

N O T I C E

THIS DOCUMENT HAS BEEN REPRODUCED FROM
MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT
CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED
IN THE INTEREST OF MAKING AVAILABLE AS MUCH
INFORMATION AS POSSIBLE

(NASA-CR-154624-Vol-3) THREE DIMENSIONAL
THERMAL POLLUTION MODELS. VOLUME 3: FREE
SURFACE MODELS Final Report (Miami Univ.)
231 p HC A11/MF A01

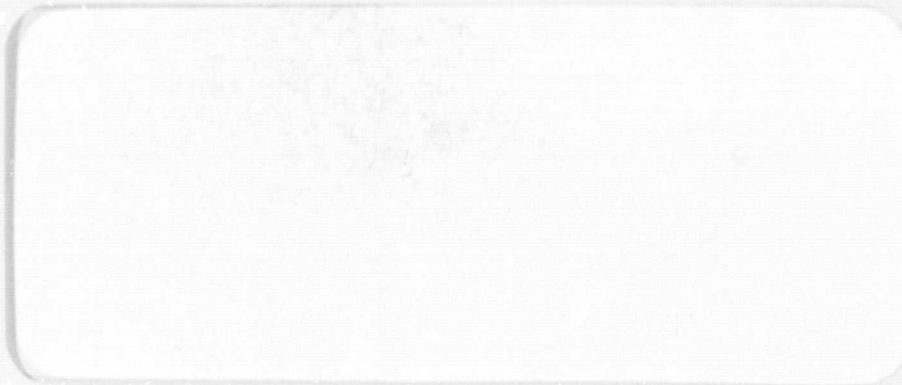
CSSL 13B

N80-26752

Unclas

G3/43 33430

DEPARTMENT OF MECHANICAL ENGINEERING

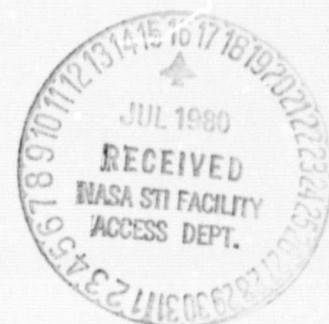


School of Engineering and Architecture

UNIVERSITY OF MIAMI



Coral Gables, Florida 33124



CR-154624
May 1978

Volume III: Free Surface Models

Three Dimensional Thermal Pollution Models

National Aeronautics and
Space Administration

John F. Kennedy Space Center

NASA

Volume III: Free Surface Models

Three Dimensional Thermal Pollution Models

NASA Contractor Report CR-154624

Contract NAS10-8926
May 1978

Samuel S. Lee and Sabrata Sengupta

**THREE-DIMENSIONAL THERMAL
POLLUTION MODELS
VOLUME III - FREE SURFACE 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 third of a three volume set presenting the description and program documentation of a mathematical model package for thermal pollution analyses and prediction. Two sets of programs, both in the free-surface formulation, are presented and clearly explained in this volume. These programs were developed by the Thermal Pollution Group at the University of Miami, and were funded by NASA, thus the program names NASUM II and NASUM III were given to reflect this joint effort.

These models are three-dimensional and time dependent using the primitive equation approach. They have sufficient generalality in programing procedure to allow application at sites with diverse topographical features. Both programs predict surface height variations, velocity field and temperature field for the "complete field". In the case of NASUM II a far-field formulation is used without including the plant thermal discharge; and in the case of NASUM III, a horizontal stretching is used to take account of the plant thermal discharge, and also to include far-field influences such as varying tide and ambient currents at points sufficiently far from the point of discharge.

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

TABLE OF CONTENTS, VOLUME III

1	Preface	i
ii	List of Symbols for Equations	1
1.	Introduction	5
2.	Program Descriptions	8
2.1	NASUM II (Far-Field)	8
2.1.1	Description - Algorithm	8
2.1.2	Flow Chart	9
2.2	NASUM III (Horizontally Stretched)	
2.2.1	Description - Algorithm	9
2.2.2	Flow Chart	10
3.	List of Program Symbols with Explanation (Far Field and Horizontally Stretched)	11
4.	Main Programs	20
4.1	NASUM II (Far Field)	20
4.2	NASUM III (Horizontally Stretched)	22
5.	INPUT DATA	23
5.1	NASUM II (Far Field)	23
5.1.1	Data Required and Format	24
5.2	NASUM III (Horizontally Stretched)	25
5.2.1	Data Required and Format	25
6.	Sample Cases	26
6.1	NASUM II (Far Field)	26
6.1.1	Problem Statement	26
6.1.2	Choice of Subroutine Programs	26

6.1.3	Calculation of Input Parameters	26
6.1.4	Sample Input	29
6.1.5	Program Execution Procedure	32
6.1.6	Sample Output	33
6.2	NASUM III (Horizontally Stretched)	41
6.2.1.	Problem Statement	41
6.3	Calculation of Input Parameters	41
6.3.1	Grid System	41
6.3.2	Calculation of Discharge Velocity	42
6.3.3	Reference Quantities	42
6.3.4	Calculation of Input Data as It Appears in the Main Program	43
6.3.5	Program Execution Procedure (NASUM III)	47
6.3.6	Sample Input	48
6.3.7	Sample Output	49
7.	Description of Computer Programs	
7.1	Main Program and Subroutines for NASUM II.	53
7.2	Main Program and Subroutines for NASUM III.	149
8.	References	217
9.	Table and Figures	218
10.	Appendices	231
A.	Wind Stresses	231
B.	Heat Transfer Mechanisms	232
C.	The Equilibrium Temperature and the Surface Heat Transfer Coefficient	236

(ii) LIST OF SYMBOLS

The following list of symbols which are obtained from Volume 1 are presented here for convenience.

A_1	First term in $\eta_0(t)$, which is defined below
A_2	Coefficient of second term in $\eta_0(t)$
C_0	phase velocity of surface gravity waves, \sqrt{gH}
C_p	specific heat at constant pressure
f	Coriolis parameter
g	acceleration due to gravity
h	depth relative to the mean water level
H	depth contour relative to free surface, $h + \eta$
I	grid index in x-direction or α direction
J	grid index in y-direction or β direction
K	grid index in z-direction or σ direction
k	thermal conductivity
K_H	horizontal eddy viscosity
K_V	vertical eddy viscosity
K_s	surface heat transfer coefficient
l	width of bay at ocean-bay interface
L	horizontal length scale
P	pressure
P_s	surface pressure

T	Temperature
T_{air}	Air temperature
T_{amb}	Water ambient temperature
T_e	Equilibrium temperature
t	time
t_ϕ	time lag in $\eta_0(t)$, which is defined below
u	velocity in x-direction (dimensional)
v	velocity in y-direction (dimensional)
v_0	Amplitude of inlet tidal velocity $v_0(t)$
w	Velocity in z-direction (dimensional)
x	Horizontal coordinate
y	Horizontal coordinate
z	Vertical coordinate

Greek Letters

α	horizontal coordinate in stretched system, = x
β	horizontal coordinate in stretched system, = y
σ	vertical coordinate in stretched system, = z/H
ρ	density
ϕ	phase angle in tidal current velocity
Ω	transformed vertical velocity
ω	angular frequency of tidal wave
τ_{xz}	surface shear stress in x-direction
τ_{yz}	surface shear stress in y-direction

η free surface elevation

$\eta_0(t)$ inlet tide level = $A_1 + A_2 \cos \omega(t+t_\phi)$

Horizontal Stretching Parameters

X horizontal stretching coordinate in x-direction

Y horizontal stretching coordinate in y-direction

X' $\frac{dX}{d\alpha}$

X'' $\frac{d^2X}{d\alpha^2}$

Y' $\frac{dY}{d\beta}$

Y'' $\frac{d^2Y}{d\beta^2}$

a the distance at which minimum step size is desired in x-direction (see transformation relation below)

b the distance at which minimum step size is desired in y-direction (see transformation relation below)

a, b, c₁, c₂, c₃, c₄, d and e are related and defined by the following relationships

$$\alpha = a + c_1 \sinh \{c_2(X-d)\}$$

$$\beta = b + c_3 \sinh \{c_4(Y-e)\}$$

1. INTRODUCTION

This volume contains descriptions of the NASUM II and NASUM III computer programs together with instructions on how to operate these programs. As outlined in Volume I of this report, NASUM II is a three-dimensional, time dependent, free-surface model intended for use in large domains where rather coarse resolution is satisfactory. In NASUM II the horizontal distance between nodes in the rectangular grid employed in the model is the same throughout the domain. NASUM III has the same basic three-dimensional, free-surface character as NASUM II but includes a form of "horizontal stretching" which provides fine resolution in some parts of the domain and coarse resolution in other parts. The horizontal distance between nodes in the rectangular grid employed in the model is consequently a function of location in the domain.

The program descriptions, associated algorithms, flow charts, program symbols, choice of input data, and sample problems for NASUM II and NASUM III are contained herein for the ready access of the computer programs by the user. Note, that the governing equations, approximations, simplifying assumptions, and numerical methods of solution are presented in Volume I.

NASUM II (the far-field version of the free surface model) has been applied to South Biscayne Bay, Florida and is presented with computer results in Lee and Sengupta's (1977) report on Three-Dimensional Thermal Pollution Models and in a paper by Sengupta, Lee and Miller (1977). The South Biscayne Bay is a relatively shallow estuary with the principal driving mechanism being tidal flux at the ocean-bay interface, although wind

effects are clearly evidenced in the northern part of the bay.

NASUM III (the horizontally stretched version of the free surface model) has been applied to Hutchinson Island, St. Lucie, Florida, which is a coastal site with a submerged discharge. Horizontal stretching was used in order to obtain resolution in the neighborhood of the discharge, while at the same time a large horizontal domain could be covered. If a constant grid size had been used, this would have required an excessively large number of grid points to cover the same extent of the boundaries of the domain. Therefore, in order to circumvent this problem, a hyperbolic sine (SINH) stretching transformation was used in both the horizontally lateral and transverse directions, respectively, to obtain a small grid size in the neighborhood of the discharge and an increasingly larger grid size as distance from the discharge point increased. Waldrop and Farmer (1973) suggested a tangent (TAN) stretching transformation; however, the SINH stretching transformation was found to have advantages in this study. The details of the comparison between the tangent and sinh stretching formulas are presented in Volume I. Lee and Sengupta (1977) and Tsai (1977) present the results of this Hutchinson Island investigation.

The effects of variable bottom topography, spatio-temporal free surface variations, surface heat transfer based on the equilibrium temperature concept introduced by Edinger and Geyer (1971), tide level variation, resultant ambient currents, and meteorological conditions have been factored into these models. In addition, turbulence has been modelled by using the eddy transport concept, and the effects of baroclinicity have been

included. Again, the user should refer to Volume I for the complete mathematical formulations, approximations and assumptions, and the numerical methods of solution.

2. Program Descriptions

This section presents the computer program algorithms and associated flow charts (in standard notation. c.f. Murrill and Smith (1975) for the NASUM II and NASUM III.

2.1 NASUM II (The Far Field Model)

2.1.1 Description of Program Algorithm

The program algorithm for NASUM II is as follows:

- a) Integrate the surface height equation using forward-time, central-space differencing initially (FTCS), and, thereafter, central-time, central-space is used (CTCS).
- b) Integrate the u-momentum equation using forward-time, central-space differencing initially (FTCS), and, thereafter, central-time, central-space is used (CTCS) with DuFort-Frankel differencing applied to the vertical momentum diffusion term as given, for example, in Roache (1972).
- c) Integrate the v-momentum equation using forward-time, central-space differencing initially (FTCS), and, thereafter, central-time, central-space is used (CTCS) with DuFort-Frankel differencing applied to the vertical momentum diffusion term.
- d) The equivalent vertical velocity, Ω , is then computed by knowing H, u and v. The spatial integration is performed by applying Simpson's rule. (c.f. Crandell (1955)).
- e) The energy equation is then integrated over time using forward-time, central-space (FTCS) throughout.
- f) The density is calculated from the equation of state.
- g) The pressure field is calculated from the hydrostatic equation using the trapezoidal rule for spatial integration.
- h) Then, the value of computed real time (or simulation time) is checked and the steps a) through g) repeated if so

desired. Reference to the flow chart presented in Fig. 1 will clarify this last step in the program algorithm.

2.1.2 Flow Chart

The flow chart for NASUM II is presented in Fig. 1.

2.2 NASUM III (The horizontally stretched model)

2.2.1 Description of Program Algorithm

The program algorithm for NASUM III is as follows:

- a) Integrate the surface height equation using forward-time, central-space differencing initially (FTCS), and , thereafter, central-time, central-space is used (CTCS).
- b) Integrate the u-momentum equation using forward-time, central-space differencing initially (FTCS), and, thereafter, central-time, central -space is used (CTCS) without DuFort - Frankel differencing applied to the vertical momentum diffusion then, since the vertical diffusion term does not govern the time step value as it does for a shallow estuary.
- c) Integrate the v-momentum equation using forward-time, central-space differencing initially (FTCS), and, thereafter, central-time, central-space is used (CTCS).
- d) The equivalent vertical velocity, Ω , is then computed by knowing H, u and v. However, for a submerged discharge Ω at the bottom of the basin is not zero (c.f. Volume I). The spatial integration is performed by applying the trapezoidal rule.
- e) The energy equation is then integrated over time using forward-time, central-space (FTCS) initially, and, thereafter, central-time, central-space (CTCS) is used.
- f) The density is calculated from the equation of state.
- g) The pressure field is calculated from the hydrostatic equation using the trapezoidal rule for spatial integration.

h) Then, the value of computed real time (or simulation time) is checked and the steps a) through g) repeated if so desired. Reference to the flow chart presented in Fig. 2 will clarify this last step in the program algorithm.

2.2.2 Flow Chart

The flow chart for NASUM III is presented in Fig. 2.

3. LIST OF PROGRAM SYMBOLS

3.1 Symbols Common to Far-Field and Horizontally Stretched Model Programs

This section presents in alphabetical order the program symbols, in FORTRAN language, and their definition for those symbols which are common to the Far-Field and Horizontally Stretched Models. In many cases the definition is shortened by referring to algebraic symbols already defined in section (ii) of this volume.

A

- A1 : first term in $\eta_0(t)$
 A2 : Coefficient of second term in $\eta_0(t)$

B

- BH : B_H
 BV : B_V

C

- CI : Coefficient of inertia term in momentum equations
 CC : Coefficient of Coriolis term in momentum equations
 CP : Coefficient of pressure term in momentum equations
 CH : Coefficient of horizontal diffusion term in momentum equations
 CV : Coefficient of vertical diffusion term in momentum equations

D

D(I,J,K)	:	$u(\alpha, \beta, \sigma)$ at $t = t + \Delta t$
DT	:	time step, Δt
DX	:	Grid size in α - direction, $\Delta\alpha$
DY	:	Grid size in β - direction, $\Delta\beta$
DZ	:	Grid size in α - direction, $\Delta\sigma$
DUM	:	$\frac{\sigma}{H} \int_0^1 \left\{ \frac{\partial(Hu)}{\partial\alpha} + \frac{\partial(Hv)}{\partial\beta} \right\} d\sigma$
DIHUTX	:	$\frac{\partial(HuT)}{\partial\alpha}$
DIHUUX	:	$\frac{\partial(Huu)}{\partial\alpha}$
DIHUVX	:	$\frac{\partial(Huv)}{\partial\alpha}$
DIHUVY	:	$\frac{\partial(Huv)}{\partial\beta}$
DIHUX	:	$\frac{\partial(Hu)}{\partial\alpha}$ at $k = k$
DIHUX1	:	$\frac{\partial(Hu)}{\partial\alpha}$ at $k = k-1$
DIHVTY	:	$\frac{\partial(HvT)}{\partial\beta}$
DIHVVY	:	$\frac{\partial(Hvv)}{\partial\beta}$
DIHVVY	:	$\frac{\partial(Hv)}{\partial\beta}$ at $k = k$
DIHVY1	:	$\frac{\partial(Hv)}{\partial\beta}$ at $k = k-1$
DIPX	:	$\frac{\partial p}{\partial\alpha}$
DIPY	:	$\frac{\partial p}{\partial\beta}$

$$D1TX : \frac{\partial T}{\partial \alpha}$$

$$D2TX : \frac{\partial^2 T}{\partial \alpha^2}$$

$$D1TY : \frac{\partial T}{\partial \beta}$$

$$D2TY : \frac{\partial^2 T}{\partial \beta^2}$$

$$D2TZ : \frac{\partial^2 T}{\partial \sigma^2}$$

$$D1UX : \frac{\partial u}{\partial \alpha}$$

$$D2UX : \frac{\partial^2 u}{\partial \alpha^2}$$

$$D1UY : \frac{\partial u}{\partial \beta}$$

$$D2UY : \frac{\partial^2 u}{\partial \beta^2}$$

$$D1UWZ : \frac{\partial (u\Omega)}{\partial \sigma}$$

$$D2UZ : \frac{\partial^2 u}{\partial \sigma^2}$$

$$D1VX : \frac{\partial v}{\partial \alpha}$$

$$D2VX : \frac{\partial^2 v}{\partial \alpha^2}$$

$$D1VY : \frac{\partial v}{\partial \beta}$$

$$D2VY : \frac{\partial^2 v}{\partial \beta^2}$$

$$D1VWZ : \frac{\partial (v\Omega)}{\partial \sigma}$$

$$D2VZ : \frac{\partial^2 v}{\partial \sigma^2}$$

$$D1WZ : \frac{\partial (\Omega T)}{\partial \sigma}$$

E E(I, J, F) : v(α, β, σ) et t=t+Δt

ETA(I, J) : η(α, β)

ETAX(I, J) : $\frac{\partial \eta}{\partial \alpha}$

ETAY(I, J) : $\frac{\partial \eta}{\partial \beta}$

F FF : f

G G(I, J, K) : u(α, β, σ) at t=t

GR : g

H H(I, J, K) : u(α, β) at t=t
 H(I, J) : H(α, β) at t=t in energy equation

$$HDUM(I, J) : - \int_0^1 \left\{ \frac{\partial (Hu)}{\partial \alpha} + \frac{\partial (Hv)}{\partial \beta} \right\} \alpha \sigma$$

HI(I,J): $h(\alpha, \beta)$
 HNI(I,J): $H(\alpha, \beta)$ at $t=t + \Delta t$ in energy equation
 HS: K_s
 HT(I,J): $H(\alpha, \beta)$ at $t=t - \Delta t$ in momentum equations
 HTD(I,J): $H(\alpha, \beta)$ at $t=t$ in momentum equations
 HTE(I,J): $H(\alpha, \beta)$ at $t=t + \Delta t$ in momentum equations
 HTMIN: Minimum value of $H(\alpha, \beta)$ in bay
 HX(I,J): $\frac{\partial h}{\partial \alpha}$
 HY(I,J): $\frac{\partial h}{\partial \beta}$

I

I: Index in α - direction
 I1: Lower index for inlet along MAR=1*
 I2: Upper index for inlet along MAR=1
 I3: Lower index for outlet along MAR=2
 I4: Upper index for outlet along MAR=2
 I5: Index for inlet along MAR=3
 I6: Index for outlet along MAR=4
 IBAY: Parameter which when specified either provides
 a constant time step for a shallow bay or for a
 deep bay.
 (=0 for shallow bay; =1 for deep bay)
 IHTE: =0 for specifying initial surface
 =1 for not specifying initial surface
 IN: Number of grid points in α - direction
 INLET: =1 for inlet along MAR=1
 =2 for inlet along MAR=3
 IRUN: =0 for first run
 =1 thereafter

*NOTE: The MAR(I,J) matrix is explained in section 6.1.3. Fig.3 illustrates the location of I1 through I6 and J1 through J6.

- J J: Index in β - direction
 J1: Index for inlet along MAR=1
 J2: Index for outlet along MAR=2
 J3: Lower index for inlet along MAR=3
 J4: Upper index for inlet along MAR=3
 J5: Lower index for outlet along MAR=4
 J6: Upper index for outlet along MAR=4
 JN: Number of grid points in β - direction
- K K: Index in σ - direction
 KH: K_H
 KN: Number of grid points in σ - direction
 KV: K_V
- L L: Index of time cycle without energy equation
 LL: Index of time cycle with energy equation
 LN: Number of time cycles without energy equation
 LNI: Number of time cycles with energy equation
- M M: Parameter for either specifying $V_o(t)$ at the inlet or
 specifying $\eta_o(t)$ at the inlet (=1 for V_o , =2 for η_o case)
 MAR(I,J): Numbering system for grid system - used to
 distinguish between different boundary finite difference
 schemes.
- P P(I,J,K): P(α, β, σ)
- R RO(I,J,K): $\rho(\alpha, \beta, \sigma)$ for variable density case
 RR: $\rho(\alpha, \beta, \sigma)$ for constant density case
- T T(I,J,K): T(α, β, σ) at $t=t$
 TA: Te

TAUX: τ_{zx}
 TAUY: τ_{zy}
 TFLAT: Time interval from initial flat surface ($\eta_0=0$) to some desired hour
 THT: Time interval from start-up to high tide
 TI: Initial temperature for isothermal bay
 TNI(I,J,K): $T(\alpha, \beta, \sigma)$ at $t=t + \Delta t$
 TPH: Time lag for $V_0(t)$
 TPHI: Time lag for $\eta_0(t)$
 TT: TTOT + TTOTI
 TTOT: Total run time without energy equation
 TTOTb: Total run time with energy equation

U U(I,J,K): $u(\alpha, \beta, \sigma)$ at $t = t - \Delta t$

V V(I,J,K): $v(\alpha, \beta, \sigma)$ at $t=t - \Delta t$
 VO: V_0 amplitude of $V_0(t)$ at inlet

W W(I,J,K): $\Omega(\alpha, \beta, \sigma)$
 WUD: $-\frac{1}{H} \int_0^\sigma \left[\frac{\partial (Hu)}{\partial \alpha} + \frac{\partial (Hv)}{\partial \beta} \right] d\sigma$
 WZ(I,J,K): $\omega(\alpha, \beta, \sigma)$

3.2 Additional Symbols for Horizontally Stretched Model Program

This section presents in alphabetical order the program symbols, in FORTRAN language, and their definition for those additional symbols for the Horizontally Stretched Model. Again, symbols not defined here have already been defined in section (ii).

A

A: Value of $(X - d)/C_1$

B

B: Value of $(Y - e)/C_3$

D

DEEX: Value of C_1

DEEY: Value of C_3

DELX: Grid size in α - direction, $\Delta\alpha$

DELY: Grid size in β - direction, $\Delta\beta$

DHDX: $\frac{\partial H}{\partial \alpha}$ at $t=t$

DHDY: $-\frac{\partial H}{\partial \beta}$ at $t=t$

E

EEEX: Value of d

EEEY: Value of e

H

HK: K_s

HTX: $\frac{\partial H}{\partial \alpha}$

HTY: $-\frac{\partial H}{\partial \beta}$

T

T(I,J,K): $T(\alpha, \beta, \sigma)$ at $t=t - \Delta t$

TN(I,J,K): $T(\alpha, \beta, \sigma)$ at $t=t$

TF(I,J,K): $T(\alpha, \beta, \sigma)$ at $t=t + \Delta t$

TAIR: Air temperature

TAM: Ambient temperature of water body

U

UM: $u(\alpha, \beta, \sigma)$

V

VM: $v(\alpha, \beta, \sigma)$

W

WH: $w(\alpha, \beta, \sigma)$

X

XX: $\frac{dX}{d\alpha}$

XXX: $\frac{d^2X}{d\alpha^2}$

Y

$$YY: \frac{dY}{d\beta}$$

$$YYY: \frac{d^2 Y}{d\beta^2}$$

4. MAIN PROGRAMS

This section presents a detailed description of the main programs for the NASUM II and NASUM III. The main programs themselves appear in Section 7.1.

4.1 NASUM II (Far-Field Main Program)

The following main program outline and associated description is for the far-field version of the free surface model. The main program name is FMAIN, and appears in Section 7.1.

- a) Specify number of grid points, IN, JN and KN in PARAMETER statement (although the geometry of the domain of solution under consideration will not cover all the grid points; MAR(I,J)=0 covers range of grid points outside the domain of solution, where MAR(I,J) is constructed as shown in Fig. 5 for application to the South Biscayne Bay).
- b) Specify IRUN=0 or 1. The value 0 is used for the first run only, and 1 is used thereafter.
 - For IRUN = 0, READ2 and INITIA are used.
 - For IRUN = 1, READ1 is used.
- c) Specify LN, LN1, M, INLET, IBAY, IHITE, I1, I2, I3, I4, I5, I6, J1, J2, J3, J4, J5, J6, VO, TPH, TPH1, A1, A2, HTMIN, THT, TFLAT, CI, CC, CP, CH, CV, GR, FF, RR, DX, DY, DZ, KH, KV, BH, BV, TI. See section 3.1 for definition of these symbols, and refer to section 6.1.4, to follow, for a sample input of these parameters.
- d) Specify TAUX, TAUY, TA, and HS, as defined in section 3.1, each hour.
- e) Specify DT as defined in section 3.1. (in seconds)
- f) For L=1, TTOT=0.0: Energy equation is not coupled to the system of governing equations. The following subroutines are used:
 - HEIGHT
 - TIDE or VEL
 - DATA
 - UVVEL
 - WVEL
 - PRES
 - ETT
- g) For L>1: Energy equation is not coupled to the system of governing equations, but central-time is used now after the first time step has been executed (for L=1).

HEILN
TIDE or VEL
DATA
UVVELN
WVEL
PRES
ETT
OLDHT
OLDUV

h) For $LL > 1$, $TT - TTOT + TTOT1 > 0$: Energy equation is coupled to the system of governing equations. The following subroutines are used:

HEILN
TIDE or VEL
TIDAL
DATA
UVVELN
WVEL
PRES
ETT
OLDHT
OLDUV
TEMP
OLDT

i) After the final time cycle is computed, the following subroutines are used for printing and storing on magnetic tape:

PRPARA
PRETA
PRUV
WW
PRW
PRTEMP
STORE

4.2 NASUM III (Horizontally Stretched Main Program)

The following main program outline and associated description is for the horizontally stretched version of the free surface model. The main program name is FMAIN, and appears in Section 7.1.

- a) Specify number of grid points, JN, JN and KN in PARAMETER statement for the domain of interest.
- b) Read in all the data required and logic parameters IRUN, LN, CI, CC, CP, CH, CV, GR, FF, RR, HK, DX, DY, DZ, KH, KV, BH, BV, TAUX, TAVY, TAIR, DT, DELX, DELY, DEEX, DEEY, EEEEX, EEEY. See sections 3.1 and 3.2 for definition of these symbols; and refer to Section 6.2.4, to follow, for a sample input of these parameters.
- c) Generate a two-dimensional matrix MAR(I,J) for locating the position of the points in the domain.
- d) Initialize all the necessary quantities, as defined in Section 3.1; specify the discharge conditions and bottom topography.
- e) Convert the real vertical velocities W into the transformed sigma coordinate vertical velocities, Ω .
- f) Calculate the horizontal stretching parameters, X' , X'' , Y' , Y'' for the set of governing equations.
- g) Calculate the new predicted dependent variables.
- h) Store the data and new predicted dependent variables on magnetic tape and print out these values at the desired time step.
- i) For L=1, TTOT=0.0: Forward-time differencing is used. The following subroutines are used:

HEIGHT
 UVVEL
 TEMP
 PRES
 ETT

- j) For $L > 1$ or $TTOT > DT$: Central-time differencing is used.
The following subroutines are used:

HEILN
UVVELN
WVEL
TEMPN
PRES
ETT
OLDUVT
OLDUV

- k) After the final time cycle is computed, the following subroutines are used for printing and storing on magnetic tape:

STORE
PRPARA
PRETA
PRUV
PRW
PRTEM

5. INPUT DATA

The data that is required for the execution of the main program in either NASUM II or NASUM III is called Input Data. The data required is listed in the order it appears in the respective programs, and the corresponding FORMAT (in FORTRAN language) is given corresponding to each data symbol. Section 5.1.1 lists the data input required for running NASUM II, and Section 5.2.1 lists the data input required for running NASUM III. The actual calculation required for several of the input data is given in Section 6.1.3 for NASUM II, and in 6.2.3 for NASUM III. Note, the data input symbols have already been defined in Section 3 of this volume.

5.1 NASUM II(Far-Field Model)

The following number of computer data cards, with proper FORMAT, is now given in order as they appear in the main program in order to execute NASUM II (refer to Section 3.1 for definition of these FORTRAN symbols). The data that must be calculated beforehand is given in Section 6.1.3.

5.1.1 DATA REQUIRED AND FORMAT

<u>CARD NO.</u>	<u>DATA</u>	<u>FORMAT</u>
1	IRUN	I5
2	LN	I5
3	LN1	I5
4	M	I5
5	INLET	I5
6	IBAY	I5
7	IHITE	I5
8	I1, I2, I3, I4, I5, I6	6I5
9	J1, J2, J3, J4, J5, J6	6I5
10	VO, TPH, TPH1, A1, A2	Free
11	HTMIN, THT	Free
12	CI, CC, CP, CH, CV	Free
13	GR, FF, RR	Free
14	DX, DY, DZ	Free
15	KH, KV	Free
16	BH, BV	Free
17	TI	Free
18	DTAUX(13)**	Free
19	DTAUY(13)	Free
20	DTA(13)	Free
21	DHS(13)	Free
22	DT	Free

**NOTE: 13 values of TAUX and TAUY and 13 values of TA and HS are read in for variation each hour. Note, the letter "D" precedes previously defined symbols (Section 3.1), since this was necessary for computer convenience.

5.2 NASUM III (Horizontally Stretched Model)

The following number of computer data cards, with proper FORMAT, is now given in order as they appear in the main program in order to execute NASUM III (refer to Section 3.1 and 3.2) for definition of these FORTRAN symbols). The data that must be calculated beforehand is given in Section 6.2.3.

5.2.1 DATA REQUIRED AND FORMAT

<u>CARD NO.</u>	<u>DATA</u>	<u>FORMAT</u>
1	IRUN	I5
2	LN	I5
3	CI, CC, CP, CH, CV	Free
4	GR, FF, RR, HK	Free
5	DX, DY, DZ	Free
6	KH, KV, BH, BV	Free
7	TAUX, TAUY	Free
8	TAIR	Free
9	DT	Free
10	DELX	Free
11	DELY	Free
12	DEEX	Free
13	DEEY	Free
14	EEXX	Free
15	EEY	Free

6. SAMPLE CASES

The following sample cases will illustrate and clarify to the user the proper choice of programs, subprograms (or subroutines), calculation of input parameters, sample input and sample output for NASUM II and NASUM III.

6.1 NASUM II (Far Field Model)

6.1.1 Problem Statement - Application to Biscayne Bay

Given Biscayne Bay, in Dade County, Florida, as an example application site, compute the surface heights, η , velocity field u, v, w , and the temperature distribution T at 2:00 P.M. knowing the meteorological data, the IR data base, and the tide data base for April 15, 1975. The IR data base is assumed synoptic at 2:00 P.M., the wind velocity and ambient temperature is known every hour, and the tide height, with respect to the mean water level, is known as a function of time at the ocean bay interface.

Use the NASUM II far-field, free surface model program to obtain the desired results

6.1.2 Choice of Subroutine Programs

Section 4.1 is followed in order to choose the proper subroutine programs for this sample case. Sections 6.1.3 and 6.1.4 to follow next, will clearly illustrate what steps the user must follow in order to obtain the desired results for this sample case.

6.1.3 Calculation of Input Parameters

a) Construct a three-dimensional grid system for the Biscayne Bay. Fig. 4 illustrates the horizontal grid for the bay superimposed on the actual geometry of the domain of interest. The governing equations have been transformed into the α, β, σ coordinate system, which maps

the variable depth basin into a constant depth basin. Then, depending on the desired resolution of vertical structure, the number of vertical grid points is selected, that is KN. The values of IN and JN are then selected with consideration of desired horizontal resolution versus computer storage and overall computation time. Then, the next step is to specify IN, JN, KN in the main program, as mentioned in section 4.1. For this application IN=34, JN=11, KN=5.

b) Next, LN and LN1 as defined in section 3.1 are specified. First, however, the time step DT is computed based on the criteria given in Volume I. For the Biscayne Bay, DT is computed by the vertical momentum diffusion criterion:

$$DT = \Delta t < \frac{(z \text{ minimum})^2}{2} / K_v = \frac{(.25)^2 (60.96 \text{ cm})^2}{2} / 5 \text{ cm}^2 / \text{sec.}$$

Thus, DT = 10 sec. is chosen for this sample case. in order to ensure numerical stability.

For starting the program at $\eta_0 (t=0) = 0$ for April 15, 1975 LN= 1342, LN1=1380, since the program is started at 6:26 a.m. and run without the energy equation to 10:10 am, at which time the IR data base is read in, as an initial condition, from sub-routine TIDAL. Then, from 10:10 am to 2:00 P.M. the energy equation is included. This procedure ignores the effect of density currents on the momentum and surface height equations from 6:26 am to 10:10am. However, these density currents are quite small for the Biscayne Bay which is dominated by the wind and the tidal flux at the ocean-bay interface.

c) The eddy viscosity coefficients have been estimated by applying the "4/3 scaling law" to previously known water basin values used by other researchers.

d) Next, the matrix MAR(I,J) is constructed based on this particular grid system as shown in Fig. 5.

e) The depth matrix HI (I,J) is then constructed by specifying the depth below the mean water level, $h(\alpha, \beta)$ at each horizontal grid point.

f) The initial temperature matrix T (I,J,K) is constructed for the bay by first plotting the IR data base surface isotherms on the horizontal grid, and then interpolating to specify the temperature at each grid point (I,J). (See Fig. 6) The bay is shallow and well mixed vertically, hence, the vertical temperature variation is initially set equal to zero. Subroutine TIDAL reads in from data cards T(I,J,K).

g) The tidal current velocity amplitude V_0 is computed from the following formula, assuming a 90° phase shift between the tide height and tidal current velocity at the ocean-bay interface, (Ippen 1966):

$$V_0 = \left(\frac{2aC_0}{h} \right) \left(\frac{2\pi l}{\lambda} \right)$$

where $a = |\eta_0| = 37$ cm for April 15, 1975

$$C_0 = \sqrt{gh} \approx 4.3 \times 10^2 \text{ cm/sec}$$

$$l = 10\Delta\beta = 16 \times 10^5 \text{ cm}$$

$$\lambda = C_0 T = 12C_0 = 1.86 \times 10^7 \text{ cm} \gg l$$

$\langle h \rangle = 190$ cm, average at ocean-bay interface.

Thus, $V_0 = 90$ cm/sec

h) TPH = 3138.7 sec. is computed by letting $V_0(t) = 0$ at 10:10 A.M. (high tide where $V_0(t) = V_0 \cos \left[(t + \text{TPH}) \frac{2\pi}{T} \right]$ and $t=0$ at 8:00 A.M. on 4/15/75.

TPH1 = 7800 + FLAT = 13440 sec. (where 7800 sec = 8:00 A.M. to 10:00 A.M.)

where TFLAT = 5640 sec., the time from $\eta_0(t) = 0$ at 6:26 A.M. for April 15, 1975 to the beginning of the $V_0(t)$ run at 8:00 AM on April 15, 1975.

i) The equilibrium temperature, T_A , and the equilibrium coefficient of surface heat transfer H_S , are computed following Harleman and Stolzenbach (1973). Appendix C presents these formula.

j) The wind stresses $TAUX$, $TAUY$ are determined as shown in Appendix A.

6.1.4 Sample Input

The far-field solution is obtained by either specifying $V_o(t)$ or $n_o(t)$ at the ocean-bay interface. These two sample cases will now be given:

6.1.4.1 Velocity Case

IRUN = 0
 LN = 360 (begin at 0800 EST, 4/15/75, and compute until 0900 EST)
 LN1=1
 M=1 ($V_o(t)$ specified at inlet)
 INLET = 1 (ocean-bay interface along MAR=1)
 IBAY = 0 (shallow bay), i.e., constant time step of 10sec is used even in shallow regions - this was done to avoid inordinately small time steps during low water.
 IHITE = 1 (regression surface not read in at time of high tide)
 I1, I2, I3, I4, I5, I6= 7, 16, 31, 33, 35, 35(refer to Fig. 3)
 J1, J2, J3, J4, J5, J6= 11, 1, 12, 12, 12, 12(refer to Fig. 3)
 V_o , TPH, TPH1, A1, A2 = -90, 3138.7, -13420, 11.8872, 37.1856
 HTMIN, THT = 60.96, 7800.0
 TFLAT = 0.0
 CI, CC, CP, CH, CV=1., 1., 1., 1.,1.
 GR, FF, RR = 980., .00006, 1.
 DX, DY, DZ = 160,000., 160,000., .25
 KH, KV = 10,000., 5.
 BH, BV = 10,000., 5.

TI = 24.5

DTAUX (1) = -.37

DTAUY (1) = .15

DTA (1) = 31.7

DHS (1) = .00129

.

.

.

DT = 10

Next, IRUN = 1, LN = 420, LN1 = 1 (10:10am)

Next, IRUN = 1, LN = 1, LN1 = 1380 (2:00pm)

6.1.4.2 Tide Height Case

IRUN = 0

LN = 264 (begin at 0626 est, 4/15/75 and compute until 0710 est)

LN1 = 1

M = 2 ($\eta_0(t)$ specified at inlet)

INLET = 1 (ocean-bay interface closing MAR=1)

IBAY = 0 (shallow bay)

IHITE = 1 (regression surface not read in at time of high tide)

I1, I2, I3, I4, I5, I6 (refer to Fig. 3)

J1, J2, J3, J4, J5, J6 (refer to Fig. 3)

Vo, TPH, TPH1, A1, A2 = -90, 3138.7, -13420., 11.8872, 37.1856.

HIMIN, THT = 60.96, 13420.

TFLAT = 5640

CI, CC, CP, CH, CV = 1., 1., 1., 1., 1.

GR, FF, RR = 980., .00006, 1.

DX, DY, DZ = 160,000., 160,000., .25.

KH, KV = 10,000., 5.

BH, BV = 10,000, 5.

