SIMULATION OF OIL SLICK TRANSPORT IN GREAT LAKES CONNECTING CHANNELS

Volume III: User's Manual for the Lake-River Oil Spill Simulation Model

H.T. Shen
P.D. Yapa
M.E. Petroski

Report No. 86-3
March 1986

Department of Civil and Environmental Engineering
Clarkson University
Potsdam • New York • 13676

SIMULATION OF OIL SLICK TRANSPORT IN GREAT LAKES CONNECTING CHANNELS

Volume III: User's Manual for the Lake-River Oil Spill Simulation Model (LROSS)

by

Hung Tao Shen, Poojitha D. Yapa and Mark E. Petroski

Report No. 86-3

Department of Civil and Environmental Engineering
Clarkson University
Potsdam, New York 13676

March 1986

Sponsored by

U.S. Army Corps of Engineers
Contract No. DACA33-85-C-0001

PREFACE

The growing concern over the possible impacts of oil spills on aquatic environments has led to the development of a large number of computer models for simulating the transport and spreading of oil slicks in surface water bodies. Almost all of these models were developed for coastal environments. With the increase in inland navigation activities, oil slick simulation models for rivers and lakes are needed.

In this study, two computer models named as ROSS and LROSS are developed for simulating oil slick transport in rivers and lakes, respectively. The study was originated by the Detroit District, U.S. Army Corps of Engineers in relation to the Great Lakes limited navigation season extension study. The oil slick transformation processes considered in these models include advection, spreading, evaporation and dissolution. These models can be used for slicks of any shape originated from instantaneous or continuous spills in rivers and lakes with or without ice covers. Although developed for the need of the connecting channels in the upper Great Lakes, including the Detroit River, Lake St. Clair, St. Clair River, and St. Mary's River, these models are site independent and can be used [illegible] other rivers and lakes.

The programs are written in FORTRAN programming language to be compatible with FORTRAN77 compiler. In addition, a user-friendly, menu driven program with graphics capability is developed for the IBM-PC AT computer, so that these models can be easily used to assist the oil spill cleanup action in the connecting channels should a spill occur.

This report series is organized in four volumes, to provide a complete description of the analytical formulation of the models, the logic and structures of the computer programs, and the instructions for using the models. The title of these volume are:

Volume I:	Theory and Model Formulation
Volume II:	User's Manual for the River Oil Spill Simulation Model (ROSS)
Volume III:	User's Manual for the Lake-River Oil Spill Simulation Model (LROSS)
Volume IV:	User's Manual for the Microcomputer-Based Interactive Program

ACKNOWLEDGEMENTS

This study was supported by the U.S. Army Corps of Engineers under Contract No. DACA33-85-C-0001. Steve F. Daly and Mike Ferrick of the U.S. Army Cold Regions Research and Engineering Laboratory are the contracting officer's technical representatives. The writers would like to thank both of them as well as Dan Thompson and Don Williams of the Detroit District, U.S. Army Corps of Engineers, for their cooperation and assistance throughout the study period.

The writers would also like to acknowledge the assistance provided by the following individuals during various stages of this study: J.R. Weiser and R. Thomas, Detroit District, U.S. Army Corps of Engineers; F.H. Quinn and D.J. Schwab, Great Lakes Environmental Research Laboratory, NOAA; J. Galt and T. Kaiser, NOAA; M. Sydor, Inland Water Directorate, Canada; G. Tsang and R.O. Ramseir, Environment Canada; D. Mackay, University of Toronto; J.A. McCorquodale, University of Windsor; and S. Venkatesh, Atmospheric Environment Service, Canada.

TABLE OF CONTENTS

VOLUME III. User's Manual for the Lake-River Oil Spill Simulation Model (LROSS)

LIST OF FIGURES

LIST OF FIGURES (cont.)

LIST OF FIGURES (cont.)

CHAPTER I

INTRODUCTION

In this volume the computer model LROSS for lake-river oil spill simulation is presented along with instructions for using the computer model. The model simulates the transport of oil slick in a lake and traces this transport process as the slick moves into and along a river. This model is an extension of the model ROSS presented in Volume II. The analytical formulation of the computer model is presented in Volume I. Formulations on the oil slick transformation and river current distribution is the same as those developed in the model ROSS. The lake current distribution is computed using the rigid-lid lake circulation model (Schwab, et al., 1981 and 1984; Bennet, et al., 1983). The flow chart presented in Fig. 1 outlines the structure of the model. Discussions on some of the computer logic and techniques which were not discussed in Volumes I and II will be given in the following sections. Detailed presentations of the computer model, input data files and model output are given in later chapters.

