Approved for Public Release;
Distribution Unlimited.

ENVPREDRSCHFAC
Technical Note No. 2-74

A MULTILAYER HYDRODYNAMICAL-NUMERICAL MODEL

(W. HANSEN TYPE)

MODEL DESCRIPTION AND OPERATING/RUNNING INSTRUCTIONS

By

TAIVO LAEVASTU

ENVIRONMENTAL PREDICTION RESEARCH FACILITY

Part 2 of a series of four reports

prepared for:

ENVIRONMENTAL PROTECTION AGENCY
PACIFIC NORTHWEST ENVIRONMENTAL
RESEARCH LABORATORY
CORVALLIS, OREGON

JANUARY 1974

ENVIRONMENTAL PREDICTION RESEARCH FACILITY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93940

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER ENVPREDRSCHFAC Technical Note No. 2-74

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
A Multilayer Hydrodynamical-Numerical Model (W. Hansen): Model Description and Operating/Running Instructions

5. TYPE OF REPORT & PERIOD COVERED

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Taivo Laevastu

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Environmental Prediction Research Facility
Naval Postgraduate School
Monterey, California 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Naval Air Systems Command
Department of the Navy
Washington, D.C. 20361

12. REPORT DATE
January 1974

13. NUMBER OF PAGES
60

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for Public Release; Distribution Unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Multilayer Hydrodynamical-Numerical Model
Dispersion and diffusion of pollutants

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The multilayer Hydrodynamical-Numerical (HN) model (W. Hansen type) is, in many respects, similar to Hansen's single-layer HN model. The hydrodynamical formulas present vertically integrated motion through given layers. The lower layer is partly driven by friction, which is proportional to the mean density relations of the layers, and partly by horizontal pressure gradients caused by the inclination of the surface as well as of the interface.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (Continued)

The finite difference forms are similar to those used in single-layer models, except for the additional terms dictated by the second layer. The model uses a continuity equation in its direct, physically simple and correct form.

In respect to the inputs, the model is also similar to the single-layer HN model, with the exception of additional inputs of second layer thickness and internal tides in the second layer.

Besides the basic formulas of the model, this report describes the numerical model in FORTRAN. The description of the input parameters, abbreviations used, and the program itself are given in the Appendixes.

The verification of the model is described in another report.

The model described in FORTRAN has not been fully optimized for routine production runs.

FOREWORD

This documentation of a two-layer hydrodynamical-Numerical (HN) model is the second in a series of reports from Environmental Prediction Research Facility (EPRF), to Environmental Protection Agency (EPA).

This two-layer HN model is in many respects similar to the single-layer HN model of type Walter Hansen and reference is often made to the latter in this report. The model was applied to the New York Bight and the results and verification are given in another report. It could be pointed out that similar models with three and six layers have been applied at EPRF for other areas of the Northern Hemisphere oceans.

The project was funded by Environmental Protection Agency, Pacific Northwest Environmental Research Laboratory, Corvallis, Oregon. The project officer was Mr. R. J. Callaway, Chief, Physical Oceanography Branch, National Coastal Pollution Research Program of the above mentioned EPA laboratory.

The project was carried out by the Oceanography Department of the Environmental Prediction Research Facility under the supervision of Dr. Taivo Laevastu.

CONTENTS

FOREWORD iii

1. THE BASIC MULTILAYER HYDRODYNAMICAL EQUATIONS AND THEIR FINITE DIFFERENCE FORMS 1

2. THE ESSENTIALS OF THE MULTILAYER HN MODEL 6

2.1 The Stability Criteria and Horizontal Viscosity Coefficient 6
2.2 The Grid Net and the Inputs 7
2.3 Treatment of Open Boundaries 7
2.4 Other General Information, Including Notes on Verification 8

3. REVIEW OF THE FORTRAN PROGRAM 10

3.1 The Control Program 10
3.2 Subroutine J02 10
3.3 Subroutine J03 11
3.4 Subroutine Vector 12
3.5 Subroutine J04 12
3.6 Subroutine J05 13
3.7 The Transport Function 16
3.8 Subroutine S01 16
3.9 Subroutine S08 16
3.10 Subroutine J0B6, J0B7 and the Function Subroutines 16

REFERENCES 17

APPENDIX A - INPUT PARAMETERS AND FORMATS 20

APPENDIX B - LIST OF SYMBOLS 23

APPENDIX C - PROGRAM NYAREA LISTING 29

I. BASIC MULTILAYER HYDRODYNAMICAL EQUATIONS AND THEIR FINITE DIFFERENCE FORMS

The layer-by-layer vertically integrated hydrodynamical equations were proposed by Professor Walter Hansen (personal communication). They are analogous to the well-proven single-layer HN model of the Walter Hansen type.

$$\dot{\zeta}_1 - \dot{\zeta}_2 + H_{u1}(U1_x) + H_{v1}\,(V1_y) = 0$$

$$\dot{\zeta}_2 + H_{u2}\,(U2_x) + H_{v2}\,(V2_y) = 0$$

$$\dot{U1} + \frac{r\sqrt{U1^2 + V1^2}}{H_{u1}}\,U1 - fV1 - g\,\zeta_{1x} = K^{(x)}$$

$$\dot{U2} + \frac{r\sqrt{U2^2 + V2^2}}{H_{u2}}\,U2 - fV2 - g\,\frac{\rho 1}{\rho 2}\,\zeta_{1x} - g\,(1 - \frac{\rho 1}{\rho 2})\,\zeta_{2x} = 0$$

$$\dot{V1} + \frac{r\sqrt{U1^2 + V1^2}}{H_{v1}}\,V1 + fU1 - g\,\zeta_{1y} = K^{(y)}$$

$$V2 + \frac{r\sqrt{U2^2 + V2^2}}{H_{v2}}\,V2 + fU2 - g\,\frac{\rho 1}{\rho 2}\,\zeta_{1y} - g\,(1 - \frac{\rho 1}{\rho 2})\,\zeta_{2y} = 0$$

where:

ζ_1 - surface elevation;

ζ_2 - deviation of MLD (mixed layer depth) from mean

U1, V1 - u,v components in upper layer

U2, V2 - u.v components in lower layer

r - friction coefficient

f - Coriolis parameter

g - acceleration of gravity

H - depth

ρ_1, ρ_2 - densities

$K^{(x)}$, $K^{(y)}$ - external forces

There are two inter-dependent continuity equations, one for each layer. These compute the change of sea level, the change of depth, and thickness of the layers.

The equations of motion for each layer are vertically integrated through this given layer. The lower layer is driven by internal friction and by pressure gradients.

The finite difference forms used for computation are essentially the same as in W. Hansen's single-layer model, except for the additional terms dictated by the presence of a second layer.

$$\zeta^{t+\tau}_{(n,m,1)} = \bar{\zeta}^{t-\tau}_{(n,m,1)} - \frac{\tau}{\ell} \{H^t_{u(n,m,1)} U^t_{(n,m,1)} - H^t_{u(n,m-1,1)} U^t_{(n,m-1,1)} + H^t_{v(n-1,m,1)} V^t_{(n-1,m,1)} - H^t_{v(n,m,1)} V^t_{(n,m,1)} \} - \frac{\tau}{\ell} \{H^t_{u(n,m,2)} U^t_{(n,m,2)} - H^t_{u(n,m-1,2)} U^t_{(n,m-1,2)} + H^t_{v(n-1,m,2)} V^t_{(n-1,m,2)} - H^t_{v(n,m,2)} V^t_{(n,m,2)}\}$$

$$\zeta^{t+\tau}_{(n,m,2)} = \bar{\zeta}^{t-\tau}_{(n,m,2)} - \frac{\tau}{\ell} \{H^t_{u(n,m,2)} U^t_{(n,m,2)} - H^t_{u(n,m-1,2)} U^t_{(n,m-1,2)} + H^t_{v(n-1,m,2)} + V^t_{(n-1,m,2)} - H^t_{v(n,m,2)} V^t_{(n,m,2)}\}$$

$$U^{t+2\tau}_{(n,m,1)} = \{1-[2\tau r/H^{t+2\tau}_{u(n,m,1)}] \sqrt{U^{t}_{(n,m,1)}{}^{2} + V^{*t}_{(n,m,1)}{}^{2}}\}$$

$$U^{t}_{(n,m,1)} + 2\tau f\ V^{*t}_{(n,m,1)} - \frac{\tau g}{\ell} \{\zeta^{t+\tau}_{(n,m+1,1)} -$$

$$\zeta^{t+\tau}_{(n,m,1)}\} + 2\tau X^{t+2\tau}_{(n,m)}$$

$$U^{t+2\tau}_{(n,m,2)} = \{1-[2\tau r/H^{t+2\tau}_{u(n,m,2)}] \sqrt{U^{t}_{(n,m,2)}{}^{2} + V^{*t}_{(n,m,2)}{}^{2}}\}$$

$$U^{t}_{(n,m,2)} + 2\tau f\ V^{*t}_{(n,m,2)} - \frac{\tau g}{\ell} \{\frac{\rho 1}{\rho 2} [\zeta^{t+\tau}_{(n,m+1,1)}$$

$$- \zeta^{t+\tau}_{(n,m,1)}]\} - \frac{\tau g}{\ell} \{[1 - \frac{\rho 1}{\rho 2}] [\zeta^{t+\tau}_{(n,m+1,2)}$$

$$- \zeta^{t+\tau}_{(n,m,2)}]\}$$

$$V^{t+2\tau}_{(n,m,1)} = \{1-[2\tau r/H^{t+2\tau}_{v(n,m,1)}] \sqrt{V^{t}_{(n,m,1)}{}^{2} + U^{*t}_{(n,m,1)}{}^{2}}\}$$

$$-2\tau f\ U^{*t}_{(n,m,1)} - \frac{\tau g}{\ell} \{\zeta^{t+\tau}_{(n,m,1)} - \zeta^{t+\tau}_{(n+1,m,1)}\}$$

$$+2\tau Y^{t+2\tau}_{(n,m)}$$

$$V^{t+2\tau}_{(n,m,2)} = \{1-[2\tau r/H^{t+2\tau}_{v(n,m,2)}] \sqrt{V^{t}_{(n,m,2)}{}^{2} + U^{*t}_{(n,m,2)}{}^{2}}\}$$

$$-2\tau f\ U^{*t}_{(n,m,2)} - \frac{\tau g}{\ell} \{\frac{\rho 1}{\rho 2} [\zeta^{t+\tau}_{(n,m,1)} - \zeta^{t+\tau}_{(n+1,m,1)}]\}$$

$$- \frac{\tau g}{\ell} \{[1-\frac{\rho 1}{\rho 2}] [\zeta^{t+\tau}_{(n,m,2)} - \zeta^{t+\tau}_{(n+1,m,2)}]\}$$

The computation of $\bar{U}$ and U^* is done as in the single-layer model

$$\bar{U}^t_{(n,m)} = \alpha\, U^t_{(n,m)} + \frac{1-\alpha}{4} \{U^t_{(n-1,m)} + U^t_{(n+1,m)} + U^t_{(n,m+1)} + U^t_{(n,m-1)}\}$$

$$U^{*t}_{(n,m)} = \frac{1}{4} \{ U^t_{(n,m-1)} + U^t_{(n+1,m-1)} + U^t_{(n,m)} + U^t_{(n+1,m)}\}$$

The computation of $\bar{V}$ and V^* are analogous to the corresponding U computations above. The actual depth (and thickness of the layer) is computed with the following equations:

$$H^{t+2\tau}_{u(n,m,1)} + h_{u(n,m,1)} + \frac{1}{2} \{ \zeta^{t+\tau}_{(n,m,1)} + \zeta^{t+\tau}_{(n,m+1,1)}\}$$

$$H^{t+2\tau}_{v(n,m,1)} = h_{v(n,m,1)} + \frac{1}{2} \{ \zeta^{t+\tau}_{(n,m,1)} + \zeta^{t+\tau}_{(n+1,m,1)}\}$$

The following symbols were used in the finite difference formulas above:

x,y	space coordinates
t	time
U,V	components of velocity
u,v	indicators of U and V point (location) in the grid
h	initial depth (when $\zeta = 0$)
ζ	surface elevation (for the second layer, it indicates the deviation of the depth of the layer from its prescribed mean value)
H	total depth ($H = h + \zeta$)
H_u, H_v	depths at u and v points respectively

X, Y	components of external forces
g	acceleration of gravity
f	Coriolis parameter
r	friction coefficient (bottom stress for lower layer)
α	coefficient of horizontal eddy viscosity (also acts as a smoothing coefficient)
n, m	coordinates of the grid point, 1 and 2 indicate the first (surface) or second (deep) layer
τ	half time step
ℓ	half grid length
ρ_1, ρ_2	densities of first and second layer

2. ESSENTIALS OF THE MULTILAYER HN MODEL

This chapter reviews briefly the inputs to the model, the differences between the single-layer and multilayer model, and the treatment of open boundaries. As most of the essential features of this multilayer model are similar to the single-layer model, the description of the latter should be consulted (see Part 1 of this series).

2.1 THE STABILITY CRITERIA AND HORIZONTAL VISCOSITY COEFFICIENT

The same stability criteria as for the single-layer model applies for the multilayer model:

$$0.5\ \Delta t \leq \frac{0.5\ell}{\sqrt{2gH_{max}}}$$

The H_{max} is the maximum depth or thickness of any of the layers to be expected during the computation. Thus, if original data indicates that the maximum depth (thickness) in the lower layer is 520 m, one should add, for example, 80 m for a contingent increase of this depth due to possible upwards movement of the upper boundary of the layer during computations.

The horizontal viscosity coefficient (also as a smoothing coefficient) has the same value for the upper layer as in single layer models (0.98 to 0.99). However, it has been found necessary to decrease this coefficient for the lower layers (0.90 to 0.92) in order to smooth out some disturbances in the interface, especially at the initial state of the computation. Using a higher value for the coefficient results in longer, more costly computations, as output should be taken when these initial disturbances have dissipated. The use of a lower value for the coefficient saves computer time. The final results are not affected to any great extent.

