ENVPREDRSCHFAC
Technical Note No. 1-74

A VERTICALLY INTEGRATED HYDRODYNAMICAL-NUMERICAL MODEL (W. HANSEN TYPE)

MODEL DESCRIPTION AND OPERATING/RUNNING INSTRUCTIONS

By

OCEANOGRAPHY DEPARTMENT
ENVIRONMENTAL PREDICTION RESEARCH FACILITY

Part 1 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. 1-74	2. GOVT ACCESSION NO.
	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): A Vertically Integrated Hydrodynamical-Numerical Model (W. Hansen Type) Model Description and Operating/Running Instructions.	5. TYPE OF REPORT & PERIOD COVERED
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Oceanography Department, Environmental Prediction Research Facility	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: 69
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)

Hydrodynamical-Numerical Model
Transport and diffusion of pollutants
Analysis and prediction of tides
Storm surge prediction
Drift computations

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

This report presents an operational description of the single-layer vertically integrated Hydrodynamical-Numerical (HN) model. The model is an adaptation of one developed by Professor Walter Hansen, University of Hamburg, and tested over the course of two decades. The description includes the setting up of the model for a given area and the selection and preparation of input data. A brief background of the

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)

hydrodynamical formulas and their finite difference forms are also presented. Certain new features which have been added to the program are described.

For coastal applications, it is now possible to run the model with three open boundaries. The treatment of the boundaries is described in this paper.

Another new feature makes possible the inclusion of the permanent (thermohaline component) current as well as tidal and wind currents. The new model also allows the specification and subsequent modification of the mixed layer depth in case there is thermal and haline stratification present in the area. The wind current will be effective only in the upper mixed layer.

The transport and diffusion equation is included in the new program. Thus, the program can be used for computing the distribution of pollutants. In case of oil pollution or other flotsam on the surface, the oil patch or flotsam is given an additional push from the wind.

The program (see Appendix C) is written in FORTRAN and thus will run on almost any computer. If the field size is less than 1,000 grid points (e.g., 25x40) the model can be run on a computer with a 32K memory. If the time step is not too small, a 24-hr prediction requires about 30 minutes of computer time on a 32K machine.

FOREWORD

The single-layer, vertically integrated Hydrodynamical-Numerical (HN) model has been used at Environmental Prediction Research Facility for analysis and prediction of tides, storm surges, real-time currents and for the computation of drift for search and rescue missions. Current components are computed in short time intervals and, thus, the model is ideally suited for analysis and prediction of the dispersal and diffusion of pollutants. After testing various finite difference forms for inclusion of diffusion into the model, the one presented in this program description was found most suited for models of small grid size.

The model was applied to three different size areas of the New York Bight. The results and verification of the model are given in other reports in this series.

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 Environmental Prediction Research Facility under the supervision of Dr. Taivo Laevastu.

CONTENTS

I. THE BASIC HYDRODYNAMICAL EQUATIONS AND THEIR FINITE DIFFERENCE FORMS

The use of the numerical hydrodynamical method for the computation of tides and currents was originally proposed by Hansen in 1938.[1] With the development and availability of high-speed electronic computers the use of this method became economically feasible and practically possible.

The derivation of vertically integrated hydrodynamical equations is described by Hansen (1956),[2] Sündermann (1966),[3] Jensen, Weywadt and Jensen (1966),[4] and Brettschneider (1967).[5] The following basic hydrodynamical equations are used in the single-layer model:

$$\frac{\partial u}{\partial t} + fv - \nu\nabla^2 u + \frac{r}{H}u\sqrt{u^2+v^2} + g\frac{\partial \zeta}{x} = X + \frac{\tau^{(x)}}{H} \qquad (1)$$

---

[1] Hansen, W., 1938: Amplitudenverhältnis und Phasenunterschied der harmonischen Konstanten in der Nordsee. Ann. d. Hydr. u Marit. Met. 66 (9):429-443.

[2] Hansen, W., 1956: Theorie zur Errechnung des Wasserstandes und der Strömungen in Randmeeren nebst Anwendungen. Tellus, 8:287-300.

[3] Sündermann, J., 1966: Ein Vergleich zwischen der analytischen under der numerischen Berechung winderzengter Strömungen und Wasserstände in einem Modellmeer mit Anwendungen auf die Nordsee. Mitteil. Inst. Meeresk., Hamburg 4:73 pp. tables & figures.

[4] Jensen, H. D., S. Weywadt and A. Jensen, 1966: Forecasting of storm surges in the North Sea. Part I. NATO Subcomm. Oceanogr. Res. Techn. Rpt. 28 (Mimeo).

[5] Brettschneider, G., 1967: Anwendung des hydrodynamisch-numerischen Verfahrens zur Ermittlung der M_2 - Mitschwingungsgezeit der Nordsee. Mitteil. Inst. Meeresk., Hamburg, 7:65 pp and figures.

$$\frac{\partial v}{\partial t} - fu - \nu\nabla^2 v + \frac{r}{H}v\sqrt{u^2+v^2} + g\frac{\partial \zeta}{y} = Y + \frac{\tau(y)}{H} \qquad (2)$$

$$\frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x}(Hu) + \frac{\partial}{\partial y}(Hv) = 0 \qquad (3)$$

Equations (1) and (2) are vertically integrated equations of motion, and equation (3) is an equation of continuity. $\tau^{(x)}$ and $\tau^{(y)}$ (wind stress) are usually expressed as:

$$\begin{aligned} \tau^{(x)} &= \lambda \cdot W_x \sqrt{W_x^2 + W_y^2} \\ \tau^{(y)} &= \lambda\, W_y \sqrt{W_x^2 + W_y^2} \end{aligned} \qquad (4)$$

The bottom stress (friction) ($\tau^{(b)}$) in equations (1) and (2) is:

$$\frac{r}{H}\, u \sqrt{u^2 + v^2};\ \frac{r}{H}\, v \sqrt{u^2 + v^2} \qquad (5)$$

Analytical solutions to equations (1) through (3) are of little value since exact solutions are possible only for basins with a regular shape and constant depth and wind distributions. However, Hansen (1956) and his later collaborators (e.g., Südermann (1966) and Brettschneider (1967)) have developed an explicit method for achieving time-dependent solutions to these formulas using a finite difference approach. The finite difference approximations for equations (1) through (3) are:

$$\zeta^{t+\tau}(n,m) = \bar{\zeta}^{t-\tau}(n,m) - \frac{\tau}{\ell}\Big\{H_u^t(n,m)U^t(n,m) - H_u^t(n,m-1)$$
$$U^t(n,m-1) + H_v^t(n-1,m)V^t(n-1,m) - H_v^t(n,m)V^t(n,m) \qquad (6)$$

$$U^{t+2\tau}(n,m) = \left\{1 - \left[2\tau r/H_u^{t+2\tau}(n,m)\right] \sqrt{\bar{U}^t(n,m)^2+V^{*t}(n,m)^2}\right\}$$

$$\bar{U}^t(n,m) + 2\tau\ fV^{*t}(n,m) - \frac{\tau g}{\ell}\left\{\zeta^{t+\tau}(n,m+1) - \zeta^{t+\tau}(n,m)\right\} +$$

$$2\tau X^{t+2\tau}(n,m) \qquad (7)$$

$$V^{t+2\tau}(n,m) = \left\{1 - \left[2\tau r/H_v^{t+2\tau}(n,m)\right] \sqrt{\bar{V}^t(n,m)^2+U^{*t}(n,m)^2}\right\}$$

$$\bar{V}^t(n,m) - 2\tau\ fU^{*t}(n,m) - \frac{\tau g}{\ell}\left\{\zeta^{t+\tau}(n,m) - \zeta^{t+\tau}(n+1,m)\right\} +$$

$$2\tau Y^{t+2\tau}(n,m) \qquad (8)$$

The following symbols are used in the preceding formulas:

x,y	space coordinates
t	time
u,v	components of velocity
h	initial depth (when $\zeta=0$)
ζ	surface elevation
H	total depth ($H=h+\zeta$)
H_u, H_v	depths at u and v points respectively
X,Y	components of external forces
$\tau^{(x)},\tau^{(y)}$	components of wind stress
g	acceleration of gravity
f	Coriolis parameter
r	friction coefficient (bottom stress)
ν	coefficient of horizontal eddy viscosity
∇^2	Laplacian
λ	coefficient of friction (drag coefficient)
W_x,W_y	wind speeds
$\tau^{(b)}$	bottom stress
n,m	coordinates of the grid point
τ	half time step } in finite difference equations
ℓ	grid length } in finite difference equations

The averaged velocity and water elevation (sea level) components are:

$$\bar{U}^t(n,m) = \alpha\, U^t(n,m) + \frac{1-\alpha}{4}\left\{U^t(n-1,m) + U^t(n+1,m) + U^t(n,m+1) + U^t(n,m-1)\right\}.$$

$\bar{V}^t(n,m)$ and $\bar{\zeta}^t(n,m)$ are analogous to $\bar{U}^t(n,m)$ above. (9)

(The factor α can be interpreted as a "horizontal viscosity parameter." Its normal value is 0.99).

$$U^{*t}(n,m) = \frac{1}{4}\left\{U^t(n,m-1) + U^t(n+1,m-1) + U^t(n,m) + U^t(n+1,m)\right\} \qquad (10)$$

$V^{*t}(n,m)$ is analogous to the $U^{*t}(n,m)$ above.

The time step is 2τ. The total depth (H_u, H_v) is computed as:

$$H_u^{t+2\tau}(n,m) = h_u(n,m) + \frac{1}{2}\left[\zeta^{t+\tau}(n,m) + \zeta^{t+\tau}(n,m+1)\right] \qquad (11)$$

The effects of wind (external force) are computed with the following formulas:

$$X^t = \frac{\lambda\, W_x^t \sqrt{(W_x^t)^2 + (W_y^t)^2}}{H} - \frac{1}{\rho}\frac{\partial P_o}{\partial x} \qquad (12)$$

$$Y^t = \frac{\lambda\, W_y^t \sqrt{(W_x^t)^2 + (W_y^t)^2}}{H} - \frac{1}{\rho}\frac{\partial P_o}{\partial y} \qquad (13)$$

where P_o is atmospheric pressure

2. THE BASIC HYDRODYNAMICAL-NUMERICAL (HN) MODEL IN FORTRAN II

2.1 THE GRID NET

The grid net (staggered grid) is shown in Figure 1. It consists of three different sets of grid points: (1) the water elevation points (z) at the intersection of the grid; (2) the point for u-velocity component to the right and (3) the V-velocity component below the corresponding z point. Each of these three points has the same coordinate designation (n,m).

The geographical orientation of the grid is usually selected so that the x axis (m coordinate) is parallel to the entrance of the bay (or to the tidal input boundary). The relationship between the geographic coordinates and computation coordinates is shown in Figure 2. This orientation must be considered in the input of wind directions, which are introduced in computational grid coordinates. The direction of the current is printed (and plotted) in geographical direction.

The correction of the current direction to geographic direction is made within the program. The geographic direction is computed with the following formula:

$$\emptyset_{geogr.} = 90° - \emptyset_{comp.} + \emptyset_{rot} \qquad (14)$$

where $\emptyset_{geogr.}$ is the desired geographical direction; $\emptyset_{comp.}$ is the direction as computed in the program, and $\emptyset_{rot}$ is the rotation of the grid from its normal position, and this angle is introduced as an input card. The selection of grid size is described in the next section in connection with stability criteria.

The coastline must pass through u and v points and not through z points; thus, it becomes a "step line" along irregular coasts.

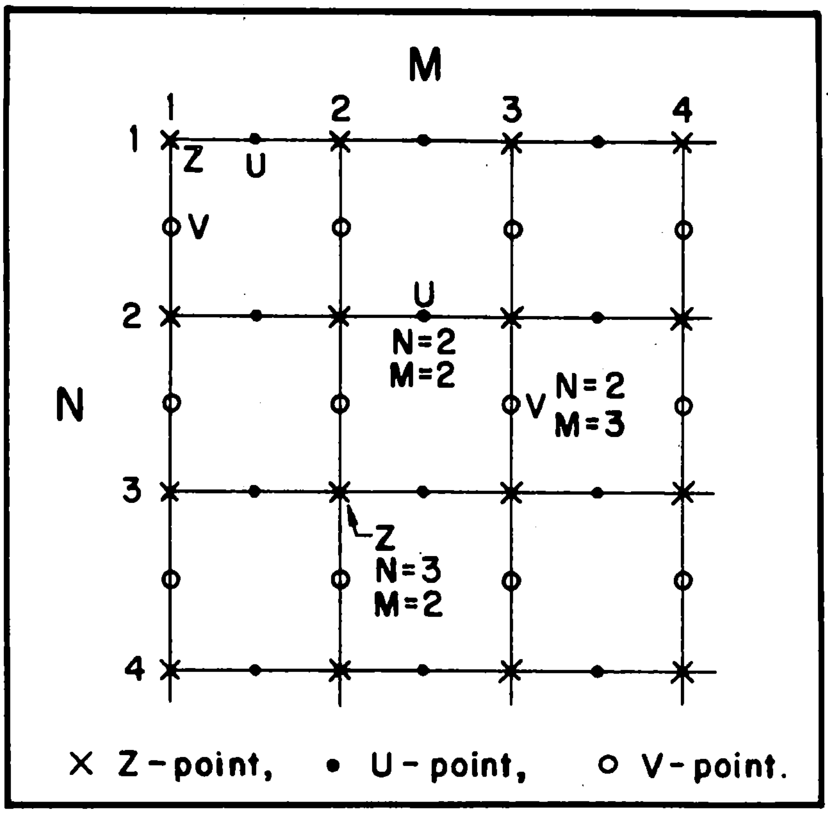

Figure 1. Scheme of the Grid Net

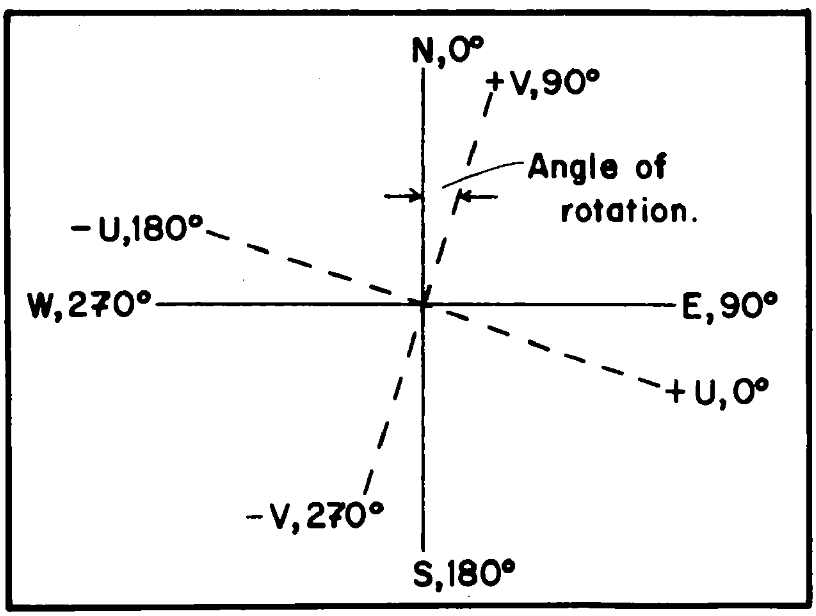

Figure 2. Relations Between the Geographic Coordinates and Computation Coordinates

2.5 INPUT OF DEPTHS AND DEFINITION OF BOUNDARIES

Three sets of depth values must be read into the computer. The first set is symbolic depth (sea and land table) at z points: the z points over land are designated 0; at the boundary on the coast, -1 (immediately outside the boundary line); over the water 1; and at the input boundaries -2 (and -3). The second and third sets of depth values are read at u and v points respectively. The depth values over the land are 0; at the boundary points, -1; and over the sea, the actual depth in centimeters.