I.1 Initial Input

Grid Control Data and River Geometry

The variables which govern the size and number of lake and river grids are first read into the program. Those dealing primarily with the lake are used to determine whether a point is located in either the lake or the river. This is extremely important since the correct grid size must be used when performing specific calculations. Again, it is emphasized that the

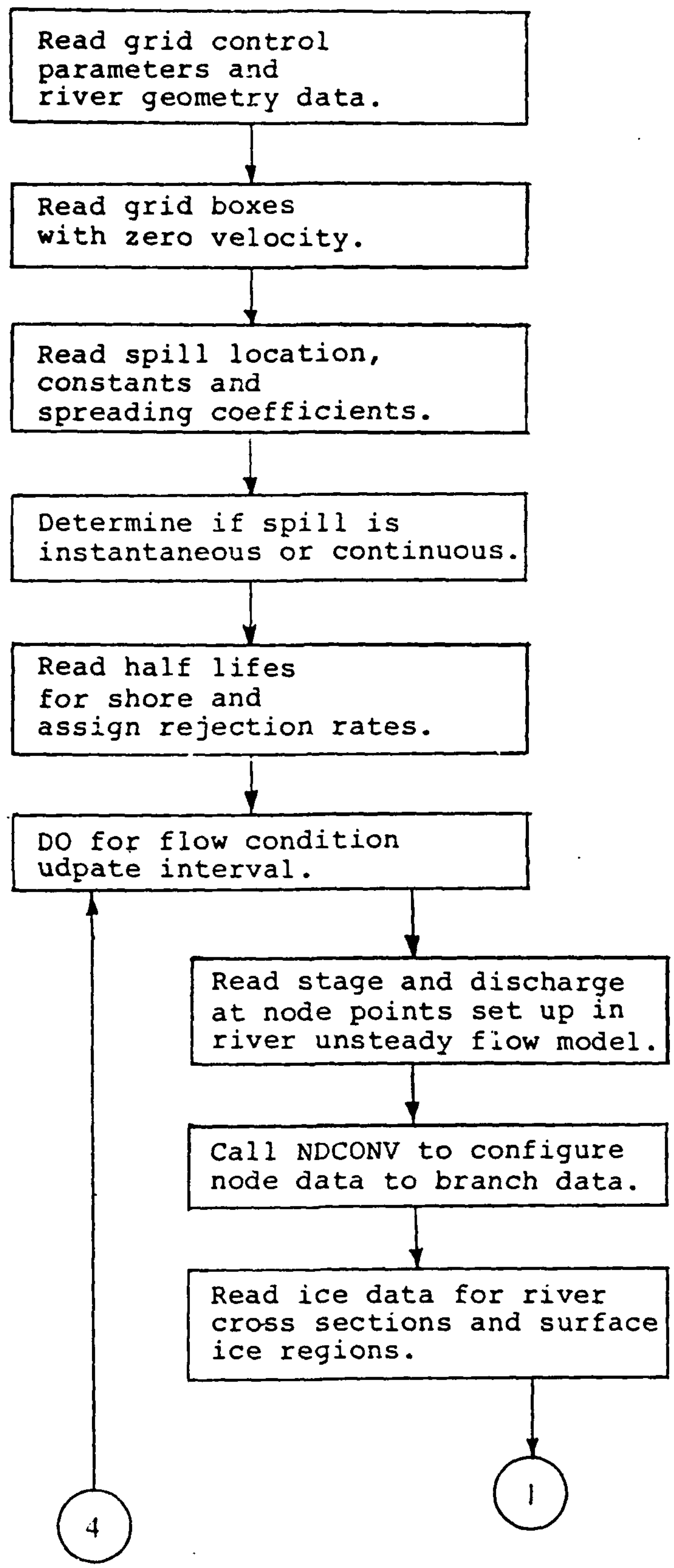


Figure 1. Block diagram of computer simulation model

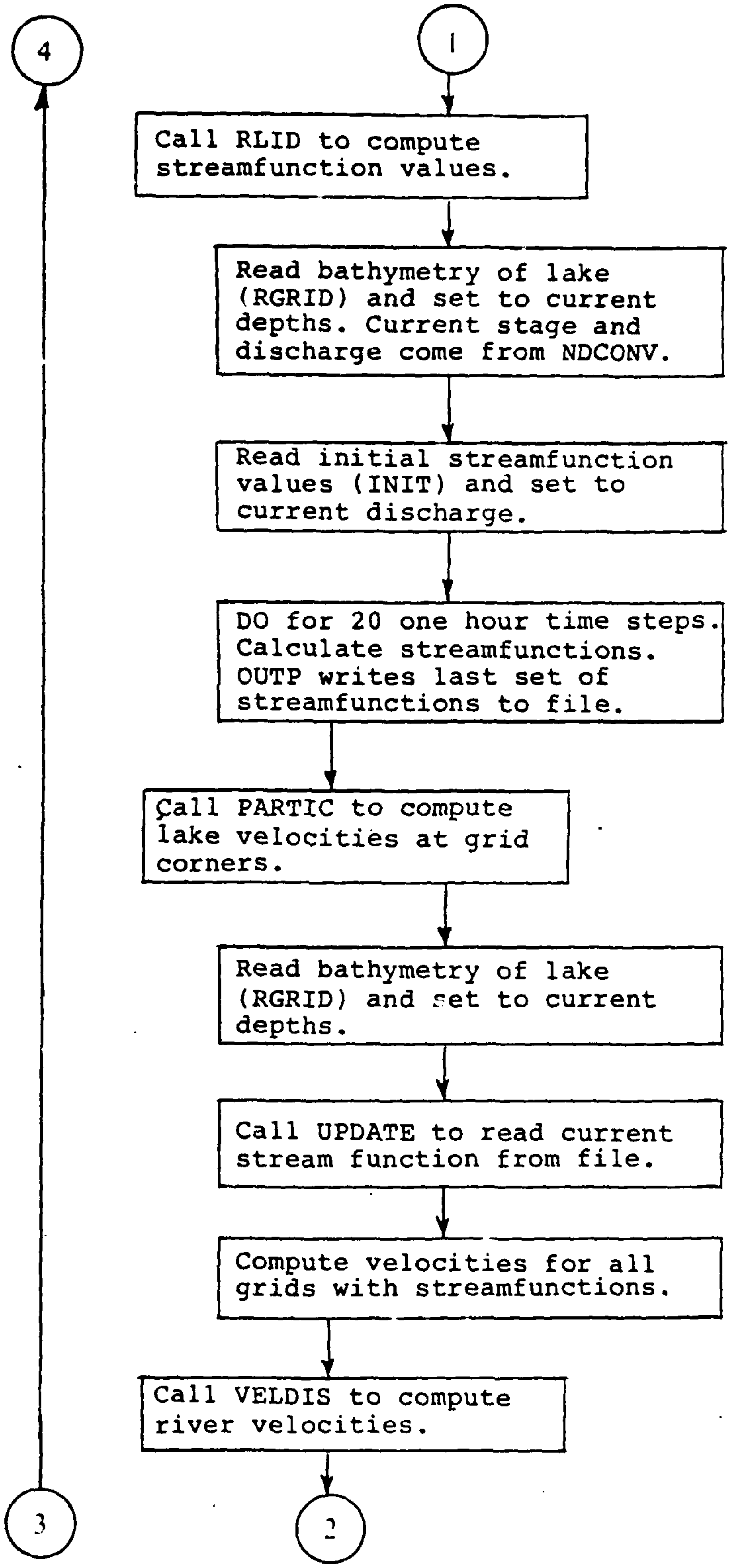


Figure 1. Block diagram of computer simulation model

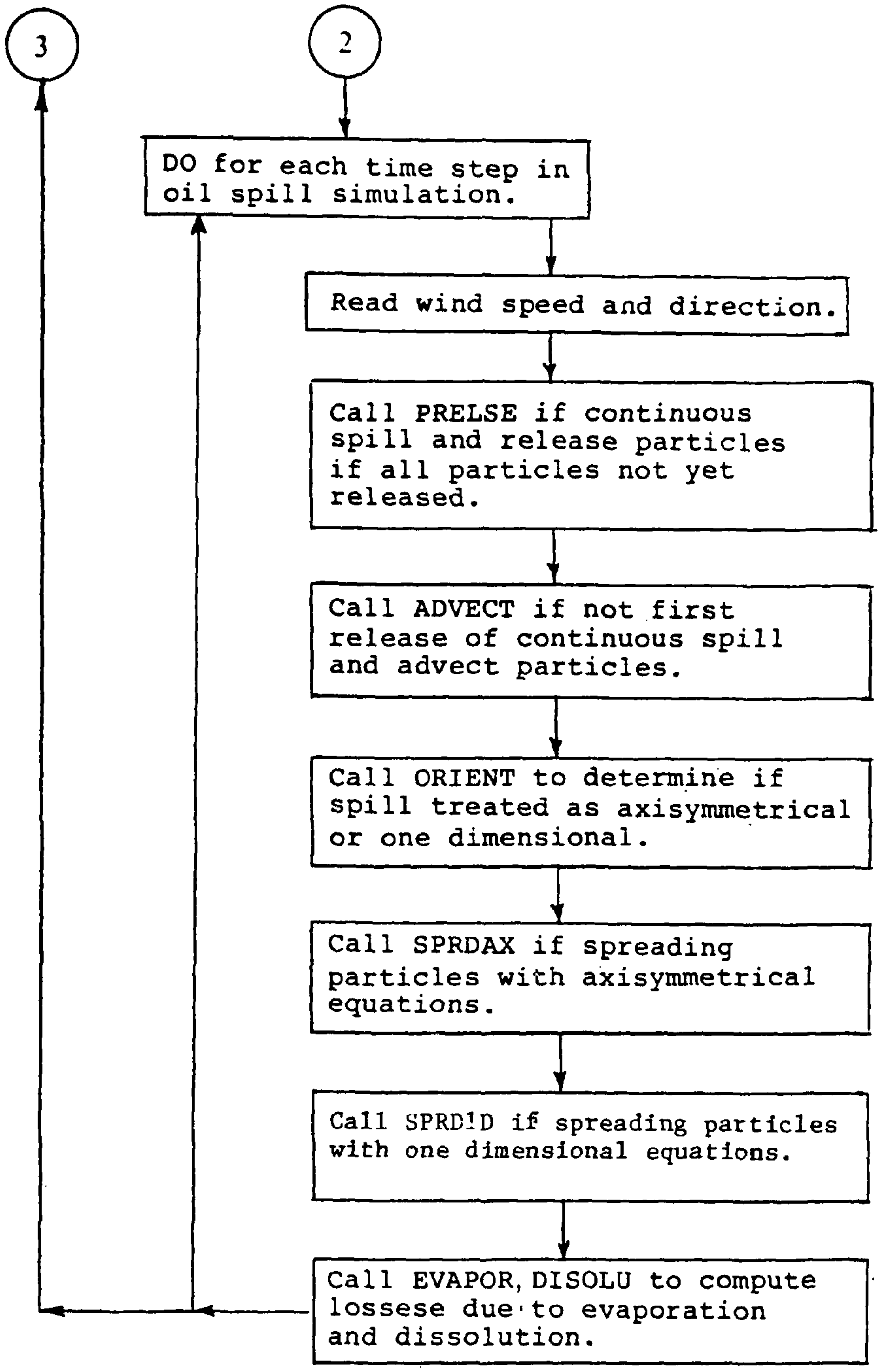


Figure 1. Block diagram of computer simulation model

information describing the lake and river grid schemes is found in Chapter II (with variable definitions in Chapter III).

