NASA CR-

# DEVELOPMENT OF A THREE-DIMENSIONAL TIME-DEPENDENT FLOW FIELD MODEL

(NASA-CR-120762) DEVELOPMENT OF A THREE-DIMENSIONAL TIME-DEPENDENT FLOW FIELD MODEL (Louisiana State Univ.) 95 p HC $4.75 CSCL 08H N75-24070
Unclas
G3/43 22759

by

R. C. Farmer, W. R. Waldrop,

F. H. Pitts & K. R. Shah

Prepared under Contract No. NAS8-30380

Department of Chemical Engineering

Louisiana State University

Baton Rouge, La. 70803



January, 1975

NASA CR-

# DEVELOPMENT OF A THREE-DIMENSIONAL TIME-DEPENDENT FLOW FIELD MODEL

by

R. C. Farmer, W. R. Waldrop,

F. H. Pitts & K. R. Shah

Prepared under Contract No. NAS8-30330

Department of Chemical Engineering

Louisiana State University

Baton Rouge, La. 70803

January, 1975

NASA CR-

# DEVELOPMENT OF A THREE-DIMENSIONAL TIME-DEPENDENT FLOW FIELD MODEL

by

R. C. Farmer, W. R. Waldrop,

F. H. Pitts & K. R. Shah

Prepared under Contract No. NAS8-30330

Department of Chemical Engineering

Louisiana State University

Baton Rouge, La. 70803

January, 1975

This report was prepared by Louisiana State University under contract NAS8-30380, DEVELOPMENT OF A THREE-DIMENSIONAL FLOW FIELD MODEL, for the George C. Marshall Space Flight Center of the National Aeronautics and Space Administration. The contracting officer's representative was Mr. H. B. Wilson, Jr.

## ABSTRACT

A three-dimensional, time-dependent mathematical model to represent Mobile Bay was developed. The objective of this study was to develop computer programs which would numerically solve the appropriate conservation equations for predicting bay and estuary flow fields. The model will be most useful for analyzing the dispersion of sea water into fresh water and the transport of sediment. Also, the model serves as a useful tool for relating field and physical model data. The unique feature of this model is that it correctly accounts for momentum transfer in the governing flows; thereby, making it far more realistic than any previously devised. NASA's ERTS and Skylab programs resulted in high quality photographs of Mobile Bay. U.S. Army Corps of Engineers have also studied this bay extensively. All these data have been reviewed for comparison to this mathematical model.

# TABLE OF CONTENTS

## LIST OF FIGURES



## LIST OF TABLES

## NOMENCLATURE

English Letters:

| | |
|---|---|
| c | heat capacity |
| $C_1$, $C_2$ | stretching constants |
| E | sum of specific internal, kinetic, and potential energy |
| g | gravity force |
| h | free surface height |
| I | unit tensor |
| J | diffusion - flux with respect to number-average velocity |
| K | diffusion - flux with respect to mass-average velocity |
| M | molecular mass (or weight) |
| N | molarity |
| P | pressure |
| q | heat-flux vector |
| $q_r$ | heat-flux vector due to radiation |
| S | concentration of salt-water in a binary mixture of fresh water salt-water |
| t | time |
| T | temperature |
| U | number-average velocity |
| V | mass-average velocity |
| x, y, z | Cartesian coordinates |
| Y | transformed y coordinate |

Greek Letters:

| | |
|---|---|
| $\mu$ | viscosity |
| $\rho$ | density |
| $\tau$ | shear-stress tensor |
| $\Omega$ | earth's rotation vector |
| $\Phi$ | viscous generation of heat |

Subscripts:

| | |
|---|---|
| f | fresh-water property |
| s | sea-water property |
| x, y, z | component of property in specified direction |

Mathematical:

| | |
|---|---|
| - | vector |
| = | 2nd order tensor |
| $\nabla$ | del operator |
| $\Sigma$ | summation |
| $\partial$ | partial |
| D( )/Dt | substantial derivative with respect to number-average velocity |
| \| | variable preceding symbol is evaluated at the location which follows the symbol |

## INTRODUCTION

This report describes the development of a three-dimensional, time-dependent flow field model to represent Mobile Bay. This model is a mathematical representation of the appropriate conservation equations which has been evaluated with a digital computer. Since all important physical phenomena cannot be mathematically modeled in a definitive fashion, the errors introduced by approximations made to make the model tractable must be evaluated. Therefore, the model was prepared specifically to represent Mobile Bay.

Mobile Bay is not only a commercially important but also a technically interesting bay to study. The bay is connected by a tidal inlet to the Gulf of Mexico, thereby forming a region in which sea water is measurably diluted with land water drainage. Such a region is called an estuary. The drainage occurs by several large rivers discharging both water and sediment into the bay. A ship channel extends from end-to-end of the bay; the channel is dredged and maintained by the Corps of Engineers. The bay is shallow and therefore greatly affected by prevailing winds.

All estuaries are so complex that any mathematical model which may currently be implemented on a computer will contain certain simplifications to make a solution of the model tractable. These simplifications are so severe that once the model is developed it must be tested to prove that it is a valid representation of the estuary. Ideally, one would like to have sufficient experimental data from direct field measurements to test the model. Usually, sufficient data are not available. Even when they are, initial attempts at modeling will invariably show some discrepancy, then the model will have to be "improved".

Methods of making such improvements are not, usually, obvious, because the very complexity of estuaries defies establishing cause and effect phenomena. Results from physical model studies contribute much more to this stage of mathematical model development. This is due to the fact that physical model experiments can be designed to emphasize specific study of select phenomena, i.e. complicating factors can simply be eliminated in these studies. Use of only "real world" field data cannot be so utilized because our environment cannot be so controlled. Of course, physical model data is also often used without collaborating field or mathematical model data.

These general comments concerning the interactions of field, mathematical model, and physical model data precisely describe our current state-of-knowledge of Mobile Bay. Several attempts have been made to measure the flow properties within the bay; the most complete and recent study was performed by the Corps of Engineers to provide verification data for a physical model study which is being conducted, also by the Corps at Vicksburg, Mississippi. The purpose of the physical model experiments is to study the impact of opening another ship channel within Mobile Bay. Photographic data from NASA are also available.

This report summarizes the Corps of Engineers data pertinent to Mobile Bay, describes the mathematical model which has been developed to describe Mobile Bay, and compares data from all three sources.

## SUMMARY OF FIELD DATA

Results from several field studies of Mobile Bay have been reported (1, 2). Although these reports present many measurements descriptive of the flow-field, there is not sufficient information at any given instant of time to constitute adequate boundary conditions and check-points to be compared to a mathematical model. An adequate experiment was performed by the Mobile District Corps of Engineers for verifying the physical model study which will be described in the next section of this report.

These verification measurements (3) consisted of flow and tide level data through the four major river inlets to the north and the two major passes in the south for a complete tidal cycle. At the passes, vertical salinity profile data were also collected. Cross-sectional areas for the inlet and outlets were determined. Finally, tide, velocity, and salinity vertical-profile data were also taken at select stations within the bay. These regions are shown in Figure 1.

There are other rivers which were not measured during this study, presumably because there is little flow through them. However, what was measured corresponds exactly with the streams which are included in the physical model. Also, no data on winds or on flow into or out of the northern marshes are given. There is an airport weather station nearby from which some wind data may be obtained.

The four river inlets were taken to be where the highway crosses the north end of the bay. These inlets and cross-sectional areas were the: Mobile, 2,120 sq. m (22,810 sq. ft.); Tensaw, 2,850 sq. m (30,660 sq. ft.); Apalachee, 2,013 sq. m (21,660 sq. ft.); and Blakeley, 2,499 sq. m (26,880 sq. ft). These rivers are shown in Figure 2. Notice that part of the flow from the upper Mobile River passes into the bay through the Tensaw inlet, and that all of the flow from the upper Tensaw enters through the Tensaw, Apalachee, and Blakeley.

The mouth of Mobile Bay is 30,265 sq. m (325,600 sq. ft.) in cross-section. The pass connecting Mobile Bay to the Mississippi sound between Cedar Point and Dauphin Island is 6,144 sq. m (66,100 sq. ft.) in cross-section. All cross-sections are the area between mean-sea level and the bottom. The bottom location changes significantly with time, probably because of sediment deposition.

The previously mentioned data are available for further study. Only two points need further commment at this time. Low tide at

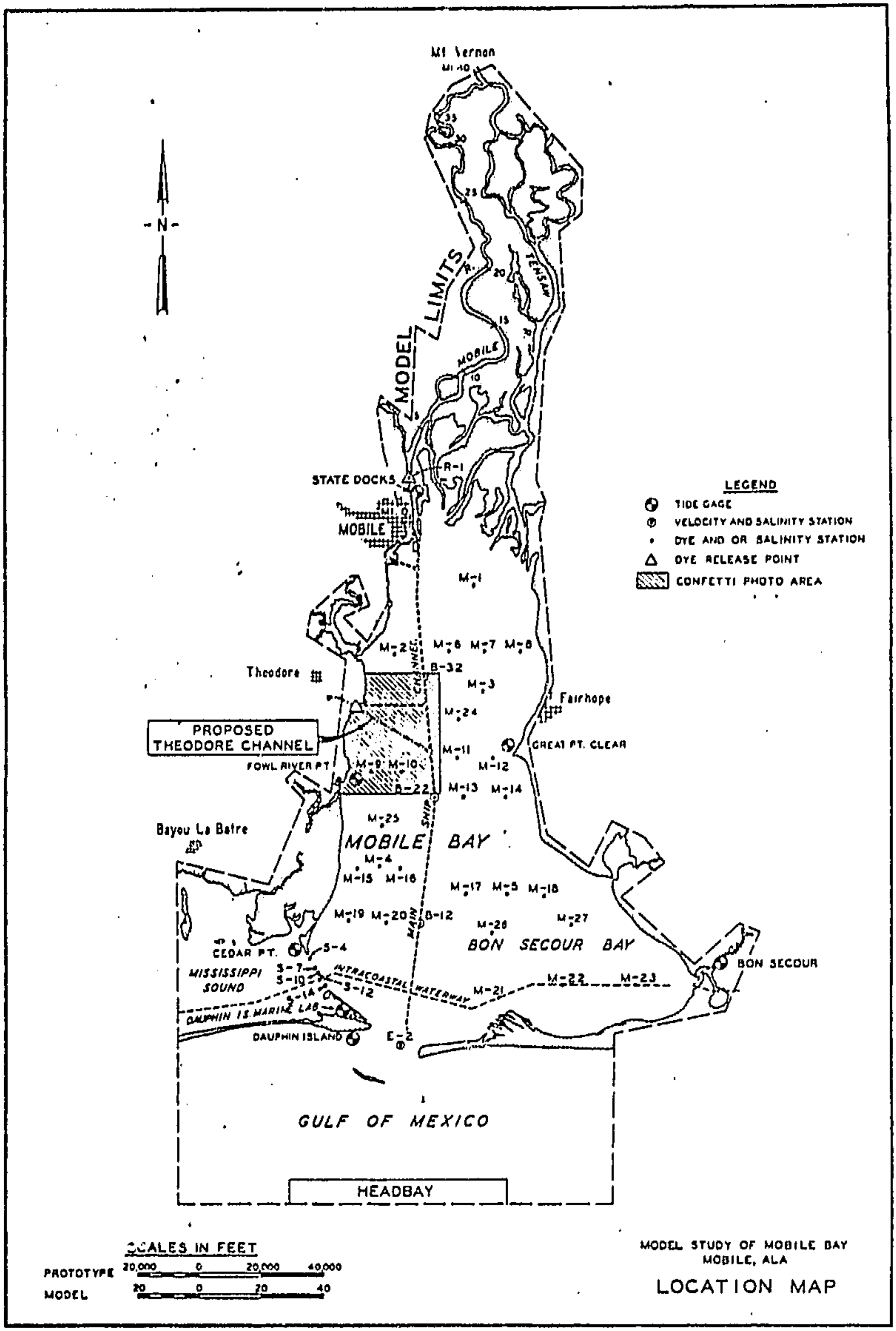


Figure 1. Mobile Bay

ORIGINAL PAGE IS
OF POOR QUALITY

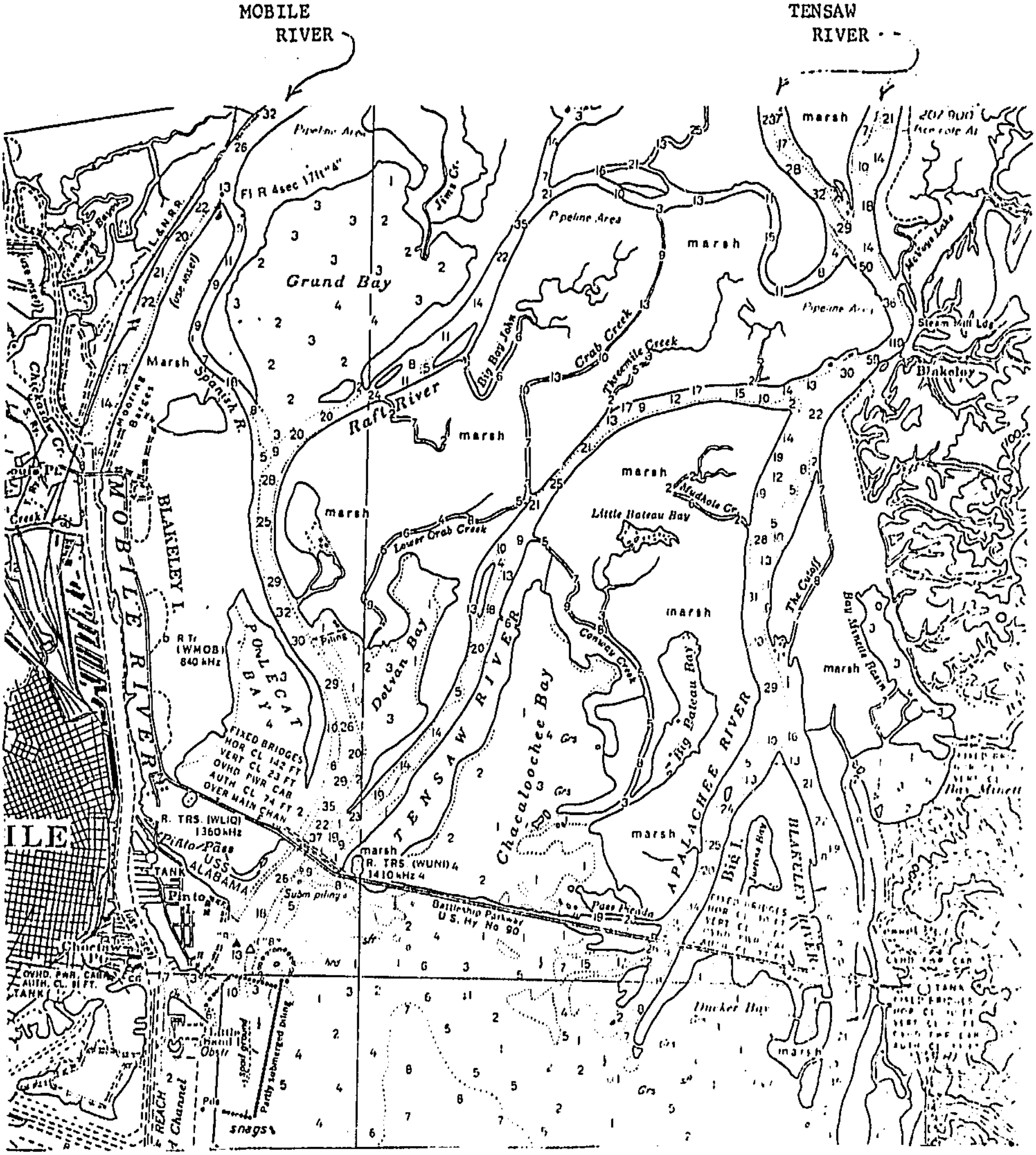


Figure 2. River Inlets

Mobile Point occurs some 4 hours before low tide at the Mobile River inlet; high tide about 2.5 hours before. This demonstrates that there are significant tidal phenomena within the bay.

The second point is that the high tide causes flow reversal in the river inlets at the north end of the bay. This implies that one would have to go up-stream before the rivers would be discharging, i.e. the bay, marshes, and lower river channels are filling with water on a rising tide. This fact complicates the mathematical specification of boundary conditions at the northern end of the bay. These marsh regions are clearly discernible in the ERTS IR photograph shown in Figure 3. The dark regions in the river valleys north of the bay are the marshes. Notice that these marshes do not extend down either side of the bay.

Figure 4 is a companion picture to Figure 3, but it was taken at visual wavelength. Although quantative data cannot be obtained from such photographs, the long distances over with the river plumes retain their shape and the dead water region in Bon Secour are clearly evident. Similar data in true color from Skylab photographs are also available.

## PHYSICAL MODEL DATA

In order to estimate possible effects caused by building a new ship channel, to be named Theodore Channel, the Corps of Engineers have performed physical model studies of Mobile Bay. First a set of base data for the Bay as it now exists was collected, then the new channel and several possible new islands which serve as spoilage banks were modeled and studied. The base data are of interest to this study; such data were made available to these investigators by the Mobile District Office of the Corps (4).

The region of the Bay which was physically modeled is shown in Figure 1. Salinities and velocities at the numbered surface locations during a tide cycle are reported for the "surface" and "bottom" elevations. The Mobile and Tensaw Rivers flows were metered and reported, presumably well up-stream of the northern reaches of the Bay. Note this does not correspond to the location where the field data are reported. The river volumetric flow rates were reported as (cubic feet/second) of prototype flow, and tide levels in (ft) of prototype surface elevation. The scaling laws used for such unit conversions are not presently known to these investigators. Note also should be made that "turbulence strips" were used in these model studies. Therefore, the physical model data will be considered as its representation of the actual Bay, which is how the modelers intended it should be used.

Tide data at the north, south, and middle points of the Bay are shown in Figure 5. Note that high tide occurs about 3 hours later at the state docks location than at Dauphin Island. This difference between physical model and field data is thought to be significant.

Figure 3. IR Photograph of Mobile Bay from NASA's ERTS Program

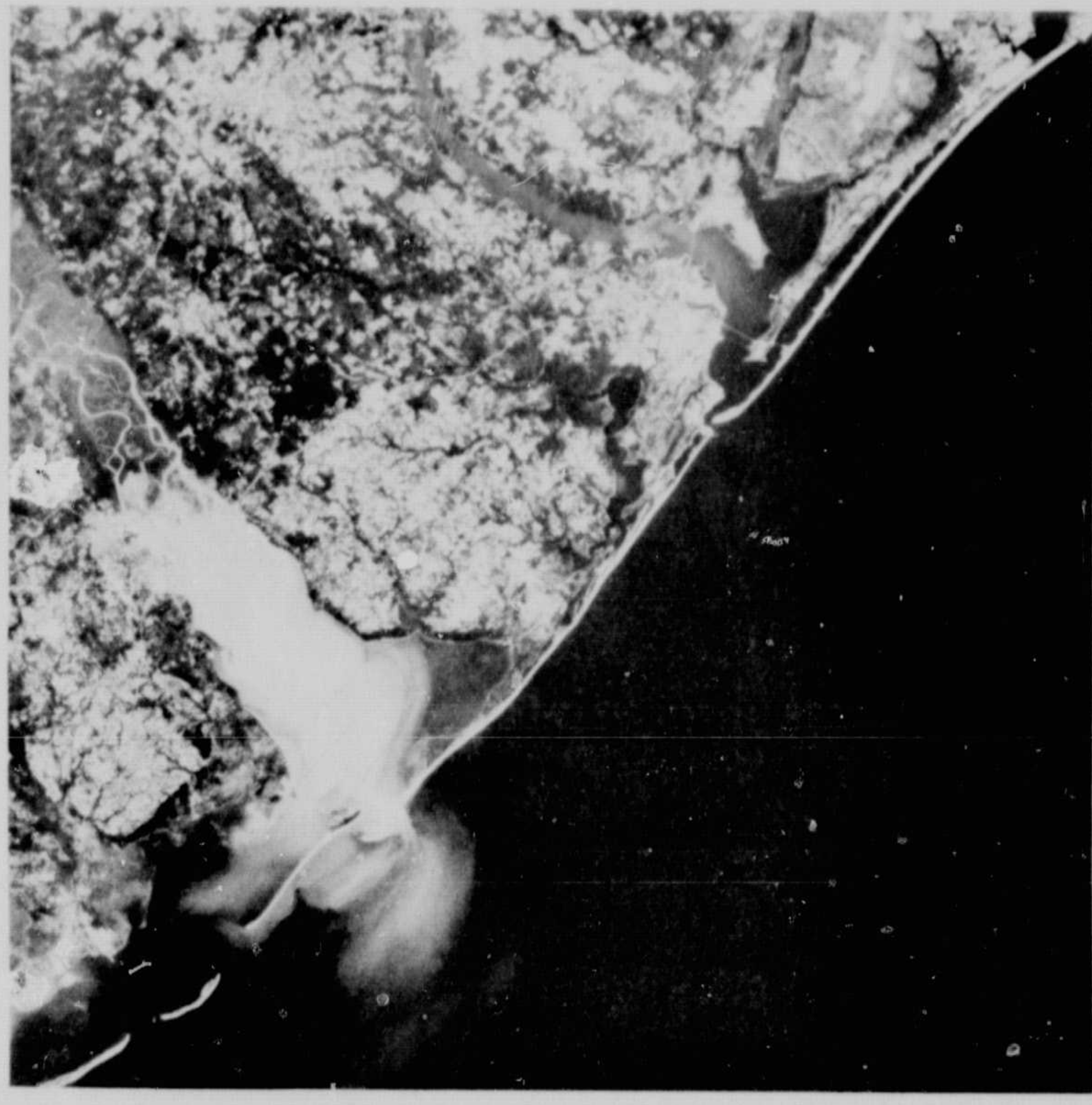

Figure 4. Black and White Photograph of Mobile Bay from NASA's ERTS Program

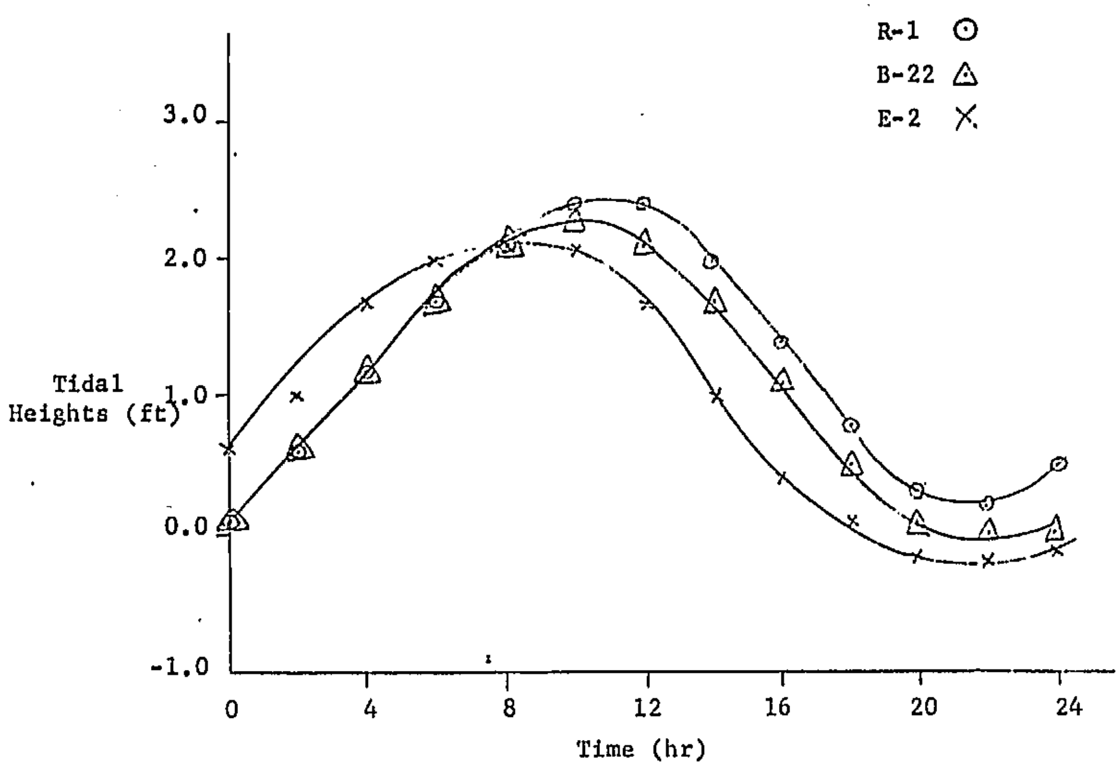


Figure 5. Tide Heights at Various Bay Stations

All velocities are reported as north or south velocity components only. Data are available at many more model stations than field stations. Interior stations at B-12 and B-32 are the only two field stations.