All the boundary values should be defined if the program is to work correctly. Thus, the boundaries along the coast are defined as closed boundaries (i.e., no flow into or from the land). This is done by setting either u or v components at 0 on the coast (dependent on the direction of the coast) by defining the depths at u (respectively v) points as -1 for boundary identification purposes.

2.6 INPUT OF TIDES

The tidal heights are introduced at each time step at the second line from the open boundary. To avoid computational problems at coastal singularities, open boundaries should preferably be along the X axis (M) on the top of the grid (Y or N=2), or on the right hand side along (N) grid, whereby the land is located to the left of this input line.

Normally, four tidal components are used for the input of sea level elevation on the open boundary:

$$Z = A_1 \cos(\alpha_1 t - \kappa_1) + A_2 \cos(\alpha_2 t - \kappa_2) + A_3 \cos(\alpha_3 t - \kappa_3) + A_4 \cos(\alpha_4 t - \kappa_4) \quad (16)$$

In this equation, the speeds of the constituents (α) must be given in radians per second; the epoch of the constituent (κ) is in radians, and the amplitude of the constituent (A) is given in cm.

It should be noted that, by introducing only astronomical tidal amplitudes at the open boundary, the possible effects of atmospheric tidal components (caused by prevailing winds) and sea-level deformation by permanent currents are neglected. Furthermore, the tidal harmonic constants are derived from tide recordings near the coast and their extrapolation over deep water is uncertain. Experiments have shown, however, that the model corrects the unsatisfactory input assumptions quite rapidly and that reliable results can be extracted a few grid lengths inside the input boundary. In some experiments, the input tidal amplitudes have been made an inverse function of depth. This somewhat improves the results near the input boundary.

If the tidal constituents differ considerably at opposite ends of the input boundary (or along a coast), a possibility is provided in the program for the input of both sets of tidal harmonics and their interpolation.

2.7 INPUT OF PERMANENT (THERMOHALINE) CURRENT

There are several possibilities for input of permanent currents. The more correct (but time consuming) method is to introduce u and v components of permanent currents at the input boundary before tides are introduced. Computations are then made until stability is reached in the area (a few hours to 20 hours in real-time), at which time the u and v current field is extracted, stored and added later in each time step, while computing with tides and winds. This method is not shown in the program given in the Appendix. The second and fully satisfactory input of permanent currents is accomplished by prescribing a slope to an open boundary, or a longitudinal or transversal slope over the whole computational area.

2.8 INPUT OF WIND AND DERIVATION OF "LAYER AND A HALF" MODEL

The wind speed is given on the input card in m sec^{-1}. The wind direction must be given in grid coordinates and in the direction toward which the wind is blowing. Thus, a west wind is coming from -x toward +x and should be given an angle of 0°; a north wind has an angle of 270°; east wind, 180°, and south wind, 90°. If the coordinate system is turned (tilted) in relation to the geographic system, the angle of tilt must be added.

In the program given in Appendix C, the atmospheric pressure effects are neglected. This can be done if computations are made over a small area. In larger area computations, the effects of atmospheric pressure changes should be included in the program. For this purpose, it can be assumed that the sea level acts as an inverted barometer.

If the area on which the computation is made is deep or has thermal and haline stratification, it can be assumed that the wind current reaches only to the top of the thermocline. An empirical approach to this effect is provided in the program. An initial mixed layer depth must be specified and this initial depth is changed in inverse proportion to sea-level changes. The wind current is computed to the depth of the computed mixed layer. When the depth is shallower than the mixed layer, the wind current is computed to the bottom. This empirical approach is not fully satisfactory in all instances as the change of MLD set must be computed from divergence, which is derived from transport computations. This is properly done in multi-layer HN models (see Laevastu, 1974a)[7]. Furthermore, it should be borne in mind that the resulting computed current is vertically integrated over total depth.

---

[7] Laevastu, T., 1974a: A multilayer hydrodynamical numerical model (W. Hansen type) Model description and operating/running instructions. Report to EPA, Part 2.

2.9 TREATMENT OF TIDAL FLATS AND INUNDATIONS

The symbolic depth values at z points can be used for a variety of indicative purposes. In the computation for areas with tidal flats, the z points can be used for testing whether the flats are dry (e.g., using indicator -5). At the same time, indicators of the absence of water must be used in depth fields for u and v points. This is done in actual depth fields (HGU and HGV in the program) by assigning a fixed negative number (e.g., -3) to points that are temporarily dry. A test on the flow from one to another grid point must be performed in the case of extensive tidal flats.

If in storm-surge computations the extent of inundation of bars and low coastal areas is required, negative depth (i.e., height) values must be coded over land at u and v as well as at z points. In this case the -1 boundary values also get actual (but negative) height values.

3. TREATMENT OF OPEN BOUNDARIES

The HN model has been used with success for the computation of currents in straits and semi-closed seas with two or more entrances and along an open coast. In these models the principal entrance is the entrance from which the major tides move into the area. This principal entrance should preferably be parallel to the x coordinate at the top of the grid (usually along N=2 grid line).

Tidal heights are also introduced at the secondary open boundary (which in the case of straits is usually the last line of grid points along the x axis). Care should be taken that the tidal epochs along the second input line have a proper phase lag from the principal tidal entrance. The proper phase lag can usually be obtained through a tidal harmonic analysis from a location near the second entrance. If the phase lag is correctly computed, no discrepancies will occur between the z values at the second open boundary and the preceding grid point z values.

The main requirement in models with multiple open boundaries, which are applicable to open coasts, is that "non-input" boundaries (i.e., boundaries which are not prescribed in each time step) be kept open for in- and out-flow as dictated by currents and sea-level changes inside the area. The first successful method for treatment of the "along the coast" model was termed the "lazyman method." In this method, the tides were introduced at a boundary perpendicular to the coast so that the coast was located at the left side in the computational area. The tidal height was made inversely proportional to the depth along the input boundary. There was no special treatment of the seaward open boundary, except that the computations of u, v and z were extended toward this boundary as far as the finite difference approaches permitted (thus the "lazyman method") and the missing values of U at the right hand boundary and of V at the lower boundary were assumed to be the same as at the previous column (m-1) and row (n-1) respectively. The derivation of the open boundary values of z, u, and v by linear extrapolation (from the gradients between second and third grid points inside the area to the boundary) has not given fully satisfactory results.

4. COMPUTATION OF DIFFUSION AND DISPERSAL OF POLLUTANTS

As the HN models provide a current vector at each grid point at each time step, one is tempted to solve the interacting advection and diffusion problem using a time-stepping finite difference method. It should be borne in mind that diffusion (mixing) is irreversible, whereas advection is reversible. This model treats the dispersion of those substances which can be readily mixed in shallow water or in a surface mixed layer. If these problems can be properly solved by the HN model, no difficulties are foreseen in treating specific problems such as dispersion of oil, brackish water, etc. The following were considered in designing the dispersion portion of the model:

a. Vertical diffusion should be neglected (can be included in multi-layer models).
b. A basic Lagrangian approach should be used because of its application to computer solutions for grid arrays in finite-difference time-stepping.
c. Eulerian modifications should be used for instantaneous releases where the size of the blob is small in comparison to the mesh length.
d. Fickian diffusion with constant diffusity is used since advection and diffusion equations are solved separately in short, interlocking time-steps (similar to the "two-step method" of Schönfeld and Groen, 1961).[8]
e. The Austausch coefficient for eddy diffusion per se must be related to grid length.

---

[8]Schönfeld, J. C., and P. Groen, 1961: Mixing and exchange processes. Radioactive waste disposal into the sea. <u>IAEA Safety Series</u> 5:100-132.

Diffusion in water bodies has been presented in many formulas, a few of which are presented here (neglecting vertical diffusion).

The general diffusion formula is:

$$\frac{\partial^2 S}{\partial x^2} + \frac{\partial^2 S}{\partial y^2} - \frac{1}{A}\frac{\partial S}{\partial t} = 0$$

The basic dispersion formula:

$$\frac{\partial S}{\partial t} = Y - \frac{S}{n} - \frac{\partial}{\partial x}(U_x + pS_x) - \frac{\partial}{\partial y}(U_y + pS_y)$$

The Fickian equation:

$$\frac{\partial S}{\partial t} = Y - \frac{S}{n} - K\nabla^2 S - \frac{\partial}{\partial x}(S_u U) - \frac{\partial}{\partial y}(S_v V)$$

where:

Y - addition (release); n - decay; S - concentration; $pS_{x,y}$ - concentration velocity component; t- time; K (coefficient) = $\beta a v_r$ (a - depth; $v_r = u^2 + v^2$; $\beta = 0.003$); S_u, S_v - concentration gradients in u and v direction; and A - diffusion coefficient (Austausch coefficient).

The Lagrangian approach of diffusion in finite difference form was adopted by Wolff, Hansen and Joseph (1970):[9]

$$S^{t+\tau}_{n,m} = S^t_{n,m}\left(1 - \frac{4\tau A}{\ell^2}\right) + \frac{A\tau}{\ell^2}\left(S^t_{n-1,m} + S^t_{n+1,m} + S^t_{n,m-1} + S^t_{n,m+1} - 4S^t_{n,m}\right)$$

where

t is time; τ - time step; ℓ - grid size; S - concentration; A - diffusion coefficient; n and m are grid coordinates.

---

[9] Wolff, P. M., W. Hansen and J. Joseph, 1970: Investigation and prediction of dispersion of pollutants in the sea with Hydrodynamical-Numerical (HN) model. Marine Pollution and Sealife (Fishing News Ltd., London): 146-150.

The above finite equation deviates from the usual finite difference diffusion formula only in the addition of the last term (-4S), which makes the solution similar to the solution of the "Laplacian" ($K\nabla^2 S$). Secondly, the advection is computed linearly in finite difference form:

$$S^{t+\tau}_{n,m} = S^{t}_{n,m} - \tau \left| U^{t+\tau}_{n,m} \right| \left(\frac{S^{t}_{n,m} - S^{t}_{n,m\pm1}}{\ell} \right) - \tau \left| V^{t+\tau}_{n,m} \right| \left(\frac{S^{t}_{n,m} - S^{t}_{n\pm1,m}}{\ell} \right)$$

where: Sn,m-1 or n,m+1 (respectively n-1, n+1) are used, depending on the direction (sign) of U and V.

The Lagrangian approach used by Wolff, Hansen and Joseph (1970), though reproducing the diffusion process well, does not conserve absolutely the amount of the dispersing substance. The following modified formula was, however, found to be conservative, provided the proper A (Austausch coefficient) is chosen and corresponds to the chosen time step and grid size:

$$S^{t+\tau}_{n,m} = S^{t}_{n,m} - \frac{4\tau A}{\ell^2} S^{t}_{n,m} + \frac{\tau A}{\ell^2} \left(S^{t}_{n-1,m} + S^{t}_{n,m-1} + S^{t}_{n+1,m} + S^{t}_{n,m+1} - 4S^{t}_{n,m} \right) + \frac{\tau A}{\ell^2} \left(S^{t}_{n-1,m-1} + S^{t}_{n-1,m+1} + S^{t}_{(n+1,m-1)} + S^{t}_{(n+1,m+1)} \right)$$

It should be mentioned that the transport equation used in EPRF programs is very similar to the "upwind" difference scheme used in air pollution problems (Pandolfo et al, 1971).[10]

---

[10] Pandolfo, T. P., M. A. Atwater and G. E. Anderson, 1971: Prediction by numerical models of transport and diffusion in an urban boundary layer. Cent. Env. and Man, Inc., Hartford. Rpt. 4082: 139 pp.

This scheme requires that:

$$\frac{U\Delta t}{\Delta \ell} < 1.$$

The Austausch coefficient (A) is a function of grid size and time-step. In an experiment designed to investigate the conservation of diffusing substances, one of the main criteria was found to be a relationship between grid size and time-step. This relation of A to grid size with a 1,000 sec time-step is shown on Figure 3. The correct value of A is found from this graph and from the relation: $A = \frac{1,000}{\Delta t} A$ (graph). With a small time-step, the A approaches that found empirically by Okubo and Ozmidov (1970),[11] and Kullenberg (1972).[12] An idea of the proper A to be used in different grid sizes can be obtained also from the Joseph and Sendner (1958)[13] formulation. There is still some slight uncertainty about the dependence of the horizontal Austausch coefficient (K_h) on the length scale:

$$K_H = k_1 \times 10^{-3} \ \ell^{k2}$$

Kullenberg (1972) gives the values for the coefficients k_1=1.3 and k_2=1.31; whereas, Okubo (1971)[14] gives the corresponding values as 1.03 and 1.15. Both lines are shown in Figure 3.