A lower value of the "bottom friction" coefficient can be used for the upper layer, as it acts as an internal friction coefficient. There is no data available as to what this coefficient should be. Preliminary numerical experiments have indicated a plausible value of 0.0015.

2.2 THE GRID NET AND THE INPUTS

The same grid net as in single-layer models is used. The sea and land table and the depths at u and v points are also introduced as in the single-layer model.

The depth of the mixed layer (the thickness of the upper layer) is given with a data statement as an initial flat field. However, if its topography is known, it can be read-in from data cards.

The program computes the deviation of the sea surface elevation from a flat field, and the deviation of the depth of the upper boundary of the lower layer either from a flat field or from a prescribed field. The program allows the disappearance of the lower layer (e.g., interception with the bottom) and/or of the upper layer (e.g., upwelling). The geographical reorientation of the grid follows the same roles as in the single layer model.

It has been found that the model behaves best when the open boundary is chosen such that the iterations are forward-stepped from the boundary.

2.3 TREATMENT OF OPEN BOUNDARIES

The input of tides at a primary open boundary is done in every time step for the upper layer the same way as in a single-layer model, using tidal harmonic constants.

$$Z = A_1\cos(\alpha_1 t-\kappa_1) + A_2\cos(\alpha_2 t-\kappa_2) \ldots$$

The internal tides must also be introduced into the second layer. Usually the magnitudes of the internal tidal waves and their phase lag is unknown. On the basis of available time series recordings of surface-to-bottom temperature profiles, it can be generalized that in most areas over the continental shelf, the tidal internal wave is 4 to 8 times higher than the surface tidal wave, and lags 1 to 3 hours behind the surface wave. This preliminary knowledge has been utilized for the input of internal tides at the open boundary.

The other open boundaries (other than with tidal input) must be left entirely open for inflow and outflow. Many ways have been tested to treat these boundaries. It could be mentioned that the bounds of various Do-loops should be properly adjusted in treatment of open boundaries in the model, depending on the orientation of the grid and input boundary. The missing values of u at the right hand boundary and v at the lower boundary can be assumed to be the same as at the previous column (m-1) and row (n-1) respectively. The method of "lubricated wall" for open boundaries is described in paragraph 3.6.

The permanent thermohaline currents are introduced into the model by prescribing properly precomputed inclinations (pressure gradient) at the boundaries. The manner of its application varies with the application area of the model. This method of input allows the input of the permanent current independently in two layers; for example, for simulation of estuarine circulation (i.e., outflow at the surface, inflow in lower layer). The outflow of rivers must be prescribed at the gridpoints in or off the river at each time step.

2.4 OTHER GENERAL INFORMATION, INCLUDING NOTES ON VERIFICATION

The multilayer HN model can be used for the same tasks (and more) as the single-layer model. The dispersion and diffusion is handled in this model in the same way as in the single-layer model, except that it can be treated separately in two layers.

The two-layer model was originally set up by Professor Walter Hansen during his visit to Monterey a few years ago. The original version was programmed by Mr. R. Mann, utilizing to a large extent, a single-layer program originally written by Professor J. Sündermann. Since then considerable changes, improvements, and adaptions have been made by the author.

The program has been tested and compared in a number of areas. However, the results have not been published previously, except for some results with a two-layer model for the Strait of Gibraltar and some with a three-layer model for the Labrador Sea by Hamilton, Williams and Laevastu (1973).

The two-layer model can also be partly verified with mareograph data. Contrary to single-layer model verification, however, total and thorough verification of the two-layer model requires data on currents and mixed layer depth fluctuations.

A description of the results of the application of a two-layer model to the New York Bight and its partial verification are described in Part 4 of this series.

3. REVIEW OF THE FORTRAN PROGRAM

The program of the two-layer model, presented in Appendix C,is written in FORTRAN and will run, with very few changes, on any machine which has a FORTRAN compiler. The necessary changes are pointed out in the below description of this model. The main peculiarity of this model (which might vary from machine to machine) is the reading and writing of extended core on CDC 6000 and 7000 series machines.

3.1 THE CONTROL PROGRAM

The set-up of the program is simple and convenient. For example, the dimension, common and equivalence statements are essentially the same in all subroutines, though all the variables are not necessarily used in all subroutines. Some equivalences are used to reduce the core requirements. Some of the variables, especially in the second layer, are put into a second common block. An extended core map in the form of comment cards is given in the control program. The rest of the control program calls-in various subroutines and counts for the real time. At the end, it rewinds the tapes used for various outputs which will be described later. It should be noted that tapes 4 and 5 are used for a particular plotting program which is not documented here.

3.2 SUBROUTINE J02

This subroutine reads-in data from control cards as well as the basic data of depth and sea/land table. The control cards and their formats are described in Appendix A to this report. A small Do-loop (DO 702) is for conversion of depth from fathoms to centimeters in case the depths are read from a navigation chart in fathoms.

The next series of print statements is for printing the input data for checking purposes, especially for checking the digitization of the depth fields. It should be noted that

errors are often made in digitization as well as in the setting of boundaries along the coast. These boundary settings must be carefully checked, and any necessary corrections made prior to running the programs. Following the printing, a Do-loop (DO 900) is used for setting the maximum depth. It often happens that in the area of interest there is a section of deep water, the maximum depth of which is used to compute the required time step.

Do-loops DO 26 and DO 25 are for setting the thickness of the upper layer. In this particular model, a flat mixed layer depth over the whole area is used in the initial state. However, if the topography of the mixed layer depth for a given area is known or is constructed on the basis of available knowledge, it can be read-in with control cards just as depths are read-in. Finally, Subroutine J02 contains the setting of some computational parameters, which includes the zeroing of three fields (AA1, BB2 and CC3). These are used later to compute the transport through predetermined sections.

3.3 SUBROUTINE J03

Subroutine J03 is used for the initialization of the fields. In the beginning, it contains a possibility to read-in Z, U, and V values from a previous computation which might have been interrupted and the values written on the tape. It serves the same purpose as the so-called "checkpoint" feature in CDC machines. The next section is for the initialization of actual depth which is mainly necessary in case the Z, U and V values were read from a previous computation. This subroutine also contains several calls for reading and writing fields into extended core. This must be changed if the program is run on a machine which does not have the extended core. At the end of the subroutine there are Do-loops for zeroing a number of fields and for initial setting of counters. It should be noted that if the Z, U, and V values are continued

from a previous computation, the counters TIC and TIT must be set as they were at the end of previous computation. This depends very much on the set up of the program and will be described in later paragraphs.

3.4 SUBROUTINE VECTOR

This subroutine is called by Subroutine J04. This serves to compute the current direction and speed from computed U and V components. This is done only during the output cycle which can be hourly, or any desired time interval specified on the control card.

It should be noted that in this particular program system, subroutine calls are in FORTRAN II (e.g., ABSF, SQRTF and others). The so-called "F functions" do not exist in many newer FORTRAN compilers. There are two possibilities to overcome this. The first, which offers the greatest savings in machine time, is to recut the cards leaving out the F at the end of the compiler subroutines. The second possibility is to make special function subroutines at the end of the program, as is the case with this program. Furthermore, it should be noted that statement #433 contains (at the end) a number +180. This is the tilt angle of the grid in relation to the north-south orientation. This problem is discussed in the description of the single-layer model and the value should be changed according to the tilt taken in a particular program.

3.5 SUBROUTINE J04

This subroutine is for output of the computed fields. How often it is called in real time is determined by the parameter T1 which is read-in from an input card in Subroutine J02. It should be noted that when a new area with new dimensions is printed out, the dimension statements as well as the Do-loops must be changed correspondingly. First, the water level variations (astronomic tides and meteorological tides, or the water level changes caused by winds) are printed. The

second printed field is the deviation of the mixed layer depth from the mean value which was originally given to this parameter. The subsequent two fields give the direction and speed of the current in the upper and lower layers respectively.

The program contains an option to compute the so-called "rest current" after a full tidal cycle (ca., 24 hours, 50 minutes). The following Do-loop and few additional statements are for the preparation of rest currents for the output (using the same printing statements and formats as in printing the current direction and speed).

The following little section with Do-loops 9 and 1,000 is for the output of data at selected points on the tape which will be printed later with Subroutine S08. The positions and names of the special points from which output is desired is put in by a control card in J02. Thereafter follows a section for output of transport through selected sections (in cubic meters per hour) and zeroing the sections after the output. Finally, Subroutine J04 includes the printing-out of concentration fields (e.g., pollution) if dispersion and diffusion computations are made.

3.6 SUBROUTINE J05

This is the main computation subroutine and is called in at every time step. First it contains the mean Z and Z2L fields (smoothing of the"height" fields). It should be noted that, in statement 22, different smoothers are used for upper and lower layers. A little heavier smoothing in lower layers has been found necessary, whereas the smoothing coefficient for the upper layers is normally 0.98 or less, in the lower layers 0.92 or 0.90 is used. Do-loop 70 is the main Do-loop for the computation of Z and Z2L values at grid points over the sea. After the computation of Z and Z2L fields, the outer open boundaries are set in this particular model, as can be seen from this boundary setting in the program. The open

boundaries presented in the program in Appendix C are essentially closed for in- and out-flow, but they move up and down in the fashion of the so-called "lubricated wall." Do-loop 278 is for the input of tides at the primary open boundary. Preceding this Do-loop, and within the Do-loop itself, a specific adjustment in this model is made to compute spring tides of the area. A time lag of 5,400 seconds is also introduced which was required in a double input of tides in another run of this model but is not documented here. Thereafter follows a specific adjustment of boundaries to allow some permanent flow through the areas as known from empirical measurements. This is a very special input and varies from one model to another, the description of which will be found in describing the results of any output from the model run.

As the new elevation values (here, water level and mixed layer depth changes) are computed, new layer thickness and depth values must be computed for use in various current component computations. This is done in Do-loop 85 which is essentially the same as found in Subroutine J03. Thereafter, Do-loop 760 contains a check of whether some of the points have gone dry or the lower or upper layer have vanished due to sinking of rising of the mixed layer depth.

Before the computation of U and V components, Subroutine S01 is called if the winds are used as input in the model.

The following section contains the computation of the U component. It starts with the computation of $\bar{U}$ and V* and the corresponding values for the lower layers. Before computing the U and U2L in Do-loops 412 and 1050, the boundaries for $\bar{U}$ and V* are set. It should be noted that U and U2L are needed from the previous time step for the computation of V. Thereafter, the boundaries of U and U2L are set, U* and U2L* computed and the new U and U2L values are transferred from the intermediate fields (ZM and ZM2) into the proper fields.

Before the computation of V, $\bar{V}$ and $\overline{V2L}$ are computed, boundaries are set and the V components are computed in Do-loops 428 and 1100.

It can be noted that a special additional inflow is set in every time step in this program at point 26/40, which is the location of The Narrows on the large scale grid. Such additional flows can and should be set if outflow from rivers is known or can be estimated.

After setting the V boundaries, a following section contains the computation of net flow through predetermined sections. In this program the net flow is output hourly during a full tidal cycle. Two approaches which are essentially identical are used in this section for the computation of transport. In one approach (sections AA1 and BB2) the transports are computed by listing the corresponding depth and current values at desired grid points. The other part uses a special transport function for computation. This function is described later in this document. Before coming to the computation of dispersion and diffusion, the rest currents are computed and accumulated in special fields during a full tidal cycle.

The dispersion and diffusion of pollutants is usually computed throughout one tidal cycle. Both the instantaneous source and continuous source can be used and examples of setting the sources are shown in the program. There are several different possible approaches to compute the dispersion and diffusion. If the grid size is small, the diffusion can be computed as indicated in this program. On the other hand, if the grid size is large, the approach used in this program is unsatisfactory. In this case the central position of the instantaneous point source is followed and the shape of the resulting fields is computed in a special program which uses the output of the current fields from this program and computes the shape of the pollution blob. At the very end of

the subroutine, another method of computing diffusion (a simplified Rammig method) is indicated. It gives essentially the same results as the subroutine above.

3.7 THE TRANSPORT FUNCTION

This function is used to compute transport through the section as mentioned earlier. It contains several comments which makes the function easily understandable.

3.8 SUBROUTINE S01

Subroutine S01 is used for computing the surface wind current component. It contains the provision to compute the surface current from surface to bottom (in case of shallow water) or through the upper layer, or the mixed layer depth can be used. In the latter case, the last variable in statements 5 and 7 must be changed to HMLD. This is a very simple but quite satisfactory wind current computation. There are much more complex and, theoretically, more accurate subroutines now available in EPRF which use, among other methods, a prescribed pressure field to obtain the wind.

3.9 SUBROUTINE S08

This subroutine is called at the very end of the program for output of Z, U and V values in both layers at desired special points.

3.10 SUBROUTINE JOB6, JOB7 AND THE "F FUNCTIONS"

These latter two subroutines are used for output on a special plotting tape and could be omitted in case this plotting program is not available. In this case, corresponding calls for the subroutine should be deleted from the control program. The "F function" subroutines are indicated at the very end of the program but could be deleted as described earlier if the few statements containing these in the program are repunched.

REFERENCES

Hamilton, G.D., F.R. Williams and T. Laevastu, 1973: Computation of Tides and Currents With Multilayer Hydrodynamical-Numerical (MHN) Models With Several Open Boundaries. Proc. of the 1973 Summer Computer Simulation Conference, Montreal, Canada, 634-639.

Hansen, W., 1966: The Reproduction of the Motion in the Sea by Means of Hydrodynamical-Numerical Methods. Mitteil. Inst. Meeresk., Hamburg, 5:57 pp. and figures.

Hansen, W., Personal communication (Including unpublished version of the two-layer model development).

Reports in this series:

Part 1: A vertically integrated hydrodynamical-numerical model (W. Hansen type): Model description and operating/running instructions.

Part 2: A multi-layer hydrodynamical-numerical model (W. Hansen type): Model description and operating/running instructions. (This report.)

