GO TO 7
 13             IF (K.EQ.KNM1) GO TO 6
 14             IF (TT.GE.T(I,J,K+1)) GO TO 111
 15             MARK=1
 16             AVT=(TT+T(I,J,K+1))/2.0
 17             T(I,J,K)=AVT-TX
 18             T(I,J,K+1)=AVT
 19             GO TO 5
 20       7     CONTINUE
 21             IF (TT.GE.T(I,J,K+1)) GO TO 111
 22             MARK=1
 23             AVT=(TT+2*T(I,J,K+1))/3.0
 24             T(I,J,K)=AVT-TX
 25             T(I,J,K+1)=AVT
 26             GO TO 5
 27       6     CONTINUE
 28             IF (TT.GE.T(I,J,K+1)) GO TO 111
 29             MARK=1
 30             AVT=(2*TT+T(I,J,K+1))/3.0
 31             T(I,J,K)=AVT-TX
 32             T(I,J,K+1)=AVT
 33             GO TO 5
 34       5     CONTINUE
 35       111   CONTINUE
 36       8     CONTINUE
 37       9     CONTINUE
 38       10    CONTINUE
 39             RETURN
 40             END
```

### 9.2.63 TPRIN1

This program prints the input parameters.

```
*DOC.TPRIN1
  1              SUBROUTINE TPRIN1(TAI,TAH,TAV,CB,CW,AKT,TREF,RREF,EUL,A,B,C,TE,TO)
  2              PRINT 10, TAI,TAH,TAV,CB,CW,AKT,TREF,RREF,EUL,A,B,C,TE,TO
  3        10    FORMAT (/' TAI=',E15.7,/' TAH=',E15.7,/' TAV=',E15.7,/' CB=',E15.7
  4             C,/' CW=',E15.7,/' AKT=',E15.7,/' TREF=',E15.7,/' RREF=',E15.7,/
  5             C' EUL=',E15.7,/' A=',E15.7,/' B=',E15.7,/' C=',E15.7,/' TE=',E15.7
  6             C,/' TO=',E15.7,/)
  7              RETURN
  8              END
```

### 9.2.64 TPRIN2

This program prints temperature and density fields.

```
*DOC.TPRIN2
  1       C**********************************************************************
  2       C        THIS PROGRAM PRINTS THE VALUES OF T,TW,RO,ROW
  3       C**********************************************************************
  4             SUBROUTINE TPRIN2(I,J,K,IN,JN,KN,T,RO,TREF)
  5             DIMENSION T(IN,JN,KN),RO(IN,JN,KN)
  6             DO 10 K=1,KN
  7             DO 10 I=1,IN
  8             PRINT 11,K,I,(T(I,J,K),J=1,JN)
  9       10    PRINT 12,(RO(I,J,K),J=1,JN)
 10       11    FORMAT(/' K=',I3,3X,'I=',I3/' TEMPERATURE'/(5X,8E15.7))
 11       12    FORMAT(' DENSITY'/(5X,8E15.7))
 12             DO 100 K=1,KN
 13             DO 100 J=1,JN
 14             DO 100 I=1,IN
 15       100   T(I,J,K)=(1.+T(I,J,K))*TREF
 16             DO 150 K=1,KN
 17             WRITE(6,105) K
 18             DO 140 I=1,IN
 19             WRITE(6,106) (T(I,J,K),J=1,JN)
 20       140   CONTINUE
 21       150   CONTINUE
 22       105   FORMAT('1',' TEMPERATURES FOR K=',I5)
 23       106   FORMAT(//,22F5.1)
 24             RETURN
 25             END
```

9.2.65 TPRIN9

This subroutine is used by the far field models. Itprints temperature and density fields.

```
M*DULL(1).TPRIN9
C*****************************************************************************
C        THIS PROGRAM PRINTS THE VALUES OF T,TW,RO,ROW
C*****************************************************************************
      SUBROUTINE TPRIN9(I,J,K,IN,JN,KN,T,RO,TREF,MAR)
      DIMENSION T(IN,JN,KN),RO(IN,JN,KN),MAR(IN,JN)
      GO TO 512
      DO 10 K=1,KN
      DO 10 I=1,IN
      PRINT 11,K,I,(T(I,J,K),J=1,JN)
10    PRINT 12,(RO(I,J,K),J=1,JN)
11    FORMAT(/' K=',I3,3X,'I=',I3/' TEMPERATURE'/(5X,8E15.7))
12    FORMAT(' DENSITY'/(5X,8E15.7))
512   CONTINUE
      DO 100 K=1,KN
      DO 100 J=1,JN
      DO 100 I=1,IN
      T(I,J,K)=(1.+T(I,J,K))*TREF
      IF(MAR(I,J).EQ.0) T(I,J,K)=1000000.00
100   CONTINUE
      DO 150 K=1,KN
      WRITE(6,105) K
      DO 140 I=1,IN
      WRITE(6,106) (T(I,J,K),J=1,JN)
140   CONTINUE
150   CONTINUE
105   FORMAT('1',,' TEMPERATURES FOR K=',I5,/)
106   FORMAT(//,18F7.1)
      RETURN
      END
```

9.2.66 TPRINA

This subroutine is used by the far field printing programs. The subroutine is similar to TPRIN2 with the difference that it is tailored to the Lake Belews application.