---

[11] Okubo, A., and R. V. Ozmidov, 1970: Empirical dependence of the coefficient of horizontal turbulent diffusion in the ocean on the scale of the phenomenon in question. Izv. Atmosph. and Ocean. Phys. 6 (5):534-536.

[12] Kullenberg, G., 1972: Apparent horizontal diffusion in stratified vertical shear flow. Tellus, 24(1): 17-28.

[13] Joseph, J., and H. Sendner, 1958: Uber die horizontale Diffusion im Meere. Dtsch. Hydrogr. Zeitschr. 11(2): 49-77.

[14] Okubo, A., 1971: Oceanic diffusion diagrams. Deep Sea Res. 18, (8): 789-802.

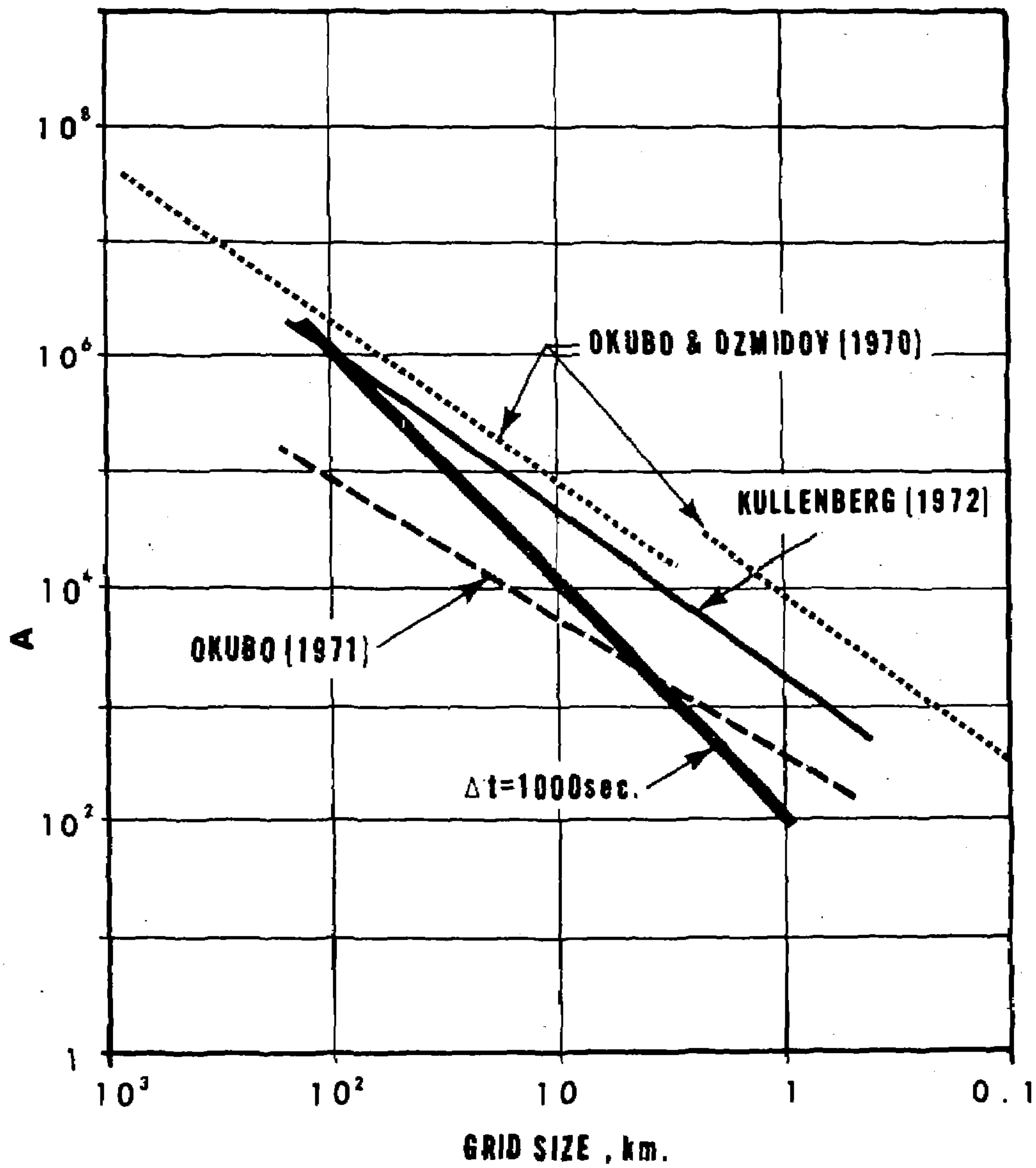

Figure 3. Dependence of Austausch Coefficient (A) on grid size (Δt=1000 sec)

If the dimensions of a diffusion blob are small in relation to grid size (in the case of an instantaneous release), it is preferable to use a Eulerian approach in the initial time stage. In this treatment, the center of the blob is advected by the currents and its position is recorded in every time step. The concentrations of neighboring grid points as well as the change of central concentration are computed with, for example, the Joseph-Sendner (1958) formula or other available formulations. This approach is repeated until a number of surrounding grid points have some concentrations above a predetermined minimum level, whereafter the Lagrangian treatment is continued.

It should be noted that if the grid size is large (e.g., 1 km or larger), the results of diffusion computations are not fully satisfactory as yet (see Laevastu et al, 1974c and 1974d).[15&16]

Decay (uptake by biota, remineralization, etc.) was not included in the tests reported above; however, its inclusion would not be difficult. Furthermore, "thermal pollution" can be treated by including the heat exchange terms in the model. (see Laevastu, and Hubert, 1965).[17]

---

[15] Laevastu, T., in collaboration with M. Clancy and A. Stroud, 1974: Currents, tides, and dispersal of pollutants in Lower Bay and Approaches to New York with fine and medium grid size hydrodynamical-numerical models. Report to EPA, Part 3.

[16] Laevastu, T. and R. Callaway in collaboration with A. Stroud and M. Clancy, 1974: 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. Report to EPA, Part 4.

[17] Laevastu, T., and W. E. Hubert, 1967: Analysis and prediction of the depth of the thermocline and near-surface thermal structure. Navy Weather Research Facility Rpt. 36-0667-128:90 pp.

Rammig (1971)[18] used a somewhat different approach for the computation of diffusion. His basic formula is:

$$\frac{\partial s}{\partial t} + U\frac{\partial s}{\partial x} + V\frac{\partial s}{\partial y} = A\frac{\partial^2 s}{\partial x^2} + B\frac{\partial^2 s}{\partial y^2}$$

assuming that A = B, his finite difference form (for a square grid) for the u component is:

$$S^{t+\tau}_{n,m} = S^{t}_{n,m} - \tau/\ell\; U^{t}_{n,m}\left(S^{t}_{n,m+1} - 2S^{t}_{n,m} + S^{t}_{n,m+1}\right) + 2A\tau/_{\ell}2$$

$$\left(S^{t}_{n,m-1} - 2S^{t}_{n,m} + S^{t}_{n,m+1}\right) + 2A\tau/_{\ell}2\left(S^{t}_{n-1,m} - 2S^{t}_{n,m} + S^{t}_{n+1,m}\right)$$

The V component is analogous to the above. This formula is also contained with blocked comments in column one in the program in Appendix C.

The stability criterion of this diffusion formula is:

$$\Delta t < \frac{\ell^2}{4A}$$

Rammig (1971) derived a new Austausch coefficient at each point, which is a function of depth and current speed:

$$A = 0.324\ H\ |u|$$

The values for A derived with the above formula might be appropriate in shallow, turbulent rivers, but tend to be somewhat too low for coastal waters. However, further tests on this formula with actual data are required.

---

[18]Rammig, H. G., 1971: Hydrodynamisch-Numerische Untersuchungen über Strömungsabhängige Horizontalausbreitungen von Stoffen in Modellflüssen mit Anwendungen auf die Elbe. Mitteil. Inst. Meeresk., Hamburg 14:64 pp and figures.

In case of computing the movement of drifting oil or other flotsam at the surface, it is necessary to give additional movement to this flotsam by wind (leeway), i.e., the oil at the surface is pushed with wind. The present model computes the leeway as a percentage of the wind. This percentage must be introduced with a control card.

In the program (see Appendix C) the initial position of oil or drifting objects is introduced with a control card. The drift due to tidal currents and the additional leeway caused by wind are computed in each time step. The position is printed out each time output is requested (e.g., hourly). Any number of initial positions can be included in the program.

5. REVIEW OF THE PROGRAM AND OUTPUT FORMATS

The computer program in FORTRAN is reproduced in Appendix C. The control statments are contained in the first subroutine, called Control Program. This is followed by Subroutine J01 which is for reading the input data (which are printed-out for checking purposes) and initialization of the fields. If no u, v and z values are read from previous computations, these fields and the intermediates (UPB,VPB) are initialized 0 over land, but over the water a low initial value (0.2) is read-in to facilitate the initial computations. The interpolation of tidal input is done next, if two sets of tidal constituents are used.

The main computations are done in J02, which is repeated once every time step. It consists essentially of input of tides, smoothing and averaging of u, v and z fields, their computation, and setting of boundary values.

If wind fields are specified, the subroutine of wind current computation is called in J02. If diffusion computations are desired, the diffusion and transport field (S(N,M)) is computed before the wind current component computation. However, additional drift of oil floating on the surface, as well as drifting object computation, is done after computing the wind current components.

The program contains an option for the computation of rest currents (e.g., after a full tidal cycle) and two sets of diffusion computation formulas. The formula from Rammig (1971) is blocked with comment statements in the appended program, as is the computation of wind currents with mixed layer depth.

The horizontal fields of water level and current speed and direction are printed out every one or two hours, or in any other desired time interval in Subroutine J03. Further, the data for selected special points are written on tape 2 every time field output is requested. The data at special points are printed out at the very end of the computation (J05).

A plotting program (not reproduced here) has been added to the HN model to allow the plotting of results on the Calcomp plotter (Bauer, 1969).[19] A series of statements are added to the basic program to capture the arrays of data that are required to plot the water level and velocity vectors. The plotting program is adequately documented by Bauer (1969) and is not described further in this document.

It should be pointed out that the model must run six to ten hours of real time (sometimes more, depending on the size of the area and grid length) before equilibrium is established and correct outputs can be taken.

The computations toward the coast (i.e., coast located to the right or in the lower part of the grid field) have not been successful yet. This is mainly due to the necessity of complex treatment of singularities (boundary problems) at the coast, which is not fully included in the present program.

---

[19] Bauer, R. A., 1969: Modification to program Hansen. Comp. Appl., Inc., San Diego, MS Rpt.

REPORTS IN THIS SERIES

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

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

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, FORMATS AND SET-UP OF TAPES

INPUT PARAMETERS AND DATA CARDS

<u>Card 1</u> Format 3A8

TIT

TIT Title of the computation

<u>Card 2</u>

JA, NE, ME, KO, IUE, KKE, NURU, KI, LE, LU, LI, LOP, LUN, LA, LIV, LEET

JA	Indicator, whether computation starts from initially 0 fields (JA = 0: Z = 0, U = 0, V = 0) or from the values of Z, U and V from tape). NOTE: If JA = 1, T on card 4 could be specified as the time of the computation of the input values or any other adjusted "real" time.
NE,ME	Delimeters of entire field (size of the computational grid).
KO	Number of points on the open boundary where tidal constituents 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
KI	Indicator, whether one or two sets of harmonic constants are used. In the latter case, the constants are interpolated linearly to each input point. KI = 0 - one set of constants; KI = 1 - two sets of constants.

LE Indicator, whether rest current computation over one tidal cycle (or any specified time interval) is desired. LE = 0 - no rest current computation; LE = 1 - rest current computation (specify beginning and end times on card 4).

LU Indicator and number of input points of permanent current. LU = 0 - no permanent current input; LU = 12 - permanent current input at 12 points (specify the N and M positions of input (NL, ML, card 21)).

LI Indicator for computation of diffusion and transport. LI = 0 - no diffusion computation; LI = 1 - diffusion computation.

LOP Indicator for plotting program. LOP = 0 - no plotting; LOP = 1 - plotting required (requires additional tape).

LUN Indicator for depth input. LUN = 0 - depths at U and V point different and read from the cards; LUN = 1 - depth for U and V points are identical (changes have to be made at the boundaries).

LA Number of points on second open boundary (NG and MG - i.e., the N and M of the points must be specified on card 17).

LIV Indicator for drift and oil drift calculations. LIV = 0 - no drift calculations; LIV = 1 - object drift calculation; LIV = 2 - oil drift calculation.

LEET Indicator, whether data to be put on tape and read from it.

Card 3 Format 9F8.3

G, ALPHA, RBETA, C1, TIL

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., 0.75)

CI	Wind indicator; 0 - no wind; 1 or 2 - wind (Specify beginning and end times on card 4). The number indicates the number of wind field changes.
TIL	Angle between true N and 90° of computational grid (+V).

Card 4 Format 9F8.0

DT, TE, TW, T1, T2, SI, T, TRB, TRE

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., 72000 sec).
SI	Time when wind stops (sec) (e.g., 180000)
T	Time (initialized 0. If previous computations made and z, u, v, read from tape, give the time previous computation ended).
TRB	Starting time for rest current computation. Used also for starting time for drift computations.
TRE	Ending time for rest current computation. Used also for ending time for drift computations.

Card 5 Format 9F8.0

T4

T4	Time when data on Z, U and V to be put on tape.

Card 6 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}) (not used in enclosed program).

R Friction coefficient (0.003).