Part 3: Computation of tides, currents and dispersal of pollutants in Lower Bay and Approaches to New York with fine and medium grid size hydrodynamical-numerical models.

Part 4: Computation of tides, currents, and dispersal of pollutants in the New York Bight from Block Island to Atlantic City with large grid size, single and two-layer hydrodynamical-numerical models.

APPENDIXES

APPENDIX A - INPUT PARAMETERS AND FORMATS

APPENDIX B - LIST OF SYMBOLS

APPENDIX C - PROGRAM NYAREA LISTING

APPENDIX A

INPUT PARAMETERS AND FORMATS

THE DATA INPUT CARDS

Card 1 Format 24I3

JA, NE, ME, IZE, IUE, KKE, NURU, NG, NS, LI

JA	Indicator, JA=1, read initial values; JA=0, z=0, u=0, v=0
NE,ME	Delimeters of entire field (size of the computational grid)
IZE	Number of points on the open boundary where tidal heights are introduced
IUE	Twice the number of wind fields
KKE	Number of wind field characteristics (A(I))'s (=(IUE/2)*3)
NURU	Number of selected special points
NG	Number of points at the second open input boundary
NS	(not used)
LI	Parameter for computation of dispersion and diffusion of pollutants: LI=0 - no computation; LI=1 - computation to be performed

Card 2 Format 9F8.3

AFGN, G, ALPHA, RBETA, C1

AFGN	Problem number
G	Acceleration of gravity (e.g., 978.0 cm sec^2)
ALPHA	Smoothing parameter (e.g., 0.998)
RBETA	Coefficient of geostrophic wind (e.g., 0.65)
C1	Wind indicator: 0 - no wind; 1 or 2 - wind

Card 3 Format 9F8.0

DT, TE, TW, T1, T2, SI, T, (T3)

DT	1/2 time step (sec) (e.g., 30)
TE	Length of computation (sec) (e.g., 180000)
TW	Time when wind starts (sec) (e.g., 7200)
T1	Time interval between printouts (e.g., 3600 sec)
T2	Field output counter (0 if outputs desired from the start of the computation; otherwise, any other delayed starting time - e.g., 7200 sec)
SI	Time when wind stops (sec) (e.g., 180000)
T	Time (initialized 0, if previous computations were made and s, u and v read from tape, give the time previous computation ended)
(T3)	If plot desired, time when fields are required for plotting

Card 4 Format 6E12.4

DL, F, SIGMA, R, ROL, C

DL	1/2 space step (half grid size in cm - e.g., 185000)
F	Coriolis parameter (e.g., 8.55×10^{-5})
SIGMA	Angular velocity of M_2 tide (1.4088×10^{-4})
R	Friction coefficient (0.003)
ROL	Density of the air (e.g., 1.1627×10^{-3})
C	Drag coefficient (e.g., 3.2×10^{-6})

Card 5 Format 24I3 (one or more cards)

NZ, MZ

NZ	N coordinate
MZ	M coordinate

First the coordinates of the points at the first open boundary (input boundary) and therafter the coordinates of special selected points where special outputs are desired.

Card 6	Format 24I3
NA, MA	N and M coordinates at second input boundary
Card 7	Format 24I3
NU, MU	
NU	N coordinate
MU	M coordinate

Coordinates of the upper right and lower left corners of the wind field

Card 8	Format 10A5
V(I)	Names of the selected special points
Card 9	Format 9F8.2
A(I)	
A(1)	Time when wind starts (sec)
A(2)	Wind speed (m sec^{-1})
A(3)	Wind direction (in degrees in computation coordinates)
Card 10	Format 12F6.0 (usually more than 10 cards)
HTZ	Symbolic depths at the z points
Card 11	Format 12F6.0 (usually more than 10 cards
HTU	Depths at u points
Card 12	Format 12F6.0 (usually more than 10 cards)
HTV	Depths at v points

Cards 13 and 14

Input cards for plotting parameters

APPENDIX B

LIST OF SYMBOLS

A	Characteristics of wind field (A1 - time when wind starts (sec); A2 - wind speed (m sec^{-1}); A3 - wind direction (degrees, counted clockwise from 0° grid coordinate)
AA1	A array for computation of transport through given section (BB2, CC3, etc. serve the same purpose)
AFGN	Program number
ALPHA	Smoothing parameter (0.96 to 0.998)
ALP	Smoothing parameter for second layer (Z2L) (0.88 to 0.90)
ANG ANG2	Compass direction of the current (upper and lower layers)
ARG	Argument of tidal component, (Cos ($\alpha t - \kappa$)) (primary input boundary)
AUS	Austausch coefficient
A1	2 x DT
Ab	F x A1
A3	R x A1
A4	DT/DL
A5	G x A4
B	Intermediate in tidal input
BB2	See AA1
BETA	(1 - ALPHA)/4
BET	(1 - ALP)/4
B4	2 * DL

CON	Argument of tidal component, (Cos ($\alpha t - \kappa$)) (secondary input boundary)
C1	Wind indicator; 0 = no wind; 1 = wind
DIR	Compass direction of rest current
DL	Half grid size (cm)
DL2	Same as B4, used in diffusion computation
DT	Half time step
F	Coriolis parameter
G	Acceleration of gravity
GRZ	A3 x $\sqrt{ZM^2 + ZST^2}$
HGU	Thickness of upper layer at u points
HGU2	Depth of lower layer (thickness of lower layer) at u points
HGV	Thickness of upper layer at v points
HGV2	Depth of lower layer, at v points
HMLD	Mixed layer depth (not used in this program)
HTL	Initial thickness of upper layer (cm)
HTU	Initial depth at u points (0 = land; -1 = boundary)
HTU2	Water depth (initial) in lower layer, at u points
HTV	Initial depth at v points (0 = land; -1 = boundary)
HTV2	Water depth (initial) in lower layer at v points
HTZ	Sea-land table (0 = land; -1 = boundary; 1 = sea)
I	Index
IUE	Twice the number of wind fields
IZE	Number of boundary points at the primary input boundary

J	Index
JA	Input indicator
LEN	A counter
M	Field index in x direction
MA	M coordinate of tidal input at the secondary boundary
ME	Total width of the grid (x direction)
MEH	ME - 1
MU	M coordinate of the wind field delimeter
MZ	M coordinate of primary open boundary (and M coordinate of special point)
N	Field index in y direction
NA	N coordinate of tidal input at the secondary boundary
NE	Total length of the grid (y direction)
NEH	NE - 1
NG	Number of points at secondary input boundary
NURU	Number of special points
NU	N coordinate of the wind field delimeter
NZ	N coordinate of primary open boundary (and N coordinate at special point)
PZ	Intermediate parameter in adjustment of boundaries
RAD	Resultant speed for upper layer
RAD2	Resultant speed for lower layer
RB	Intermediate parameter in tidal input
RBETA	Coefficient of geostrophic wind
REST	Resultant speed for rest current (after a full tidal cycle)

RHO1	Density for top layer
RHO2	Density for lower layer
ROL	Density of the air (not used)
S	Concentration field (e.g., pollutant)
SI	End of wind input (sec)
SIGMA	Angular speed of M_2 tide (not used)
SSH SSV	Intermediate for computing advection of pollutants
T	Time (counter) (sec)
T1	Output interval (sec)
T2	Initial time for first output (sec)
TIC	Index (used in summation of rest current)
TIT	A counter
TRB	Start of rest current output or pollution input
TRE	End of rest current output or pollution input
TW	Time when wind starts (sec)
U	U component of velocity for upper layer
US	Accumulator of u component for computation of advection of pollutant
USR	Accumulator for computation of rest currents
U1 U2	Amplitudes of tidal components at primary boundary
U2L	U component of velocity for lower layer
U3 U4	Amplitudes of tidal components at secondary boundary
V	V component of speed in upper layer
VS	Accumulator of v component for computation of advection of pollutant

VSR	Accumulator for computation of rest current
V1	Names of special points
V2	N and M components of special points
V2L	V component of speed in lower layer
V3 V4	Speed and direction in upper layer at special points
V5	Mixed layer depth at special points (thickness of upper layer)
V6 V7	Speed and direction in lower layer at special points
WERTL WERTL2 WERTO WERTOL	Intermediates for smoothing and averaging
WERTOL2 WERTOR WERTOR2 WERTR WERTR2 WERTU WERTUL WERTUL2 WERTUR WERTUR2 WERTU2	Intermediates for smoothing and averaging
WURTZEL	$\sqrt{ZM^2 + ZST^2}$
X	Intermediate tidal parameter at primary open boundary
XK	x component of wind current

Y	Intermediate tidal parameter at secondary open boundary
YK	y component of wind current
Z	Water level (elevation) (cm)
ZM	Intermediate for smoothing (upper layer)
ZM2	Intermediate for smoothing (lower layer)
ZST	Intermediate for averaging (upper layer)
ZST2	Intermediate for averaging (lower layer)
Z1 Z2	Amplitudes of tidal components at primary boundary
Z2L	Deviation of mixed layer depth from mean
Z3 Z4	Amplitudes of tidal components at secondary boundary

APPENDIX C

PROGRAM NYAREA LISTING

```
      PROGRAM NYAREA(INPUT ,OUTPUT,TAPE2,TAPE4,TAPE3=INPUT ,TAPE5,TAPE6)
C     TWO LAYER MODEL,H M,
      DIMENSION Z(42,47),U(42,47),V(42,47),ZM(42,47),ZST(42,47),HGU(42,4
     17),HGV(42,47),HTZ(42,47),HTU(42,47),HTV(42,47),XK(42,47),YK(42,47)
     2,A(12),NZ(90),MZ(90),NU(10),MU(10),Z1(80),Z2(80),V1(20),V2(20),U1(
     320),U2(20),X(8),ARG(8),RAD(42,47),ANG(42,47),V3(20),V4(20),
     4HTL(42,47),NA(60),MA(60),Y(8),Z3(80),Z4(80),U3(80),U4(80),CON(8)
     5,V5(20),V6(20),V7(20),US(42,47),VS(42,47),S(42,47)
     6,ANG2(42,47),RAD2(42,47)
     7,USR(42,47),VSR(42,47)
      EQUIVALENCE(ZM,RAD),(ZST,ANG),(XK,Z2L),(HGU,RAD2),(YK,U2L),
     1(HGV,ANG2),(ZM2,HTV),(ZST2,HTU)
      COMMON E,U,V,ZM,ZST,HGU,HGV,HTZ,HTU,HTV,XK,YK,A,NZ,MZ,NU,MU,Z1,Z2,
     1V1,V2,U1,U2,X,ARG,V3,V4,HTL,NA,MA,Y,Z3,Z4,U3,U4,CON
     2,V5,V6,V7,US,VS,S,USR,VSR
     3,F,G,SIGMA,ALPHA,R,ROL,RBETA,C,DL,DT,T,T1,T2,NE,ME,IE,KKE,IZE,IUE,
     4TW,BETA,A1,A2,A3,A4,A5,C1,C3,S1,NURU,NURV,JA,NEH,MEH,NEHH,MEHH,POC
     5,NG,T3,STV,ETV,TINCV,STH,ETH,TINCH,TRAN,T4,ARGN,TIC,LEN,NS
     6,TRB,TRE
     7,AA2,AA3,AA4,BB3,BB4,BB5,BB6,BB7,BB8,CC4,CC5,CC6,CC7,CC8
      DIMENSION Z2L(42,47),U2L(42,47),V2L(42,47),ZM2(42,47),ZST2(42,47),
     1HGU2(42,47),HGV2(42,47),HTU2(42,47),HTV2(42,47),NU(50),MD(50)
      COMMON/LAYER2/V2L,HGU2,HGV2,HTU2,HTV2,ND,MD,AA1,BB2,CC3
C     ECS MAP
C         ZM          1
C         ZST      1975
C         XK       3949
C         YK       5923
C         HGU      7897
C         HGV      9871
C         ZM2     11845
C         ZST2    13819
C         RAD     15793
C         ANG     17767
C         RAD2    23689
C         ANG2    25663
C         Z2L     31585
C         U2L     33559
C         HTV     35533
C         HTU     37507
C
      REWIND 2
      CALL J02
      CALL J03
      CALL J0B6
      IF(T)6,5,1
    1 CALL J04
```

Extended core map for a particular model.

```
    6 CALL JOB7
    8 CONTINUE
      T2=T2+T1
    2 T=T+A1
      CALL JO5
      IF(T-TE)3,5,5
    3 IF(ABSF(T-T2)-A1/2.)1,1,4
    4 GO TO 2
    5 LEN=5
      LOP=5
      CALL JO4
      CALL JO7
      END FILE 2
      REWIND 2
      END FILE 6
      REWIND 6
      CALL SO8
      END FILE 4
      REWIND 4
      END FILE 5
      REWIND 5
      STOP
      END
      SUBROUTINE JO2
      DIMENSION Z(42,47),U(42,47),V(42,47),ZM(42,47),ZST(42,47),HGU(42,4
     17),HGV(42,47),HTZ(42,47),HTU(42,47),HTV(42,47),XK(42,47),YK(42,47)
     2,A(12),NZ(90),MZ(90),NU(10),MU(10),Z1(80),Z2(80),V1(20),V2(20),U1(
     320),U2(20),X(8),ARG(8),RAD(42,47),ANG(42,47),V3(20),V4(20),
     4HTL(42,47),NA(60),MA(60),Y(8),Z3(80),Z4(80),U3(80),U4(80),CON(8)
     5,V5(20),V6(20),V7(20),US(42,47),VS(42,47),S(42,47)
     6,ANG2(42,47),RAD2(42,47)
     7,USR(42,47),VSR(42,47)
      EQUIVALENCE(ZM,RAD),(ZST,ANG),(XK,Z2L),(HGU,RAD2),(YK,U2L),
     1(HGV,ANG2),(ZM2,HTV),(ZST2,HTU)
      COMMON Z,U,V,ZM,ZST,HGU,HGV,HTZ,HTU,HTV,XK,YK,A,NZ,MZ,NU,MU,Z1,Z2,
     1V1,V2,U1,U2,X,ARG,V3,V4,HTL,NA,MA,Y,Z3,Z4,U3,U4,CON
     2,V5,V6,V7,US,VS,S,USR,VSR
     3,F,G,SIGMA,ALPHA,R,ROL,RBETA,C,DL,DT,I,T1,T2,NE,ME,IE,KKE,IZE,IUE,
     4TW,BETA,A1,A2,A3,A4,A5,C1,C3,SI,NURU,NURV,JA,NEH,MEH,NEHH,MEHH,POC
     5,NG,T3,STV,ETV,TINCV,STH,ETH,TINCH,TRAN,T4,AFGN,TIC,LEN,NS
     6,TRB,TRE
     7,AA2,AA3,AA4,BB3,BB4,BB5,BB6,BB7,BB8,CC4,CC5,CC6,CC7,CC8
      DIMENSION Z2L(42,47),U2L(42,47),V2L(42,47),ZM2(42,47),ZST2(42,47),
     1HGU2(42,47),HGV2(42,47),HTU2(42,47),HTV2(42,47),NU(50),MD(50)
      COMMON/LAYER2/V2L,HGU2,HGV2,HTU2,HTV2,ND,MD,AA1,BB2,CC3
      DATA HTL1/2500./
  100 FORMAT(6F10.2)
  101 FORMAT(10A5)
  102 FORMAT(24I3)
  103 FORMAT(9F8.3)
  104 FORMAT(9F8.0)
  105 FORMAT(6E12.4)
  106 FORMAT(12F6.0)
```