Next, the data describing the river geometry is read. This information includes: 1) branch start and end cross sections, 2) cross section locations, orientation and connection sequence, 3) points describing cross section geometry, and 4) boundary grid boxes for river and lake shorelines.

The detailed procedure for creating and organizing the river data can be found in Volume II. It is important to realize that the river is organized into a series of branches. Each branch covers a specified stretch of the river and contains a number of cross sections depending upon available field data and accuracy requirements of this computer model.

Spill Data and Spill Type

The information which describes the actual spill is now provided. This data controls the size of the spill, the number of particles used to represent the spill, and the time scales for both the duration of oil spill simulation and spill duration. Also, the coefficients and constants used in the spreading, evaporation and dissolution phases are read. All of this data is user dependent. This implies that the user has the option of locating a hypothetical or real spill anywhere in the lake or river system with the desired physical properties. The spill simulation time step and spill duration are used to establish the spill type. If the spill duration is greater than half of the simulation time step, the spill will be considered continuous. Otherwise, the spill will be considered as an instantaneous spill. Details for releasing particles as a continuous spill will be discussed shortly.

I.2 River and Lake Computations

Updating Flow Conditions

The computer model has the capability to re-compute the depth-averaged surface velocities in both the lake and river at a specified time interval. The interval is the time step used in the river unsteady flow model. Its magnitude is dependent upon the need for updated flow conditions over the course of the spill simulation. For example, the flow conditions may be updated every 3 hours in a total simulation of 24 hours. At the time interval, stage and discharge conditions for all nodes in the unsteady river model (Thomas, 1984) are read. Subroutine NDCONV converts this information into the stage and discharge boundary conditions for each branch of the river. The lake circulation model requires the stage and discharge at the beginning of the first river branch (the lake-river interface) as boundary conditions for computing the correct stream function values.

Lake Circulation

Subroutine RLID is next called to calculate the stream function values, ψ, at the grid corners of the lake grids. The lake depths, initial stream function values, and meteorological data (wind speed, direction and location of meteorological station) are required input into this routine. The depths are given for all grids comprising the lake and are read in through subroutine RGRID along with various additional parameters (Chapter III). Initial stream function values are read using subroutine INIT for these same grids in addition to the grids needed to maintain the "no-flow" into the shoreline boundary condition (Chapter IV). Finally, the wind data must be supplied at the unsteady flow model time interval.

Ice Data

Data describing the location and extent of ice in both the river and lake is read in next. In the river, the cross section ice information is used to calculate the ice cover effects when computing the streamtube velocities by increasing the hydraulic radius. In the lake, the ice region data serves as an index for handling the shear stress term in the governing equations. For both the river and the lake, this information describes the regions where ice is encountered in order that the proper spreading and advection equations may be applied.

Ice Stress and Wind Stress

If an ice cover is present, there will be no wind stress. However, an additional shear stress is present due to friction on the underside of the cover. To index the presenc of an ice cover, two arrays are initialized in RLID for each grid box in the lake. One array ZWND (I,J), is set to either one or zero depending upon whether the ice cover is or is not present. The other array, FR(I,J), represents the drag coefficient in the quadratic drag law. Without an ice cover, the drag coefficient equals the bed drag coefficient only. With an ice cover, the ice drag coefficiewnt is added to the bed drag coefficient.

Lake Boundary and Initial Conditions

The stream function values and depths are initially set in the model for a reference discharge with the condition that inflow equals outflow. If the initial stage and discharge read from the river unsteady flow model is different from these reference values, the stream functions and depths must be changed to reflect the change. The depths merely require the increase or decrease in size depending upon whether the new stage is higher or lower. However, even after a quick adjustment to the stream function values, the circulation model is run for a minimum of twenty, one-hour time steps to obtain the quasi-steady state stream function distribution at the initial flow condition.

These quasi-steady state stream function values are saved for later computation of lake velocities for the initial period. The stream function output is controlled by subroutine OUTP. When the boundary conditions for the river branches are again updated by the river unsteady flow model time interval, the stream function values must be updated as well according to the new boundary condition at the lake-river interface. However, a smaller number of one-hour time steps in the update interval (3 to 6 hours) is needed instead of 20 hours.