ROL Density of the air (e.g., 1.1627×10^{-3}) (not used in enclosed program).

C Drag coefficient (e.g., 3.2×10^{-6})

<u>Card 7</u> Format 4E10.2

AUS

AUS Austausch coefficient (e.g., 1.0E5).

<u>Card 8</u> Format 24I3

NK, MK

NK N coordinate

MK M coordinate

Coordinates of the points at the first open boundary (input boundary).

<u>Card 9</u> (one or more cards Format 24I3

NZ, MZ

NZ N coordinate

MZ M coordinate

Coordinates of special selected points where special outputs are desired.

<u>Card 10</u> Format 24I3

NU, MU

NU N coordinate

MU M coordinate

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

Card 11 Format 10A5

V(I) Names of the selected special points.

Card 12 Format 9F8.2

A(I)

A(I) Interval (in sec) between wind speed and direction change. If only one wind speed is used, then duration of the wind (SI - TW) (For the program in Appendix C the time intervals between the wind changes must be equal).

A(2) Wind speed (m sec^{-1})

A(3) Wind direction (in degrees in computation coordintes).

Card 13 Format 14F5.1

H

H Amplitudes of tidal constituents

Card 14

B

B Amplitudes of tidal constituents at second boundary. Used only if two sets of tidal constituents are used and interpolated, or tidal input is made at a second boundary.

Card 15 Format 7F10.9

AL

AL Speeds of the tidal constituents (α) in radians per second.

Card 16 Format 7F10.9

CAPA

CAPA Phase angles (κ) of the tidal constituents.

Card 17

CAPB

CAPB — Phase angles (κ) of the tidal constituents. Used only if two sets of tidal constituents are used and interpolated, or tidal input is made at a second boundary.

Card 18 (one or more cards) Format 24I3

NG, MG

NG — N coordinate

MG — M coordinate

Coordinates of the points on the second open boundary. Used only if two or more open boundaries are used in the area.

Card 19 (several cards) Format 12F6.0

HTZ

Symbolic depths at the z points

Card 20 (several cards) Format 12F6.0

HTU

Depths at u points (cm)

Card 21 (several cards) Format 12F6.0

HTV

Depths at v points (cm)

Card 22 Format 24I3

NL, ML

N and M coordinates of the input for permanent current. Used only if permanent current input is included.

Card 23 (one or more) Format 14F5.1

PERU

PERU — U component of permanent current. Used only if permanent current input is included.

<u>Card 24</u> (one or more) Format 14F5.1

PERV

PERV V component of permanent current

<u>Card 25</u> Format 9F8.2

POSN, POSM, DRI

POSN N coordinate of the initial position of drifting object.

POSM M coordinate of the initial position of drifting object.

DRI Percentage of wind speed for additional drift (leeway)

<u>Card 26, 27</u> Plotting parameters. See special publication. Used only if plotting program is included.

SET-UP OF TAPES

The input parameters are read into the computer after the program, usually from cards (or from tape). Input tape 50 on FORTRAN 63 CO-OP monitor indicates card reader. For running the program on the CDC 6500, no tapes are required if no plotting is desired. The program and data input cards are read with the card reader. A disk is assigned to tape 2 (intermediate tape for writing special point outputs). Outputs go directly to the printer. In case outputs are required for plotting, one tape is required at the end of the program.

For running of the program on the CDC 1604 or other smaller computer (with FORTRAN 60 or FORTRAN 63), the following tape units are required:

Tape 1 is the FORTRAN system tape;

Tape 2 is the intermediate outputs of special points;

Tape 3 can be for input of the program and data if no card reader is used and may be used also for outputs for plotting purposes;

Tape 4 can be used as the printer if no printer is available;

Tape 5 can be used for input and output of intermediate data on u, v, and z.

APPENDIX B

ABBREVIATIONS AND PARAMETERS USED IN THE PROGRAM

Note: Symbols marked with an * are further explained in Appendix A

A(I)*	Characteristics of the wind field
AH(I,J)	Amplitudes of tidal constituents at individual input points
AINC(I)	Intermediate parameter in interpolation of tidal constituents
AL(I)*	Speed of tidal constituents
ALPHA*	Smoothing parameter
ANG(N,M)	Current direction
ARG(I,J)	$(\alpha t - \kappa)$ Argument of tidal constituent
AUS*	Austausch coefficient
A1 to A5	Intermediate parameters (see statement 60-65)
B(I)*	Amplitudes of second set of tidal constituents
BETA	1 - ALPHA
B1 to B5	Intermediate parameters (see statements 412-415)
C*	Drag coefficient (3.2×10^{-6})
CAINC, CAINCU	Intermediate parameters in interpolation of tidal constituents
CAPA(I)*	κ of first set of tidal constituents
CAPAP(I,J)	κ of tidal constituents at individual input points
CAPB(I)*	κ of second set of tidal constituents
CNCU	Intermediate parameter in interpolation of tidal constituents
CW	A counter
C1*	Wind indicator
C3	Intermediate parameter
DL*	Half grid size
DRI*	Leeway (drift speed in % of wind speed)

DT*	Half time step
ETC* } ETH* }	Plotting parameters
H(I)*	Amplitudes of first set of tidal constituents
HGU(N,M) } HGV(N,M) }	Actual depths at u and v points
HT	Counter (in plotting)
HTU(N,M)* } HTV(N,M)* }	Water depths at u and v points
HTZ(N,M)*	Symbolic depths at z points
HMLD	Mixed layer depth
I	A counter, delimeter
IU	Number of wind fields
IUE*	Twice the number of wind fields
J	A counter
JA*	Initiation indicator
K	A counter
KI*	Tidal input indicator
KKE*	Number of wind field characteristics
KO*	Number of points at the open boundary
L	A counter
LA*	Number of points on second input boundary
LE*	Indicator for rest current computations
LEET*	Indicator for writing intermediate output on the tape
LI*	Indicator for computation of diffusion and transport
LIV*	Indicator for drift and oil drift calculations
LOP*	Indicator for plotting program
LU*	Indicator for permanent current input
LUN*	Indicator for simplified depth input
M*	Grid index (x)
MD*	Coordinate for drift computation

ME*	Delimeter of grid size (x)
MEH	ME-1
MG*	M coordinate of input points at second open boundary
MK*	M coordinate of input points at first open boundary
ML*	M coordinates of input points of permanent current
MU*	Wind field delimeter
MUM	A counter
MZ*	M coordinate for output at special points
M1, M2	Indices
N*	Grid index (y)
ND*	Coordinate for drift computation
NE*	Delimeter of grid size (y)
NEH	NE-1
NG*	N coordinates of input points at second open boundary
NK*	N coordinates of input points at first open boundary
NL*	N coordinates of input points of permanent current
NON	A counter
NU*	Wind field delimeter
NURU*	Number of special output points
NZ*	N coordinate of output at special points
N1,N2	Indices
OTC(I)	Intermediate for meteorological overtides input at the coast
PERU(I)*, PERV(I)*	u and v components of permanent current
POSM*, POSN*	M and N coordinates of the drifting object

PREC(I,J)	Intermediate for interpolation of tidal input
PROD	Percentage of wind speed for additional drift of oil
R*	Friction coefficient
RAD(N,M)	Current speed (resultant)
RBETA*	Coefficient of geostrophic wind (0.75)
ROL*	Density of the air (1.1627×10^{-3})
ROOT	Intermediate (square root)
S(N,M)	Concentration of diffusing substance
SI*	Time when wind stops
SIGMA*	Angular velocity of M_2 tide
SSH, SSV	Intermediates in diffusion computation
STH*, STV*	Plotting parameters
T*	Time counter
TAR(I)	Intermediate for input of meteorological overtides at the coast
TE*	Length of computation
TIC	A counter
TIL*	Tilt of the grid
TINCH*, TINCV*	Plotting parameters
TIT*	Title
TRB*	Starting time for rest current computation (sec)
TRE*	Ending time for rest current computation (sec)
TW*	Time when wind starts (sec)
T1*	Time interval between printouts
T2*	Field output counter
T4*	Time when intermediate u, v and z values to be put on tape
U(N,M)	U component of current
UDIS	Intermediate for drift computation

UML, UMR	Intermediates for oil drift computation
UPB(N,M)	Intermediate field for smoothing and averaging
US(N,M)	U component of rest current
V(N,M)	V component of current
VALE, VALO, VARI, VAUP	Intermediates for smoothing
VDIS	Intermediate for drift computation
VEC	Plotting parameter
VLE, VLL, VLO, VLR	Intermediates for averaging
VMB, VMO	Intermediates for oil drift computation
VPB(N,M)	Intermediate field for smoothing and averaging
VRI	Intermediate for averaging
VS(N,M)	V component of rest current
VUL, VUP, VUR	Intermediates for averaging
V1*	Names of special output points
V2 to V4	Intermediates for output at special points
XK(N,M), YK(N,M)	U and v components of wind current
Y1*	Decay of pollutant
Z(N,M)	Sea level

APPENDIX C

PROGRAM NYBIGHT LISTING

```
      PROGRAMNYBIGHT(INPUT,OUTPUT,TAPE50=INPUT,TAPE2,TAPE3,TAPE5)
      DIMENSION HTZ(42,47),HTU(42,47),HTV(42,47),Z(42,47),U(42,47),V(42,
     147),US(42,47),VS(42,47),RAD(42,47),ANG(42,47),XK(42,47),YK(42,47),
     2     NU(6),MU(6),NK(85),MK(85),AH(5,85),CAPAP(5,85),PREC(5,85),ARG
     3(5,85),PERU(85),PERV(85),NL(85),ML(85),NZ(10),MZ(10),V1(10),UPB(42
     4,47),VPB(42,47),S(42,47),A(85)        ,V2(10),V3(10),V4(10),HGU(42,4
     57),HGV(42,47),H(5),B(5),AL(5),CAPA(5),CAPB(5),AINC(5),CAINC(5),
     6                  NG(40),MG(40),TIT(3),AUS(5),TAR(85),OTC(85)
     7,USR(42,47),VSR(42,47)
      COMMON HTZ,HTU,HTV,Z,U,V,US,VS,RAD,ANG,XK,YK,  NU,MU,NK,MK,AH,CAPA
     1P,PREC,ARG,PERU,PERV,NL,ML,NZ,MZ,V1,UPB,VPB,S,A    ,V2,V3,V4,HGU,HG
     2V,H,B,AL,CAPA,CAPB,AINC,CAINC,NG,MG,TIT,AUS,TAR,OTC,USR,VSR,
     3T1,T2,T4,TE,TW,A1,A2,A3,A4,DT,BETA,ALPHA,F,R,DL,G,C,C3,NE,ME,NEH,M
     4EH,JA,IUE,KKE,NURU,SI,T,RBETA,C1,SIGMA,ROL,A5,RU,RV,KO,LI,LE,LU,KI
     5,LOP,TRB,TRE,TIC,             LUN,LA,TIL,LIV,LEET
     6,POSN,POSM,DRI,STV,ETV,TINCV,STH,ETC,TINCH,CNCU,CAINCU
     7,AA1,BB2,CC3,TSC,DL2
     8,AA2,AA3,AA4,BB3,BB4,BB5,BB6,BB7,BB8,CC4,CC5,CC6,CC7,CC8
C     CONTROLL PROGRAM
    4 REWIND 2
    5 CALL JOB1
 1007 CONTINUE
    6 CALL JOB3
 1009 CONTINUE
 1010 IF(LOP) 7,7,1011
 1011 CALL JOB7
      IF(T-TE)7,240,240
  240 IF(LOP)1016,1016,241
  241 IF(LOP-5)1013,1016,1016
    7 T2=T2+T1
    8 T=T+A1
    9 CALL JOB2
   10 IF(T-TE)11,1012,1012
   11 IF(ABSF(T-T2)-A1/2.)6,6,12
   12 GO TO 8
 1012 IF(LOP)13,13,1013
 1013 LOP=5
   13 CALL JOB3
 1014 IF(LOP)1016,1016,1015
 1015 CALL JOB7
 1016 END FILE 2
   14 REWIND 2
   15 CALL JOB5
 1017 IF(LOP)16,16,245
  245 END FILE 3
```