Control program. JOB6 and JOB7 are for a particular plotting program. Tapes 4 and 5 are for the same plotting program.

```
107 FORMAT(12F6.1)
112 FORMAT(5X,19HHALF TIME STEP DT=,F5.0,5H SEC,29X,20HHALF SPACE STEP
   1 DL=,E11.4,4H CM///)
113 FORMAT(5X,27HSMOOTHING PARAMETER ALPHA=,F6.3,25X,23HFRICTION COEFF
   1ICENT R=,E11.4///)
114 FORMAT(5X,22HCORIOLIS PARAMETER F=,E11.4,7H  1/SEC,10X,39HANGULAR
   1VELOCITY OF THE M2 TIDE SIGMA=,E11.4,7H 1/SEC/////)
115 FORMAT(5X,13HWIND IS CALM,/////)
116 FORMAT(5X,21HTHE WIND BEGINS AFTER,F7.0,6H  SEC,,19X,13HAIR DENSIT
   1Y=,E11.4,9H  G/CM**3///)
117 FORMAT(5X,23HREDUCTION FACTOR BETA=,F6.3,26X,25HDRAG COEFFICIENT L
   1AMBDA=,E11.4,11H (CONSTANT)/////)
120 FORMAT(5X,54H(N,M) COORDINATES OF THE Z POINTS ON THE OPEN BOUNDAR
   1Y//5X,12(1H(,I2,1H,,I2,1H),2X)//5X,12(1H(,I2,1H,,I2,1H),2X)//)
   2//5X,12(1H(,I2,1H,,I2,1H),2X)//5X,12(1H(,I2,1H,,I2,1H),2X)
   3//5X,9(1H(,I2,1H,,I2,1H),2X)
121 FORMAT(1H1,14X,36HSYMBOLIC WATER DEPTH AT THE Z POINTS///8X,14I8/)
122 FORMAT(///8X,4I8//)
 23 FORMAT(1H1,14X,33HWATER DEPTH AT THE U2 POINTS (CM)///8X,14I8/)
123 FORMAT(1H1,14X,33HWATER DEPTH AT THE U1 POINTS (CM)///8X,14I8/)
 24 FORMAT(1H1,14X,33HWATER DEPTH AT THE V2 POINTS (CM)///8X,14I8/)
124 FORMAT(1H1,14X,33HWATER DEPTH AT THE V1 POINTS (CM)///8X,14I8/)
130 FORMAT(I8,14F8.0)
133 FORMAT(1X,45HWIND INPUT(STARTING TIME,MAGNITUDE,DIRECTION),/1X,
   16F8.2//)
134 FORMAT(I8,4F8.0)
    READ (3,102)JA,NE,ME,IZE,IUE,KKE,NURU,NG,NS,LI
    READ (3,103)AFGN,G,ALPHA,RBETA,C1
    READ (3,104)DT,TE,TW,T1,T2,SI,I
   6,T3
    READ (3,105)DL,F,SIGMA,R,ROL,C
    IF(IZE)2,2,1
  1 READ (3,102) (NZ(I),MZ(I),I=1,IZE)
  2 READ (3,102) (NZ(I+39),MZ(I+39),I=1,NURU)
    READ (3,102) (NA(I),MA(I),I=1,NG)
    READ (3,102) (NU(I),MU(I),I=1,IUE)
    READ (3,101)(V1(I),I=1,NURU)
    READ (3,100) (A(I),I=1,KKE)
    READ (3,106) ((HTZ(N,M),M=1,ME),N=1,NE)
    READ (3,106) ((HTU(N,M),M=1,ME),N=1,NE)
    READ (3,106) ((HTV(N,M),M=1,ME),N=1,NE)
    DO 9967 N=1,41
    HTZ(N,1)=0.
    HTZ(N,2)=-2.
9967 CONTINUE
 700 DO 702 N=1,NE
     DO 702 M=1,ME
     IF(HTU(N,M))701,701,703
 703 HTU(N,M)=HTU(N,M)*185.201
 701 IF(HTV(N,M))702,702,704
 704 HTV(N,M)=HTV(N,M)*185.201
 702 CONTINUE
```

Reading of input and control cards

Special input boundary setting.

Conversion of depths in fathoms to cm.

```
      PRINT 112,DT,DL
      PRINT 113,ALPHA,R
      PRINT 114,F,SIGMA
      IF(C1)4,3,4
    3 PRINT 115
      GO TO 5
    4 PRINT 116,TW,ROL
      PRINT 117,RBETA,C
      PRINT 133,(A(I),I=1,KKE)
    5 PRINT 120,(NZ(I),MZ(I),I=1,IZE)
    8 PRINT 121,(N,N=1,14)
      PRINT 130,(N,(HTZ(N,M),M=1,14),N=1,NE)
      PRINT 222,(N,N=15,28)
      PRINT 234,(N,(HTZ(N,M),M=15,28),N=1,NE)
      PRINT 222,(N,N=29,42)
      PRINT 234,(N,(HTZ(N,M),M=29,42),N=1,NE)
      PRINT 223,(N,N=43,47)
      PRINT 235,(N,(HTZ(N,M),M=43,47),N=1,NE)
  222 FORMAT(///8X,14I8//)
  223 FORMAT(///8X,5I8//)
  234 FORMAT(I8,14F8.0)
  235 FORMAT(I8,5F8.0)
      PRINT 153,(N,N=1,14)
      PRINT 130,(N,(HTU(N,M),M=1,14),N=1,NE)
      PRINT 652,(N,N=15,28)
      PRINT 656,(N,(HTU(N,M),M=15,28),N=1,NE)
      PRINT 612,(N,N=29,42)
      PRINT 656,(N,(HTU(N,M),M=29,42),N=1,NE)
      PRINT 660,(N,N=43,47)
      PRINT 661,(N,(HTU(N,M),M=43,47),N=1,NE)
      PRINT 124,(N,N=1,14)
      PRINT 130,(N,(HTV(N,M),M=1,14),N=1,NE)
      PRINT 652,(N,N=15,28)
      PRINT 656,(N,(HTV(N,M),M=15,28),N=1,NE)
      PRINT 622,(N,N=29,42)
      PRINT 656,(N,(HTV(N,M),M=29,42),N=1,NE)
      PRINT 660,(N,N=43,47)
      PRINT 661,(N,(HTV(N,M),M=43,47),N=1,NE)
      DO 900 N=1,NE
      DO 900 M=1,ME
      IF(HTU(N,M).GT.150000)901,902
  901 HTU(N,M)=150000.
  902 IF(HTV(N,M).GT.150000)903,900
  903 HTV(N,M)=150000.
  900 CONTINUE
      DO 26 N=1,NE
      DO 26 M=1,ME
      HTL(N,M)=HTL1
   26 CONTINUE
      DO 25 N=1,NE
      DO 25 M=1,ME
      IF (HTU(N,M)) 10,12,14
```

Printing of input data for checking.

Maximum depth setting.

```
 10 HTU2(N,M) = -1.0
    GO TO 15
 12 HTU2(N,M) = 0.0
    GO TO 15
 14 HTU2(N,M) = HTU(N,M) - HTL(N,M)
    IF(HTU2(N,M))201,201,15
201 HTU2(N,M)=0.
 15 IF(HTV(N,M)) 17,19,21
 17 HTV2(N,M) = -1.0
    GO TO 25
 19 HTV2(N,M) = 0.0
    GO TO 25
 21 HTV2(N,M) = HTV(N,M) - HTL(N,M)
    IF(HTV2(N,M))202,202,25
202 HTV2(N,M)=0.
 25 CONTINUE
652 FORMAT(///5X,14I8/)
656 FORMAT(I5,14F8.0)
660 FORMAT(I5,5I8/)
661 FORMAT(I5,5F8.0)
    BETA=(1.-ALPHA)/4.
    LOP=1
    T4=T1
    A1=2.*DT
    A2=F*A1
    A3=R*A1
    A4=DT/DL
    A5=G*A4
    C3=C*A1*10000.
    AA1=0.
    BB2=0.
    CC3=0.
    NEH=NE-1
    MEH=ME-1
    NEHH=NE-2
    MEHH=ME-2
    RETURN
    END
    SUBROUTINE JOS
    DIMENSION Z(42,47),U(42,47),V(42,47),ZM(42,47),ZST(42,47),HGU(42,4
   17),HGV(42,47),HTZ(42,47),HTU(42,47),HTV(42,47),XK(42,47),YK(42,47)
   2,A(12),NZ(90),MZ(90),NU(10),MU(10),Z1(80),Z2(80),V1(20),V2(20),U1(
   320),U2(20),X(8),ARG(8),RAD(42,47),ANG(42,47),V3(20),V4(20),
   4HTL(42,47),NA(60),MA(60),Y(8),Z3(80),Z4(80),U3(80),U4(80),CON(8)
   5,V5(20),V6(20),V7(20),US(42,47),VS(42,47),S(42,47)
   6,ANG2(42,47),RAD2(42,47)
   7,USR(42,47),VSR(42,47)
    EQUIVALENCE(ZM,RAD),(ZST,ANG),(XK,Z2L),(HGU,RAD2),(YK,U2L),
   1(HGV,ANG2),(ZM2,HTV),(ZST2,HTU)
    COMMON Z,U,V,ZM,ZST,HGU,HGV,HTZ,HTU,HTV,XK,YK,A,NZ,MZ,NU,MU,Z1,Z2,
   1V1,V2,U1,U2,X,ARG,V3,V4,HTL,NA,MA,Y,Z3,Z4,U3,U4,CON
   2,V5,V6,V7,US,VS,S,USR,VSR
   3,F,G,SIGMA,ALPHA,R,ROL,RBETA,C,DL,DT,T,T1,T2,NE,ME,IE,KKE,IZE,IUE,
```

Setting of the thickness of layers at U and V points.

Setting of some computational parameters.

```
     4TW,BETA,A1,A2,A3,A4,A5,C1,C3,S1,NURU,NURV,JA,NEH,MEH,NEHH,MEHH,POC
     5,NG,T3,STV,ETV,TINCV,STH,ETH,TINCH,TRAN,T4,AFGN,TIC,LEN,NS
     6,TRB,TRE
     7,AA2,AA3,AA4,BB3,BB4,BB5,BB6,BB7,BB8,CC4,CC5,CC6,CC7,CC8
      DIMENSION Z2L(42,47),U2L(42,47),V2L(42,47),ZM2(42,47),ZST2(42,47),
     1HGU2(42,47),HGV2(42,47),HTU2(42,47),HTV2(42,47),ND(50),MD(50)
      COMMON/LAYER2/V2L,HGU2,HGV2,HTU2,HTV2,ND,MD,AA1,BB2,CC3
140   FORMAT(18F4.1)
 100  FORMAT(9F8.3)
      IF(JA)2,7,2
   2  READ (3,100) ((Z(N,M),M=1,ME),N=1,NE)
      READ (3,100) ((U(N,M),M=1,ME),N=1,NE)
      READ (3,100) ((V(N,M),M=1,ME),N=1,NE)
      GO TO 11
   7  CONTINUE
      IF(JA)11,410,11
 410  DO 12 N=1,NE
      DO 12 M=1,ME
      IF(HTZ(N,M))201,200,201
 200  Z(N,M)=0.
      Z2L(N,M) = 0.0
      GO TO 202
 201  Z(N,M)=0.2
      Z2L(N,M) = 0.2
 202  IF(HTU(N,M))203,203,204
 203  U(N,M)=0.
      U2L(N,M) = 0.0
      GO TO 205
 204  U(N,M)=0.2
      U2L(N,M) = 0.2
 205  IF(HTV(N,M))206,206,207
 206  V(N,M)=0.
      V2L(N,M) = 0.0
      GO TO 12
 207  V(N,M)=0.2
      V2L(N,M) = 0.2
  12  CONTINUE
      CALL WRITEC(HTU,37507,1974)
      CALL READEC(ZST2,13819,1974)
      CALL WRITEC(HTV,35533,1974)
      CALL READEC(ZM2,11845,1974)
      DO 208 N=1,NE
      DO 208 M=1,ME
      IF(HTZ(N,M))209,209,210
 209  ZM(N,M)=0.
      ZM2(N,M) = 0.0
      ZST(N,M)=0.
      ZST2(N,M) = 0.0
      GO TO 208
 210  ZM(N,M)=0.2
      ZM2(N,M) = 0.2
      ZST(N,M)=0.2
      ZST2(N,M) = 0.2
```

Reading of Z, U and V values from previous computation.