## THE MATHEMATICAL MODEL

The philosophy of developing and using a mathematical model is that a certain set of equations represents the phenomena of interest in the region of interest and that these equations may be solved for a particular set of auxiliary conditions. The solution would represent the behavior that one wishes to predict, and the auxiliary conditions and/or specified parameters in the equations would represent alternatives that one wishes to either understand or control.

Any and all errors which might be in the model would manifest themselves as inaccuracies in the flow description, hence, calculated results should be verified before a model is used. All known information concerning the flow region of interest should be considered to establish this verification. It is impossible to either completely measure or model a large, complex flow-field; therefore, model development and verification proceed in a "hand-over-fist" fashion. That is, as more is learned of the flow better predictions can be made, and as better predictions are made the more important regions of the flow can be identified for further field study.

Any model study is bounded by the desire for completeness and the necessity for economy; therefore, a modeling scheme which attemps to optimize this balance was developed. Since flow within the ship channel is believed to exert a dominating influence on the flow of sea water into the bay and on the sediment deposition trends within the bay, a three-dimensional, time dependent flow field model was deemed essential to describe the type flow found in Mobile Bay.

### Model Development

All solutions to fluid mechanics problems are based on conservation principles. These principles, usually expressed as a conservation of mass, momentum and energy, are presented in most standard texts of fluid mechanics for single-component flows. The phenomena of interest is that involving the relative motion and the mixing of fresh and saline water. Sediment transport is also of interest, but its study will be postponed to a future time.

Because the fluid in the mixing region may have two components some care must be exercised to insure that the most useful form of the pertinent conservation laws is determined. Also, it is fruitful to consider and compare experimental results and analytical methods which arise from studies of heated water discharges. The relationship between the mass density of fresh water and sea water and various mixtures of the same may be represented as a linear function of the amount of salt in the mixture, but the number of moles (or particles) in a unit volume is a constant. This means that any diffusion which occurs must behave as a replacement process, i.e. a molecule of "salt-water" replaces a molecule of fresh water. Furthermore, by introducing a fictitious molecular weight for salt water, the fluid may be described as a binary mixture of fresh water molecules and salt-water molecules. This behaviour of the number density allows one to specify the condition for incompressible flow by stating that the substantial derivative of the number density of fluid particles with respect to a number average velocity is zero, or that the divergence of the number average velocity is zero for an isothermal fluid. A similar relationship for the effect of modest temperature changes on the density of water exists. Now all the conservation laws may be stated.

Very generally, the conservation of mass equation states that the amount of mass accumulated in a given volume is equal to the difference between the mass convected out of the volume and the mass convected in. For multicomponent fluids, the net flux of mass into the volume by diffusion must also be considered.

The conservation of momentum is derived from Newton's second law of motion. Expressed for fluids, this states that the rate of change of momentum of the fluid in a given volume is equal to the summation of the vector forces acting on the fluid. Pertinent forces may consist of pressure forces, shearing stresses, and gravity forces. An apparent force known as the Coriolis force may also be included when the equation is expressed in a coordinate system which is moving relative to an inertially fixed system. Unfortunately, the momentum and energy equations are cumbersome when written in terms of the number average velocity; therefore, both the number average velocity and the mass average velocity will be used in formulating the conservation laws.

The conservation of energy equation results from applying the first law of thermodynamics to the moving fluid; it states that the rate of increase in total energy is equal to the sum of the rate of work done on the fluid and the rate of heat added from external sources. The total energy of a given volume of fluid consists of three types: internal energy which may be expressed as a function of the temperature, kinetic energy due to velocity of the fluid, and potential energy which is a function of elevation of the particles of fluid. The rate of work done to the fluid results from pressure forces, gravity forces, and viscous and turbulent shearing forces.

The energy equation reduces to a trivial form for incompressible, isothermal flows when velocities are such that there is little heat generated by viscous dissipation. Thus, the conservation of mass and momentum equations produce an independent and complete set.

In addition to the general conservation equations, an equation of state is required to express density as a function of composition and temperature. For liquids, density and temperature are usually independent of pressure, i.e. incompressible.

The conservation laws and equations are summarized in Table 1. Note that interpretation of the transport coefficients as either molecular or eddy values allows the stated equations to be used for both laminar and turbulent flows. The complete conservation equations just described represent a complex set of second order, nonlinear partial differential equations. Even so, these equations do not precisely describe certain types of flow. This inexactness is due to lack of properly specifying specific terms of these equations as certainly the basic laws must be observed. It is an easy matter to indicate turbulent transport with a coefficient; specifying the local value of such a coefficient is another matter entirely. Furthermore, the microscopic interaction of the fluid particles and solid material (i.e. sediment) is not completely known, and is not, therefore, represented by the equations in Table 1.

The actual solution of these equations will use an approximation presented by Frank-Kamenetskii (Ref 5) and utilized by Daly and Pracht (Ref 6) which states that negligible differences result from using the number average velocity instead of the mass average velocity in the conservation of momentum and energy equations. This then yields a set of equations with only one velocity.

Another approximation suggested by Yih (Ref. 7) is quite useful in simplifying the conservation equations. Yih recognizes that the coefficient of expansion $\partial\rho/\partial T$ is very small compared with unity for most liquids and, through a perturbation analysis, shows that $\nabla \cdot \overline{U} \cong 0$ when the range of temperatures are small relative to the mean absolute temperature. The implication of this assumption is that the volume dilation due to thermal expansion is negligible. This approximation is analogous to that involving the use of number-average velocity to simplify the solution of the two-component system of equations.

Before the equations are solved, one must choose a suitable coordinate system. A Cartesian system was chosen.

The resulting equations, including the approximations described above, which apply directly to non-isothermal, two-component fluid motions for the case of interest are may be solved by the method reported herein. The transport co-efficients may differ in each of the spatial directions. It should also be noted that the radiative heat flux vector $q_r$ of the energy equation need only be considered at the surface of the fluid; hence, it could be justly considered as a boundary condition. Furthermore, the two most common situations arise when the fluid is one-component ($S = 0$) or is isothermal ($T$ = constant); seldom must both variations be considered simultaneously.

Although the species continuity equation of Table 1 is written for the salt concentration S, it is equally applicable for the

Table 1. General Conservation Laws for Modeling Bays and Estuaries

Overall Continuity Equation.

$$\frac{\partial\rho}{\partial t} + \overline{U} \cdot \nabla\rho + \rho(\nabla\cdot\overline{U}) = \rho\nabla\cdot(\overline{U} - \overline{V}) \qquad (1)$$

Approximate Equation of State.

$$\rho = \rho_f + \left(\frac{\partial\rho}{\partial S}\right)_T S + \left(\frac{\partial\rho}{\partial T}\right)_S (T - T_o) \qquad (2)$$

Diffusive Behavior.

$$\rho\nabla\cdot(\overline{U} - \overline{V}) = \left(\frac{\partial\rho}{\partial S}\right)_T \left(\frac{\partial S}{\partial t} + \overline{U}\cdot\nabla S\right) \qquad (3)$$

Compressive Behavior.

$$\rho(\nabla\cdot\overline{U}) = -\left(\frac{\partial\rho}{\partial T}\right)_S \left(\frac{\partial T}{\partial t} + \overline{U}\cdot\nabla T\right) \qquad (4)$$

Momentum Equation.

$$\frac{\partial(\rho\overline{V})}{\partial t} + \overline{V}\cdot(\nabla\rho\overline{V}) = -\nabla P + \nabla\cdot\mu\nabla\overline{V} - \rho\overline{g} - \rho(2\Omega\times\overline{V}) \qquad (5)$$

Energy Equation.

$$\frac{\partial(\rho E)}{\partial t} + \nabla\cdot(\rho E\overline{V}) + \nabla\cdot\overline{q} + \nabla\cdot(\bar{\bar{\tau}} - \bar{\bar{I}}P)\cdot\overline{V} - \Sigma\overline{g}_i\cdot\overline{J}_i = 0 \qquad (6)$$

Relationship Between Mass-Average and Number-Average Velocity.

$$\overline{U} - \overline{V} = \overline{J}_s/[S + (M_s\rho)/(M_f - M_s)] \qquad (7)$$

Relationship Between Mass Flux Vector and Molar Flux Vector.

$$\overline{J}_s = \overline{K}_s[1 + (S/\rho)(M_f - M_s)/M_s] \qquad (8)$$

conservation of any chemical or biological species with the inclusion of a rate of generation (or degeneration) term. These equations could then be applied to the dispersion of any pollutant discharged into a river or bay. Should the species have a negligible effect upon the density as is often the case, then the species continuity equation could be decoupled from the other conservation equations. For this case, the flow field could be computed and the resulting velocities used to compute the advection, diffusion and generation of the species. For cases where the density is affected by the concentration of the species, this decoupling is not applicable. The species continuity equation should then be solved simultaneously with the other conservation equations.

To model Mobile Bay, the flow is considered isothermal and to be a binary mixture of salt and water; therefore, the equations given in Table 2 apply. Furthermore, the viscous terms in the z-momentum equation were also negleted, since an order of magnitude analysis showed them to be two orders lower than the pressure and gravity terms in this equation.

## Solution Technique

The set of equations as presented in Table 2 is sufficient to describe all the dependent variables for a given set of eddy coefficients, but the equations are in a form for which there is no known analytical solution. These equations will be approximated by a set of finite-difference equations and solved by using a time-dependent technique. Before writing the equations in finite-difference form, a stretching transformation that will create a more efficient use of grid points is suggested. For example, by letting

$$y = \frac{1}{C_1} \tan \frac{Y}{C_2} \qquad (15)$$

where $C_1$ and $C_2$ are arbitrary constants, the equations of Table 2 can be expressed as functions of Y instead of y. The implications of this transformation can be more easily understood by rewriting it as

$$Y = C_2 \tan^{-1} (C_1 y) \qquad (16)$$

When $y = 0$, then $Y = 0$. Grid planes located at even increments of $\Delta Y$ will produce ever-increasing increments of $\Delta y$. This tranformation will increase resolution where it is most needed and make specifying boundary conditions less critical.

The philosophy of using this numerical procedure is to begin with a set of initial values of the dependent variables at all grid intersections and allow the flow field to adjust asymtotically until all the conservation equations and boundary conditions are satisfied. The final result will be a steady-state flow field. Naturally, the better the estimate for the initial data, the more rapid the convergence, but even a crude guess, one conveniently

Table 2. Specific Conservation Laws for the Proposed Mobile Bay Model

Species Continuity Equation.

$$\frac{DS}{Dt} = D_x \frac{\partial^2 S}{\partial x^2} + D_y \frac{\partial^2 S}{\partial y^2} + D_z \frac{\partial^2 S}{\partial z^2} \qquad (9)$$

Equation of State.

$$\rho = \rho_f + \left(\frac{\partial \rho}{\partial S}\right)_T S \qquad (10)$$

Compressive Behavior.

$$\nabla \cdot \overline{U} = \frac{\partial U_x}{\partial x} + \frac{\partial U_y}{\partial y} + \frac{\partial U_z}{\partial z} = 0 \qquad (11)$$

Momentum Equation.

$$\frac{D(\rho U_x)}{Dt} = -\frac{\partial P}{\partial x} + \frac{\partial}{\partial x}\left(\mu_x \frac{\partial U_x}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu_y \frac{\partial U_x}{\partial y}\right) + \frac{\partial}{\partial z}\left(\mu_z \frac{\partial U_x}{\partial z}\right) + 2\rho\Omega_z U_y \qquad (12)$$

$$\frac{D(\rho U_y)}{Dt} = -\frac{\partial P}{\partial y} + \frac{\partial}{\partial x}\left(\mu_x \frac{\partial U_y}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu_y \frac{\partial U_y}{\partial y}\right) + \frac{\partial}{\partial z}\left(\mu_z \frac{\partial U_y}{\partial z}\right) - 2\rho\Omega_z U_x \qquad (13)$$

$$\frac{D(\rho U_z)}{Dt} = -\frac{\partial P}{\partial z} - \rho g_z \qquad (14)$$

specified, is sufficient. The boundary conditions will then be replaced with conditions which depend on time, and the asymptotic solution will become an initial condition. An unsteady solution will result from further calculations. An efficient method for specifying initial conditions when making a series of calculations with similar geometry is to store the results of one run on tape and recall these data as initial conditions for the next run. Experience has shown that this can reduce computation times for subsequent runs by a factor of four.

The numerical technique for calculating the dependent variables actually involves a sequence of calculations for each time step. The first step of the sequence involves computing new values of $U_x$, $U_y$, and S (and T, when needed) at each grid intersection by stepping forward in time by $\Delta t$. This is best explained by considering the following truncated Taylor series:

$$U_x\{n+1,i,j,k\} = U_x\{n,i,j,k\} + \frac{\partial U_x}{\partial t}\{n,i,j,k\}\,\Delta t \tag{17}$$

The braces indicate functionality, n is the index of the time step, and i,j,k indicate specific x,Y, and z grid planes, respectively. A finite difference approximation to the transformed right-hand side of the x-momentum equation of Table 2 can be substituted for $\partial U_x/\partial t$. Similar expressions can be written for $U_Y$ and S or T.

The maximum possible time step for this type of numerical scheme is approximately given by the Courant-Friedricks-Lewy (CFL) condition, but in actuality this maximum time step is seldom obtained. There are various procedures for increasing the stable time step of a truncated Taylor's series such as the one shown, but they all pay a penalty in computation time per step and/or computer core storage. A method which has proven quite useful to these authors is best explained by considering another truncated Taylor series:

$$U_x\{n+1,i,j,k\} = U_x\{n,i,j,k\} + \left[\frac{\partial U_x}{\partial t}\{n,i,j,k\}\right]\Delta t + \left[\frac{\partial^2 U_x}{\partial t^2}\{n,i,j,k\}\right]\frac{(\Delta t)^2}{2} \tag{18}$$

The braces indicate functionality, n is the index of the time step, and i,j,k indicate specific x,Y, and z grid planes, respectively. When $(\partial^2 U_x)/(\partial t^2)$ is approximated by a forward difference as a function of $(\partial U_x)/(\partial t)$, one has

$$U_x\{n+1,i,j,k\} = U_x\{n,i,j,k\} + \left[\frac{\partial U_x}{\partial t}\{n,i,j,i\} + \frac{\partial U_x}{\partial t}\{n+1,i,j,k\}\right]\frac{\Delta t}{2} \tag{19}$$

The $(\partial U_x)/(\partial t)$ {n,i,j,k} can easily be computed from a finite-difference approximation to Equation 12. Upwind differencing was used for the convective terms. For the case of positive $U_x$ and negative $U_Y$ and $U_z$ components of velocity at grid point i,j,k, the finite difference approximation of Equation 12 becomes

$$\frac{\partial U_x}{\partial t}\{n,i,j,k\} = -\Big[\frac{1}{\Delta x}\Big([U_x^2\{n,i+1,j,k\} - U_x^2\{n,i,j,k\}]$$

$$- \frac{0.5}{\rho\{n,i,j,k\}}[P\{n,i+1,j,i\} - P\{n,i-1,j,k\}]\Big)$$

$$+ \frac{Y\{j\}}{\Delta y}(U_x\{n,i,j,k\}\ U_Y\{n,i,j,k\} - U_x\{n,i,j-1,k\}\ U_Y\{n,i,j-1,k\})$$

$$+ \frac{1}{\Delta z}(U_x\{n,i,j,k\}\ U_z\{n,i,j,k\} - U_x\{n,i,j,k-1\}\ U_z\{n,i,j,k-1\})]$$

$$+ \epsilon_H\Big[\frac{1}{(\Delta x)^2}(U_x\{n,i+1,j,k\} - 2U_x\{n,i,j,k\} + U_x\{n,i-1,j,k\})$$

$$+ \frac{Y'\{j\}}{\Delta y}^2(U_x\{n,i,j+1,k\} - 2U_x\{n,i,j,k\} + U_x\{n,i,j-1,k\})$$

$$+ \frac{Y''\{j\}}{2\Delta y}(U_x\{n,i,j+1,k\} - U_x\{n,i,j-1,k\})]$$

$$+ \epsilon_z\Big[\frac{1}{(\Delta z)^2}(U_x\{n,i,j,k+1\} - 2U_x\{n,i,j,k\} + U_x\{n,i,j,k-1\})] \quad (20)$$

where

$$Y'\{j\} = \frac{dY}{dy}\{j\}$$

and

$$Y''\{j\} = \frac{d^2Y}{dy^2}\{j\}$$

Computing $\partial U_x/\partial t\{n+1,i,j,k\}$ is more difficult because dependent variables at time n+1 are required but are not known. For this reason, solutions of Equation 19 usually fall into two classes: the implicit schemes which involve matrix inversion, and two-step schemes which estimate provisional values at time step n+1. From a practical stand-point, the primary reason for including $(\partial^2 U_x)/(\partial t^2)$ term of Equation 18 is to dampen numerical oscillations of $(\partial U_x)/(\partial t)$ by averaging gradients at time n and n+1, as shown in Equation 19. This effect may also be effectively accomplished by averaging gradients at times n and n-1, which would give

$$U_x\{n+1,i,j,k\} = U_x\{n,i,j,k\} + \Big[\frac{\partial U_x}{\partial t}\{n,i,j,k\} + \frac{\partial U_x}{\partial t}\{n-1,i,j,k\}\Big]\frac{\Delta t}{2} \quad (21)$$

Equation 21 was, therefore an efficient method for computing $U_x\{n+1,i,j,k\}$. Since $(\partial U_x)/(\partial t)$ $\{n-1,i,j,k\}$ was already available, less manipulation per time step was required than for other commonly used procedures. Equation 21 provided stability comparable to that of a two-step technique. Both techniques have time step limitations as determined by the Courant-Friedericks-Lewy conditions.

It is conceded that using Equation 21 instead of Equation 19 possibly sacrifices some accuracy in describing unsteady behavior, but if a steady-state solution is the desideratum, then lagging $(\partial u)/(\partial t)$ by half a time step is inconsequential because all partials with respect to time should asymptotically approach zero.

Once new values of $U_x$ and $U_Y$ have been computed at each grid intersection $U_z$ may be calculated from a finite-difference representation of equation 11. Spatial marching can be used in the vertical direction, beginning at the bottom, where $U_z = 0$.

The free surface height h at each horizontal grid intersection may then be calculated. The incompressibility of the fluid requires that the net mass of fluid convected into a volume of fixed horizontal dimensions near the surface has to be accompanied by a comparable adjustment to the surface height of the volume. Since $U_x$, $U_Y$ and $U_z$ at time n+1 have already been computed in the previous sequence, sufficient data are available to calculate $(\partial h)/(\partial t)$ $\{n+1,i,j\}$. The new surface height can be computed from:

$$h\{n+1,i,j\} = h\{n,i,j\} + \left[\frac{\partial h}{\partial t}\{n,i,j\} + \frac{\partial h}{\partial t}\{n+1,i,j\}\right]\frac{\Delta t}{2} \qquad (22)$$

The gradient $(\partial h)/(\partial t)\{n,i,j\}$ will be known from the previous time step. Note that h is related to the size of the grid cells which contain the surface; therefore, these cells change volume during the course of the calculations.