Figure 2 gives a clearer interpretation of the method of 1) reading initial stream function values, 2) running the lake circulation model for twenty, one-hour time steps, 3) using the last set of stream functions, calculate the lake velocities and use them from initial spill time up to the first update of flow conditions, 4) re-reading boundary conditions at the update interval, and 5) re-computing stream functions and velocities to be

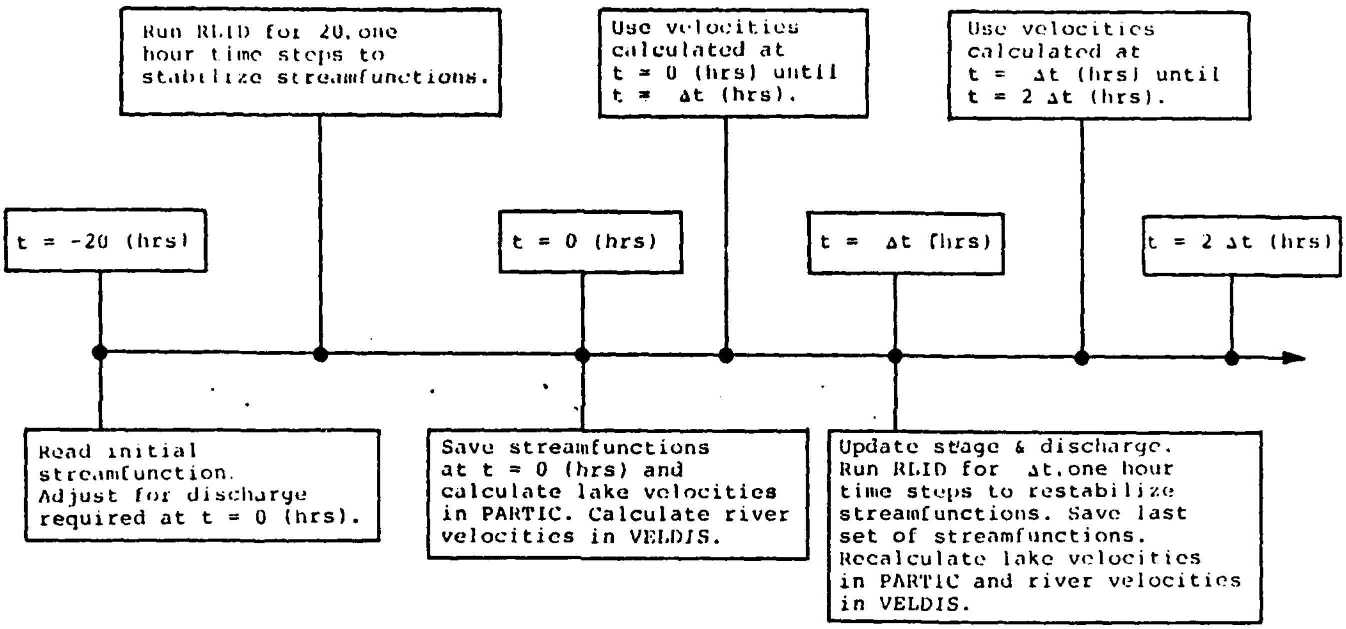


Figure 2 Time line for computing and re-computing stream function and lake velocity distributions

used up to the next update of flow conditions.

Lake Velocities

Once the stream function values are known for each grid box, the grid box velocity is computed in subroutine PARTIC. It is necessary to reread the bathymetric data to update the depth array before use in PARTIC. Subroutine UPDATE reads the current stream function values prior to calculation of lake velocities. Depth-averaged velocities are calculated for every grid containing a stream function value. The model first computes the transports M and N between adjacent values of stream functions as shown in Figure 3. Then, the velocity components are computed at the transport points and shifted back to the defined stream function point for that grid. Finally, a four point average is taken using velocities at all corners of the grid and assigned to the grid center.

River Velocities

The depth-averaged velocities for the river are calculated in subroutine VELDIS. Using the streamtube approach, velocities are calculated and assigned coordinates corresponding to the center of each streamtube. A velocity is then assigned to the grid box in which the coordinates lie. This procedure is carried out from one branch to the next for each cross section in a branch. A predetermined number of interpolated velocities are next calculated at equidistant points between consecutive cross sections in the same streamtube. These velocities are assigned to the grid boxes in which they lie. Once the interpolations have been completed for all streamtubes between all cross sections, the river is scanned for grid boxes requiring a velocity. Starting

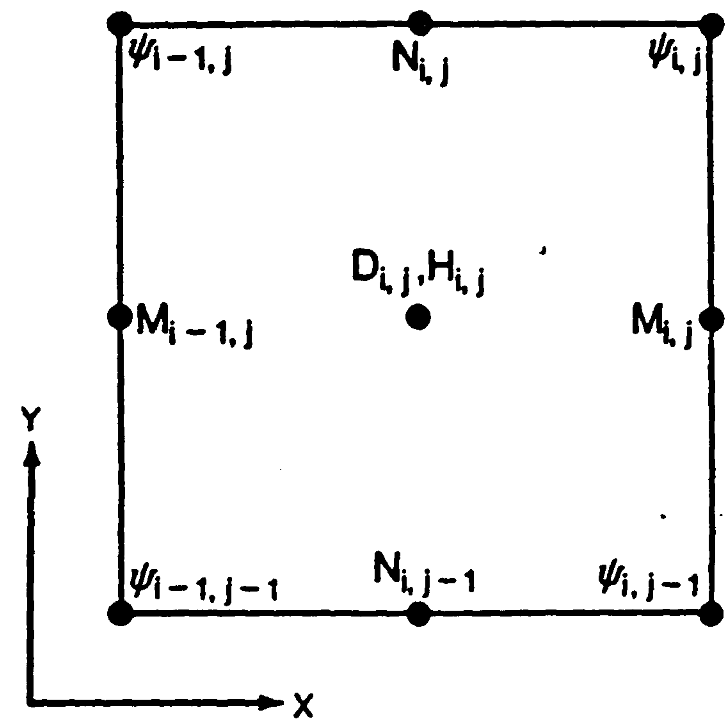


Figure 3 Position of variables in finite difference grid of lake circulation model (Schwab, et al., 1981)

from the beginning of the river, velocities in the adjoining grid boxes above, below, and on either side of a grid without velocities, are averaged and assigned to that grid.