Control Program
JOB6 and JOB7 are
for a special plot
program, as is Tape 3

```
      REWIND 3
 1018 CONTINUE
   16 STOP
      END
      SUBROUTINE JOB1
      DIMENSION HTZ(42,47),HTU(42,47),HTV(42,47),Z(42,47),U(42,47),V(42,
     147),US(42,47),VS(42,47),RAD(42,47),ANG(42,47),XK(42,47),YK(42,47),


     2     NU(6),MU(6),NK(85),MK(85),AH(5,85),CAPAP(5,85),PREC(5,85),ARG
     3(5,85),PERU(85),PERV(85),NL(85),ML(85),NZ(10),MZ(10),V1(10),UPB(42
     4,47),VPB(42,47),S(42,47),A(85)      ,V2(10),V3(10),V4(10),HGU(42,4
     57),HGV(42,47),H(5),B(5),AL(5),CAPA(5),CAPB(5),AINC(5),CAINC(5),
     6              NG(40),MG(40),TIT(3),AUS(5),TAR(85),OTC(85)
     7,USR(42,47),VSR(42,47)
      COMMON HTZ,HTU,HTV,Z,U,V,US,VS,RAD,ANG,XK,YK,  NU,MU,NK,MK,AH,CAPA
     1P,PREC,ARG,PERU,PERV,NL,ML,NZ,MZ,V1,UPB,VPB,S,A    ,V2,V3,V4,HGU,HG
     2V,H,B,AL,CAPA,CAPB,AINC,CAINC,NG,MG,TIT,AUS,TAR,OTC,USR,VSR,
     3T1,T2,T4,TE,TW,A1,A2,A3,A4,DT,BETA,ALPHA,F,R,DL,G,C,C3,NE,ME,NEH,M
     4EH,JA,IUE,KKE,NURU,SI,T,RBETA,C1,SIGMA,ROL,A5,RU,RV,KO,LI,LE,LU,KI
     5,LOP,TRB,TRE,TIC,            LUN,LA,TIL,LIV,LEET
     6,POSN,POSM,DRI,STV,ETV,TINCV,STH,ETC,TINCH,CNCU,CAINCU
     7,AA1,BB2,CC3,TSC,DL2
     8,AA2,AA3,AA4,BB3,BB4,BB5,BB6,BB7,BB8,CC4,CC5,CC6,CC7,CC8
C     READING OF VALUES AND INITIATION
      ETH = ETC
   29 CONTINUE
   30 FORMAT(24I3)
   31 FORMAT(9F8.3)
   32 FORMAT(9F8.0)
   33 FORMAT(6E12.4)
   34 FORMAT(15A5)
   35 FORMAT(9F8.2)
   36 FORMAT(14F5.1)
   37 FORMAT(7F10.9)
   38 FORMAT(12F6.0)
   39 FORMAT(3A8)
   55 FORMAT(E10.2)
   56 READ (50,39)(TIT(I),I=1,3)
   40 READ (50,30)JA,NE,ME,KO,IUE,KKE,NURU,KI,LE,LU,LI,LOP
     1,LA,LUN,LIV,LEET
   41 READ (50,31)G,ALPHA,RBETA,C1,TIL
   42 READ (50,32)DT,TE,TW,T1,T2,SI,T,TRB,TRE
 1006 IF(LEET) 43,43,1008
 1008 READ (50,32)T4
   43 READ (50,33)DL,F,SIGMA,R,ROL,C
   57 READ (50,55)AUS(1)
   44 READ (50,30)(NK(I),MK(I),I=1,KO)
   45 READ (50,30)(NZ(I),MZ(I),I=1,NURU)
   46 READ (50,30)(NU(I),MU(I),I=1,IUE)
```

```
  47 READ (50,34)(V1(I),I=1,NURU)
1030 NON=1
1036 CW = 1.
  48 READ (50,35)(A(I),I=NON,KKE)
1032 CW=CW+1.
1033 NON=KKE+1
1034 KKE=KKE+NON-1
1037 IF(C1-CW) 49,48,48
  49 READ (50,36)(H(I),I=1,5)
  58 IF(KI)50,50,59
  59 READ (50,36)(B(I),I=1,5)
  50 READ (50,37)(AL(I),I=1,5)
  51 READ (50,37)(CAPA(I),I=1,5)
  69 IF(KI) 1020,1020,70
  70 READ (50,37)(CAPB(I),I=1,5)
1020 IF(LA)52,52,1021
1021 READ (50,30)(NG(I),MG(I),I=1,LA)
  52 READ (50,38)((HTZ(N,M),M=1,ME),N=1,NE)
  68 MEH=ME-1
1022 IF(LUN) 53,53,54
  53 READ (50,38)((HTU(N,M),M=1,ME),N=1,NE)




  54 READ (50,38)((HTV(N,M),M=1,ME),N=1,NE)
  71 IF(LU) 92,92,72
  72 READ (50,30)(NL(I),ML(I),I=1,LU)
  73 READ (50,36)(PERU(I),I=1,LU)
  74 READ (50,36)(PERV(I),I=1,LU)
  92 IF(JA) 1040,1040,93
  93 READ (5,31)((Z(N,M),M=1,ME),N=1,NE)
  94 READ (5,31)((U(N,M),M=1,ME),N=1,NE)
  98 READ (5,31)((V(N,M),M=1,ME),N=1,NE)
1098 READ (5,32)T
1040 IF(LIV)1005,1005,1041
1041 IF(LIV-3) 1042,1005,1042
1042 READ (50,35)POSN,POSM,DRI
1005 IF(LOP)60,60,200
 200 CALL JOB6
     DO 9967 N = 1,41
     HTZ(N,2)=-2.
9967 HTZ(N,1)=0.
     HTV(30,17)=-1
     DO 2031 N = 1,NE
     DO 2031 M = 1,ME
     IF(HTU(N,M).GT.0) HTU(N,M) = HTU(N,M) * 185.201
     IF(HTV(N,M).GT.0) HTV(N,M) = HTV(N,M) * 185.201
     IF(HTU(N,M).LE.0.) GO TO 2033
     IF(HTU(N,M)-200.)2032,2032,2033
```

Reading of input data

Calling of initial plotting program; setting an input boundary

Converting of depths from fathoms to cm and setting of minimum depth.

```
2032 HTU(N,M) = 200.
2033 IF(HTV(N,M).LE.0.) GO TO 2031
     IF(HTV(N,M)-200.) 2034,2034,2031
2034 HTV(N,M) = 200.
2031 CONTINUE
  60 BETA=(1.-ALPHA)/4.
```

```
  61 A1=2.*DT
  62 A2=F*A1
  63 A3=R*A1
  64 A4=DT/DL
  65 A5=G*A4
  66 C3=C*A1*10000.
  67 NEH=NE-1
```

Setting of some intermediate parameters.

```
1052 KKE = (IUE/2) * 3
  99 TIC=0.
     TSC = 0.
     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.
```

Zeroing of transport arrays.

```
  75 DO 100 N=1,NE
  76 DO 100 M=1,ME
  77 IF(HTZ(N,M))78,78,80
  78 Z(N,M)=0.
```

```
     VPB(N,M)=0.
  79 GO TO 81
  80 Z(N,M)=0.2
  81 IF(HTU(N,M)) 82,82,84
  82 U(N,M)=0.
     UPB(N,M)=0.
  83 GO TO 85
  84 U(N,M)=0.2
     UPB(N,M)=0.2
  85 IF(HTV(N,M))86,86,88
  86 V(N,M)=0.
     VPB(N,M)=0.
  87 GO TO 95
  88 V(N,M)=0.2
```

Initialization of arrays

```
      VPB(N,M)=0.2
   95 XK(N,M)=0.
   96 YK(N,M)=0.
      S(N,M)=0.
      US(N,M) = 0.
      VS(N,M) = 0.
      USR(N,M)=0.
      VSR(N,M)=0.
  100 CONTINUE
 1060 IF(LUN)101,101,1061
 1061 DO 1064 N=1,NE
 1062 DO 1064 M=1,ME
 1063 HTU(N,M)=HTV(N,M)
 1064 CONTINUE
  101 IF(KI)120,120,102
  102 DO 107 I=1,5
  103 AINC(I)=(H(I)-B(I))/(KO-1)
 1101 IF(CAPA(I)-90.)1102,1102,1103
 1102 IF(CAPB(I)-270.)1103,1103,1105
 1103 IF(CAPA(I)-270.)104,104,1104
 1104 IF(CAPB(I)-90.)1107,1107,104
 1105 CAPA(I)=CAPA(I)+360.
 1106 GO TO 104
 1107 CAPB(I)=CAPB(I)+360.
  104 CAINC(I)=(CAPA(I)-CAPB(I))/(KO-1)
  107 CONTINUE
  108 DO 114 I = 1,5
 1065 CNCU=0.
 1066 CAINCU=0.
  109 DO 114 J = 1,KO
  110 AH(I,J) = H(I) - CNCU
  111 CNCU = CNCU + AINC(I)
  112 CAPAP(I,J) = CAPA(I) - CAINCU
 1108 IF(CAPAP(I,J)-360.)1111,1111,1109
 1109 CAPAP(I,J)=CAPAP(I,J)-360.
 1110 GO TO 113
 1111 IF(CAPAP(I,J))1112,1112,113
 1112 CAPAP(I,J)=CAPAP(I,J)+360.
  113 CAINCU = CAINCU + CAINC(I)
  114 CONTINUE
  115 GO TO 125
  120 DO 124 J=1,KO
  121 DO 124 I=1,5
  122 AH(I,J)=H(I)
  123 CAPAP(I,J)=CAPA(I)
  124 CONTINUE
  125 CONTINUE
 1130 PRINT 100,(TIT(I),I=1,3)
```

Special setting of depths if only one set was read in.

Interpolation of tidal input parameters. (If required)

```
1131 PRINT 181,DT,DL
1132 PRINT 182,F,AUS(1)
1133 PRINT 183,ALPHA,C1
1134 PRINT 184,TE,T1,TW,SI
1114 PRINT 1141,T2,TRB,TRE
1115 PRINT 1142,TIL
1116 PRINT 1143,KI,LE,LU,LI,LOP,LUN,LA,LIV
1117 IF(LIV)1120,1120,1118
1118 IF(LIV-2)1119,1120,1119
1119 PRINT 1144, POSN,POSM,DRI
1120 CONTINUE
1135 PRINT 185,(NK(I),MK(I),I=1,KO)
1136 PRINT 186,H(1),H(2),H(3),H(4),B(1),B(2),B(3),B(4)
1137 PRINT 187,AL(1),AL(2),AL(3),AL(4)
1138 PRINT 188,CAPA(1),CAPA(2),CAPA(3),CAPA(4),CAPB(1),CAPB(2),CAPB(3),
    1CAPB(4)
1140 MUM = XFIXF(C1*3.)
1139 PRINT 189,(A(I),I=1,MUM)
1121 IF(LOP)1124,1124,1122
1122 PRINT 1145,STV,ETV,TINCV
1123 PRINT 1145,STH,ETH,TINCH
1124 CONTINUE
1125 IF(LU)1128,1128,1126
1126 PRINT 1146,(NL(I),ML(I),I=1,LU)
1127 PRINT 1147, (PERU(I),PERV(I),I=1,LU)
1128 CONTINUE
 180 FORMAT(1H1,35X,3A8///)
 181 FORMAT(5X,19HHALF TIME STEP DT =,F5.0,5H  SEC,29X,20HHALF GRID SIZ
    1E DL = ,E11.4,4H  CM//)
 182 FORMAT(5X,21HCORIOLIS PARAMETER =,E12.4,5X,16HAUSTAUSCH COEF.=,
    1E10.2//)
 183 FORMAT(5X,27HSMOOTHING PARAMETER ALPHA =,F6.3,20X,19HWIND INDICATO
    1R C1 =,F8.3//)
 184 FORMAT(5X,13HCOMP. LENGTH ,F8.0,5X,15HOUTPUT INTERV. ,F8.0,5X,15HW
    1IN  STARTS AT ,F8.0,5X,11HWIND STOPS ,F8.0,6HSECONS,//)
 185 FORMAT(5X,49H(N,M) COORDINATES OF INPUT POINTS AT THE BOUNDARY ,//
    15X,12(1H(,I2,1H,,I2,1H),2X)//5X,12(1H(,I2,1H,,I2,1H),2X)//5X,5(1H(
    2,I2,1H,,I2,1H),2X)//)
 186 FORMAT(5X,29HAMPLITUDES OF TIDE CONSITUTES,5X,8F10,1//)
 187 FORMAT(5X,20HSEEDS OF TIDE CONST.,5X,8F10.7,//)
 188 FORMAT(5X,21HKAPPAS OF TIDE CONST.,5X,8F10,7,//)
 189 FORMAT(5X,25HWIND.INTERV.,SPEED,DIR.  ,9F8,2,//5X,12F8,2,//)
1141 FORMAT(5X,25HWIND BEG.,DRIFT BEG.,ENDS,3X,3F8,2,//)
1142 FORMAT(5X,12HTILT OF GRID,5X,F4.0,//)
1143 FORMAT(5X,3HKI=,I3,3HLE=,I3,3HLU=,I3,3HLI=,I3,4HLOP=,I3,4HLUN=,I3,
    13HLA=,I3,4HLIV=,I3,//)
1162 FORMAT(///8X,15I8)
1163 FORMAT(1X,I8,15F8.0)
1144 FORMAT(5X,25HSTARTING POS. OF DRIFT,N=,F6.2,2X,2HM=,F6.2,18HPERCEN
    1TAGE OF WIND,F6.2,//)
1145 FORMAT((10X,15HPLOTTING PARAM.,3F10.0/))
1146 FORMAT(5X,24HCOORD.OF PERM.CURR.INPUT,/5X,12(1H(,I2,1H,,I2,1H),2X)
```

Printing of input for checking. (statements 1131–1127)

```
   1/5X,12(1H(,I2,1H,,I2,1H),2X)/)
1147 FORMAT(5X,26HU AND V COMP.OF PERM.CURR,,/5X,8(1H(,F4.1,1H,,F4.1,1H
    1),2X)/5X,8(1H(,F4.1,1H,,F4.1,1H),2X)/5X,8(1H(,F4.1,1H,,F4.1,1H),2X
    2)/)
 140 PRINT 160,(N,N=1,16)
 141 PRINT 161,(N,(HTZ(N,M),M=1,16),N=1,NE)
 142 PRINT 162,(N,N=17,32)
 143 PRINT 163,(N,(HTZ(N,M),M=17,32),N=1,NE)
     PRINT1162,(N,N=33,47)
     PRINT1163,(N,(HTZ(N,M),M=33,47),N=1,NE)
 144 PRINT 164,(N,N=1,16)
 145 PRINT 161,(N,(HTU(N,M),M=1,16),N=1,NE)
 146 PRINT 162,(N, N=17,32)
 147 PRINT 163,(N,(HTU(N,M),M=17,32),N=1,NE)
     PRINT1162,(N,N=33,47)
     PRINT1163,(N,(HTU(N,M),M=33,47),N=1,NE)
 148 PRINT 165,(N,N=1,16)
 149 PRINT 161,(N,(HTV(N,M),M=1,16),N=1,NE)
 150 PRINT 162,(N,N=17,32)
 151 PRINT 163,(N,(HTV(N,M),M=17,32),N=1,NE)
     PRINT1162,(N,N=33,47)
     PRINT1163,(N,(HTV(N,M),M=33,47),N=1,NE)
```

Printing of input for checking.

```
     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
```

Setting of maximum depth

```
 160 FORMAT(1H1,16X,36HSYMBOLIC WATER DEPTH AT THE Z POINTS//8X,16I8/)
 161 FORMAT(I8,16F8.0)
 162 FORMAT(///8X,16I8)
 163 FORMAT(I8,16F8.0)
 164 FORMAT(1H1,16X,32HWATER DEPTH AT THE U POINTS (CM)//8X,16I8/)
 165 FORMAT(1H1,16X,32HWATER DEPTH AT THE V POINTS (CM)//8X,16I8/)
 170 DO 175 N=1,NE
 171 DO 175 M=1,ME
 173 HGU(N,M)=HTU(N,M)
 174 HGV(N,M)=HTV(N,M)
```