The size of the surface elements may be accounted for by identifying $(h - z_{max-1}) \equiv a$ and then writing the divergence and momentum equations as follows:

$$\frac{U_{xi} - U_{xi-1}}{x_i - x_{i-1}} + \frac{U_{yj} - U_{yj-1}}{y_j - y_{j-1}} + \frac{U_{zh} - U_{zmax-1}}{a} + \frac{\Delta a}{a\Delta t} = 0 \qquad (23)$$

x - momentum (y - momentum is transformed version of this equation)

$$\frac{\langle\rho a\rangle_t}{a}\frac{\Delta U_x}{\Delta t} + \langle\rho U_x\rangle_x\left(\frac{U_{xi} - U_{xi-1}}{x_i - x_{i-1}}\right) + \langle\rho U_y\rangle_y\left(\frac{U_{yj} - U_{yj-1}}{y_j - y_{j-1}}\right)$$
$$+ \langle\rho U_z\rangle_z\left(\frac{U_{xh} - U_{xzmax-1}}{a}\right) = -\frac{(P_{xi+1} - P_{xi-1})}{2\Delta x} + \left(\frac{\tau_{xx}|_{xi} - \tau_{xx}|_{xi-1}}{x_i - x_{i-1}}\right)$$
$$+ \left(\frac{\tau_{yx}|_{yj} - \tau_{yx}|_{yj-1}}{y_j - y_{j-1}}\right) + \left(\frac{\tau_{zx}|_h - \tau_{zx}|_{zmax-1}}{a}\right) \qquad (24)$$

z - momentum

$$\frac{\langle \rho a \rangle_t}{a} \frac{\Delta U_z}{\Delta t} + \langle \rho U_z \rangle_z \left( \frac{U_{zh} - U_{zmax-1}}{a} \right) + \frac{\langle \rho U_y \rangle_y \left( U_z \big|_{yj} - U_z \big|_{yj-1} \right)}{y_j - y_{j-1}}$$

$$+ \langle \rho U_x \rangle_x \left( \frac{U_z \big|_{xi} - U_z \big|_{xi-1}}{x_i - x_{i-1}} \right) = - \left( \frac{P_h - P_{zmax-1}}{a} \right) - \rho g \qquad (25)$$

where $\langle \; \rangle$ means the average of the quantity within the brackets in the direction of the subscript on the last bracket. The stretched coordinate and the salt continuity equation could have been shown also but these add nothing to the discussion at this point. If $h = zmax$, $a = \Delta z$, these equations are identical to the interior equations. If the surface does not coincide with a z grid point (as it never would), the $U_z|_h = 0$. This is the normal behavior of these equations. For convenience, a certain number of z-grid points, zmax, are always used, in which case a might be negative. This feature imposed the condition that "a" never excedes $[z_{max} - z_{max-1}]$ in magnitude. This creates no problem; it is merely described so that the logic of the calculation may be understood.

Pressure, the last dependent variable to be calculated in the sequence of each time step, can be computed by a spatial marching procedure similar to that used to calculate $U_z$. At the free surface the pressure will be known, and pressure at lower grid rows can be computed from an integrated form of the z-momentum equation. An initial adjustment must be made to the pressure at the top grid plane to account for the free surface height.

Boundary Conditions

Boundary conditions for each dependent variable must be enforced after the sequence in which the variable is computed during the time step. Boundary conditions were applied as shown in Figure 6.

A no-slip condition was enforced along the bottom by forcing the velocity at grid points adjacent to the bottom to fit a boundary-layer-type profile. Grid points lying under the bottom were bypassed. The vertical shear component was set equal to zero as a boundary condition. Wind shear at the surface was empirically related to wind velocity. Heat flux to and from the atmosphere, although neglected in this analysis, could have been included as surface boundary conditions.

Reflection principles were used along the bay banks, where it was felt that the coarse horizontal grid spacing was insufficient to enforce a no-slip condition.

Velocities through the tidal inlets were obtained by extrapolation, for a given tide level and cut-point on the salinity distribution. The cut-point is defined as a level below which there is pure salt water. These conditions could be held constant with

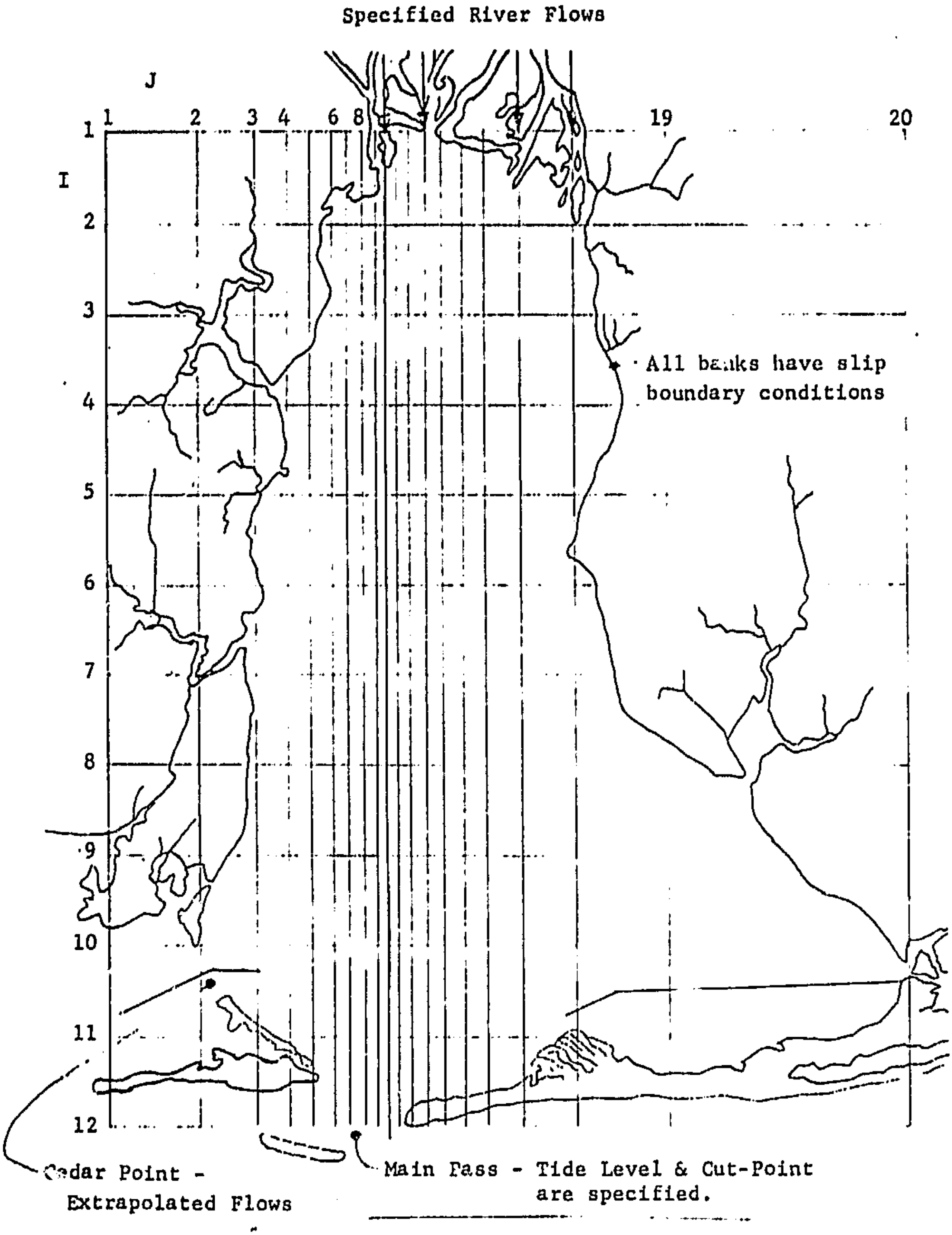


Figure 4a. Boundary Conditions and Grid Locations.

CROSS-SECTIONS PERPENDICULAR
TO X AXIS

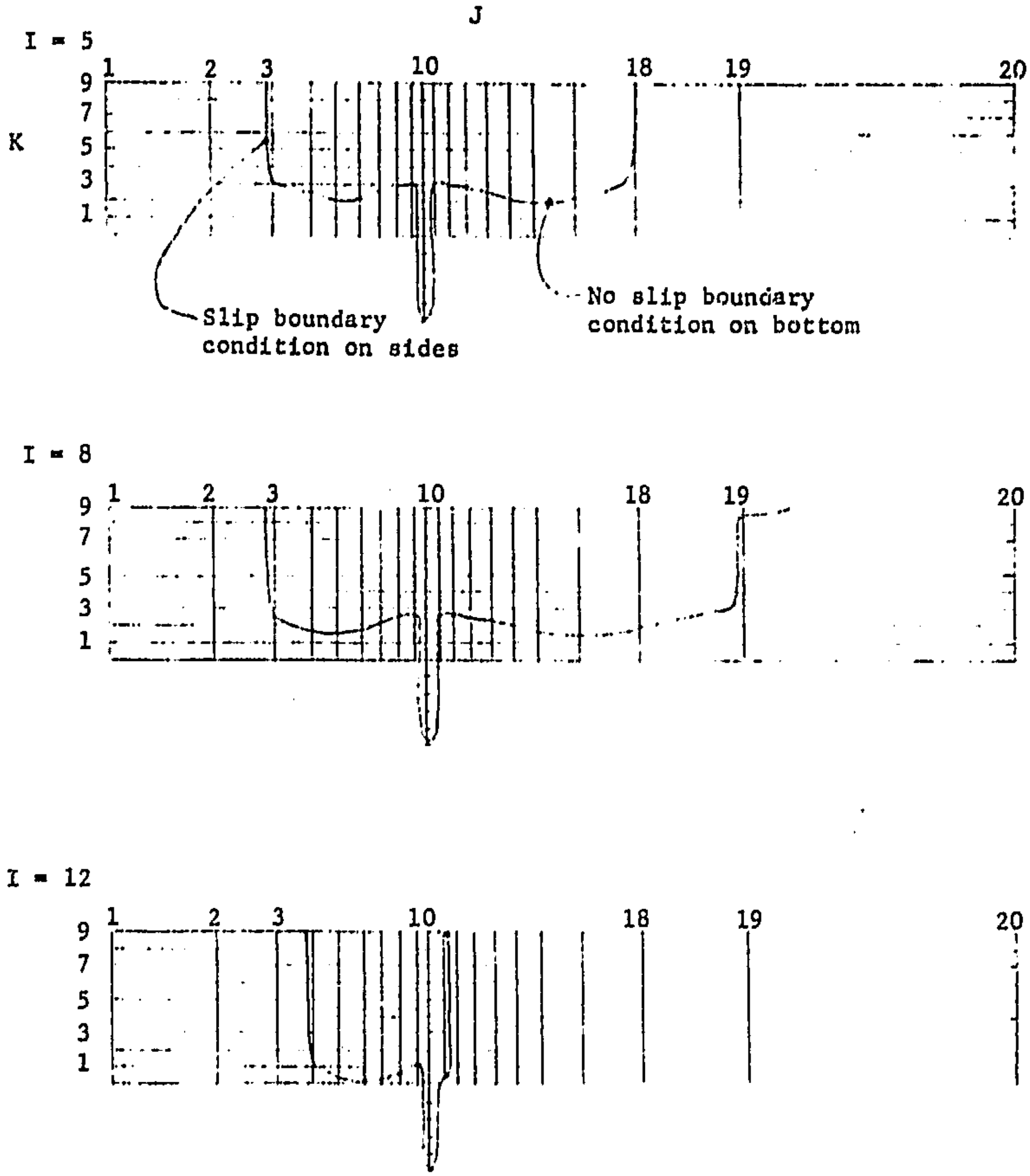




Figure 6b. The Vertical Grid System

time to obtain a steady-state solution, or varied as a known function of time to obtain a time-dependent solution. The tidal effects will be modeled by imposing time varying boundary conditions at the seaward inlets to the harbor. Over part of the tide cycle the river flows will also have to be time dependent. Flows to and from marshes should also be included as part of the specified boundary conditions. These latter two effects have not yet been included in the model.

## RESULTS

The equations presented in the previous chapter constitute the mathematical model of Mobile Bay. The solution technique presented has been implemented with a computer program; this program is listed in the Appendix. Mathematically, the model is a set of partial differential equations which are solved for a particular set of auxiliary conditions. Since the desired solution is a function of time, initial conditions throughout the bay are required; these are not available. In fact, one of the major benefits of using a model is to first predict a quasi-steady or a periodic motion such that an accurate statement of initial conditions is not required. A proposed scheme of generating the necessary initial conditions is presented. The speed and economy of obtaining a solution with this method can now be determined. Initial calculations indicate that more improvement in efficiency is warranted before extensive calculations are made. Several methods of accomplishing this improvement are outlined.

The concept for modeling Mobile Bay is that the three-dimensional, unsteady conservation laws are solved for a set of boundary conditions and an arbitrary set of assumed initial conditions. The boundary conditions are held fixed such that the solution to the equation is driven to an asymptotic limit. This limit then becomes an initial value for a truly unsteady solution. This idea was presented in (8) and applied to modeling a bayou flow problem in (9). The river flows are held fixed. The tide level and a cut-point below which there is only sea water is fixed at the Mobile Point entrance to the bay. Flow through the Cedar Point - Dauphin Island entrance is obtained by extrapolation. These boundary conditions and an initial estimate of all dependent variables are sufficient to begin the sequence of calculations.

The program to perform these calculations is listed as CHANNEL. CHANNEL describes the bay for the stated type of boundary conditions and, in addition, places a specified wind shear on the surface and a no slip condition on the bottom. CHANNEL is a combination of MOBILE 2 and the bayou program from ref. (9). The bayou calculation is included as a description of the ship channel. MOBILE 2* is also

*The 2 in its name means it is the second grid configuration which has been used. Generally, there is no difficulty in changing the number of grid points in a calculation, but for this bay calculation some care is necessary, because internal program logic requires that adjacent boundary grids differ by no more than one. This means that the ratios of grid points used should be kept roughly the same.

a three-dimensional, unsteady bay model, but the somewhat more simple bottom location specification allows an overall more simple solution. This development of programs by addition of pre-tested subroutines to a basic main program is by design; this allows one to test and incorporate " " features into the calculation without undue program rewriting.

It should be noted that CHANNEL is the culmination of several previous studies which has resulted in significant improvements in this basic analytical approach. Specifically, these improvements are:

1) An equation set which has been optimized to run faster by eliminating terms which an order of magnitude analysis showed to be of neglible magnitude.

2) A windward differencing scheme was used since it appears to be the fastest explicit scheme yet devised.

3) Wind-shear is included as a surface boundary condition.

4) The method of applying tidal conditions has been proved to be adequate on the more simple bayou analysis.

Starting with these program features a grid system was devèloped for Mobile Bay as shown in Fig. 6. Calculations were then initiated to describe the Bay.

A zero mean-sea-level tide condition and a set-point corresponding to field observations at Mobile Point were used as boundary conditions to seek an asymptotic solution. The constancy and smallness of the program generated error functions indicate that suitable progress has been made to begin seeking an unsteady solution. The error functions (SBAR, UBAR, VBAR) are gross measures of the changes in S, $U_x$, $U_y$ between computation steps.

Study of the calculations and review of som. of the development steps necessary to obtain these calculations yield the following results.

Stable calculation time is about 0.5 seconds. Modest changes in the grid system point density and using MOBILE 2 rather than CHANNEL does not appreciably change this step size; however, both of these changes do affect the total amount of time necessary to perform a given number of calculation steps. The significant factor which must be considered is the amount of real time which is modeled in a unit of computation time. For CHANNEL and MOBILE 2, these numbers are, about $6^{-1}$, for grid system number 2 (12 x 20 x 9). (Actually, the nine is an eight in MOBILE 2.) Grid system 1 (17 x 30 x 15) gives about $25^{-1}$ as the real time to computing time ratio.

Data at intermediate numbers of calculation steps reveal that velocity and surface elevation fields establish before concentration fields. As expected for the case run the velocities are small, probably because the major driving force in creating flow is the tidal oscillation which have not yet been calculated.

Initially guessed fields take a long time to change - of the order of hours of real time.

The most significant result is that the program has been developed, is running without apparent errors, and is responding as expected to applied boundary conditions.

An unsteady solution has not yet been sought, mainly because the program has only now been developed and run to obtain these initial conditions. Secondly, several observations can be made on the calculations and available experimental data which have a bearing on the immediate steps which need to be made in researching this problem.

River flows as boundary conditions need to be specified so that they can reverse. This means specifying them as a function of time during a tide cycle. Experimental field data at one river stage condition are available and can be used for such a specification.

The ratio of real time to computer time needs to be improved by some or all of the following means.

1) Determining if using a larger number of time steps in the species equation than in the momentum equations improves computational efficiency.

2) Running on a faster computer. Some runs on machines available to NASA would expedite this study.

3) Transform the time variable to reduce the real time length of a tide cycle. Reductions in cycle from 24 hours to about 4 (the lag time for a tidal flucturation to travel from the south to the north end of the Bay) should prove feasible.

4) Devise an improved computation procedure. Possibilities from the simple expedient of devising a new guessing procedure for initialization to using a new differencing formulation will be considered.

In summary, a mathematical model of Mobile Bay has been developed. This model is now ready to use for parametric studies to predict the bay's dynamic behavior. Economy dictates that the exact procedure for accomplishing the required computation be carefully planned and executed.

REFERENCES


1. Austin, G. B., "On the Circulation and Tidal Flushing of Mobile Bay, Alabama, Part I", Ref. 54-20T, Texas A&M, April, 1954.

2. McPhearson, Jr., R. M., "The Hydrography of Mobile Bay and Mississippi Sound, Alabama", J. Marine Science, 1, pp. 1-83, August, 1970.

3. Kerr, A. W., Mobile District, Corps of Engineers personal communications of September 10 and November 14, 1974.

4. Drago, E. A., Mobile District, Corps of Engineers, personal communication of August 16, 1974.

5. Frank-Kamenetskii, D. A., translated by J. P. Appleton, Diffusion and Heat Transfer in Chemical Kinetics, Plenum Press, New York, 1969.

6. Daly, B. J. and W. E. Pracht, "Numerical Study of Density-Current Surges", The Physics of Fluids, 11, pp. 15-30, 1968.

7. Yih, Chia-Shun, Dynamics of Nonhomogeneous Fluids, The MacMillan Co., New York, 1965.

8. Waldrop, W. R. and R. C. Farmer, "Three-Dimensional Flow and Sediment Transport at River Mouths", Louisiana State University. Coastal Studies Institute, TR-150 Sept. 1973.

9. Waldrop, W. R., R. C. Farmer and P. A. Bryant, "Saltwater Intrusion into a Flowing Stream", Louisiana State University. Coastal Studies Institute, TR-161, May, 1974.

## APPENDIX

Discussion of the Program: This appendix describes a computer program which can be used for modeling a bay or estuary with a shipping channel. The mathematical model used in this program is a three-dimensional, time-dependent flow field model which can be used to describe the type flow found in any bay or estuary by changing appropriate boundary conditions and geometry.

Mobile Bay was selected as the bay to be modeled. The computational techniques and equations solved are presented in the chapter on model techniques of the main paper. Comment cards are used throughout in the computer program to indicate which operations are being performed. A simplified flow sheet is presented in Figure 1.

Input Guide: All data cards to be used in the program are read by SUBROUTINE PRELIM. Proceeding each READ statement are comment cards which define each term to be read and specify the appropriate units. These data are nondimensionalized before using them for computations. Two basic formats are used for input, I10 and F10.5. The I10 is an integer format consisting of 10 columns right justified; the F10.5 is a floating point format occupying 10 columns with a decimal point punched on the card.

At the completion of the run, data are written on a tape in SUBROUTINE ECRIRE. This provides data for a restart capability or a plot program.

When the restart capability is used, data are read from a tape by SUBROUTINE LIRE. The option of calling this subroutine is controlled by a data card as described in SUBROUTINE PRELIM.

Table 1 provides the details of the card input.

Description of Printout: This section presents a description of the CHANNEL program output.

Initially, the input parameters are printed in order to identify the problem and to verify that correct values were used. Then a complete description of the grid systems in the three coordinate directions are printed. This description includes index numbers and dimensionless and dimensional values for the locations of the grid planes as well as stretching function values where applicable.

The physical location of the boundaries of the bay are printed as it was read into the program and as the program describes it in terms of the above grid systems. If desired, a description of the floor of the bay can be printed. This portion of the printout is generally suppressed by branching around it with a GØ TØ statement because of its excessive length and low usage. This same comment also applies to a portion of the printout that describes the heights of the banks along the boundaries of the bay. Then the maximum velocities, east and west bank locations and depths of the rivers entering the north end of the bay are printed.

Next, terms monitoring the overall changes in the salinities and horizontal components of velocity for a single time step are printed for selected time steps. If the number of time steps specified in the input data is achieved, a message stating that a stable run was made is given and then values of the flow field variables are printed at selected Y-Z grid planes[1] with velocities in M/S, salinities as fractions of Gulf water salinity and pressures in non-dimensional form. If the surface becomes too rough or a negative pressure below the surface is encountered, a message identifying the cause for premature termination along with its location in the grid system is given. Values of the flow field variables are then printed as above. Finally, the free surface heights relative to the top horizontal grid plane are printed in CM in the form of a map for ease of interpretation.

Program Listing: The following pages contain a listing of CHANNEL and the required input data.

[1]Which Y-Z grid planes are selected is controlled by LOGICAL IF statements that can be modified as needed.