```
M*DULL(1).TPRINA
C*****************************************************************************
C        THIS PROGRAM PRINTS THE VALUES OF T,TW,RO,ROW
C*****************************************************************************
      SUBROUTINE TPRINA(I,J,K,IN,JN,KN,T,RO,TREF,MAR)
      DIMENSION T(IN,JN,KN),RO(IN,JN,KN),MAR(IN,JN)
      DO 10 K=1,KN
      DO 10 I=1,IN
      PRINT 11,K,I,(T(I,J,K),J=1,JN)
10    PRINT 12,(RO(I,J,K),J=1,JN)
11    FORMAT(/' K=',I3,3X,'I=',I3/' TEMPERATURE'/(5X,8E15.7))
12    FORMAT(' DENSITY'/(5X,8E15.7))
      DO 100 K=1,KN
      DO 100 J=1,JN
      DO 100 I=1,IN
      T(I,J,K)=(1.+T(I,J,K))*TREF
      IF(MAR(I,J).EQ.0) T(I,J,K)=1000000.00
100   CONTINUE
      DO 150 K=1,KN
      WRITE(6,105) K
      DO 140 I=1,IN
      WRITE(6,106) (T(I,J,K),J=1,JN)
140   CONTINUE
150   CONTINUE
105   FORMAT('1',,' TEMPERATURES FOR K=',I5,/)
106   FORMAT(//,18F7.1)
      RETURN
      END
```

92.67 UANVC

This subroutine adds coriolis component to H and G which are obtained from subroutine "UV"

```
I*DOC.UANVC
  1              SUBROUTINE UANVC(I,J,K,IN,JN,KN,ABR,DT,U,V,H,G,HI,MAR)
  2              DIMENSION H(IN,JN,KN),G(IN,JN,KN),U(IN,JN,KN),V(IN,JN,KN),
  3             CHI(IN,JN),MAR(IN,JN)
  4              KNM1=KN-1
  5              DO 10 I=1,IN
  6              DO 10 J=1,JN
  7              IF (MAR(I,J).LT.11) GO TO 9
  8              DO 8 K=2,KNM1
  9              H(I,J,K)=H(I,J,K)+ABR*HI(I,J)*V(I,J,K)*DT
 10              G(I,J,K)=G(I,J,K)-ABR*HI(I,J)*U(I,J,K)*DT
 11        8     CONTINUE
 12        9     CONTINUE
 13        10    CONTINUE
 14              RETURN
 15              END
```

9.2.68 UV

This subroutine computes new horizontal velocities without including the Coriolis component which is added later in the subroutine UANVC. This program uses U and V momentum equations and computes velocities in the interior points after one time step and are stored as H and G.