I.3 Slick Transformation

Wind Component of Drift Velocity

The wind component of the drift velocity is considered to have the same magnitude and direction over the entire lake-river region. However, the wind data is input for every time step in the spill simulation thus providing more flexibility in its use. By inputting the predicted or expected wind conditions along the path of the slick, the wind information is used only in the area in which it pertains despite its overall constancy.

Continuous Spills

If the spill is determined to be continuous, subroutine PRELSE is called to control the release of oil particles. The logic in PRELSE used to model a continuous spill considers the total spill as a series of particle releases. In this way, the oil can be released in the model continuously but the volume of oil released up to a point in time can be spread as if it were an instantaneous spill. The number of releases is equal to the spill duration divided by the simulation time step. The release of particles is done uniformly in time over the spill simulation time step.

The actual sequence used is as follows. At the first time step of the oil spill simulation, a group of particles are released uniformly in time, advected (in PRELSE) and then spread according to the total volume they represent. When the subsequent calculations are completed for that time step, another group of particles is released and advected (in PRELSE) for the next time step. The particles which were released prior to this time must be advected as well. This is done using another subroutine (ADVECT) which will eventually be the sole means of advecting the particles once the entire continuous spill has been released. Furthermore, when the spreading is computed, the entire spilled volume up to that time is used, not just the volume of the particles released last.

Advection

Open Water

The source of the advective wind velocity has already been described. The appropriate water velocity to use depends in which grid a particle is currently located. All points in a grid are considered to experience the velocity assigned to that grid so if a particle is situated anywhere inside of a grid, that grids' velocity is used when computing the overall drift velocity for the particle.

Ice Covered Region

If a particle is in an ice covered region, the first condition for advection under ice to occur, is that the threshold velocity be exceeded.

Spreading in Open Water

Axisymmetrical Spreading

If the criteria for using axisymmetrical spreading are met, subroutine SPRDAX will perform the necessary computations for this type of spreading. Use of the axisymmetrical equations is accomplished by first dividing the slick into eight segments each encompassing $\pi/4$ radians. This allows for the probable distortion in the slick from a truly circular slick. Each pie segment will contain a number of particles depending upon the location of the particles in the slick. The particles in each segment will be spread radially according to a computer spreading rate. Since the spreading equations are based upon a circular slick, the volume used to compute the spreading rate equals eight times the volume of oil in the segment. In this way, the correct magnitude of the spreading rate is computed.

The spreading rates computed are considered directly applicable to particles at the mean radius of the segment. The magnitude of the spreading rate for other particles is weighted according to the ratio of the particles position relative to the slick centroid and the distance to the mean segment radius.

One Dimensional Spreading

If the criteria for using one dimensional spreading are met, subroutine SPRD1D will perform the necessary computations for this type of spreading. The technique used to model the one dimensional case is similar to the axisymmetrical case except that the slick is broken up into strips instead of circular segments. These strips lie perpendicular to the major axis of the slick. The major axis is found by using the moments of inertia. Each strip is one grid box long in the direction of the major axis and as many grid boxes wide in the direction of the minor axis to accommodate the particles in the strip.

Spreading rates are computed independently on each side of the strip centroid. Since the one dimensional equations apply to a symmetrically shaped strip, the volume used to calculate one sides spreading rate equals twice the volume actually present. In this way, the correct magnitude of the spreading rate is computed and deviations, in the slick shape from a symmetric shape along the entire slick centroid, can be accounted for. Again, the spreading rate is applicable to particles at the mean strip width on one side of the slick. The spreading rate for the remainder of the particles is weighted according to the ratio of the distance of the particle from the strip centroid and the mean (upper or lower side) strip width.

Spreading Under Ice Cover

When an ice region is encountered, the choice of using open water spreading or spreading under an ice cover first depends upon whether or not the oil is still leaking from its source. No spreading under the ice cover will occur for an instantaneous spill or once the continuous leak stops. If the leak is in progress and conforms to an axisymmetrical shape, the segments under ice will spread.

Weathering Effects

Oil losses due to evaporation and dissolution are computed in subroutine EVAPOR and DISOLU respectively. Once the evaporative loss has been computed, the representative oil volume of each particle is reduced. The amount of oil losses due to dissolution are small compared to those from evaporation and this loss neither significantly changes the oil volume nor significantly changes the computed spreading rates. However, the amount of dissolved oil is calculated and accumlated for use in assessing the impact of the oil on the marine environment.

I.4 Shoreline Conditions

During the advection and spreading phases, oil particles can be moved beyond the boundary grids describing the river and island shorelines. Therefore, after completion of either phase, a check is made to determine if a particle has been moved onto a land grid. Arrays are used to keep track of land trapped particles so that upon entry into subroutine BOUNDR, the reaction of the oil with the shoreline can be assessed.