FIGURE 1

SIMPLIFIED FLOW CHART FOR CHANNEL

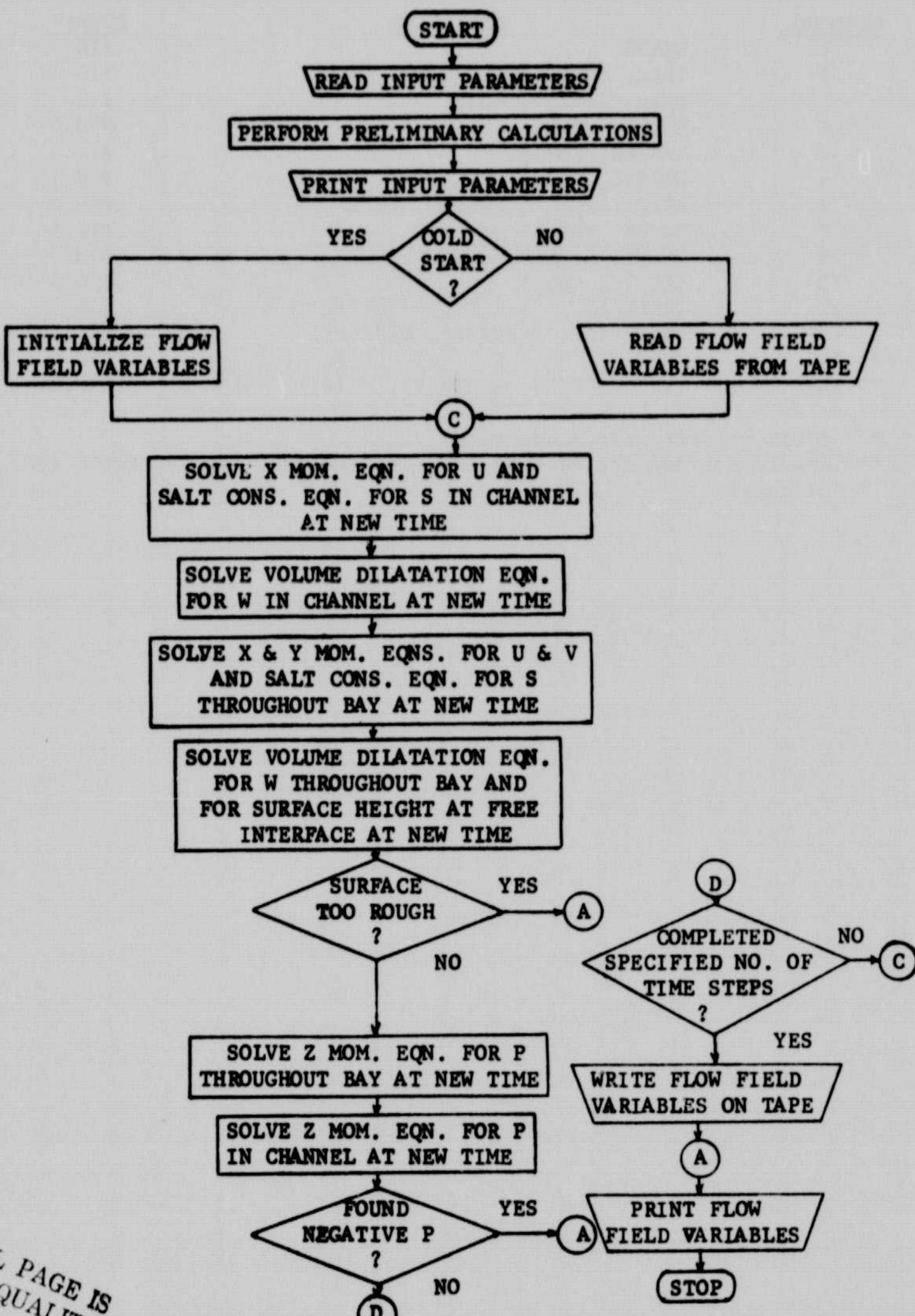

TABLE 1
INPUT PARAMETERS FOR CHANNEL

| CARD NO. | VARIABLES | FORMAT |
|---|---|---|
| 1 | LTAPE | I10 |
| 2 | IMAX, JMAX, KMAX, JWAG, KMXC | I10 |
| 3 | NMAX, NPRINT, NBAR | I10 |
| 4 | FETCH, YMAX | F10.5 |
| 5 | XBANKN, XBANKS | F10.5 |
| 6 | DTREAL, YK2, ACCEL | F10.5 |
| 7 | HREF, SIGMAT, PERIØD, TIDE | F10.5 |
| 8 | ROUGH, VØNKAR | F10.5 |
| 9 | VWIND, DIRECT, PHI | F10.5 |
| 10 | CDEPTH, CUTPT | F10.5 |
| 11 | IIMAX | I10 |
| 12* | SØUTH(II), EAST(II), WEST(II) | F10.5 |
| 13 | CØNVRT | F10.5 |
| 14** | YW(I), YE(I), DEPTH(I), URIV(I), RIV | F10.5 |

* IIMAX of these cards are read.
** A card is read for each of the 4 rivers entering the north end of the bay.

```
      COMMON/DIMEN1/U(2,12,20,11),V(2,12,20,11),W(12,20,11)
      COMMON/DIMEN2/S(2,12,20,11),P(12,20,11)
      COMMON/DIMEN3/DUDT(12,20,11),DVDT(12,20,11),DHDT(12,20),
     1              SURF(12,20)
      COMMON/DIMEN4/UC(2,12,11),WC(12,11),SC(2,12,11),PC(12,11),
     &       DUCDT(12,11)
      COMMON/FLOOR/KFLOOR(12,20),ZB(12,20),KFLORC(12),ZBC(12)
      COMMON/PULL/X(12),Y(20),Z(11),SY(20),SYY(20),YK2,ZC(11)
      COMMON/LIMITS/IMAX,IMAXM1,JMAX,JMAXM1,KMAX,KMAXM1,JWAG,KMXC,
     &       KMXCM1,KCHANL
      COMMON/UNITS/VREF,HREF,GRAV,PI,BETA,FETCH,YMAX,OMEGA,CDEPTH
      COMMON/GRID/DX,DY,DZ,DT,DT02,DXINV,DYINV,DZINV,SYJ,SY2,SYJP1,SYJM1
      COMMON/VELCTY/UNEW,UOLD,UIP1,UIM1,UJP1,UJM1,UKP1,UKM1,
     &              VNEW,VOLD,VIP1,VIM1,VJP1,VJM1,VKP1,VKM1,
     &              WNEW,WIP1,WIM1,WJP1,WJM1,WKP1,WKM1
      COMMON/STEP/N,MN,MO,NBAR,NPRINT,NMAX,KEYOUT,LTAPE
      COMMON/COORD/SOUTH(22),EAST(22),WEST(22),JEAST(22),JWEST(22)
      COMMON/MISC/TOPLYR,SBAR,UBAR,VBAR,VISC,DIFFUS,PREFIX,ACCEL
      COMMON/WEDGE/ZW(17)
      COMMON/CODES/LW,LR,LT
      DIMENSION NI(21)
      CALL PRELIM
      RHO=1.0+BETA
   10 N=N+1
      MO=3-MO
      MN=3-MN
      CALL CHANEL
C        STEP FORWARD IN TIME TO CALC. NEW VALUES OF SALINITY AND THE HORIZONTAL
C        COMPONENTS OF VELOCITY.
      CALL SETUP
C        ADJUST VERTICAL COMPONENTS OF VELOCITY AND SURFACE HEIGHT FOR
C        COMPATIBILITY WITH THE HORIZONTAL COMPONENTS OF VELOCITY.
      CALL VOLDIL
      IF(KEYOUT.NE.0)GO TO 100
C        UPDATE THE PRESSURES OVER THE FIELD.
      CALL PRESS
```

```
      CALL PRESCI
      IF(KEYOUT.NE.0)GO TO 100
C        ALL DEPENDENT VARIABLES HAVE NOW BEEN COMPUTED FOR THIS TIME STEP.
      IF(N.NE.NBAR*(N/NBAR))GO TO 20
      WRITE(LW,1000)N,SBAR,UBAR,VBAR
      NBAR=2*NBAR
   20 IF(N.NE.NPRINT*(N/NBAR))GO TO 30
C        AN INTERMEDIATE WRITE STATEMENT COULD GO HERE.
   30 IF(N.LT.NMAX)GO TO 10
C        DATA IS WRITTEN ON TAPE HERE FOR RESTART OR PLOTTING.
      CALL ECRIRE
      WRITE(LW,1003)
  100 WRITE(LW,1000)N,SBAR,UBAR,VBAR
      WRITE(LW,1001)
      DO 110 I = 1,IMAX
      JW=JWEST(I)
      JE=JEAST(I)
      DO 110 J=JW,JE
      KBOT=KFLOOR(I,J)
      WRITE(LW,1004)
      DO 110 K=KBOT,KMAX
      SNEW=S(MN,I,J,K)
      PNEW=P(I,J,K)
      UNEW=U(MN,I,J,K)*VREF
      VNEW=V(MN,I,J,K)*VREF
      WNEW=W(I,J,K)*VREF
C        VELOCITIES ARE PRESENTED IN METERS/SEC.
      WRITE(LW,1002)UNEW,VNEW,WNEW,SNEW,PNEW,I,J,K
  110 CONTINUE
      DO 106 I=1,IMAX
      DO 106 J=1,JMAX
      SURF(I,J)=(SURF(I,J)-DZ)*HREF*100.0
  106 CONTINUE
      WRITE(6,1500)
      DO 16 I=1,IMAX
   16 NI(I)=I
```

```
      WRITE(6,1501)(NI(I),I=1,IMAX)
      DO 17 JJ = 1,JMAX
      J = JMAX+1-JJ
   17 WRITE(6,1502)J,(SURF(I,J),I=1,IMAX)
      WRITE(LW,1011)
      DO 210 K=1,KMXC
      WRITE(LW,1015)
      DO 230 I=1,IMAX
      IF(ZC(K).GT.ZB(I,JWAG))GO TO 210
      U1=UC(MN,I,K)*VREF
      W1=WC(I,K)*VREF
      WRITE(LW,1012)I,K,U1,W1,SC(MN,I,K),PC(I,K)
  230 CONTINUE
  210 CONTINUE
      CALL SWEDGE
      WRITE(LW,1013)
      DO 220 I=1,IMAX
      ZB1=ZBC(I)*HREF
      XTRUE=X(I)*HREF
      ZW1=ZW(I)*HREF
      WRITE(LW,1014)I,ZB1,XTRUE,KFLORC(I),ZW1
  220 CONTINUE
 1000 FORMAT(5X,2HN=I4,5X,5HSBAR=1PE13.6,5X,5HUBAR=1PE13.6,5X,5HVBAR=1PE
     &13.6)
 1001 FORMAT(//,13X,1HU,19X,1HV,19X,1HW,15X,10HSALT CONC.,11X,8HPRESSURE
     &       ,8X,1HI,5X,1HJ,5X,1HK,/)
 1002 FORMAT(5(7X,1PE13.6),3(3X,I3))
 1003 FORMAT(//,2X,20HY'ALL DID JUST FINE ,
     & /,2X,'YOU HAD A STABLE RUN AND YOU WROTE ON THE TAPE.',
     & /,2X,33HY'ALL COME SEE US AGAIN. BYE NOW.,//)
 1004 FORMAT(/)
 1011 FORMAT(//,22X,'I',7X,'K',8X,'UC',13X,'WC',13X,'SC',13X,'PC',/)
 1012 FORMAT(15X,2(5X,I3),4(5X,E12.5))
 1013 FORMAT(//,22X,'I',9X,'ZB-M',10X,'X-M',9X,'KFLORC',5X,
     &       'Z-WEDGE-METERS',/)
 1014 FORMAT(20X,I3,9X,F11.5,9X,F9.1,5X,I3,7X,F11.5)
```

```
1015 FORMAT(/)
1500 FORMAT(///,55X,22HSURFACE HEIGHTS IN CM.,//)
1501 FORMAT(4X,2HI=,21I6,//)
1502 FORMAT(' J=',I2,3X,F5.2,20(1X,F5.2))
     STOP
     END
```

```
      SUBROUTINE PRELIM
      COMMON/DIMEN1/U(2,12,20,11),V(2,12,20,11),W(12,20,11)
      COMMON/DIMEN2/S(2,12,20,11),P(12,20,11)
      COMMON/DIMEN4/UC(2,12,11),WC(12,11),SC(2,12,11),PC(12,11),
     &       DUCDT(12,11)
      COMMON/FLOOR/KFLOOR(12,20),ZB(12,20),KFLORC(12),ZBC(12)
      COMMON/PULL/X(12),Y(20),Z(11),SY(20),SYY(20),YK2,ZC(11)
      COMMON/LIMITS/IMAX,IMAXM1,JMAX,JMAXM1,KMAX,KMAXM1,JWAG,KMXC,
     &       KMXCM1,KCHANL
      COMMON/UNITS/VREF,HREF,GRAV,PI,BETA,FETCH,YMAX,OMEGA,CDEPTH
      COMMON/GRID/DX,DY,DZ,DT,DTO2,DXINV,DYINV,DZINV,SYJ,SYJ2,SYJP1,SYJM1
      COMMON/LN/FUNLN1,FUNKAP
      COMMON/STEP/N,MN,MO,NBAR,NPRINT,NMAX,KEYOUT,LTAPE
      COMMON/FORCES/FWINDX,FWINDY,F
      COMMON/TURB/COEFVH,COEFVZ,COEFDH,COEFDZ,RICH,CRICH
      COMMON/RIVERS/YW(4),YE(4),DEPTH(4),URIV(4),RMOMAF(4),JRIV(4)
      COMMON/COORD/SOUTH(22),EAST(22),WEST(22),JEAST(22),JWEST(22)
      COMMON/MISC/TOPLYR,SBAR,UBAR,VBAR,VISC,DIFFUS,PREFIX,ACCEL
      COMMON/CODES/LW,LR,LT
      COMMON/INDEX/I,J,K
      COMMON/PASS/XBANKN,XBANKS,CUTPT,KCUT
C        IF A COLD START IS DESIRED, SET LTAPE =0
C        IF YOU WANT TO READ FROM THE TAPE, SET LTAPE=1
      READ(LR,1000) LTAPE
C        IMAX,JMAX,KMAX ARE THE MAXIMUM NUMBER OF GRID POINTS IN THE X,Y,Z
C        DIRECTIONS RESPECTIVELY.
C        JWAG IS THE NUMBER OF THE GRID POINT ALONG THE CENTERLINE OF THE SHIP
C        CHANNEL OF THE BAY.
C        KMXC IS THE MAXIMUM NUMBER OF VERTICAL GRID POINTS IN 2-D MODEL
C        OF SHIPPING CHANNEL.
      READ(LR,1000)IMAX,JMAX,KMAX,JWAG,KMXC
C        NMAX IS THE MAXIMUM NUMBER OF TIME STEPS.
C        NPRINT IS THE NUMBER OF A TIME STEP FOR WHICH AN INTERMEDIATE WRITE
C        STATEMENT CAN BE USED, VALUES ARE WRITTEN ON TAPE HERE ALSO.
C        NBAR IS THE TIME STEP NUMBER WHERE SBAR,UBAR,VBAR ARE PRINTED.
C        LET NBAR=1
```

```
      READ(LR,1000)NMAX,NPRINT,NBAR
C        FETCH IS THE LENGTH OF THE BAY FROM NORTH TO SOUTH  IN KILOMETERS
C        YMAX IS THE MAX. DISTANCE FROM THE CENTER OF THE SHIP CHANNEL(J=JWAG)
C        TO THE EAST BANK.
C        YMAX SHOULD BE SLIGHTLY GREATER THAN THE DISTANCE
C        FROM THE SHIP CHANNEL TO THE EAST BANK OF THE BAY  IN KILOMETERS
      READ(LR,1001) FETCH,YMAX
C        XBANKN AND XBANKS ARE LOCATIONS OF THE NORTH AND SOUTH BANKS OF
C        A PASS OPENING TO A LARGE BAY TO THE WEST OF THE BAY BEING
C         ANALYZED ---- IN KILOMETERS
      READ(LR,1001) XBANKN,XBANKS
C        DTREAL IS THE SIZE OF THE TIME STEP IN SECONDS.
C        ACCEL IS AN ACCELERATION FACTOR EXPEDITING CONVERGENCE OF THE SALT
C        CONSERVATION EQUATION.
C        YK1 & YK2 ARE STRETCHING CONSTANTS IN THE EAST-WEST DIRECTION.
C        SEE SUBROUTINE RUBER FOR DEFINITIONS, & ADJUST AS REQUIRED. BUT ALWAYS
C        MAKE YK2 LESS THAN PI/2.0. YK1 IS CALC. IN SUBROUTINE RUBER
      READ(LR,1001) DTREAL,YK2,ACCEL
C        HREF IS THE DEPTH IN METERS WHICH IS SLIGHTLY DEEPER THAN THE DEEPEST
C        POINT IN THE BAY EXCLUSIVE OF THE SHIP CHANNEL. THIS DEPTH CORRESPONDS
C        TO K=0
C        SIGMAT SPECIFIES DENSITY OF SEA WATER RELATIVE TO FRESH WATER
C        DENSITY OF FRESH WATER = 1.0 GM PER CUBIL CM.
C        SIGMAT = (DENSITY OF SEA WATER - 1.0) * 1000.0
C        PERIOD IS THE TIME REQUIRED FOR A COMPLETE TIDAL CYCLE IN HOURS
C        TIDE IS HEIGHT DIFFERENCE IN CM. BETWEEN HIGH AND LOW TIDE
      READ(LR,1001) HREF,SIGMAT,PERIOD,TIDE
C        ROUGH IS THE DIAMETER OF THE AVERAGE PARTICLE WHICH GENERATES THE
C        TURBULENCE ON THE BOTTOM. ROUGH IS MEASURED IN MILLIMETERS.
C        VONKAR IS THE VON KARMAN UNIVERSAL BOUNDARY LAYER CONSTANT.
      READ(LR,1001) ROUGH,VONKAR
C        VWIND IS THE WIND SPEED IN CM/SEC MEASURED 10 METERS ABOVE THE SURFACE
C        DIRECT IS THE DIRECTION WHICH THE WIND IS BLOWING MEASURED IN DEGREES
C        AND COUNTERCLOCKWISE FROM THE POSITIVE XAXIS
C        PHI IS THE AVERAGE LATITUDE OF THE BAY.
      READ(LR,1001) VWIND,DIRECT,PHI
```

```
C        CDEPTH IS THE DEPTH OF THE SHIPPING CHANNEL BELOW MEAN SEA
C        LEVEL IN METERS.
C        CUTPT IS HEIGHT OF THE SALT WEDGE IN BAY ABOVE K=0 IN METERS.
      READ(LR,1001)CDEPTH,CUTPT
      WRITE(LW,1002)LTAPE,DTREAL,ACCEL
      WRITE(LW,1003)SIGMAT,ROUGH,VONKAR
      WRITE(LW,1004)IMAX,JMAX,KMAX,JWAG,KMXC,NMAX,NPRINT,NBAR
      WRITE(LW,1006)HREF,PERIOD,TIDE
      WRITE(LW,1007)FETCH,YMAX,YK2
      WRITE(LW,1008)VWIND,DIRECT,PHI
      WRITE(LW,1018)CDEPTH,CUTPT
      WRITE(LW,1017) XBANKN,XBANKS
C        READ IIMAX DATA CARDS TO SPECIFY.........
C          1. THE DISTANCE SOUTH OF THE NORTHERN BOUNDARY OF THE BAY.(BEGIN
C          WITH 0.0 & END WITH FETCH)
C          2. THE DISTANCE FROM THE SHIP CHANNEL TO THE EASTERN BOUNDARY OF
C          THE BAY.
C          3. THE DISTANCE FROM THE SHIP CHANNEL TO THE WESTERN BOUNDARY OF THE
C          BAY(NOTE...ALWAYS NEGATIVE)
C        READ 1 DATA CARD TO SPECIFY THE CONVERSION FACTOR OF THE UNITS ABOVE
C        TO METERS.
      READ(LR,1000) IIMAX
      XBANKN=(1000.0*XBANKN)/HREF
      XBANKS=(1000.0*XBANKS)/HREF
      DO 2 II=1,IIMAX
    2 READ(LR,1001) SOUTH(II),EAST(II),WEST(II)
      READ(LR,1001) CONVRT
      IMAXM1=IMAX-1
      JMAXM1=JMAX-1
      KMAXM1=KMAX-1
      KMXCM1=KMXC-1
      CALL RUBER
      CUTPT=CUTPT/HREF
      K=0
    1 K=K+1
      IF(Z(K).LE.CUTPT) GO TO 1
```

```
      KCUT=K-1
      VREF=SQRT(GRAV*HREF)
      WRITE(LW,1010)
      DO 4 II=1,IIMAX
      SOUTH(II)=SOUTH(II)*CONVRT/HREF
      EAST(II)=EAST(II)*CONVRT/HREF
      WEST(II)=WEST(II)*CONVRT/HREF
      SOUTH1=SOUTH(II)*HREF
      EAST1=EAST(II)*HREF
      WEST1=WEST(II)*HREF
    4 WRITE(LW,1011) II,SOUTH1,EAST1,WEST1
      WRITE(LW,1012)
      I=1
      J=JWAG
    6 J=J+1
      IF(Y(J).LT.EAST(1)) GO TO 6
      JEAST(I)=J-1
      J=JWAG
    8 J=J-1
      IF(Y(J).GT.WEST(1)) GO TO 8
      JWEST(I)=J+1
      YW1=WEST(1)*HREF
      YE1=EAST(1)*HREF
      YEAST=EAST(1)
      YWEST=WEST(1)
      CALL BOTTOM (YEAST,YWEST)
      WRITE(LW,1013) I,YW1,YE1,JWEST(1),JEAST(1)
      DO 16 I=2,IMAX
      II=1
   10 II=II+1
      IF(SOUTH(II).LT.X(I)) GO TO 10
      WATE=(X(I)-SOUTH(II-1))/(SOUTH(II)-SOUTH(II-1))
      YEAST=EAST(II)*WATE+EAST(II-1)*(1.0-WATE)
      YWEST=WEST(II)*WATE+WEST(II-1)*(1.0-WATE)
      J=JWAG
   12 J=J+1
```

```
      IF(Y(J).LT.YEAST) GO TO 12
      JEAST(I)=J-1
      J=JWAG
   14 J=J-1
      IF(Y(J).GT.YWEST) GO TO 14
      JWEST(I)=J+1
      YE1=YEAST*HREF
      YW1=YWEST*HREF
      CALL BOTTOM (YEAST,YWEST)
      WRITE(LW,1013) I,YW1,YE1,JWEST(I),JEAST(I)
   16 CONTINUE
      GO TO 9000
      WRITE(LW,1015)
      DO 20 I=1,IMAX
      J1=JWEST(I)
      J2=JEAST(I)
      DO 20 J=J1,J2
      XREAL=X(I)*HREF
      YREAL=Y(J)*HREF
      ZBREAL=ZB(I,J)*HREF
   20 WRITE(LW,1016)I,J,KFLOOR(I,J),ZB(I,J),XREAL,YREAL,ZBREAL
 9000 CONTINUE
C        WANT TO READ THE COORDINATES OF THE WEST & EAST BANKS AND THE DEPTHS
C        OF THE RIVERS FLOWING INTO THE NORTH END OF THE BAY. USE THE SAME UNITS
C        FOR THE EAST AND WEST COORD. AS FOR THE SHORE WITH THE SHIP CHANNEL
C        AS Y=0.0, AND INPUT THE DEPTH IN METERS.
C        AS Y=0.0,AND INPUT THE DEPTH IN METERS.URIV(I) ARE THE MAXI-
C        MUM RIVER VELOCITIES IN METERS/SEC.
      COEFDH=0.0025
      COEFDZ=0.0025
      COEFVH=0.005
      COEFVZ=0.005
      DO 30 I=1,4
      READ(LR,1001) YW(I),YE(I),DEPTH(I),URIV(I),RIV
   30 JRIV(I)=IFIX(RIV+0.5)
      DO 22 I=1,4
```