TI = 24.5

DTAUX(1) = -.37

DTAUY(1) = .15

DTA(1) = 31.7

DHS (1) = .00129

DT = 10

Next, IRUN = 1, LN = 300, LN1 = 1 (8:00am)

Next, IRUN = 1, LN = 300, LN1 = 1 (9:00am)

Next, IRUN = 1, LN = 420, LN1 = 1 (10:10am)

Next, IRUN = 1, LN = 1, LN1 = 1380 (2:00pm)

6.1.5 Program Execution Procedure (NASUM II)

This section describes the procedure by which the user executes the NASUM II program.

a) Input Parameters: The user must first follow the steps outlined in section 4.1, and become quite familiar with all the input parameters listed in section 5.1 and section 6.1.3.

b) First Run: In order to obtain surface heights and three-dimensional velocity and three-dimensional temperature, the main program FMAIN is executed. In FMAIN there are two tape units. One is a READ unit designated as Unit 7. The other is a STORE 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 on a magnetic tape.

c) Run Continuation: For extending the results, the run has to be continued. The magnetic tape which was "unit 8" in the first run will now be read "Unit 7", for reading the previously stored results. Another magnetic tape is now 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 extended runs IRUN=1.

6.1.6 Sample Output

The output from the model sample run is listed as follows:

- a) Parameters
- b) Surface heights
- c) Horizontal components of velocity
- d) Vertical velocity component
- e) Temperatures

ORIGINAL PAGE IS
OF POOR QUALITY

HEILN	S(1)	021715	023415	S(0)	140410	140614
	S(2)			S(2)	BLANKSCOMM	
UVVEL	S(1)	023416	030605	S(0)	140615	141215
	S(2)			S(2)	ELANKSCOMM	
UVVELN	S(1)	030606	036142	S(0)	141216	141626
	S(2)			S(2)	BLANKSCOMM	
MVEL	S(1)	036143	037413	S(0)	141627	141741
	S(2)			S(2)	ELANKSCOMM	
MVEL1	S(1)	037414	040735	S(0)	141742	142013
	S(2)			S(2)	BLANKSCOMM	
MVEL2	S(1)	040736	043073	S(0)	142014	142105
	S(2)			S(2)	ELANKSCOMM	
PRES	S(1)	043074	043346	S(0)	142106	142160
	S(2)			S(2)	BLANKSCOMM	
MW	S(1)	043347	044353	S(0)	142161	142317
	S(2)			S(2)	BLANKSCOMM	
TEMP	S(1)	044354	046154	S(0)	142320	142523
	S(2)			S(2)	ELANKSCOMM	
OLDUV	S(1)	046155	046425	S(0)	142524	142570
	S(2)			S(2)	BLANKSCOMM	
OLOHT	S(1)	046426	046546	S(0)	142571	142621
	S(2)			S(2)	BLANKSCOMM	
OLDT	S(1)	046547	046701	S(0)	142622	142656
	S(2)			S(2)	ELANKSCOMM	
EY	S(1)	046702	047024	S(0)	142657	142714
	S(2)			S(2)	BLANKSCOMM	
PRPARA	S(1)	047025	047170	S(0)	142715	143052
	S(2)			S(2)	ELANKSCOMM	
PRETA	S(1)	047171	047266	S(0)	143053	143103
	S(2)			S(2)	BLANKSCOMM	
PRUV	S(1)	047267	047453	S(0)	143104	143154
	S(2)			S(2)	BLANKSCOMM	
PRW	S(1)	047454	047602	S(0)	143155	143221
	S(2)			S(2)	ELANKSCOMM	
PRTEMP	S(1)	047603	047736	S(0)	143222	143263
	S(2)			S(2)	BLANKSCOMM	
STORE	S(1)	047737	050663	S(0)	143264	143356
	S(2)			S(2)	BLANKSCOMM	

SYSSPLIBS. LEVEL 7C-1
END OF COLLECTION - TIME 8.447 SECONDS

CI=	.1000000001
CNE	.1000000001
CVE	.1000000001
CPE	.1000000001
CCE	.1000000001
DX=	.1000000006
DVE	.1000000006
OZE	.2500000000
OFE	.1000000002
TAUX=	.1100000001
TAUVE	.2500000000
TIOTE	.1342000005
CNE	.5000000002

FF 6.300000
PR 1.2943101
KH 1.00000005
KV 5.00000001

BH 1.00000005
BV 5.00000001
ORAD 0.000000
TF 2.45000002
TTOT 1.00000005
TA 3.72000002
HS 1.16000002

I = 1
WAVE-HEIGHT
0.000000 0.000000 0.000000 0.000000 0.000000
-0.252646202 -0.1769363002 -0.000000 -0.000000 -0.000000 -0.109609602

I = 2
WAVE-HEIGHT
0.000000 0.000000 0.000000 0.000000 0.000000
-0.223393002 -0.662096001 -0.000000 -0.000000 -0.000000 -0.173919502 -0.915571801

I = 3
WAVE-HEIGHT
0.000000 0.000000 0.000000 0.000000 0.000000
-0.252667602 -0.1709285002 -0.140955902 -0.000000 -0.000000 -0.127990202 -0.253021602 -0.160366902

I = 4
WAVE-HEIGHT
0.000000 0.000000 0.000000 0.000000 0.000000
-0.210167902 -0.982902701 -0.1948603902 -0.000000 -0.000000 -0.116007902 -0.222151702 -0.110204502

I = 5
WAVE-HEIGHT
0.000000 0.000000 0.000000 0.000000 0.000000
-0.243498502 -0.1414262002 -0.231654202 -0.000000 -0.000000 -0.268748802 -0.166750002 -0.257505102 -0.150790402

I = 6
WAVE-HEIGHT
0.000000 0.000000 0.000000 0.000000 0.000000
-0.213975202 -0.977041601 -0.226070102 -0.000000 -0.000000 -0.229567002 -0.103448202 -0.219504002 -0.160009002

I = 7
WAVE-HEIGHT
0.000000 0.000000 0.000000 0.000000 0.000000
-0.239714002 -0.1555955002 -0.250012502 -0.000000 -0.000000 -0.247583802 -0.155172102 -0.241703802 -0.151037702

I = 8
WAVE-HEIGHT
0.000000 0.000000 0.000000 0.000000 0.000000
-0.200577002 -0.1208065002 -0.250012502 -0.102710902 -0.2170927002 -0.979178201 -0.216366902 -0.102237002

I = 9
WAVE-HEIGHT
0.000000 0.000000 0.000000 0.000000 0.000000
-0.200577002 -0.1208065002 -0.250012502 -0.154056702 -0.247007902 -0.149955902 -0.231111202 -0.140029202

U
U

K= 1 I = 9

W -VELOCITY
.0000000
.1974573-02
.0000000
.2589752-02
.3492988-02
.3153560-02
.2776921-02
.2250031-02
.2101176-02

K= 1 I = 10

W -VELOCITY
.0000000
.3009129-02
.4560243-02
.0000000
.4543842-02
.1110550-02
.2907990-02
.3610662-02
.2530645-02

K= 1 I = 11

W -VELOCITY
.0000000
.3277628-02
.3944439-02
.0000000
.3072628-02
.2588939-02
.2342776-02
.2360796-02
.3168049-02

K= 1 I = 12

W -VELOCITY
.0000000
.1065260-01
.3046363-02
.0000000
.3270246-02
.2509137-02
.7209129-03
.2564036-02
.2581190-02

K= 1 I = 13

W -VELOCITY
.0000000
.8151436-02
.2661427-02
.0000000
.2880202-02
.3254401-02
.2629987-02
.3296794-02
.2601037-02

K= 1 I = 14

W -VELOCITY
.0000000
.3155266-02
.1775953-02
.0000000
.4961285-02
.3032716-02
.2947769-02
.2601068-02
.2122584-02

K= 1 I = 15

W -VELOCITY
.0000000
.1105785-02
.3602575-02
.0000000
.4775560-02
.3377179-02
.2663010-02
.3091018-02
.2127827-02

K= 1 I = 16

W -VELOCITY
.0000000
.1262246-02
.2488969-02
.0000000
.2180497-02
.3309158-02
.2647301-02
.1655575-02
.1699278-02

K= 1 I = 17

W -VELOCITY
.1472839-01
.5953945-02
.3333424-02
.3397438-02
.3040734-02
.3280763-02
.3921161-02
.2890130-02
.3089816-02

K= 1 I = 18

W -VELOCITY
.4737235-02
.2558575-02
.2529672-03
.2329056-02
.1922037-02
.3780229-02
.2662869-02
.4022093-02
.3340506-02

K= 1 I = 19

W -VELOCITY
.4059941-02
.4446410-02
.2792152-02
.1631194-02
.2024414-02
.2435638-02
.2208436-02
.2463631-02
.2279371-02

K= 1 I = 20

W -VELOCITY
.0000000
.1631194-02
.2792152-02
.1631194-02
.2024414-02
.2435638-02
.2208436-02
.2463631-02
.2279371-02

.2520161*02 .2595073*02 .2601051*02

K= 1 I= 1C
TEMPERATURE

.2734221*02 .2712409*02 .2572978*02 .2655C99*02 .2668170*02
.2617515*02 .2625217*02 .2693283*02 .2718791*02 .2653142*02 .2709911*02 .2707349*02

K= 1 I= 11
TEMPERATURE

.2733733*02 .2737013*02 .2653142*02 .2709911*02 .2707349*02
.2613595*02 .2614964*02 .2616285*02 .2613145*02 .2613145*02

K= 1 I= 12
TEMPERATURE

.2841986*02 .2842696*02 .2823477*02 .2675550*02 .2818305*02 .2636088*02 .2722109*02
.2615415*02 .2613145*02 .2851672*02 .2634614*02 .2795894*02 .2636244*02 .2650430*02

K= 1 I= 13
TEMPERATURE

.2661139*02 .2879942*02 .2851672*02 .2634614*02 .2795894*02 .2636244*02 .2650430*02
.2639211*02 .2632941*02 .2841464*02 .2689815*02 .2647697*02 .26495362*02 .2668609*02

K= 1 I= 14
TEMPERATURE

.0000000 .2840187*02 .2841464*02 .2689815*02 .2647697*02 .26495362*02 .2668609*02
.2508745*02 .2655899*02 .2653741*02 .2653741*02 .2653741*02 .2653741*02 .2653741*02

K= 1 I= 15
TEMPERATURE

.2885116*02 .2815197*02 .2768264*02 .2703379*02 .2581671*02 .2641289*02 .2700215*02
.2733484*02 .2684851*02 .2693085*02 .2693085*02 .2693085*02 .2693085*02 .2693085*02

K= 1 I= 16
TEMPERATURE

.2954193*02 .2630793*02 .2760856*02 .2725777*02 .2532173*02 .2604870*02 .2803829*02
.2611205*02 .2638505*02 .2631118*02 .2631118*02 .2631118*02 .2631118*02 .2631118*02

K= 1 I= 17
TEMPERATURE

.2922907*02 .2892068*02 .2754472*02 .2664028*02 .2508039*02 .2402145*02 .2919292*02
.2570540*02 .2415605*02 .2415605*02 .2415605*02 .2415605*02 .2415605*02 .2415605*02

K= 1 I= 18
TEMPERATURE

.2934978*02 .2761890*02 .2707746*02 .2662792*02 .2732573*02 .2580046*02 .2771115*02
.2544942*02 .2544942*02 .2544942*02 .2544942*02 .2544942*02 .2544942*02 .2544942*02

K= 1 I= 19
TEMPERATURE

.2822298*02 .2770765*02 .2730077*02 .2641872*02 .2726056*02 .2842338*02 .2602752*02
.2568649*02 .30337807*02 .30335936*02 .30335936*02 .30335936*02 .30335936*02 .30335936*02

K= 1 I= 20
TEMPERATURE

.2652089*02 .2798062*02 .2689562*02 .2677064*02 .26335954*02 .2607883*02 .2690063*02
.2851732*02 .2851732*02 .2851732*02 .2851732*02 .2851732*02 .2851732*02 .2851732*02

HORIZONTALLY STRETCHED MODEL

6.2. NASUM III (Horizontal Stretched Model):

6.2.1. Problem Statement:

Given the Hutchinson Island, St Lucie, Florida as an example application site, compute the three dimensional velocity and temperature distribution for the following discharge and meteorological conditions.

Discharge volume from the	}	- 363,000 6 p.m.
Condensers of Power Plant		
Discharge Temperature		= 35.0°C
Ambient Temperature		= 25.0°C
Air Temperature		= 30.0°C
Current		= 2 cm/sec South

6.3. Calculation of Input Parameters:

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

6.3.1. Grid System

The remote sensing data and ground truth data was available for the Hutchinson Island site and it is used to determine the size of the domain. The domain selected was 2380 m x 2000 m. The domain has a variable depth and so a variable bottom topography is used. A horizontally stretched grid system as shown in Fig. () is used. This would give more resolution of the plume in the near field where the effects of the thermal discharge are predominant.

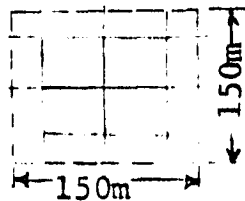
The grid system is also stretched in the vertical direction for ease of programming. In this way, same number of grid points can be used in shallow and deep regions of the basin.

6.3.2. Calculation of Discharge Velocity:

In the numerical model a 9 point discharge is chosen. The discharge velocity is calculated by balancing the mass as shown below.

For the numerical grid system, the mass into the domain = (Discharge Area x Velocity) = $A \times V$

The grid size chosen at the discharge is the minimum grid and equal to 50 m. So the discharge area is equal to $(150 \times 150) \text{ m}^2$ as shown below



$$\therefore A = (150 \times 150) \times 10^4 \text{ cm}^2$$

$$\begin{aligned} V \times A &= \text{Discharged volume (given)} \\ &= 363,000 \text{ G.P.M} \end{aligned}$$

$$V \times (150 \times 150 \times 10^4) = \frac{363,000 \times 0.0038 \times 10^6}{60} \frac{\text{cm}^3}{\text{sec}}$$

$$V = 0.102 \text{ cm/sec}$$

\therefore Velocity at discharge or inlet velocity for the model is 0.102 cm/sec.

6.3.3. Reference Quantities:

The reference eddy viscosity and diffusivity are determined using the 4/3 power law as below

$$A_{\text{ref}} = 0.0025 (L)^{4/3}$$

Where A_{ref} is the reference eddy viscosity and L is the maximum length of the domain in centimeters.

$$A_{ref} = 0.0025 (2380 \times 10^2)^{4/3}$$

$$\approx 40,000 \text{ cm}^2/\text{sec}$$

For turbulent prandtl number of 1 ($Pr_t = \frac{A_{ref}}{B_{ref}}$)

$$B_{ref} \text{ (Eddy diffusivity)} = 40,000 \frac{\text{cm}^2}{\text{sec}}$$

The vertical eddy viscosity and diffusivity chosen was $10 \frac{\text{cm}^2}{\text{sec}}$

6.3.4. Calculation of Input Data as It Appears In the Main Program

<u>Card No</u>	<u>Fortran Quantity</u>
1	IRUN

IRUN = 0 for the first run and equal to 1 for later runs.

<u>Card No</u>	<u>Fortran Quantity</u>
2	LN

LN is the number of cycles required. It is always advised to run the program for 10 or 15 cycles and check how the model is running.

<u>Card No</u>	<u>Fortran Quantity</u>
3	KSTORE

If KSTORE is equal to zero the model will store the results on the tape to be provided and if it is equal to 1 the model will not store results on the tape. For KSTORE equal to zero or 1 the model will print results at the end of the run.

<u>Card No</u>	<u>Fortran Quantity</u>
4	CI, CC, CH, CV, CP

The values of these quantities are equal to 1.0

<u>Card No</u>	<u>Fortran Quantity</u>
5	GR, FF, RR, HK

GR is gravity = 980.0 cm/sec

FF is coriolis term ≈ 0.0006

For small domain this terms is negligible and may be kept equal to 0.0 for all practical purposes

RR is density = 1.0

$$HK = \frac{h}{PC_p KV}$$

$h = 1200 \text{ BTU/day } ^\circ\text{F} - \text{ft} = 0.00678 \text{ Ca/sec, } ^\circ\text{C cm}$

$$HK = \frac{0.00678}{1.0 \times 1.0 \times 1.0} = 0.000678$$

<u>Card No</u>	<u>Fortran Quantity</u>
6	DX, DY, DZ

DX, DY, DZ are the minimum grid sizes chosen and they are

$$DX = 5000.0$$

$$DY = 5000.0$$

$$DZ = \frac{1}{KN-1} = \frac{1}{5-1} = \frac{1}{4} = 0.25$$

When KN is the number of grids in the vertical direction

<u>Card No</u>	<u>Fortran Quantity</u>
7	KH, KV, BH, BV

These are reference horizontal and vertical eddy viscosities and diffusivities. These are calculated in section 6.3.3.

$$KH = 40000.0$$

$$KV = 10.0$$

$$BH = 40000.0$$

$$BV = 10.0$$

<u>Card No</u>	<u>Fortran Quantity</u>
8	DELX, DELY, DEEX, DEEY, EEEX, EEEY

DELX and DELY are minimum grid sizes chosen = 5000 cm
 In order to determine DEEX, DEEY, EEEX, EEEY, another main
 program "CONST" has to be run twice. The input cards needed
 for the program "CONST" are

First time XB, A, DX, AN

Second time YB, B, DY, BN

Where XB = X boundary = 238000.0 cm

A = a constant = 38,000.0

DX = DELTA X = 5000.0 cm

AN = Number of grids in x direction = 20

Similarly YB = 200000.0 cm

B = 100000.0

DY = 5000.0 cm

BN = 20

The output of this main program will give the constants
 C1,D and C1,D

First C1,D are equal to EEEX and DEEX

Second C1,D are equal to EEEY and DEEY.

For the sample problem they are equal to

DEEX = 29333.03

DEEY = 47503.21

EEEX = 22947.29

EEEY = 20957.83

<u>Card No</u>	<u>Fortran Quantity</u>
9	TAUX, TAUY

For the sample problem the effect of wind is neglected and
 they are equal to 0.0,0.0. If the effects wind are to be
 considered, they can be easily calculated as explained in
 Appendix A.

Card No
10

Fortran Quantity
TAIR

TAIR, the air temp = 30.0°C

Card No
11

Fortran Quantity
DT

The value of DT used in the sample problem is 5 sec. In general it is always advised to start with a small value and increase it to the point when the model would not go unstable.

Note: The sample problem presented is a simplified version of the Hutchinson Island discharge problem.

6.3.5. Program Execution Procedure (NASUM III)

This section describes the procedure by which the User executes the NASUM III program.

a) Input Parameter:

The User must first follow the steps outlined in the sample problem and become quite familiar with all the input parameters.

b) First Run:

In order to obtain three-dimensional velocity and three-dimensional temperature, the main program TMAIN3 is executed. In TMAIN3 there are two tape units. One is a READ unit designated as Unit 7. During the first run, there is no need for unit 7, and unit 8 has to be provided to store results on a magnetic tape.

c) Run Continuation:

For extending the results, the run has to be continued. The magnetic tape which was "unit 8" in the first run will now be read "unit 7", for reading the previously stored results. Another magnetic tape is now 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 extended runs $IRUN = 1$.

6.3.6. Sample Input

<u>Card No</u>	<u>Symbol</u>	<u>Value & Format</u>
1	IRUN	♢ ♢ ♢ ♢ 0
2	LN	♢ ♢ 700
3	KSTORE	♢ ♢ ♢ ♢ 0
4	CI, CC, CP, CH, CV	1.0, 1.0, 1.0, 1.0, 1.0
5	GR, FF, RR, HK	980.0, 0.00006, 1.0, 0.000678
6	DX, DY, DZ	5000.0, 5000.0, 0.25
7	KH, KV, BH, BV	40000.0, 10.0, 40000.0, 10.0
8	DELX, DELY, DEEX, DEEY, EEEX, EEEY	5000.0, 5000.0, 29333.03, 47503.21, 22947.29, 20957.83
9	TAUX, TAUY	0.0, 0.0
10	TAIR	30.0
11	DT	5.0

6.3.7. Sample Output

Some of the output from the model sample run are listed as follows:

- a) Parameters
- b) Surface (K=1) horizontal velocities, u and v.
- c) Vertical velocity, Ω , at K=2
- d) Surface temperatures (J=1 at left-hand side)
(J=JN at right-hand side)
I=1 at top
I=IN at bottom

The surface isotherms obtained after 1 hour of simulation are shown in Fig. (44).

KE 1 IF 6
 U-VELOCITY
 .1728563*01
 .1890646*01
 .1731295*01
 .164384P*01
 .1791741*01
 .1752443*01
 .1752052*01
 .1728563*01
 .1890646*01
 .1731295*01
 .164384P*01
 .1791741*01
 .1752443*01
 .1752052*01

KE 1 IF 6
 U-VELOCITY
 .3735599*00
 .4386481*00
 .3743002*00
 .4167932*00
 .3769361*00
 .3957166*00
 .3892725*00
 .3613283*00
 .3735599*00
 .4386481*00
 .3743002*00
 .4167932*00
 .3769361*00
 .3957166*00
 .3892725*00
 .3613283*00

KE 1 IF 6
 U-VELOCITY
 .1799613*01
 .2117109*01
 .1605433*01
 .2010663*01
 .1827742*01
 .1968917*01
 .1877497*01
 .1839366*01
 .1799613*01
 .2117109*01
 .1605433*01
 .2010663*01
 .1827742*01
 .1968917*01
 .1877497*01
 .1839366*01

KE 1 IF 7
 U-VELOCITY
 .3837645*00
 .5005581*00
 .3857023*00
 .4541865*00
 .3928139*00
 .4170398*00
 .4100549*00
 .3957368*00
 .3837645*00
 .5005581*00
 .3857023*00
 .4541865*00
 .3928139*00
 .4170398*00
 .4100549*00
 .3957368*00

KE 1 IF 7
 U-VELOCITY
 .1651303*01
 .2415021*01
 .1860526*01
 .2191283*01
 .1894846*01
 .2015611*01
 .1974075*01
 .1939035*01
 .1651303*01
 .2415021*01
 .1860526*01
 .2191283*01
 .1894846*01
 .2015611*01
 .1974075*01
 .1939035*01

KE 1 IF 8
 U-VELOCITY
 .3907066*00
 .5782492*00
 .3932578*00
 .4908179*00
 .3927516*00
 .4359409*00
 .4274591*00
 .4033369*00
 .3907066*00
 .5782492*00
 .3932578*00
 .4908179*00
 .3927516*00
 .4359409*00
 .4274591*00
 .4033369*00

KE 1 IF 8
 U-VELOCITY
 .1884846*01
 .2790002*01
 .1897045*01
 .2368155*01
 .1942869*01
 .2123270*01
 .2063817*01
 .1958829*01
 .1884846*01
 .2790002*01
 .1897045*01
 .2368155*01
 .1942869*01
 .2123270*01
 .2063817*01
 .1958829*01

KE 1 IF 9
 U-VELOCITY
 .3951667*00
 .6222743*00
 .3980469*00
 .5130446*00
 .4068385*00
 .4470165*00
 .4390759*00
 .4124351*00
 .3951667*00
 .6222743*00
 .3980469*00
 .5130446*00
 .4068385*00
 .4470165*00
 .4390759*00
 .4124351*00

KE 1 IF 9
 U-VELOCITY
 .1906390*01
 .3002522*01
 .1920205*01
 .2475497*01
 .1972297*01
 .2156797*01
 .2113417*01
 .1988976*01
 .1906390*01
 .3002522*01
 .1920205*01
 .2475497*01
 .1972297*01
 .2156797*01
 .2113417*01
 .1988976*01

KE 1 IF 10
 U-VELOCITY
 .3980718*00
 .6058703*00
 .4007681*00
 .5073906*00
 .4107968*00
 .4467816*00
 .4376726*00
 .4148562*00
 .3980718*00
 .6058703*00
 .4007681*00
 .5073906*00
 .4107968*00
 .4467816*00
 .4376726*00
 .4148562*00

KE 1 IF 10
 U-VELOCITY
 .1920436*01
 .2923429*01
 .1933377*01
 .2446259*01
 .1981790*01
 .2155701*01
 .2111518*01
 .1999555*01
 .1920436*01
 .2923429*01
 .1933377*01
 .2446259*01
 .1981790*01
 .2155701*01
 .2111518*01
 .1999555*01

KE 1 IF 11
 U-VELOCITY
 .4203613*00
 .4022570*00
 .422570*00
 .4096276*00
 .4007183*00
 .4251276*00
 .4251276*00
 .4096276*00
 .4203613*00
 .4022570*00
 .422570*00
 .4096276*00
 .4007183*00
 .4251276*00
 .4251276*00
 .4096276*00

ORIGINAL PAGE IS
 OF POOR QUALITY

-.5986951-06
-.5303639-06

K= 2 I = 3

M -VELOCITY

-.2335555-05
-.7093769-06
-.6708136-06

-.6065294-06
-.5355279-06

-.2335585-05
-.9423040-06
-.6841850-06

-.5999640-06
-.5370959-06

-.5822766-06
-.9190950-06
-.6881060-06

-.5795549-06
-.5770955-05

-.5986260-06
-.8429713-06
-.6881060-06

-.5514443-06

-.6637482-06
-.7421081-06

-.5271942-06

-.7096751-06
-.6624772-06

-.5172344-06

-.7672427-06
-.6334571-06

-.5216362-06

K= 2 I = 4

M -VELOCITY

-.2521592-05
-.1354209-05
-.6693902-06

-.2521592-05
-.1457493-05
-.7042214-06

-.7143459-06
-.1358190-05
-.7136579-06

-.6528320-06
-.1087410-05
-.7136579-06

-.7089765-06
-.7684996-06

-.7891596-06
-.5669872-06

-.9341541-06
-.5315889-06

-.3146161-05
-.5977216-06

K= 2 I = 5

M -VELOCITY

-.2633294-05
-.2292829-05
-.5865590-06

-.2603294-05
-.2545871-05
-.6622075-06

-.7760259-06
-.2206031-05
-.6806713-06

-.6495495-06
-.1391721-05
-.6806713-06

-.7052560-06
-.5407673-06

-.8500963-06
-.1603215-06

-.1163251-05
-.2040340-06

-.1698952-05
-.4176738-06

K= 2 I = 6

M -VELOCITY

-.2620122-05
-.4122076-05
-.4610948-06

-.2620122-05
-.4902348-05
-.5905427-06

-.7929078-06
-.3889488-05
-.6210384-06

-.6174251-06
-.1675533-05
-.6210384-06

-.6820693-06
-.3564357-06

-.9239129-06
-.7958045-06

-.1528038-05
-.4072236-06

-.2650884-05
-.1231863-06

K= 2 I = 7

M -VELOCITY

-.2350354-05
-.7459896-05
-.5556851-07

-.2350354-05
-.9650002-05
-.2564375-06

-.5274738-06
-.6618491-05
-.3006552-06

-.3193416-06
-.9060569-06
-.3608552-06

-.3989619-06
-.3579620-05

-.7550837-06
-.3047501-05

-.1744114-05
-.1640463-05

-.3884589-05
-.5252009-06

K= 2 I = 8

M -VELOCITY

-.2106413-05
-.1594233-04
-.3010527-06

-.2106413-05
-.1641256-03
-.3159564-07

-.2813387-06
-.1544334-03
-.2542270-07

-.5852473-07
-.1392077-03
-.2542270-07

-.1536991-06
-.1252752-04

-.6161365-06
-.6503024-05

-.2026971-05
-.3044076-05

-.5695058-05
-.3146569-05

K= 2 I = 9

M -VELOCITY

-.2098593-05
-.1723106-04
-.2774375-06

-.2098593-05
-.1652601-03
-.6326812-07

-.2743979-06
-.1524090-03
-.1476744-08

-.4365710-07
-.1341178-03
-.1476744-08

-.1448524-06
-.1699834-04

-.6561055-06
-.8630913-05

-.2253665-05
-.3854697-05

-.6493865-05
-.14348507-05

K= 2 I = 10

M -VELOCITY

-.2074426-05
-.1474647-04
-.3854445-06

-.2074426-05
-.1608121-03
-.8116115-07

-.2633026-06
-.1483602-03
-.1797993-07

-.2693833-07
-.1318735-03
-.1797993-07

-.1144627-06
-.1688310-04

-.5743054-06
-.8440196-05

-.1963840-05
-.3778330-05

-.5573972-05
-.1391255-05

K= 2 I = 11

M -VELOCITY

-.2205004-05
-.4538576-05
-.1455406-06

-.2205004-05
-.3311065-05
-.6369181-07

-.4193062-06
-.5855864-06
-.1329028-06

-.1757205-06
-.5261433-05
-.1329028-06

-.2264984-06
-.6255967-05

-.5150626-06
-.4368697-05

-.1270631-05
-.2282364-05

-.2717766-05
-.6438079-06

TEMPERATURE AT 1 = 1

26.49 26.49 25.78 25.51 25.43 25.43 25.46 25.51 25.57 25.62 25.66 25.68 25.66 25.61 25.54 25.46 25.41 25.40 25.39 25.39
26.49 25.79 25.51 25.43 25.43 25.46 25.51 25.57 25.62 25.66 25.68 25.66 25.61 25.54 25.46 25.41 25.40 25.39 25.39
26.35 25.73 25.43 25.42 25.44 25.51 25.62 25.75 25.87 25.96 25.99 25.96 25.85 25.68 25.52 25.42 25.38 25.37 25.37
26.21 25.69 25.47 25.42 25.48 25.62 25.66 26.12 26.38 26.58 26.65 26.57 26.33 25.98 25.65 25.45 25.38 25.37 25.37
26.10 25.67 25.48 25.45 25.55 25.77 26.20 26.65 27.18 27.59 27.71 27.52 27.05 26.42 25.85 25.52 25.41 25.38 25.38
26.04 25.67 25.51 25.50 25.62 25.66 26.64 27.65 28.64 29.17 29.27 28.61 27.96 26.96 26.11 25.62 25.46 25.42 25.42
26.04 25.68 25.56 25.57 25.76 26.17 27.31 28.56 30.90 31.52 31.51 30.42 29.00 27.54 26.38 25.74 25.53 25.48 25.48
26.05 25.67 25.60 25.63 25.67 26.44 27.54 30.47 35.00 35.00 35.00 32.19 29.97 28.04 26.61 25.84 25.59 25.54 25.54
26.04 25.62 25.61 25.66 25.92 26.56 26.20 30.68 35.00 35.00 35.00 32.77 30.43 28.30 26.72 25.89 25.62 25.57 25.57
26.01 25.62 25.60 25.66 25.69 26.47 27.99 30.55 35.00 35.00 35.00 32.42 30.16 28.16 26.67 25.88 25.62 25.57 25.57
25.93 25.63 25.63 25.58 25.60 25.77 26.10 27.09 28.47 30.04 30.72 30.85 30.07 28.84 27.49 26.39 25.78 25.58 25.54
25.84 25.64 25.59 25.50 25.52 25.56 25.64 26.13 26.75 27.39 27.81 27.96 27.76 27.27 26.61 26.00 25.65 25.53 25.50
25.78 25.38 25.38 25.50 25.52 25.56 25.64 25.77 25.96 26.15 26.29 26.36 26.32 26.17 25.94 25.71 25.56 25.51 25.50
25.72 25.72 25.37 25.50 25.52 25.53 25.55 25.58 25.63 25.67 25.70 25.72 25.71 25.68 25.63 25.57 25.53 25.52 25.52
25.67 25.57 25.36 25.50 25.53 25.53 25.54 25.55 25.58 25.63 25.66 25.66 25.55 25.55 25.55 25.54 25.53 25.53 25.53
25.63 25.63 25.35 25.50 25.53 25.54 25.54 25.54 25.54 25.54 25.54 25.54 25.54 25.54 25.54 25.54 25.54 25.54 25.54
25.59 25.59 25.34 25.50 25.53 25.54 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55
25.55 25.55 25.33 25.49 25.53 25.54 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55 25.55
25.53 25.53 25.33 25.50 25.54 25.55 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56
25.53 25.53 25.33 25.50 25.54 25.55 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56 25.56

7.1 MAIN PROGRAM FOR NASUM II

```

1      C      FMAIN
2      C*****
3      C      THIS PROGRAM CALCULATES THE SURFACE ELEVATIONS, CIRCULATION AND
4      C      TEMPERATURE DISTRIBUTION FOR THE THREE-DIMENSIONAL FREE SURFACE
5      C      FAR-FIELD MODEL
6      C*****
7      PARAMETER IN=34,JN=11,KN=5
8      REAL KH,KV
9      DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),
10     CHT(IN,JN),HI(IN,JN),ETA(IN,JN),HTE(IN,JN),
11     CH(IN,JN,KN),G(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
12     CHX(IN,JN),HY(IN,JN),MAR(IN,JN),HTD(IN,JN),
13     CRO(IN,JN,KN),P(IN,JN,KN),HDUM(IN,JN)
14     C,T(IN,JN,KN),TN1(IN,JN,KN),WZ(IN,JN,KN)
15     C,II(IN),DTA(13),DHS(13),DTAUX(13),DTAUY(13)
16     READ 1,IRUN
17     READ 1,LN
18     READ 1,LN1
19     READ 1,M
20     READ 1,INLET
21     READ 1,IBAY
22     READ 1,IH1TE
23     1      FORMAT (I5)
24     READ 11,I1,I2,I3,I4,I5,I6
25     READ 11,J1,J2,J3,J4,J5,J6
26     11     FORMAT(6J5)
27     READ 2,V0,TPH,TPH1,A1,A2
28     READ 2,HTMIN,TH T
29     READ 2,TFLAT
30     READ 2, CI,CC,CF,CH,CV
31     READ 2, GR,FF,RP
32     READ 2, DX,DY,DZ
33     READ 2, KH,KV
34     READ 2,BH,BV
35     READ 2, TI
36     READ 2,(DTAUX(I),I=1,13)
37     READ 2,(DTAUY(I),I=1,13)
38     READ 2,(DTA(I),I=1,13)
39     READ 2,(DHS(I),I=1,13)
40     DO 100 I=1,13
41     DHS(I)=DHS(I)/100.
42     100    CONTINUE
43     READ 2,DT
44     2      FORMAT( )
45     IF(IRUN.GT.0) GO TO 4
46     CALL READ2(IN,JN,MAR,HI,HX,HY,DX,DY)
47     CALL INITIA(IN,JN,KN,ETA,HT,HI,HTD,HTE,II,U,V,W,RO,P,GR,RR,DZ,D,
48     CE,H,G,TI,T,TN1)
49     TTOT=0.0
50     TTOT1=0.0
51     TAUX=0.0
52     TAUY=0.0
53     GO TO 6
54     4      CONTINUE
55     CALL READ1(IN,JN,KN,U,V,W,HI,HT,HTD,HX,HY,MAR,ETA,P,RO,CI,
56     CCC,CH,CV,CP,DX,DY,DZ,DT,TAUX,TAUY,ITOT,H,G,HTE,T,TTOT1,WZ)

```

```

57      6      CONTINUE
58      DO 5 L=1,LN
59      IF (TTOT.GE.THT) GO TO 1000
60      TTOT=TTOT+DT
61      TT=TTOT
62      IF (M.EQ.1) GO TO 33
63      CALL TIDE (IN,JN,TT,L,DT,HTD,HI,HTE,INLET,I1,I2,J1,J3,J4,I5,TPH1
64      C,A1,A2)
65      GO TO 34
66      33      CALL VEL (IN,JN,KN,U,V,H,G,D,E,TT,VO,TPH,I1,I2,I5,J1,J3,J4,INLET)
67      34      CONTINUE
68      CALL DATA (TAUX,TAUY,TT,TA,HS,DTAUX,DTAUY,DTA,DHS,TFLAT)
69      ITTOT=TTOT
70      IF (L.GT.1.OR.ITTOT.GT.0) GO TO 50
71      CALL HEIGHT (I,J,K,IN,JN,KN,MAR,U,V,HT,HTD,DZ,DT,DX,DY,HDUM,M,I1,
72      CI2,I3,I4,I5,I6,J1,J2,J3,J4,J5,J6)
73      CALL UVVEL (IN,JN,KN,U,V,H,G,DX,DY,DZ,W,TAUX,TAUY,DT,HT,
74      CHX,HY,ETA,P,MAR,CT,CC,CP,CH,CV,KH,KV,GR,RR,FF,HTD,RO,T,I1,I2,I3,
75      CI4,I5,I6,J1,J2,J3,J4,J5,J6,M)
76      CALL WVEL (IN,JN,KN,H,G,W,HTD,DX,DY,DZ,MAR,M,I1,I2,J3,J4)
77      CALL PRES (IN,JN,KN,HTD,RO,GR,P,DZ)
78      CALL ETT (IN,JN,HTD,HI,MAR,ETA)
79      GO TO 80
80      50      CONTINUE
81      CALL HEILN (IN,JN,KN,MAR,H,G,HTD,HT,HTE,DZ,DT,DX,DY,HDUM,M,I1,I2,
82      CI3,I4,I5,I6,J1,J2,J3,J4,J5,J6)
83      CALL UVVELN (IN,JN,KN,U,V,H,G,D,E,DX,DY,DZ,W,TAUX,TAUY,DT,
84      CHT,HTD,HTE,HX,HY,ETA,P,MAR,KH,KV,GR,RR,FF,CP,CC,CI,CH,CV,RO,T,I1
85      C,I2,I3,I4,I5,I6,J1,J2,J3,J4,J5,J6,M)
86      CALL WVEL (IN,JN,KN,D,E,W,HTE,DX,DY,DZ,MAR,M,I1,I2,J3,J4)
87      CALL PRES (IN,JN,KN,HTE,RO,GR,P,DZ)
88      CALL ETT (IN,JN,HTE,HI,MAR,ETA)
89      CALL OLDUV (IN,JN,KN,U,V,H,G,D,E)
90      CALL OLDHT (IN,JN,HTE,HTD,HT)
91      80      CONTINUE
92      5      CONTINUE
93      1000     CONTINUE
94      DO 15 LL=1,LN1
95      TT=TTOT1+TTOT
96      IF (TT.LT.THT) GO TO 3000
97      IF (IBAY.EQ.1) GO TO 5002
98      DO 211 I=1,IN
99      DO 211 J=1,JN
100     IF (ETA(I,J).LT.0.0) GO TO 65
101     GO TO 211
102     65      CONTINUE
103     IHI=HI(I,J)
104     IHTMIN=HTMIN
105     IF (IHI.EQ.IHTMIN) GO TO 4000
106     GO TO 4001
107     4000     CONTINUE
108     HTM=HTMIN+30.48
109     HI(I,J)=HTM
110     HT(I,J)=HI(I,J)+ETA(I,J)
111     HTD(I,J)=HT(I,J)
112     HTE(I,J)=HT(I,J)
113     4001     CONTINUE

```

```

114      A=-30.48
115      HTM=HTMIN+30.48
116      IF(ETA(I,J).LE.A.AND.HI(I,J).LE.HTM) GO TO 311
117      GO TO 211
118
119      311  CONTINUE
119      HI(I,J)=HTM+30.48
120      HT(I,J)=ETA(I,J)+HI(I,J)
121      HTD(I,J)=HT(I,J)
122      HTE(I,J)=HT(I,J)
123      211  CONTINUE
124      5002 CONTINUE
125      IF(M.EQ.1) GO TO 330
126      CALL TIDE(IN,JN,TT,LL,DT,HTD,HI,HTE,INLET,I1,I2,J1,J3,J4,I5,TPH1
127      C,A1,A2)
128      GO TO 340
129      330  CALL VEL(IN,JN,KN,U,V,H,G,D,E,TT,VO,TPH,I1,I2,I5,J1,J3,J4,INLET)
130      340  CONTINUE
131      CALL DATA(TAUX,TAUY,TT,TA,HS,DTAUX,DTAUY,DTA,DHS,TFLAT)
132      ITT=TT
133      ITHT=THT
134      IF(ITT.EQ.ITHT) GO TO 2000
135      TTOT1=TTOT1+DT
136      500  CONTINUE
137      CALL HEILN(IN,JN,KN,MAR,H,G,HTD,HT,HTE,DZ,DT,DX,DY,HJUM,M,I1,
138      CI2,I3,I4,I5,I6,J1,J2,J3,J4,J5,J6)
139      GO TO 2001
140      2000 CONTINUE
141      CALL TIDAL(IN,JN,KN,ETA,HI,HT,HTD,HTE,II,T,IHITE)
142      TTOT1=TTOT1+DT
143      IF(IHITE.EQ.1) GO TO 500
144      2001 CONTINUE
145      CALL UVVELN(IN,JN,KN,U,V,H,G,D,E,DX,DY,DZ,W,TAUX,TAUY,DT,
146      CHT,HTD,HTE,HX,HY,ETA,P,MAR,KH,KV,GR,RR,FF,CP,CC,CI,CH,CV,RO,T,I1,
147      CI2,I3,I4,I5,I6,J1,J2,J3,J4,J5,J6,H)
148      CALL WVEL(IN,JN,KN,D,E,W,HTE,DX,DY,DZ,MAR,M,I1,I2,J3,J4)
149      CALL PRES(IN,JN,KN,HTE,RO,GR,P,DZ)
150      CALL ETT(IN,JN,HTE,HI,MAR,ETA)
151      CALL OLDUV(IN,JN,KN,U,V,H,G,D,E)
152      CALL OLDHT(IN,JN,HTE,HTD,HT)
153      CALL TEMP(IN,JN,KN,T,H,G,W,DX,DY,DZ,DT,HT,BH,BV,MAR,TN1,HTD,QRAD,
154      CRR,HS,TA,RO)
155      CALL OLD T(IN,JN,KN,T,TN1)
156      15  CONTINUE
157      3000 CONTINUE
158      CALL PRPARA(CI,CH,CV,CP,CC,DX,DY,DZ,DT,TAUX,TAUY,TTOT,GR,FF,RR,
159      CKH,KV,BH,BV,QRAD,TI,TTOT1,TA,HS)
160      CALL PRETA(I,J,IN,JN,ETA)
161      CALL PRUV(I,J,K,IN,JN,KN,H,G)
162      CALL WW(IN,JN,KN,HT,HTD,ETA,H,G,W,WZ,MAR,DX,DY,DZ,DT,HX,HY)
163      CALL PRW(IN,JN,KN,WZ)
164      CALL PRTEMP(I,J,K,IN,JN,KN,T)
165      CALL STORE(IN,JN,KN,U,V,W,HI,HT,HTD,HX,HY,MAR,ETA,P,RO,CI,
166      CCC,CH,CV,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,H,G,HTE,T,TTOT1,WZ)
167      STOP
168      END

```

ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE PROGRAMS FOR NASUM II

~~SECRET~~ SECRET ~~SECRET~~

7.1.1 READ 2

This subroutine is used by specifying $IRUN=0$. First, the two-dimensional MAR matrix, $MAR(I,J)$ is read in from data cards (a sample problem will illustrate this in Section 6.1).