Initialization of fields (0 or a small value over the sea).

```
208 CONTINUE
 11 CONTINUE
800 CALL WRITEC(ZM2,11845,1974)
    CALL READEC(HTV,35533,1974)
    CALL WRITEC(ZST2,13819,1974)
    CALL READEC(HTU,37507,1974)
    CALL READEC(HGU,7892,1974)
    CALL READEC(HGV,9871,1974)
 15 DO 14 N=1,NEH
    DO 14 M=1,MEH
    IF(HTU(N,M))5,5,4
  5 HGU(N,M)=0.
    GO TO 501
  4 HGU(N,M)=HTL(N,M)+(Z(N,M)+Z(N,M+1))/2.
501 IF(HTU2(N,M))503,503,502
502 HGU2(N,M) = HTU2(N,M) + (Z2L(N,M) + Z2L(N,M+1))/2.
    GO TO 13
503 HGU2(N,M) = 0.0
 13 IF(HTV(N,M))8,8,6
  8 HGV(N,M)=0.
    GO TO 505
  6 HGV(N,M)=HTL(N,M)+(Z(N,M)+Z(N+1,M))/2.
505 IF(HTV2(N,M))507,507,506
506 HGV2(N,M) = HTV2(N,M) + (Z2L(N,M) + Z2L(N+1,M))/2.
    GO TO 14
507 HGV2(N,M) = 0.0
 14 CONTINUE
```

Initialization of actual depth.

```
    CALL WRITEC(HGU,7897,1974)
    CALL WRITEC(HGV,9871,1974)
    CALL WRITEC(U2L,33559,1974)
    CALL WRITEC(Z2L,31585,1974)
    CALL WRITEC(HTU,37507,1974)
    CALL WRITEC(HTV,35533,1974)
    CALL READEC(XK,3949,1974)
    CALL READEC(YK,5923,1974)
    CALL READEC(ANG2,25663,1974)
    CALL READEC(RAD2,23689,1974)
    DO 10 N=1,NE
    DO 10 M=1,ME
    XK(N,M)=0.
    YK(N,M)=0.
    USR(N,M)=0.
    VSR(N,M)=0.
    S(N,M) = 0
    US(N,M) = 0.
    VS(N,M) = 0.
    ANG2(N,M) = 0
    RAD2(N,M) = 0.
 10 CONTINUE
    AA2 = 0.
    AA3 = 0.
    AA4 = 0.
    BB3 = 0.
    BB4 = 0.
```

Zeroing of fields and arrays.

```
      BB5 = 0.
      BB6 = 0.
      BB7 = 0.
      BB8 = 0.
      CC4 = 0.
      CC5 = 0.
      CC6 = 0.
      CC7 = 0.
      CC8 = 0.
      CALL WRITEC(ANG2,25663,1974)
      CALL WRITEC(RAD2,23689,1974)
      CALL WRITEC(XK,3949,1974)
      CALL WRITEC(YK,5923,1974)
      CALL READEC(Z2L,31585,1974)
      CALL READEC(U2L,33559,1974)
      TIC=0.
      TIT=0.
      LEN=0
      RETURN
      END
      SUBROUTINE VECTOR(HTZ,U,V,RAD,ANG,NE,ME)
      DIMENSION HTZ(42,47),U(42,47),V(42,47),RAD(42,47),ANG(42,47)
      DO 432 N=1,NE
      DO 432 M=1,ME
      IF(HTZ(N,M))509,101,509
  509 IF(V(N,M)**2)100,141,100
  141 IF(U(N,M)**2)100,101,100
  100 RAD(N,M)=SQRTF(ABSF(V(N,M))**2+ABSF(U(N,M))**2)
      GO TO 102
  101 RAD(N,M)=0.
      ANG(N,M)=999.
      GO TO 432
  102 ANG(N,M)=ABSF(ASINF(V(N,M)/RAD(N,M)))
      ANG(N,M)=ANG(N,M)*(360./6.28318530/3)
      IF(V(N,M))143,142,142
  142 IF(U(N,M))500,433,433
  143 IF(U(N,M))501,501,144
  144 GO TO 502
  500 ANG(N,M)=180.-ANG(N,M)
      GO TO 433
  501 ANG(N,M)=180.+ANG(N,M)
      GO TO 433
  502 ANG(N,M)=360.-ANG(N,M)
  433 ANG(N,M)=(90.-ANG(N,M))+180.
  503 IF(ANG(N,M))145,432,432
  145 ANG(N,M)=360.+ANG(N,M)
      GO TO 503
  432 CONTINUE
      RETURN
      END
      SUBROUTINE J04
      DIMENSION Z(42,47),U(42,47),V(42,47),ZM(42,47),ZSI(42,47),HGU(42,4
```

Setting some counters.

Subroutine for computation of current direction and speed from U and V components.

```
     17),HGV(42,47),HTZ(42,47),HTU(42,47),HTV(42,47),XK(42,47),YK(42,47)
     2,A(12),NZ(90),MZ(90),NU(10),MU(10),Z1(80),Z2(80),V1(20),V2(20),U1(
     320),U2(20),X(8),ARG(8),RAD(42,47),ANG(42,47),V3(20),V4(20),
     4HTL(42,47),NA(60),MA(60),Y(8),Z3(80),Z4(80),U3(80),U4(80),CON(8)
     5,V5(20),V6(20),V7(20),US(42,47),VS(42,47),S(42,47)
     6,ANG2(42,47),RAD2(42,47)
     7,USR(42,47),VSR(42,47)
      EQUIVALENCE(ZM,RAD),(ZST,ANG),(XK,Z2L),(HGU,RAD2),(YK,U2L),
     1(HGV,ANG2),(ZM2,HTV),(ZST2,HTU)
      COMMON Z,U,V,ZM,ZST,HGU,HGV,HTZ,HTU,HTV,XK,YK,A,NZ,MZ,NU,MU,Z1,Z2,
     1V1,V2,U1,U2,X,ARG,V3,V4,HTL,NA,MA,Y,Z3,Z4,U3,U4,CON
     2,V5,V6,V7,US,VS,S,USR,VSR
     3,F,G,SIGMA,ALPHA,R,ROL,RBETA,C,DL,DT,T,T1,T2,NE,M=,IE,KKE,IZE,IUE,
     4TW,BETA,A1,A2,A3,A4,A5,C1,C3,S1,NURU,NURV,JA,NEH,MEH,NEHH,MEHH,POC
     5,NG,T3,STV,ETV,TINCV,STH,ETH,TINCH,TRAN,T4,AFGN,TIC,LEN,NS
     6,TRB,TRI
     7,AA2,AA3,AA4,BB3,BB4,BB5,BB6,BB7,BB8,CC4,CC5,CC6,CC7,CC8
      DIMENSI N Z2L(42,47),U2L(42,47),V2L(42,47),ZM2(42,47),ZST2(42,47),
     1HGU2(42,47),HGV2(42,47),HTU2(42,47),HTV2(42,47),NU(50),MD(50)
      COMMON/LAYER2/V2L,HGU2,HGV2,HTU2,HTV2,ND,MD,AA1,BB2,CC3
      CALL WRITEC(ZM,1,1974)
      CALL WRITEC(ZST,1975,1974)
      CALL WRITEC(HGU,7897,1974)
      CALL WRITEC(HGV,9871,1974)
      CALL READEC(RAD,15793,1974)
      CALL RE DEC(ANG,17767,1974)
      CALL READEC(RAD2,23689,1974)
      CALL RE DEC(ANG2,25663,1974)
CJ04  WRI FIE
      PRINT 499,T,(N,N=1,14)
      PRINT 500,(N,(Z(N,M),M=1,14),N=1,NE)
      PRINT 502,(N,N=15,28)
      PRINT 500,(N,(Z(N,M),M=15,28),N=1,NE)
      PRINT 502,(N,N=29,42)
      PRINT 500,(N,(Z(N,M),M=29,42),N=1,NE)
      PRINT 501,(N,N=43,47)
      PRINT 500,(N,(Z(N,M),M=29,42),N=1,NE)
      PRINT 503,(N,(Z(N,M),M=43,47),N=1,NE)
      PRINT 500,(N,(Z(N,M),M=29,42),N=1,NE)
  494 FORMAT(1H1,14X,64HRESULTANT CURRENT SPEED (CM/SEC) AND DIRECTION
     1(DEG.)  AFTER T=F9.0,/5X,I7,13I9/)
  495 FORMAT(//5X,I7,4I9/)
  496 FORMAT((I5,14(F4.0,1H,,F4.0)/))
  497 FORMAT((I5,5(F4.0,1H,,F4.0)/))
  498 FORMAT(//5X,I7,13I9/)
  499 FORMAT(1H1,14X,25HWATER LEVEL (CM) AFTER T=,F9.0/5X,I7,13I8)
  500 FORMAT(I5,14F8.0/)
  501 FORMAT(//5X,I7,4I8/)
  502 FORMAT(//5X,I7,13I8/)
  503 FORMAT(I5,5F8.0/)
      PRINT 559,T,(N,N=1,14)
      PRINT 500,(N,(Z2L(N,M),M=1,14),N=1,NE)
```

Printing of Z values

```
      PRINT 502,(N,N=15,28)
      PRINT 500,(N,(Z2L(N,M),M=15,28),N=1,NE)
      PRINT 502,(N,N=29,42)
      PRINT 500,(N,(Z2L(N,M),M=29,42),N=1,NE)
      PRINT 501,(N,N=43,47)
      PRINT 503,(N,(Z2L(N,M),M=43,47),N=1,NE)
```

Printing of Z2L values.

```
  554 FORMAT(1H1,14X,64HDEEP LAYER CURRENT, SPEED (CM/SEC) AND DIRECTION
     1(DEG.)  AFTER T=,F9.0,/5X,I7,13I9/)
  559 FORMAT(1H1,14X,23HMLD DEPTH CM  AFTER T=,F9.2/5X,I7,13I8)
 1634 CALL VECTOR(HTZ,U,V,RAD,ANG,NE,ME)
      PRINT 494,T,(N,N=1,14)
      PRINT 496,(N,(RAD(N,M),ANG(N,M),M=1,14),N=1,NE)
      PRINT 498,(N,N=15,28)
      PRINT 496,(N,(RAD(N,M),ANG(N,M),M=15,28),N=1,NE)
      PRINT 498,(N,N=29,42)
      PRINT 496,(N,(RAD(N,M),ANG(N,M),M=29,42),N=1,NE)
      PRINT 495,(N,N=43,47)
      PRINT 497,(N,(RAD(N,M),ANG(N,M),M=43,47),N=1,NE)
      IF(LOP-10)1635,10,1635
 1635 CALL VECTOR(HTZ,U2L,V2L,RAD2,ANG2,NE,ME)
      PRINT 554,T,(N,N=1,14)
      PRINT 496,(N,(RAD2(N,M),ANG2(N,M),M=1,14),N=1,NE)
      PRINT 498,(N,N=15,28)
      PRINT 496,(N,(RAD2(N,M),ANG2(N,M),M=15,28),N=1,NE)
      PRINT 498,(N,N=29,42)
      PRINT 496,(N,(RAD2(N,M),ANG2(N,M),M=29,42),N=1,NE)
      PRINT 495,(N,N=43,47)
      PRINT 497,(N,(RAD2(N,M),ANG2(N,M),M=43,47),N=1,NE)
```

Printing of directions and speed of upper and lower layers.

```
      IF(LOP-10)1630,10,1630
 1630 IF(LOP-5)10,1631,10
 1631 DO 1633 N=1,NE
      DO 1633 M=1,ME
      IF(HTZ(N,M))1633,1633,1632
 1632 V(N,M)=VSR(N,M)/(0,01*TIC)
      U(N,M)=USR(N,M)/(0,01*TIC)
 1633 CONTINUE
```

Preparation of rest currents for output.

```
      LOP=10
      GO TO 1634
   10 DO 9 I=1,NURU
      N=NZ(I+39)
      M=MZ(I+39)
      V2(I)=Z(N,M)
      V3(I)=RAD(N,M)
      V4(I)=ANG(N,M)
    9 CONTINUE
      WRITE (2)T,(V2(I),I=1,NURU),(V3(I),I=1,NURU),(V4(I),I=1,NURU)
      DO1000I=1,NURU
      N=NZ(I+39)
      M=MZ(I+39)
      V5(I)=Z'L(N,M)
      V6(I)=RAD2(N,M)
      V7(I)=ANG2(N,M)
 1000 CONTINUE
```

Output of data from selected points on tape to be printed later with subroutine SO8

```
      WRITE (6)T,(V5(I),I=1,NURU),(V6(I),I=1,NURU),(V7(I),I=1,NURU)
      TRB=54600.
      TRE=144000.
      IF(T-TRB)3620,3611,3611
3611  IF(T-TRE)3612,3612,3620
3612  AA1=AA1/T1
      AA2 = AA2/T1
      AA3 = AA3/T1
      AA4 = AA4/T1
      BB2=BB2/T1
      BB3 = BB3/T1
      BB4 = BB4/T1
      BB5 = BB5/T1
      BB6 = BB6/T1
      BB7 = BB7/T1
      BB8 = BB8/T1
      CC3=CC3/T1
      CC4 = CC4/T1
      CC5 = CC5/T1
      CC6 = CC6/T1
      CC7 = CC7/T1
      CC8 = CC8/T1
      PRINT 5101,AA1,BB2,CC3
      PRINT 5101,AA2,AA3,AA4
      PRINT 5101,BB3,BB4,BB5
      PRINT 5101,BB6,BB7,BB8
      PRINT 5101,CC4,CC5,CC6
      PRINT 5101,CC7,CC8
5101  FORMAT(1H1,//10X,26HTRANSPORT,SECTION A, M3=  ,F10.0,//28X,2HB,,
     15X,F10.0,//28X,2HC,,5X,F10.0,/7)
      AA1=0.
      AA2 = 0.
      AA3 = 0.
      AA4 = 0.
      BB2=0.
      BB3 = 0.
      BB4 = 0.
      BB5 = 0.
      BB6 = 0.
      BB7 = 0.
      BB8 = 0.
      CC3=0.
      CC4 = 0.
      CC5 = 0.
      CC6 = 0.
      CC7 = 0.
      CC8 = 0.
3620  CONTINUE
      PRINT 614,T,(N,N=1,14)
      PRINT 500,(N,(S(N,M),M=1,14),N=1,NE)
      PRINT 502,(N,N=15,28)
```

Time control for transport computation.