ORIGINAL PAGE IS
OF POOR QUALITY

```
      YW(I)=YW(I)*CONVRT
      YE(I)=YE(I)*CONVRT
      WRITE(LW,1014)I,YW(I),I,YE(I),I,DEPTH(I)
   22 WRITE(LW,1005)I,URIV(I)
      DO 24 I=1,4
      YW(I)=YW(I)/HREF
      YE(I)=YE(I)/HREF
      DEPTH(I)=DEPTH(I)/HREF
   24 URIV(I)=URIV(I)/VREF
      BETA=0.001*SIGMAT
      CRICH=-2.0*BETA*DZ
      F=2.0*OMEGA*SIN(PHI)*HREF/VREF
      ROUGH=0.001*ROUGH/HREF
      FUNLNI=ALOG(0.2/ROUGH)
      FUNKAP=VONKAR*8.5
      FWIND=1.64E-4*(VWIND**1.333)
      FWINDX=FWIND*COS(DIRECT*PI/180.0)
      FWINDY=FWIND*SIN(DIRECT*PI/180.0)
      WRITE(LW,1009) FWIND,FWINDX,FWINDY
      FWINDY=FWINDY/((100.0*VREF)**2)
      FWINDX=FWINDX/((100.0*VREF)**2)
      DT=DTREAL*VREF/HREF
      DTO2=0.5*DT
      IF(LTAPE.EQ.0) CALL INITAL
      IF(LTAPE.EQ.1) CALL LIRE
      CALL PRESS
      CALL PRESSC
      DO 50 I=1,IMAX
      JWM1=JWEST(I)-1
      JEP1=JEAST(I)+1
      DO 44 J=1,JWM1
      KBOT=KFLOOR(I,JWM1)
      DO 44 K=KBOT,KMAX
   44 P(I,J,K)=P(I,JWM1+1,K)
      DO 46 J=JEP1,JMAX
      KBOT=KFLOOR(I,JEP1)
```

```
      DO 46 K=KBOT,KMAX
   46 P(I,J,K)=P(I,JEP1-1,K)
   50 CONTINUE
      DO 60 I=1,IMAX
      DO 61 K=1,KMXC
      UC(MO,I,K) = UC(MN,I,K)
      SC(MO,I,K) = SC(MN,I,K)
   61 CONTINUE
      DO 60 J=1,JMAX
      DO 60 K=1,KMAX
      U(MO,I,J,K)=U(MN,I,J,K)
      V(MO,I,J,K)=V(MN,I,J,K)
      S(MO,I,J,K)=S(MN,I,J,K)
   60 CONTINUE
 1000 FORMAT(8I10)
 1001 FORMAT(9F10.5)
 1002 FORMAT(/,5X,'LTAPE =',I3,/,5X,'DTREAL =',F9.4,' SECONDS',
     &      5X,'ACCEL=',F9.4)
 1003 FORMAT(/,5X,'SIGMAT =',F9.4,/,5X,'ROUGH =',F9.4,' MILLIMETERS',/,
     & 5X,'VON KARMAN CONSTANT =',F9.4)
 1004 FORMAT(/,5X,'IMAX =',I3,/,5X,'JMAX =',I3,/,5X,'KMAX =',I3,/,5X,
     & 'JWAG =',I3,/,5X,'KMXC =',I3,/,5X,'NMAX =',I4,/,5X,'NPRINT =',I4,
     & /,5X,'NBAR =',I3)
 1005 FORMAT(/,5X,'MAXIMUM VELOCITY OF RIVER ',I1,' = ',F9.4,' METERS/SE
     1C')
 1006 FORMAT(/,5X,'HREF =',F9.4,' METERS',/,5X,'PERIOD =',F9.4,' HOURS',
     & /,5X,'TIDE =',F9.4,' CM.')
 1007 FORMAT(/,5X,'LENGTH OF BAY FROM NORTH TO SOUTH =',F9.4,' KM.',/,
     & 5X,'MAX. DIST. FROM CENTER OF SHIP CHANNEL TO EAST BANK =',F9.4,
     & ' KM.',/,5X,'YK2 =',F9.4)
 1008 FORMAT(/,5X,'WIND SPEED 10 METERS ABOVE SURFACE =',F9.4,' CM/SEC',
     & /5X,'DIRECTION WHICH WIND IS BLOWING =',F9.4,' DEG. ANTICLOCKWISE
     & FROM POSITIVE X-AXIS',/,5X,'AVERAGE LATITUDE OF BAY =',F9.4,
     & ' DEGREES')
 1009 FORMAT(/,5X,6HFWIND=1PE10.3,2X,'DYNE/CMSQ',
     1          5X,7HFWINDX=1PE10.3,2X,'DYNE/CMSQ',
```

```
     2          5X,7HFWINDY=1PE10.3,2X,'DYNE/CMSQ',//)
1010 FORMAT(/,15X,'II',15X,'SOUTH1(M)',15X,'EAST1(M)',15X,'WEST1(M)',/)
1011 FORMAT(13X,I4,13X,1PE13.6,2(10X,1PE13.6))
1012 FORMAT(/15X,'I',15X,'YWEST',15X,'YEAST',15X,'JWEST',15X,'JEAST',/)
1013 FORMAT(13X,I3,10X,1PE13.6,7X,1PE13.6,12X,I4,16X,I4)
1014 FORMAT(/,5X,'YW',I1,' = ',F9.3,5X,'YE',I1,' = ',F9.3,5X,'DEPTH',I1
     1          ,' = ',F9.3)
1015 FORMAT(/10X,'I',10X,'J',10X,'KFLOOR(I,J)',6X,'ZB(I,J)',15X,
     1          'XREAL',14X,'YREAL',15X,'ZBREAL',/)
1016 FORMAT(8X,I3,8X,I3,10X,I4,10X,1PE13.6,10X,1PE13.6,7X,1PE13.6,
     1          5X,1PE13.6)
1017 FORMAT(/,5X,'XBANKN=',F9.4,5X,'XBANKS=',F9.4)
1018 FORMAT(/,5X,'CDEPTH =',F9.4,/,5X,'CUTPT =',F9.4)
     RETURN
     END
```

```
      SUBROUTINE LIRE
C
C        A SUBROUTINE TO READ DATA FROM TAPE WHEN LTAPE IS NOT = 0
C
      COMMON/DIMEN1/U(2,12,20,11),V(2,12,20,11),W(12,20,11)
      COMMON/DIMEN2/S(2,12,20,11),P(12,20,11)
      COMMON/DIMEN3/DUDT(12,20,11),DVDT(12,20,11),DHDT(12,20),
     1              SURF(12,20)
      COMMON/DIMEN4/UC(2,12,11),WC(12,11),SC(2,12,11),PC(12,11),
     &       DUCDT(12,11)
      COMMON/FLOOR/KFLOOR(12,20),ZB(12,20),KFLORC(12),ZBC(12)
      COMMON/LIMITS/IMAX,IMAXM1,JMAX,JMAXM1,KMAX,KMAXM1,JWAG,KMXC,
     &       KMXCM1,KCHANL
      COMMON/COORD/SOUTH(22),EAST(22),WEST(22),JEAST(22),JWEST(22)
      COMMON/PULL/X(12),Y(20),Z(11),SY(20),SYY(20),YK2,ZC(11)
      COMMON/STEP/N,MN,MO,NBAR,NPRINT,NMAX,KEYOUT,LTAPE
      COMMON/RIVERS/YW(4),YE(4),DEPTH(4),URIV(4),RMOMAF(4),JRIV(4)
      COMMON/SHORE/ZBANKN(31),ZBANKS(31),ZBANKE(31),ZBANKW(31)
      REWIND 3
      READ(3)  IMAX,JMAX,KMAX,KMXC,JWAG,
     &         (URIV(I),RMOMAF(I),JRIV(I),I=1,4)
      READ(3) (ZBANKN(J),J=1,JMAX)
      READ(3) (ZBANKS(J),J=1,JMAX)
      READ(3) (ZBANKE(J),J=1,JMAX)
      READ(3) (ZBANKW(J),J=1,JMAX)
      READ(3) ((DHDT(I,J),I=1,IMAX),J=1,JMAX)
      READ(3) (((DUDT(I,J,K),I=1,IMAX),J=1,JMAX),K=1,KMAX)
      READ(3) (((DVDT(I,J,K),I=1,IMAX),J=1,JMAX),K=1,KMAX)
      READ(3) (((U(MN,I,J,K), I=1,IMAX), J=1,JMAX), K=1,KMAX)
      READ(3) (((V(MN,I,J,K), I=1,IMAX), J=1,JMAX), K=1,KMAX)
      READ(3) (((W(I,J,K),I=1,IMAX),J=1,JMAX),K=1,KMAX)
      READ(3) (((S(MN,I,J,K), I=1,IMAX), J=1,JMAX), K=1,KMAX)
      READ(3) (((P(I,J,K),I=1,IMAX),J=1,JMAX),K=1,KMAX)
      READ(3) ((SURF(I,J), I=1,IMAX), J=1,JMAX)
      READ(3) ((ZB(I,J), I=1,IMAX), J=1,JMAX)
      READ(3) ((KFLOOR(I,J), I=1,IMAX), J=1,JMAX)
```

```
      READ(3)  ((UC(MN,I,K), I=1,IMAX),K=1,KMXC)
      READ(3)  ((WC(I,K), I=1,IMAX),K=1,KMXC)
      READ(3)  ((SC(MN,I,K), I=1,IMAX),K=1,KMXC)
      READ(3)  ((PC(I,K), I=1,IMAX),K=1,KMXC)
      READ(3)  ((DUCDT(I,K), I=1,IMAX),K=1,KMXC)
      READ(3)  (ZBC(I), I=1,IMAX)
      READ(3)  (KFLORC(I), I=1,IMAX)
      RETURN
      ENTRY ECRIRE
C
C        AN ENTRY TO WRITE UPDATED DATA ON THE TAPE
C
      REWIND 3
      WRITE(3)  IMAX,JMAX,KMAX,KMXC,JWAG,
     &         (URIV(I),RMOMAF(I),JRIV(I),I=1,4)
      WRITE(3) (ZBANKN(J),J=1,JMAX)
      WRITE(3) (ZBANKS(J),J=1,JMAX)
      WRITE(3) (ZBANKE(J),J=1,JMAX)
      WRITE(3) (ZBANKW(J),J=1,JMAX)
      WRITE(3) ((DHDT(I,J),I=1,IMAX),J=1,JMAX)
      WRITE(3) (((DUDT(I,J,K),I=1,IMAX),J=1,JMAX),K=1,KMAX)
      WRITE(3) (((DVDT(I,J,K),I=1,IMAX),J=1,JMAX),K=1,KMAX)
      WRITE(3) (((U(MN,I,J,K), I=1,IMAX), J=1,JMAX), K=1,KMAX)
      WRITE(3) (((V(MN,I,J,K), I=1,IMAX), J=1,JMAX), K=1,KMAX)
      WRITE(3) (((W(I,J,K),I=1,IMAX),J=1,JMAX),K=1,KMAX)
      WRITE(3) (((S(MN,I,J,K), I=1,IMAX), J=1,JMAX), K=1,KMAX)
      WRITE(3) (((P(I,J,K),I=1,IMAX),J=1,JMAX),K=1,KMAX)
      WRITE(3) ((SURF(I,J), I=1,IMAX), J=1,JMAX)
      WRITE(3) ((ZB(I,J), I=1,IMAX), J=1,JMAX)
      WRITE(3) ((KFLOOR(I,J), I=1,IMAX), J=1,JMAX)
      WRITE(3)  ((UC(MN,I,K), I=1,IMAX),K=1,KMXC)
      WRITE(3)  ((WC(I,K), I=1,IMAX),K=1,KMXC)
      WRITE(3)  ((SC(MN,I,K), I=1,IMAX),K=1,KMXC)
      WRITE(3)  ((PC(I,K), I=1,IMAX),K=1,KMXC)
      WRITE(3)  ((DUCDT(I,K), I=1,IMAX),K=1,KMXC)
      WRITE(3)  (ZBC(I), I=1,IMAX)
```

```
      WRITE(3)  (KFLORC(I), I=1,IMAX)
      WRITE(3) (X(I),I=1,IMAX)
      WRITE(3) (Y(J),J=1,JMAX)
      WRITE(3) (Z(K),K=1,KMAX)
      WRITE(3) (ZC(K),K=1,KMXC)
      WRITE(3) (SOUTH(I),I=1,22)
      WRITE(3) (WEST(I),I=1,22)
      WRITE(3) (EAST(I),I=1,22)
      WRITE(3) (JWEST(I),I=1,IMAX)
      WRITE(3) (JEAST(I),I=1,IMAX)
      RETURN
      END
```

```
      SUBROUTINE RUBER
      COMMON/PULL/X(12),Y(20),Z(11),SY(20),SYY(20),YK2,ZC(11)
      COMMON/LIMITS/IMAX,IMAXM1,JMAX,JMAXM1,KMAX,KMAXM1,JWAG,KMXC,
     &      KMXCM1,KCHANL
      COMMON/GRID/DX,DY,DZ,DT,DTO2,DXINV,DYINV,DZINV,SYJ,SY2,SYJP1,SYJM1
      COMMON/UNITS/VREF,HREF,GRAV,PI,BETA,FETCH,YMAX,OMEGA,CDEPTH
      COMMON/CODES/LW,LR,LT
      WRITE(LW,1000)
      DX=(FETCH*1000.0/HREF)/FLOAT(IMAXM1)
      DXINV=1.0/DX
      X1=-DX
      DO 10 I=1,IMAX
      X1=X1+DX
      X(I)=X1
      XREAL=X1*HREF
   10 WRITE(LW,1001)I,X1,XREAL
      WRITE(LW,1002)
      DY=1.0/FLOAT(JMAX-JWAG)
      DYINV=1.0/DY
      YK1=(YMAX*1000.0)/(HREF*TAN(YK2))
      CONSQ=YK1*YK1
      DO 20 J=1,JMAX
      CAPY=DY*FLOAT(J-JWAG)
      Y1=YK1*TAN(YK2*CAPY)
      Y(J)=Y1
      YREAL=Y1*HREF
      YSQ=Y1*Y1
      DENOM=CONSQ+YSQ
      SY(J)=(1.0/YK2)*YK1/DENOM
      SYY(J)=-(2.0/YK2)*YK1*Y1/(DENOM*DENOM)
   20 WRITE(LW,1003)J,Y1,SY(J),SYY(J),YREAL
      WRITE(LW,1004)
      DZ=1.0/FLOAT(KMAX)
      DZINV=1.0/DZ
      Z1=0.0
      DO 30 K=1,KMAX
```

```
      Z1=Z1+DZ
      Z(K)=Z1
      ZREAL=Z1*HREF
   30 WRITE(LW,1005)K,Z(K),ZREAL
      WRITE(LW,1006)
      Z1=1.-CDEPTH/HREF
      KCHANL=-IFIX(Z1*DZINV)+2
      Z1=-FLOAT(KCHANL)*DZ
      DO 40 K=1,KMXC
      Z1=Z1+DZ
      ZC(K)=Z1
      ZREAL=Z1*HREF
   40 WRITE(LW,1005) K,ZC(K),ZREAL
 1000 FORMAT(//,13X,1HI,18X,1HX,15X,8HXREAL(M),/)
 1001 FORMAT(10X,I4,11X,1PE13.6,5X,1PE13.6)
 1002 FORMAT(//,13X,1HJ,18X,1HY,15X,2HSY,15X,3HSYY,15X,8HYREAL(M),/)
 1003 FORMAT(10X,I4,11X,1PE13.6,3(5X,1PE13.6))
 1004 FORMAT(//,13X,1HK,18X,1HZ,15X,8HZREAL(M),/)
 1005 FORMAT(10X,I4,11X,1PE13.6,5X,1PE13.6)
 1006 FORMAT(//,13X,'K',17X,'ZC',15X,'REAL(M)',/)
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE INITAL
      COMMON/DIMEN1/U(2,12,20,11),V(2,12,20,11),W(12,20,11)
      COMMON/DIMEN2/S(2,12,20,11),P(12,20,11)
      COMMON/DIMEN3/DUDT(12,20,11),DVDT(12,20,11),DHDT(12,20),
     1               SURF(12,20)
      COMMON/DIMEN4/UC(2,12,11),WC(12,11),SC(2,12,11),PC(12,11),
     &       DUCDT(12,11)
      COMMON/FLOOR/KFLOOR(12,20),ZB(12,20),KFLORC(12),ZBC(12)
      COMMON/PULL/X(12),Y(20),Z(11),SY(20),SYY(20),YK2,ZC(11)
      COMMON/LIMITS/IMAX,IMAXM1,JMAX,JMAXM1,KMAX,KMAXM1,JWAG,KMXC,
     &       KMXCM1,KCHANL
      COMMON/STEP/N,MN,MO,NBAR,NPRINT,NMAX,KEYOUT,LTAPE
      COMMON/GRID/DX,DY,DZ,DT,DTO2,DXINV,DYINV,DZINV,SYJ,SY2,SYJP1,SYJM1
      COMMON/COORD/SOUTH(22),EAST(22),WEST(22),JEAST(22),JWEST(22)
      COMMON/RIVERS/YW(4),YE(4),DEPTH(4),URIV(4),RMOMAF(4),JRIV(4)
      COMMON/SHORE/ZBANKN(31),ZBANKS(31),ZBANKE(31),ZBANKW(31)
      COMMON/PASS/XBANKN,XBANKS,CUTPT,KCUT
C        WILL PUT A FENCE AROUND THE BAY & WILL LATER CUT HOLES WHERE
C        APPROPRIATE
      DO 2 I=1,IMAX
      ZBANKW(I)=1.01
    2 ZBANKE(I)=1.01
      DO 4 J=1,JMAX
      ZBANKN(J)=1.01
    4 ZBANKS(J)=1.01
C        WANT TO SPECIFY INITIAL CONDITIONS IN EACH OF THE RIVER MOUTHS.
      I=1
      DO 14 NN=1,4
      YWW= YW(NN)
      YEE = YE(NN)
      DEPTHH = DEPTH(NN)
      URIVV = URIV(NN)
      WIDTH1=YEE-YWW
      AREA1=WIDTH1*DEPTHH
      J=JRIV(NN)
      KBOT=KFLOOR(I,J)
```

```
      ZBOT=ZB(I,J)
      ZBANKN(J)=ZBOT
      JM1=J-1
      JP1=J+1
      WIDTH2=0.5*(Y(JP1)-Y(JM1))
      IF(NN.EQ.1)GO TO 6
      AREA2=WIDTH2*(1.0+0.01*DZ-ZBOT)
      URIVV=URIVV*AREA1/AREA2
      URIV(NN)=URIVV
      RMOMAF(NN)=AREA2**2/AREA1
      GO TO 8
    6 AREA2=WIDTH2*DEPTHH
      URIVV=URIVV*AREA1/AREA2
      URIV(NN)=URIVV
      RMOMAF(NN)=AREA2**2/AREA1
      ZBOT=ZBC(I)
    8 DO 14 K=KBOT,KMAX
      VERT=(Z(K)-ZBOT)/(1.0-ZBOT)
      PROFIL=1.0-(1.0-VERT)**7
      U(MN,I,J,K)=URIVV*PROFIL
      V(MN,I,J,K)=0.0
      W(I,J,K)=0.0
      S(MN,I,J,K)=0.0
   14 CONTINUE
      DO 2C J=1,JMAX
      SURF(I,J)=1.01*DZ
      K=0
   18 K=K+1
      IF(K.GT.KMAX) GO TO 20
      IF(Z(K).GE.ZBANKN(J)) GO TO 20
      U(MN,I,J,K)=0.0
      V(MN,I,J,K)=0.0
      W(I,J,K)=0.0
      S(MN,I,J,K)=0.0
      GO TO 18
   20 CONTINUE
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C        WILL NOW SPECIFY INITIAL CONDITIONS IN SHIPPING CHANNEL
      URIVV = URIV(I)
      ZBOT=ZBC(I)
      KBOT=KFLORC(I)
      DO 10 K=KBOT,KMXC
      VERT=(ZC(K)-ZBOT)/(1.0-ZBOT)
      PROFIL=1.0-(1.0-VERT)**7
      UCHANL=URIVV*PROFIL
      UC(MN,1,K)=UCHANL
      WC(1,K)=0.0
      SC(MN,1,K)=0.0
      DUCDT(1,K)=0.0
      DO 10 I=2,IMAX
      UC(MN,I,K)=UCHANL
      WC(I,K)=0.0
      SC(MN,I,K)=X(I)/X(IMAX)
      DUCDT(I,K)=0.0
   10 CONTINUE
      DO 11 I=1,IMAX
      KBOTM1 = KFLORC(I) - 1
      DO 11 K=1,KBOTM1
      UC(MN,I,K) = 0.
      WC(I,K) = 0.
      SC(MN,I,K) = 0.
      PC(I,K) = 0.
   11 DUCDT(I,K) = 0.
C        WILL NOW SPECIFY INITIAL CONDITIONS AT THE MAIN PASS INTO THE
C        SEA
      I=IMAX
      JW=JWEST(I)
      JE=JEAST(I)
      DO 24 J=JW,JE
      ZBANKS(J)=ZB(I,J)
      IF(J.EQ.JWAG) ZBANKS(J)=1.0E-6
      K=0
   22 K=K+1
```

```
      IF(K.GT.KMAX) GO TO 24
      IF(Z(K).LT.ZBANKS(J)) GO TO 22
      U(MN,I,J,K)=0.0
      IF(K.GT.KCUT) U(MN,I,J,K)=URIV(1)*0.10
      V(MN,I,J,K)=0.0
      W(I,J,K)=0.0