```
*DOC.UV
  1              SUBROUTINE UV(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,D,E,H,G,DX,DY,DZ,
  2             CW,
  3             CDT,AI,AP,AH,AV,A3,HI,HX,HY,P,MAR)
  4              DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
  5             CH(IN,JN,KN),G(IN,JN,KN),HI(IN,JN),HX(IN,JN),HY(IN,JN),P(IWN,JWN),
  6             CMAR(IN,JN)
  7              DIMENSION W(IN,JN,KN)
  8              DIMENSION A3(KN)
  9              KNM1=KN-1
 10              A=DT*AH*(1/(DX*DX)+1/(DY*DY))
 11              DO 10 I=1,IN
 12              DO 10 J=1,JN
 13              IW=I
 14              JW=J
 15              IF (MAR(I,J).LT.11) GO TO 9
 16              D1PX=(P(IW,JW)-P(IW-1,JW)+P(IW,JW-1)-P(IW-1,JW-1))/(2*DX)
 17              D1PY=(P(IW,JW)-P(IW,JW-1)+P(IW-1,JW)-P(IW-1,JW-1))/(2*DY)
 18              DO 8 K=2,KNM1
 19              B=DT*AV*A3(K)/(DZ*DZ)
 20              C=(HI(I,J)+A*HI(I,J)+B/HI(I,J))
 21              D1HUUX=(U(I+1,J,K)*U(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
 22             C*U(I-1,J,K)*HI(I-1,J))/(2*DX)
 23              D1HUVY=(U(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-U(I,J-1,K)
 24             C*V(I,J-1,K)*HI(I,J-1))/(2*DY)
 25              D1HUVX=(U(I+1,J,K)*V(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
 26             C*V(I-1,J,K)*HI(I-1,J))/(2*DX)
 27              D1HVVY=(V(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-V(I,J-1,K)*
 28             CV(I,J-1,K)*HI(I,J-1))/(2*DY)
 29              D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
 30              D1UY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
 31              D1VX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
 32              D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
 33              D1UWZ=(U(I,J,K+1)*W(I,J,K+1)-U(I,J,K-1)*W(I,J,K-1))/(2*DZ)
 34              D1VWZ=(V(I,J,K+1)*W(I,J,K+1)-V(I,J,K-1)*W(I,J,K-1))/(2*DZ)
 35              DDUX=(U(I+1,J,K)+U(I-1,J,K)-D(I,J,K))/(DX*DX)
 36              DDUY=(U(I,J+1,K)+U(I,J-1,K)-D(I,J,K))/(DY*DY)
 37              DDUZ=(U(I,J,K+1)+U(I,J,K-1)-D(I,J,K))/(DZ*DZ)
 38              DDVX=(V(I+1,J,K)+V(I-1,J,K)-E(I,J,K))/(DX*DX)
 39              DDVY=(V(I,J+1,K)+V(I,J-1,K)-E(I,J,K))/(DY*DY)
 40              DDVZ=(V(I,J,K+1)+V(I,J,K-1)-E(I,J,K))/(DZ*DZ)
 41              H(I,J,K)=(DT/C)*(-AI*(D1HUUX+D1HUVY+HI(I,J)*D1UWZ)-HI(I,J)*AP
 42             C*D1PX+AH*HI(I,J)*(DDUX+DDUY)+AH*HX(I,J)*D1UX+AH*HY(I,J)*D1UY
 43             C+AV*A3(K)*DDUZ/HI(I,J))+HI(I,J)*U(I,J,K)/C
 44              G(I,J,K)=(DT/C)*(-AI*(D1HUVX+D1HVVY+HI(I,J)*D1VWZ)-HI(I,J)*AP
 45             C*D1PY+AH*HI(I,J)*(DDVX+DDVY)+AH*HX(I,J)*D1VX+AH*HY(I,J)*D1VY
 46             C+AV*A3(K)*DDVZ/HI(I,J))+HI(I,J)*V(I,J,K)/C
 47       8      CONTINUE
 48       9      CONTINUE
 49       10     CONTINUE
 50              RETURN
 51              END
```

9.2.69 UVT

This program is called in by TMAIN. This program computes U and V for variable density model, at successive time steps. $\alpha$ and $\beta$ momentum equations are used for computations. Results are stored as H (for U) and G (for V).

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

### 9.2.70 UVTB

This subroutine is used by the far field stratified model. The subroutine is similar to UVT with the difference that it calls for subroutine VERTDF which supplies the value for vertical viscosity and its derivative.