Output of transport through selected sections (m^3/hour)

Setting of sections to 0

```
      PRINT 500,(N,(S(N,M),M=15,28),N=1,NE)
      PRINT 502,(N,N=29,42)
      PRINT 500,(N,(S(N,M),M=29,42),N=1,NE)
      PRINT 501,(N,N=43,47)
      PRINT 503,(N,(S(N,M),M=43,47),N=1,NE)
  614 FORMAT(1H1,14X,37HCONCENTRATION OF LOWER LAYER AFTER T=,F9.0,//8X,
     1I8,15I8 /)
  452 RETURN
      END
      SUBROUTINE J05
      DIMENSION Z(42,47),U(42,47),V(42,47),ZM(42,47),ZST(42,47),HGU(42,4
     17),HGV(42,47),HTZ(42,47),HTU(42,47),HTV(42,47),XK(42,47),YK(42,47)
     2,A(12),NZ(90),MZ(90),NU(10),MU(10),Z1(80),Z2(80),V1(20),V2(20),U1(
     320),U2(20),X(8),ARG(8),RAD(42,47),ANG(42,47),V3(20),V4(20),
     4HTL(42,47),NA(60),MA(60),Y(8),Z3(80),Z4(80),U3(80),U4(80),CUN(8)
     5,V5(20),V6(20),V7(20),US(42,47),VS(42,47),S(42,47)
     6,ANG2(42,47),RAD2(42,47)
     7,USR(42,47),VSR(42,47)
      EQUIVALENCE(ZM,RAD),(ZST,ANG),(XK,Z2L),(HGU,RAD2),(YK,U2L),
     1(HGV,ANG2),(ZM2,HTV),(ZST2,HTU)
      COMMON Z,U,V,ZM,ZST,HGU,HGV,HTZ,HTU,HTV,XK,YK,A,NZ,MZ,NU,MU,Z1,Z2,
     1V1,V2,U1,U2,X,ARG,V3,V4,HTL,NA,MA,Y,Z3,Z4,U3,U4,CUN
     2,V5,V6,V7,US,VS,S,USR,VSR
     3,F,G,SIGMA,ALPHA,R,ROL,RBETA,C,DL,DT,T,T1,T2,NE,ME,IE,KKE,IZE,IUE,
     4TW,BETA,A1,A2,A3,A4,A5,C1,C3,SI,NURU,NURV,JA,NEH,MEH,NEHH,MEHH,POC
     5,NG,T3,STV,ETV,TINCV,STH,ETH,TINCH,TRAN,T4,AFGN,TIC,LEN,NS
     6,TRB,TRE
     7,AA2,AA3,AA4,BB3,BB4,BB5,BB6,BB7,BB8,CC4,CC5,CC6,CC7,CC8
      DIMENSION Z2L(42,47),U2L(42,47),V2L(42,47),ZM2(42,47),ZST2(42,47),
     1HGU2(42,47),HGV2(42,47),HTU2(42,47),HTV2(42,47),NU(50),MU(50)
      COMMON/LAYER2/V2L,HGU2,HGV2,HTU2,HTV2,ND,MD,AA1,BB2,CC3
      DATA RH 1,RHO2/1,026,1.029/
C005  COMP
      CALL WRITEC(RAD,15793,1974)
      CALL WRITEC(ANG,17767,1974)
      CALL WRITEC(RAD2,23689,1974)
      CALL WRITEC(ANG2,25663,1974)
      CALL READEC(ZM,1,1974)
      CALL READEC(ZST,1975,1974)
      CALL READEC(HGU,7897,1974)
      CALL READEC(HGV,9871,1974)
      CALL WRITEC(HTV,35533,1974)
      CALL READEC(ZM2,11845,1974)
      ALP=0.90
      BET=(1.-ALP)/4.
      DO 3110 N=1,NE
      DO 3110 M=1,ME
      ZST(N,M)=0.
 3110 ZM2(N,M)=0.
      DO 470 N=2,NEH
      DO 470 M=2,MEH
      IF(HTZ(N,M))710,470,75
  710 IF(HTZ(N,M)+2.)470,75,470
```

Printing of concentration fields (eg. pollution).

```
 75 IF(1-N)28,6,28
 28 IF(HTZ(N-1,M))57,6,57
 57 WERTO=Z(N-1,M)
    WERTO2 = Z2L(N-1,M)
    GO TO 7
  6 WERTO=Z(N,M)
    WERTO2 = Z2L(N,M)
  7 IF(NE-N)18,9,18
 18 IF(HTZ(N+1,M))58,9,58
 58 WERTU=Z(N+1,M)
    WERTU2 = Z2L(N+1,M)
    GO TO 10
  9 WERTU=Z(N,M)
    WERTU2 = Z2L(N,M)
 10 IF(1-M)29,12,29
 29 IF(HTZ(N,M-1))59,12,59
 59 WERTL=Z(N,M-1)
    WERTL2 = Z2L(N,M-1)
    GO TO 13
 12 WERTL=Z(N,M)
    WERTL2 = Z2L(N,M)
 13 IF(ME-M)17,15,17
 17 IF(HTZ(N,M+1))60,15,60
 60 WERTR=Z(N,M+1)
    WERTR2 = Z2L(N,M+1)
    GO TO 22
 15 WERTR=Z(N,M)
    WERTR2 = Z2L(N,M)
 22 ZM(N,M)=ALPHA*Z(N,M)+BETA*(WERTO+WERTU+WERTL+WERTR)
    ZM2(N,M) = ALP  *Z2L(N,M) + BET *(WERTO2 + WERTU2 + WERTL2+WERTR2)
470 CONTINUE
```

Computation of $\overline{Z}$ and $\overline{Z2L}$ ("smoothing").

```
    DO 70 N=2,NE
    DO 70 M=3,ME
604 IF(HTZ(N,M))70,70,149
149 Z(N,M)=ZM(N,M)-A4*(HGU(N,M)*U(N,M)-HGU(N,M-1)*U(N,M-1)+HGV(N-1,M)*
   1V(N-1,M)-HGV(N,M)*V(N,M)) - A4*(HGU2(N,M)*U2L(N,M) - HGU2(N,M-1)*
   2U2L(N,M-1) + HGV2(N-1,M)*V2L(N-1,M) - HGV2(N,M)*V2L(N,M))
    Z2L(N,M) = ZM2(N,M) - A4*(HGU2(N,M)*U2L(N,M) - HGU2(N,M-1)*U2L(N,
   1M-1) + HGV2(N-1,M)*V2L(N-1,M) - HGV2(N,M)*V2L(N,M))
 70 CONTINUE
```

Computation of Z and Z2L.

```
    DO 570 M=1,ME
    Z(2,M)=Z(3,M)
    Z(1,M)=Z(2,M)
    Z2L(2,M)=Z2L(3,M)
    Z2L(1,M)=Z2L(2,M)
    Z2L(2,M)=Z2L(3,M)
570 CONTINUE
```

Setting of boundaries for Z and Z2L.

```
    TD=104400.+T
    TG=TD-5400.
    TGT=TG-TD
    DO 78 N=2,40
    RB=N
    B=(RB-2.)/38.
```

Input of tides at one open boundary.

```
      Z(N,2)=28,3*COS(,00014092*(TD+B*  TGT  )-4,5736)+5,9*COS(
     1,000145416*(TD+B*  TGT  )-4,7656)+7,7*COS(,000137859*(TD+B*  TGT
     2)-4.2124)+6.6*COS(,000072908*(TD+B*  TGT  )-1,8340)+5,0*COS(
     3,000067585*(TD+B*  TGT  )-2,4256)
C
      Z(N,2) = 0.70*Z(N,2)
   78 CONTINUE
      DO 380 N=2,40
      PZ=FLOAT(N)
      Z(N,2)=Z(N,2)-4,/PZ
  380 CONTINUE
      DO 390  =2,44
      PZ=FLOAT(M)
      Z(2,M)=Z(2,M)-0.1*PZ
      Z(1,M)=Z(2,M)-0.06*PZ
  390 CONTINUE
      CALL WRITEC(ZM2,11845,1974)
      CALL WRITEC(ZST2,13819,1974)
      CALL READEC(HTU,37507,1974)
      CALL READEC(HTV,35533,1974)
 2003 DO 85 N=1,NEH
      DO 85 M=1,MEH
      IF(HTU(N,M))76,76,81
   81 HGU(N,M)=HTL(N,M)+(Z(N,M)+Z(N,M+1))/2,
   76 IF(HTU2(N,M))77,77,83
   83 HGU2(N,M)=HTU2(N,M)+(Z2L(N,M)+Z2L(N,M+1))/2,
   77 IF(HTV(N,M))85,85,87
   87 HGV(N,M)=HTL(N,M)+(Z(N,M)+Z(N+1,M))/2,
   82 IF(HTV2(N,M))85,85,84
   84 HGV2(N,M)=HTV2(N,M)+(Z2L(N,M)+Z2L(N+1,M))/2
   85 CONTINUE
      DO 760 N=1,NE
      DO 760 M=1,ME
      IF(HGU(N,M)-30,)761,761,762
  761 HGU(N,M)=0.
  762 IF(HGU2(N,M)-30,)763,763,764
  763 HGU2(N, )=0.
  764 IF(HGV(N,M)-30,)765,765,766
  765 HGV(N,M)=0.
  766 IF(HGV2(N,M)-30.)767,767,760
  767 HGV2(N,M)=0.
  760 CONTINUE
      IF(C1)101,103,161
  161 IF(T-TW)103,162,162
  162 CALL S01
  103 CALL WRITEC(HTU,37507,1974)
      CALL WRITEC(HTV,35533,1974)
      CALL READEC(ZST2,13819,1974)
      CALL READEC(ZM2,11845,1974)
      DO 3112 N=1,NE
      DO 3112 M=1,ME
 3112 ZST2(N,M)=0.
      DO 512 N=2,NEH
```

Specific adjustaments to boundaries.

Computations of actual depths and layer thickness.

Calling of wind subroutine (S01)

```
      DO 512 M=2,MEH
      IF(HGU(N,M))512,512,413
  413 IF(1-N)128,106,128
      IF(HGU(N,M))512,512,413
  413 IF(1-N)128,106,128
  128 IF(HGU(N-1,M))157,106,157
  157 WERTO=U(N-1,M)
      WERTO2= U2L(N-1,M)
      GO TO 107
  106 WERTO=U(N,M)
      WERTO2 = U2L(N,M)
  107 IF(NE-N)118,109,118
  118 IF(HGU(N+1,M))158,109,158
  158 WERTU=U(N+1,M)
      WERTU2 = U2L(N+1,M)
      GO TO 110
  109 WERTU=U(N,M)
      WERTU2 = U2L(N,M)
  110 IF(1-M)129,112,129
  129 IF(HGU(N,M-1))159,112,159
  159 WERTL=U(N,M-1)
      WERTL2 = U2L(N,M-1)
      GO TO 113
  112 WERTL=U(N,M)
      WERTL2 = U2L(N,M)
  113 IF(ME-M)117,115,117
  117 IF(HGU(N,M+1))160,115,160
  160 WERTR=U(N,M+1)
      WERTR2 = U2L(N,M+1)
      GO TO 23
  115 WERTR=U(N,M)
      WERTR2 = U2L(N,M)
   23 ZM(N,M)=ALPHA*U(N,M)+BETA*(WERTO+WERTU+WERTL+WERTR)
      ZM2(N,M) = ALPHA*U2L(N,M) + BETA*(WERTO2+WERTU2+WERTL2+WERTR2)
      IF(HGV(N-1,M))61,32,61
   61 WERTOL=V(N-1,M)
      WERTOL2 = V2L(N-1,M)
      IF(HGV(N-1,M+1))62,35,62
   62 WERTOR=V(N-1,M+1)
      WERTOR2 = V2L(N-1,M+1)
      GO TO 36
   32 WERTOL=V(N-1,M+1)
      WERTOL2 = V2L(N-1,M+1)
   35 WERTOR=WERTOL
      WERTOR2 = WERTOL2
   36 IF(HGV(N,M))63,38,63
   63 WERTUL=V(N,M)
      WERTUL2 = V2L(N,M)
      IF(HGV(N,M+1))64,41,64
   64 WERTUR=V(N,M+1)
      WERTUR2 = V2L(N,M+1)
      GO TO 42
   38 WERTUL=V(N,M+1)
```

Computation of $\overline{U}$ and $\overline{U2L}$ (smoothing).