Initial setting of variable depth

```
 175 CONTINUE
1175 CONTINUE
     RETURN
     END
     SUBROUTINE JOB2
     DIMENSION HTZ(42,47),HTU(42,47),HTV(42,47),Z(42,47),U(42,47),V(42,
    147),US(42,47),VS(42,47),RAD(42,47),ANG(42,47),XK(42,47),YK(42,47),
    2     NU(6),MU(6),NK(85),MK(85),AH(5,85),CAPAP(5,85),PREC(5,85),ARG
    3(5,85),PERU(85),PERV(85),NL(85),ML(85),NZ(10),MZ(10),V1(10),UPB(42
    4,47),VPB(42,47),S(42,47),A(85)      ,V2(10),V3(10),V4(10),HGU(42,4
    57),HGV(42,47),H(5),B(5),AL(5),CAPA(5),CAPB(5),AINC(5),CAINC(5),
    6                NG(40),MG(40),TIT(3),AUS(5),TAR(85),OTC(85)
    7,USR(42,47),VSR(42,47)
```

```
      COMMON HTZ,HTU,HTV,Z,U,V,US,VS,RAD,ANG,XK,YK,  NU,MU,NK,MK,AH,CAPA
     1P,PREC,ARG,PERU,PERV,NL,ML,NZ,MZ,V1,UPB,VPB,S,A    ,V2,V3,V4,HGU,HG
     2V,H,B,AL,CAPA,CAPB,AINC,CAINC,NG,MG,TIT,AUS,TAR,OTC,USR,VSR,
     3T1,T2,T4,TE,TW,A1,A2,A3,A4,DT,BETA,ALPHA,F,R,DL,G,C,C3,NE,ME,NEH,M
     4EH,JA,IUE,KKE,NURU,SI,T,RBETA,C1,SIGMA,ROL,A5,RU,RV,KO,LI,LE,LU,KI
     5,LOP,TRB,TRE,TIC,          LUN,LA,TIL,LIV,LEET
     6,POSN,POSM,DRI,STV,ETV,TINCV,STH,ETC,TINCH,CNCU,CAINCU
     7,AA1,BB2,CC3,TSC,DL2
     8,AA2,AA3,AA4,BB3,BB4,BB5,BB6,BB7,BB8,CC4,CC5,CC6,CC7,CC8
C     COMPUTATIONS
  309 CONTINUE
      TD = 104400. + T
      TG=TD-6100.
      TGT=TG-TD
      DO 384 N=2,40
      RN=N
      D=(RN-2.)/39.                                      Tidal input
      PY=FLOAT(N)
      Z(N,2)=H(1)*COS(AL(1)*(TD+D*TGT)-CAPA(1))+H(2)*COS(AL(2)*(TD+D*TGT
     1)-CAPA(2))+H(3)*COS(AL(3)*(TD+D*TGT)-CAPA(3))+H(4)*COS(AL(4)*(TD+D
     2*TGT)-CAPA(4))+H(5)*COS(AL(5)*(TD+D*TGT)-CAPA(5))
      Z(N,2) = 0.86*Z(N,2)
  384 CONTINUE
      DO 386 N=2,40
      PZ=FLOAT(N)
      Z(N,2)=Z(N,2)-4./PZ
  386 CONTINUE                                           Special boundary setting
      DO 1215 M=2,44
      PZ=FLOAT(M)
      Z(2,M)=Z(2,M)-0.1*PZ
      Z(1,M)=Z(2,M)-0.08*PZ
 1215 CONTINUE
  390 DO 394 N=1,NE
  391 DO 394 M=1,ME
  392 UPB(N,M)=0.
  393 VPB(N,M)=0.
  394 CONTINUE
  500 DO 523 N=1,NE
  501 DO 523 M=1,ME
  502 IF(HTZ(N,M))523,523,503
  503 IF(1-N)504,507,504
  504 IF(HTZ(N-1,M))505,507,505
  505 VAUP=Z(N-1,M)
  506 GO TO 508
  507 VAUP=Z(N,M)                                        Computation of Z̄
  508 IF(NE-N)509,512,509
  509 IF(HTZ(N+1,M))510,512,510
  510 VALO=Z(N+1,M)
  511 GO TO 513
  512 VALO=Z(N,M)
  513 IF(1-M)514,516,514
```

```
 514 IF(HTZ(N,M-1))1514,516,1514
1514 VALE=Z(N,M-1)
 515 GO TO 517
 516 VALE=Z(N,M)
 517 IF(ME-M)518,521,518
 518 IF(HTZ(N,M+1))519,521,519
 519 VARI=Z(N,M+1)
 520 GO TO 522
 521 VARI=Z(N,M)
 522 VPB(N,M)=ALPHA*Z(N,M)+BETA*(VAUP+VALO+VALE+VARI)
 523 CONTINUE
 530 DO 535 N=2,NE
 531 DO 535 M=3,MEH
 532 IF(HTZ(N,M)) 535,535,2533
2533 IF(HTZ(N,M)-3.)533,535,533                              Computation of Z
 533 Z(N,M)=VPB(N,M)-A4*(HGU(N,M)*U(N,M)-HGU(N,M-1)*U(N,M-1)+HGV(N-1,M)
    1*V(N-1,M)-HGV(N,M)*V(N,M))
 535 CONTINUE
1537 DO 1543 N = 1,NEH
1538 DO 1543 M = 1,MEH
1539 IF(HTU(N,M)) 1541,1541,1540
1540 HGU(N,M) = HTU(N,M) +(Z(N,M)+Z(N,M+1))/2.
1541 IF(HTV(N,M)) 1543,1543,1542                          Computation of
1542 HGV(N,M) = HTV(N,M)+(Z(N,M)+Z(N+1,M))/2.            real depths.
1543 CONTINUE
1560 DO 1568 N=1,NE
1561 DO 1568 M=1,ME
1562 IF(HTU(N,M))1565,1565,1563
1563 IF(HGU(N,M)-10.)1564,1564,1565
1564 HGU(N,M)=-3.                                             Checking of
1565 IF(HTV(N,M))1568,1568,1566                               "drying" of tidal
1566 IF(HGV(N,M)-10.)1567,1567,1568                           flats.
1567 HGV(N,M)=-3.
1568 CONTINUE
 540 DO 585 N=2,NEH
 541 DO 585 M=2,MEH
 542 IF(HTU(N,M))585,585,543
 543 IF(1-N)544,547,544
 544 IF(HTU(N-1,M))545,547,545
 545 VUP=U(N-1,M)
 546 GO TO 548
 547 VUP=U(N,M)
 548 IF(NE-N)549,552,549
 549 IF(HTU(N+1,M))550,552,550
 550 VLO=U(N+1,M)
 551 GO TO 553
 552 VLO=U(N,M)                                               Computation of Ū
 553 IF(1-M)554,557,554
 554 IF(HTU(N,M-1))555,557,555
 555 VLE=U(N,M-1)
```

```
 556 GO TO 558
 557 VLE=U(N,M)
 558 IF(ME-M)559,562,559
 559 IF(HTU(N,M+1))560,562,560
 560 VRI=U(N,M+1)
 561 GO TO 563
 562 VRI=U(N,M)
 563 UPB(N,M)=ALPHA*U(N,M)+BETA*(VUP+VLO+VLE+VRI)
 564 IF(HTV(N-1,M))565,569,565
 565 VUL=V(N-1,M)
 566 IF(HTV(N-1,M+1))567,570,567
 567 VUR=V(N-1,M+1)
 568 GO TO 571
 569 VUL=V(N-1,M+1)
 570 VUR=VUL
 571 IF(HTV(N,M))572,576,572
 572 VLL=V(N,M)
 573 IF(HTV(N,M+1))574,577,574                    Computation of
 574 VLR=V(N,M+1)                                 V*.
 575 GO TO 578
 576 VLL=V(N,M+1)
 577 VLR=VLL
 578 VPB(N,M)=(VUL+VLL+VUR+VLR)/4.
 585 CONTINUE
 586 DO 595 N=2,NEH
 587 DO 595 M=2,MEH
 588 IF(HTU(N,M)) 595,595,1570
1570 IF(HGU(N,M)-3.) 595,1571,589
1571 VPB(N,M)=0.
1572 GO TO 595                                    Computation of
 589 IF(UPB(N,M)*UPB(N,M))595,590,591             U.
 590 IF(VPB(N,M)*VPB(N,M))595,595,591
 591 ROOT=SQRTF(UPB(N,M)*UPB(N,M)+VPB(N,M)*VPB(N,M))
 592 GRZ=A3*ROOT
 593 VPB(N,M)= (1. - GRZ/HGU(N,M))*UPB(N,M) + A2*VPB(N,M) - A5*
    1(Z(N,M+1) - Z(N,M)) + XK(N,M)
 595 CONTINUE
1590 CONTINUE
3596 DO 3600 N=1,NE
     IF(HTU(N,1))3600,3600,3597
3597 IF(HTU(N,2))3600,3600,3598                   Special boundary
3598 VPB(N,1)=VPB(N,2)*HTU(N,1)/HTU(N,2)          settings.
3600 CONTINUE
 900 DO 918 N=1,NEH
 901 DO 918 M = 2,ME
 902 IF(HTV(N,M))918,918,903
 903 IF(HTU(N,M-1))904,908,904
 904 VUL=U(N,M-1)
 905 IF(HTU(N+1,M-1))906,909,906
 906 VLL=U(N+1,M-1)
 907 GO TO 910
```

```
 908 VUL=U(N+1,M-1)
 909 VLL=VUL
 910 IF(HTU(N,M))911,915,911
 911 VUR=U(N,M)
 912 IF(HTU(N+1,M))913,916,913
 913 VLR=U(N+1,M)
 914 GO TO 917
 915 VUR=U(N-1,M)
 916 VLR=VUR
 917 UPB(N,M)=(VUL+VLL+VUR+VLR)/4.
 918 CONTINUE
```

Computation of U^*

```
     GO TO 920
 920 DO 924 N=1,NE
 921 DO 924 M=1,ME
 922 IF(HTU(N,M))924,924,923
 923 U(N,M) = VPB(N,M)
 924 CONTINUE
```

Transfer of new U into proper array.

```
C
 930 DO 954 N=1,NE
 931 DO 954 M=1,ME
 932 IF(HTV(N,M))954,954,933
 933 IF(1-N)934,937,934
 934 IF(HTV(N-1,M))935,937,935
 935 VUP=V(N-1,M)
 936 GO TO 938
 937 VUP=V(N,M)
 938 IF(NE-N)939,942,939
 939 IF(HTV(N+1,M))940,942,940
 940 VLO=V(N+1,M)
 941 GO TO 943
 942 VLO=V(N,M)
 943 IF(1-M)944,947,944
 944 IF(HTV(N,M+1))945,947,945
 945 VLE=V(N,M-1)
 946 GO TO 948
 947 VLE=V(N,M)
 948 IF(ME-M)949,952,949
 949 IF(HTV(N,M+1))950,952,950
 950 VRI=V(N,M+1)
 951 GO TO 953
 952 VRI=V(N,M)
 953 VPB(N,M)=ALPHA*V(N,M)+BETA*(VUP+VLO+VLE+VRI)
 954 CONTINUE
```

Computation of $\bar{V}$

```
 960 DO 969 N = 2,NEH
 961 DO 969 M=2,MEH
 962 IF(HTV(N,M)) 969,969,1990
1990 IF(HGV(N,M)-3.) 969,1991,963
1991 V(N,M)=0.
1992 GO TO 969
 963 IF(VPB(N,M)*VPB(N,M))969,964,965
 964 IF(UPB(N,M)*UPB(N,M))969,969,965
 965 ROOT=SQRTF(VPB(N,M)*VPB(N,M)+UPB(N,M)*UPB(N,M))
 966 GRZ=A3*ROOT
 967 V (N,M) = (1. - GRZ/HGV(N,M))*VPB(N,M) - A2*UPB(N,M) - A5*(Z(N,M)
```