The MAR numbering system is used for distinguishing between spatial differencing of the terms of the system of governing equations in the interior of the domain of solution, on the boundaries, and outside the domain. Points outside the domain are assigned a value $MAR=0$, and calculations are not performed. The MAR matrix, as will be shown in a sample problem, is constructed by the user by first establishing a grid system which closely follows the geometry of the application site. Then the MAR numbering system is specified as follows:

$MAR(I,J) = 0$ points outside domain

$MAR(I,J) = 1$ upper horizontal boundary

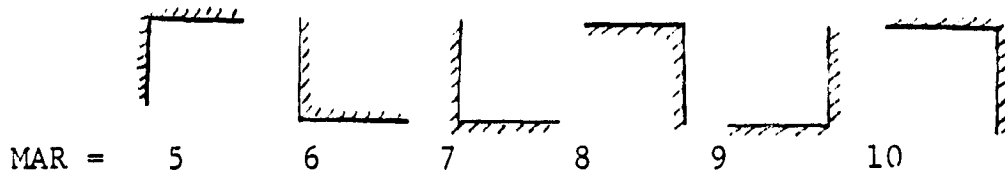
$MAR(I,J) = 2$ lower horizontal boundary

$MAR(I,J) = 3$ left vertical boundary

$MAR(I,J) = 4$ right vertical boundary

$MAR(I,J) = 5$ through $MAR(I,J) = 10$ are boundary corners and are specified below:

$MAR(I,J) = 11$ interior of domain.



Next, this subroutine reads in from data cards the two-dimensional matrix, $HI(I,J)$, which specifies the depth (in feet) below the mean water level at each grid point. Then the depths are converted into centimeters for calculation of the bottom gradients, $HX(I,J)$ and $HY(I,J)$ in the x and y directions, respectively. Note, that calculation of the bottom gradients is performed by using central differencing in the interior and the point single-sided differencing on the boundary. Again, for $MAR=6$ and $MAR=8$ central differencing is used, since these particular corners may be treated as being interior points.

```

1 C*****
2 C THIS SUBROUTINE READS IN DATA FOR THE APPLICATION SITE DOMAIN
3 C AND ACTUAL BOTTCM TOPOGRAPHY
4 C*****
5 SUBROUTINE READ2(IN,JN,MAR,HI,HX,HY,DX,DY)
6 DIMENSION MAR(IN,JN),HI(IN,JN),HX(IN,JN),HY(IN,JN)
7 DO 10 I=1,IN
8     10 READ 1,(MAR(I,J),J=1,JN)
9     DO 99 I=1,IN
10    99 READ 1,(HI(I,J),J=1,JN)
11    1 FORMAT( )
12    DO 15 I=1,IN
13    DO 15 J=1,JN
14    HI(I,J)=30.48*HI(I,J)
15    15 CONTINUE
16    DO 50 I=1,IN
17    DO 50 J=1,JN
18    IF(MAR(I,J).EQ.0) GO TO 50
19    IF(MAR(I,J).EQ.1) GO TO 31
20    IF(MAR(I,J).EQ.2) GO TO 32
21    IF(MAR(I,J).EQ.3) GO TO 33
22    IF(MAR(I,J).EQ.4) GO TO 34
23    IF(MAR(I,J).EQ.5) GO TO 35
24    IF(MAR(I,J).EQ.6) GO TO 36
25    IF(MAR(I,J).EQ.7) GO TO 37
26    IF(MAR(I,J).EQ.8) GO TO 38
27    IF(MAR(I,J).EQ.9) GO TO 39
28    IF(MAR(I,J).EQ.10) GO TO 40
29    HX(I,J)=(HI(I+1,J)-HI(I-1,J))/(2.*DX)
30    HY(I,J)=(HI(I,J+1)-HI(I,J-1))/(2.*DY)
31    GO TO 50
32    31 CONTINUE
33    HX(I,J)=(HI(I+1,J)-HI(I-1,J))/(2.*DX)
34    HY(I,J)=(3*HI(I,J)+HI(I,J-2)-4*HI(I,J-1))/(2*DY)
35    GO TO 50
36    32 CONTINUE
37    HX(I,J)=(HI(I+1,J)-HI(I-1,J))/(2.*DX)
38    HY(I,J)=(4*HI(I,J+1)-3*HI(I,J)-HI(I,J+2))/(2*DY)
39    GO TO 50
40    33 CONTINUE
41    HX(I,J)=(4*HI(I+1,J)-3*HI(I,J)-HI(I+2,J))/(2*DX)
42    HY(I,J)=(HI(I,J+1)-HI(I,J-1))/(2.*DY)
43    GO TO 50
44    34 CONTINUE
45    HX(I,J)=(3*HI(I,J)+HI(I-2,J)-4*HI(I-1,J))/(2*DX)
46    HY(I,J)=(HI(I,J+1)-HI(I,J-1))/(2.*DY)
47    GO TO 50
48    35 CONTINUE
49    HX(I,J)=(4*HI(I+1,J)-3*HI(I,J)-HI(I+2,J))/(2*DX)
50    HY(I,J)=(3*HI(I,J)+HI(I,J-2)-4*HI(I,J-1))/(2*DY)
51    GO TO 50
52    36 CONTINUE
53    HX(I,J)=(HI(I+1,J)-HI(I-1,J))/(2.*DX)
54    HY(I,J)=(HI(I,J+1)-HI(I,J-1))/(2.*DY)
55    GO TO 50
56    37 CONTINUE

```

```

57      HX(I,J)=(4*HI(I+1,J)-3*HI(I,J)-HI(I+2,J))/(2*DX)
58      HY(I,J)=(4*HI(I,J+1)-3*HI(I,J)-HI(I,J+2))/(2*DY)
59      GO TO 50
60      38  CONTINUE
61      HX(I,J)=(HI(I+1,J)-HI(I-1,J))/(2*DX)
62      HY(I,J)=(HI(I,J+1)-HI(I,J-1))/(2*DY)
63      GO TO 50
64      39  CONTINUE
65      HX(I,J)=(3*HI(I,J)+HI(I-2,J)-4*HI(I-1,J))/(2*DX)
66      HY(I,J)=(4*HI(I,J+1)-3*HI(I,J)-HI(I,J+2))/(2*DY)
67      GO TO 50
68      40  CONTINUE
69      HX(I,J)=(3*HI(I,J)+HI(I-2,J)-4*HI(I-1,J))/(2*DX)
70      HY(I,J)=(3*HI(I,J)+HI(I,J-2)-4*HI(I,J-1))/(2*DY)
71      50  CONTINUE
72      RETURN
73      END

```

ORIGINAL PAGE IS
OF LOW QUALITY

INTENTIONAL COPY

7.1.2 INITIA

This subroutine is used by specifying IRUN =0. The initial values of η , H, u, v, Ω , T, P and ρ are specified as was outlined in section 4.1.

$$\eta = 0$$

$$u = 0$$

$$v = 0$$

$$\Omega = 0$$

$$*T = T_{unif} = \text{const.}$$

$$P = P_0 + \rho g H \sigma$$

$$\rho = \rho_{unif}$$

In terms of the symbols used in the program, we have:

$$ETA(I,J) = 0$$

$$HT(I,J) = HI(I,J)$$

$$HTD(I,J) = HI(I,J)$$

$$HTE(I,J) = HI(I,J)$$

$$U(I,J,K) = 0$$

$$H(I,J,K) = 0$$

$$D(I,J,K) = 0$$

$$V(I,J,K) = 0$$

$$G(I,J,K) = 0$$

$$E(I,J,K) = 0$$

$$W(I,J,K) = 0$$

$$T(I,J,K) = TI$$

$$TN1(I,J,K) = TI$$

$$RO(I,J,K) = RR$$

$$P(I,J,K) = RR * GR * HT(I,J) * (K-1) * DZ$$

*NOTE: The energy equation is only coupled to the system of governing equations after the IR data base is inputted at 10:10pm for 4/15/75 into the program, and therefore an initial value of T is not actually required. However, if no IR data is available an isothermal bay may be assumed initially and the coupling to the energy equation initially.

```

1      C      THIS PROGRAM INITIALIZES VARIABLES FOR CONSTANT DENSITY MODEL
2      C
3      C*****
4      C      THIS SUBROUTINE INITIALIZES THE VARIABLES FOR THE VARIABLE DENSITY "
5      C*****
6      SUBROUTINE INITIA(IN,JN,KN,ETA,HT,HI,HTD,HTE,II,U,V,W,RO,P,GR,RR,
7      CDZ,D,E,H,G,II,T,TN1)
8      DIMENSION ETA(IN,JN),HT(IN,JN),HI(IN,JN),HTD(IN,JN),HTE(IN,JN),
9      CII(IN),U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),W(IN,JN,KN)
10     C,RO(IN,JN,KN),H(IN,JN,KN),G(IN,JN,KN),P(IN,JN,KN),T(IN,JN,KN),
11     CTN1(IN,JN,KN)
12     DO 20 I=1,IN
13     DO 20 J=1,JN
14     ETA(I,J)=0.0
15     HT(I,J)=HI(I,J)+ETA(I,J)
16     HTD(I,J)=HT(I,J)
17     HTE(I,J)=HT(I,J)
18
19     20  CONTINUE
20     DO 8 I=1,IN
21     DO 8 J=1,JN
22     DO 8 K=1,KN
23     U(I,J,K)=0.
24     H(I,J,K)=0.
25     D(I,J,K)=0.
26     V(I,J,K)=0.
27     G(I,J,K)=0.
28     E(I,J,K)=0.
29     W(I,J,K)=0.
30     T(I,J,K)=TI
31     TN1(I,J,K)=TI
32     RO(I,J,K)=1.029431-.000020*T(I,J,K)-.0000048*(T(I,J,K)**2)
33
34     8  CONTINUE
35     DO 25 I=1,IN
36     DO 25 J=1,JN
37     DO 21 K=1,KN
38     RR=RO(I,J,K)
39     P(I,J,K)=GR*HT(I,J)+RR*(K-1)*DZ
40
41     21  CONTINUE
42     25  CONTINUE
43     RETURN
44     END

```

7.1.3 READ 1

This subroutine is used by specifying IRUN=1. The results of a previous run are read in from a magnetic tape for the purpose of running the program over a long period of time in segments. The system variables η , H, u, v, Ω , T, P and ρ are read in as well as physical and numerical parameters, as follows:

HI(I,J) = depths at each grid points (in cm)

HX(I,J) = bottom gradient in x-direction

HY(I,J) = bottom gradient in y-direction

MAR(I,J) = domain numbering system

CI, CC, CH, CV, CP = constants always = 1

DX = horizontal grid size in x-direction (in cm)

DY = horizontal grid size in y-direction (in cm)

DZ = vertical grid size = $\Delta Z/HI(I,J)$

*DT = time step (in seconds)

TAUX = wind stress component in x-direction (dynes/cm²)

TAUY = wind stress component in y-direction (dynes/cm²)

TTOT = total run time without energy equation (in seconds)

TTOT1 = total run time with energy equation (in seconds)

* NOTE: The determination of Δt is presented in the sample problem in section 6.1.

```

1      C
2      C*****
3      C   THIS SUBROUTINE READS IN (FROM MAGNETIC TAPE) VALUES FOR THE VARIABLE.
4      C   AND PHYSICAL AND NUMERICAL PARAMETERS FROM A PREVIOUS COMPUTER RUN
5      C   FOR THE VARIABLE DENSITY MODEL
6      C*****
7      SUBROUTINE READ I(IN,JN,KN,U,V,W,HI,HT,HTD,HX,HY,MAR,ETA,P,RO,CI,
8      CCC,CH,CV,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,H,G,HTE,T,TTOT1,WZ)
9      DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),P(IN,JN,KN),
10     CHI(IN,JN),HT(IN,JN),HTD(IN,JN),HX(IN,JN),HY(IN,JN),MAR(IN,JN),
11     CETA(IN,JN),RO(IN,JN,KN),H(IN,JN,KN),G(IN,JN,KN),HTE(IN,JN)
12     C,T(IN,JN,KN),WZ(IN,JN,KN)
13     READ (7) (((U(I,J,K),K=1,KN),J=1,JN),I=1,IN),
14     C(((V(I,J,K),K=1,KN),J=1,JN),I=1,IN),
15     C(((W(I,J,K),K=1,KN),J=1,JN),I=1,IN),
16     C(((H(I,J,K),K=1,KN),J=1,JN),I=1,IN),
17     C(((G(I,J,K),K=1,KN),J=1,JN),I=1,IN),
18     C(((P(I,J,K),K=1,KN),J=1,JN),I=1,IN),
19     C(((RO(I,J,K),K=1,KN),J=1,JN),I=1,IN),
20     C((HTD(I,J),J=1,JN),I=1,IN),
21     C((HTE(I,J),J=1,JN),I=1,IN),
22     C((HI(I,J),J=1,JN),I=1,IN),
23     C((HX(I,J),J=1,JN),I=1,IN),
24     C((HY(I,J),J=1,JN),I=1,IN),
25     C((MAR(I,J),J=1,JN),I=1,IN),
26     C((HT(I,J),J=1,JN),I=1,IN),
27     C((ETA(I,J),J=1,JN),I=1,IN),
28     C(((T(I,J,K),K=1,KN),J=1,JN),I=1,IN),
29     C(((WZ(I,J,K),K=1,KN),J=1,JN),I=1,IN),
30     CCI,CC,CH,CV,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,TTOT1
31     RETURN
32     END

```

7.1.4 TIDE (M=2)

This subroutine specifies the tide height at the ocean-bay opening as a function of time (i.e. TTOT or TTOT + TTOT1) in the form:

$$\eta_0(t) = A_1 + A_2 \cos \omega(t+t\phi)$$

or in terms of the program symbols:

$$ETA = A1 + A2 * \cos ((TT + TPH1) * (6.23/12.15 * 3600))$$

where A1 and A2 are read in from data cards, TT = TTOT or =TTOT + TTOT1 and TPH1 is computed as will be illustrated in the sample problem.

This is the actual tidal condition case for tidal flux at the ocean-bay opening. Note, this subroutine is not used when the current velocity is specified at this open boundary (i.e. M=1 for specifying $V_0(t)$). It is further pointed out that the user must start the computer run at $t=0$ (TTOT=0) with a flat surface). $ETA(I,J)=0$, and $ETA=0$ at the open boundary. Otherwise, the large step in surface height at the open boundary will result in numerical instability, since the governing equations do not adjust sufficiently fast to yield a compatible and realistic situation between surface heights and currents. Since, the domain of solution is assumed to be "still" (i.e. $u=v=\Omega=0$) initially for the case of not having an adequate initial data base for specifying currents, the above noted specification of $ETA(I,J)$ at $t=0$ must be followed to insure numerical stability.


```

1 C*****
2 C   THIS SURROUTINE CALCULATES THE SURFACE ELEVATION AT THE OCEAN-BAY
3 C   OPENING AT EVERY TIME STEP
4 C*****
5   SUBROUTINE TIDE(IN,JN,TTOT,L,DT,HTD,HI,HTE,INLET,I1,I2,J1,J3,J4
6   C,IS,TPH1,A1,A2)
7   DIMENSION HTD(IN,JN),HI(IN,JN),HTE(IN,JN)
8   ETA=A1+A2*COS((TTOT*TPH1)+(6.28/(12.15*3600.0)))
9   IF(INLET.GT.1) GO TO 2000
10  DO 40 I=I1,I2
11     ITTOT=TTOT
12     IF(L.GT.1.OR.ITTOT.GT.0) GO TO 5
13     HTD(I,J1)=ETA+HI(I,J1)
14     GO TO 40
15     5   HTE(I,J1)=ETA+HI(I,J1)
16     40  CONTINUE
17     GO TO 1000
18     2000 CONTINUE
19     DO 50 J=J3,J4
20     ITTOT=TTOT
21     IF(L.GT.1.OR.ITTOT.GT.0) GO TO 15
22     HTD(IS,J)=ETA+HI(IS,J)
23     GO TO 50
24     15  HTE(IS,J)=ETA+HI(IS,J)
25     50  CONTINUE
26     1000 CONTINUE
27     RETURN
28     END

```

7.1.5 VEL(M=1)

This subroutine specifies the current velocity at the ocean-bay opening as a function of time (ie TTOT or TTOT + TTOT1) in the form:

$$V_o(t) = V_o \cos \omega (t + t\phi) \quad \text{or:}$$

$$U_o(t) = U_o \cos \omega (t + t\phi) \quad \text{depending on which axis the open boundary is aligned.}$$

Then, in terms of the program symbols:

$$V_o = V_o * \cos ((TT + TPH) * (6.24) / (12.15 * 3600))$$

Where V_o is computed as an estimate and TPH is computed as will be illustrated in the sample problem. $TT = TTOT$ or $TT = TTOT + TTOT1$. Similarly, $V_o(t) = 0$ at $t = 0$ is recommended along with an initially 'flat' surface to insure compatibility between the surface heights and the velocity field.

```

1 C*****
2 C   THIS SUBROUTINE CALCULATES THE CURRENT VELOCITY AT THE OCEAN-BAY
3 C   OPENING AT EVERY TIME STEP
4 C*****
5   SUBROUTINE VEL(IN,JN,KN,U,V,H,G,D,E,TT,VO,TPH,I1,I2,I5,J1,J3,J4,
6   CINLET)
7   DIMENSION U(IN,JN,KN),V(IN,JN,KN),H(IN,JN,KN),G(IN,JN,KN)
8   C,D(IN,JN,KN),E(IN,JN,KN)
9   KN1=KN-1
10  DO 1000 K=1,KN1
11  IF(INLET.GT.1) GO TO 2000
12  DO 100 I=I1,I2
13  V(I,J1,K)=VO*COS((TT+TPH)*(6.28/(12.15*3600.0)))
14  G(I,J1,K)=V(I,J1,K)
15  E(I,J1,K)=G(I,J1,K)
16      100  CONTINUE
17      GO TO 1000
18      2000 CONTINUE
19      DO 200 J=J3,J4
20      U(I5,J,K)=VO*COS((TT+TPH)*(6.28/(12.15*3600.0)))
21      H(I5,J,K)=U(I5,J,K)
22      D(I5,J,K)=H(I5,J,K)
23      200  CONTINUE
24      1000 CONTINUE
25      RETURN
26      END

```

7.1.6 TIDAL

This subroutine specifies $T(I,J,K)$ as an initial temperature distribution which is constructed by the user from an IR-Data Base. This will be illustrated in the sample problem. Specification of an initial surface as constructed from an adequate tide data base may be inputted into the model but is left optional for the user.

```

1 C*****
2 C   THIS SUBROUTINE READS IN THE INITIAL VALUES FOR THE TEMPERATURE
3 C   DISTRIBUTION AND THE INITIAL VALUES FOR THE SURFACE
4 C   ELEVATIONS(OPTIONAL)
5 C*****
6   SUBROUTINE TIDAL(IN,JN,KN,ETA,HI,HT,HTD,HTE,II,T,IHTE)
7   DIMENSION II(IN),ETA(IN,JN),HI(IN,JN),HT(IN,JN),HTD(IN,JN),
8   CHTE(IN,JN),T(IN,JN,KN)
9   DO 100 I=1,IN
10  READ 5,(T(I,J,1),J=1,JN)
11      5   FORMAT( )
12  100   CONTINUE
13      DO 101 I=1,IN
14      DO 101 J=1,JN
15      DO 102 K=2,KN
16      T(I,J,K)=T(I,J,1)
17  102   CONTINUE
18  101   CONTINUE
19      IF(IHTE.EQ.1) GO TO 25
20      DO 10 I=1,IN
21      READ 1,II(I),(ETA(I,J),J=1,JN)
22      1   FORMAT(I3,11F7.2)
23  10    CONTINUE
24      DO 20 I=1,IN
25      DO 20 J=1,JN
26      HT(I,J)=ETA(I,J)+HI(I,J)
27      HTD(I,J)=HT(I,J)
28      HTE(I,J)=HT(I,J)
29  20    CONTINUE
30  25    CONTINUE
31      RETURN
32      END

```

7.1.7 DATA

This subroutine specifies the wind stresses, equilibrium temperature and the surface heat transfer coefficient every hour. However, these physical parameters must be calculated by the user as will be shown in the sample problem. Note, that this subroutine has been programmed for a maximum computer run of 12 hours. However, this subroutine can be used for less than 12 hours, or it may be easily modified by the user for computer runs in excess of 12 hours, by merely reading in values of these physical parameters for the desired increase in number of hours,

```

1  C*****
2  C   THIS SUBROUTINE READS IN HOURLY VALUES OF SURFACE WIND SHEAR STRESS,
3  C   EQUILIBRIUM TEMPERATURE, AND SURFACE HEAT TRANSFER COEFFICIENT
4  C*****
5  SUBROUTINE DATA (TAUX,TAUY,TTOT,TA,HS,DTAUX,DTAUY,DTA,DHS,TFLAT)
6  DIMENSION DTAUX(13),DTAUY(13),DTA(13),DHS(13)
7  TTOT=TTOT-TFLAT
8  IF(TTOT.LT.0.0) GO TO 50
9  IF(TTOT.GE.0.0.AND.TTOT.LT.3600.0) GO TO 1
10 IF(TTOT.LT.7200.AND.TTOT.GE.3600.0) GO TO 2
11 IF(TTOT.LT.10800.AND.TTOT.GE.7200.0) GO TO 3
12 IF(TTOT.LT.14400.AND.TTOT.GE.10800.0) GO TO 4
13 IF(TTOT.LT.18000.AND.TTOT.GE.14400.0) GO TO 5
14 IF(TTOT.LT.21600.AND.TTOT.GE.18000.0) GO TO 6
15 IF(TTOT.LT.25200.AND.TTOT.GE.21600.0) GO TO 7
16 IF(TTOT.LT.28800.AND.TTOT.GE.25200.0) GO TO 8
17 IF(TTOT.LT.32400.AND.TTOT.GE.28800.0) GO TO 9
18 IF(TTOT.LT.36000.AND.TTOT.GE.32400.0) GO TO 10
19 IF(TTOT.LT.39600.AND.TTOT.GE.36000.0) GO TO 11
20 IF(TTOT.LT.43200.AND.TTOT.GE.39600.0) GO TO 12
21 IF(TTOT.LT.46800.AND.TTOT.GE.43200.0) GO TO 13
22 1 CONTINUE
23 TAUX=DTAUX(1)
24 TAUY=DTAUY(1)
25 TA=DTA(1)
26 HS=DHS(1)
27 GO TO 50
28 2 CONTINUE
29 TAUX=DTAUX(2)
30 TAUY=DTAUY(2)
31 TA=DTA(2)
32 HS=DHS(2)
33 GO TO 50
34 3 CONTINUE
35 TAUX=DTAUX(3)
36 TAUY=DTAUY(3)
37 TA=DTA(3)
38 HS=DHS(3)
39 GO TO 50
40 4 CONTINUE
41 TAUX=DTAUX(4)
42 TAUY=DTAUY(4)
43 TA=DTA(4)
44 HS=DHS(4)
45 GO TO 50
46 5 CONTINUE
47 TAUX=DTAUX(5)
48 TAUY=DTAUY(5)
49 TA=DTA(5)
50 HS=DHS(5)
51 GO TO 50
52 6 CONTINUE
53 TAUX=DTAUX(6)
54 TAUY=DTAUY(6)
55 TA=DTA(6)
56 HS=DHS(6)

```

```
57          GO TO 50
58          7      CONTINUE
59             TAUX=DTAUX(7)
60             TAUY=DTAUY(7)
61             TA=DTA(7)
62             HS=DHS(7)
63          GO TO 50
64          8      CONTINUE
65             TAUX=DTAUX(8)
66             TAUY=DTAUY(8)
67             TA=DTA(8)
68             HS=DHS(8)
69          GO TO 50
70          9      CONTINUE
71             TAUX=DTAUX(9)
72             TAUY=DTAUY(9)
73             TA=DTA(9)
74             HS=DHS(9)
75          GO TO 50
76          10     CONTINUE
77             TAUX=DTAUX(10)
78             TAUY=DTAUY(10)
79             TA=DTA(10)
80             HS=DHS(10)
81          GO TO 50
82          11     CONTINUE
83             TAUX=DTAUX(11)
84             TAUY=DTAUY(11)
85             TA=DTA(11)
86             HS=DHS(11)
87          GO TO 50
88          12     CONTINUE
89             TAUX=DTAUX(12)
90             TAUY=DTAUY(12)
91             TA=DTA(12)
92             HS=DHS(12)
93          GO TO 50
94          13     CONTINUE
95             TAUX=DTAUX(13)
96             TAUY=DTAUY(13)
97             TA=DTA(13)
98             HS=DHS(13)
99          50     CONTINUE
100         TTOT=TTOT+TFLAT
101         RETURN
102         END
```

PRECEDING PAGE BLANK NOT FILMED

7.1.8 HEIGHT

This subroutine calculates $H(x,y)$ the depth contour with respect to the free surface, at $t = \Delta t$ ($=HTD(I,J)$) by forward differencing the surface height equation in time with respect to the initial depth contour matrix $H(x,y)$ at $t=0$ ($=HT(I,J)$). Note, this subroutine is used only for the first time cycle. The integration in this subroutine is performed by applying Simpson's Rule.* The general inlet and outlet conditions are specified by reading in from data cards parameters which impose the location of the inlet, either on the upper horizontal boundary or on the left vertical boundary of the grid system. However, this subroutine can be easily modified by the user for having an inlet on the lower horizontal boundary or on the right vertical boundary. The only change required is respecifying $MAR(I,J)$ corresponding to the inlet location and reading in from data cards the values of I and J which properly locate the inlet.

The derivatives in the integral are obtained by central differencing in space for interior points, including $MAR=6$ and $MAR=8$. Three point single sided differencing is performed on the boundaries. These different schemes are given in Volume I.

* For $KN = 5$:
$$\int_{x_1}^{x_5} F(x) dx \approx \Delta x \left[\frac{1}{3}F(x_1) + \frac{4}{3}F(x_2) + \frac{2}{3}F(x_3) + \frac{4}{3}F(x_4) + \frac{1}{3}F(x_5) \right]$$

```

1      C
2      C*****
3      C    THIS SUBROUTINE CALCULATES THE TOTAL DEPTH AT EACH X-Y LOCATION
4      C    IN THE DOMAIN FOR THE FIRST TIME STEP USING A FORWARD DIFFERENCING
5      C    SCHEME IN TIME
6      C*****
7      SUBROUTINE HEIGHT(I,J,K,IN,JN,KN,MAR,U,V,HT,HTD,DZ,DT,DX,DY,HDUM,
8      CM,I1,I2,I3,I4,I5,I6,J1,J2,J3,J4,J5,J6)
9      DIMENSION MAR(IN,JN),U(IN,JN,KN),V(IN,JN,KN),HT(IN,JN),HTD(IN,JN),
10     CHDUM(IN,JN)
11     KNM1=KN-1
12     DO 50 I=1,IN
13     DO 50 J=1,JN
14     HDUM(I,J)=0.0
15     DO 60 K=1,KN
16     IF(MAR(I,J).EQ.0) GO TO 50
17     IF(MAR(I,J).EQ.11) GO TO 11
18     IF(MAR(I,J).EQ.3) GO TO 12
19     IF(MAR(I,J).EQ.5) GO TO 19
20     IF(MAR(I,J).EQ.2) GO TO 13
21     IF(MAR(I,J).EQ.4) GO TO 14
22     IF(MAR(I,J).EQ.1) GO TO 20
23     IF(MAR(I,J).EQ.7) GO TO 15
24     IF(MAR(I,J).EQ.9) GO TO 16
25     IF(MAR(I,J).EQ.10) GO TO 17
26     IF(MAR(I,J).EQ.6) GO TO 11
27     IF(MAR(I,J).EQ.8) GO TO 11
28     11  D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
29     D1HUY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
30     GO TO 24
31     12  CONTINUE
32     IF(I.EQ.I5.AND.J.GE.J3.AND.J.LE.J4.AND.M.GT.1) GO TO 50
33     D1HUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
34     CU(I+2,J,K))/(2.*DX)
35     D1HUY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
36     GO TO 24
37     14  CONTINUE
38     D1HUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
39     CU(I-1,J,K))/(2.*DX)
40     D1HUY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
41     GO TO 24
42     13  CONTINUE
43     D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
44     D1HUY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
45     CV(I,J+2,K))/(2.*DY)
46     GO TO 24
47     20  CONTINUE
48     IF(J.EQ.J1.AND.I.GE.I1.AND.I.LE.I2.AND.M.GT.1) GO TO 50
49     D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
50     D1HUY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
51     CV(I,J-1,K))/(2.*DY)
52     GO TO 24
53     15  CONTINUE
54     D1HUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
55     CU(I+2,J,K))/(2.*DX)
56     D1HUY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*

```

```

57      CV(I,J+2,K))/(2.*DY)
58      GO TO 24
59      19  CONTINUE
60      DIHUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
61      CU(I+2,J,K))/(2.*DX)
62      DIHVV=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
63      CV(I,J-1,K))/(2.*DY)
64      GO TO 24
65      16  CONTINUE
66      DIHUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
67      CU(I-1,J,K))/(2.*DX)
68      DIHVV=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
69      CV(I,J+2,K))/(2.*DY)
70      GO TO 24
71      17  CONTINUE
72      DIHUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
73      CU(I-1,J,K))/(2.*DX)
74      DIHVV=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
75      CV(I,J-1,K))/(2.*DY)
76      GO TO 24
77      24  CONTINUE
78      C.....SIMPSON'S RULE IS USED FOR INTEGRATION
79      IF(K.EQ.1.OR.K.EQ.5) GO TO 101
80      IF(K.EQ.2.OR.K.EQ.4) GO TO 102
81      HDUM(I,J)=((DIHUX+DIHVV)*DZ*(2./3.))+HDUM(I,J)
82      GO TO 103
83      101  HDUM(I,J)=((DIHUX+DIHVV)*DZ/3.))+HDUM(I,J)
84      GO TO 103
85      102  HDUM(I,J)=((DIHUX+DIHVV)*DZ*(4./3.))+HDUM(I,J)
86      103  CONTINUE
87      60  CONTINUE
88      HTD(I,J)=HT(I,J)-HDUM(I,J)
89      50  CONTINUE
90      RETURN
91      END

```

7.1.9 HEILN

This subroutine calculates H at time level $n+1$ ($=HTE(I,J)$) from H at time level n ($=HTD(I,J)$) and H at time level $n-1$ ($=HT(I,J)$) by using central differencing in time.

Volume I gives the detailed finite difference scheme used by this subroutine for solving the surface height equation. The integration, once again, is performed by using Simpson's Rule. The general inlet and outlet conditions are incorporated as was just discussed for subroutine HEIGHT.

```

1      C
2      C*****
3      C   THIS SUBROUTINE CALCULATES THE TOTAL DEPTH AT EACH X-Y LOCATION
4      C   IN THE DOMAIN FOR THE SECOND TIME STEP AND THEREAFTER USING A CENTRAL
5      C   DIFFERENCING SCHEME IN TIME
6      C*****
7      SUBROUTINE HEILN(IN,JN,KN,MAR,U,V,HT,HTD,HTE,DZ,DT,DX,DY,HDUM,M
8      C,I1,I2,I3,I4,I5,I6,J1,J2,J3,J4,J5,J6)
9      DIMENSION MAR(IN,JN),U(IN,JN,KN),V(IN,JN,KN),HT(IN,JN),HTD(IN,JN),
10     CHDUM(IN,JN),HTE(IN,JN)
11     KNM1=KN-1
12     DO 50 I=1,IN
13     DO 50 J=1,JN
14     HDUM(I,J)=0.0
15     DO 60 K=1,KN
16     IF(MAR(I,J).EQ.0) GO TO 50
17     IF(MAR(I,J).EQ.11) GO TO 11
18     IF(MAR(I,J).EQ.3) GO TO 12
19     IF(MAR(I,J).EQ.5) GO TO 19
20     IF(MAR(I,J).EQ.2) GO TO 13
21     IF(MAR(I,J).EQ.1) GO TO 20
22     IF(MAR(I,J).EQ.4) GO TO 14
23     IF(MAR(I,J).EQ.7) GO TO 15
24     IF(MAR(I,J).EQ.9) GO TO 16
25     IF(MAR(I,J).EQ.10) GO TO 17
26     IF(MAR(I,J).EQ.6) GO TO 11
27     IF(MAR(I,J).EQ.8) GO TO 11
28     11  D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
29     D1HUY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
30     GO TO 24
31     12  CONTINUE
32     IF(I.EQ.I5.AND.J.GE.J3.AND.J.LE.J4.AND.M.GT.1) GO TO 50
33     D1HUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
34     CU(I+2,J,K))/(2.*DX)
35     D1HUY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
36     GO TO 24
37     14  CONTINUE
38     D1HUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
39     CU(I-1,J,K))/(2.*DX)
40     D1HUY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
41     GO TO 24
42     13  CONTINUE
43     D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
44     D1HUY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
45     CV(I,J+2,K))/(2.*DY)
46     GO TO 24
47     20  CONTINUE
48     IF(J.EQ.J1.AND.I.GE.I1.AND.I.LE.I2.AND.M.GT.1) GO TO 50
49     D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
50     D1HUY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
51     CV(I,J-1,K))/(2.*DY)
52     GO TO 24
53     15  CONTINUE
54     D1HUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
55     CU(I+2,J,K))/(2.*DX)
56     D1HUY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*

```

```

57      CV(I,J+2,K))/(2.*DY)
58      GO TO 24
59      19  CONTINUE
60      DIHUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
61      CU(I+2,J,K))/(2.*DX)
62      DIHVY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
63      CV(I,J-1,K))/(2.*DY)
64      GO TO 24
65      16  CONTINUE
66      DIHUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
67      CU(I-1,J,K))/(2.*DX)
68      DIHVY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
69      CV(I,J+2,K))/(2.*DY)
70      GO TO 24
71      17  CONTINUE
72      DIHUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
73      CU(I-1,J,K))/(2.*DX)
74      DIHVY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
75      CV(I,J-1,K))/(2.*DY)
76      GO TO 24
77      24  CONTINUE
78      C....SIMPSON'S RULE IS USED FOR INTEGRATION
79      IF(K.EQ.1.OR.K.EQ.5) GO TO 101
80      IF(K.EQ.2.OR.K.EQ.4) GO TO 102
81      HDUM(I,J)=((DIHUX+DIHVY)*DZ*(2./3.))+HDUM(I,J)
82      GO TO 103
83      101  HDUM(I,J)=((DIHUX+DIHVY)*DZ/3.))+HDUM(I,J)
84      GO TO 103
85      102  HDUM(I,J)=((DIHUX+DIHVY)*DZ*(4./3.))+HDUM(I,J)
86      103  CONTINUE
87      60  CONTINUE
88      HTE(I,J)=HTD(I,J)-HDUM(I,J)*2*DT
89      50  CONTINUE
90      RETURN
91      END

```

7.1.10 UVVEL

This subroutine calculates the horizontal components of velocity, u and v , at $t = \Delta t$ ($=H(I,J,K)$ and $G(I,J,K)$), respectively) from u and v at $t=0$ ($=U(I,J,K)$ and $V(I,J,K)$) by using a forward differencing in time.