      S(MN,I,J,K)=1.0
      GO TO 22
   24 CONTINUE
      DO 28 J=1,JMAX
      SURF(I,J)=DZ
      K=0
   26 K=K+1
      IF(K.GT.KMAX) GO TO 28
      IF(Z(K).GE.ZBANKS(J)) GO TO 28
      U(MN,I,J,K)=0.0
      V(MN,I,J,K)=0.0
      W(I,J,K)=0.0
      IF(K.GT.KCUT) GO TO 29
      S(MN,I,J,K)=1.0
      GO TO 30
   29 CONTINUE
      S(MN,I,J,K)=0.0
   30 GO TO 26
   28 CONTINUE
C         WILL NOW SPECIFY INITIAL CONDITIONS IN THE PASS TO THE BAY ON
C         THE WEST
      DO 34 I=2,IMAX
      IF(X(I).LT.XBANKN.OR.X(I).GT.XBANKS) GO TO 34
      JW=JWEST(I)
      ZBANKW(I)=ZB(I,JW)
   34 CONTINUE
C         WILL NOW SPECIFY INITIAL CONDITIONS IN THE BAY PROPER
      DO 39 J=1,JMAX
   39 DHDT(1,J)=0.0
      DO 40 I=2,IMAX
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SALT = (X(I)/X(IMAX))**2
      DO 40 J=1,JMAX
      DHDT(I,J)=0.0
      SURF(I,J)=DZ
      DO 40 K=1,KMAX
      DUDT(I,J,K)=0.0
      DVDT(I,J,K)=0.0
      IF(I.EQ.IMAX) GO TO 40
      S(MN,I,J,K) = SALT
   42 U(MN,I,J,K) = 0.001
      IF(K.GT.KCUT) U(MN,I,J,K)=0.2*URIV(I)
      V(MN,I,J,K)=0.0
      W(I,J,K)=0.0
   40 CONTINUE
      GO TO 9000
      DO 70 J=1,JMAX
   70 WRITE(6,1000)J,ZBANKN(J),ZBANKS(J)
      WRITE(6,1001)
      DO 80 I=1,IMAX
   80 WRITE(6,1000)I,ZBANKW(I),ZBANKE(I)
 1000 FORMAT(5X,I3,2(2X,1PE10.3))
 1001 FORMAT(/)
 9000 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BOTTOM (YEAST,YWEST)
C
C        THE BOTTOM LOCATION IS COMPUTED
C
      COMMON/INDEX/I,J,K
      COMMON/COORD/SOUTH(22),EAST(22),WEST(22),JEAST(22),JWEST(22)
      COMMON/FLOOR/KFLOOR(12,20),ZB(12,20),KFLORC(12),ZBC(12)
      COMMON/PULL/X(12),Y(20),Z(11),SY(20),SYY(20),YK2,ZC(11)
      COMMON/LIMITS/IMAX,IMAXM1,JMAX,JMAXM1,KMAX,KMAXM1,JWAG,KMXC,
     &        KMXCM1,KCHANL
      COMMON/UNITS/VREF,HREF,GRAV,PI,BETA,FETCH,YMAX,OMEGA,CDEPTH
      DELTAY=1.0/HREF
      DELTAX=2.0/HREF
      XPRIME=X(I)/X(IMAX)
      XFUN=DELTAX*(1.0-XPRIME**3)
      DO 10 J=1,JWAG
      YPRIME=Y(J)/YWEST
      YFUN=0.5*DELTAY*(1.0+COS(2.0*PI*YPRIME))
      ZB(I,J)=XFUN+YFUN+1.0E-6
   10 CONTINUE
      DO 20 J=JWAG,JMAX
      YPRIME=Y(J)/YEAST
      YFUN=0.5*DELTAY*(1.0+COS(2.0*PI*YPRIME))
      ZB(I,J)=XFUN+YFUN+1.0E-6
   20 CONTINUE
      DO 40 J=1,JMAX
      K=0
   30 K=K+1
      IF(Z(K).LT.ZB(I,J)) GO TO 30
      KFLOOR(I,J)=K
      KFLORC(I)=2
      ZBC(I)=1-CDEPTH/HREF
   40 CONTINUE
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE SETUP
C
C        VALUES ARE CALLED FROM DIMENSIONAL ARRAYS AND ASSIGNED TO
C        NONDIMENSIONED
C        VARIABLES WHICH WILL BE USED IN CALC. NEW VALUES OF U,V,& S
      COMMON/DIMEN1/U(2,12,20,11),V(2,12,20,11),W(12,20,11)
      COMMON/DIMEN2/S(2,12,20,11),P(12,20,11)
      COMMON/DIMEN3/DUDT(12,20,11),DVDT(12,20,11),DHDT(12,20),
     1              SURF(12,20)
      COMMON/DIMEN4/UC(2,12,11),WC(12,11),SC(2,12,11),PC(12,11),
     &     DUCDT(12,11)
      COMMON/FLOOR/KFLOOR(12,20),ZB(12,20),KFLORC(12),ZBC(12)
      COMMON/COORD/SOUTH(22),EAST(22),WEST(22),JEAST(22),JWEST(22)
      COMMON/PULL/X(12),Y(20),Z(11),SY(20),SYY(20),YK2,ZC(11)
      COMMON/LIMITS/IMAX,IMAXM1,JMAX,JMAXM1,KMAX,KMAXM1,JWAG,KMXC,
     &     KMXCM1,KCHANL
      COMMON/PASS/XBANKN,XBANKS,CUTPT,KCUT
      COMMON/GRID/DX,DY,DZ,DT,DT02,DXINV,DYINV,DZINV,SYJ,SY2,SYJP1,SYJM1
      COMMON/STEP/N,MN,MC,NBAR,NPRINT,NMAX,KEYOUT,LTAPE
      COMMON/UNITS/VREF,HREF,GRAV,PI,BETA,FETCH,YMAX,OMEGA,CDEPTH
      COMMON/CONC/SNEW,SOLD,SIP1,SIM1,SJP1,SJM1,SKP1,SKM1,RHO,RHOINV
      COMMON/VELCTY/UNEW,UOLD,UIP1,UIM1,UJP1,UJM1,UKP1,UKM1,
     &              VNEW,VOLD,VIP1,VIM1,VJP1,VJM1,VKP1,VKM1,
     &              WNEW,WIP1,WIM1,WJP1,WJM1,WKP1,WKM1
      COMMON/PRES/PK,PIP1,PIM1,PJP1,PJM1
      COMMON/TURB/COEFVH,COEFVZ,COEFDH,COEFDZ,RICH,CRICH
      COMMON/SHORE/ZBANKN(31),ZBANKS(31),ZBANKE(31),ZBANKW(31)
      COMMON/LN/FUNLN1,FUNKAP
      COMMON/INDEX/I,J,K
      COMMON/MISC/TOPLYR,SBAR,UBAR,VBAR,VISC,DIFFUS,PREFIX,ACCEL
      UBAR=0.0
      VBAR=0.0
      SBAR=0.0
      DO 30 I=2,IMAX
      X1=X(I)
      JW=JWEST(I)
```

```
      JE=JEAST(I)
      IP1=I+1
      IM1=I-1
      JWM1=JW-1
      SYJ=SY(JWM1)
      SYJP1=SY(JW)
      DO 20 J=JW,JE
      JP1=J+1
      JM1=J-1
      SY1=SY(J)
      SY2=SYY(J)
      KBOT=KFLOOR(I,J)
      KBOTP1=KBOT+1
      K = KBOT
      ZB1=ZB(I,J)
      TOPLYR=SURF(I,J)-0.5*DZ
      SYJM1=SYJ
      SYJ=SYJP1
      SYJP1=SY(JP1)
      PREFIX=SYJ*DYINV
C        THIS IS A SERIES APPROXIMATION OF A LN FUNCTION
      ALPHA1=5.0*(Z(KBOTP1)-ZB1)-1.0
      ALPHA2=ALPHA1*ALPHA1
      ALPHA3=ALPHA2*ALPHA1
      ALPHA4=ALPHA3*ALPHA1
      ALPHA5=ALPHA4*ALPHA1
      FUNLN2=ALPHA1-0.5*ALPHA2+0.3333*ALPHA3-0.25*ALPHA4+0.2*ALPHA5
      FUNLN=(1.0-DZ/((Z(KBOT)-ZB1+0.5*DZ)*(FUNLN1+FUNLN2+FUNKAP)))
      SOLD=S(MO,I,J,KBOT)
      UOLD=U(MO,I,J,KBOT)
      VOLD=V(MO,I,J,KBOT)
      WNEW=W(I,J,KBOT)
      SKM1=SOLD
      SKP1=S(MO,I,J,KBOTP1)
      UKP1=U(MO,I,J,KBOTP1)
      VKP1=V(MO,I,J,KBOTP1)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      WKP1=W(I,J,KBOTP1)
      SJM1=S(MO,I,JM1,KBOT)
      SJP1=S(MO,I,JP1,KBOT)
      VJM1 = V(MO,I,JM1,KBCT)
      VJP1 = V(MO,I,JP1,KBOT)
      SIM1=S(MO,IM1,J,KBCT)
      UIM1=U(MO,IM1,J,KBCT)
      IF(I.EQ.IMAX)GO TO 4
      SIP1=S(MO,IP1,J,KBCT)
      UIP1=U(MO,IP1,J,KBOT)
      GO TO 5
    4 SIP1=1.0
      UIP1=UOLD
    5 CONTINUE
      IF(J.NE.JWAG)GO TO 6
      KCMAX=KBOT+KCHANL-1
      SKM1=SC(MO,I,KCMAX)
      UKM1=UC(MO,I,KCMAX)
      VKM1=0.0
      WKM1 = WC(I,KCMAX)*0.1906
      KIM1=KFLOOR(IM1,J)
      IF(KIM1.LE.KBOT) GO TO 3
      KCIM1=KIM1+KCHANL-1
      UIM1=UC(MO,IM1,KCIM1)
      SIM1=SC(MO,IM1,KCIM1)
      PIM1=PC(IM1,KCIM1)
      GO TO 2
    3 PIM1=P(IM1,J,KBOT)
    2 PIP1=P(IP1,J,KBOT)
      RICH=0.0
      RHOINV = 1.0-BETA*SOLD
      CALL UVSNEW
      U(MN,I,J,K)=UNEW
      UBAR=UBAR+ABS(UNEW-UOLD)
      S(MN,I,J,K)=SNEW
      SBAR=SBAR+ABS(SNEW-SOLD)
```

```
      GO TO 9
    6 CONTINUE
C       A NEW VALUE OF S AT THE BOTTOM GRID POINT IS NOW COMPUTED
      CALL BOTGRD
      S(MN,I,J,KBOT)=SNEW
      SBAR=SBAR+ABS(SNEW-SOLD)
    9 CONTINUE
      DO 10 K=KBOTP1,KMAXM1
      KP1=K+1
      KM1=K-1
      SKM1=SOLD
      SOLD=SKP1
      SKP1=S(MO,I,J,KP1)
      UKM1=UOLD
      UOLD=UKP1
      UKP1=U(MO,I,J,KP1)
      VKM1=VOLD
      VOLD=VKP1
      VKP1=V(MO,I,J,KP1)
      WKM1=WNEW
      WNEW=WKP1
      WKP1=W(I,J,KP1)
      SIM1=S(MO,IM1,J,K)
      SJM1=S(MO,I,JM1,K)
      SJP1=S(MO,I,JP1,K)
      UIM1=U(MO,IM1,J,K)
      UJM1=U(MO,I,JM1,K)
      UJP1=U(MO,I,JP1,K)
      VIM1=V(MO,IM1,J,K)
      VJM1=V(MO,I,JM1,K)
      VJP1=V(MO,I,JP1,K)
      PIM1=P(IM1,J,K)
      PJM1=P(I,JM1,K)
      PJP1=P(I,JP1,K)
      IF(I.EQ.IMAX)GO TO 7
      SIP1=S(MO,IP1,J,K)
```

```
      UIP1=U(MO,IP1,J,K)
      VIP1=V(MO,IP1,J,K)
      PIP1=P(IP1,J,K)
      GO TO 8
    7 SIP1=1.0
      UIP1=UOLD
      VIP1=VOLD
      PIP1=1.5*P(I,J,K)-0.5*PIM1
    8 CONTINUE
C        THE RICHARDSON NO. IS COMPUTED
      QKP1=SQRT(UKP1*UKP1+VKP1*VKP1)
      QKM1=SQRT(UKM1*UKM1+VKM1*VKM1)
      DQ=QKP1-QKM1
      RICH=CRICH*(SKP1-SKM1)/(DQ*DQ+1.0E-6)
      IF(RICH.LT.0.0)RICH=0.0
      RHOINV=1.0-BETA*SOLD
C        NEW VALUES OF U,V,& S AT INTERIOR GRID PTS. ARE COMPUTED
      CALL UVSNEW
      S(MN,I,J,K)=SNEW
      SBAR=SBAR+ABS(SNEW-SOLD)
      U(MN,I,J,K)=UNEW
      UBAR=UBAR+ABS(UNEW-UOLD)
      V(MN,I,J,K)=VNEW
      VBAR=VBAR+ABS(VNEW-VOLD)
   10 CONTINUE
C        NEW VALUES OF U & V AT THE BOTTOM GRID PT. IS DETERMINED BY A
C        LN CURVE FIT
      IF(J.EQ.JWAG)GO TO 13
      U(MN,I,J,KBOT)=U(MN,I,J,KBOTP1)*FUNLN
   13 CONTINUE
      V(MN,I,J,KBOT)=V(MN,I,J,KBOTP1)*FUNLN
      K=KMAX
      SIM1=S(MO,IM1,J,K)
      SJM1=S(MO,I,JM1,K)
      SJP1=S(MO,I,JP1,K)
      UIM1=U(MO,IM1,J,K)
```

```
      UJM1=U(MO,I,JM1,K)
      UJP1=U(MO,I,JP1,K)
      VIM1=V(MO,IM1,J,K)
      VJM1=V(MO,I,JM1,K)
      VJP1=V(MO,I,JP1,K)
      PIM1=P(IM1,J,K)
      PJM1=P(I,JM1,K)
      PJP1=P(I,JP1,K)
      SKM1=SOLD
      SOLD=SKP1
      UKM1=UOLD
      UOLD=UKP1
      VKM1=VOLD
      VOLD=VKP1
      WKM1 = WNEW
      WNEW = WKP1
      WKP1 = 0.0
      RHOINV=1.0-BETA*SOLD
C        VALUES OF U,V,&S ABOVE THE SURFACE ARE THE SAME AS THOSE ON THE
C        SURFACE.
      IF(I.EQ.IMAX) GO TO 11
      PIP1=P(IP1,J,K)
      SIP1=S(MO,IP1,J,K)
      UIP1=U(MO,IP1,J,K)
      VIP1=V(MO,IP1,J,K)
      GO TO 12
   11 SIP1=1.0
      UIP1=UOLD
      VIP1=VOLD
      PIP1=1.5*P(I,J,K)-0.5*PIM1
   12 RICH=0.5*RICH
      CALL SURFAC
      S(MN,I,J,K)=SNEW
      SBAR=SBAR+ABS(SNEW-SOLD)
      U(MN,I,J,K)=UNEW
      UBAR=UBAR+ABS(UNEW-UOLD)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      V(MN,I,J,K)=VNEW
      VBAR=VBAR+ABS(VNEW-VCLD)
   20 CONTINUE
      JWM1=JW-1
      JEP1=JE+1
      KBOTW=KFLOOR(I,JWM1)
      KBOTE=KFLOOR(I,JEP1)
      DO 24 K=KBOTW,KMAX
      IF(Z(K).GE.ZBANKW(I)) GO TO 22
      S(MN,I,JWM1,K)=S(MN,I,JW,K)
   22 IF(X1.LT.XBANKN.OR.X1.GT.XBANKS) GO TO 24
      U(MN,I,JWM1,K)=U(MN,I,JW,K)
      V(MN,I,JWM1,K)=V(MN,I,JW,K)
      S(MN,I,JWM1,K)=S(MN,I,JW,K)
   24 CONTINUE
      DO 28 K=KBOTE,KMAX
      IF(Z(K).GE.ZBANKE(I)) GO TO 28
      S(MN,I,JEP1,K)=S(MN,I,JE,K)
   28 CONTINUE
   30 CONTINUE
      I=1
      JW=JWEST(1)
      JE=JEAST(1)
      DO 32 J=JW,JE
      KBOT=KFLOOR(1,J)
      DO 32 K=KBOT,KMAX
      IF(Z(K).GE.ZBANKN(J)) GO TO 32
      S(MN,I,J,K)=S(MN,2,J,K)
   32 CONTINUE
      I=IMAX
      JW=JWEST(IMAX)
      JE=JEAST(IMAX)
      DO 34 J=JW,JE
      KBOT=KFLOOR(IMAX,J)
      DO 34 K=KBOT,KMAX
      IF(Z(K).GE.ZBANKS(J)) GO TO 34
```

```
      S(MN,I,J,K)=S(MN,IMAXM1,J,K)
   34 CONTINUE
      DO 36 I=2,IMAX
      JW=JWEST(I)
      JE=JEAST(I)
      DO 36 J=JW,JE
      KBOT=KFLOOR(I,J)
      IF(KBOT.EQ.1) GO TO 36
      KBOTM1=KBOT-1
      S(MN,I,J,KBOTM1)=S(MN,I,J,KBOT)
   36 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE UVSNEW
C        THIS SUBROUTINE CALC. NEW VALUES OF U,V,& S
      COMMON/DIMEN3/DUDT(12,20,11),DVDT(12,20,11),DHDT(12,20),
     1              SURF(12,20)
      COMMON/STEP/N,MN,MC,NBAR,NPRINT,NMAX,KEYOUT,LTAPE
      COMMON/GRID/DX,DY,DZ,DT,DT02,DXINV,DYINV,DZINV,SYJ,SY2,SYJP1,SYJM1
      COMMON/LIMITS/IMAX,IMAXM1,JMAX,JMAXM1,KMAX,KMAXM1,JWAG,KMXC,
     &       KMXCM1,KCHANL
      COMMON/VELCTY/UNEW,UOLD,UIP1,UIM1,UJP1,UJM1,UKP1,UKM1,
     &              VNEW,VOLD,VIP1,VIM1,VJP1,VJM1,VKP1,VKM1,
     &              WNEW,WIP1,WIM1,WJP1,WJM1,WKP1,WKM1
      COMMON/CONC/SNEW,SOLD,SIP1,SIM1,SJP1,SJM1,SKP1,SKM1,RHO,RHOINV
      COMMON/PRES/PK,PIP1,PIM1,PJP1,PJM1
      COMMON/FORCES/FWINDX,FWINDY,F
      COMMON/TURB/COEFVH,COEFVZ,COEFDH,COEFDZ,RICH,CRICH
      COMMON/INDEX/I,J,K
      COMMON/MISC/TOPLYR,SBAR,UBAR,VBAR,VISC,DIFFUS,PREFIX,ACCEL
      COMMON/RIVERS/YW(4),YE(4),DEPTH(4),URIV(4),RMOMAF(4),JRIV(4)
      COMMON/PASS/XBANKN,XBANKS,CUTPT,KCUT
      WINDX=0.0
      WINDY=0.0
      GO TO 5
C        THIS IS A SPECIAL ENTRY FOR THE SURFACE
      ENTRY SURFAC
      COMP=RHOINV/TOPLYR
      WINDX=FWINDX*COMP
      WINDY=FWINDY*COMP
    5 CONTINUE
      IF(UOLD.GT.0.0) GO TO 10
      COMPU1=UIP1*UIP1-UOLD*UOLD
      COMPV1=VIP1*UIP1-VOLD*UOLD
      COMPS1=0.5*(UIP1+UOLD)*(SIP1-SOLD)
      GO TO 15
   10 IF(I.NE.2)GO TO 11
      DO 12 NN=1,4
      IF(JRIV(NN).EQ.J)GO TO 13
```

```
   12 CONTINUE
      GO TO 11
   13 COMPU1=UOLD*UOLD-UIM1*UIM1*RMOMAF(NN)
      GO TO 14
   11 COMPU1=UOLD*UOLD-UIM1*UIM1
   14 COMPV1=VOLD*UOLD-VIM1*UIM1
      COMPS1=0.5*(UOLD+UIM1)*(SOLD-SIM1)
   15 COMPU2=0.5*RHOINV*(PIP1-PIM1)
      COMPV2=0.5*RHOINV*(PJP1-PJM1)
      IF(VOLD.GT.0.0) GO TO 20
      SY1=0.5*(SYJ+SYJP1)
      COMPU3=UJP1*VJP1-UOLD*VOLD
      COMPV3=VJP1*VJP1-VOLD*VOLD
      COMPS3=0.5*(VJP1+VOLD)*(SJP1-SOLD)
      GO TO 25
   20 SY1=0.5*(SYJ+SYJM1)
      COMPU3=UOLD*VOLD-UJM1*VJM1
      COMPV3=VOLD*VOLD-VJM1*VJM1
      COMPS3=0.5*(VOLD+VJM1)*(SOLD-SJM1)
   25 IF(WNEW.GT.0.0) GO TO 30
      COMPU4=UKP1*WKP1-UOLD*WNEW
      COMPV4=VKP1*WKP1-VOLD*WNEW
      COMPS4=0.5*(WKP1+WNEW)*(SKP1-SOLD)
      GO TO 35