Computation of V

```
     1 - Z(N+1,M)) + YK(N,M)
  969 CONTINUE
      V(26,40) = V(26,40) + 1.3   ------ special inflow setting.
C     COMPUTATIONS OF NET FLOW THROUGH THE SECTIONS
      DL2 = 2. * DL
      IF(T-TRB) 400,5250,5250
 5250 IF(T-TRE) 5251,5251,400
 5251 AA1=AA1+(V(25,40)*HTV(25,40)*A1*DL2)/1000000.
      BB2=BB2+(U(23,39)*HTU(23,39)*A1*DL2+U(24,39)*HTU(24,39)*A1*DL2
     1 +U(25,39)*HTU(25,39)*A1*DL2)/1000000.
      CC3 = CC3+(V(22,39)*HTV(22,39)*A1*DL2+V(23,39) *HTV(23,39)*A1*DL2+
     1V(24,39)*HTV(24,39)*A1*DL2)/1000000.
C
C     TRANSPORT THROUGH SECTIONS
C     TRANS(UORV,HTUORHTV,N,M,KEY,NUMBER)
      TRX = TRANS(U,HTU,36,7,2,5)
      AA2 = TRX*A1*DL2/1000000.0
      TRX = TRANS(U,HTU,37,15,2,3)
      AA3 = TRX*A1*DL2/1000000.0
      TRX = TRANS(U,HTU,32,27,2,5)
      AA4 = TRX*A1*DL2/1000000.0
      TRX = TRANS(U,HTU,23,39,2,3)
      BB3 = TRX*A1*DL2/1000000.0          Computations of currents
      TRX = TRANS(U,HTU,23,37,2,3)        through special sections.
      BB4 = TRX*A1*DL2/1000000.0
      TRX = TRANS(U,HTU,20,35,2,6)
      BB5 = TRX*A1*DL2/1000000.0
      TRX = TRANS(U,HTU,17,27,2,10)
      BB6 = TRX*A1*DL2/1000000.0
      TRX = TRANS(U,HTU,12,18,2,18)
      BB7 = TRX*A1*DL2/1000000.0
      TRX = TRANS(U,HTU,6,9,2,28)
      BB8 = TRX*A1*DL2/1000000.0
      TRX = TRANS(V,HTV,23,37,1,3)
      CC4 = TRX*A1*DL2/1000000.0
      TRX = TRANS(V,HTV,20,35,1,5)
      CC5 = TRX*A1*DL2/1000000.0
      TRX = TRANS(V,HTV,17,27,1,13)
      CC6 = TRX*A1*DL2/1000000.0
      TRX = TRANS(V,HTV,12,18,1,23)
      CC7 = TRX*A1*DL2/1000000.0
      TRX = TRANS(V,HTV,6,9,1,34)
      CC8 = TRX*A1*DL2/1000000.0
 1960 CONTINUE
C     COMPUTATION OF REST CURRENTS
  400 IF(LE)410,410,401
  401 IF(T-TRB)410,402,402
  402 IF(T-TRE)403,403,410
  403 TIC=TIC+A1                          Accumulation for rest
      DO 4090 N=1,NE                      current computation.
      DO 4090 M=1,ME
      IF(HTZ(N,M))4090,4090,4091
```

```
4091 USR(N,M)=USR(N,M)+(U(N,M)*2*DT)/100.
     VSR(N,M)=VSR(N,M)+(V(N,M)*2*DT)/100.
4090 CONTINUE
 404 DO 409 N=1,NE
 405 DO 409 M=1,ME
 407 IF(S(N,M))2401,2401,2403
2401 IF(S(N,M-1))2402,2402,2403
2402 IF(S(N,M+1))2404,2404,2403
2403 US(N,M)=US(N,M)+U(N,M)*2*DT
     GO TO 2405
2404 US(N,M)=0.
2405 IF(S(N,M))2406,2406,2408
2406 IF(S(N-1,M))2407,2407,2408
2407 IF(S(N+1,M))2409,2409,2408
2408 VS(N,M)=VS(N,M)+V(N,M)*2*DT
     GO TO 409
2409 VS(N,M)=0.
 409 CONTINUE
     B4=2.*DL
     DO 2610 N=1,NE
     DO 2610 M=1,ME
     IF(ABSF(US(N,M))/B4-1.) 2604,2601,2601
2601 IF(US(N,M)) 2602,2603,2603
2602 US(N,M)=-B4
     GO TO 2604
2603 US(N,M)=B4
2604 IF(ABSF(VS(N,M))/B4-1.) 2610,2605,2605
2605 IF(VS(N,M)) 2606,2607,2607
2606 VS(N,M)=-B4
     GO TO 2610
2607 VS(N,M)=B4
2610 CONTINUE
 410 IF(LI)984,984,411
 411 CONTINUE
     IF(T-10800.) 412,4500,412
4500 CONTINUE
     S(22,35) = 10000.
     S(22,27) = 10000.
     S(5,6)   = 10000.
     S(4,28)  = 10000.
     S(7,18)  = 10000.
     S(6,39)  = 10000.
     S(12,9)  = 10000.
     S(12,32) = 10000.
     S(17,18) = 10000.
     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.
```

Setting of current "accumulations" for computations of advection of pollutant

Instantaneous release of pollutants.

```
 412 B1=DL*DL
     AUS(1)=50000.
 413 B2 = (4.0*DT*AUS(1))/B1
 415 B5 = (AUS(1) * DT)/B1
 416 Y1=0.
     S(26,40) = S(26,40) + 3. -----continuous release of pollutants.
C    COMPUTATION OF DISPERSION AND DIFFUSION
 417 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))          Computation of advection
     US(N,M)=0.                              parametefs.
     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
2218 SSV=S(N,M)/(DT*ABS(V(N,M))
     VS(N,M)=0.
     GO TO 4428
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(U(N,M))*SSH)-(DT*ABSF(V(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))
  429 IF(S(N,M)) 2429,2429,300
 2429 S(N,M) = 0.
 3000 CONTINUE
  300 CONTINUE
```

Computation of diffusion and advection.

```
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)=S(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 IF(C1)470,472,470
  470 IF(T-TW)472,472,471
  471 CALL JOB4
  472 CONTINUE
      RETURN
      END
      FUNCTION TRANS(FV,FH,N,M,KEY,NUM)
      DIMENSION FV(42,47),FH(42,47)
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
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
```

Function to compute transport through section.

```
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 JOB3
        DIMENSION HTZ(42,47),HTU(42,47),HTV(42,47),Z(42,47),U(42,47),V(42,
       147),US(42,47),VS(42,47),RAD(42,47),ANG(42,47),XK(42,47),YK(42,47),
       2     NU(6),MU(6),NK(85),MK(85),AH(5,85),CAPAP(5,85),PREC(5,85),ARG
       3(5,85),PERU(85),PERV(85),NL(85),ML(85),NZ(10),MZ(10),V1(10),UPB(42
       4,47),VPB(42,47),S(42,47),A(85)        ,V2(10),V3(10),V4(10),HGU(42,4
       57),HGV(42,47),H(5),B(5),AL(5),CAPA(5),CAPB(5),AINC(5),CAINC(5),
       6                 NG(40),MG(40),TIT(3),AUS(5),TAR(85),DTC(85)
       7,USR(42,47),VSR(42,47)
        COMMON HTZ,HTU,HTV,Z,U,V,US,VS,RAD,ANG,XK,YK,   NU,MU,NK,MK,AH,CAPA
       1P,PREC,ARG,PERU,PERV,NL,ML,NZ,MZ,V1,UPB,VPB,S,A    ,V2,V3,V4,HGU,HG
       2V,H,B,AL,CAPA,CAPB,AINC,CAINC,NG,MG,TIT,AUS,TAR,DTC,USR,VSR,
       3T1,T2,T3,TE,TW,A1,A2,A3,A4,DT,BETA,ALPHA,F,R,DL,G,C,C3,NE,ME,NEH,M
       4EH,JA,IUE,KKE,NURU,SI,T,RBETA,C1,SIGMA,ROL,A5,RU,RV,KO,LI,LE,LU,KI
       5,LOP,TRB,TRE,TIC,          LUN,LA,TIL,LIV,LEET
       6,POSN,POSM,DRI,STV,ETV,TINCV,STH,ETC,TINCH,CNCU,CAINCU
       7,AA1,BB2,CC3,TSC,DL2
       8,AA2,AA3,AA4,BB3,BB4,BB5,BB6,BB7,BB8,CC4,CC5,CC6,CC7,CC8
C       WRITING OF OUTPUT FIELDS
   31 FORMAT(9F8.3)
   32 FORMAT(9F8.0)
  601 CONTINUE
  610 PRINT602,T,(N,N=1,16)
  611 PRINT604,(N,(Z(N,M),M=1,16),N=1,NE)
  612 PRINT603,(N,N=17,32)
  613 PRINT604,(N,(Z(N,M),M=17,32),N=1,NE)
        PRINT1605,(N,N=33,47)
        PRINT1604,(N,(Z(N,M),M=33,47),N=1,NE)
 1605 FORMAT(//8X,15I8)
 1604 FORMAT(1X,I8,15F8.0,/)
  614 DO 639 N=1,NE
  617 DO 639 M=1,ME
  618 IF(V(N,M))620,619,620
  619 IF(U(N,M))620,622,620
  620 RAD(N,M)=SQRTF(ABSF(V(N,M))**2+ABSF(U(N,M))**2)
  621 GO TO 625
  622 RAD(N,M)=0.
  623 ANG(N,M)=999.
  624 GO TO 639
  625 ANG(N,M)=ABSF(ASINF(V(N,M)/RAD(N,M)))
  626 ANG(N,M)=ANG(N,M)*(360./6.2831853073)
  627 IF(V(N,M))629,628,628
  628 IF(U(N,M))631,636,636
  629 IF(U(N,M))633,633,630
  630 GO TO 635
```

```
 631 ANG(N,M)=180.-ANG(N,M)                                      
 632 GO TO 636                                                   
 633 ANG(N,M)=180.+ANG(N,M)                                      Computation of speed
 634 GO TO 636                                                   and direction from U
 635 ANG(N,M)=360.-ANG(N,M)                                      and V components.
 636 ANG(N,M) = (90.-ANG(N,M)) + TIL
 637 IF(ANG(N,M))638,639,639
 638 ANG(N,M)=360.+ANG(N,M)
2638 GO TO 637
 639 CONTINUE
 650 PRINT 607,T,(N,N=1,14)
 651 PRINT 606,(N,(RAD(N,M),ANG(N,M),M=1,14),N=1,NE)
 652 PRINT 609,(N,N=15,28)
 653 PRINT 608,(N,(RAD(N,M),ANG(N,M),M=15,28),N=1,NE)
     PRINT 609,(N,N=29,42)
     PRINT 608,(N,(RAD(N,M),ANG(N,M),M=29,42),N=1,NE)
     PRINT1609,(N,N=43,47)
     PRINT1608,(N,(RAD(N,M),ANG(N,M),M=43,47),N=1,NE)
1608 FORMAT((1X,I8,5(F4.0,1H,,F4.0)/))
1609 FORMAT(//8X, 5I8)                                           Printing of output
 660 IF(LI) 1660,1660,661                                        fields.
 661 PRINT 670,T,(N,N=1,16)
 662 PRINT 604,(N,(S(N,M),M=1,16),N=1,NE)
     PRINT 605,(N,N=17,32)
     PRINT 604,(N,(S(N,M),M=17,32),N=1,NE)
     PRINT1605,(N,N=33,47)
     PRINT1604,(N,(S(N,M),M=33,47),N=1,NE)
 670 FORMAT(1H1,14X,22HCONCENTRATION AFTER T=,F9.0,//8X,I8,15I8//)
1660 IF(LEET) 1650,1650,1661
1661 IF(T-T4) 1650,1662,1650
1662 WRITE(5,31)((Z(N,M),M=1,ME),N=1,NE)
1664 WRITE (5,31)((U(N,M),M=1,ME),N=1,NE)                        Writing of outputs on
1665 WRITE (5,31)((V(N,M),M=1,ME),N=1,NE)                        tape.
1666 WRITE (5,32)T
1667 ENDFILE 5
1650 IF(LOP-10) 1630,640,1630
1630 IF(LOP-5)640,1631,640
1631 DO 1636 N=1,NE
1632 DO 1636 M=1,ME
1633 IF(VSR(N,M))1634,1635,1634                                  Rest current output.
1634 V(N,M)=VSR(N,M)/(0.01*TIC)
1635 IF(USR(N,M))1636,1637,1636
1636 U(N,M)=USR(N,M)/(0.01*TIC)
1637 CONTINUE
1638 LOP=LOP+10
1639 GO TO 624
 640 DO 646 I=1,NURU
 641 N=NZ(I)
```

```
 642 M=MZ(I)
 643 V2(I)=Z(N,M)
 644 V3(I)=RAD(N,M)
 645 V4(I)=ANG(N,M)
 646 CONTINUE
 647 WRITE(2)T,(V2(I),I=1,NURU),(V3(I),I=1,NURU),(V4(I),I=1,NURU)
```

Writing of data from selected points on tape.

```
 602 FORMAT(1H1,14X,21HWATER HEIGHT AFTER T=,F9.0,9H SEC (CM) /8X,I8,15
    1I8/ )
 604 FORMAT(I8,16F8.0/)
 605 FORMAT(//8X,I4,16I8/)
 606 FORMAT((I8,14(F4.0,1H,,F4.0)/))
 607 FORMAT(1H1,14X,76HRESULTANT CURRENT VELOCITY MAGNITUDE(CM/SEC) AND
    1DIRECTION(DEGREES) AFTER T= ,F9.0,4H SEC,//8X,I7,13I9)
 608 FORMAT((I8,14(F4.0,1H,F4.0)/))
 609 FORMAT(8X,I7,13I9/)
1670 FORMAT(///,5X,31HPOSITION OF DRIFTING OBJECT, N=,F6.2,2HM=,F6.2,//
    1/)
1651 IF(LIV)654,654,1652
1652 IF(LIV-2)1653,654,1653
1653 PRINT 1670,POSN,POSM
 654 CONTINUE
```

Output of positions of drifting objects.

```
3610 IF(T-TRB) 3620,3611,3611
3611 IF(T-TRE) 3612,3612,3620
3612 AA1=AA1/3600.
     AA2 = AA2/3600.
     AA3 = AA3/3600.
     AA4 = AA4/3600.
     BB2=BB2/3600.
     BB3 = BB3/3600.
     BB4 = BB4/3600.
     BB5 = BB5/3600.
     BB6 = BB6/3600.
     BB7 = BB7/3600.
     BB8 = BB8/3600.
     CC3=CC3/3600.
     CC4 = CC4/3600.
     CC5 = CC5/3600.
     CC6 = CC6/3600.
     CC7 = CC7/3600.
     CC8 = CC8/3600.
     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
```

Output of transport through sections.

```
5101 FORMAT(1H1,//10X,26HTRANSPORT,SECTION A, M3=    ,F10.0,//28X,2HB,,
    15X,F10.0,//28X,2HC,,5X,F10.0,//)
     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
      RETURN
      END
```