Volume I details how the u and v momentum equations are solved.

Note, this subroutine is used only for the first time cycle.

The general inlet and outlet conditions are specified by reading in from data cards values of the parameters which set the location properly or the boundary. Modification, as mentioned earlier in the description of subroutine HEIGHT, may be easily incorporated by the user.

The spatial derivatives have been replaced by central differencing in the interior of the domain and three point single sided differencing on the boundaries, as shown in Volume I. Again, MAR=6 and MAR=8 boundary corners are tested as interior points.

```

1      C
2      C*****
3      C   THIS SUBROUTINE CALCULATES THE HORIZONTAL VELOCITIES, U, V, AT EACH
4      C   X-Y LOCATION AND DEPTH IN THE DOMAIN FOR THE FIRST TIME STEP USING
5      C   A FORWARD DIFFERENCING SCHEME IN TIME
6      C*****
7      SUBROUTINE UVVEL (IN, JN, KN, U, V, H, G, DX, DY, DZ, W, TAUX, TAUY,
8      CDT, HT, HX, HY, ETA, P, MAR, CI, CC, CP, CH, CV, KH, KV, GR, RR, FF, HTD, RO, T, I1
9      C, I2, I3, I4, I5, I6, J1, J2, J3, J4, J5, J6, M)
10     REAL KH, KV
11     DIMENSION U(IN, JN, KN), V(IN, JN, KN), H(IN, JN, KN), G(IN, JN, KN),
12     CHT(IN, JN), HX(IN, JN), HY(IN, JN), ETA(IN, JN),
13     CP(IN, JN, KN), MAR(IN, JN), HTD(IN, JN), W(IN, JN, KN)
14     C, RO(IN, JN, KN), T(IN, JN, KN)
15     KN1=KN-1
16     DO 10 I=1, IN
17     DO 10 J=1, JN
18     IF (MAR(I, J).EQ.0) GO TO 10
19     IF (MAR(I, J).EQ.5) GO TO 10
20     IF (MAR(I, J).EQ.7) GO TO 10
21     IF (MAR(I, J).EQ.9) GO TO 10
22     IF (MAR(I, J).EQ.10) GO TO 10
23     IF (MAR(I, J).EQ.3) GO TO 10
24     IF (MAR(I, J).EQ.4) GO TO 10
25     DO 8 K=1, KN1
26     IF (MAR(I, J).EQ.6) GO TO 11
27     IF (MAR(I, J).EQ.8) GO TO 11
28     IF (MAR(I, J).EQ.11) GO TO 11
29     IF (MAR(I, J).EQ.1) GO TO 101
30     IF (MAR(I, J).EQ.2) GO TO 102
31     11 CONTINUE
32     ETAX=(ETA(I+1, J)-ETA(I-1, J))/(2*DX)
33     ETAY=(ETA(I, J+1)-ETA(I, J-1))/(2*DY)
34     DIPX=(P(I+1, J, K)-P(I-1, J, K))/(2*DX)
35     DIHUUX=(U(I+1, J, K)*U(I+1, J, K)*HT(I+1, J)-U(I-1, J, K)*U(I-1, J, K)*
36     CHT(I-1, J))/(2*DX)
37     DIHUVY=(U(I, J+1, K)*V(I, J+1, K)*HT(I, J+1)-U(I, J-1, K)*V(I, J-1, K)*
38     CHT(I, J-1))/(2*DY)
39     DIUX=(U(I+1, J, K)-U(I-1, J, K))/(2*DX)
40     D2UX=(U(I+1, J, K)+U(I-1, J, K)-2*U(I, J, K))/(DX*DX)
41     DIUY=(U(I, J+1, K)-U(I, J-1, K))/(2*DY)
42     D2UY=(U(I, J+1, K)+U(I, J-1, K)-2*U(I, J, K))/(DY*DY)
43     GO TO 100
44     101 CONTINUE
45     IF (J.EQ.J1.AND.I.GE.I1.AND.I.LE.I2) GO TO 10
46     DHY=(3*HT(I, J)+HT(I, J-2)-4*HT(I, J-1))/(2*DY)
47     ETAX=(ETA(I+1, J)-ETA(I-1, J))/(2*DX)
48     ETAY=(3*ETA(I, J)+ETA(I, J-2)-4*ETA(I, J-1))/(2*DY)
49     DIPX=(P(I+1, J, K)-P(I-1, J, K))/(2*DX)
50     DIHUUX=(U(I+1, J, K)*U(I+1, J, K)*HT(I+1, J)-U(I-1, J, K)*U(I-1, J, K)*
51     CHT(I-1, J))/(2*DX)
52     DIHUVY=(3*U(I, J, K)*V(I, J, K)*HT(I, J)+U(I, J-2, K)*V(I, J-2, K)
53     C*HT(I, J-2)-4*U(I, J-1, K)*V(I, J-1, K)*HT(I, J-1))/(2*DY)
54     DIUX=(U(I+1, J, K)-U(I-1, J, K))/(2*DX)
55     D2UX=(U(I+1, J, K)+U(I-1, J, K)-2*U(I, J, K))/(DX*DX)
56     DIUY=0.0

```



```

57      D2UY=(U(I,J,K)+U(I,J-2,K)-2*U(I,J-1,K))/(DY*DY)
58      GO TO 100
59      102  CONTINUE
60      IF(J.EQ.J2.AND.I.GE.I3.AND.I.LE.I4) GO TO 10
61      ETAX=(ETA(I+1,J)-ETA(I-1,J))/(2*DX)
62      ETAY=(4*ETA(I,J+1)-3*ETA(I,J)-ETA(I,J+2))/(2*DY)
63      DHY=(4*HT(I,J+1)-3*HT(I,J)-HT(I,J+2))/(2*DY)
64      DIPX=(P(I+1,J,K)-P(I-1,J,K))/(2*DX)
65      D1HUUX=(U(I+1,J,K)+U(I+1,J,K)*HT(I+1,J)-U(I-1,J,K)+U(I-1,J,K)*
66      CHT(I-1,J))/(2*DX)
67      D1HUVY=(4*U(I,J+1,K)*V(I,J+1,K)+HT(I,J+1)-3*U(I,J,K)+V(I,J,K)
68      C*HT(I,J)-U(I,J+2,K)+V(I,J+2,K)*HT(I,J+2))/(2*DY)
69      D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
70      D2UX=(U(I+1,J,K)+U(I-1,J,K)-2*U(I,J,K))/(DX*DX)
71      D1UY=0.0
72      D2UY=(U(I,J,K)+U(I,J+2,K)-2*U(I,J+1,K))/(DY*DY)
73      100  CONTINUE
74      RO(I,J,K)=1.029431-.000020*T(I,J,K)-.0000048*(T(I,J,K)**2)
75      RR=RO(I,J,K)
76      IF(K.EQ.1) GO TO 70
77      D1UWZ=(U(I,J,K+1)*W(I,J,K+1)-U(I,J,K-1)*W(I,J,K-1))/(2*DZ)
78      D2UZ=(U(I,J,K+1)+U(I,J,K-1)-2*U(I,J,K))/(DZ*DZ)
79      GO TO 80
80      70  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)*
81      CW(I,J,K+2))/(2.*DZ)
82      D2UZ=(2*U(I,J,K+1)+(TAUX/KV)*HT(I,J)*2*DZ-2*U(I,J,K))/(DZ*DZ)
83      80  CONTINUE
84      UI=CI*(D1HUUX+D1HUVY+HT(I,J)*D1UWZ)
85      UP=CP*HT(I,J)*(ETAX*GR)*(-1.)
86      UC=CC*HT(I,J)*FF*V(I,J,K)
87      UH=CH*KH*(HX(I,J)*D1UX+ETAX*D1UX+HT(I,J)*D2UX)+CH*KH*(HY(I,J)*D1UY
88      C+ETAY*D1UY+HT(I,J)*D2UY)
89      UV=CV*KV*D2UZ/HT(I,J)
90      H(I,J,K)=((-UI+UP-UC+UH+UV)*DT+HT(I,J)*U(I,J,K))/HTD(I,J)
91      8  CONTINUE
92      10  CONTINUE
93      IF(M.EQ.1) GO TO 7000
94      IF(INLET.EQ.1) GO TO 7000
95      DO 6000 K=1,KN1
96      SUM=0.0
97      DO 6002 J=J3,J4
98      SUM=SUM+H(I5+1,J,K)
99      6002 CONTINUE
100     DO 6003 J=J3,J4
101     H(I5+1,J,K)=SUM/(J4-J3+1)
102     6003 CONTINUE
103     6000 CONTINUE
104     7000 CONTINUE
105     DO 970 I=1,IN
106     DO 970 J=1,JN
107     DO 960 K=1,KN1
108     IF(I.EQ.I5.AND.J.GE.J3.AND.J.LE.J4.AND.M.GT.1) GO TO 5002
109     IF(I.EQ.I6.AND.J.GE.J5.AND.J.LE.J6) GO TO 2005
110     GO TO 970
111     5002 H(I,J,K)=H(I+1,J,K)
112     GO TO 970
113     2005 H(I,J,K)=H(I-1,J,K)

```

```

114          960  CONTINUE
115          970  CONTINUE
116          DO 30 I=1,IN
117          DO 30 J=1,JN
118          IF(MAR(I,J).EQ.0) GO TO 30
119          IF(MAR(I,J).EQ.5) GO TO 30
120          IF(MAR(I,J).EQ.7) GO TO 30
121          IF(MAR(I,J).EQ.9) GO TO 30
122          IF(MAR(I,J).EQ.10) GO TO 30
123          DO 7 K=1,KN1
124          IF(MAR(I,J).EQ.6) GO TO 12
125          IF(MAR(I,J).EQ.8) GO TO 12
126          IF(MAR(I,J).EQ.11) GO TO 12
127          IF(MAR(I,J).EQ.1) GO TO 201
128          IF(MAR(I,J).EQ.2) GO TO 202
129          IF(MAR(I,J).EQ.3) GO TO 203
130          IF(MAR(I,J).EQ.4) GO TO 204
131          12  CONTINUE
132          ETAX=(ETA(I+1,J)-ETA(I-1,J))/(2*DX)
133          ETAY=(ETA(I,J+1)-ETA(I,J-1))/(2*DY)
134          DIPY=(P(I,J+1,K)-P(I,J-1,K))/(2*DY)
135          DIHUVX=(U(I+1,J,K)*V(I+1,J,K)*HT(I+1,J)-U(I-1,J,K)*V(I-1,J,K)*
136          CHT(I-1,J))/(2*DX)
137          DIHVY=(V(I,J+1,K)*V(I,J+1,K)*HT(I,J+1)-V(I,J-1,K)*V(I,J-1,K)*
138          CHT(I,J-1))/(2*DY)
139          DIVX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
140          D2VX=(V(I+1,J,K)+V(I-1,J,K)-2*V(I,J,K))/(DX*DX)
141          DIVY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
142          D2VY=(V(I,J+1,K)+V(I,J-1,K)-2*V(I,J,K))/(DY*DY)
143          GO TO 200
144          201  CONTINUE
145          GO TO 30
146          202  CONTINUE
147          GO TO 30
148          203  CONTINUE
149          IF(I.EQ.15.AND.J.GE.J3.AND.J.LE.J4) GO TO 30
150          ETAX=(4*ETA(I+1,J)-3*ETA(I,J)-ETA(I+2,J))/(2*DX)
151          ETAY=(ETA(I,J+1)-ETA(I,J-1))/(2*DY)
152          DHX=(4*HT(I+1,J)-3*HT(I,J)-HT(I+2,J))/(2*DX)
153          DIPY=(P(I,J+1,K)-P(I,J-1,K))/(2*DY)
154          DIHUVX=(4*U(I+1,J,K)*V(I+1,J,K)*HT(I+1,J)-3*U(I,J,K)*V(I,J,K)*
155          C*HT(I,J)-U(I+2,J,K)*V(I+2,J,K)*HT(I+2,J))/(2*DX)
156          DIHVY=(V(I,J+1,K)*V(I,J+1,K)*HT(I,J+1)-V(I,J-1,K)*V(I,J-1,K)*
157          CHT(I,J-1))/(2*DY)
158          DIVX=0.0
159          D2VX=(V(I,J,K)+V(I+2,J,K)-2*V(I+1,J,K))/(DX*DX)
160          DIVY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
161          D2VY=(V(I,J+1,K)+V(I,J-1,K)-2*V(I,J,K))/(DY*DY)
162          GO TO 200
163          204  CONTINUE
164          IF(I.EQ.16.AND.J.GE.J5.AND.J.LE.J6) GO TO 30
165          ETAX=(3*ETA(I,J)+ETA(I-2,J)-4*ETA(I-1,J))/(2*DX)
166          ETAY=(ETA(I,J+1)-ETA(I,J-1))/(2*DY)
167          DHX=(3*HT(I,J)+HT(I-2,J)-4*HT(I-1,J))/(2*DX)
168          DIPY=(P(I,J+1,K)-P(I,J-1,K))/(2*DY)
169          DIHUVX=(3*U(I,J,K)*V(I,J,K)*HT(I,J)+U(I-2,J,K)*V(I-2,J,K)*
170          C*HT(I-2,J)-4*U(I-1,J,K)*V(I-1,J,K)*HT(I-1,J))/(2*DX)

```

```
171      D1HVVY=(V(I,J+1,K)+V(I,J+1,K)*HT(I,J+1)-V(I,J-1,K)+V(I,J-1,K)*
172      CHT(I,J-1))/(2*DY)
173      DIVX=0.0
174      D2VX=(V(I,J,K)+V(I-2,J,K)-2*V(I-1,J,K))/(DX*DX)
175      DIVY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
176      D2VY=(V(I,J+1,K)+V(I,J-1,K)-2*V(I,J,K))/(DY*DY)
177      200  CONTINUE
178      RO(I,J,K)=1.029431-.000020*T(I,J,K)-.0000048*(T(I,J,K)**2)
179      RR=RO(I,J,K)
180      IF(K.EQ.1) GO TO 90
181      DIVWZ=(V(I,J,K+1)+W(I,J,K+1)-V(I,J,K-1)+W(I,J,K-1))/(2*DZ)
182      D2VZ=(V(I,J,K+1)+V(I,J,K-1)-2*V(I,J,K))/(DZ*DZ)
183      GO TO 95
184      90    DIVWZ=(4*V(I,J,K+1)+W(I,J,K+1)-3*V(I,J,K)+W(I,J,K)-V(I,J,K+2)*
185      CW(I,J,K+2))/(2.*DZ)
186      D2VZ=(2*V(I,J,K+1)+(TAUY/KV)*HT(I,J)+2*DZ-2*V(I,J,K))/(DZ*DZ)
187      95    CONTINUE
188      60    VI=CI*(D1HUVX+D1HVVY+HT(I,J)+DIVWZ)
189      VP=CP*HT(I,J)+(ETAY*GR)*(-1.)
190      VC=CC*HT(I,J)*FF*U(I,J,K)
191      VH=CH*KH*(HX(I,J)+D1VX+ETAX*D1VX+HT(I,J)+D2VX)+CH*KH*(HY(I,J)+D1VY
192      C+ETAY*D1VY+HT(I,J)+D2VY)
193      VV=CV*KV*D2VZ/HT(I,J)
194      G(I,J,K)=((-VI+VP+VC+VH+VV)*DT+HT(I,J)+V(I,J,K))/HTJ(I,J)
195      7    CONTINUE
196      30    CONTINUE
197      IF(M.EQ.1) GO TO 700
198      IF(INLET.GT.1) GO TO 700
199      DO 600 K=1,KN1
200      SUM=0.0
201      DO 602 I=I1,I2
202      SUM=SUM+G(I,J1-1,K)
203      602  CONTINUE
204      DO 603 I=I1,I2
205      G(I,J1-1,K)=SUM/(I2-I1+1)
206      603  CONTINUE
207      600  CONTINUE
208      700  CONTINUE
209      DO 97 I=1,IN
210      DO 97 J=1,JN
211      DO 96 K=1,KN1
212      IF(J.EQ.J2.AND.I.GE.I3.AND.I.LE.I4) GO TO 205
213      IF(J.EQ.J1.AND.I.GE.I1.AND.I.LE.I2.AND.M.GT.1) GO TO 502
214      GO TO 97
215      205  CONTINUE
216      G(I,J,K)=G(I,J+1,K)
217      GO TO 97
218      502  CONTINUE
219      G(I,J,K)=G(I,J-1,K)
220      96    CONTINUE
221      97    CONTINUE
222      RETURN
223      END
```

7.1.11 UVVELN

This subroutine calculates the horizontal components of velocity u and v at time level $n+1$ ($=D(I,J,K)$ and $E(I,J,K)$ respectively) from u and v at time level n ($=H(I,J,K)$ and $G(I,J,K)$) and u and v at time level $n-1$ ($=U(I,J,K)$ and $V(I,J,K)$) by using central differencing in time. The numerical scheme used for solving these equations is given in Volume I. Again, the general inlet and outlet conditions are specified and may be modified by the user. Note, DuFort-Frankel differencing is used for the vertical momentum diffusion term. The spatial derivatives have been differenced as shown in Volume I.

```

1      C
2      C*****
3      C THIS SUBROUTINE CALCULATES THE HORIZONTAL VELOCITIES, U,V, AT EACH
4      C X-Y LOCATION AND DEPTH IN THE DOMAIN FOR THE SECOND TIME STEP AND
5      C THEREAFTER USING A CENTRAL DIFFERENCING SCHEME IN TIME
6      C*****
7      SUBROUTINE UVVELN(IN,JN,KN,U,V,H,G,D,E,DX,DY,DZ,W,TAUX,TAUY,DT,
8      CHT,HTD,HTE,HX,MY,ETA,P,MAR,KH,KV,GR,RR,FF,CP,CC,CI,CA,CV,RO,T,I,
9      C,I2,I3,I4,I5,I6,J1,J2,J3,J4,J5,J6,M)
10     REAL KH,KV
11     DIMENSION U(IN,JN,KN),V(IN,JN,KN),H(IN,JN,KN),G(IN,JN,KN),
12     CD(IN,JN,KN),E(IN,JN,KN),HT(IN,JN),HTD(IN,JN),HTE(IN,JN),HX(IN,JN),
13     CHY(IN,JN),ETA(IN,JN),P(IN,JN,KN),MAR(IN,JN),W(IN,JN,KN)
14     C,RO(IN,JN,KN),T(IN,JN,KN)
15     KN1=KN-1
16     DO 10 I=1,IN
17     DO 10 J=1,JN
18     IF(MAR(I,J).EQ.0) GO TO 10
19     IF(MAR(I,J).EQ.5) GO TO 10
20     IF(MAR(I,J).EQ.7) GO TO 10
21     IF(MAR(I,J).EQ.9) GO TO 10
22     IF(MAR(I,J).EQ.10) GO TO 10
23     IF(MAR(I,J).EQ.3) GO TO 10
24     IF(MAR(I,J).EQ.4) GO TO 10
25     DO 8 K=1,KN1
26     IF(MAR(I,J).EQ.6) GO TO 11
27     IF(MAR(I,J).EQ.8) GO TO 11
28     IF(MAR(I,J).EQ.11) GO TO 11
29     IF(MAR(I,J).EQ.1) GO TO 101
30     IF(MAR(I,J).EQ.2) GO TO 102
31     11 CONTINUE
32     ETAX=(ETA(I+1,J)-ETA(I-1,J))/(2*DX)
33     DHX=(HT(I+1,J)-HT(I-1,J))/(2*DX)
34     DHY=(HT(I,J+1)-HT(I,J-1))/(2*DY)
35     DIPX=(P(I+1,J,K)-P(I-1,J,K))/(2*DX)
36     DIHUUX=(H(I+1,J,K)*H(I+1,J,K)*HTD(I+1,J)-H(I-1,J,K)*
37     CH(I-1,J,K)*HTD(I-1,J))/(2*DX)
38     DIHUVY=(H(I,J+1,K)*G(I,J+1,K)*HTD(I,J+1)-H(I,J-1,K)*
39     CG(I,J-1,K)*HTD(I,J-1))/(2*DY)
40     DIUX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
41     D2UX=(U(I+1,J,K)+U(I-1,J,K)-2*U(I,J,K))/(DX*DX)
42     DIUY=(U(I,J+1,K)-U(I,J-1,K))/(2*DY)
43     D2UY=(U(I,J+1,K)+U(I,J-1,K)-2*U(I,J,K))/(DY*DY)
44     GO TO 100
45     101 CONTINUE
46     IF(J.EQ.J1.AND.I.GE.I1.AND.I.LE.I2) GO TO 10
47     ETAX=(ETA(I+1,J)-ETA(I-1,J))/(2*DX)
48     DHX=(HT(I+1,J)-HT(I-1,J))/(2*DX)
49     DHY=(3*HT(I,J)+HT(I,J-2)-4*HT(I,J-1))/(2*DY)
50     DIPX=(P(I+1,J,K)-P(I-1,J,K))/(2*DX)
51     DIHUUX=(H(I+1,J,K)*H(I+1,J,K)*HTD(I+1,J)-H(I-1,J,K)*
52     CH(I-1,J,K)*HTD(I-1,J))/(2*DX)
53     DIHUVY=(3*H(I,J,K)*G(I,J,K)*HTD(I,J)+H(I,J-2,K)*G(I,J-2,K)
54     C*HTD(I,J-2)-4*H(I,J-1,K)*G(I,J-1,K)*HTD(I,J-1))/(2*DY)
55     DIUX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
56     D2UX=(U(I+1,J,K)+U(I-1,J,K)-2*U(I,J,K))/(DX*DX)

```

```

57      D1UY=0.0
58      D2UY=(U(I,J,K)+U(I,J-2,K)-2*U(I,J-1,K))/(DY*DY)
59      GO TO 100
60      102  CONTINUE
61      IF(J.EQ.J2.AND.I.GE.I3.AND.I.LE.I4) GO TO 10
62      ETAX=(ETA(I+1,J)-ETA(I-1,J))/(2*DX)
63      DHX=(HT(I+1,J)-HT(I-1,J))/(2*DX)
64      DHY=(4*HT(I,J+1)-3*HT(I,J)-HT(I,J+2))/(2*DY)
65      D1PX=(P(I+1,J,K)-P(I-1,J,K))/(2*DX)
66      D1HUUX=(H(I+1,J,K)*H(I+1,J,K)*HTD(I+1,J)-H(I-1,J,K)*
67      CH(I-1,J,K)*HTD(I-1,J))/(2*DX)
68      D1HUVY=(4*H(I,J+1,K)*G(I,J+1,K)*HTD(I,J+1)-3*H(I,J,K)*G(I,J,K)
69      C*HTD(I,J)-H(I,J+2,K)*G(I,J+2,K)*HTD(I,J+2))/(2*DY)
70      D1UX=(U(I+1,J,K)-U(I-1,J,K))/(2*DX)
71      D2UX=(U(I+1,J,K)+U(I-1,J,K)-2*U(I,J,K))/(DX*DX)
72      D1UY=0.0
73      D2UY=(U(I,J,K)+U(I,J+2,K)-2*U(I,J+1,K))/(DY*DY)
74      100  CONTINUE
75      RO(I,J,K)=1.029431-.000020*T(I,J,K)-.0000048*(T(I,J,K)**2)
76      RR=RO(I,J,K)
77      IF(K.EQ.1) GO TO 70
78      D1UWZ=(H(I,J,K+1)*W(I,J,K+1)-H(I,J,K-1)*W(I,J,K-1))/(2*DZ)
79      UZ=H(I,J,K+1)*H(I,J,K-1)-U(I,J,K)
80      GO TO 80
81      70   D1UWZ=(4*H(I,J,K+1)*W(I,J,K+1)-3*H(I,J,K)*W(I,J,K)-H(I,J,K+2)*
82      CW(I,J,K+2))/(2.*DZ)
83      UZ=2*H(I,J,K+1)*(TAUX/KV)*HTD(I,J)+2*DZ-U(I,J,K)
84      80   CONTINUE
85      UI=C1*(D1HUUX+D1HUVY+HTD(I,J)+D1UWZ)
86      UP=CP*HTD(I,J)*(ETAX*GR)*(-1.)
87      UC=CC*HTD(I,J)*FF*G(I,J,K)
88      UH=CH*KH*(DHX*D1UX+HT(I,J)*D2UX)+CH*KH*(DHY*D1UY+HT(I,J)*D2UY)
89      D(I,J,K)=((-UI+UP-UC+UH)*2*DT+((CV*KV+UZ)/(DZ*DZ*HTD(I,J)))*2*
90      CDT+U(I,J,K)*HT(I,J))/HTE(I,J)/(1.+(CV*KV/(DZ*DZ*HTD(I,J)))*2*DT
91      C/HTE(I,J))
92      8   CONTINUE
93      10  CONTINUE
94      IF(M.EQ.1) GO TO 7000
95      IF(INLET.EQ.1) GO TO 7000
96      DO 6000 K=1,KN1
97      SUM=0.0
98      DO 6002 J=J3,J4
99      SUM=SUM+D(I5+1,J,K)
100     6002 CONTINUE
101     DO 6003 J=J3,J4
102     D(I5+1,J,K)=SUM/(J4-J3+1)
103     6003 CONTINUE
104     6000 CONTINUE
105     7000 CONTINUE
106     DO 970 I=1,IN
107     DO 970 J=1,JN
108     DO 960 K=1,KN1
109     IF(I.EQ.I5.AND.J.GE.J3.AND.J.LE.J4.AND.M.GT.1) GO TO 5002
110     IF(I.EQ.I6.AND.J.GE.J5.AND.J.LE.J6) GO TO 2005
111     GO TO 970
112     5002 D(I,J,K)=D(I+1,J,K)
113     GO TO 970

```

```

114      2005  D(I,J,K)=D(I-1,J,K)
115      960  CONTINUE
116      970  CONTINUE
117      DO 30 I=1,IN
118      DO 30 J=1,JN
119      IF(MAR(I,J).EQ.C) GO TO 30
120      IF(MAR(I,J).EQ.5) GO TO 30
121      IF(MAR(I,J).EQ.7) GO TO 30
122      IF(MAR(I,J).EQ.9) GO TO 30
123      IF(MAR(I,J).EQ.10) GO TO 30
124      DO 7 K=1,KN1
125      IF(MAR(I,J).EQ.6) GO TO 12
126      IF(MAR(I,J).EQ.8) GO TO 12
127      IF(MAR(I,J).EQ.11) GO TO 12
128      IF(MAR(I,J).EQ.1) GO TO 201
129      IF(MAR(I,J).EQ.2) GO TO 202
130      IF(MAR(I,J).EQ.3) GO TO 203
131      IF(MAR(I,J).EQ.4) GO TO 204
132      12  CONTINUE
133      ETAY=(ETA(I,J+1)-ETA(I,J-1))/(2*DY)
134      DHX=(HT(I+1,J)-HT(I-1,J))/(2*DX)
135      DHY=(HT(I,J+1)-HT(I,J-1))/(2*DY)
136      DIPY=(P(I,J+1,K)-P(I,J-1,K))/(2*DY)
137      DIHUVX=(H(I+1,J,K)+G(I+1,J,K)*HTD(I+1,J)-H(I-1,J,K)+G(I-1,J,K)*
138      CHTD(I-1,J))/(2*DX)
139      DIHVY=(G(I,J+1,K)+G(I,J+1,K)*HTD(I,J+1)-G(I,J-1,K)+
140      CG(I,J-1,K)*HTD(I,J-1))/(2*DY)
141      DIVX=(V(I+1,J,K)-V(I-1,J,K))/(2*DX)
142      D2VX=(V(I+1,J,K)+V(I-1,J,K)-2*V(I,J,K))/(DX*DX)
143      DIVY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
144      D2VY=(V(I,J+1,K)+V(I,J-1,K)-2*V(I,J,K))/(DY*DY)
145      GO TO 200
146      201  CONTINUE
147      GO TO 30
148      202  CONTINUE
149      GO TO 30
150      203  CONTINUE
151      IF(I.EQ.15.AND.J.GE.J3.AND.J.LE.J4) GO TO 30
152      ETAY=(ETA(I,J+1)-ETA(I,J-1))/(2*DY)
153      DHX=(4*HT(I+1,J)-3*HT(I,J)-HT(I+2,J))/(2*DX)
154      DHY=(HT(I,J+1)-HT(I,J-1))/(2*DY)
155      DIPY=(P(I,J+1,K)-P(I,J-1,K))/(2*DY)
156      DIHUVX=(4*H(I+1,J,K)+G(I+1,J,K)*HTD(I+1,J)-3*H(I,J,K)+G(I,J,K)
157      C*HTD(I,J)-H(I+2,J,K)+G(I+2,J,K)*HTD(I+2,J))/(2*DX)
158      DIHVY=(G(I,J+1,K)+G(I,J+1,K)*HTD(I,J+1)-G(I,J-1,K)+
159      CG(I,J-1,K)*HTD(I,J-1))/(2*DY)
160      DIVX=0.0
161      D2VX=(V(I,J,K)+V(I+2,J,K)-2*V(I+1,J,K))/(DX*DX)
162      DIVY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
163      D2VY=(V(I,J+1,K)+V(I,J-1,K)-2*V(I,J,K))/(DY*DY)
164      GO TO 200
165      204  CONTINUE
166      IF(I.EQ.16.AND.J.GE.J5.AND.J.LE.J6) GO TO 30
167      ETAY=(ETA(I,J+1)-ETA(I,J-1))/(2*DY)
168      DHX=(3*HT(I,J)+HT(I-2,J)-4*HT(I-1,J))/(2*DX)
169      DHY=(HT(I,J+1)-HT(I,J-1))/(2*DY)
170      DIPY=(P(I,J+1,K)-P(I,J-1,K))/(2*DY)

```

```

171      D1HUVX=(3*H(I,J,K)+G(I,J,K)*HTD(I,J)+H(I-2,J,K)+G(I-2,J,K)
172      C*HTD(I-2,J)-4*H(I-1,J,K)+G(I-1,J,K)*HTD(I-1,J))/(2*DX)
173      D1HVY=(G(I,J+1,K)+G(I,J-1,K)*HTD(I,J+1)-G(I,J-1,K)*
174      CG(I,J-1,K)+HTD(I,J-1))/(2*DY)
175      DIVX=0.0
176      D2VX=(V(I,J,K)+V(I-2,J,K)-2*V(I-1,J,K))/(DX*DX)
177      DIVY=(V(I,J+1,K)-V(I,J-1,K))/(2*DY)
178      D2VY=(V(I,J+1,K)+V(I,J-1,K)-2*V(I,J,K))/(DY*DY)
179      200  CONTINUE
180      RO(I,J,K)=1.029431-.000020*(I,J,K)-.0000048*(I(I,J,K)**2)
181      RR=RO(I,J,K)
182      IF(K.EQ.1) GO TO 90
183      D1VMZ=(G(I,J,K+1)*W(I,J,K+1)-G(I,J,K-1)*W(I,J,K-1))/(2*DZ)
184      VZ=G(I,J,K+1)+G(I,J,K-1)-V(I,J,K)
185      GO TO 95
186      90  D1VMZ=(4*G(I,J,K+1)*W(I,J,K+1)-3*G(I,J,K)*W(I,J,K)-G(I,J,K+2)*
187      CW(I,J,K+2))/(2.*DZ)
188      VZ=2*G(I,J,K+1)+(TAUY/KV)*HTD(I,J)*2*DZ-V(I,J,K)
189      95  CONTINUE
190      VI=CI*(D1HUVX+D1HVY+HTD(I,J)*DIVZ)
191      VP=CP*HTD(I,J)*(ETAY*GR)*(-1.)
192      VC=CC*HTD(I,J)*GF*H(I,J,K)
193      VH=CH*KH*(DHX*D1VX+HT(I,J)*D2VX)+CH*KH*(DHY*D1VY+HT(I,J)*D2VY)
194      E(I,J,K)=((-VI+VP+VC+VH)*2*DT+((CV*KV+VZ)/(DZ*DZ*HT(I,J)))*2*DT
195      C+V(I,J,K)*HT(I,J))/HTE(I,J))/(1.+(CV*KV/(DZ*DZ*HT(I,J)))*2*DT
196      C/HTE(I,J))
197      7  CONTINUE
198      30  CONTINUE
199      IF(M.EQ.1) GO TO 700
200      IF(INLET.GT.1) GO TO 700
201      DO 600 K=1,KN1
202      SUM=0.0
203      DO 602 I=I1,I2
204      SUM=SUM+E(I,J1-1,K)
205      602  CONTINUE
206      DO 603 I=I1,I2
207      E(I,J1-1,K)=SUM/(I2-I1+1)
208      603  CONTINUE
209      600  CONTINUE
210      700  CONTINUE
211      DO 97 I=1,IN
212      DO 97 J=1,JN
213      DO 96 K=1,KN1
214      IF(J.EQ.J2.AND.I.GE.I3.AND.I.LE.I4) GO TO 205
215      IF(J.EQ.J1.AND.I.GE.I1.AND.I.LE.I2.AND.M.GT.1) GO TO 502
216      GO TO 97
217      205  CONTINUE
218      E(I,J,K)=E(I,J+1,K)
219      GO TO 97
220      502  CONTINUE
221      E(I,J,K)=E(I,J-1,K)
222      96  CONTINUE
223      97  CONTINUE
224      RETURN
225      END

```


7.1.12 WVEL

This subroutine calculates the equivalent vertical velocity $\Omega(I,J,K)$ in the α, β, σ ($=x,y,\sigma$) coordinate system at each x, y location and depth σ in the domain $\Omega(I,J,K)$ at $t = \Delta t$. $(=W(I,J,K))$ is calculated from $u, v,$ and H at $t = \Delta t$ ($=H(I,J,K)$) $G(I,J,K), HTD(I,J,K)$ as shown in Volume I. Thereafter, $\Omega(I,J,K)$ at time level $n+1$ is calculated from u, v and H at time level $n+1$ ($=D(I,J,K), E(I,J,K), HTE(I,J)$) Simpson's Rule is used for performing the integration.

```

1      C
2      C*****
3      C   THIS SUBROUTINE CALCULATES THE EQUIVALENT VERTICAL VELOCITY IN THE SI
4      C   COORDINATE SYSTEM AT EACH X-Y LOCATION AND DEPTH IN THE DOMAIN
5      C   AT EACH TIME STEP
6      C*****
7      SUBROUTINE WVEL (IN,JN,KN,U,V,W,HT,DX,DY,DZ,MAR,M,I1,I2,J3,J4)
9      DIMENSION U(IN,JN,KN),V(IN,JN,KN),HT(IN,JN),W(IN,JN,KN),MAR(IN,JN)
9      KN1=KN-1
10     DO 10 I=1,IN
11     DO 10 J=1,JN
12     DUM=0.
13     DO 9 K=1,KN
14     IF(MAR(I,J).EQ.C) GO TO 10
15     IF(MAR(I,J).EQ.11) GO TO 11
16     IF(MAR(I,J).EQ.6) GO TO 11
17     IF(MAR(I,J).EQ.8) GO TO 11
18     IF(MAR(I,J).EQ.1.AND.I.GE.I1.AND.I.LE.I2.AND.M.GT.1) GO TO 12
19     IF(MAR(I,J).EQ.3.AND.J.GE.J3.AND.J.LE.J4.AND.M.GT.1) GO TO 12
20     IF(MAR(I,J).LT.11) GO TO 10
21     11  D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
22     D1HUY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
23     GO TO 24
24     12  CONTINUE
25     CALL WVEL1(I,J,K,IN,JN,KN,U,V,HT,DX,DY,MAR,D1HUX,D1HUY)
26     24  CONTINUE
27     IF(K.EQ.1.OR.K.EQ.5) GO TO 27
28     IF(K.EQ.2.OR.K.EQ.4) GO TO 28
29     DUM=DUM+DZ*(2./3.)*(D1HUX+D1HUY)/HT(I,J)
30     GO TO 9
31     27  DUM=DUM+(DZ/3.)*(D1HUX+D1HUY)/HT(I,J)
32     GO TO 9
33     28  DUM=DUM+DZ*(4./7.)*(D1HUX+D1HUY)/HT(I,J)
34     9   CONTINUE
35     WUD=0.
36     DO 8 K=2,KN
37     IF(MAR(I,J).EQ.C) GO TO 10
38     IF(MAR(I,J).EQ.11) GO TO 111
39     IF(MAR(I,J).EQ.6) GO TO 111
40     IF(MAR(I,J).EQ.8) GO TO 111
41     IF(MAR(I,J).EQ.1.AND.I.GE.I1.AND.I.LE.I2.AND.M.GT.1) GO TO 112
42     IF(MAR(I,J).EQ.3.AND.J.GE.J3.AND.J.LE.J4.AND.M.GT.1) GO TO 112
43     IF(MAR(I,J).LT.11) GO TO 10
44     111  D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
45     D1HUX1=(HT(I+1,J)*U(I+1,J,K-1)-HT(I-1,J)*U(I-1,J,K-1))/(2.*DX)
46     D1HUY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
47     D1HUY1=(HT(I,J+1)*V(I,J+1,K-1)-HT(I,J-1)*V(I,J-1,K-1))/(2.*DY)
48     GO TO 200
49     112  CONTINUE
50     CALL WVEL2(I,J,K,IN,JN,KN,U,V,HT,DX,DY,MAR,D1HUX,D1HUY,
51     CD1HUX1,CD1HUY1)
52     200  CONTINUE
53     IF(K.EQ.2) GO TO 101
54     IF(K.EQ.4) GO TO 101
55     IF(K.EQ.3) GO TO 102
56     102  WUD=WUD+DZ*((2./3.)*(D1HUX1+D1HUY1)+(1./3.)*(D1HUX+D1HUY))/

```

```
57      CHT(I,J)
58      GO TO 300
59      101  WUD=WUD+DZ*((1./3.)*(DIHUX1+DIHVV1)+(2./3.)*(DIHUX+DIHVV))/
60      CHT(I,J)
61      GO TO 300
62      300  CONTINUE
63      W(I,J,K)=-WUD+DUM*(K-1)*DZ
64      8    CONTINUE
65      10   CONTINUE
66      RETURN
67      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

PRECEDING PAGE BLANK NOT FILMED

7.1.13 WVELL

This subroutine calculates the differential terms, in differenced form, in the definite integral for the equivalent vertical velocity, at each time step, from u , v , H at $t=\Delta t$, and at the time level $n+1$, thereafter.