The logic behind BOUNDR is rather straightforward. Referring to Fig. 4, if a land trapped particle is found below shore 1 or above shore 2, it is moved to the first land grid on the appropriate side of the river. If the land trapped particle does not meet the above condition, it must be on an island. In that case it will be moved to the closest island boundary grid. Once all particles are moved to the river and/or island boundary, the rejection rate is used to re-entrain excess particles into the river. All rejected particles are assigned to the centroid of the closest water grid.

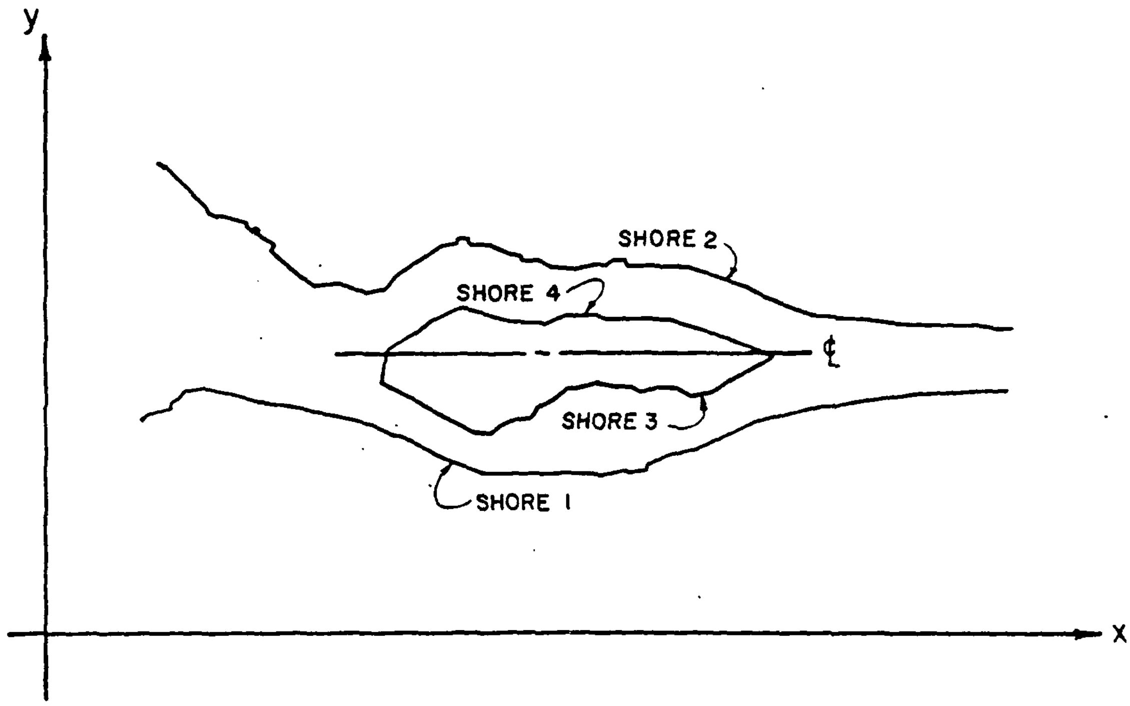


Figure 4 Indexing for shorelines

Islands

Although not given special attention up to this point, the overall model does have the capability to deal with islands as follows:

1. Island grids in the lake are treated as shallow water since RLID does not have the capability to handle the proper boundary contraints.

2. The streamtube method employed in the river can handle the main channel division around one island when computing the river velocities. Additional islands which would cause the main channel to further divide into sub-channels will be treated as shallow water and later have their corresponding grid box velocities set to zero.

3. The method used in BOUNDR to move particles into land boundary grids is limited to four shorelines. This means that when there are several islands in the same river cross section, only one island can be correctly modeled. There is no limitation if islands are in series with regards to the x axis. So, when assigning boundary grids, the most significant island should be selected for shores three and four.

4. Using oil particles is convenient since the slick can easily divide when proceeding around an island. However, the model will only spread one slick at a time. Therefore, if the slick does separate into two patches around an island, prior to termination of spreading, oil particles will be shifted to one side of the island where the appropriate spreading techniques can be used. Afterwards, the oil particles are shifted back again.

CHAPTER II

THE GRID SYSTEM

Since the model tracks the movement of oil on the water surface, it is necessary to have the capability for quick identification of the slick position. Both river and lake have their corresponding surfaces limited by finite boundaries. The computer must be able to recognize these limits for purposes of determining where to assign current velocities, where the oil will move, and when it will hit the shoreline.

A systematic technique was developed to reference any location on the two dimensional surfaces of either the lake or river. This technique requires that a fixed grid network be superimposed over both water bodies. The grids in the lake and river serve both similar and dissimilar functions. The similarity is velocities will be assigned to the grid centers for use in computing the advection of the slick and indexing the grid boxes controls where oil will hit the boundaries. The dissimilarity is due to separate computations and model structure of the lake circulation model when compared to the calculation of river currents.