```
M*DULL(1).UVTB
    1              SUBROUTINE UVTB(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,U,V,D,E,H,G,DX,DY,DZ,
    2             CRINTX,RINTY,EUL,
    3             CW,
    4             CDT,AI,AP,AH,AV,A3,HI,HX,HY,P,MAR,T,TREF,CONS,AVMX,AVMN)
    5              DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
    6             CH(IN,JN,KN),G(IN,JN,KN),HI(IN,JN),HX(IN,JN),HY(IN,JN),P(IWN,JWN),
    7             CMAR(IN,JN)
    8              DIMENSION W(IN,JN,KN),T(IN,JN,KN)
    9              DIMENSION A3(KN)
   10              DIMENSION RINTX(IN,JN,KN),RINTY(IN,JN,KN)
   11              KNM1=KN-1
   12              A=DT*AH*(1/(DX*DX)+1/(DY*DY))
   13              DO 10 I=1,IN
   14              DO 10 J=1,JN
   15              IW=I
   16              JW=J
   17              IF (MAR(I,J).LT.11) GO TO 9
   18              DO 8 K=2,KNM1
   19              D1PX=(P(IW,JW)-P(IW-1,JW)+P(IW,JW-1)-P(IW-1,JW-1))/(2*DX)
   20             C+EUL*RINTX(I,J,K)
   21              D1PY=(P(IW,JW)-P(IW,JW-1)+P(IW-1,JW)-P(IW-1,JW-1))/(2*DY)
   22             C+EUL*RINTY(I,J,K)
   23              CALL VERTDF(I,J,K,IN,JN,KN,HI,AB3,D1A3Z,D1B3Z,DZ,T,A3,TREF
   24             C,CONS,AVMX,AVMN)
   25              A3(K)=AB3
   26              B=DT*AV*A3(K)/(DZ*DZ)
   27              C=(HI(I,J)+A*HI(I,J)+B/HI(I,J))
   28              D1HUUX=(U(I+1,J,K)*U(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
   29             C*U(I-1,J,K)*HI(I-1,J))/(2*DX)
   30              D1HUVY=(U(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-U(I,J-1,K)
   31             C*V(I,J-1,K)*HI(I,J-1))/(2*DY)
   32              D1HUVX=(U(I+1,J,K)*V(I+1,J,K)*HI(I+1,J)-U(I-1,J,K)
   33             C*V(I-1,J,K)*HI(I-1,J))/(2*DX)
   34              D1HVVY=(V(I,J+1,K)*V(I,J+1,K)*HI(I,J+1)-V(I,J-1,K)*
   35             CV(I,J-1,K)*HI(I,J-1))/(2*DY)
   36              D1UZ=(U(I,J,K+1)-U(I,J,K-1))/(2.*DZ)
   37              D1VZ=(V(I,J,K+1)-V(I,J,K-1))/(2.*DZ)
   38              D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
   39              D1UY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
   40              D1VX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
   41              D1VY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
   42              D1UWZ=(U(I,J,K+1)*W(I,J,K+1)-U(I,J,K-1)*W(I,J,K-1))/(2*DZ)
   43              D1VWZ=(V(I,J,K+1)*W(I,J,K+1)-V(I,J,K-1)*W(I,J,K-1))/(2*DZ)
   44              DDUX=(U(I+1,J,K)+U(I-1,J,K)-D(I,J,K))/(DX*DX)
   45              DDUY=(U(I,J+1,K)+U(I,J-1,K)-D(I,J,K))/(DY*DY)
   46              DDUZ=(U(I,J,K+1)+U(I,J,K-1)-D(I,J,K))/(DZ*DZ)
   47              DDVX=(V(I+1,J,K)+V(I-1,J,K)-E(I,J,K))/(DX*DX)
   48              DDVY=(V(I,J+1,K)+V(I,J-1,K)-E(I,J,K))/(DY*DY)
   49              DDVZ=(V(I,J,K+1)+V(I,J,K-1)-E(I,J,K))/(DZ*DZ)
   50              H(I,J,K)=(DT/C)*(-AI*(D1HUUX+D1HUVY+HI(I,J)*D1UWZ)-HI(I,J)*AP
   51             C*D1PX+AH*HI(I,J)*(DDUX+DDUY)+AH*HX(I,J)*D1UX+AH*HY(I,J)*D1UY
   52             C+AV*(A3(K)*DDUZ+D1B3Z*D1UZ)/HI(I,J))+HI(I,J)*U(I,J,K)/C
   53              G(I,J,K)=(DT/C)*(-AI*(D1HUVX+D1HVVY+HI(I,J)*D1VWZ)-HI(I,J)*AP
   54             C*D1PY+AH*HI(I,J)*(DDVX+DDVY)+AH*HX(I,J)*D1VX+AH*HY(I,J)*D1VY
   55             C+AV*(A3(K)*DDVZ+D1B3Z*D1VZ)/HI(I,J))+HI(I,J)*V(I,J,K)/C
   56        8     CONTINUE
```

```
      9      CONTINUE
     10      CONTINUE
             RETURN
             END
```

9.2.71 UVTOP

This progtam computes U and V at the top using wind stress boundary conditions. Computations are made for MAR < 11 only. (Interior grid points).

```
*DOC.UVTOP
      C*****************************************************************************
      C       THIS PROGRAM CALCULATES U AND V VELOCITIES AT THE SURFACE USING STRESS
      C                BOUNDARY CONDITIONS
      C*****************************************************************************
            SUBROUTINE UVTOP(H,G,TAUX,TAUY,I,J,K,DZ,IN,JN,KN,HI,MAR)
            DIMENSION H(IN,JN,KN),G(IN,JN,KN)
            DIMENSION HI(IN,JN),MAR(IN,JN)
            DO 800 I=1,IN
            DO 800 J=1,JN
            IF (MAR(I,J).LT.11) 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
```

9.2.72 VERTDF

This program compute vertical viscosity and diffusivity as a function of temperature gradient in the vertical direction. The program also computes first derivative of vertical eddy viscosity and diffusivity with respect to $\gamma$.