```

1      C***
2      C   THIS SUBROUTINE CALCULATES THE DIFFERENTIAL (DIFFERENCED) TERMS IN THE
3      C   DEFINITE INTEGRAL FOR THE EQUIVALENT VERTICAL VELOCITY AT
4      C   EACH TIME STEP
5      C*****
6      SUBROUTINE WVEL1(I,J,K,IN,JN,KN,U,V,HT,DX,DY,MAR,DIHUX,DIHUY)
7      DIMENSION U(IN,JN,KN),V(IN,JN,KN),HT(IN,JN),MAR(IN,JN)
8      IF(MAR(I,J).EQ.3) GO TO 12
9      IF(MAR(I,J).EQ.5) GO TO 19
10     IF(MAR(I,J).EQ.2) GO TO 13
11     IF(MAR(I,J).EQ.1) GO TO 20
12     IF(MAR(I,J).EQ.4) GO TO 14
13     IF(MAR(I,J).EQ.7) GO TO 15
14     IF(MAR(I,J).EQ.9) GO TO 16
15     IF(MAR(I,J).EQ.10) GO TO 17
16     12   CONTINUE
17         DIHUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
18         CU(I+2,J,K))/(2.*DX)
19         DIHUY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
20         GO TO 24
21     14   CONTINUE
22         DIHUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
23         CU(I-1,J,K))/(2.*DX)
24         DIHUY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
25         GO TO 24
26     13   CONTINUE
27         IF(J.EQ.1.AND.I.GE.31.AND.I.LE.33) GO TO 31
28         DIHUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
29         DIHUY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
30         CV(I,J+2,K))/(2.*DY)
31         GO TO 24
32     31   CONTINUE
33         DIHUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
34         DIHUY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
35         CV(I,J+2,K))/(2.*DY)
36         GO TO 24
37     20   CONTINUE
38         IF(J.EQ.11.AND.I.GE.7.AND.I.LE.16) GO TO 32
39         DIHUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
40         DIHUY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
41         CV(I,J-1,K))/(2.*DY)
42         GO TO 24
43     32   CONTINUE
44         DIHUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
45         DIHUY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
46         CV(I,J-1,K))/(2.*DY)
47         GO TO 24
48     15   CONTINUE
49         DIHUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
50         CU(I+2,J,K))/(2.*DX)
51         DIHUY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
52         CV(I,J+2,K))/(2.*DY)
53         GO TO 24
54     19   CONTINUE
55         DIHUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
56         CU(I+2,J,K))/(2.*DX)

```

```

57      D1HVY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
58      CV(I,J-1,K))/(2.*DY)
59      GO TO 24
60      16      CONTINUE
61      D1HUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
62      CU(I-1,J,K))/(2.*DX)
63      D1HVV=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
64      CV(I,J+2,K))/(2.*DY)
65      GO TO 24
66      17      CONTINUE
67      D1HUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
68      CU(I-1,J,K))/(2.*DX)
69      D1HVY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
70      CV(I,J-1,K))/(2.*DY)
71      24      CONTINUE
72      RETURN
73      END

```

PRECEDING PAGE BLANK NOT FILMED

7.1.14 WVEL2

This subroutine calculates the differential time, in differenced form, in the indefinite integral for the equivalent vertical velocity at each time step. u , v , and HT are used at $t = \Delta t$ and D , E , HTE are used at time level $n+1$, thereafter.

```

1 C*****
2 C THIS SUBROUTINE CALCULATES THE DIFFERENTIAL (DIFFERENCED) TERMS IN
3 C THE INDEFINITE INTEGRAL FOR THE EQUIVALENT VERTICAL VELOCITY
4 C AT EACH TIME STEP
5 C*****
6 SUBROUTINE WVEL2(I,J,K,IN,JN,KN,U,V,HT,DX,DY,MAR,DIHUX,DIHUY,
7 COIHUX1,DIHUY1)
8 DIMENSION U(IN,JN,KN),V(IN,JN,KN),HT(IN,JN),MAR(IN,JN)
9 IF(MAR(I,J).EQ.3) GO TO 112
10 IF(MAR(I,J).EQ.5) GO TO 119
11 IF(MAR(I,J).EQ.2) GO TO 113
12 IF(MAR(I,J).EQ.1) GO TO 120
13 IF(MAR(I,J).EQ.4) GO TO 114
14 IF(MAR(I,J).EQ.7) GO TO 115
15 IF(MAR(I,J).EQ.9) GO TO 116
16 IF(MAR(I,J).EQ.10) GO TO 117
17 112 CONTINUE
18 DIHUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
19 CU(I+2,J,K))/(2.*DX)
20 DIHUX1=(4*HT(I+1,J)*U(I+1,J,K-1)-3*HT(I,J)*U(I,J,K-1)-HT(I+2,J)*
21 CU(I+2,J,K-1))/(2.*DX)
22 DIHUY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
23 DIHUY1=(HT(I,J+1)*V(I,J+1,K-1)-HT(I,J-1)*V(I,J-1,K-1))/(2.*DY)
24 GO TO 200
25 114 CONTINUE
26 DIHUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
27 CU(I-1,J,K))/(2.*DX)
28 DIHUX1=(3*HT(I,J)*U(I,J,K-1)+HT(I-2,J)*U(I-2,J,K-1)-4*HT(I-1,J)*
29 CU(I-1,J,K-1))/(2.*DX)
30 DIHUY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
31 DIHUY1=(HT(I,J+1)*V(I,J+1,K-1)-HT(I,J-1)*V(I,J-1,K-1))/(2.*DY)
32 GO TO 200
33 113 CONTINUE
34 IF(J.EQ.1.AND.I.GE.31.AND.I.LE.33) GO TO 311
35 DIHUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
36 DIHUX1=(HT(I+1,J)*U(I+1,J,K-1)-HT(I-1,J)*U(I-1,J,K-1))/(2.*DX)
37 DIHUY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
38 CV(I,J+2,K))/(2.*DY)
39 DIHUY1=(4*HT(I,J+1)*V(I,J+1,K-1)-3*HT(I,J)*V(I,J,K-1)-HT(I,J+2)*
40 CV(I,J+2,K-1))/(2.*DY)
41 GO TO 200
42 311 CONTINUE
43 DIHUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
44 DIHUX1=(HT(I+1,J)*U(I+1,J,K-1)-HT(I-1,J)*U(I-1,J,K-1))/(2.*DX)
45 DIHUY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
46 CV(I,J+2,K))/(2.*DY)
47 DIHUY1=(4*HT(I,J+1)*V(I,J+1,K-1)-3*HT(I,J)*V(I,J,K-1)-HT(I,J+2)*
48 CV(I,J+2,K-1))/(2.*DY)
49 GO TO 200
50 120 CONTINUE
51 IF(J.EQ.11.AND.I.GE.7.AND.I.LE.16) GO TO 321
52 DIHUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
53 DIHUX1=(HT(I+1,J)*U(I+1,J,K-1)-HT(I-1,J)*U(I-1,J,K-1))/(2.*DX)
54 DIHUY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
55 CV(I,J-1,K))/(2.*DY)
56 DIHUY1=(3*HT(I,J)*V(I,J,K-1)+HT(I,J-2)*V(I,J-2,K-1)-4*HT(I,J-1)*

```

ORIGINAL PAGE IS
OF POOR QUALITY


```

57      CV(I,J-1,K-1))/(2.*DY)
58      GO TO 200
59      321  CONTINUE
60      D1HUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
61      D1HUX1=(HT(I+1,J)*U(I+1,J,K-1)-HT(I-1,J)*U(I-1,J,K-1))/(2.*DX)
62      D1HVY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
63      CV(I,J-1,K))/(2.*DY)
64      D1HVY1=(3*HT(I,J)*V(I,J,K-1)+HT(I,J-2)*V(I,J-2,K-1)-4*HT(I,J-1)*
65      CV(I,J-1,K-1))/(2.*DY)
66      GO TO 200
67      115  CONTINUE
68      D1HUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
69      CU(I+2,J,K))/(2.*DX)
70      D1HUX1=(4*HT(I+1,J)*U(I+1,J,K-1)-3*HT(I,J)*U(I,J,K-1)-HT(I+2,J)*
71      CU(I+2,J,K-1))/(2.*DX)
72      D1HVY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
73      CV(I,J+2,K))/(2.*DY)
74      D1HVY1=(4*HT(I,J+1)*V(I,J+1,K-1)-3*HT(I,J)*V(I,J,K-1)-HT(I,J+2)*
75      CV(I,J+2,K-1))/(2.*DY)
76      GO TO 200
77      119  CONTINUE
78      D1HUX=(4*HT(I+1,J)*U(I+1,J,K)-3*HT(I,J)*U(I,J,K)-HT(I+2,J)*
79      CU(I+2,J,K))/(2.*DX)
80      D1HUX1=(4*HT(I+1,J)*U(I+1,J,K-1)-3*HT(I,J)*U(I,J,K-1)-HT(I+2,J)*
81      CU(I+2,J,K-1))/(2.*DX)
82      D1HVY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
83      CV(I,J-1,K))/(2.*DY)
84      D1HVY1=(3*HT(I,J)*V(I,J,K-1)+HT(I,J-2)*V(I,J-2,K-1)-4*HT(I,J-1)*
85      CV(I,J-1,K-1))/(2.*DY)
86      GO TO 200
87      116  CONTINUE
88      D1HUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
89      CU(I-1,J,K))/(2.*DX)
90      D1HUX1=(3*HT(I,J)*U(I,J,K-1)+HT(I-2,J)*U(I-2,J,K-1)-4*HT(I-1,J)*
91      CU(I-1,J,K-1))/(2.*DX)
92      D1HVY=(4*HT(I,J+1)*V(I,J+1,K)-3*HT(I,J)*V(I,J,K)-HT(I,J+2)*
93      CV(I,J+2,K))/(2.*DY)
94      D1HVY1=(4*HT(I,J+1)*V(I,J+1,K-1)-3*HT(I,J)*V(I,J,K-1)-HT(I,J+2)*
95      CV(I,J+2,K-1))/(2.*DY)
96      GO TO 200
97      117  CONTINUE
98      D1HUX=(3*HT(I,J)*U(I,J,K)+HT(I-2,J)*U(I-2,J,K)-4*HT(I-1,J)*
99      CU(I-1,J,K))/(2.*DX)
100     D1HUX1=(3*HT(I,J)*U(I,J,K-1)+HT(I-2,J)*U(I-2,J,K-1)-4*HT(I-1,J)*
101     CU(I-1,J,K-1))/(2.*DX)
102     D1HVY=(3*HT(I,J)*V(I,J,K)+HT(I,J-2)*V(I,J-2,K)-4*HT(I,J-1)*
103     CV(I,J-1,K))/(2.*DY)
104     D1HVY1=(3*HT(I,J)*V(I,J,K-1)+HT(I,J-2)*V(I,J-2,K-1)-4*HT(I,J-1)*
105     CV(I,J-1,K-1))/(2.*DY)
106     200  CONTINUE
107     RETURN
108     END

```

7.1.15 WW

This subroutine converts Ω to W , that is the vertical component of velocity. The following analytic expression is used for this conversion:

$$W = \Omega H + (\sigma) \frac{dh}{dt} + (\sigma - 1) \frac{dn}{dt}$$

The actual vertical velocity component, W is defined as $WZ(I,J,K)$ in the model program, and it is calculated at each x, y, σ . Since $WZ(I,J,K)$ is not used in solving the system of governing equations, this subroutine is used only after the last time cycle for each computer run.

```

1 C*****
2 C THIS SUBROUTINE TRANSFORMS THE EQUIVALENT VERTICAL VELOCITY
3 C (IN THE SIGMA COORDINATE SYSTEM) INTO THE ACTUAL VERTICAL VELOCITY
4 C (IN THE X-Y-Z COORDINATE SYSTEM) AT EACH X-Y LOCATION AND DEPTH
5 C IN THE DOMAIN
6 C*****
7 SUBROUTINE WZ(IN,JN,KN,HTD,HTE,ETA,D,E,W,WZ,MAR,DX,DY,DZ,DT,HX,HY)
8 DIMENSION HTD(IN,JN),HTE(IN,JN),ETA(IN,JN),W(IN,JN,KN),
9 CWZ(IN,JN,KN),MAR(IN,JN),D(IN,JN,KN),E(IN,JN,KN),HX(IN,JN)
10 C,MY(IN,JN)
11 KN1=KN-1
12 DO 10 I=1,IN
13 DO 10 J=1,JN
14 DO 9 K=1,KN1
15 WZ(I,J,K)=0.0
16 IF(MAR(I,J).EQ.C) GO TO 9
17 IF(MAR(I,J).EQ.11) GO TO 11
18 IF(MAR(I,J).EQ.7) GO TO 11
19 IF(MAR(I,J).EQ.8) GO TO 11
20 IF(MAR(I,J).EQ.1) GO TO 101
21 IF(MAR(I,J).EQ.2) GO TO 102
22 IF(MAR(I,J).EQ.3) GO TO 103
23 IF(MAR(I,J).EQ.4) GO TO 104
24 IF(MAR(I,J).EQ.5) GO TO 105
25 IF(MAR(I,J).EQ.7) GO TO 107
26 IF(MAR(I,J).EQ.9) GO TO 109
27 IF(MAR(I,J).EQ.10) GO TO 110
28
29 11 CONTINUE
30 ETAX=(ETA(I+1,J)-ETA(I-1,J))/(2.*DX)
31 ETAY=(ETA(I,J+1)-ETA(I,J-1))/(2.*DY)
32 GO TO 100
33 101 ETAX=(ETA(I+1,J)-ETA(I-1,J))/(2.*DX)
34 ETAY=(3*ETA(I,J)+ETA(I,J-2)-4*ETA(I,J-1))/(2.*DY)
35 GO TO 100
36 102 ETAX=(ETA(I+1,J)-ETA(I-1,J))/(2.*DX)
37 ETAY=(4*ETA(I,J+1)-3*ETA(I,J)-ETA(I,J+2))/(2.*DY)
38 GO TO 100
39 103 ETAX=(4*ETA(I+1,J)-3*ETA(I,J)-ETA(I+2,J))/(2.*DX)
40 ETAY=(ETA(I,J+1)-ETA(I,J-1))/(2.*DY)
41 GO TO 100
42 104 ETAX=(3*ETA(I,J)+ETA(I-2,J)-4*ETA(I-1,J))/(2.*DX)
43 ETAY=(ETA(I,J+1)-ETA(I,J-1))/(2.*DY)
44 GO TO 100
45 105 ETAX=(4*ETA(I+1,J)-3*ETA(I,J)-ETA(I+2,J))/(2.*DX)
46 ETAY=(3*ETA(I,J)+ETA(I,J-2)-4*ETA(I,J-1))/(2.*DY)
47 GO TO 100
48 107 ETAX=(4*ETA(I+1,J)-3*ETA(I,J)-ETA(I+2,J))/(2.*DX)
49 ETAY=(4*ETA(I,J+1)-3*ETA(I,J)-ETA(I,J+2))/(2.*DY)
50 GO TO 100
51 109 ETAX=(3*ETA(I,J)+ETA(I-2,J)-4*ETA(I-1,J))/(2.*DX)
52 ETAY=(4*ETA(I,J+1)-3*ETA(I,J)-ETA(I,J+2))/(2.*DY)
53 GO TO 100
54 110 ETAX=(3*ETA(I,J)+ETA(I-2,J)-4*ETA(I-1,J))/(2.*DX)
55 ETAY=(3*ETA(I,J)+ETA(I,J-2)-4*ETA(I,J-1))/(2.*DY)
56 CONTINUE
57 WZ(I,J,K)=HTE(I,J)+W(I,J,K)+((K-1)*DZ-1.)*(HTE(I,J)-HTD(I,J))/DT

```

```
57      C=D(I,J,K)*ETAX+E(I,J,K)*ETAY+((K-1)*DZ)*(D(I,J,K)
58      C*HX(I,J)+C(I,J,K)*HY(I,J))
59      9      CONTINUE
60      10     CONTINUE
61      RETURN
62      END
```

PRECEDING PAGE BLANK NOT FILMED

7.1.16 PRES

This subroutine calculates the pressure field at time level $n+1$ by knowing the contour depth, H and density, ρ at $n+1$. Note, that this is the integrated form of the hydrostatic equation. The integration is performed by applying the trapezoidal rule.

```

1      C
2      C.....
3      C    THIS SUBROUTINE CALCULATES THE PRESSURE FIELD
4      C.....
5      SUBROUTINE PRES (IN, JN, KN, HT, RO, GR, P, DZ)
6      DIMENSION HT(IN, JN), RO(IN, JN, KN), P(IN, JN, KN)
7      DO 10 I=1, IN
8      DO 10 J=1, JN
9      P(I, J, 1)=0.0
10     DO 8 K=2, KN
11     P(I, J, K)=P(I, J, K-1)+GR*HT(I, J)*(RO(I, J, K-1)+RO(I, J, K))*DZ/2.0
12     8 CONTINUE
13     10 CONTINUE
14     RETURN
15     END

```

ORIGINAL PAGE IS
OF POOR QUALITY

7.1.17 TEMP

This subroutine calculates the temperature distribution $T(I,J,K)$ at x, y, σ at each time step using a forward differencing in time. $T(x, y, \sigma)$ at $t = \Delta t$ ($=T(I,J,K)$) is calculated from $T(x, y, \sigma)$ at $t=0$ ($=TN1(I,J,K)$). Thereafter, T at time level $n+1$ ($=TN1(I,J,K)$) is calculated from T at time level n ($=T(I,J,K)$). The spatial derivatives, once again, have been approximated by central differencing in the interior of the domain, and three point single sided differencing on the boundaries, except at $MAR=6$ and $MAR=8$. The numerical scheme is given in Volume I. Note, that the adiabatic approximation given in Volume I calculates the temperature at boundary points after the energy equation is solved at interior points and at $MAR=6$ and $MAR=8$.

```

1 C*****
2 C THIS SUBROUTINE CALCULATES THE TEMPERATURE DISTRIBUTION AT EACH X-Y
3 C LOCATION AND DEPTH IN THE DOMAIN AT EACH TIME STEP USING A
4 C FORWARD DIFFERENCING SCHEME IN TIME
5 C*****
6 SUBROUTINE TEMP (IN,JN,KN,T,U,V,W,DX,DY,DZ,DT,H,BH,BV,MAR,TN1,HN1,
7 CORAD,RR,RS,TA,RO)
8 DIMENSION T(IN,JN,KN),U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),H(IN,JN)
9 C,MAR(IN,JN),TN1(IN,JN,KN),HN1(IN,JN)
10 C,RO(IN,JN,KN)
11 DO 10 I=1,IN
12 DO 10 J=1,JN
13 IF(MAR(I,J).EQ.0) GO TO 10
14 DO 8 K=2,KN
15 IF(MAR(I,J).EQ.11) GO TO 11
16 IF(MAR(I,J).EQ.6) GO TO 11
17 IF(MAR(I,J).EQ.8) GO TO 11
18 IF(MAR(I,J).LT.11) GO TO 10
19 11 CONTINUE
20 DHX=(H(I+1,J)-H(I-1,J))/(2.*DX)
21 DHY=(H(I,J+1)-H(I,J-1))/(2.*DY)
22 D1TX=(T(I+1,J,K)-T(I-1,J,K))/(2.*DX)
23 D1TY=(T(I,J+1,K)-T(I,J-1,K))/(2.*DY)
24 D2TX=(T(I+1,J,K)+T(I-1,J,K)-2.*T(I,J,K))/(DX*DX)
25 D2TY=(T(I,J+1,K)+T(I,J-1,K)-2.*T(I,J,K))/(DY*DY)
26 D1HUTX=(H(I+1,J)*U(I+1,J,K)-T(I+1,J,K)-H(I-1,J)*U(I-1,J,K)*
27 CT(I-1,J,K))/(2.*DX)
28 D1HVTY=(H(I,J+1)*V(I,J+1,K)+T(I,J+1,K)-H(I,J-1)*V(I,J-1,K)*
29 CT(I,J-1,K))/(2.*DY)
30 IF(K.EQ.5) GO TO 71
31 D1WTZ=(W(I,J,K+1)+T(I,J,K+1)-W(I,J,K-1)+T(I,J,K-1))/(2.*DZ)
32 D2TZ=(T(I,J,K+1)+T(I,J,K-1)-2.*T(I,J,K))/(DZ*DZ)
33 GO TO 80
34 71 D1WTZ=(3*T(I,J,K)*W(I,J,K)+T(I,J,K-2)*W(I,J,K-2)-4*T(I,J,K-1)*
35 CW(I,J,K-1))/(2.*DZ)
36 D2TZ=2.*(T(I,J,K-1)-T(I,J,K))/(DZ*DZ)
37 80 CONTINUE
38 TC=(D1HUTX+D1HVTY+H(I,J)*D1WTZ)
39 TKX=BH*(DHX*D1TX+H(I,J)*D2TX)
40 TKY=BH*(DHY*D1TY+H(I,J)*D2TY)
41 TKZ=BV*(D2TZ/H(I,J))
42 TN1(I,J,K)=((-TC+TKX+TKY+TKZ)*DT+H(I,J)*T(I,J,K))/HN1(I,J)
43 8 CONTINUE
44 10 CONTINUE
45 DO 1000 I=1,IN
46 DO 1000 J=1,JN
47 IF(MAR(I,J).EQ.5) GO TO 1000
48 IF(MAR(I,J).EQ.6) GO TO 1000
49 IF(MAR(I,J).EQ.7) GO TO 1000
50 IF(MAR(I,J).EQ.8) GO TO 1000
51 IF(MAR(I,J).EQ.9) GO TO 1000
52 IF(MAR(I,J).EQ.10) GO TO 1000
53 IF(MAR(I,J).EQ.11) GO TO 1000
54 DO 1001 K=2,KN
55 IF(MAR(I,J).EQ.0) TN1(I,J,K)=0.0
56 IF(MAR(I,J).EQ.1) TN1(I,J,K)=TN1(I,J-1,K)

```



```

57          IF(MAR(I,J).EQ.2) TN1(I,J,K)=TN1(I,J+1,K)
58          IF(MAR(I,J).EQ.3) TN1(I,J,K)=TN1(I+1,J,K)
59          IF(MAR(I,J).EQ.4) TN1(I,J,K)=TN1(I-1,J,K)
60      1001  CONTINUE
61      1000  CONTINUE
62          DO 2003 I=1,IN
63          DO 2000 J=1,JN
64          IF(MAR(I,J).EQ.0) GO TO 2000
65          IF(MAR(I,J).EQ.1) GO TO 2000
66          IF(MAR(I,J).EQ.2) GO TO 2000
67          IF(MAR(I,J).EQ.3) GO TO 2000
68          IF(MAR(I,J).EQ.4) GO TO 2000
69          IF(MAR(I,J).EQ.6) GO TO 2000
70          IF(MAR(I,J).EQ.8) GO TO 2000
71          IF(MAR(I,J).EQ.11) GO TO 2000
72          DO 2001 K=2,KN
73          IF(MAR(I,J).EQ.5) TN1(I,J,K)=(TN1(I+1,J,K)+TN1(I,J-1,K))/2.
74          IF(MAR(I,J).EQ.7) TN1(I,J,K)=(TN1(I+1,J,K)+TN1(I,J+1,K))/2.
75          IF(MAR(I,J).EQ.9) TN1(I,J,K)=(TN1(I-1,J,K)+TN1(I,J+1,K))/2.
76          IF(MAR(I,J).EQ.10) TN1(I,J,K)=(TN1(I-1,J,K)+TN1(I,J-1,K))/2.
77      2001  CONTINUE
78      2000  CONTINUE
79          DO 100 I=1,IN
80          DO 100 J=1,JN
81          IF(MAR(I,J).EQ.0) TN1(I,J,1)=0.0
82          RO(I,J,1)=1.029431-.000020*T(I,J,1)-.0000048*(T(I,J,1)**2)
83          RR=RO(I,J,1)
84          TERM=(DZ*HS*HN1(I,J))/(RR*BV)
85          TN1(I,J,1)=(TN1(I,J,2)+TA*TERM)/(1.0+TERM)
86      100   CONTINUE
87          RETURN
88          END

```

PRECEDING PAGE CONTAINS

7.1.18 OLDHT

This subroutine transforms $H(x,y,\sigma)$ at time level $n+1$ ($=HTE(I,J)$) into H at time level n ($=HTD(I,J)$) and H at time level n is transformed to H at time level $n-1$ ($=HT(I,J)$). These transformations are performed at the end of each time step.

```
1 C
2 C*****
3 C THIS SUBROUTINE TRANSFORMS MATRIX HTE INTO HT FOR THE NEXT TIME CYCL
4 C CALCULATION OF THE TOTAL DEPTH AT EACH X-Y LOCATION IN THE DOMAIN
5 C*****
6 SUBROUTINE OLDHT(IN, JN, HTE, HTD, HT)
7 DIMENSION HTD(IN, JN), HTE(IN, JN), HT(IN, JN)
8 DO 10 I=1, IN
9 DO 10 J=1, JN
10 HT(I, J)=HTE(I, J)
11 HTD(I, J)=HTE(I, J)
12 CONTINUE
13 RETURN
14 END
```

NO. 1000
QUALITY

7.1.19 OLDUV

This subroutine transforms u and v at time level $n=1$ ($= D(I,J,K)$ and $E(I,J,K)$) into u and v at time level $n(=H(I,J,K)$ and $G(I,J,K))$, and u and v at time level n is transformed into u and v at time level $n-1$ ($=U(I,J,K)$ and $V(I,J,K)$). These transformations are performed at the end of each time cycle.

```
1 C
2 C*****
3 C THIS SUBROUTINE TRANSFORMS MATRICES D,E INTO U,V, RESPECTIVELY,
4 C FOR THE NEXT TIME CYCLE CALCULATION OF HORIZONTAL VELOCITIES AT
5 C EACH X-Y LOCATION AND DEPTH IN THE DOMAIN
6 C*****
7 SUBROUTINE OLDUV(IN,JN,KN,U,V,H,G,D,E)
8 DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
9 CH(IN,JN,KN),G(IN,JN,KN)
10 DO 10 K=1,KN
11 DO 10 I=1,IN
12 DO 10 J=1,JN
13 U(I,J,K)=H(I,J,K)
14 V(I,J,K)=G(I,J,K)
15 H(I,J,K)=D(I,J,K)
16 G(I,J,K)=E(I,J,K)
17 10 CONTINUE
18 RETURN
19 END
```

7.1.20 OLDT

This subroutine transforms T at time level n+1 (=TN1(I,J,K)) into T at time level n(=T(I,J,K)). This transformation is performed at the end of each time cycle.

```
1 C*****
2 C   THIS SUBROUTINE TRANSFORMS MATRIX TN1 INTO T FOR THE NEXT TIME CYCLE
3 C   CALCULATION OF TEMPERATURE AT EACH X-Y LOCATION AND DEPTH IN THE DOMA
4 C*****
5     SUBROUTINE OLDT(IN,JN,KN,T,TN1)
6     DIMENSION T(IN,JN,KN),TN1(IN,JN,KN)
7     DO 10 K=1,KN
8     DO 10 I=1,IN
9     DO 10 J=1,JN
10    T(I,J,K)=TN1(I,J,K)
11    10 CONTINUE
12    RETURN
13    END
```

7.1.21 ETT

This subroutine calculates the wave height, $\eta(x,y)$
(=ETA(I,J)) at the end of each time step.

$$\text{ETA(I,J)} = \text{HT(I,J)} - \text{HI(I,J)}$$


```
1 C
2 C*****
3 C THIS SUBROUTINE CALCULATES THE WAVE HEIGHT(SURFACE ELEVATION ABOVE
4 C MEAN WATER LEVEL) AT EACH X-Y LOCATION IN THE DOMAIN AT EACH TIME SL
5 C*****
6 SUBROUTINE ETT(IN,JN,HT,HI,PAR,ETA)
7 DIMENSION HT(IN,JN),HI (IN,JN),PAR(IN,JN),ETA(IN,JN)
8 DO 10 I=1,IN
9 DO 10 J=1,JN
10 IF(PAR(I,J).EQ.0) GO TO 10
11 ETA(I,J)=HT(I,J)-HI(I,J)
12 10 CONTINUE
13 RETURN
14 END
```

7.1.22 PRPARA

This subroutine writes out the physical and numerical parameters at the end of each computer run, i.e. after the last time cycle for a particular computer run.

```

1 C
2 C*****
3 C   THIS SUBROUTINE PRINTS THE PHYSICAL AND NUMERICAL PARAMETERS FOR THE
4 C   VARIABLE DENSITY MODEL AT THE END OF EACH COMPUTER RUN
5 C*****
6   SUBROUTINE PRPARA(CI,CH,CV,CP,CC,DX,DY,DZ,DT,TAUX,TAUY,TTOT,GR,FF,
7   CRR,KH,KV,BH,BV,GRAD,TI,TTOT1,TA,HS)
8   IF(TTOT1.GT.D) TTOT1=TTOT1-DT
9   PRINT 1,CI,CH,CV,CC,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,GR,FF,RR,KH,KV
10  C,BH,BV,GRAD,TI,TTOT1,TA,HS
11  1  FORMAT(/' CI=',E15.7,/' CH=',E15.7,/' CV=',E15.7,/' CP=',E15.7,
12  C/' CC=',E15.7,/' DX=',E15.7,/' DY=',E15.7,/' DZ=',E15.7,/' DT=',
13  CE15.7,/' TAUX=',E15.7,/' TAUY=',E15.7,/' TTOT=',E15.7,/' GR=',
14  CE15.7,/' FF=',E15.7,/' RR=',E15.7,/' KH=',E15.7,/' KV=',E15.7,/'
15  C/' BH=',E15.7,/' BV=',E15.7,/' GRAD=',E15.7,/' TI=',E15.7,/'
16  C' TTOT1=',E15.7,/' TA=',E15.7,/' HS=',E15.7,/)
17  TTOT1=TTOT1+DT
18  RETURN
19  END

```

7.1.23 PRETA

This subroutine writes out $ETA(I,J)$ at the end of the computer run.

```
1 C
2 C.....
3 C THIS SUBROUTINE PRINTS THE WAVE HEIGHT AT EACH X-Y LOCATION IN THE
4 C DOMAIN AT THE END OF EACH COMPUTER RUN
5 C SUBROUTINE PRETA(I,J,IN,JN,ETA)
6 C DIMENSION ETA(IN,JN)
7 C DO 10 I=1,IN
8 10 PRINT 11,I,(ETA(I,J),J=1,JN)
9 11 FORMAT(/ ' I =',I3/' WAVE-HEIGHT'/(5X,8E15.7))
10 RETURN
11 END
```

7.1.24 PRUV

This subroutine writes out $U(I,J,K)$ and $V(I,J,K)$ at the end of the computer run.

```

1      C
2      C*****
3      C*****
4      C   THIS SUBROUTINE PRINTS THE HORIZONTAL VELOCITIES U,V AT EACH X-Y LOC
5      C   AND DEPTH IN THE DOMAIN AT THE END OF EACH COMPUTER RUN
6      C*****
7      SUBROUTINE PRUV (I,J,K,IN,JN,KN,U,V)
8      DIMENSION U(IN,JN,KN),V(IN,JN,KN)
9      KN1=KN-1
10     DO 10 K=1,KN1
11     DO 10 I=1,IN
12     PRINT 11,K,I,(U(I,J,K),J=1,JN)
13     PRINT 12,(V(I,J,K),J=1,JN)
14     11  FORMAT(/' K=' ,I3,3X,' I=' ,I3,/' U-VELOCITY'/(5X,8E15.7))
15     12  FORMAT(' V-VELOCITY'/(5X,8E15.7))
16     RETURN
17     END

```

7.1.25 PRW

This subroutine writes out $WZ(I,J,K)$ at the end of the computer run.


```
1 C*****
2 C   THIS SUBROUTINE PRINTS THE ACTUAL VERTICAL VELOCITY AT EACH X-Y
3 C   LOCATION AND DEPTH IN THE DOMAIN AT THE END OF EACH COMPUTER RUN
4 C*****
5   SUBROUTINE PRW(IN,JN,KN,W)
6   DIMENSION W(IN,JN,KN)
7   KN1=KN-1
8   DO 10 K=1,KN1
9   DO 10 I=1,IN
10  PRINT 11,K,I,(W(I,J,K),J=1,JN)
11  FORMAT(/ ' K=',I3,3X, ' I =',I3,/' ' W -VELOCITY'/(5X,8E15.7))
12  RETURN
13  END
```

7.1.25 PRTEMP

This subroutine writes out $T(I,J,K)$ at the end of the computer run.

```
1 C*****
2 C THIS SUBROUTINE PRINTS THE TEMPERATURE AT EACH X-Y LOCATION AND DEPTH
3 C IN THE DOMAIN AT THE END OF EACH COMPUTER RUN
4 C*****
5 SUBROUTINE PRTEMP(I,J,K,IN,JN,KN,T)
6 DIMENSION T(IN,JN,KN)
7 DO 10 K=1,KN
8 DO 10 I=1,IN
9 PRINT 11,K,I,(T(I,J,K),J=1,JN)
10 FORMAT(/ ' K=',I3,3X, ' I=',I3, / ' TEMPERATURE' / (5X,8E15.7))
11 CONTINUE
12 RETURN
13 END
```

7.1.27 STORE

This subroutine writes on magnetic tape all calculated system variables and numerical parameters, DX, DY, DZ, DT, TTOT and TTOT1.

```

1      C
2      C*****
3      C   THIS SUBROUTINE WRITES ON MAGNETIC TAPE, FOR THE VARIABLE DENSITY
4      C   MODEL, THE VALUES FOR THE VARIABLES AND PHYSICAL AND NUMERICAL
5      C   PARAMETERS FOR STORAGE AND FOR READING IN DATA FOR THE NEXT COMPUTER
6      C*****
7      SUBROUTINE STORE (IN, JN, KN, U, V, W, HI, HT, HTD, HX, HY, MAR, ETA, P, RO, CI,
8      CCC, CH, CV, CP, DX, DY, DZ, DT, TAUX, TAUY, TTOT, H, G, HTE, T, TTOT1, WZ)
9      DIMENSION U(IN, JN, KN), V(IN, JN, KN), W(IN, JN, KN), P(IN, JN, KN),
10     CHI(IN, JN), HT(IN, JN), HTD(IN, JN), HX(IN, JN), HY(IN, JN), MAR(IN, JN),
11     CETA(IN, JN), RO(IN, JN, KN), H(IN, JN, KN), G(IN, JN, KN), HTE(IN, JN)
12     C, T(IN, JN, KN), WZ(IN, JN, KN)
13     WRITE (8) ((U(I, J, K), K=1, KN), J=1, JN), I=1, IN),
14     C(((V(I, J, K), K=1, KN), J=1, JN), I=1, IN),
15     C(((W(I, J, K), K=1, KN), J=1, JN), I=1, IN),
16     C(((HI(I, J, K), K=1, KN), J=1, JN), I=1, IN),
17     C(((G(I, J, K), K=1, KN), J=1, JN), I=1, IN),
18     C(((P(I, J, K), K=1, KN), J=1, JN), I=1, IN),
19     C(((RO(I, J, K), K=1, KN), J=1, JN), I=1, IN),
20     C((HTD(I, J), J=1, JN), I=1, IN),
21     C((HTE(I, J), J=1, JN), I=1, IN),
22     C((HI(I, J), J=1, JN), I=1, IN),
23     C((HX(I, J), J=1, JN), I=1, IN),
24     C((HY(I, J), J=1, JN), I=1, IN),
25     C((MAR(I, J), J=1, JN), I=1, IN),
26     C((HT(I, J), J=1, JN), I=1, IN),
27     C((ETA(I, J), J=1, JN), I=1, IN),
28     C(((T(I, J, K), K=1, KN), J=1, JN), I=1, IN),
29     C(((WZ(I, J, K), K=1, KN), J=1, JN), I=1, IN),
30     CCI, CC, CH, CV, CP, DX, DY, DZ, DT, TAUX, TAUY, TTOT, TTOT1
31     END FILE 8
32     RETURN
33     END

```

7.2 MAIN PROGRAM FOR NASUM III

7.2.1. TMAIN3

This is the main program for free-surface complete field model. This program reads in the data, initializes the necessary quantities, co-ordinates the subroutines and calculates the velocity and temperatures in the whole domain under consideration. The parameter statement defines the size of the computational domain. The subroutine "XYSH" does horizontal stretching. The subroutine "READ 2" reads the MAR matrices which distinguishes the various points in the domain. The subroutine "INITIB" sets the initial conditions on velocities, temperatures, surface height and reads the depths at various points in the domain. The subroutine "CURNT" which is called after "INITIB" sets the velocities everywhere in the domain equal to the current velocity. The subroutine "INLET" puts the discharge velocity and temperature at the discharge location. Then it follows a set of subroutines to calculate the velocity and temperature field for the entire domain. The values of variable at different time levels are given in the Table (1).