Computation of V^* and $V2L^*$.

```
      WERTUL2 = V2L(N,M+1)
   41 WERTUR=WERTUL
      WERTUR2 = WERTUL2
   42 ZST(N,M)=(WERTOL+WERTUL+WERTUR+WERTUR)/4.
      ZST2(N,M) = (WERTOL2 + WERTUL2 + WERTUR2 + WERTUR2) / 4.0
  512 CONTINUE
      DO 473 M=2,ME
      ZM2(NE,M)=ZM2(NEH,M)
      ZM (NE,M)=ZM (NEH,M)
      ZST(NE,M)=ZST(NEH,M)
      ZST2(NE,M)=ZST2(NEH,M)
  473 CONTINUE
      CALL WRITEC(Z2L,31585,1974)
      CALL READEC(XK,3949,1974)
      DO 412 N=2,NE
      DO 412 M=2,MEH
      IF(HGU(N,M))412,412,513
  513 IF(ZM(N,M)*ZM(N,M))412,517,518
  517 IF(ZST(N,M)*ZST(N,M))412,412,518
  518 WURZEL=SQRTF(ZM(N,M)*ZM(N,M)+ZST(N,M)*ZST(N,M))
      GRZ=A3*WURZEL
      IF(HGU(N,M)-GRZ)94,94,14
   94 GRZ=HGU(N,M)
   14 ZM(N,M)=(1.-GRZ/HGU(N,M))*ZM(N,M)+A2*ZST(N,M)-A5*(Z(N,M+1)-Z(N,M))
     1+XK(N,M)
  412 CONTINUE
      CALL WRITEC(XK,3949,1974)
      CALL READEC(Z2L,31585,1974)
      DO 1050 N=2,NE
      DO 1050 M=2,MEH
      IF(HTU2(N,M)) 1050,1050,1005
 1005 IF(ZM2(N,M)*ZM2(N,M)) 1050,1010,1015
 1010 IF(ZST2(N,M)*ZST2(N,M)) 1050,1050,1015
 1015 WURZEL = SQRTF(ZM2(N,M)**2 + ZST2(N,M)**2)
      GRZ = A3*WURZEL
      IF(HGU2(N,M)-GRZ) 1030,1030,1025
 1030 GRZ=HGU2(N,M)
 1025 ZM2(N,M) = (1.-GRZ/HGU2(N,M))*ZM2(N,M) + A2*ZST2(N,M) - A5*(RHO1/
     1RHO2)*(Z(N,M+1)-Z(N,M)) - A5*(1.-RHO1/RHO2)*(Z2L(N,M+1)-Z2L(N,M
     2))
 1050 CONTINUE
      DO 474 M=2,NE
      ZM2(N,ME)=ZM2(N,MEH)
      ZM(N,ME)=ZM(N,MEH)
  474 CONTINUE
      DO 528 N=1,NEH
      DO 528 M=2,MEH
      IF(HGV(N,M))528,528,429
  429 IF(HGU(N,M-1))65,44,65
   65 WERTOL=U(N,M-1)
      WERTOL2 = U2L(N,M-1)
      IF(HGU(N+1,M-1))66,46,66
   66 WERTUL=U(N+1,M-1)
```

Setting of boundaries.

Computation of U and U2L.

Setting of boundaries.

```
      WERTUL2 = U2L(N+1,M-1)
      GO TO 47
   44 WERTOL=U(N+1,M-1)
      WERTOL2 = U2L(N+1,M-1)
   46 WERTUL=WERTOL
      WERTUL2 = WERTOL2
   47 IF(HGU(N,M))67,49,67
   67 WERTOR=U(N,M)
      WERTOR2 = U2L(N,M)
      IF(HGU(N+1,M))68,52,68
   68 WERTUR=U(N+1,M)
      WERTUR2 = U2L(N+1,M)
      GO TO 53
   49 WERTOR=U(N+1,M)
      WERTOR2 = U2L(N+1,M)
   52 WERTUR=WERTOR
      WERTUR2 = WERTOR2
   53 ZST(N,M)=(WERTOL+WERTUL+WERTOR+WERTUR)/4,
      ZST2(N,M) = (WERTOL2 + WERTUL2 + WERTOR2 + WERTUR2) /4,U
  528 CONTINUE
      CALL WRITEC(ZST2,13819,1974)
      CALL READEC(HTU,37507,1974)
      DO 430 N=1,NE
      DO 430 M=1,ME
      IF(HTU(N,M))430,430,163
  163 IF(HGU(N,M))170,170,171
  170 U(N,M)=0.
      GO TO 172
  171 U(N,M)=ZM(N,M)
  172 IF(HGU2(N,M))173,173,174
  173 U2L(N,M)=0.
      GO TO 430
  174 U2L(N,M)=ZM2(N,M)
  430 CONTINUE
C
      CALL WRITEC(HTU,37507,1974)
      CALL READEC(ZST2,13819,1974)
      DO 431 N=2,NEH
      DO 431 M=2,MEH
      IF(HGV(N,M))431,431,432
  432 IF(1-N)228,206,228
  228 IF(HGV(N-1,M))257,206,257
  257 WERTO=V(N-1,M)
      WERTO2 = V2L(N-1,M)
      GO TO 207
  206 WERTO=V(N,M)
      WERTO2 = V2L(N,M)
  207 IF(NE-N)218,209,218
  218 IF(HGV(N+1,M))258,209,258
  258 WERTU=V(N+1,M)
      WERTU2 = V2L(N+1,M)
      GO TO 210
  209 WERTU=V(N,M)
```

Computation of U^* and $U2L^*$.

Transfer of new U and U2L into proper fields

Computation of $\overline{V}$ and $\overline{V2L}$ (smoothing).

```
      WERTU2 = V2L(N,M)
  210 IF(1-M)229,212,229
  229 IF(HGV(N,M-1))259,212,259
  259 WERTL=V(N,M-1)
      WERTL2 = V2L(N,M-1)
      GO TO 213
  212 WERTL=V(N,M)
      WERTL2 = V2L(N,M)
  213 IF(ME-M)217,215,217
  217 IF(HGV(N,M+1))260,215,260
  260 WERTR=V(N,M+1)
      WERTR2 = V2L(N,M+1)
      GO TO 25
  215 WERTR=V(N,M)
      WERTR2 = V2L(N,M)
   25 ZM(N,M)=ALPHA*V(N,M)+BETA*(WERTO+WERTU+WERTL+WERTR)
      ZM2(N,M) = ALPHA*V2L(N,M) + BETA*(WERTO2 + WERTU2 + WERTL2 +WERTR2
     1)
  431 CONTINUE
      DO 475 N=2,NE
      ZST(N,ME)=ZST(N,MEH)
      ZST2(N,ME)=ZST2(N,MEH)
  475 CONTINUE
      CALL WRITEC(U2L,33559,1974)
      CALL READEC(YK,5923,1974)
      DO 428 N=2,NEH
      DO 428 M=2,ME
      IF(HGV(N,M))428,428,529
  529 IF(ZM(N,M)*ZM(N,M))428,617,618
  617 IF(ZST(N,M)*ZST(N,M))428,428,618
  618 WURZEL=SQRTF(ZM(N,M)*ZM(N,M)+ZST(N,M)*ZST(N,M))
      GRZ=A3*WURZEL
      IF(HGV(N,M)-GRZ)300,300,415
  300 GRZ=HGV(N,M)
  415 V(N,M)=(1.-GRZ/HGV(N,M))*ZM(N,M)-A2*ZST(N,M)-A5*(Z(N,M)-Z(N+1,M))+
     1YK(N,M)
  428 CONTINUE
      V(26,40) = V(26,40) + 1.3
      CALL WRITEC(YK,5923,1974)
      CALL READEC(U2L,33559,1974)
      DO 1100 N=2,NEH
      DO 1100 M=2,ME
      IF(HGV2(N,M)) 1100,1100,1055
 1055 IF(ZM2(N,M)*ZM2(N,M)) 1100,1060,1065
 1060 IF(ZST2(N,M)*ZST2(N,M)) 1100,1100,1065
 1065 WURZEL = SQRTF(ZM2(N,M)**2 + ZST2(N,M)**2)
      GRZ = A3*WURZEL
      IF(HGV2(N,M) - GRZ) 1080,1080,1075
 1080 GRZ=HGV2(N,M)
 1075 V2L(N,M) = (1.-GRZ/HGV2(N,M))*ZM2(N,M) - A2*ZST2(N,M) + A5*(RHO1/
     1RHO2)*(Z(N,M) - Z(N+1,M)) - A5*(1. - RHO1/RHO2)*(Z2L(N,M) - Z2L(N+
     11,M))
 1100 CONTINUE
      DO 476 M=2,ME
```

Setting of boundaries.

Computation of V.

setting of special inflow.

Computation of V2L.

```
      S(17,37) = 10000,
      S(22,9)  = 10000,
      S(27,18) = 10000,
      S(32,7)  = 10000,
      S(34,27) = 10000,
      S(38,6)  = 10000,
      S(38,15) = 10000,
      S(26,40) = S(26,40) + 3,
 4412 B1=DL*DL
      DL2=2.*DL
      AUS=5.0E4
 4413 B2 = (4.0*DT*AUS    )/B1
  414 B4=2.*DL
 4415 B5 = (AUS    * DT)/B1
  416 Y1=0.
      S(26,40) = S(26,40) + 3.
C     COMPUTATION OF DISPERSION AND DIFFUSION
 4417 DO 3000 N=2,NEH
      DO 3000 M=2,MEH
      IF(HTZ(N,M))3000,3000,201
  201 IF(HTU(N,M))202,202,204
  202 SSH=0.
      GO TO 225
  204 IF(U(N,M))205,205,211
  205 IF(US(N,M)+B4)2209,2206,2209
 2206 IF(S(N,M+1))2207,2207,2208
 2207 SSH=S(N,M)/(DT*ABS(U(N,M)))
      US(N,M)=0.
      GO TO 225
 2208 SSH=S(N,M+1)/(DT*ABS(U(N,M)))
      US(N,M)=0.
      GO TO 225
 2209 SSH=(S(N,M)-S(N,M+1))/DL2
      GO TO 225
  211 IF(US(N,M)-B4)2213,2210,2210
 2210 IF(S(N,M-1))2211,2211,2212
 2211 SSH=S(N,M)/(DT*ABS(U(N,M)))
      US(N,M)=0.
      GO TO 225
 2212 SSH=S(N,M-1)/(DT*ABS(U(N,M)))
      US(N,M)=0.
      GO TO 225
 2213 SSH=(S(N,M)-S(N,M-1))/DL2
  225 IF(HTV(N,M))2214,2214,2215
 2214 IF(HTU(N,M))3000,3000,2230
 2230 SSV=0.
      GO TO 4428
 2215 IF(V(N,M))2216,2216,2221
 2216 IF(VS(N,M)+B4)2220,2217,2220
 2217 IF(S(N-1,M))2218,2218,2219
 2218 SSV=S(N,M)/(DT*ABS(V(N,M)))
      VS(N,M)=0.
      GO TO 4428
```

continuous source.

Computation of dispersion and diffusion of pollutants.

```
 2219 SSV=S(N-1,M)/(DT*ABS(V(N,M)))
      VS(N,M)=0.
      GO TO 4428
 2220 SSV=(S(N,M)-S(N-1,M))/DL2
      GO TO 4428
 2221 IF(VS(N,M)-B4)2225,2222,2225
 2222 IF(S(N+1,M))2223,2223,2224
 2223 SSV=S(N,M)/(DT*ABS(V(N,M)))
      VS(N,M)=0.
      GO TO 4428
 2224 SSV=S(N+1,M)/(DT*ABS(V(N,M)))
      VS(N,M)=0.
      GO TO 4428
 2225 SSV=(S(N,M)-S(N+1,M))/DL2
 4428 S(N,M)=S(N,M)-B2*S(N,M)+B5*(S(N-1,M)+S(N+1,M)+S(N,M-1)+S(N,M+1)
     1-4.*S(N,M))
     2-(DT*ABSF(U2L(N,M))*SSH)-(DT*ABSF(V2L(N,M))*SSV)+Y1
     3+B5*(S(N-1,M-1)+S(N-1,M+1)+S(N+1,M-1)+S(N+1,M+1))
 4429 IF(S(N,M)) 2429,2429,3000
 2429 S(N,M) = 0.
 3000 CONTINUE
C 415 B5=2.0*B4
C 416 DO 422 N=2,NEH
C 417 DO 422 M=2,MEH
C 418 IF(HGU(N,M))420,420,419
C 419 S(N,M)=S(N,M)-A1*U(N,M)*(S(N,M+1)/B4-S(N,M-1)/B4)+B2*(S(N,M-1)/2.-
C 420 IF(HGV(N,M))422,422,421
C     1S(N,M)+S(N,M+1)/2.)+B2*(S(N-1,M)/2.-S(N,M)+S(N+1,M)/2.)
C 421 S(N,M)= (N,M)-A1*V(N,M)*(S(N+1,M)/B4-S(N-1,M)/B4)+B2*(S(N-1,M)/2.-
C     2S(N,M)+S(N+1,M)/2.)+B2*(S(N,M-1)/2.-S(N,M)+S(N,M+1)/2.)
C 422 CONTINUE
  984 CONTINUE
   11 RETURN
      END
      FUNCTION TRANS(FV,FH,N,M,KEY,NUM)
      DIMENSION FV(42,47),FH(42,47)
C
C     FV = U OR V COMPONENT OF VELOCITY
C     FH = HEIGHT ON U OR V GRID
C     N = INITIAL VALUE OF N....MIN
C     M = INITIAL VALUE OF M....MIN
C      KEY = 1...SECTION IS ALONG CONSTANT N
C      KEY = 2...SECTION IS ALONG CONSTANT M
C      NUM = NUMBER OF POINTS ON THE SECTION
C
      TOT = 0
      M1 = M-1
      N1 = N-1
C
      GO TO(100,200) KEY
C
  100 CONTINUE
```