Zeroing of transport array.

```
      SUBROUTINE JOB4
      DIMENSION HTZ(42,47),HTU(42,47),HTV(42,47),Z(42,47),U(42,47),V(42,
     147),US(42,47),VS(42,47),RAD(42,47),ANG(42,47),XK(42,47),YK(42,47),
     2     NU(6),MU(6),NK(85),MK(85),AH(5,85),CAPAP(5,85),PREC(5,85),ARG
     3(5,85),PERU(85),PERV(85),NL(85),ML(85),NZ(10),MZ(10),V1(10),UPB(42
     4,47),VPB(42,47),S(42,47),A(85)       ,V2(10),V3(10),V4(10),HGU(42,4
     57),HGV(42,47),H(5),B(5),AL(5),CAPA(5),CAPB(5),AINC(5),CAINC(5),
     6                 NG(40),MG(40),TIT(3),AUS(5),TAR(85),OTC(85)
     7,USR(42,47),VSR(42,47)
      COMMON HTZ,HTU,HTV,Z,U,V,US,VS,RAD,ANG,XK,YK,  NU,MU,NK,MK,AH,CAPA
     1P,PREC,ARG,PERU,PERV,NL,ML,NZ,MZ,V1,UPB,VPB,S,A    ,V2,V3,V4,HGU,HG
     2V,H,B,AL,CAPA,CAPB,AINC,CAINC,NG,MG,TIT,AUS,TAR,OTC,USR,VSR,
     3T1,T2,T3,TE,TW,A1,A2,A3,A4,DT,BETA,ALPHA,F,R,DL,G,C,C3,NE,ME,NEH,M
     4EH,JA,IUE,KKE,NURU,SI,T,RBETA,CI,SIGMA,ROL,A5,RU,RV,KO,LI,LE,LU,KI
     5,LOP,TRA,TRE,TIC,          LUN,LA,TIL,LIV,LEET
     6,POSN,POSM,DRI,STV,ETV,TINCV,STH,ETC,TINCH,CNCU,CAINCU
     7,AA1,BB2,CC3,TSC,DL2
     8,AA2,AA3,AA4,BB3,BB4,BB5,BB6,BB7,BB8,CC4,CC5,CC6,CC7,CC8
C     WIND CURRENT COMPUTATION AND COMPUTATION OF
C     DRIFT OF BOATS AND OIL
  704 CONTINUE
  705 IF(SI-T)745,745,706
  706 KKE=(IUE/2)*3
  707 IU=IUE/2
  708 K=XFIXF ((T-TW+A(1))/A(1))+1
  709 L = K +1
  715 DO 743 I=1,IU
  716 N1=NU(2*I-1)
  717 N2=NU(2*I)
  718 M1=MU(2*I-1)
  719 M2=MU(2*I)
  720 DO 743 N = N1,N2
  721 DO 743 M=M1,M2
  722 IF(HTU(N,M))733,733,723
  723 IF(HGU(N,M)) 724,733,724
  724 IF(HGU(N,M)+3.) 733,733,725
 725  HMLD = 2500. + 8.* Z(N,M)
  726 IF(HMLD-HGU(N,M)) 730,727,727
```

Computation of wind current component and a drift component due to wind.

```
727 HMLD = HGU(N,M)
730 XK(N,M)=C3*A(K)**2*COSF(A(L)*0.0174533)/HMLD
731 IF(LIV)733,733,732
732 UPB(N,M)=A(K)*COSF(A(L))*100.
733 IF(HTV(N,M))743,743,734
734 IF(HGV(N,M)) 735,743,735
735 IF(HGV(N,M)+3.) 743,743,736
736 HMLD=2500.+8.*Z(N,M)
    IF(HMLD-HGV(N,M)) 740,737,737
737 HMLD = HGV(N,M)
740 YK(N,M)=C3*A(K)**2*SINF(A(L)*0.0174533)/HMLD
741 IF(LIV)743,743,742
742 VPB(N,M)=A(K)*SINF(A(L))*100.
743 CONTINUE
744 GO TO 750
745 DO 749 N=1,NE
746 DO 749 M=1,ME
747 XK(N,M)=0.
748 YK(N,M)=0.
749 CONTINUE
750 CONTINUE
751 IF(LIV)780,780,752
752 IF(T-TRB)780,753,753
753 IF(T-TRE)754,754,780
754 IF(LIV-2)770,755,755
755 PROD=0.015
756 DO 766 N=2,NEH
757 DO 766 M=2,MEH
758 IF(HTZ(N,M)) 766,766,759
759 UMR=UPB(N,M)*PROD*A4
760 UML=UPB(N,M-1)*PROD*A4
761 VMB=VPB(N,M)*PROD*A4
762 VMO=VPB(N-1,M)*PROD*A4
763 S(N,M)=S(N,M)+UMR*(S(N,M)-S(N,M+1))+UML*(S(N,M-1)-S(N,M))-VMB*(S(N
   1,M)-S(N+1,M))-VMO*(S(N-1,M)-S(N,M))
764 IF(S(N,M))765,766,766
765 S(N,M)=0.
766 CONTINUE
770 IF(LIV-2) 771,780,780
771 ND=XFIXF(POSN)
772 MD=XFIXF(POSM)
773 UDIS = (A1 * (U(ND,MD) + UPB(ND,MD) * DRI)) / 2. * DL
774 POSM=POSM+UDIS
775 VDIS = (A1 * (V(ND,MD) + VPB(ND,MD) * DRI)) / 2. * DL
776 POSN=POSN+VDIS
780 CONTINUE
    RETURN
    END
    SUBROUTINE JOB5
    DIMENSION HTZ(42,47),HTU(42,47),HTV(42,47),Z(42,47),U(42,47),V(42,
   147),US(42,47),VS(42,47),RAD(42,47),ANG(42,47),XK(42,47),YK(42,47),
   2     NU(6),MU(6),NK(85),MK(85),AH(5,85),CAPAP(5,85),PREC(5,85),ARG
   3(5,85),PERU(85),PERV(85),NL(85),ML(85),NZ(10),MZ(10),V1(10),UPB(42
```

Computation of drift of objects and oil at the surface (due to wind)

```
     4,47),VPB(42,47),S(42,47),A(85)      ,V2(10),V3(10),V4(10),HGU(42,4
     57),HGV(42,47),H(5),B(5),AL(5),CAPA(5),CAPB(5),AINC(5),CAINC(5),
     6               NG(40),MG(40),TIT(3),AUS(5),TAR(85),OTC(85)
     7,USR(42,47),VSR(42,47)
      COMMON HTZ,HTU,HTV,Z,U,V,US,VS,RAD,ANG,XK,YK,  NU,MU,NK,MK,AH,CAPA
     1P,PREC,ARG,PERU,PERV,NL,ML,NZ,MZ,V1,UPB,VPB,S,A   ,V2,V3,V4,HGU,HG
     2V,H,B,AL,CAPA,CAPB,AINC,CAINC,NG,MG,TIT,AUS,TAR,OTC,USR,VSR,
     3T1,T2,T4,TE,TW,A1,A2,A3,A4,DT,BETA,ALPHA,F,R,DL,G,C,C3,NE,ME,NEH,M
     4EH,JA,IUE,KKE,NURU,SI,T,RBETA,C1,SIGMA,ROL,A5,RU,RV,KO,LI,LE,LU,KI
     5,LOP,TRB,TRE,TIC,            LUN,LA,TIL,LIV,LEET
     6,POSN,POSM,DRI,STV,ETV,TINCV,STH,ETC,TINCH,CNCU,CAINCU
     7,AA1,BB2,CC3,TSC,DL2
     8,AA2,AA3,AA4,BB3,BB4,BB5,BB6,BB7,BB8,CC4,CC5,CC6,CC7,CC8
C    OUTPUT OF SPECIAL POINTS
  802 CONTINUE
  803 FORMAT(1H1,5X,114HWATER ELEVATION(CM) AND CURRENT VELOCITY RESULTA
     1NT(MAGNITUDE IN CM(SEC AND DIRECTION IN DEGREES) AT SPECIAL POINTS
     2///7X,4HTIME,4X,10(4X,A5)/15X,5(4X,A5)/)
  804 FORMAT(9X,5H(SEC),4X,10(2X,I3,1H,,I3)/15X,5(2X,I3,1H,,I3)//)
  805 FORMAT(F11.0,4X,10F9.2/15X,5F9.2)          ┐
  806 FORMAT(5X,14HEND OF PROGRAM)               │
  807 FORMAT(15X,10(F4.0,1H,,F4.0)/15X,5(F4.0,1H,,F4.0)/)
  808 PRINT 803,(V1(I),I=1,NURU)                 │
  809 PRINT 804,(NZ(I),MZ(I),I=1,NURU)           ├── Output at special points.
  810 READ (2)T,(V2(I),I=1,NURU),(V3(I),I=1,NURU),(V4(I),I=1,NURU)
  813 PRINT 805,T,(V2(I),I=1,NURU)               │
  814 PRINT 807,(V3(I),V4(I),I=1,NURU)           ┘
  815 IF (TE-T)816,816,810
  816 PRINT 806
      RETURN
      END
      SUBROUTINE JOB6
      DIMENSION HTZ(42,47),HTU(42,47),HTV(42,47),Z(42,47),U(42,47),V(42,
     147),US(42,47),VS(42,47),RAD(42,47),ANG(42,47),XK(42,47),YK(42,47),
     2    NU(6),MU(6),NK(85),MK(85),AH(5,85),CAPAP(5,85),PREC(5,85),ARG
     3(5,85),PERU(85),PERV(85),NL(85),ML(85),NZ(10),MZ(10),V1(10),UPB(42
     4,47),VPB(42,47),S(42,47),A(85)      ,V2(10),V3(10),V4(10),HGU(42,4
     57),HGV(42,47),H(5),B(5),AL(5),CAPA(5),CAPB(5),AINC(5),CAINC(5),
     6               NG(40),MG(40),TIT(3),AUS(5),TAR(85),OTC(85)
     7,USR(42,47),VSR(42,47)
      COMMON HTZ,HTU,HTV,Z,U,V,US,VS,RAD,ANG,XK,YK,  NU,MU,NK,MK,AH,CAPA
     1P,PREC,ARG,PERU,PERV,NL,ML,NZ,MZ,V1,UPB,VPB,S,A   ,V2,V3,V4,HGU,HG
     2V,H,B,AL,CAPA,CAPB,AINC,CAINC,NG,MG,TIT,AUS,TAR,OTC,USR,VSR,
     3T1,T2,T4,TE,TW,A1,A2,A3,A4,DT,BETA,ALPHA,F,R,DL,G,C,C3,NE,ME,NEH,M
     4EH,JA,IUE,KKE,NURU,SI,T,RBETA,C1,SIGMA,ROL,A5,RU,RV,KO,LI,LE,LU,KI
     5,LOP,TRB,TRE,TIC,            LUN,LA,TIL,LIV,LEET
     6,POSN,POSM,DRI,STV,ETV,TINCV,STH,ETC,TINCH,CNCU,CAINCU
     7,AA1,BB2,CC3,TSC,DL2
     8,AA2,AA3,AA4,BB3,BB4,BB5,BB6,BB7,BB8,CC4,CC5,CC6,CC7,CC8
C     ENTRY TO READ VECTOR + HEIGHT CARDS
      ETH = ETC
  200 READ(50,215)STV,ETV,TINCV                  ┐
  201 PRINT 216,STV,ETV,TINCV                    │
  202 READ(50,215)STH,ETC,TINCH                  │
```

```
  203 PRINT 216,STH,ETC,TINCH
  215 FORMAT(3F10.0)
  216 FORMAT(10X,3F10.0)                          Outputs for plotting.
C       PAUSE 3 IS FOR MOUNTING OF PLOTTING TAPE
C 217 PAUSE 3
  218 WRITE (3)HTZ,V1,NURU,ME,NE
      RETURN
      END
      SUBROUTINE JOB7
      DIMENSION HTZ(42,47),HTU(42,47),HTV(42,47),Z(42,47),U(42,47),V(42,
     147),US(42,47),VS(42,47),RAD(42,47),ANG(42,47),XK(42,47),YK(42,47),
     2     NU(6),MU(6),NK(85),MK(85),AH(5,85),CAPAP(5,85),PREC(5,85),ARG
     3(5,85),PERU(85),PERV(85),NL(85),ML(85),NZ(10),MZ(10),V1(10),UPB(42
     4,47),VPB(42,47),S(42,47),A(85)       ,V2(10),V3(10),V4(10),HGU(42,4
     57),HGV(42,47),H(5),B(5),AL(5),CAPA(5),CAPB(5),AINC(5),CAINC(5),
     6                  NG(40),MG(40),TIT(3),AUS(5),TAR(85),OTC(85)
     7,USR(42,47),VSR(42,47)
      COMMON HTZ,HTU,HTV,Z,U,V,US,VS,RAD,ANG,XK,YK,  NU,MU,NK,MK,AH,CAPA
     1P,PREC,ARG,PERU,PERV,NL,ML,NZ,MZ,V1,UPB,VPB,S,A    ,V2,V3,V4,HGU,HG
     2V,H,B,AL,CAPA,CAPB,AINC,CAINC,NG,MG,TIT,AUS,TAR,OTC,USR,VSR,
     3T1,T2,TB,TE,TW,A1,A2,A3,A4,DT,BETA,ALPHA,F,R,DL,G,C,C3,NE,ME,NEH,M
     4EH,JA,IUE,KKE,NURU,SI,T,RBETA,C1,SIGMA,ROL,A5,RU,RV,KO,LI,LE,LU,KI
     5,LOP,TRB,TRE,TIC,            LUN,LA,TIL,LIV,LEET
     6,POSN,POSM,DRI,STV,ETV,TINCV,STH,ETC,TINCH,CNCU,CAINCU
     7,AA1,BB2,CC3,TSC,DL2
     8,AA2,AA3,AA4,BB3,BB4,BB5,BB6,BB7,BB8,CC4,CC5,CC6,CC7,CC8
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.                                     Outputs for plotting.
  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     (3)T,VEC,HT,RAD,ANG,Z
  239 CONTINUE
      RETURN
      END
      FUNCTION ASINF(X)
      ASINF = ASIN(X)
      RETURN
      END
      FUNCTION SINF(X)
      SINF = SIN(X)
      RETURN
      END
      FUNCTION COSF(X)
      COSF = COS(X)
      RETURN
      END
      FUNCTION SQRTF(X)
      SQRTF = SQRT(X)
      RETURN
      END
```

```
      FUNCTION ABSF(X)
      ABSF = ABS(X)
      RETURN
      END
      FUNCTION XFIXF(X)
      XFIXF = IFIX(X)
      RETURN
      END
C
```