```

1     PARAMETER IN=20,JN=20,KN=5
2     REAL KH,KV,RH,BV
3     DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),UM(IN,JN,KN),
4     CHT(IN,JN),HI(IN,JN),ETA(IN,JN),HTE(IN,JN),VM(IN,JN,KN),
5     CXX(IN),YY(JN),XXX(IN),YYY(JN),X(IN),Y(JN),A(IN),B(JN),
6     CH(IN,JN,KN),G(IN,JN,KN),O(IN,JN,KN),E(IN,JN,KN),WH(IN,JN,KN),
7     CMAR(IN,JN),HTD(IN,JN),PO(IN,JN,KN),P(IN,JN,KN)
8     C,T(IN,JN,KN),TN(IN,JN,KN),TF(IN,JN,KN),TAM(IN,JN,KN)
9     READ 1,IRUN
10    READ 1,LN
11    READ 1,KSTORF
12    1   FORMAT (I5)
13    READ 2, CI,CC,CP,CH,CV
14    READ 2, GP,FF,RR,HK
15    READ 2, DX,DY,DZ
16    READ 2, KH,KV,RH,BV
17    READ 2,DELX,DELY,DEEX,DEEY,EEEX,EEEY
18    CALL XYSH(IN,JN,DELX,DELY,DEEX,DEEY,EEEX,EEEY,XX,YY,XXX,YYY,A,B,
19    CX,Y)
20    2   FORMAT( )
21    IF(IRUN.GT.0) GO TO 4
22    CALL READ2(IN,JN,MAR)
23    CALL INITIB(IN,JN,KN,ETA,HT,HI,U,V,RO,P,GR,PR,DZ,HTD,D,E,H,G,
24    CHT,T,TN,TF,TAM,UM,VM,W,DX,DY)
25    CALL CUPNT(I,J,K,IN,JN,KN,U,V,H,G,D,E)
26    CALL INLET(I,J,K,IN,JN,KN,WH,T,TN,TF)
27    CALL WHTON(IN,JN,KN,HTE,HTD,HTD,ETA,H,G,W,WH,DX,DY,DZ,DT,
28    CMAR,XX,YY)
29    TTOT=0.0
30    GO TO 6
31    4   CONTINUE
32    CALL READ1(IN,JN,KN,U,V,W,HI,HT,HTD,MAR,ETA,P,RO,CI,UM,VM,
33    CCC,CH,CV,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,H,G,HTE,T,TN,TF,TAM,TAIR,D,
34    CE)
35    6   CONTINUE
36    READ 2,TAUX,TAUY
37    READ 2,TAIR
38    READ 2,DT
39    DO 5 L=1,LN
40    TTOT=TTOT+DT
41    CALL UVEL3(IN,JN,KN,U,V,H,G,D,E,DX,DY,DZ,W,TAUX,TAUY,DT,
42    CHTD,HTD,HTE,HX,HY,ETA,P,MAR,KH,KV,GR,RR,FF,
43    CCP,CC,CI,CH,CV,RO,IN,XX,YY,XXX,YYY)
44    CALL WVEL(IN,JN,KN,H,G,W,HTD,DX,DY,DZ,MAR,XX,YY,WH)
45    CALL TEMS(IN,JN,KN,HTD,HTD,HTE,DX,DY,DZ,DT,RH,BV,T,TN,TF,
46    CW,H,G,MAR,HK,TAIR,TAM,RO,XX,YY,XXX,YYY,L,LN)
47    CALL BOUND2(I,J,K,IN,JN,KN,U,V,H,G,D,E,W,HI,HTE,T,TN,TF)
48    CALL DENSITY(IN,JN,KN,RO,TF)
49    CALL PRES(IN,JN,KN,HTE,RO,GR,P,DZ)
50    CALL ETT(IN,JN,HTE,HI,MAR,ETA)
51    CALL OLDR(IN,JN,HTE,HTD,HT)
52    CALL CLOUDT(IN,JN,KN,U,V,H,G,O,F,T,TN,TF)
53    CALL INLET(I,J,K,IN,JN,KN,WH,T,TN,TF)
54    CALL MIXT(I,J,K,IN,JN,KN,TN)
55    CALL INLET(I,J,K,IN,JN,KN,WH,T,TN,TF)
56    CALL TIDE(I,J,K,IN,JN,KN,U,V,H,G,D,E,T,TN,TF)

```



```

57          5  CONTINUE
58             IF(KSTORE.CT.C)GO TO 1000
59             CALL STORE(IN,JN,KN,U,V,W,HI,HT,HTD,HAR,ETA,P,RO,CI,UM,VM,CC,CH,
60             CCV,CP,DX,DY,CZ,DT,TAUX,TAUY,TTOT,M,G,HTE,T,YN,TF,TAM,TAIR,D,E)
61          1000 CONTINUE
62             CALL PNPAPA(CT,CH,CV,CP,CC,DX,DY,DZ,DT,TAUX,TAUY,TTOT,GR,FF,RR,
63             CMH,KV,PH,BV,TAIR)
64             CALL PRFA(T,J,IN,JN,ETA)
65             CALL PRVVI(J,K,IN,JN,KN,M,G)
66             CALL PRK(IN,JN,KN,W)
67             CALL PRTEM(IN,JN,KN,TN)
68             CALL PRINTP(IN,JN,KN,P)
69             CALL PRINTH(IN,JN,HI)
70             STOP
71             END

```

PRECEDING PAGE BLANK NOT FILMED

SUBROUTINE PROGRAMS FOR NASUM III

PRECEDING PAGE BLANK NOT FILMED

7.1.2. ROUND2

This subroutine sets the boundary conditions for all variables. The boundary conditions for velocity are no slip and no normal velocity on the bottom and on the shore line (y-axis). For temperature, the bottom and shore line are treated as adiabatic. For open boundaries, the boundary conditions on velocity and temperature are $\frac{\partial v}{\partial n} = 0$ and $-\frac{\partial T}{\partial n} = 0$ respectively. The temperature boundary condition on the surface is specified in temperature subroutine "TEM5".

```

1      SUBROUTINE BOUND2(I,J,K,IN,JN,KN,U,V,H,G,D,E,W,HI,HTE,T,TN,TF)
2      DIMENSION U(IN,JN,KN),V(IN,JN,KN),H(IN,JN,KN),
3      CG(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),T(IN,JN,KN),
4      CTN(IN,JN,KN),TF(IN,JN,KN),W(IN,JN,KN),HI(IN,JN),HTE(IN,JN)
5      KN1=KN-1
6      IN1=IN-1
7      JN1=JN-1
8      DO 5 K=1,KN
9      DO 5 I=1,IN
10     C
11     C      FAR J BOUNDARY
12     C
13     T(I,JN,K)=T(I,JN1,K)
14     TN(I,JN,K)=TN(I,JN1,K)
15     TF(I,JN,K)=TF(I,JN1,K)
16     HTE(I,JN)=HTE(I,JN1)
17     D(I,JN,K)=D(I,JN1,K)
18     E(I,JN,K)=E(I,JN1,K)
19     W(I,JN,K)=W(I,JN1,K)
20     S
21     C
22     C      FAR I BOUNDARY
23     C
24     DO 10 K=1,KN
25     DO 10 J=1,JN
26     T(IN,J,K)=T(IN1,J,K)
27     TN(IN,J,K)=TN(IN1,J,K)
28     TF(IN,J,K)=TF(IN1,J,K)
29     HTE(IN,J)=HTE(IN1,J)
30     D(IN,J,K)=D(IN1,J,K)
31     E(IN,J,K)=E(IN1,J,K)
32     W(IN,J,K)=W(IN1,J,K)
33     IC
34     C
35     C      ALONG I BOUNDARY
36     C
37     DO 15 K=1,KN
38     DO 15 I=1,IN
39     T(I,1,K)=T(I,2,K)
40     TN(I,1,K)=TN(I,2,K)
41     TF(I,1,K)=TF(I,2,K)
42     HTE(I,1)=HTE(I,2)
43     D(I,1,K)=D(I,2,K)
44     E(I,1,K)=E(I,2,K)
45     W(I,1,K)=W(I,2,K)
46     15
47     C
48     C      ALONG SHORE LINE
49     C
50     DO 20 K=1,KN
51     DO 20 J=1,JN
52     T(1,J,K)=T(2,J,K)
53     TN(1,J,K)=TN(2,J,K)
54     TF(1,J,K)=TF(2,J,K)
55     HTE(1,J)=HI(1,J)
56     D(1,J,K)=D(2,J,K)

```

```
57      E(I,J,K)=E(2,J,K)
58      W(I,J,K)=W(2,J,K)
59      20  CONTINUE
60      C
61      C   BOTTOM BOUNDARY CONDITION
62      C
63      DO 25 I=1,IN
64          DO 25 J=1,JN
65              T(I,J,5)=T(I,J,4)
66              TN(I,J,5)=TN(I,J,4)
67              TF(I,J,5)=TF(I,J,4)
68      25  CONTINUE
69      RETURN
70      END
```

PRECEDING PAGE BLANK NOT FILMED

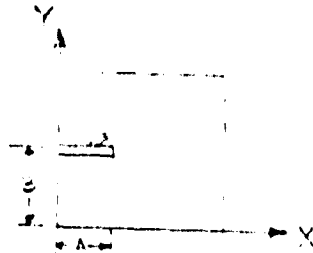
7.2.3. CONST

This is a small main program and need to be run in order to determine the constants DEEX, DEEY, EEEX, EEEY which are used in the subroutine "XYSH". The input cards reads

(1) XB, A, DX, AN

(2) YB, B, DY, BN

where XB, YB are X and Y boundary distances. A and B are distances from the shore line (y-axis) and x-axis respectively from where the stretching starts. It is shown below.



DX, DY are the minimum grid size needed. AN, BN are the number of grid points in the x and y directions.

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

```

E

```

50 READ 4,XB,A,DX,AN
4 FORMAT( )
IF(XB.LE.0.) GO TO 400
WRITE(6,11) XB,A,DX,AN
11 FORMAT('1','XBNDRY=',F10.1,' A=',F10.1,' DELTA X=',F10.1,'
INBR POINTS=',F10.1/)
E1=XB-A
E2=(AN-1.)*DX
DC=.01*A
18 C=0.
J=0
8 C=C+DC
J=J+1
D=C*ALOG(A/C+SQRT((A/C)**2+1.))
U=AMIN1((E2-D)/C,.85.)
X=A+C*SINH(U)
WRITE(6,42) C,X
IF(X.GT.XP) GO TO 8
IF(J.GT.1) GO TO 17
DC=DC/2.
GO TO 18
17 CMIN=C-DC
CMAX=C
33 CONTINUE

```

C

```

C11=CMAX
D=C11*ALOG(A/C11+SQRT((A/C11)**2+1.))
U=AMIN1((E2-D)/C11,.85.)
ERR1=-E1+C11*SINH(U)
J=1
C12=CMAX-DC/2.

```

C

```

1 J=J+1
IF(J.GT.30) GO TO 99
D=C12*ALOG(A/C12+SQRT((A/C12)**2+1.))
U=AMIN1((E2-D)/C12,.85.)
ERR2=-E1+C12*SINH(U)
WRITE(6,42) C12,D,U,ERR2
42 FORMAT(1X,4F15.7)
IF(ABS(ERR2/XP).LT..C01) GO TO 2
C13=(C11*ERR2-C12*ERR1)/(ERR2-ERR1)
C13=AMAX1(C13,CMIN)
C13=AMIN1(C13,CMAX)
C11=C12
C12=C13
ERR1=ERR2
GO TO 1

```

C

```

2 C1=C12
WRITE(6,6) C1,D
6 FORMAT(1H0,'C1.',F15.7,' D=',F15.7/)
DUM=0.
N=INT(AN)
DO 3 I=1,N
XL=(I-1)*DX

```

```
57      X=A+C1*SINH((XL-D)/C1)
58      DELTA=X-DUM
59      DUM=X
60      WRITE(6,5) I,XL,X,DELTA
61      3 CONTINUE
62      5 FORMAT(1X,'I=',I4,'   XL=',F10.2,'   X=',F10.2,'   DELTA=',F10.1)
63      GO TO 50
64      49 WRITE(6,14)
65      14 FORMAT(1X,'NBR ITERATIONS EXCEEDED 30')
66      GO TO 400
67      400 STOP
68      END
```

PRECEDING PAGE BLANK NOT FILMED

7.2.4. CURNT

This subroutine sets the velocity field in the whole domain equal to the current velocity. In this case 2 cm/sec is chosen. If the initial current is more, the values should be made equal to the measured value of current. If there is no initial current, this subroutine can be deleted from the main program.

```
1      SUBROUTINE CURNT(I,J,K,IN,JN,KN,U,V,H,G,D,E)
2      DIMENSION U(IN,JN,KN),V(IN,JN,KN)
3      C,H(IN,JN,KN),G(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN)
4      KN1=KN-1
5      DO 10 K=1,KN1
6      DO 10 I=1,IN
7      DO 10 J=1,JN
8      U(I,J,K)=0.0
9      V(I,J,K)=2.0
10     H(I,J,K)=0.0
11     G(I,J,K)=2.0
12     D(I,J,K)=0.0
13     E(I,J,K)=2.0
14     10 CONTINUE
15     RETURN
16     END
```

ORIGINAL PAGE IS
OF POOR QUALITY

7.2.5. DENSTY

This subroutine computes for the whole domain the density using the temperatures computed in the subroutine "TEM5".

```
1      SUBROUTINE DENSTY(IN,JN,KN,PO,T)
2      DIMENSION RO(IN,JN,KN),T(IN,JN,KN)
3      DO 10 I=1,IN
4      DO 10 J=1,JN
5      DO 10 K=1,KN
6      RO(I,J,K)=1.00042A-0.000019*(T(I,J,K))-0.0000046*
7      (T(I,J,K)*T(I,J,K))
8      10  CONTINUE
9      RETURN
10     END
```

ORIGINAL PAGE IS
OF POOR QUALITY

7.2.6. ETT

This subroutine calculates the wave height by subtracting mean level of water (HI) from the total height computed.

```
1      C      THIS PROGRAM COMPUTES THE WAVE-HEIGHT
2      C
3      SUBROUTINE ETT(IN,JN,HT,HI,MAP,FTP)
4      DIMENSION HT(IN,JN),HI(IN,JN),MAR(IN,JN),ETA(IN,JN)
5      DO 10 I=1,IN
6      DO 10 J=1,JN
7      ETA(I,J)=HT(I,J)-HI(I,J)
9      10      CONTINUE
9      RETURN
10     END
```

PRECEDING PAGE BLANK NOT FILMED

7.2.7. HITE2

This subroutine computes the free surface height using the equation

$$\frac{\partial H}{\partial t} = - \int_0^1 \left(X' \frac{\partial(HU)}{\partial X} + Y' \frac{\partial(HV)}{\partial Y} \right) \delta \sigma - (W_b - U_b X' \frac{\partial h}{\partial X} - V_b Y' \frac{\partial h}{\partial Y})$$

The numerical scheme used is forward difference in time and central difference in space (FTCS). Simpsons rule is used for numerical integration.

```

1      SUBROUTINE HITE2(IN,JN,KN,MAR,U,V,HT,HTD,HTE,DZ,DT,DX,DY,HDUM
2      C,XX,YY,WH)
3      DIMENSION MAR(IN,JN),U(IN,JN,KN),V(IN,JN,KN),HT(IN,JN),HTD(IN,JN),
4      HDUM(IN,JN),HTE(IN,JN),XX(IN),YY(JN),WH(IN,JN,KN)
5      KNM1=KN-1
6      IN1=IN-1
7      JN1=JN-1
8      DO 50 I=2,IN1
9      DO 50 J=2,JN1
10     HDUM(I,J)=0.0
11     DO 60 K=1,KN
12     IF(MAR(I,J).EQ.11) GO TO 11
13     GO TO 50
14     11  DIHUX=(HT(I+1,J)*U(I+1,J,K)-HT(I-1,J)*U(I-1,J,K))/(2.*DX)
15     DIHUY=(HT(I,J+1)*V(I,J+1,K)-HT(I,J-1)*V(I,J-1,K))/(2.*DY)
16     C.... SIMPSON'S RULE IS USED FOR INTEGRATION
17     IF(K.EQ.1.OR.K.EQ.5) GO TO 101
18     IF(K.EQ.2.OR.K.EQ.4) GO TO 102
19     HDUM(I,J)=((DIHUX*XX(I)+DIHUY*YY(J))*DZ*(2./3.))+HDUM(I,J)
20     GO TO 103
21     101 HDUM(I,J)=((DIHUX*XX(I)+DIHUY*YY(J))*DZ/3.))+HDUM(I,J)
22     GO TO 103
23     102 HDUM(I,J)=((DIHUX*XX(I)+DIHUY*YY(J))*DZ*(4./3.))+HDUM(I,J)
24     103 CONTINUE
25     HTE(I,J)=HTD(I,J)-HDUM(I,J)*DT
26     60  CONTINUE
27     HTE(I,J)=HTE(I,J)-WH(I,J,KN)*DT
28     50  CONTINUE
29     RETURN
30     END

```


7.2.8. INITIB

This subroutine reads the depths for a constant depth basin and initializes the values u, v, w, p, ρ, T and HI . The program sets u, v, w and wave height ETA equal to zero. The temperature is set equal to the ambient temperature. The pressure is hydrostatic.

```

1      SUBROUTINE INITID(IN,JN,KN,ETA,HT,HI,U,V,RO,P,GR,RR,DZ,HTD,D,
2      CE,H,G,HTE,T,TN,TF,TAM,UM,VM,W,DX,DY)
3      DIMENSION ETA(IN,JN),HT(IN,JN),HI(IN,JN),U(IN,JN,KN),V(IN,JN,KN),
4      CRO(IN,JN,KN),P(IN,JN,KN),HTD(IN,JN),W(IN,JN,KN)
5      C,D(IN,JN,KN),F(IN,JN,KN),H(IN,JN,KN),G(IN,JN,KN),HTE(IN,JN)
6      C,TAM(IN,JN,KN),T(IN,JN,KN),TN(IN,JN,KN),TF(IN,JN,KN)
7      C,UM(IN,JN,KN),VM(IN,JN,KN)
8      DO 500 J=1,JN
9      HI(1,J)=300.0
10     HI(2,J)=350.0
11     HI(3,J)=400.0
12     HI(4,J)=450.0
13     HI(5,J)=500.0
14     HI(6,J)=550.0
15     HI(7,J)=600.0
16     HI(8,J)=600.0
17     HI(9,J)=600.0
18     HI(10,J)=600.0
19     HI(11,J)=600.0
20     HI(12,J)=650.0
21     HI(13,J)=700.0
22     HI(14,J)=750.0
23     HI(15,J)=800.0
24     HI(16,J)=850.0
25     HI(17,J)=900.0
26     HI(18,J)=950.0
27     HI(19,J)=1000.0
28     HI(20,J)=1000.0
29     500 CONTINUE
30     DO 10 I=1,IN
31     DO 10 J=1,JN
32     ETA(I,J)=0.0
33     HT(I,J)=HI(I,J)
34     HTD(I,J)=HI(I,J)
35     HTE(I,J)=HI(I,J)
36     10 CONTINUE
37     DO 8 I=1,IN
38     DO 8 J=1,JN
39     DO 8 K=1,KN
40     UM(I,J,K)=0.0
41     VM(I,J,K)=0.0
42     W(I,J,K)=0.0
43     U(I,J,K)=UM(I,J,K)
44     V(I,J,K)=VM(I,J,K)
45     TAM(I,J,K)=25.0
46     T(I,J,K)=TAM(I,J,K)
47     P(I,J,K)=GR*HT(I,J)*RR*(K-1)*DZ
48     H(I,J,K)=U(I,J,K)
49     D(I,J,K)=U(I,J,K)
50     G(I,J,K)=V(I,J,K)
51     E(I,J,K)=V(I,J,K)
52     TN(I,J,K)=T(I,J,K)
53     TF(I,J,K)=T(I,J,K)
54     8 CONTINUE
55     RETURN
56     END

```

7.2.9. INLET

This subroutine puts the velocity and temperature of the discharge at the discharge location. The value of velocity specified in this subroutine should be calculated depending upon the mass of the discharge.

```
1      SUBROUTINE INLET(I,J,K,IN,JN,KN,WH,T,TN,TF)
2      DIMENSION T(IN,JN,KN),WH(IN,JN,KN)
3      C,TN(IN,JN,KN),TF(IN,JN,KN)
4      DO 10 I=8,10
5      DO 10 J=10,12
6      DO 10 K=1,KN
7      WH(I,J,5)=-0.35
8      T(I,J,K)=35.0
9      TN(I,J,K)=35.0
10     TF(I,J,K)=35.0
11     10 CONTINUE
12     RETURN
13     END
```

7.2.10. MIXT

This subroutine mixes the temperatures by an averaging process in such that unstable density gradients are eliminated.

7.2.11. OLDHT

This subroutine sets the values of height HTD at time level n to HT at time level n-1 and HTE at time level n+1 to HTD at time level n after all computation are completed. This is necessary in order to retain values of height at one time step lag.

```
1      C      THIS PROGRAM TRANSFERS MATRIX HTD TO HT
2      C
3      SUBROUTINE OLDHT(IN,JN,HTE,HTD,HT)
4      DIMENSION HTD(IN,JN),HTE(IN,JN),HT(IN,JN)
5      DO 10 I=1,IN
6      DO 10 J=1,JN
7      HT(I,J)=HTD(I,J)
8      HTD(I,J)=HTE(I,J)
9      10      CONTINUE
10     RETURN
11     END
```


7.2.12. OLDUVT

This subroutine sets the values of velocity and temperature at time level $n+1$ equal to the values at time level n and the values at time level n are made equal to the values at time level $n-1$. This is necessary in order to retain the values of velocity and temperature at the time step lag.

```
1      C      THIS PROGRAM TRANSFERS MATRICES U,V TO D,E RESPECTIVELY
2      C      AND T TO TN
3      C
4      SUBROUTINE OLDUVT(IN,JN,KN,U,V,H,G,D,F,T,TN,TF)
5      DIMENSION U(IN,JN,KN),V(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN),
6      CH(IN,JN,KN),G(I,J,K),T(IN,JN,KN),TN(IN,JN,KN),TF(IN,JN,KN)
7      DO 10 K=1,KN
8      DO 10 I=1,IN
9      DO 10 J=1,JN
10     U(I,J,K)=H(I,J,K)
11     V(I,J,K)=G(I,J,K)
12     H(I,J,K)=D(I,J,K)
13     G(I,J,K)=E(I,J,K)
14     T(I,J,K)=TN(I,J,K)
15     TN(I,J,K)=TF(I,J,K)
16     10 CONTINUE
17     RETURN
18     END
```

ORIGINAL PAGE IS
OF POOR QUALITY

7.2.13. PRPARA

This subroutine prints the input values and the total time the model is simulated.

```
1 C THIS SUBROUTINE PRINTS PARAMETERS FOR THE FREE SURFACE MODEL
2 C
3 SUBROUTINE PRPARA (CI,CH,CV,CP,CC,DX,DY,DZ,DT,TAUX,TAUY,TTOT,GR,FF,
4 CRR,KH,KV,RH,BV,TAIR)
5 PRINT 1,CI,CH,CV,CC,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,GR,FF,RR,KH,KV
6 C,BH,BV,TAIR
7 1 FORMAT(// CI=,F15.7,/' CH=,E15.7,/' CV=,E15.7/' CP=,E15.7,
8 C/' CC=,E15.7,/' DX=,E15.7,/' DY=,E15.7,/' DZ=,E15.7,/' DT=,
9 CE15.7,/' TAUX=,E15.7,/' TAUY=,E15.7,/' TTOT=,E15.7,/' GR=,
10 CE15.7,/' FF=,E15.7,/' RR=,E15.7,/' KH=, E15.7,/' KV=,E15.7,/'
11 C' BH=,E15.7,/' BV=,F15.7,/' TAIR=,F15.7/)
12 RETURN
13 END
```

ORIGINAL PAGE IS
OF POOR QUALITY

7.2.14. PRES

This subroutine calculates the pressure field using the updated density field.

```
1 C THIS PROGRAM CALCULATES THE PRESSURE FIELD
2 C
3 SUBROUTINE PRFS(IN,JN,KN,HT,RO,GR,P,DZ)
4 DIMENSION HT(IN,JN),HO(IN,JN,KN),P(IN,JN,KN)
5 DO 10 I=1,IN
6 DO 10 J=1,JN
7 P(I,J,1)=0.0
8 DO 8 K=2,KN
9 P(I,J,K)=P(I,J,K-1)*GR*HT(I,J)+(RO(I,J,K-1)+RO(I,J,K))*DZ/2.0
10 CONTINUE
11 10 CONTINUE
12 RETURN
13 END
```

7.2.15. PRETA

This subroutine prints wave height (ETA).

```
1 C THIS PROGRAM PRINTS THE WAVE HEIGHT
2 C
3 SUBROUTINE PRETA(I,J,IN,JN,ETA)
4 DIMENSION ETA(IN,JN)
5 DO 10 I=1,IN
6 PRINT 11,I,(ETA(I,J),J=1,JN)
7 11 FORMAT(/' I =',I3/' WAVE-HEIGHT'/(5X,8E15.7))
8 RETURN
9 END
```


7.2.16. PRTEM

This subroutine prints temperatures in the whole domain.

```
1 C THIS SUBROUTINE PRINTS THE TEMPERATURES
2 C
3 SUBROUTINE PRTEM(IN,JN,KN,T)
4 DIMENSION T(IN,JN,KN)
5 DO 40 K=1,KN
6 WRITE(6,105) K
7 105 FORMAT('1',* ' TEMPERATURE AT K =',I5,/)
8 DO 20 I=1,IN
9 20 WRITE(6,106) (T(I,J,K),J=1,JN)
10 106 FORMAT('7',20F6.2)
11 40 CONTINUE
12 RETURN
13 END
```

ORIGINAL PAGE IS
OF POOR QUALITY

7.2.17. PRUV

This subroutine prints u and v velocity in the whole domain.

```
1 C THIS PROGRAM PRINTS THE HORIZONTAL VELOCITIES
2 C
3 SUBROUTINE PPUV(I,J,K,IN,JN,KN,U,V)
4 DIMENSION U(IN,JN,KN),V(IN,JN,KN)
5 KN1=KN-1
6 DO 10 K=1,KN
7 DO 10 I=1,IN
8 PRINT 11,K,I,(U(I,J,K),J=1,JN)
9 PRINT 12,(V(I,J,K),J=1,JN)
10 FORMAT(/' K=',I3,3X,' I=',I3,/' U-VELOCITY'/(5X,8E15.7))
11 FORMAT(' V-VELOCITY'/(5X,8E15.7))
12 RETURN
13 END
```

7.2.18. PRW

This subroutine prints Ω values in the whole domain.

```
1      C      THIS SUBROUTINE PRINTS THE VERTICAL VELOCITIES
2      C
3      SUBROUTINE PRW(IN,JN,KN,W)
4      DIMENSION W(IN,JN,KN)
5      DO 10 K=1,KN
6      DO 10 I=1,IN
7      10     PRINT I,K,I,(W(I,J,K),J=1,JN)
8      11     FORMAT(/' K=' ,I3,3X,' I =' ,I3,/' W -VELOCITY'/(5X,8E15.7))
9      RETURN
10     END
```

7.2.19. PRWH

This subroutine prints w-velocity in the whole domain.

```
SUBROUTINE PRWH(IN,JN,KN,WH)
DIMENSION WH(IN,JN,KN)
DO 10 K=1,KN
DO 10 I=1,IN
10 PRINT 11,K,I,(WH(I,J,K),J=1,JN)
11 FORMAT(/' K=',I3,3X,' I =',I3,/' WH-VELOCITY'/(5X,8E15.7))
RETURN
END
```

THIS PAGE IS
PROPERTY

7.2.20. READ1

This subroutine is used to read the information stored by subroutine "STORE". This subroutine is used from second run onwards in order to use the values created in the first run (ie IRUN=0). "READ1" and "STORE" correspond to each other. The "READ 1" subroutine uses a file designated as "UNIT 7" in order to read the information on the tape.

```

1      C      THIS SUBROUTINE READS DATA FROM TAPE
2      C
3      SUBROUTINE READ1(IN,JN,KN,U,V,W,HI,HT,HTD,MAR,ETA,P,RO,CI,UM,VM,
4      CCC,CH,CV,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,H,G,HTE,T,TN,TF,TAM,TAIR,D,
5      CE)
6      DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),P(IN,JN,KN),
7      CHI(IN,JN),HT(IN,JN),HTD(IN,JN),MAR(IN,JN),
8      CETA(IN,JN),RO(IN,JN,KN),H(IN,JN,KN),G(IN,JN,KN),HTE(IN,JN),
9      CT(IN,JN,KN),TN(IN,JN,KN),TF(IN,JN,KN),TAM(IN,JN,KN)
10     C,UM(IN,JN,KN),VM(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN)
11     I      CONTINUE
12     READ(7,END=1) ((U(I,J,K),K=1,KN),J=1,JN),I=1,IN),
13     C((V(I,J,K),K=1,KN),J=1,JN),I=1,IN),
14     C((W(I,J,K),K=1,KN),J=1,JN),I=1,IN),
15     C((H(I,J,K),K=1,KN),J=1,JN),I=1,IN),
16     C((G(I,J,K),K=1,KN),J=1,JN),I=1,IN),
17     C((D(I,J,K),K=1,KN),J=1,JN),I=1,IN),
18     C((E(I,J,K),K=1,KN),J=1,JN),I=1,IN),
19     C((P(I,J,K),K=1,KN),J=1,JN),I=1,IN),
20     C((RO(I,J,K),K=1,KN),J=1,JN),I=1,IN),
21     C((UM(I,J,K),K=1,KN),J=1,JN),I=1,IN),
22     C((VM(I,J,K),K=1,KN),J=1,JN),I=1,IN),
23     C((HTD(I,J),J=1,JN),I=1,IN),((HTE(I,J),J=1,JN),I=1,IN),
24     C((HI(I,J),J=1,JN),I=1,IN),((MAR(I,J),J=1,JN),I=1,IN),
25     C((HT(I,J),J=1,JN),I=1,IN),((ETA(I,J),J=1,JN),I=1,IN),
26     C((T(I,J,K),K=1,KN),J=1,JN),I=1,IN),
27     C((TN(I,J,K),K=1,KN),J=1,JN),I=1,IN),
28     C((TF(I,J,K),K=1,KN),J=1,JN),I=1,IN),
29     C((TAM(I,J,K),K=1,KN),J=1,JN),I=1,IN),
30     CCI,CC,CH,CV,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,TAIR
31     RETURN
32     END

```

7.2.21. READ2

This subroutine reads the "MAR" matrix. The MAR numbering system is used in order to identify the points in the interior, on the boundaries and outside the domain.

The "MAR" numbering used is as follows,

MAR (I,J) = 0 for points outside the domain.

MAR (I,J) = 1 for upper horizontal boundary.

MAR (I,J) = 2 for lower horizontal boundary.

MAR (I,J) = 3 for left vertical boundary.

MAR (I,J) = 4 for right vertical boundary.

MAR (I,J) = 5 through MAR (I,J) = 10 are boundary corners and are specified as below.

MAR (I,J) = 11 for points in the interior.

MAR-5



MAR-6



MAR-7



MAR-8



MAR-9



MAR-10



```
1 C THIS SUBROUTINE DEFINES THE MAR MATRIX FOR LOCATING THE POSITIONS
2 C
3 SUBROUTINE READ2(IN,JN,MAR)
4 DIMENSION MAR(IN,JN)
5 MAR(1,1)=7
6 MAR(1,JN)=5
7 MAR(IN,1)=9
8 MAR(IN,JN)=10
9 INM1=IN-1
10 JNM1=JN-1
11 DO 10 I=2,INM1
12 MAR(I,1)=2
13 10 MAR(I,JN)=1
14 DO 20 J=2,JNM1
15 MAR(1,J)=3
16 20 MAR(IN,J)=4
17 DO 30 I=2,INM1
18 DO 30 J=2,JNM1
19 30 MAR(I,J)=11
20 RETURN
21 END
```

7.2.22. STORE

This subroutine is used to store the values at the end of all computations on a file designated as "UNIT 8".

```

1      C      THIS SUBROUTINE STORES THE DATA INTO TAPE
2      C
3      SUBROUTINE STORE (IN,JN,KN,U,V,W,HI,HT,HTD,MAR,ETA,P,RO,CI,UM,VM,
4      CCC,CH,CV,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,H,G,HTE,T,IN,TF,TAM,TAIR,
5      C0,E)
6      DIMENSION U(IN,JN,KN),V(IN,JN,KN),W(IN,JN,KN),P(IN,JN,KN),
7      CHT(IN,JN),HI(IN,JN),HTD(IN,JN),MAR(IN,JN),D(IN,JN,KN),E(IN,JN,KN),
8      CETA(IN,JN),RO(IN,JN,KN),H(IN,JN,KN),G(IN,JN,KN),HTE(IN,JN),
9      CT(IN,JN,KN),TN(IN,JN,KN),TF(IN,JN,KN),TAM(IN,JN,KN)
10     C,UM(IN,JN,KN),VM(IN,JN,KN)
11     WRITE (8)((U(I,J,K),K=1,KN),J=1,JN),I=1,IN),
12     C((V(I,J,K),K=1,KN),J=1,JN),I=1,IN),
13     C((W(I,J,K),K=1,KN),J=1,JN),I=1,IN),
14     C((HT(I,J),K=1,KN),J=1,JN),I=1,IN),
15     C((G(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((P(I,J,K),K=1,KN),J=1,JN),I=1,IN),
19     C((RO(I,J,K),K=1,KN),J=1,JN),I=1,IN),
20     C((UM(I,J,K),K=1,KN),J=1,JN),I=1,IN),
21     C((VM(I,J,K),K=1,KN),J=1,JN),I=1,IN),
22     C((HTD(I,J),J=1,JN),I=1,IN),((HTE(I,J),J=1,JN),I=1,IN),
23     C((HI(I,J),J=1,JN),I=1,IN),((MAR(I,J),J=1,JN),I=1,IN),
24     C((HT(I,J),J=1,JN),I=1,IN),((ETA(I,J),J=1,JN),I=1,IN),
25     C((T(I,J,K),K=1,KN),J=1,JN),I=1,IN),
26     C((TN(I,J,K),K=1,KN),J=1,JN),I=1,IN),
27     C((TF(I,J,K),K=1,KN),J=1,JN),I=1,IN),
28     C((TAM(I,J,K),K=1,KN),J=1,JN),I=1,IN),
29     CCI,CC,CH,CV,CP,DX,DY,DZ,DT,TAUX,TAUY,TTOT,TAIR
30     END FILE 8
31     END FILE 8
32     RETURN
33     END

```

ORIGINAL PAGE IS
OF POOR QUALITY

7.2.23. TEM5

This subroutine computes temperature only in the interior of the domain. The schemes used are forward in time and central in space (F.T.C.S.). Vertical diffusion term is treated by DuFort-Frankel scheme.

```

1      SUBROUTINE TFMS(IN,JN,N,HT,HTD,HTE,DX,DY,DZ,DT,RH,BV,T,TN,TF,
2      C,H,G,MAR,HK,TAIR,TAM,PO,XX,YY,XXY,YYY,L,LN)
3      DIMENSION HT(IN,JN),HTD(IN,JN),HTE(IN,JN),MAR(IN,JN),
4      CTAM(IN,JN,KN),C(IN,JN,KN),H(IN,JN,KN),G(IN,JN,KN),RO(IN,JN,KN),
5      CXX(IN),YY(JN),XXY(IN),YYY(JN),
6      CT(IN,JN,KN),TN(IN,JN,KN),TF(IN,JN,KN)
7      KN1=KN-1
8      IN1=IN-1
9      JN1=JN-1
10     DO 10 K=1,KN1
11     DO 10 I=2,IN1
12     DO 10 J=2,JN1
13     IF(MAR(I,J).EQ.11) GO TO 11
14     GO TO 300
15     11 CONTINUE
16     DHUTX=(HTD(I+1,J)*H(I+1,J,K)*TN(I+1,J,K)-HTD(I-1,J)
17     C*H(I-1,J,K)*TN(I-1,J,K))/(2*DX)
18     DHVTY=(HTD(I,J+1)*C(I,J+1,K)*TN(I,J+1,K)-
19     CHD(I,J-1)*C(I,J-1,K)*TN(I,J-1,K))/(2*DY)
20     DHT=(HT(I,J+1)-HT(I,J-1))/(2*DY)
21     DITY=(TN(I,J+1,K)-TN(I,J-1,K))/(2*DY)
22     D2TY=(TN(I,J+1,K)+TN(I,J-1,K)-2*TN(I,J,K))/(DY*DY)
23     DHX=(HT(I+1,J)-HT(I-1,J))/(2*DX)
24     DITY=(TN(I+1,J,K)-TN(I-1,J,K))/(2*DX)
25     D2TX=(TN(I+1,J,K)+TN(I-1,J,K)-2*TN(I,J,K))/(DX*DX)
26     IF(K.EQ.1) GO TO 50
27     D1WTZ=(W(I,J,K+1)*TN(I,J,K+1)-W(I,J,K-1)*TN(I,J,K-1))/(2*DZ)
28     D1TZ=(TN(I,J,K+1)-TN(I,J,K-1))/(2*DZ)
29     D2TZ=(TN(I,J,K+1)+TN(I,J,K-1)-T(I,J,K))/(DZ*DZ)
30     GO TO 200
31     50 D1WTZ=(4*W(I,J,K+1)*TN(I,J,K+1)-3*W(I,J,K)*TN(I,J,K)-
32     W(I,J,K-1)*TN(I,J,K-1))/(2*DZ)
33     D1TZ=HT(I,J)*HK*(TN(I,J,1)-TAIR)
34     D2TZ=(2*TN(I,J,K+1)-T(I,J,K))/(DZ*DZ)-2*D1TZ/DZ
35     200 CONTINUE
36     DHT=(HTE(I,J)-HTD(I,J))/(DT)
37     TLX=DHUTX*XX(I)
38     TLY=DHVTY*YY(J)
39     TLZ=HTD(I,J)*D1WTZ
40     TLC=((K-1)*DZ-1)*DHT*D1TZ
41     TIT=TLX+TLY+TLZ+TLC
42     TRX=RH*(DHX*XX(I)*D1TX*XX(I)+HT(I,J)*XX(I)*XX(I)+D2TX*HT(I,J)*
43     CXX(I)*D1TX)
44     TRY=RH*(DHY*YY(J)*D1TY*YY(J)+HT(I,J)*YY(J)*YY(J)+D2TY*HT(I,J)*
45     CYYY(J)*D1TY)
46     GO TO 500
47     500 TR=TRX+TRY
48     TRV=BV*D2TZ1/HTD(I,J)
49     TR=TR+TRV
50     IF(I,J,K)=((TR-TL)*DT+HTD(I,J)*TN(I,J,K))/
51     C(HTE(I,J)+KV*DT/(DZ*DZ*HTD(I,J)))
52     GO TO 8
53     300 TF(I,J,K)=TAM(I,J,K)
54     8 CONTINUE
55     10 CONTINUE
56     RETURN
57     END

```


7.2.24. TIDE

This sets the values of velocity at one boundary equal to the value of current coming into the domain.

```
1 SUBROUTINE TIDE(I,J,K,IN,JN,KN,U,V,H,G,D,E,T,TN,TF)
2 DIMENSION U(IN,JN,KN),V(IN,JN,KN),H(IN,JN,KN)
3 DIMENSION G(IN,JN,KN),D(IN,JN,KN),E(IN,JN,KN)
4 DIMENSION T(IN,JN,KN),TN(IN,JN,KN),TF(IN,JN,KN)
5 KN1=KN-1
6 DO 10 I=1,IN
7 DO 10 K=1,KN1
8 V(I,1,K)=2.0
9 G(I,1,K)=2.0
10 E(I,1,K)=2.0
11 CONTINUE
12 RETURN
13 END
```

ORIGINAL PAGE IS
OF POOR QUALITY.