```
M*DULL(1).VERTDF
    1              SUBROUTINE VERTDF(I,J,K,IN,JN,KN,HI,AB3,D1A3Z,D1B3Z,DZ,T,A3,TREF
    2             C,CONS,AVMX,AVMN)
    3              DIMENSION HI(IN,JN),T(IN,JN,KN),A3(KN)
    4              DO 50 KK=1,KN
    5              IF(KK.EQ.1) GO TO 11
    6              IF(KK.EQ.KN) GO TO 12
    7              D1TZ=(T(I,J,KK+1)-T(I,J,KK-1))/(2.*DZ)
    8              GO TO 20
    9        11    D1TZ=(4.*T(I,J,KK+1)-3.*T(I,J,KK)-T(I,J,KK+2))/(2.*DZ)
   10              GO TO 20
   11        12    D1TZ=(-4.*T(I,J,KK-1)+3.*T(I,J,KK)+T(I,J,KK-2))/(2.*DZ)
   12        20    PARA=-HI(I,J)*(((KK-1)*DZ)**2)*D1TZ
   13              A3(KK)=AVMX/(1.+PARA*CONS)
   14              IF(A3(KK).LT.AVMN) A3(KK)=AVMN
   15              DEPTH=(FLOAT(KK-1))*DZ*HI(I,J)
   16              IF(DEPTH.GT.0.30) A3(KK)=AVMN
   17        50    CONTINUE
   18              IF(K.EQ.1) D1A3Z=0.0
   19              IF(K.EQ.KN) D1A3Z=(-4.*A3(K-1)+3.*A3(K)+A3(K-2))/(2.*DZ)
   20              IF(K.GT.1.AND.K.LT.KN) D1A3Z=(A3(K+1)-A3(K-1))/(2.*DZ)
   21              AB3=A3(K)
   22              D1B3Z=D1A3Z
   23              IF(I.EQ.24.AND.J.EQ.6) WRITE(6,100) I,J,K,AB3,D1A3Z,D1B3Z
   24        100   FORMAT (3I5,3F15.8)
   25              RETURN
   26              END
RT,5 J.WHTOP
```

9.2.73 WHATIJ

This program computes the values of W at I, J from the values of WH at IW, JW.

Vertical velocity at a point on the main grid point is taken to be equal to the average of vertical velocities at four points on the half grid and lying adjacent to the point under consideration. Computations are made for interior grid points only.

```
*DOC.WHATIJ
  1       C*************************************************************************
  2       C     THIS PROGRAM CALCULATES THE VALUE OF W AT I,J FROM VALUES OF WH AT IW,JW
  3       C*************************************************************************
  4             SUBROUTINE WHATIJ(I,J,K,IW,JW,IN,JN,KN,IWN,JWN,W,WH,MAR)
  5             DIMENSION WH(IWN,JWN,KN),W(IN,JN,KN)
  6             DIMENSION MAR(IN,JN)
  7             DO 3550 I=1,IN
  8             DO 3550 J=1,JN
  9             IF (MAR(I,J).LT.11) GO TO 3540
 10             DO 3510 K=1,KN
 11             IW=I
 12             JW=J
 13             W(I,J,K)=(WH(IW,JW,K)+WH(IW,JW-1,K)+WH(IW-1,JW,K)+WH(IW-1,JW-1,K))
 14            C/4.
 15       3510  CONTINUE
 16       3540  CONTINUE
 17       3550  CONTINUE
 18             RETURN
 19             END
```

9.2.74 WHTOP

This program sets the value of WH equal to zero at the surface.

$WH = \frac{\partial \gamma}{\partial t} = \Omega = 0$ at $\gamma = 0$

No computations are made for points outside of the region of interest, defined by MRH = 0.

```
*DOC.WHTOP
  1       C*************************************************************************
  2       C     THIS PROGRAM SETS THE VALUE OF WH EQUAL TO ZERO AT THE SURFACE
  3       C*************************************************************************
  4             SUBROUTINE WHTOP(IW,JW,IWN,JWN,KN,WH,K,MRH)
  5             DIMENSION WH(IWN,JWN,KN)
  6             DIMENSION MRH(IWN,JWN)
  7             DO 3300 IW=1,IWN
  8             DO 3300 JW=1,JWN
  9             IF (MRH(IW,JW).EQ.0) GO TO 3000
 10             WH(IW,JW,1)=0
 11       3000  CONTINUE
 12       3300  CONTINUE
 13             RETURN
 14             END
```