   30 COMPU4=UOLD*WNEW-UKM1*WKM1
      COMPV4=VOLD*WNEW-VKM1*WKM1
      COMPS4=0.5*(WNEW+WKM1)*(SOLD-SKM1)
   35 CONTINUE
C
C        NOW SOLVING THE X-MOMENTUM EQUATION.
      CALL VISCUS
C
C        DU/DT= -(D(U*U)/DX + D(U*V)/DY + D(U*W)/DZ + (1.0/RHO)*DP/DX)
C               + COEFVX*(DSQU/DXSQ) + COEFVY*(DSQU/DYSQ) + COEFVZ*(DSQU/DZSQ)
C               + FWINDX/(RHO*DEPTH)
C
```

```
      DUODT=-DXINV*(COMPU1+COMPU2)-SY1*DYINV*COMPU3
     & -DZINV*COMPU4+VISC+F*VOLD+WINDX
      UNEW=UOLD+(DUODT+DUDT(I,J,K))*DT02
      DUDT(I,J,K)=DUODT
C        NOW SOLVING THE Y-MOMEMTUM EQUATION.
      CALL VVISC
C
C        DV/DT= -(D(U*V)/DX + D(V*V)/DY + D(V*W)/DZ + (1.0/RHO)*DP/DY)
C                 + COEFVX*(DSQV/DXSQ) + COEFVY*(DSQV/DYSQ) + COEFVZ*(DSQV/DZSQ)
C                 + FWINDY/(RHO*DEPTH)
C
      DVODT=-DXINV*COMPV1-SY1*DYINV*(COMPV2+COMPV3)
     & -DZINV*COMPV4+VISC-F*UOLD+WINDY
      VNEW=VOLD+(DVODT+DVDT(I,J,K))*DT02
      DVDT(I,J,K)=DVODT
C        NOW SOLVING THE CONSERVATION OF SALT EQUATION.
   40 CALL DIFUSE
      DSODT=-DXINV*COMPS1-SY1*DYINV*COMPS3
     & -DZINV*COMPS4+DIFFUS
      SNEW=SOLD+DSODT*DT*ACCEL
      IF(SNEW.LT.0.0)SNEW=0.0
      IF(I.EQ.IMAX.AND.K.LE.KCUT)SNEW=1.0
      RETURN
C        THIS IS A SPECIAL ENTRY FOR CALC. S AT THE BOTTOM GRID PT.
      ENTRY BOTGRD
      RICH=0.0
      IF(UOLD.GT.0.0) GO TO 45
      COMPS1=0.5*(UIP1+UOLD)*(SIP1-SOLD)
      GO TO 50
   45 COMPS1=0.5*(UOLD+UIM1)*(SOLD-SIM1)
   50 IF(VOLD.GT.0.0) GO TO 55
      SY1=0.5*(SYJ+SYJP1)
      COMPS3=0.5*(VJP1+VOLD)*(SJP1-SOLD)
      GO TO 60
   55 SY1=0.5*(SYJ+SYJM1)
      COMPS3=0.5*(VOLD+VJM1)*(SOLD-SJM1)
```

```
60 COMPS4=0.0
   GO TO 40
   END
```

```
      SUBROUTINE VISCUS
      COMMON/GRID/DX,DY,DZ,DT,DTO2,DXINV,DYINV,DZINV,SYJ,SY2,SYJP1,SYJM1
      COMMON/VELCTY/UNEW,UOLD,UIP1,UIM1,UJP1,UJM1,UKP1,UKM1,
     &              VNEW,VOLD,VIP1,VIM1,VJP1,VJM1,VKP1,VKM1,
     &              WNEW,WIP1,WIM1,WJP1,WJM1,WKP1,WKM1
      COMMON/CONC/SNEW,SOLD,SIP1,SIM1,SJP1,SJM1,SKP1,SKM1,RHO,RHOINV
      COMMON/TURB/COEFVH,COEFVZ,COEFDH,COEFDZ,RICH,CRICH
      COMMON/MISC/TOPLYR,SBAR,UBAR,VBAR,VISC,DIFFUS,PREFIX,ACCEL
C
C        EDDY VISCOSITY TERMS ARE CALCULATED.
C
C        THIS SECTION COMPUTES TERMS FOR THE X-MOM. EQ.
C
      DAMPV=1.0/SQRT(1.0+10.0*RICH)
      TWOU=UOLD+UOLD
      XCOMP=DXINV*DXINV*(UIP1-TWOU+UIM1)
      YCOMP1=SY2*0.5*DYINV*(UJP1-UJM1)
      YCOMP2=PREFIX*PREFIX*(UJP1-TWOU+UJM1)
      ZCOMP=DZINV*DZINV*(UKP1-TWOU+UKM1)
      GO TO 10
      ENTRY VVISC
C
C        THIS SECTION COMPUTES TERMS FOR THE Y-MOM. EQ.
C
      TWOV=VOLD+VOLD
      XCOMP=DXINV*DXINV*(VIP1-TWOV+VIM1)
      YCOMP1=SY2*0.5*DYINV*(VJP1-VJM1)
      YCOMP2=PREFIX*PREFIX*(VJP1-TWOV+VJM1)
      ZCOMP=DZINV*DZINV*(VKP1-TWOV+VKM1)
   10 VISC=COEFVH*(XCOMP+YCOMP1+YCOMP2)+COEFVZ*DAMPV*ZCOMP
      RETURN
C
      ENTRY DIFUSE
C
C        EDDY DIFFUSION TERMS ARE CALCULATED.
C
```

```
DAMPD=1.0/(1.0+3.333*RICH)**1.5
TWOS=SOLD+SOLD
XCOMP=DXINV*DXINV*(SIP1-TWOS+SIM1)
YCOMP1=SY2*0.5*DYINV*(SJP1-SJM1)
YCOMP2=PREFIX*PREFIX*(SJP1-TWOS+SJM1)
ZCOMP=DZINV*DZINV*(SKP1-TWOS+SKM1)
DIFFUS=COEFDH*(XCOMP+YCOMP1+YCOMP2)+COEFDZ*DAMPD*ZCOMP
RETURN
END
```

```
      SUBROUTINE VOLOIL
C
C        NEW VALUES OF W & SURFACE HTS. ARE CALC.
C
      COMMON/DIMEN1/U(2,12,20,11),V(2,12,20,11),W(12,20,11)
      COMMON/DIMEN3/DUDT(12,20,11),DVDT(12,20,11),DHDT(12,20),
     1              SURF(12,20)
      COMMON/DIMEN4/UC(2,12,11),WC(12,11),SC(2,12,11),PC(12,11),
     &        DUCDT(12,11)
      COMMON/FLOOR/KFLOOR(12,20),ZB(12,20),KFLORC(12),ZBC(12)
      COMMON/COORD/SOUTH(22),EAST(22),WEST(22),JEAST(22),JWEST(22)
      COMMON/LIMITS/IMAX,IMAXM1,JMAX,JMAXM1,KMAX,KMAXM1,JWAG,KMXC,
     &        KMXCM1,KCHANL
      COMMON/PULL/X(12),Y(20),Z(11),SY(20),SYY(20),YK2,ZC(11)
      COMMON/GRID/DX,DY,DZ,DT,DT02,DXINV,DYINV,DZINV,SYJ,SY2,SYJP1,SYJM1
      COMMON/STEP/N,MN,MO,NBAR,NPRINT,NMAX,KEYOUT,LTAPE
      COMMON/CODES/LW,LR,LT
      DIMENSION TEMPRY(20),SAVE(20)
      DO 5 J=1,JMAX
    5 SAVE(J)=SURF(1,J)
      JWIM1=JWEST(1)
      JEIM1=JEAST(1)
      DO 30 I=2,IMAXM1
      IM1=I-1
      IP1=I+1
      JW=JWEST(I)
      JE=JEAST(I)
      SURF1=SURF(I,JW)
      SURFJP=SURF1
      KCMAX=KFLOOR(I,JWAG)+KCHANL-1
      DO 20 J=JW,JE
      JP1=J+1
      JM1=J-1
      SY1=SY(J)
      Y1=Y(J)
      YJP1=Y(JP1)
```

```
      YJM1=Y(JM1)
      DELY=0.5*(YJP1-YJM1)
      DELYIN=1.0/DELY
      DELY1=Y1-YJM1
      DELY2=YJP1-Y1
      SURFJM=SURF1
      SURF1=SURFJP
      SURFJP=SURF(I,JP1)
      SURFIM=SURF(IM1,J)
      SURFIP=SURF(IP1,J)
      KBOT=KFLOOR(I,J)
      KBOTP1=KBOT+1
      K=KBOT
      U1=U(MN,I,J,K)
      UIM1=U(MN,IM1,J,K)
      UIP1=U(MN,IP1,J,K)
      V1=V(MN,I,J,K)
      VJM1=V(MN,I,JM1,K)
      VJP1=V(MN,I,JP1,K)
      Z1=Z(K)
      ZB1=ZB(I,J)
      ZBIM=0.5*(ZB1+ZB(IM1,J))
      ZBIP=0.5*(ZB1+ZB(IP1,J))
      ZBJM=0.5*(ZB1+ZB(I,JM1))
      ZBJP=0.5*(ZB1+ZB(I,JP1))
      UIN=0.25*(U1+UIM1)
      UOUT=0.25*(U1+UIP1)
      VIN=0.25*(V1+VJM1)
      VOUT=0.25*(V1+VJP1)
      WNEW=DXINV*(UIN*(Z1-ZBIM)-UOUT*(Z1-ZBIP))
     &     +DELYIN*(VIN*(Z1-ZBJM)-VOUT*(Z1-ZBJP))
      IF(J.NE.JWAG) GO TO 6
      WNEW=WNEW+WC(I,KCMAX)*0.1906
    6 W(I,J,KBOT) = WNEW
      DO 10 K=KBOTP1,KMAX
      KM1=K-1
```

```
      UIMKM1=UIM1
      UIM1=U(MN,IM1,J,K)
      UIPKM1=UIP1
      UIP1=U(MN,IP1,J,K)
      VJMKM1=VJM1
      VJM1=V(MN,I,JM1,K)
      VJPKM1=VJP1
      VJP1=V(MN,I,JP1,K)
      UIN=0.25*(UIM1+UIMKM1)
      UOUT=0.25*(UIP1+UIPKM1)
      VIN=0.25*(VJM1+VJMKM1)
      VOUT=0.25*(VJP1+VJPKM1)
      WIN=0.50*WNEW+0.125*((W(IM1,J,KM1)+W(IP1,J,KM1))
     & +(W(I,JM1,KM1)*DELY1+W(I,JP1,KM1)*DELY2)*DELYIN)
      WNEW=WIN-((UOUT-UIN)*DXINV+(VOUT-VIN)*DELYIN)*DZ
   10 W(I,J,K)=WNEW
      UKM1=U(MN,I,J,KMAXM1)
      U1=U(MN,I,J,KMAX)
      VKM1=V(MN,I,J,KMAXM1)
      V1=V(MN,I,J,KMAX)
      UIN=0.25*(U1+UIM1+UKM1+UIMKM1)
      UOUT=0.25*(U1+UIP1+UKM1+UIPKM1)
      VIN=0.25*(V1+VJM1+VKM1+VJMKM1)
      VOUT=0.25*(V1+VJP1+VKM1+VJPKM1)
      SURFAV=0.5*SURF1+0.125*((SURFIM+SURFIP)
     & +(SURFJM*DELY1+SURFJP*DELY2)*DELYIN)
      DHODT=WIN+0.5*((UIN*(SURF1+SURFIM)-UOUT*(SURF1+SURFIP))*DXINV
     & +(VIN*(SURF1+SURFJM)-VOUT*(SURF1+SURFJP))*DELYIN)
      SURF2=SURFAV+(DHODT+DHDT(I,J))*DTO2
      DHDT(I,J)=DHODT
      TEMPRY(J)=SURF2
      IF(SURF2.GE.0.5*DZ) GO TO 20
      WRITE(LW,1000) N,I,J
      KEYOUT=1
      RETURN
   20 CONTINUE
```

```
      DO 21 J=JWIM1,JEIM1
   21 SURF(IM1,J)=SAVE(J)
      DO 22 J=JW,JE
   22 SAVE(J)=TEMPRY(J)
      JWIM1=JW
      JEIM1=JE
      SURF(IM1,JW-1) = SURF(IM1,JW)
      SURF(IM1,JE+1) = SURF(IM1,JE)
   30 CONTINUE
      DO 39 J = JW,JE
   39 SURF(IMAXM1,J) = TEMPRY(J)
      SURF(IMAXM1,JW-1) = SURF(IMAXM1,JW)
      SURF(IMAXM1,JE+1) = SURF(IMAXM1,JE)
      DO 40 J=1,JMAX
      SURF(1,J)=SURF(2,J)
      KBOT=KFLOOR(1,J)
      DO 50 K=KBOT,KMAX
   50 W(1,J,K)=W(2,J,K)
      KBOT=KFLOOR(IMAX,J)
      DO 60 K=KBOT,KMAX
   60 W(IMAX,J,K)=W(IMAXM1,J,K)
   40 CONTINUE
 1000 FORMAT(/,5X,' I AM GOING TO QUIT BECAUSE THE SURF IS TOO ROUGH AT'
     & ,5X,2HN=I4,5X,2HI=I3,5X,2HJ=I3)
      RETURN
      END
```

```
      SUBROUTINE PRESS
      COMMON/DIMEN1/U(2,12,2C,11),V(2,12,20,11),W(12,20,11)
      COMMON/DIMEN2/S(2,12,20,11),P(12,20,11)
      COMMON/DIMEN3/DUDT(12,20,11),DVDT(12,20,11),DHOT(12,20),
     1               SURF(12,20)
      COMMON/DIMEN4/UC(2,12,11),WC(12,11),SC(2,12,11),PC(12,11),
     &        DUCDT(12,11)
      COMMON/FLOOR/KFLOOR(12,20),ZB(12,20),KFLORC(12),ZBC(12)
      COMMON/COORD/SOUTH(22),EAST(22),WEST(22),JEAST(22),JWEST(22)
      COMMON/PULL/X(12),Y(20),Z(11),SY(20),SYY(20),YK2,ZC(11)
      COMMON/LIMITS/IMAX,IMAXM1,JMAX,JMAXM1,KMAX,KMAXM1,JWAG,KMXC,
     &        KMXCM1,KCHANL
      COMMON/UNITS/VREF,HREF,GRAV,PI,BETA,FETCH,YMAX,CMEGA,CDEPTH
      COMMON/GRID/DX,DY,DZ,DT,DTO2,DXINV,DYINV,DZINV,SYJ,SY2,SYJP1,SYJM1
      CCMMON/CONC/SNEW,SOLD,SIP1,SIM1,SJP1,SJM1,SKP1,SKM1,RHO,RHOINV
      COMMON/STEP/N,MN,MC,NBAR,NPRINT,NMAX,KEYOUT,LTAPE
      COMMON/CODES/LW,LR,LT
C
C        THIS IS A SUBROUTINE TO CALCULATE PRESSURE
C
      DO 40 I=1,IMAX
      IM1=I-1
      IP1=I+1
      JE=JEAST(I)
      JW=JWEST(I)
      DO 30 J=JW,JE
      JM1=J-1
      JP1=J+1
      K=KMAX
      SY1=SY(J)
      SK=S(MN,I,J,K)
      RHO=1.0+BETA*SK
      IF(I.EQ.1.OR.I.EQ.IMAX) GO TO 12
      COMPX=(U(MN,IP1,J,K)*W(IP1,J,K)-U(MN,IM1,J,K)*W(IM1,J,K))
     & *(0.5*DXINV)
      GO TO 14
```

```
12 COMPX=0.0
14 COMPY=(V(MN,I,JP1,K)*W(I,JP1,K)-V(MN,I,JM1,K)*W(I,JM1,K))
  & *(0.5*SY1*DYINV)
   PK=RHO*(1.0+COMPX+COMPY)*(SURF(I,J)-DZ)
   P(I,J,KMAX)=PK
   WK=W(I,J,KMAX)
   KBOT=KFLOOR(I,J)
   KKMAX=KMAX-KBOT
   DO 20 KK=1,KKMAX
   K=KMAX-KK
   PKP1=PK
   SKP1=SK
   SK=S(MN,I,J,K)
   WKP1=WK
   WK=W(I,J,K)
   RHOAVE=1.0+0.5*BETA*(SK+SKP1)
   IF(I.EQ.1.OR.I.EQ.IMAX) GO TO 16
   COMPX=(U(MN,IP1,J,K)*W(IP1,J,K)-U(MN,IM1,J,K)*W(IM1,J,K))
  & *(0.5*DXINV)
   GO TO 18
16 COMPX=0.0
18 COMPY=(V(MN,I,JP1,K)*W(I,JP1,K)-V(MN,I,JM1,K)*W(I,JM1,K))
  & *(0.5*SY1*DYINV)
   COMPZ=WKP1*WKP1-WK*WK
   PK=PKP1+RHOAVE*((1.C+CCMPX+COMPY)*DZ+COMPZ)
   P(I,J,K)=PK
   IF(PK.GE.0.0) GO TO 20
   KEYOUT=1
   WRITE(LW,1000)N,I,J,K
   RETURN
20 CONTINUE
3C CONTINUE
   KBOTW=KFLOOR(I,JW)
   KBOTE=KFLOOR(I,JE)
   DO 34 K=KBOTW,KMAX
34 P(I,JW-1,K)=P(I,JW,K)
```

```
      DO 36 K=KBOTE,KMAX
   36 P(I,JE+1,K)=P(I-JE,K)
   40 CONTINUE
 1000 FORMAT(/,5X,'CONGRATULATIONS BILL. YOU HAVE DISCOVERED ANTIMATTER
     &AT',5X,2HN=I4,5X,2HI=I3,5X,2HJ=I3,5X,2HK=I3)
      RETURN
      END
```

```
      SUBROUTINE CHANEL
      COMMON/DIMEN4/UC(2,12,11),WC(12,11),SC(2,12,11),PC(12,11),
     &       DUCDT(12,11)
      COMMON/FLOOR/KFLOOR(12,20),ZB(12,20),KFLORC(12),ZBC(12)
      COMMON/PULL/X(12),Y(20),Z(11),SY(20),SYY(20),YK2,ZC(11)
      COMMON/LIMITS/IMAX,IMAXM1,JMAX,JMAXM1,KMAX,KMAXM1,JWAG,KMXC,
     &       KMXCM1,KCHANL
      COMMON/MISCC/SCBAR,UCBAR,VISCC,DIFUSC
      COMMON/VLCTYC/UCNEW,UCOLD,UCIM1,UCIP1,UCKM1,UCKP1,WCNEW,WCKM1,WCKP
     &1,WCIM1
      COMMON/STEP/N,MN,MO,NBAR,NPRINT,NMAX,KEYOUT,LTAPE
      COMMON/CODES/LW,LR,LT
C        STEP FORWARD IN TIME TO CALCULATE NEW VALUES OF UC AND SC.
      CALL SETUPC
C        ADJUST WC FOR COMPATABILITY WITH UC.
      CALL VOLDLC
      RETURN
      ENTRY PRESC1
C        UPDATE PC OVER THE FIELD.
      CALL PRESSC
      IF(KEYOUT.NE.0)GO TO 100
C        ALL DEPENDENT VARIABLES HAVE NOW BE COMPUTED FOR THIS TIME
C        STEP.
      IF(N.NE.NBAR*(N/NBAR))GO TO 20
      WRITE(LW,1000)N,SCBAR,UCBAR
   20 IF(N.NE.NPRINT*(N/NBAR))GO TO 30
C        AN INTERMEDIATF WRITE STATEMENT COULD GO HERE.
   30 IF(N.LT.NMAX) RETURN
  100 WRITE(LW,1000)N,SCBAR,UCBAR
      RETURN
 1000 FORMAT(5X,'N=',I4,5X,'SCBAR=',1PE13.6,5X,'UCBAR=',E13.6)
      END
```

```
      SUBROUTINE SWEDGE
C        LOCATE POSITION OF SALT WEDGE INTERFACE(0.5 ISOHALINE)
      COMMON/DIMEN4/UC(2,12,11),WC(12,11),SC(2,12,11),PC(12,11),
     &       DUCDT(12,11)
      COMMON/FLOOR/KFLOOR(12,20),ZB(12,20),KFLORC(12),ZBC(12)
      COMMON/GRID/DX,DY,DZ,DT,DTO2,DXINV,DYINV,DZINV,SYJ,SY2,SYJP1,SYJM1
      COMMON/PULL/X(12),Y(20),Z(11),SY(20),SYY(20),YK2,ZC(11)
      COMMON/LIMITS/IMAX,IMAXM1,JMAX,JMAXM1,KMAX,KMAXM1,JWAG,KMXC,
     &       KMXCM1,KCHANL
      COMMON/STEP/N,MN,MO,NBAR,NPRINT,NMAX,KEYOUT,LTAPE
      COMMON/WEDGE/ZW(17)
      DO 30 I=1,IMAX
      KBOT=KFLORC(I)
      IF(SC(MO,I,KBOT).LT.0.5)GO TO 25
      K=KFLOOR(I,JWAG)+KCHANL
      IF(SC(MO,I,K).GT.0.5) GO TO 35
   20 K=K-1
      IF(SC(MO,I,K).LT.0.5)GO TO 20
      ZW(I)=ZC(K)+(SC(MO,I,K)-0.5)/(SC(MO,I,K)-SC(MO,I,K+1))*DZ
      GO TO 30
   35 ZW(I) = ZC(K)
      GO TO 30
   25 ZW(I)=ZBC(I)
   30 CONTINUE
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE SETUPC
C        VALUES ARE CALLED FROM DIMENSIONED VARIABLES AND ASSIGNED TO
C        UNDIMENSIONED VARIABLES WHICH WILL BE USED TO CALCULATE NEW
C        VALUES OF UC AND SC
      COMMON/DIMEN1/U(2,12,20,11),V(2,12,20,11),W(12,20,11)
      COMMON/DIMEN2/S(2,12,20,11),P(12,20,11)
      COMMON/DIMEN4/UC(2,12,11),WC(12,11),SC(2,12,11),PC(12,11),