7.2.25. UVEL3

This subroutine computes u and v velocities from the two horizontal momentum equations. The schemes used are forward in time and central in space. DuFort-Frankel scheme is used on the vertical viscous terms. This subroutine computes velocities only in the interior.

```

1  SUBROUTINE UVEL3(IN,JN,KN,U,V,H,G,D,E,DX,DY,DZ,W,TAUX,TAUY,DT,
2  CHT,HTC,HTF,HX,HY,ETA,P,MAR,KH,KV,CR,RR,FF,CP,CC,CI,CH,CV,RO,T,
3  CXX,YY,XXX,YYY)
4  REAL KH,KV
5  DIMENSION U(IN,JN,KN),V(IN,JN,KN),H(IN,JN,KN),G(IN,JN,KN),
6  CD(IN,JN,KN),E(IN,JN,KN),HT(IN,JN),HTD(IN,JN),HTE(IN,JN),HX(IN,JN),
7  CHY(IN,JN),ETA(IN,JN),P(IN,JN,KN),MAR(IN,JN),W(IN,JN,KN)
8  C,RO(IN,JN,KN),T(IN,JN,KN),
9  CXX(IN),YY(JN),XXX(IN),YYY(JN)
10  KN1=KN-1
11  IN1=IN-1
12  JN1=JN-1
13  DO 10 I=2,IN1
14  DO 10 J=2,JN1
15  DO 3 K=1,KN1
16  IF(MAR(I,J).EQ.11) GO TO 11
17  GO TO 10
18  11  CONTINUE
19  ETAY=(ETA(I+1,J)-ETA(I-1,J))/(2*DX)
20  DHX=(HT(I+1,J)-HT(I-1,J))/(2*DX)
21  DHY=(HT(I,J+1)-HT(I,J-1))/(2*DY)
22  DIPY=(P(I+1,J,K)-P(I-1,J,K))/(2*DX)
23  DIHUX=(H(I+1,J,K)+H(I+1,J,K)+4*HTD(I+1,J)-H(I-1,J,K)*
24  CH(I-1,J,K)+HTD(I-1,J))/(2*DX)
25  DIHUY=(H(I,J+1,K)+G(I,J+1,K)+HTD(I,J+1)-H(I,J-1,K)*
26  CG(I,J-1,K)+HTD(I,J-1))/(2*DY)
27  DIUX=(H(I+1,J,K)-H(I-1,J,K))/(2*DX)
28  DIUY=(H(I+1,J,K)+H(I-1,J,K)-2*H(I,J,K))/(DX*DX)
29  DIUY=(H(I,J+1,K)-H(I,J-1,K))/(2*DY)
30  D2UY=(H(I,J+1,K)+H(I,J-1,K)-2*H(I,J,K))/(DY*DY)
31  120  CONTINUE
32  IF(K.EQ.1) GO TO 7C
33  D1UWZ=(H(I,J,K+1)+H(I,J,K+1)-H(I,J,K-1)+H(I,J,K-1))/(2*DZ)
34  D1U7=(H(I,J,K+1)-H(I,J,K-1))/(2*DZ)
35  D2U71=(H(I,J,K+1)+H(I,J,K-1)-U(I,J,K))/(DZ*DZ)
36  GO TO 8C
37  7C  D1UWZ=(4*H(I,J,K+1)+6*H(I,J,K+1)-3*H(I,J,K)+H(I,J,K)-H(I,J,K+2)*
38  CW(I,J,K+2))/(2*DZ)
39  D1U7=(TAUX*HTD(I,J))/(KV)
40  D2U71=(2*H(I,J,K+1)-U(I,J,K))/(DZ*DZ)-2*D1U7/DZ
41  130  CONTINUE
42  DHT=(HTF(I,J)-HTD(I,J))/(DT)
43  UI=CI*(DIHUX*XX(I)+DIHUY*YY(J)+HTD(I,J)+D1UWZ+
44  C((K-1)*DZ-1)*D1U2*CHT)
45  UP1=CP*HTD(I,J)*(FTAX*GR)*(-1.)*XX(I)
46  UP2=(DIFX*YX(I)/PR)+HTD(I,J)
47  UP3=(GR*XX(I)*DHX*(K-1)*CZ)*HTD(I,J)
48  UP=UP1+UP2+UP3
49  UC=CC*HTD(I,J)*FF+G(I,J,K)
50  UH=CH*KH*(DHX*XX(I)+XX(I)*DIUY+HT(I,J)*XX(I)*XX(I)*D2UX
51  C*HT(I,J)+XXX(I)*DIUY)+
52  CCH*KH*(DHY*YY(J)+YY(J)*DIUY+HT(I,J)*YY(J)+YY(J)*D2UY
53  C*HT(I,J)+YYY(J)*DIUY)
54  UV=(KV*D2U71)/HTD(I,J)
55  D(I,J,K)=(-UI+UC+UH+UV)+DT+HTD(I,J)+H(I,J,K))/
56  C(HTF(I,J)+(KV*DT)/(DZ*DZ)*HTD(I,J))

```

```

57      8      CONTINUE
58      10     CONTINUE
59          DO 30 I=2,INI
60          DO 70 J=2,JNI
61          DO 7 K=1,KN1
62          IF (MAP(I,J).EQ.11) GO TO 12
63          GO TO 30
64      12     CONTINUE
65          ETAY=(ETA(I,J+1)-ETA(I,J-1))/(2*DY)
66          DHX=(HT(I+1,J)-HT(I-1,J))/(2*DX)
67          DHY=(HT(I,J+1)-HT(I,J-1))/(2*DY)
68          DIPY=(P(I,J+1,K)-P(I,J-1,K))/(2*DY)
69          DIHUVX=(H(I+1,J,K)*G(I+1,J,K)*HTD(I+1,J)-H(I-1,J,K)*G(I-1,J,K)*
70          CHTD(I-1,J))/(2*DX)
71          DIHVVY=(G(I,J+1,K)*G(I,J+1,K)*HTD(I,J+1)-G(I,J-1,K)*
72          CG(I,J-1,K)*HTD(I,J-1))/(2*DY)
73          DIVX=(G(I+1,J,K)-G(I-1,J,K))/(2*DX)
74          D2VX=(G(I+1,J,K)+G(I-1,J,K)-2*G(I,J,K))/(DX*DX)
75          DIVY=(G(I,J+1,K)-G(I,J-1,K))/(2*DY)
76          D2VY=(G(I,J+1,K)+G(I,J-1,K)-2*G(I,J,K))/(DY*DY)
77          IF (K.EQ.1) GO TO 9C
78          DIVWZ=(G(I,J,K+1)*W(I,J,K+1)-G(I,J,K-1)*W(I,J,K-1))/(2*DZ)
79          D1VZ=(G(I,J,K+1)-G(I,J,K-1))/(2*DZ)
80          D2VZ1=(G(I,J,K+1)+G(I,J,K-1)-V(I,J,K))/(DZ*DZ)
81          GO TO 9F
82      9F     DIVWZ=(4*G(I,J,K+1)*W(I,J,K+1)-3*G(I,J,K)*W(I,J,K)-G(I,J,K+2)*
83          C*G(I,J,K+2))/(2*DZ)
84          DIVZ=(TAUY*HTD(I,J))/(KV)
85          D2VZ1=(2*G(I,J,K+1)-V(I,J,K))/(DZ*DZ)-2*D1VZ/DZ
86      9E     CONTINUE
87          DHT=(HT(I,J)-HTD(I,J))/(DT)
88          VI=CI+(DIHUVX*XX(I)+DIHVVY*YY(J)+HTD(I,J)+D1VWZ+
89          C((K-1)*DZ-1)*D1VZ*DHT)
90          VP1=CP*HTD(I,J)*(ETAY*GR)*(-1.)*YY(J)
91          VP2=- (DIPY*YY(J)/FR)*HTD(I,J)
92          VP3=(FR*YY(J)+DHY*(K-1)*DZ)*HTD(I,J)
93          VP=VP1+VP2+VP3
94          VC=CC*HTD(I,J)*FF*H(I,J,K)
95          VH=CH*KH*(DHX*XX(I)*XX(I)+DHY*HT(I,J)*XX(I)*XX(I)+D2VX
96          C*HT(I,J)*XXX(I)*G1VX)
97          C+CH*KH*(DHY*YY(J)*YY(J)+D1VY*HT(I,J)*YY(J)*YY(J)+D2VY
98          C*HT(I,J)*YYY(J)*D1VY)
99          VV=KV*D2VZ1/HTD(I,J)
100         E(I,J,K)=((-VI+VC+VH+VV)*DT+HTD(I,J)+G(I,J,K))/
101         CHTF(I,J)+(KV*DT)/(DZ*DZ*HTD(I,J))
102      7      CONTINUE
103      70     CONTINUE
104          RETURN
105          END

```

ORIGINAL PAGE IS
OF POOR QUALITY

PRECEDING PAGE BLANK NOT FILMED

7.2.26. WHTOW

This subroutine calculates the value of $W(\Omega)$ which is used in the model from the value of $WH(w)$ specified.

```

1      SUBROUTINE WHTOW(IN,JN,KN,HTD,HTE,HT,ETA,D,E,W,WH,DX,DY,DZ,DT,MAR,
2      CX,YY)
3      DIMENSION HT(IN,JN),HTD(IN,JN),HTE(IN,JN),MAR(IN,JN),D(IN,JN,KN),
4      CE(IN,JN,KN),W(IN,JN,KN),WH(IN,JN,KN),ETA(IN,JN),XX(IN),YY(JN)
5      KN1=KN-1
6      IN1=IN-1
7      JN1=JN-1
8      DO 10 K=2,KN
9      DO 10 I=2,IN1
10     DO 10 J=2,JN1
11     IF(MAR(I,J).EQ.11) GO TO 44
12     GO TO 10
13     44  OHX=(HTE(I+1,J)-HTE(I-1,J))/(2*DX)
14     OHY=(HTE(I,J+1)-HTE(I,J-1))/(2*DY)
15     ETAX=(ETA(I+1,J)-ETA(I-1,J))/(2*DX)
16     ETAY=(ETA(I,J+1)-ETA(I,J-1))/(2*DY)
17     DHT=(HTD(I,J)-HTE(I,J))/(DT)
18     W(I,J,K)=(WH(I,J,K)-((K-1)*DZ-1)*DHT-D(I,J,K)*ETAX*XX(I)
19     C-E(I,J,K)*ETAY*YY(J)+(K-1)*DZ*D(I,J,K)*OHX*XX(I)+(K-1)*DZ*
20     CE(I,J,K)*OHY*YY(J))/HT(I,J)
21     10  CONTINUE
22     RETURN
23     END

```

7.2.27. WVEL

This subroutine computes the vertical velocity

(Ω) by using the equation

$$\Omega = - \frac{1}{H} \int_0^{\sigma} \left\{ X' \frac{\partial(HU)}{\partial X} + Y' \frac{\partial(HV)}{\partial Y} \right\} d\sigma + \frac{1}{H} \int_0^{\sigma} \left\{ X' \frac{\partial(HU)}{\partial X} \right. \\ \left. + Y' \frac{\partial(HV)}{\partial Y} \right\} d\sigma + \frac{\sigma}{H} (W_b - U_b X' \frac{\partial h}{\partial X} - V_b Y' \frac{\partial h}{\partial Y})$$

the numerical scheme used is central in space and trapezoidal rule is for numerical integration.


```

1      C      THIS PROGRAM CALCULATES THE VERTICAL VELOCITY IN THE GAMMA COORDS
2      C
3      SUBROUTINE WVEL(IN,JN,KN,U,V,W,HT,DX,DY,DZ,MAR,XX,YY,WH)
4      DIMENSION U(IN,JN,KN),V(IN,JN,KN),HT(IN,JN),W(IN,JN,KN),MAR(IN,JN)
5      C,XX(IN),YY(JN),WH(IN,JN,KN)
6      KN1=KN-1
7      IN1=IN-1
8      JN1=JN-1
9      DO 10 I=2,IN1
10     DO 10 J=2,JN1
11     DUM=0.
12     DO 9 K=1,KN
13     IF(MAR(I,J).EQ.11) GO TO 20
14     GO TO 10
15     20    DIHUX=(HT(I+1,J)+U(I+1,J,K)-HT(I-1,J)+U(I-1,J,K))/(2*DX)
16     DIHUY=(HT(I,J+1)+V(I,J+1,K)-HT(I,J-1)+V(I,J-1,K))/(2*DY)
17     IF (K.EQ.1) GO TO 17
18     IF(K.EQ.KN) GO TO 17
19     DUM=DUM+DZ*(DIHUX*XX(I)+DIHUY*YY(J))/HT(I,J)
20     GO TO 9
21     17    DUM=DUM+DZ*(DIHUX*XX(I)+DIHUY*YY(J))/(2*HT(I,J))
22     9     CONTINUE
23     DUM=DUM+W(I,J,KN)
24     WUD=0.
25     DO 8 K=2,KN1
26     IF(MAR(I,J).EQ.11) GO TO 40
27     GO TO 10
28     40    DIHUX1=(HT(I+1,J)+U(I+1,J,K-1)-HT(I-1,J)+U(I-1,J,K-1))/(2*DX)
29     DIHUY1=(HT(I+1,J)+U(I+1,J,K)-HT(I-1,J)+U(I-1,J,K))/(2*DX)
30     DIHUY1=(HT(I,J+1)+V(I,J+1,K)-HT(I,J-1)+V(I,J-1,K))/(2*DY)
31     DIHUY1=(HT(I,J+1)+V(I,J+1,K-1)-HT(I,J-1)+V(I,J-1,K-1))/(2*DY)
32     WUD=WUD+DZ*(DIHUX*XX(I)+DIHUX1*XX(I)+DIHUY*YY(J)+DIHUY1*YY(J))/
33     (2*HT(I,J))
34     W(I,J,K)=-WUD+DUM+(K-1)*DZ
35     8     CONTINUE
36     10    CONTINUE
37     RETURN
38     END

```

1965
 QUALITY

7.2.28. XYSH

This subroutine does the stretching in both horizontal directions and determines the constants $XX(I)$, $YY(J)$, $XXX(I)$ and $YYY(J)$. This subroutine needs the values of $DEEX$, $DEEY$, $EEEX$ which are read in the main program. In order to obtain these values, another main program "CONST" has to be run.

```

1      C      THIS SUBROUTINE COMPUTES THE HORIZONTAL STRETCHING CONSTANT
2      C
3      SUBROUTINE XYSH(IN,JN,DELY,DELY,DEEX,DEEY,EFEX,EEFY,XX,YY,
4      CXXX,YYY,A,B,X,Y)
5      DIMENSION X(IN),XX(IN),XXX(IN),Y(JN),YY(JN),YYY(JN),A(IN),B(JN)
6      DO 10 I=1,IN
7      X(I)=(I-1)*DELY
8      A(I)=(X(I)-DEEX)/EEFX
9      XX(I)=1./COSH(A(I))
10     XXX(I)=-SINH(A(I))/(EEEX*COSH(A(I))*3)
11     CONTINUE
12     DO 20 J=1,JN
13     Y(J)=(J-1)*DELY
14     B(J)=(Y(J)-DEEY)/EEFY
15     YY(J)=1./COSH(B(J))
16     YYY(J)=-SINH(B(J))/(EEEY*COSH(B(J))*3)
17     CONTINUE
18     RETURN
19     END

```

REFERENCES

Crandell, S. H., Engineering Analysis, McGraw Hill, New York, 1965.

Edinger, J. E., and Geyer, J.C., "Heat Exchange in the Environment," E.E.I. Publication No. 65-902, Edison Electric Institute, 1965. (R1971)

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

Lee, S. S., and Sengupta, S., "Three-Dimensional Thermal Pollution Models," Phase III, NASA-CR-144858, 1976.

Murrill, P.W., and Smith, C.L., An Introduction to FORTRAN IV Programming, Intext Educational Publishers, 1975.

Sengupta, S., Lee, S.S., and Miller, H.P., "A Three-Dimensional Free-Surface Numerical Model for Transport Processes in Biscayne Bay", in Computing Methods in Geophysical Mechanics, ASME Applied Mechanics Division, 1977.

Tsai, C.F., Ph.D. dissertation, University of Miami, Coral Gables, Florida, 1977.

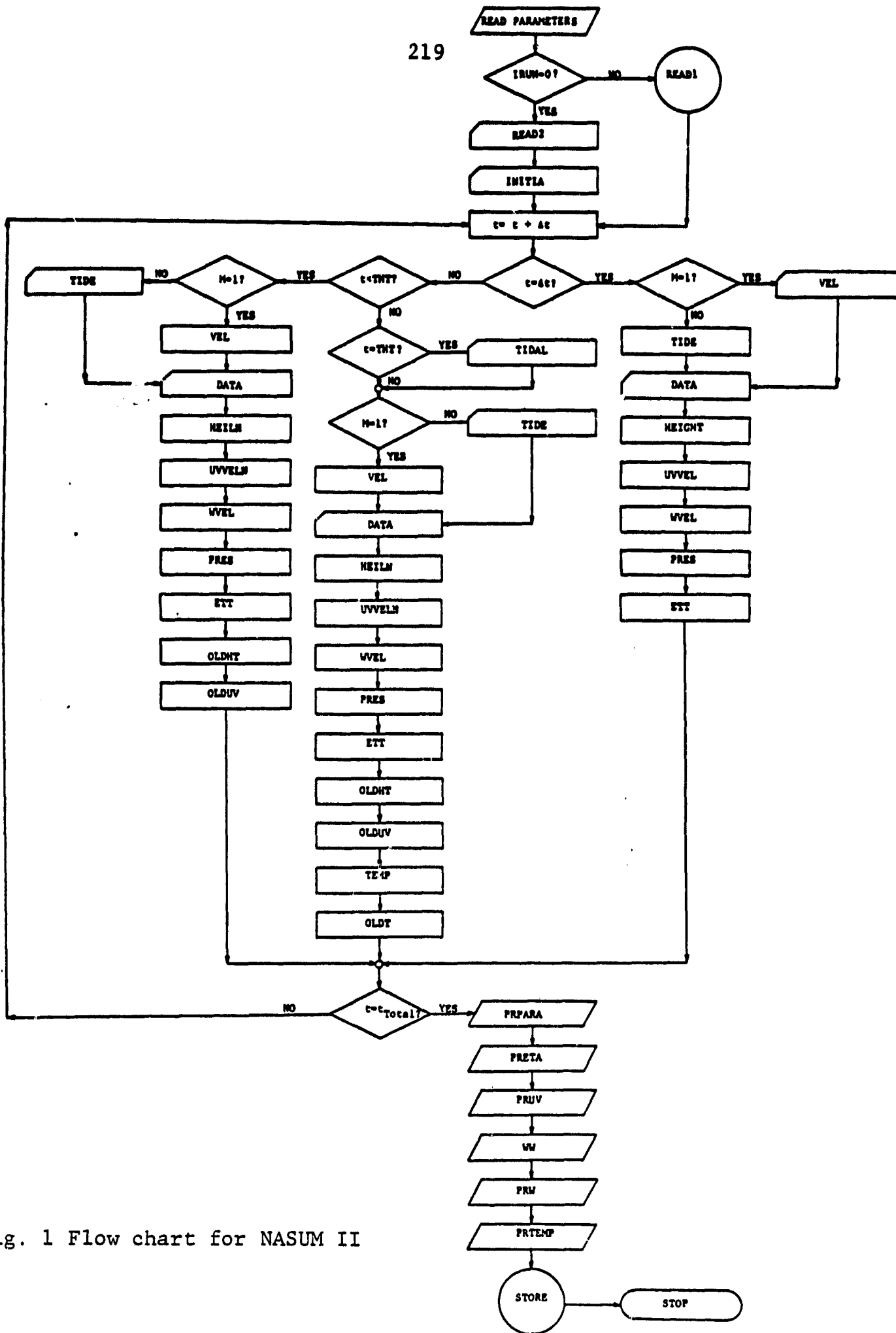
Waldrop, W., and Farmer, R.C., "Three-Dimensional Flow and Sediment Transport at River Mouths," Coastal Studies Institute, Louisiana State University, Baton Rouge. Technical Report No. 150, Sept. 1973.

ORIGINAL PAGE IS
OF POOR QUALITY

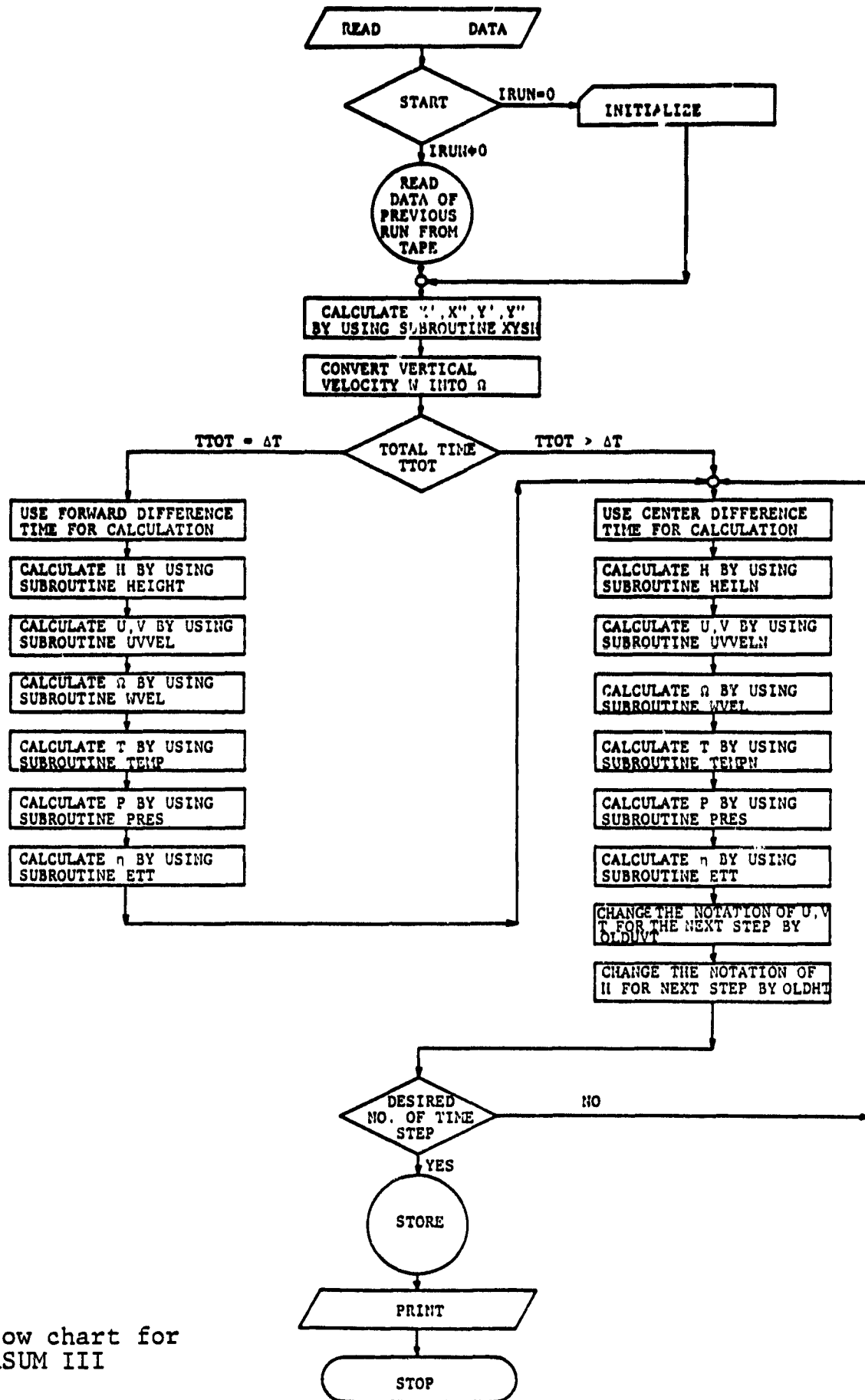
PRECEDING PAGE BLANK NOT FILMED

TABLE 1: Representation of variables at different time levels

VARIABLE	n-1	n	n+1
Height (H)	HT	HTD	HTE
Height (h)	HI	HI	HI
u-velocity	U	H	D
v-velocity	V	G	E
w-velocity	WH	WH	WH
Ω	W	W	W
Density	RR	RR	RR
Temperature	T	TN	TF
η	ETA	ETA	ETA



- Fig. 1 Flow chart for NASUM II



-Fig 2 Flow chart for NASUM III

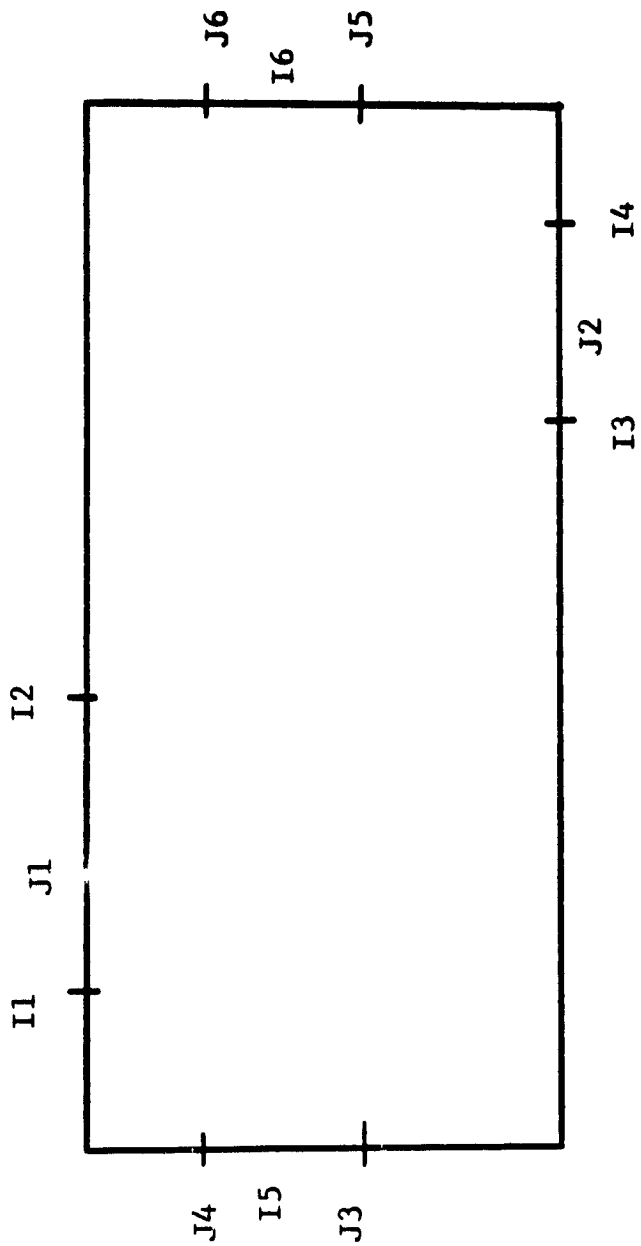


Fig.3 Location of Open Boundaries for NASUM II (Far-Field)

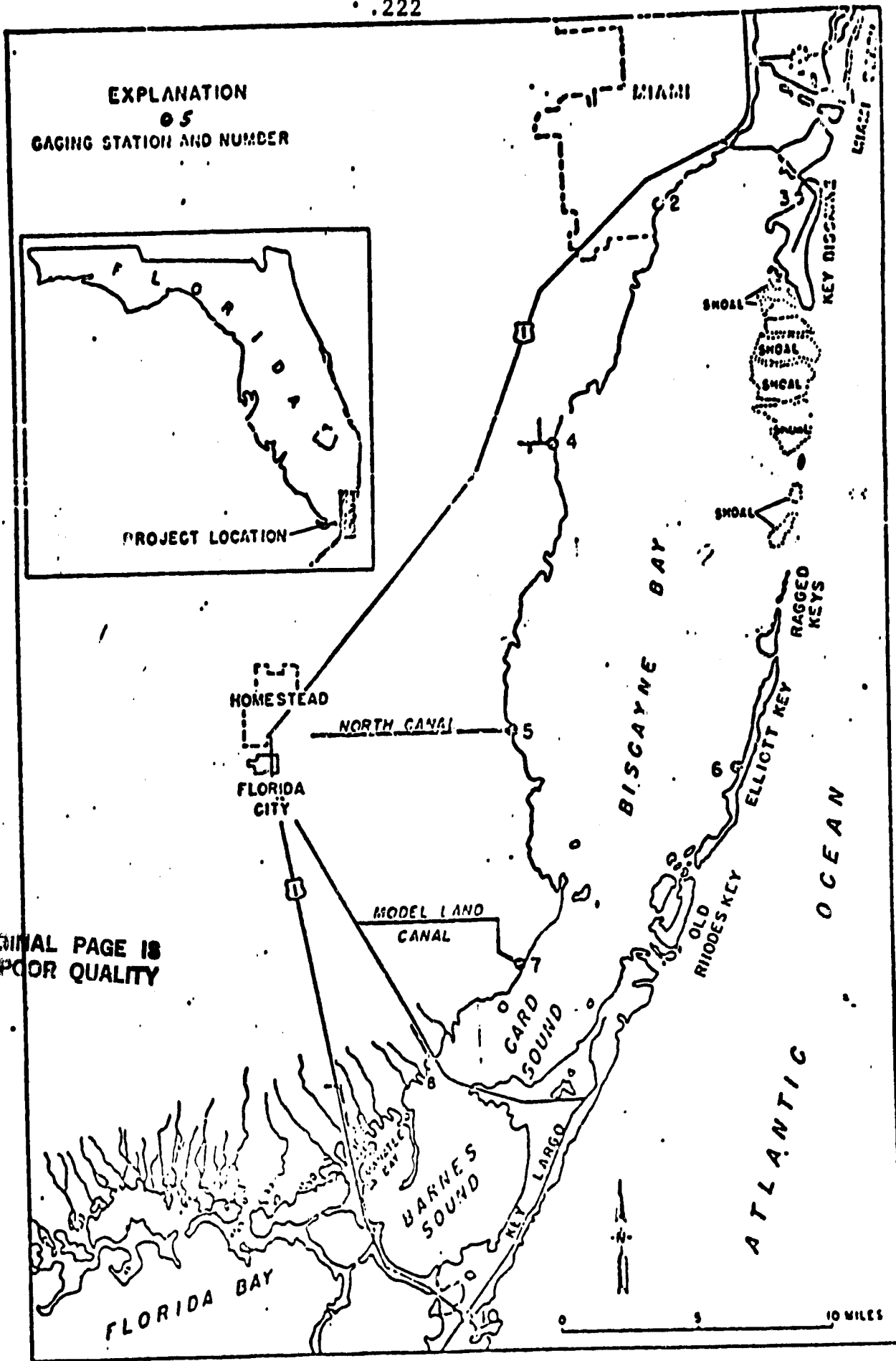


Fig. 4 Map of southeastern Dade County showing the area of investigation and location of gaging stations.

Mar Specification of Corners

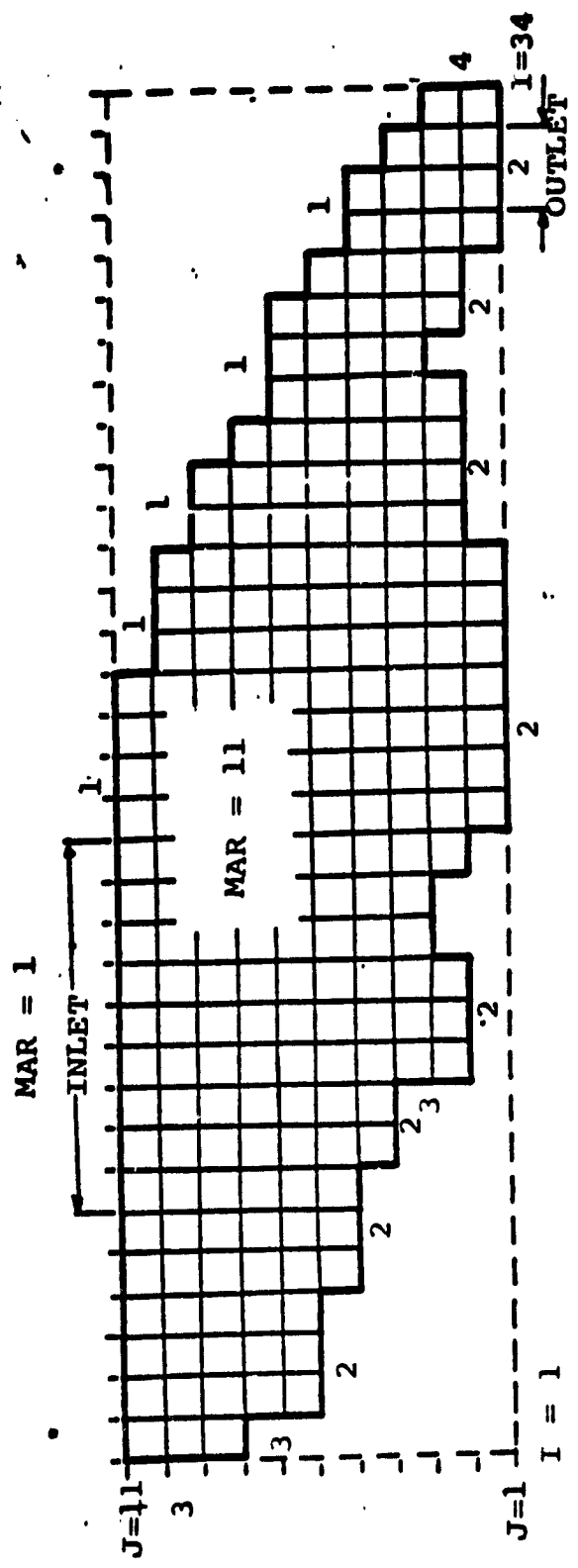


Fig. 5 Grid configuration and MAR node values.