Function for computation of transport through sections.

```
C     THIS IS A SECTION ALONG CONSTANT N ...V VELOCITY USED
      DO 150 K = 1,NUM
      TOT = TOT + FV(N,M1+K)*FH(N,M1+K)
  150 CONTINUE
      TRANS = TOT
      RETURN
C
  200 CONTINUE
C     THIS IS A SECTION ALONG CONSTANT M...U VELOCITY USED
      DO 250K = 1,NUM
      TOT = TOT + FV(N1+K,M)*FH(N1+K,M)
  250 CONTINUE
      TRANS = TOT
      RETURN
      END
      SUBROUTINE S01
      DIMENSION Z(42,47),U(42,47),V(42,47),ZM(42,47),ZST(42,47),HGU(42,4
     17),HGV(42,47),HTZ(42,47),HTU(42,47),HTV(42,47),XK(42,47),YK(42,47)
     2,A(12),NZ(90),MZ(90),NU(10),MU(10),Z1(80),Z2(80),V1(20),V2(20),U1(
     320),U2(20),X(8),ARG(8),RAD(42,47),ANG(42,47),V3(20),V4(20),
     4HTL(42,47),NA(60),MA(60),Y(8),Z3(80),Z4(80),U3(80),U4(80),CON(8)
     5,V5(20),V6(20),V7(20),US(42,47),VS(42,47),S(42,47)
     6,ANG2(42,47),RAD2(42,47)
     7,USR(42,47),VSR(42,47)
      EQUIVALENCE(ZM,RAD),(ZST,ANG),(XK,Z2L),(HGU,RAD2),(YK,U2L),
     1(HGV,ANG2),(ZM2,HTV),(ZST2,HTU)
      COMMON Z,U,V,ZM,ZST,HGU,HGV,HTZ,HTU,HTV,XK,YK,A,NZ,MZ,NU,MU,Z1,Z2,
     1V1,V2,U1,U2,X,ARG,V3,V4,HTL,NA,MA,Y,Z3,Z4,U3,U4,CON
     2,V5,V6,V7,US,VS,S,USR,VSR
     3,F,G,SIGMA,ALPHA,R,RDL,HBETA,C,DL,DT,T,T1,T2,NE,ME,IE,KKE,IZE,IUE,
     4TW,BETA,A1,A2,A3,A4,A5,C1,C3,SI,NURU,NURV,JA,NEH,MEH,NEHH,MEHH,POC
     5,NG,T3,ETV,ETV,TINCV,STH,ETH,TINCH,THAN,T4,AFGN,TIC,LEN,NS
     6,TRB,TRE
     7,AA2,AA3,AA4,BB3,BB4,BB5,BB6,BB7,BB8,CC4,CC5,CC6,CC7,CC8
      DIMENSION Z2L(42,47),U2L(42,47),V2L(42,47),ZM2(42,47),ZST2(42,47),
     1HGU2(42,47),HGV2(42,47),HTU2(42,47),HTV2(42,47),ND(50),MD(50)
      COMMON/LAYER2/V2L,HGU2,HGV2,HTU2,HTV2,ND,MD,AA1,BB2,CC3
C S01 WIND
      CALL WRITEC(Z2L,31585,1974)
      CALL WRITEC(U2L,33559,1974)
      CALL READEC(XK,3949,1974)
      CALL READEC(YK,5923,1974)
      IF(SI-T)12,10,2
    2 DO 8 N=1,NE
      DO 8 M=1,ME
      IF(HTU(N,M))4,4,3
    3 IF(HGU(N,M))4,4,50
   50 HMLD=2000.+8.*Z(N,M)
   51 IF(HMLD-HGU(N,M))5,52,52
   52 HMLD=HGU(N,M)
    5 XK(N,M)=C3*A(2)**2*COS(A(3)*0.01745)/HGU(N,M)
    4 IF(HTV( ,M))8,8,6
    6 IF(HGV(N,M))8,8,70
   70 IF(HMLD-HGV(N,M))7,61,61
```

Computation of wind current component.

```
   61 HMLD=HGV(N,M)
    7 YK(N,M)=C3*A(2)**2*SIN(A(3)*0.01745)/HGV(N,M)
    8 CONTINUE
      GO TO 10
   10 DO 11 N=1,NE
      DO 11 M=1,ME
      XK(N,M)=0.
   11 YK(N,M)=0.
   12 CALL WRITEC(XK,3949,1974)
      CALL WRITEC(YK,5923,1974)
      CALL READEC(Z2L,31585,1974)
      CALL READEC(U2L,33559,1974)
      RETURN
      END
      SUBROUTINE SUB
      DIMENSION Z(42,47),U(42,47),V(42,47),ZM(42,47),ZST(42,47),HGU(42,4
     17),HGV(42,47),HTZ(42,47),HTU(42,47),HTV(42,47),XK(42,47),YK(42,47)
     2,A(12),NZ(90),MZ(90),NU(10),MU(10),Z1(80),Z2(80),V1(20),V2(20),U1(
     320),U2(20),X(8),ARG(8),RAD(42,47),ANG(42,47),V3(20),V4(20),
     4HTL(42,47),NA(60),MA(60),Y(8),Z3(80),Z4(80),U3(80),U4(80),CON(8)
     5,V5(20),V6(20),V7(20),US(42,47),VS(42,47),S(42,47)
     6,ANG2(42,47),RAD2(42,47)
     7,USR(42,47),VSR(42,47)
      EQUIVALENCE(ZM,RAD),(ZST,ANG),(XK,Z2L),(HGU,RAD2),(YK,U2L),
     1(HGV,ANG2),(ZM2,HTV),(ZST2,HTU)
      COMMON Z,U,V,ZM,ZST,HGU,HGV,HTZ,HTU,HTV,XK,YK,A,NZ,MZ,NU,MU,Z1,Z2,
     1V1,V2,U1,U2,X,ARG,V3,V4,HTL,NA,MA,Y,Z3,Z4,U3,U4,CON
     2,V5,V6,V7,US,VS,S,USR,VSR
     3,F,G,SIGMA,ALPHA,R,ROL,RBETA,C,DL,DT,I,T1,T2,NE,ME,IE,KKE,IZE,IUE,
     4TW,BETA,A1,A2,A3,A4,A5,C1,C3,SI,NURU,NURV,JA,NEH,MEH,NEHH,MEHH,POC
     5,NG,T3,STV,ETV,TINCV,STH,ETH,TINCH,TRAN,T4,ABGN,TIC,LEN,NS
     6,TRB,TRE
     7,AA2,AA3,AA4,BB3,BB4,BB5,BB6,BB7,BB8,CC4,CC5,CC6,CC7,CC8
      DIMENSION Z2L(42,47),U2L(42,47),V2L(42,47),ZM2(42,47),ZST2(42,47),
     1HGU2(42,47),HGV2(42,47),HTU2(42,47),HTV2(42,47),NU(50),MU(50)
      COMMON/LAYER2/V2L,HGU2,HGV2,HTU2,HTV2,ND,MD,AA1,BB2,CC3
C SUB OUTPUT FOR SPECIAL POINTS
  131 FORMAT(1H1,5X,115HWATER LEVEL (CM),SEC CURRENT SPEED (CM/SEC),DIRE
     1CTION (DEG),MLD (M),DEEP CURRENT SPEED (CM/SEC) AND DIRECTION (DEG
     2)///7X,9HTIME,4X,10(2X,A7))
  132 FORMAT((6X,5H(SEC),4X,10(2X,I3,1H,,I3)//))
  133 FORMAT(F11.0,4X,10F9.2)
  140 FORMAT((15X,10(F4.0,1H,,F4.0)//15X,10F9.2//15X,10(F4.0,1H,,F4.0)//
     1)
  134 FORMAT(5X,14HEND OF PROGRAM)
      NURV=0
      PRINT 131,(V1(I),I=1,NURU)
      PRINT 132,(NZ(I+39),MZ(I+39),I=1,NURU)
    1 READ  (2)T,(V2(I),I=1,NURU),(V3(I),I=1,NURU),(V4(I),I=1,NURU)
      READ  (2)T,(V5(I),I=1,NURU),(V6(I),I=1,NURU),(V7(I),I=1,NURU)
      PRINT 133,T,(V2(I),I=1,NURU)
      PRINT 140,(V3(I),V4(I),V5(I),V6(I),V7(I),I=1,NURU)
```

Output of values from special points.

```
      IF(TE-T)2,2,1
    2 PRINT 134
      RETURN
      END
      SUBROUTINE JOB6
      DIMENSION Z(42,47),U(42,47),V(42,47),ZM(42,47),ZST(42,47),HGU(42,4
     17),HGV(42,47),HTZ(42,47),HTU(42,47),HTV(42,47),XK(42,47),YK(42,47)
     2,A(12),NZ(90),MZ(90),NU(10),MU(10),Z1(80),Z2(80),V1(20),V2(20),U1(
     320),U2(20),X(8),ARG(8),RAD(42,47),ANG(42,47),V3(20),V4(20),
     4HTL(42,47),NA(60),MA(60),Y(8),Z3(80),Z4(80),U3(80),U4(80),CON(8)
     5,V5(20),V6(20),V7(20),US(42,47),VS(42,47),S(42,47)
     6,ANG2(42,47),RAD2(42,47)
     7,USR(42,47),VSR(42,47)
      EQUIVALENCE(ZM,RAD),(ZST,ANG),(XK,Z2L),(HGU,RAD2),(YK,U2L),
     1(HGV,ANG2),(ZM2,HTV),(ZST2,HTU)
      COMMON Z,U,V,ZM,ZST,HGU,HGV,HTZ,HTU,HTV,XK,YK,A,NZ,MZ,NU,MU,Z1,Z2,
     1V1,V2,U1,U2,X,ARG,V3,V4,HTL,NA,MA,Y,Z3,Z4,U3,U4,CON
     2,V5,V6,V7,US,VS,S,USR,VSR
     3,F,G,SIGMA,ALPHA,R,ROL,RBETA,C,DL,DT,I,T1,T2,NE,ME,IE,KKE,IZE,IUE,
     4TW,BETA,A1,A2,A3,A4,A5,C1,C3,SI,NURU,NURV,JA,NEH,MEH,NEHH,MEHH,POC
     5,NG,T3,STV,ETV,TINCV,STH,ETH,TINCH,TRAN,T4,AFGN,TIC,LEN,NS
     6,TRB,TRE
     7,AA2,AA3,AA4,BB3,BB4,BB5,BB6,BB7,BB8,CC4,CC5,CC6,CC7,CC8
      DIMENSION Z2L(42,47),U2L(42,47),V2L(42,47),ZM2(42,47),ZST2(42,47),
     1HGU2(42,47),HGV2(42,47),HTU2(42,47),HTV2(42,47),ND(50),MD(50)
      COMMON/LAYER2/V2L,HGU2,HGV2,HTU2,HTV2,ND,MD,AA1,BB2,CC3
C     ENTRY TO READ VECTOR + HEIGHT CARDS
      ETH = ETC
  200 READ            (3,215)STV,ETV,TINCV
  201 PRINT 216,STV,ETV,TINCV
  202 READ            (3,215)STH,ETC,TINCH
  203 PRINT 216,STH,ETC,TINCH
  215 FORMAT(3F10.0)
  216 FORMAT(10X,3F10.0)
C     PAUSE 3 IS FOR MOUNTING OF PLOTTING TAPE
  218 WRITE      (4) HTZ,V1,NURU,ME,NE
      RETURN
      END
      SUBROUTINE JOB7
      DIMENSION Z(42,47),U(42,47),V(42,47),ZM(42,47),ZST(42,47),HGU(42,4
     17),HGV(42,47),HTZ(42,47),HTU(42,47),HTV(42,47),XK(42,47),YK(42,47)
     2,A(12),NZ(90),MZ(90),NU(10),MU(10),Z1(80),Z2(80),V1(20),V2(20),U1(
     320),U2(20),X(8),ARG(8),RAD(42,47),ANG(42,47),V3(20),V4(20),
     4HTL(42,47),NA(60),MA(60),Y(8),Z3(80),Z4(80),U3(80),U4(80),CON(8)
     5,V5(20),V6(20),V7(20),US(42,47),VS(42,47),S(42,47)
     6,ANG2(42,47),RAD2(42,47)
     7,USR(42,47),VSR(42,47)
      EQUIVALENCE(ZM,RAD),(ZST,ANG),(XK,Z2L),(HGU,RAD2),(YK,U2L),
     1(HGV,ANG2),(ZM2,HTV),(ZST2,HTU)
      COMMON Z,U,V,ZM,ZST,HGU,HGV,HTZ,HTU,HTV,XK,YK,A,NZ,MZ,NU,MU,Z1,Z2,
     1V1,V2,U1,U2,X,ARG,V3,V4,HTL,NA,MA,Y,Z3,Z4,U3,U4,CON
     2,V5,V6,V7,US,VS,S,USR,VSR
     3,F,G,SIGMA,ALPHA,R,ROL,RBETA,C,DL,DT,I,T1,T2,NE,ME,IE,KKE,IZE,IUE,
```

A plotting subroutine.

```
     4TW,BETA,A1,A2,A3,A4,A5,C1,C3,SI,NURU,NURV,JA,NEH,MEH,NEHH,MEHH,POC
     5,NG,T3,STV,ETV,TINCV,STH,ETH,TINCH,TRAN,T4,ALGN,TIC,LEN,NS
     6,TRB,TRE
     7,AA2,AA3,AA4,BB3,BB4,BB5,BB6,BB7,BB8,CC4,CC5,CC6,CC7,CC8
      DIMENSION Z2L(42,47),U2L(42,47),V2L(42,47),ZM2(42,47),ZST2(42,47),
     1HGU2(42,47),HGV2(42,47),HTU2(42,47),HTV2(42,47),ND(50),MD(50)
      COMMON/LAYER2/V2L,HGU2,HGV2,HTU2,HTV2,ND,MD,AA1,BB2,CC3
C     ENTRY TO WRITE DATA ON TAPE FOR VECTOR AND WATER HEIGHT PLOT
  230 VEC=HT=0.
  231 IF(ABSF(T-STV)-T1/2.)232,232,234
  232 STV=STV+TINCV
  233 VEC=1.
  234 IF(ABSF(T-STH)-T1/2.)235,235,237
  235 STH=STH+TINCH
  236 HT=1.
  237 IF(HT+VEC)238,239,238
  238 WRITE      (4)T,VEC,HT,RAD,ANG,Z
      WRITE (5)T,VEC,HT,RAD2,ANG2,Z2L
  239 CONTINUE
      RETURN
      END
      FUNCTION ASINF(X)
      ASINF = ASIN(X)
      RETURN
      END
      FUNCTION COSF(X)
      COSF = COS(X)
      RETURN
      END
      FUNCTION SQRTF(X)
      SQRTF = SQRT(X)
      RETURN
      END
      FUNCTION SINF(X)
      SINF = SIN(X)
      RETURN
      END
      FUNCTION ABSF(X)
      ABSF = ABS(X)
      RETURN
```

→ A plotting subroutine.