     &        DUCDT(12,11)
      COMMON/FLOOR/KFLOOR(12,20),ZB(12,20),KFLORC(12),ZBC(12)
      COMMON/PULL/X(12),Y(20),Z(11),SY(20),SYY(20),YK2,ZC(11)
      COMMON/LIMITS/IMAX,IMAXM1,JMAX,JMAXM1,KMAX,KMAXM1,JWAG,KMXC,
     &        KMXCM1,KCHANL
      COMMON/GRID/DX,DY,DZ,DT,DT02,DXINV,DYINV,DZINV,SYJ,SY2,SYJP1,SYJM1
      COMMON/STEP/N,MN,MC,NBAR,NPRINT,NMAX,KEYOUT,LTAPE
      COMMON/UNITS/VREF,HREF,GRAV,PI,BETA,FETCH,YMAX,OMEGA,CDEPTH
      COMMON/CONCC/SCNEW,SCOLD,SCIM1,SCIP1,SCKM1,SCKP1,RHO,RHOINV
      COMMON/VLCTYC/UCNEW,UCOLD,UCIM1,UCIP1,UCKM1,UCKP1,WCNEW,WCKM1,WCKP
     &1,WCIM1
      COMMON/PRESC/PCK,PCIP1,PCIM1
      COMMON/TURB/COEFVH,COEFVZ,COEFDH,COEFDZ,RICH,CRICH
      COMMON/LN/FUNLN1,FUNKAP
      COMMON/INDEX/I,J,K
      COMMON/MISCC/SCBAR,UCBAR,VISCC,DIFUSC
      UCBAR=0.0
      SCBAR=0.0
      DO 30 I=2,IMAX
      IP1=I+1
      IM1=I-1
      KBOT=KFLORC(I)
      KBOTP1=KBOT+1
      ZB1=ZBC(I)
C       THIS IS A SERIES APPROXIMATICN OF A LN FUNCTION
      ALPHA1=5.0*(ZC(KBOTP1)-ZB1)-1.0
      ALPHA2=ALPHA1*ALPHA1
      ALPHA3=ALPHA2*ALPHA1
      ALPHA4=ALPHA3*ALPHA1
```

```
      ALPHA5=ALPHA4*ALPHA1
      FUNLN2=ALPHA1-0.5*ALPHA2+0.3333*ALPHA3-0.25*ALPHA4+0.2*ALPHA5
      FUNLN=(1.0-DZ/((ZC(KBOT)-ZB1+0.5*DZ)*(FUNLN1+FUNLN2+FUNKAP)))
      SCOLD=SC(MO,I,KBOT)
      UCOLD=UC(MO,I,KBOT)
      WCNEW=WC(I,KBOT)
      SCKM1=SCOLD
      SCKP1=SC(MO,I,KBOTP1)
      UCKP1=UC(MO,I,KBOTP1)
      WCKP1=WC(I,KBOTP1)
      SCIM1=SC(MO,IM1,KBOT)
      UCIM1=UC(MO,IM1,KBOT)
      IF (I.EQ.IMAX) GO TO 4
      SCIP1=SC(MO,IP1,KBOT)
      UCIP1=UC(MO,IP1,KBOT)
      GO TO 5
    4 SCIP1=1.0
      UCIP1=UCOLD
    5 CONTINUE
C        A NEW VALUE OF SC AT THE BOTTOM GRID POINT IS NOW COMPUTED
      CALL BTGRDC
      SC(MN,I,KBOT)=SCNEW
      SCBAR=ABS(SCNEW-SCOLD)+SCBAR
      KCMXM1=KFLOOR(I,JWAG)+KCHANL-2
      DO 10 K=KBOTP1,KCMXM1
      KP1=K+1
      KM1=K-1
      SCKM1=SCOLD
      SCOLD = SCKP1
      SCKP1=SC(MO,I,KP1)
      UCKM1=UCOLD
      UCOLD=UCKP1
      UCKP1=UC(MO,I,KP1)
      WCKM1=WCNEW
      WCNEW=WCKP1
      WCKP1=WC(I,KP1)
```

```
      SCIM1=SC(MO,IM1,K)
      UCIM1=UC(MO,IM1,K)
      PCIM1=PC(IM1,K)
      IF(I.EQ.IMAX)GO TO 7
      SCIP1=SC(MO,IP1,K)
      UCIP1=UC(MO,IP1,K)
      PCIP1=PC(IP1,K)
      GO TO 8
    7 SCIP1=1.0
      UCIP1=UCOLD
      PCIP1=1.5*PC(I,K)-0.5*PCIM1
    8 CONTINUE
C     THE RICHARDSON NO. IS COMPUTED
      DQ=ABS(UCKP1)-ABS(UCKM1)
      RICH=CRICH*(SCKP1-SCKM1)/(DQ*DQ+1.0E-6)
      IF(RICH.LT.0.0)RICH=0.0
      RHOINV=1.0-BETA*SCOLD
C     NEW VALUES OF UC AND SC AT INTERIOR GRID POINTS ARE COMPUTED
      CALL USNEW
      SC(MN,I,K)=SCNEW
      SCBAR=SCBAR+ABS(SCNEW-SCOLD)
      UC(MN,I,K)=UCNEW
      UCBAR=UCBAR+ABS(UCNEW-UCOLD)
   10 CONTINUE
C     NEW VALUE OF UC AT THE BOTTOM GRID POINT IS DETERMINED
C     BY A LN CURVE FIT
      UC(MN,I,KBOT)=UC(MN,I,KBOTP1)*FUNLN
C     NEW VALUES OF UC AND SC JUST BELOW BAY-CHANNEL INTERFACE ARE
C     COMPUTED
      K=KCMXM1+1
      KP1=KFLOOR(I,JWAG)
      SCIM1=SC(MO,IM1,K)
      UCIM1=UC(MO,IM1,K)
      PCIM1=PC(IM1,K)
      SCKM1=SCOLD
      SCOLD=SCKP1
```

```
      SCKP1=S(MO,I,JWAG,KP1)
      UCKM1=UCOLD
      UCOLD=UCKP1
      UCKP1=U(MO,I,JWAG,KP1)
      WCKM1 = WCNEW
      WCNEW = WCKP1
      WCKP1=W(I,JWAG,KP1)*5.247
      IF(I.EQ.IMAX)GO TO 11
      IF(KFLOOR(IP1,JWAG).LT.KP1) GO TO 13
      SCIP1=SC(MO,IP1,K)
      UCIP1=UC(MO,IP1,K)
      PCIP1=PC(IP1,K)
      GO TO 12
   13 KIP1 = KFLOOR(IP1,JWAG)
      SCIP1=S(MO,IP1,JWAG,KIP1)
      UCIP1=U(MO,IP1,JWAG,KIP1)
      PCIP1 = P(IP1,JWAG,KIP1)
      GO TO 12
   11 SCIP1=1.0
      UCIP1=UCOLD
      PCIP1=1.5*PC(I,K)-0.5*PCIM1
   12 DQ=ABS(UCKP1)-ABS(UCKM1)
      RICH=CRICH*(SCKP1-SCKM1)/(DQ*DQ+1.0E-6)
      IF(RICH.LT.0.0) RICH=0.0
      RHOINV=1.0-BETA*SCOLD
      CALL USNEW
      SC(MN,I,K)=SCNEW
      SCBAR=SCBAR+ABS(SCNEW-SCOLD)
      UC(MN,I,K)=UCNEW
      UCBAR=UCBAR+ABS(UCNEW-UCOLD)
   30 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE  USNEW
C        THIS SUBROUTINE CALCULATE NEW VALUES OF UC AND SC
      COMMON/DIMEN4/UC(2,12,11),WC(12,11),SC(2,12,11),PC(12,11),
     &        DUCDT(12,11)
      COMMON/FLOOR/KFLOOR(12,20),ZB(12,20),KFLORC(12),ZBC(12)
      COMMON/GRID/DX,DY,DZ,DT,DTO2,DXINV,DYINV,DZINV,SYJ,SY2,SYJP1,SYJMI
      COMMON/VLCTYC/UCNEW,UCOLD,UCIM1,UCIP1,UCKM1,UCKP1,WCNEW,WCKM1,WCKP
     &1,WCIM1
      COMMON/CONCC/SCNEW,SCOLD,SCIM1,SCIP1,SCKM1,SCKP1,RHO,RHOINV
      COMMON/PRESC/PCK,PCIP1,PCIM1
      COMMON/TURB/COEFVH,COEFVZ,COEFDH,COEFDZ,RICH,CRICH
      COMMON/LIMITS/IMAX,IMAXM1,JMAX,JMAXM1,KMAX,KMAXM1,JWAG,KMXC,
     &        KMXCM1,KCHANL
      COMMON/MISC/TOPLYR,SBAR,UBAR,VBAR,VISC,DIFFUS,PREFIX,ACCEL
      COMMON/MISCC/SCBAR,UCBAR,VISCC,DIFUSC
      COMMON/INDEX/I,J,K
      IF(UCOLD.GT.0.0) GO TO 10
      COMPU1=UCIP1*UCIP1-UCOLD*UCOLD
      COMPS1=0.5*(UCIP1+UCOLD)*(SCIP1-SCOLD)
      GO TO 15
   10 COMPU1=UCOLD*UCOLD-UCIM1*UCIM1
      COMPS1=0.5*(UCOLD+UCIM1)*(SCOLD-SCIM1)
   15 COMPU2=0.5*RHOINV*(PCIP1-PCIM1)
      IF(WCNEW.GT.0.0)GO TO 30
      COMPU4=UCKP1*WCKP1-UCOLD*WCNEW
      COMPS4=0.5*(WCKP1+WCNEW)*(SCKP1-SCOLD)
      GO TO 35
   30 COMPU4=UCOLD*WCNEW-UCKM1*WCKM1
      COMPS4=0.5*(WCNEW+WCKM1)*(SCOLD-SCKM1)
   35 CONTINUE
C        NOW SOLVING THE X-MOMENTUM EQUATION.
      CALL VISCSC
C
C     DU/DT= -(D(U*U)/DX+D(U*V)/DZ+(1.0/RHO)*DP/DX)
C             +COEFVX*(DSQU/DXSQ)+COEFVZ*(DSOU/DZSQ)
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DUCODT=-DXINV*(COMPU1+COMPU2)-DZINV*COMPU4+VISCC
      UCNEW=UCOLD+(DUCODT+DUCDT(I,K))*DTO2
      DUCDT(I,K)=DUCODT
C        NOW SOLVING THE CONSERVATION OF SALT EQUATION
   40 CALL DIFSEC
      DSCODT=-DXINV*COMPS1-DZINV*COMPS4+DIFUSC
      SCNEW=SCOLD+DSCODT*DT*ACCEL
      IF(I.EQ.IMAX) SCNEW = 1.0
      RETURN
C        THIS IS A SPECIAL ENTRY FOR CALCULATING SC AT THE BOTTOM GRID
C        POINTS.
      ENTRY BTGRDC
      RICH=0.0
      IF(UCOLD.GT.0.0)GO TO 45
      COMPS1=0.5*(UCIP1+UCOLD)*(SCIP1-SCOLD)
      GO TO 50
   45 COMPS1=0.5*(UCOLD+UCIM1)*(SCOLD-SCIM1)
   50 COMPS4 = 0.0
      GO TO 40
      END
```

```
      SUBROUTINE VISCSC
      COMMON/GRID/DX,DY,DZ,DT,DTO2,DXINV,DYINV,DZINV,SYJ,SY2,SYJP1,SYJM1
      COMMON/MISCC/SCBAR,UCBAR,VISCC,DIFUSC
      COMMON/CONCC/SCNEW,SCOLD,SCIM1,SCIP1,SCKM1,SCKP1,RHO,RHOINV
      COMMON/TURB/COEFVH,COEFVZ,COEFDH,COEFDZ,RICH,CRICH
      COMMON/VLCTYC/UCNEW,UCOLD,UCIM1,UCIP1,UCKM1,UCKP1,WCNEW,WCKM1,WCKP
     &1,WCIM1
C        EDDY VISCOSITY TERM FOR X-MOMENTUM EQUATION IS CALCULATED
      DAMPV=1.0/SQRT(1.0+10.0*RICH)
      TWOUC=UCOLD+UCOLD
      XCOMP=DXINV*DXINV*(UCIP1-TWOUC+UCIM1)
      ZCOMP=DZINV*DZINV*(UCKP1-TWOUC+UCKM1)
      VISCC=COEFVH*XCOMP+COEFVZ*DAMPV*ZCOMP
      RETURN
C
      ENTRY DIFSEC
C
C        EDDY DIFFUSION TERM FOR SALT CONSERVATION EQUATIONS IS CALCULA
C        TED.
      DAMPD=1.0/(1.0+3.333*RICH)**1.5
      TWOSC=SCOLD+SCOLD
      XCOMP=DXINV*DXINV*(SCIP1-TWOSC+SCIM1)
      ZCOMP=DZINV*DZINV*(SCKP1-TWOSC+SCKM1)
      DIFUSC=COEFDH*XCOMP+COEFDZ*DAMPD*ZCOMP
      RETURN
      END
```

```
      SUBROUTINE VOLDLC
      COMMON/DIMEN4/UC(2,12,11),WC(12,11),SC(2,12,11),PC(12,11),
     &        DUCDT(12,11)
      COMMON/FLOOR/KFLOOR(12,20),ZB(12,20),KFLORC(12),ZBC(12)
      COMMON/PULL/X(12),Y(20),Z(11),SY(20),SYY(20),YK2,ZC(11)
      COMMON/LIMITS/IMAX,IMAXM1,JMAX,JMAXM1,KMAX,KMAXM1,JWAG,KMXC,
     &        KMXCM1,KCHANL
      COMMON/GRID/DX,DY,DZ,DT,DT02,DXINV,DYINV,DZINV,SYJ,SY2,SYJP1,SYJM1
      COMMON/STEP/N,MN,MO,NBAR,NPRINT,NMAX,KEYOUT,LTAPE
      COMMON/CODES/LW,LR,LT
      DO 30 I=2,IMAXM1
      IM1=I-1
      IP1=I+1
      KBOT=KFLORC(I)
      KBOTP1=KBOT+1
      K=KBOT
      UC1=UC(MN,I,K)
      UCIM1=UC(MN,IM1,K)
      UCIP1=UC(MN,IP1,K)
      ZC1=ZC(K)
      ZBC1=ZBC(I)
      ZBCIM=0.5*(ZBC1+ZBC(IM1))
      ZBCIP=0.5*(ZBC1+ZBC(IP1))
      UCIN=0.25*(UC1+UCIM1)
      UCOUT=0.25*(UC1+UCIP1)
      WCNEW=DXINV*(UCIN*(ZC1-ZBCIM)-UCOUT*(ZC1-ZBCIP))
      WC(I,KBOT)=WCNEW
      KKMAX=KFLOOR(I,JWAG)+KCHANL-1
      DO 10 K=KBOTP1,KKMAX
      KM1=K-1
      UCIMKM=UCIM1
      UCIM1=UC(MN,IM1,K)
      UCIPKM=UCIP1
      UCIP1=UC(MN,IP1,K)
      UCIN=0.25*(UCIM1+UCIMKM)
      UCOUT=0.25*(UCIP1+UCIPKM)
```

```
      WCIN=0.50*WCNEW+0.25*(WC(IM1,KM1)+WC(IP1,KM1))
      WCNEW=WCIN-(UCOUT-UCIN)*DXINV*DZ
   10 WC(I,K)=WCNEW
   30 CONTINUE
      KKMAX=KFLOOR(1,JWAG)+KCHANL-1
      KBOT=KFLORC(1)
      DO 50 K=KBOT,KKMAX
   50 WC(1,K)=WC(2,K)
      KKMAX=KFLOOR(IMAX,JWAG)+KCHANL-1
      KBOT=KFLORC(IMAX)
      DO 60 K=KBOT,KKMAX
   60 WC(IMAX,K)=WC(IMAXM1,K)
      RETURN
      END
```

```
      SUBROUTINE PRESSC
C        THIS SUBROUTINE CALCULATES THE PRESSURE.
      COMMON/DIMEN1/U(2,12,20,11),V(2,12,20,11),W(12,20,11)
      COMMON/DIMEN2/S(2,12,20,11),P(12,20,11)
      COMMON/DIMEN4/UC(2,12,11),WC(12,11),SC(2,12,11),PC(12,11),
     &        DUCDT(12,11)
      COMMON/FLOOR/KFLOOR(12,20),ZB(12,20),KFLORC(12),ZBC(12)
      COMMON/LIMITS/IMAX,IMAXM1,JMAX,JMAXM1,KMAX,KMAXM1,JWAG,KMXC,
     &        KMXCM1,KCHANL
      COMMON/GRID/DX,DY,DZ,DT,DTO2,DXINV,DYINV,DZINV,SYJ,SY2,SYJP1,SYJM1
      COMMON/STEP/N,MN,MC,NBAR,NPRINT,NMAX,KEYOUT,LTAPE
      COMMON/UNITS/VREF,HREF,GRAV,PI,BETA,FETCH,YMAX,CMEGA,CDEPTH
      COMMON/CODES/LW,LR,LT
      DO 40 I=1,IMAX
      IM1=I-1
      IP1=I+1
      KC=KFLOOR(I,JWAG)+KCHANL-1
      SCK=SC(MN,I,KC)
      RHO=1.0+BETA*SCK
      IF(I.EQ.1.OR.I.EQ.IMAX) GO TO 12
      COMPX=(UC(MN,IP1,KC)*WC(IP1,KC)-UC(MN,IM1,KC)*WC(IM1,KC))
     &     *(0.5*DXINV)
      GO TO 14
   12 COMPX=0.0
   14 KBOT=KFLOOR(I,JWAG)
      PCK=P(I,JWAG,KBOT)+RHO*(1.0+COMPX)*DZ
      PC(I,KC)=PCK
      WCK=WC(I,KC)
      KBOT=KFLORC(I)
      KKMAX=KC-KBOT
      DO 20 KK=1,KKMAX
      K=KC-KK
      PCKP1=PCK
      SCKP1=SCK
      SCK=SC(MN,I,K)
      WCKP1=WCK
```

```
      WCK=WC(I,K)
      RHOAVE=1.0+0.5*BETA*(SCK+SCKP1)
      IF(I.EQ.1.OR.I.EQ.IMAX)GO TO 16
      COMPX=(UC(MN,IP1,K)*WC(IP1,K)-UC(MN,IM1,K)*WC(IM1,K))*(0.5*DXINV)
      GO TO 18
   16 COMPX=0.0
   18 COMPZ=WCKP1*WCKP1-WCK*WCK
      PCK=PCKP1+RHOAVE*((1.0+COMPX)*DZ+COMPZ)
      PC(I,K)=PCK
      IF(PCK.GE.0.0)GO TO 20
      KEYOUT=1
      WRITE(LW,1000)N,I,K
      RETURN
   20 CONTINUE
   40 CONTINUE
      RETURN
 1000 FORMAT(/,5X,'CONGRATULATIONS FRED.YOU HAVE DISCOVERED ANTIMATTER
     &       AT',5X,'N=',I4,5X,'I=',I3,5X,'K=')
      END
```

```
      BLOCK DATA
C
C        VARIOUS CONSTANTS ARE DEFINED
C
      COMMON/UNITS/VREF,HREF,GRAV,PI,BETA,FETCH,YMAX,OMEGA,CDEPTH
      COMMON/STEP/N,MN,MO,NBAR,NPRINT,NMAX,KEYOUT,LTAPE
      COMMON/CODES/LW,LR,LT
C        LW,LR,LT ARE THE LOGICAL INPUT/OUTPUT UNIT CODES FOR WRITING, READING
C        AND USING A TAPE.
      DATA LW,LR,LT/6,5,3/
      DATA N,KEYOUT,MN,MO,NBAR/0,0,1,2,1/
      DATA GRAV,PI,OMEGA/9.790,3.14159,7.29E-5/
      END
```

```
DATA
          1
         12          20          9          10          11
       1100        1100          1
   50.586      26.313
   43.36       47.066
    0.5         1.4        5.0
     8.0       22.0        0.0        0.0
   10.0         0.4
   450.0      200.0       30.7
    12.2        3.0
         22
      0.0      6.0       -0.1
      1.3      5.9       -1.35
      2.6      5.85      -1.45
      3.9      6.2       -1.4
      5.2      6.0       -1.65
      6.5      6.35      -2.5
      7.8      6.4       -3.0
      9.1      5.9       -3.15
     10.4      5.4       -4.2
     11.7      4.25      -4.35
     13.0      5.0       -4.4
     14.3      5.2       -4.8
     15.6      5.4       -4.6
     16.9      6.5       -4.55
     18.2      8.25      -4.45
     19.5      9.8       -4.85
     20.8     10.9       -5.4
     22.1     13.1       -5.35
     23.4     14.0       -5.0
     24.7     14.2       -4.4
     26.0     11.2       -3.8
     27.3      0.5       -3.2
     1853.0
     -0.1        0.1        9.14       0.335       10.0
```

| | | | | |
|---|---|---|---|---|
| 1.1 | 1.35 | 8.53 | 0.213 | 13.0 |
| 4.4 | 4.63 | 6.10 | 0.168 | 17.0 |
| 5.5 | 5.8 | 7.92 | 0.229 | 18.0 |