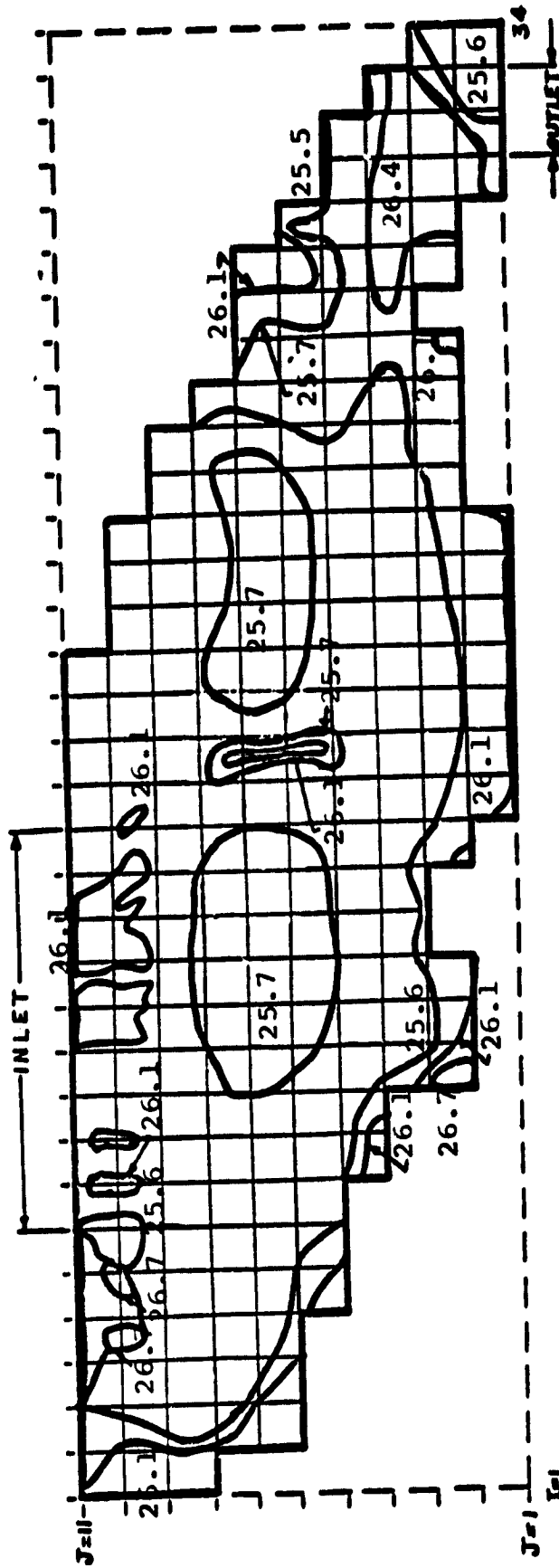


Fig. 6 Surface Isotherms For Biscayne Bay From NASA - 6
IR Scanner Data, Corrected by Ground Truth, on the
morning of April 15, 1975

**THERMAL POLLUTION LAB
UNIVERSITY OF MIAMI**

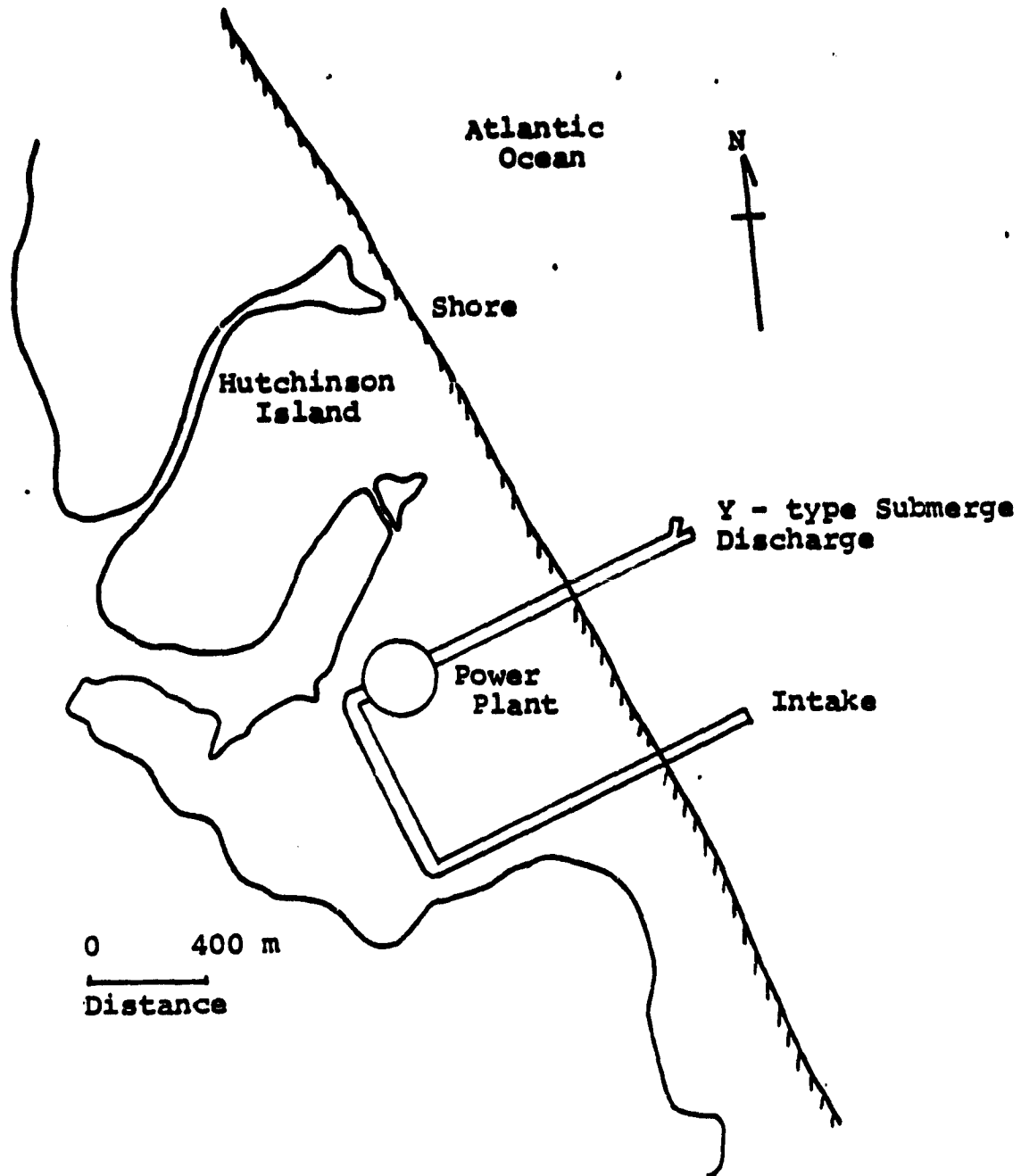


Fig. 7 Florida Power and Light Company's Hutchinson Island Site Power Plant

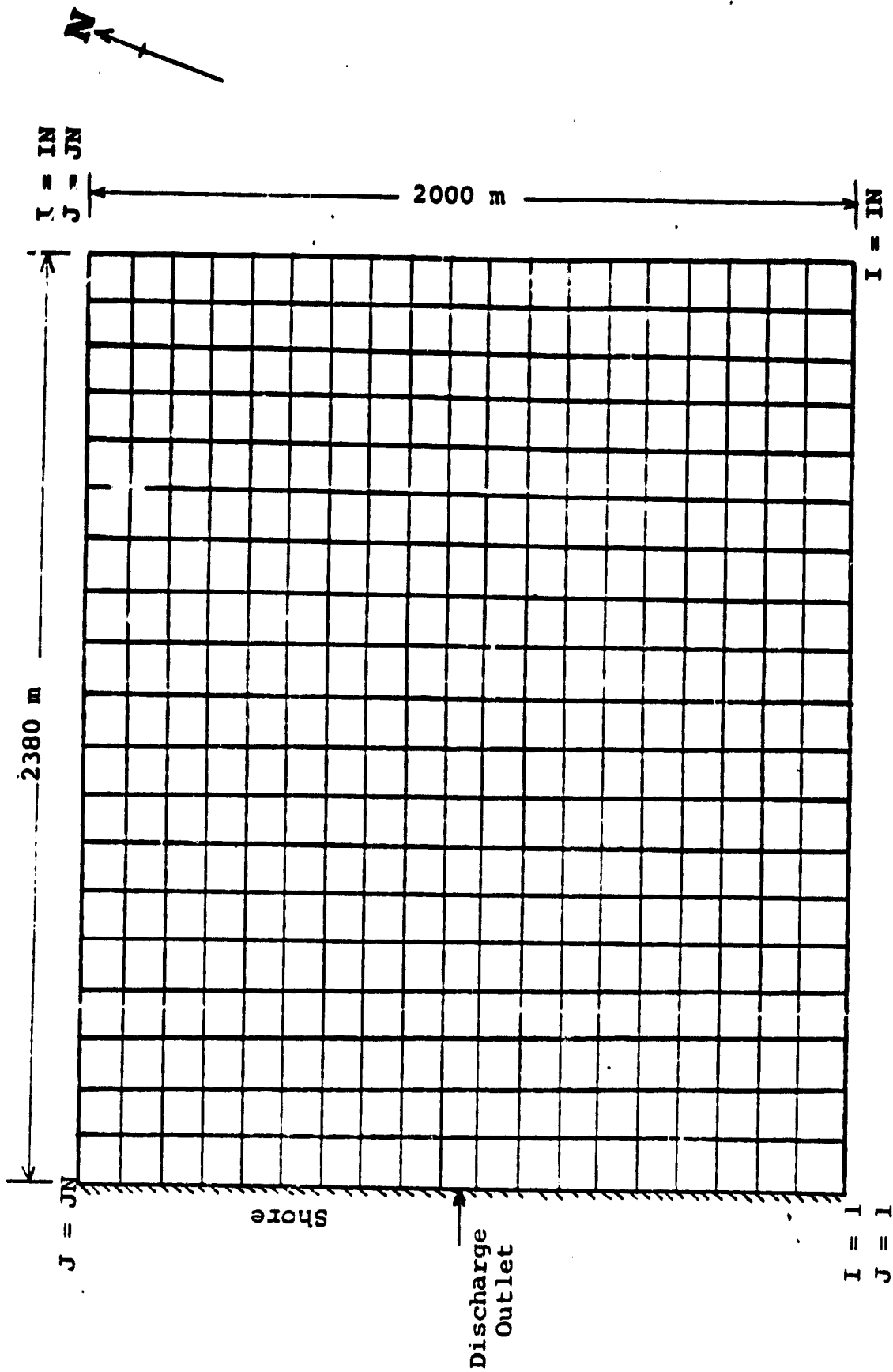


Fig. 8 Horizontal Grid Point System Without Stretching
For Free Surface Near Field Model Applied to
Hutchinson Island Site

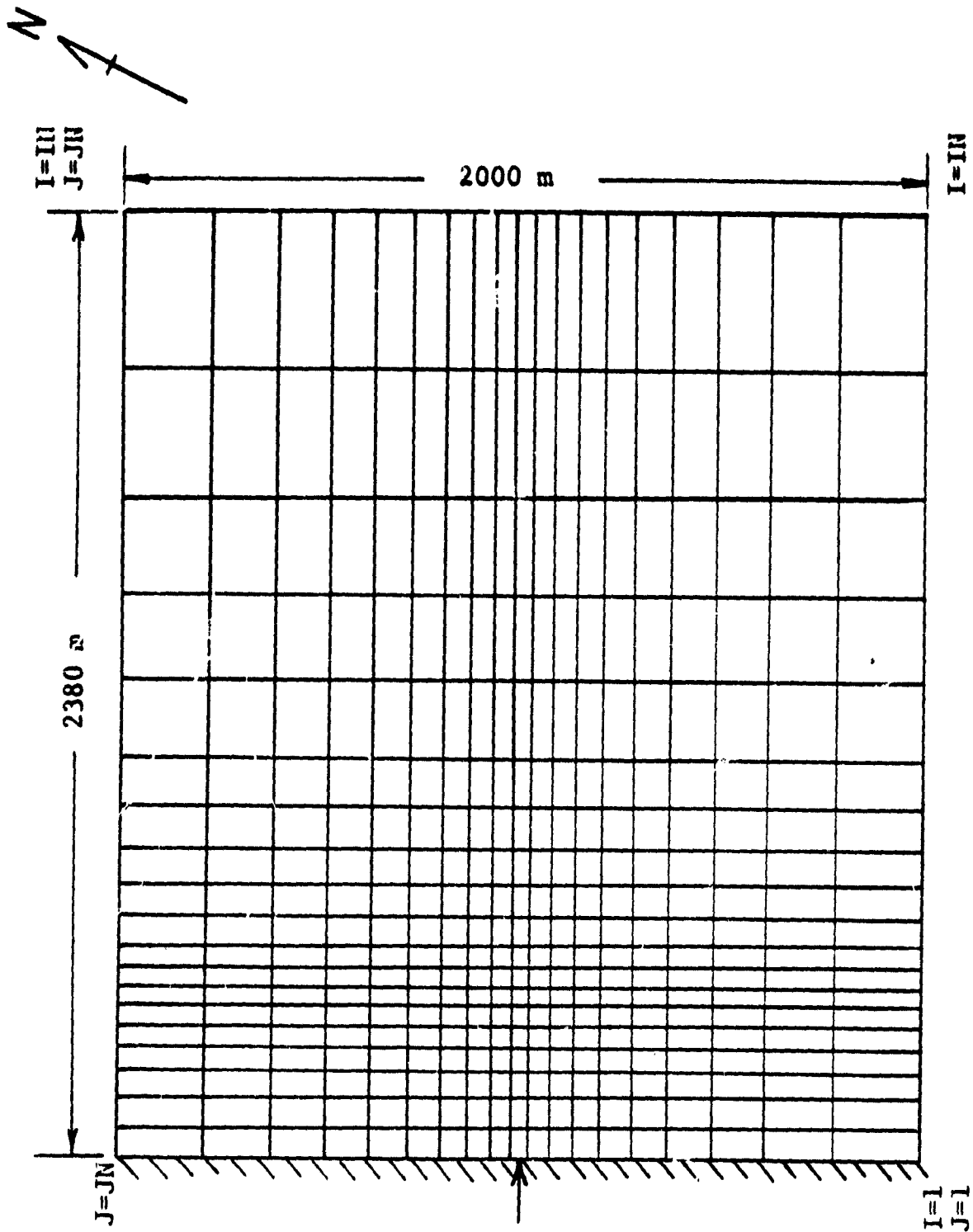


Fig.9 Physical horizontal grid point system for the free surface model sample problem applied to Hutchinson Island site.

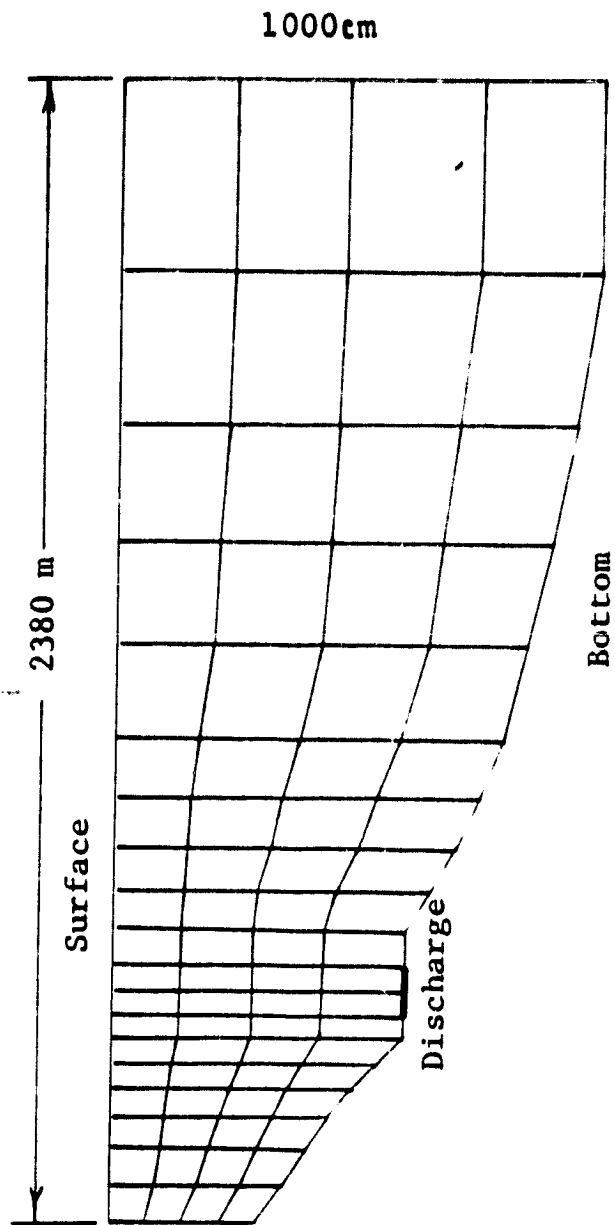


Fig. 10 Numerical grid point system on a distorted vertical section for the free surface model sample problem applied at Hutchinson Island.

Discharge volume 263,000 G.P.M
Discharge Velocity: 0.102 cm/sec
Discharge Temperature: 35.0°C
Current: 2 cm/sec
(Parallel to shoreline)
Wind: None
Total Time: 60 min.

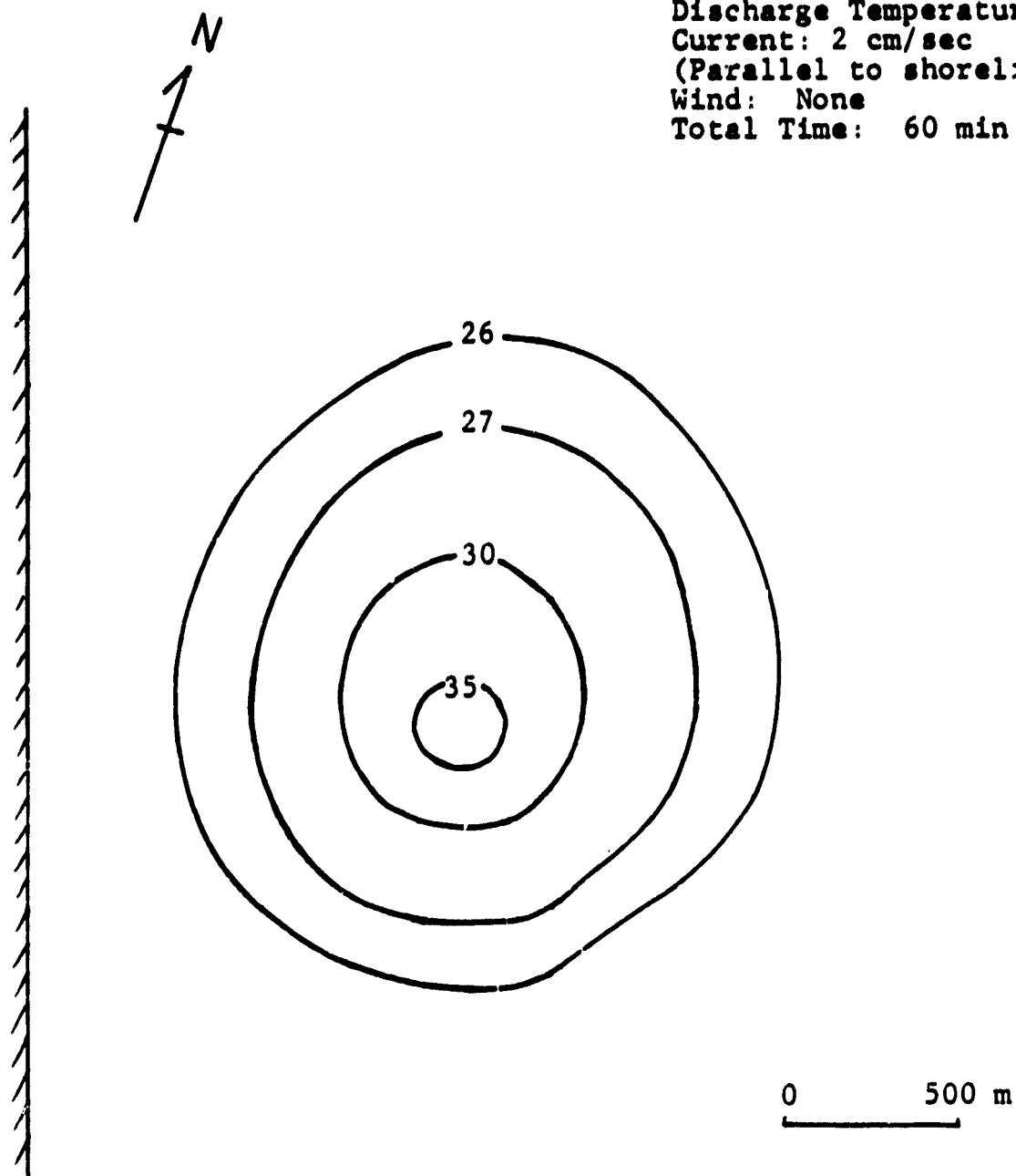


Fig.11 Surface isotherms obtained after 1 hour of simulation of the free surface model for the sample problem. (Hutchinson Island Site)

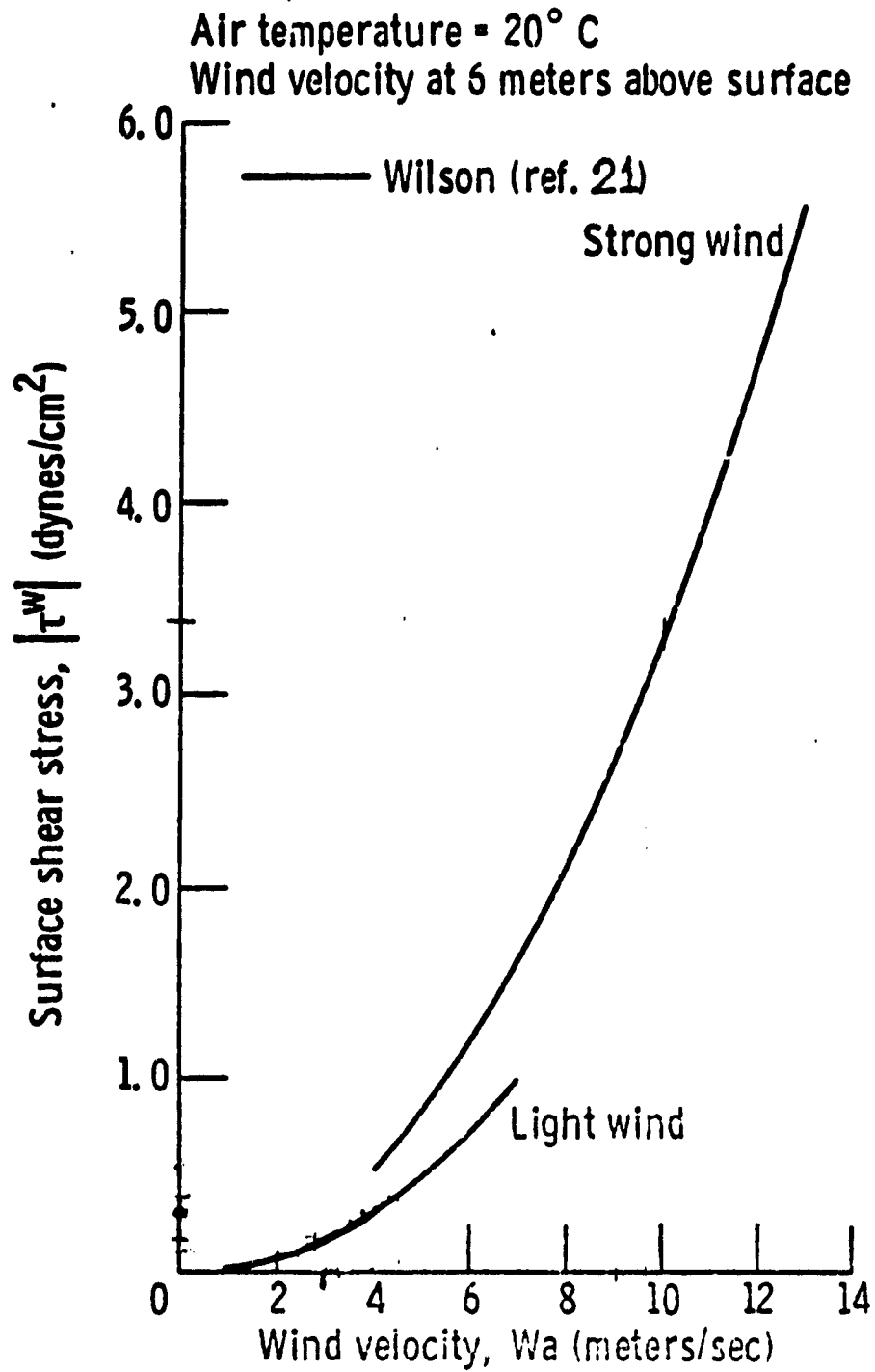
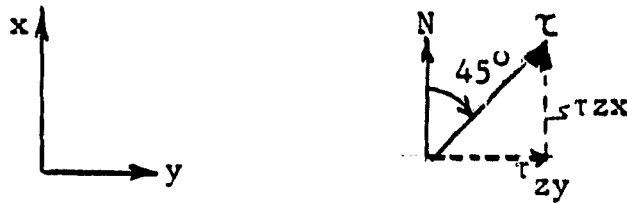


Figure 12 Wind shear stress relation.

APPENDIX A
Wind Stresses

The wind shear stresses τ_{zx} and τ_{zy} are computed by using the Wilson curve as shown in Fig.12. First, the magnitude of the wind velocity, in meters/sec., is used to read off from this curve the resultant shear stress; τ_{zx} and τ_{zy} are determined by simply resolving τ into its respective components.

As an example, consider a wind of 10 mph from the South West direction. Assume that the direction of North is in the same direction as the positive x-axis, and East is in the same direction as the positive y-axis as shown below.



$$\text{Then } \tau_{zx} = \tau \cos 45^\circ = .4 \cos 45^\circ = .283 \text{ dynes/cm}^2$$

$$\tau_{zy} = \tau \sin 45^\circ = .4 \sin 45^\circ = .283 \text{ dynes/cm}^2$$

Where $\tau = .4 \text{ dynes/cm}^2$ for 10 mph = 4.47 meters/sec.

APPENDIX B

HEAT TRANSFER MECHANISMS

The analysis in this section is taken from Harleman et al.
(1973)

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 φ_{sc} = clear sky solar radiation obtained using the 100% possible sunshine curve (given in Appendix B)

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:

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

2. Atmospheric Radiation (long wave)

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

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

Where ϵ = average emittance of the atmosphere

σ = Stefan-Boltzmann constant

T_a^* = air temperature (absolute)

However, good agreement with experimental data has indicated that φ_a 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, φ_{ac} , may be expressed as:

$$\varphi_{ac} = 1.12 \times 10^{-13} (T_a^*)^6$$

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:

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

3. Longwave Radiation from the Water Surface φ_{br}

In reference () it is noted 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.

4. Evaporative Heat Flux, φ_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 flux (mass/area/time)

ρ = density of water

W_z = windspeed at height z above surface

$F(W_z)$ = 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, φ_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 $\approx W$ 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 $F(W) = 22.4 (T_s - T_a)^{1/3} + 14W$. . . artificially heated surface
 surface

5. Conduction Heat Flux,

Bowen (see reference) 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 $R_b = C_b \left| \frac{T_s - T_a}{e_s - e_a} \right| = \text{Bowen Ratio}$

and $C_b = \text{Bowen constant} = 0.255 \text{ mm Hg}/^\circ\text{F}$

and, therefore the conduction heat flux, φ_c , may be expressed as,

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

APPENDIX C
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:

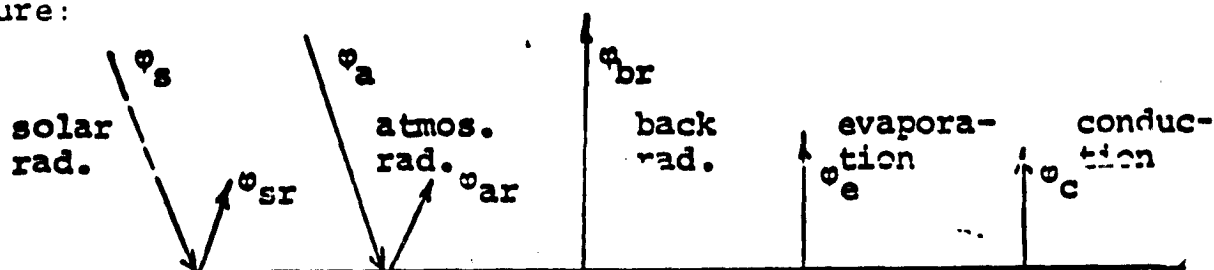


Fig. F-1. Heat Transfer Mechanisms at the Water Surface.

The following heat balance results,

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

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

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

$$\varphi_n = \varphi_r - \varphi_L \dots\dots\dots(F-2)$$

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

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

A. Equilibrium Temperature Calculation, T_e (See Appendix E)

Under equilibrium conditions equation (F-2) yields,

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

so that,

$$\phi_T = \phi_L \dots\dots\dots (F-3)$$

Then by using the approximate formulae in reference () we obtain by setting $T_s = T_e$,

$$\begin{aligned} 0.94 \phi_{sc} (1-0.65C^2) + 1.16 \times 10^{-13} (T_a^*)^6 (1+0.17C^2) \\ = 0.97 \sigma (T_e^*)^4 + F(W) [(e_s - e_a) + C_b (T_e - T_a)] \dots (F-4) \end{aligned}$$

where ϕ_{sc} = clear sky solar radiation

C = cloudiness ratio

T_a = air temperature ($^{\circ}C$ or $^{\circ}F$)

T_e = equilibrium temperature ($^{\circ}C$ or $^{\circ}F$)

T^* = absolute temperature ($^{\circ}K$ or $^{\circ}R$)

$F(W)$ = windspeed function (BTU/ft²/day, mm Hg)

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

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

σ = Stefan-Boltzmann constant = 4.1×10^{-8} BTU/ft², day, $^{\circ}R^4$

C_b = Bowen constant = 0.255 mm Hg/ $^{\circ}F$ (see Appendix E)

W = windspeed (mph)

For a natural water surface,

$$F(W) = 17W \dots\dots\dots (F-5a)$$

and, for an artificially heated surface,

$$F(W) = 22.4 (T_e - T_a)^{1/3} + 14W \dots\dots\dots (F-5b)$$

Thus, equation (F-4) becomes,

$$\begin{aligned} 0.94 \varphi_{sc} (1-0.65c^2) + 1.16 \times 10^{-13} (T_a^*)^6 (1+0.17c^2) \\ = 0.97\varphi (T_e^*)^4 + 17W [(e_s - e_a) + 0.255 (T_e - T_a)] \end{aligned}$$

or,

$$\begin{aligned} 0.94 \varphi_{sc} (1-0.65c^2) + 1.16 \times 10^{-13} (T_a^*)^6 (1+0.17c^2) \\ = 0.97\varphi (T_e^*)^4 + [22.4 (T_e - T_a)^{1/3} + 14W] \\ [(e_s - e_a) + 0.255 (T_e - T_a)] \end{aligned}$$

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

B. Surface Heat Transfer Coefficient (K)

From reference () 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,

$$\begin{aligned} K = 3.88\varphi (T_s^*)^3 + F(W) \left[\left(\frac{\partial e_s}{\partial T} \right)_{T=T_{av}} + c_b \right] \\ + [(e_s - e_a) + c_b (T_s - T_a)] \frac{\partial F(W)}{\partial T_{av}} \end{aligned}$$

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

C. Numerical Example

Consider natural water surface

$$C = 0$$

$$T_a = 25^\circ\text{C}$$

$$W = 10 \text{ mph}$$

Location - Miami (latitude 26°N)

Date - December 20

From reference (),

$$\textcircled{T}_a = 25^\circ\text{C} \rightarrow e_a \approx 0.43 \text{ psia}$$

$$\textcircled{T}_e = 27^\circ\text{C} \text{ as guess } \approx e_s = e_e \approx 0.51 \text{ psia}$$

From reference (), Figure 2.15, pg. 2-61 (see Figure F-3)

$$\varphi_{sc} \approx 425 \text{ Langleys/day} = 1560 \text{ BTU/ft}^2/\text{day} \dots \text{ using}$$

100%sunshine curve at 26°N , Dec. 20

Note: 1 Langley/min. = 220.62 BTU/ft^2 , hr. = 1 calarie/cm²min.

Then using equation (F-6) with $C = 0$,

$$\begin{aligned} 0.94 (1560) (1) + 1.16 \times 10^{-13} (5.37 \times 10^2)^6 (1) &= 4250 \\ 4 \times 10^{-8} (5.406 \times 10^2)^4 + 170 [(.255) (2) + (.08) (51.7)] \\ 4206 &\text{ close enough!} \end{aligned}$$

$$\therefore T_e \approx 27^\circ\text{C}$$

(where 1 psia = 51.7 mm Hg)

Then from equation (),

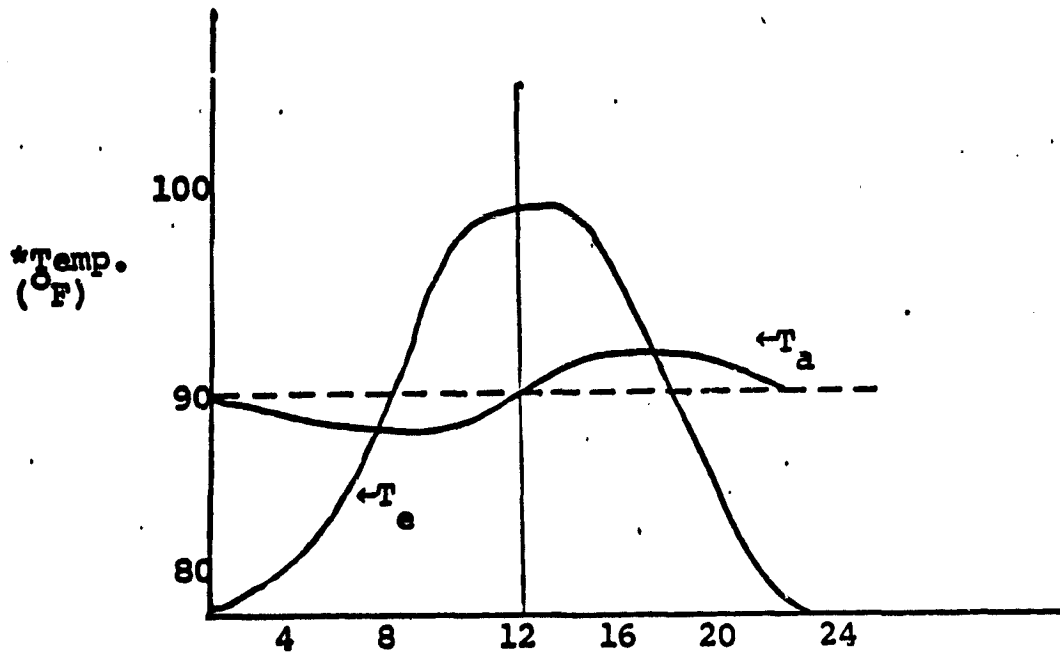
$$K = 3.88 \times 4.1 \times 10^{-8} (5.406 \times 10^2)^3 + 170 (.255 + 0.0251 \times 51.7)$$

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

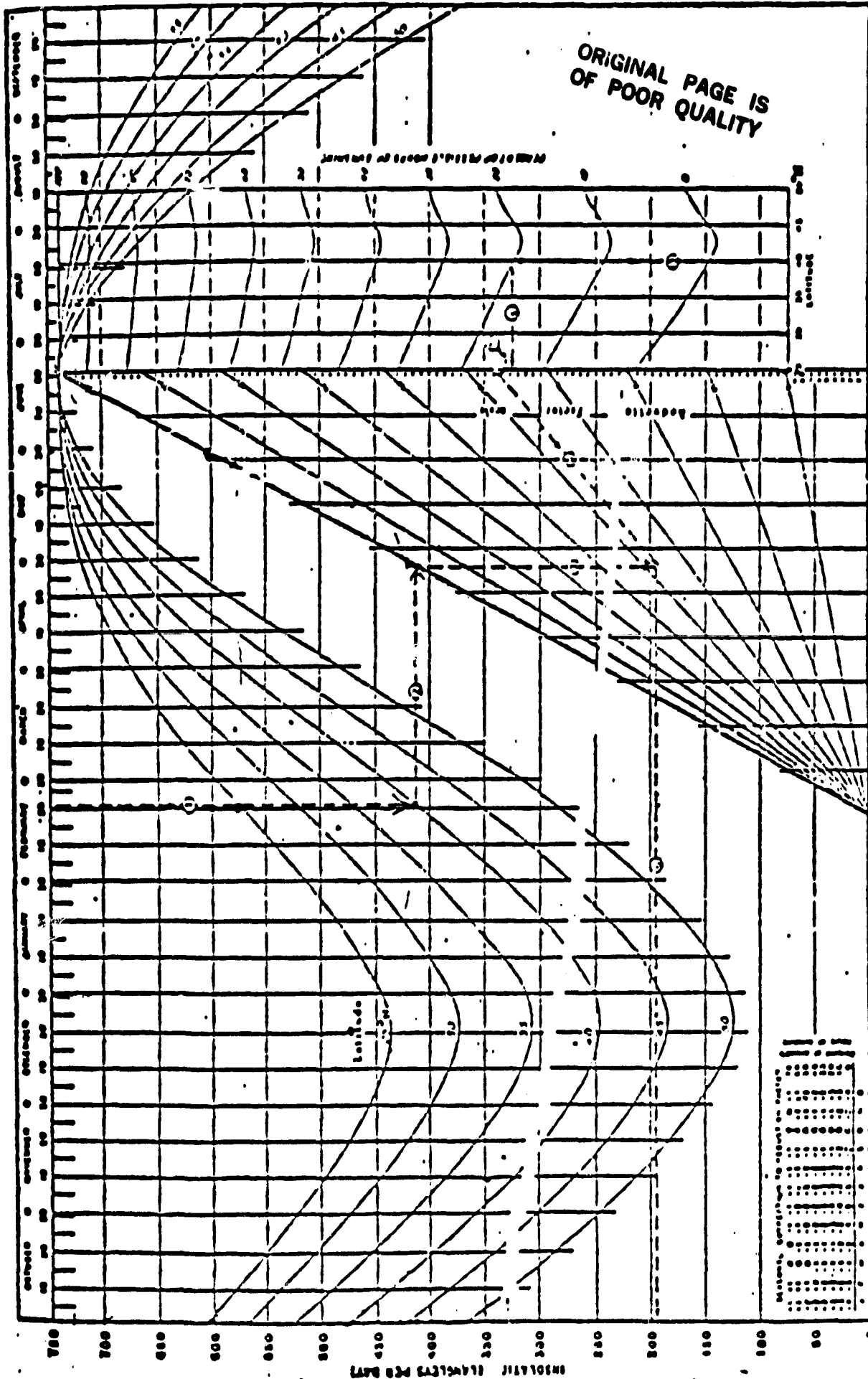
where $\left. \frac{\partial e_s}{\partial T} \right|_{T=T_{av}} \approx \frac{e_s - e_a}{T_e - T_a} = .0251$

D. Discussion

The equilibrium 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 (F-2) 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.



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



Working diagrams for computing insulation as a function of percent possible penetration, latitude, and time of year. The example shown by each of them with arrows will show the daily total number of a certain latitude 40° N. on 1 January 21 and with 10 percent possible penetration. Step 1—enter the graph at 40° N. on 1 January 21 and move horizontally to the right to the curve labeled 10. Step 2—enter the graph at 40° N. on 1 January 21 and move horizontally to the right to the curve labeled 10. Step 3—move vertically to the right to the curve labeled 10. Step 4—move vertically to the right to the curve labeled 10. Step 5—move vertically to the right to the curve labeled 10. Step 6—move vertically to the right to the curve labeled 10. Step 7—move vertically to the right to the curve labeled 10. Step 8—move vertically to the right to the curve labeled 10. Step 9—move vertically to the right to the curve labeled 10. Step 10—move vertically to the right to the curve labeled 10.

Figure F-3. Daily Average Isolation (from Hamon (1954))