II.1 The Coordinate System

The grid network is laid out according to a Cartesian system. The placement of the grids follows the x and y axis of the Cartesian plane. As shown in Fig. 5, for the Lake St. Clair-Detroit River System, these axes are originated from a pre-selected plane where the lake and river connect. The

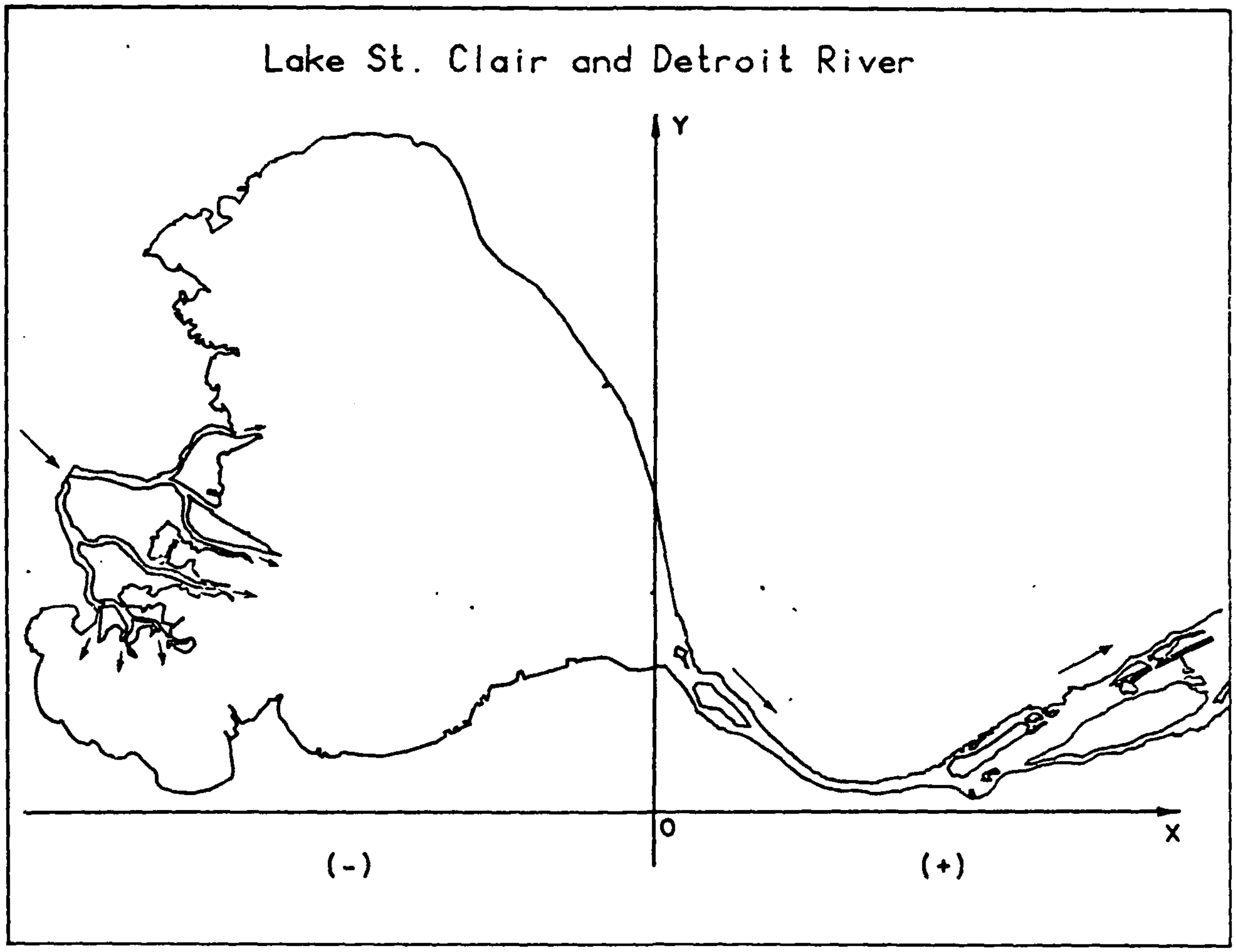


Figure 5 Location of axes in lake-river system

lake will be to the left of the y axis and the river will be to the right. If any shoreline is visualized as a string of x,y coordinates, the scheme used here will make all x coordinates for the lake negative and all x coordinates for the river positive. In either case, the y coordinates are always kept positive since the computer logic dictates the x axis as the lower reference line. The river is used to set the orientation of the Cartesian plane. The x axis follows the major orientation along the length of the river. The relative position of the x axis along the y axis is established by leaving one row of grid boxes above the x axis before reaching the lake as shown in Fig. 6.

An important distinction must be emphasized between the lake and the river, since the lake circulation is computed separately from the river currents. The lake circulation model actually requires at least one layer of grid boxes all the way around the lake which are not in any way part of the lake shoreline. These boxes must border on the x axis at the bottom row and extend one column beyond the y axis at the far right side as indicated in Fig. 6. The result is an overlap in river versus lake grid boxes at the lake-river interface. This will not cause any confusion in the model because the extra column of boxes past the y axis is only used for computational purposes in the lake circulation model. They will not have assigned velocities or serve as any part of the lake during the oil spill simulation.

II.2 Grid Sizes

The grids must be square and for purposes of maintaining flexibility, the size of the lake grid must be exactly divisible (to a whole number) by the size of the river grid. The implication here is that a lake grid and a river

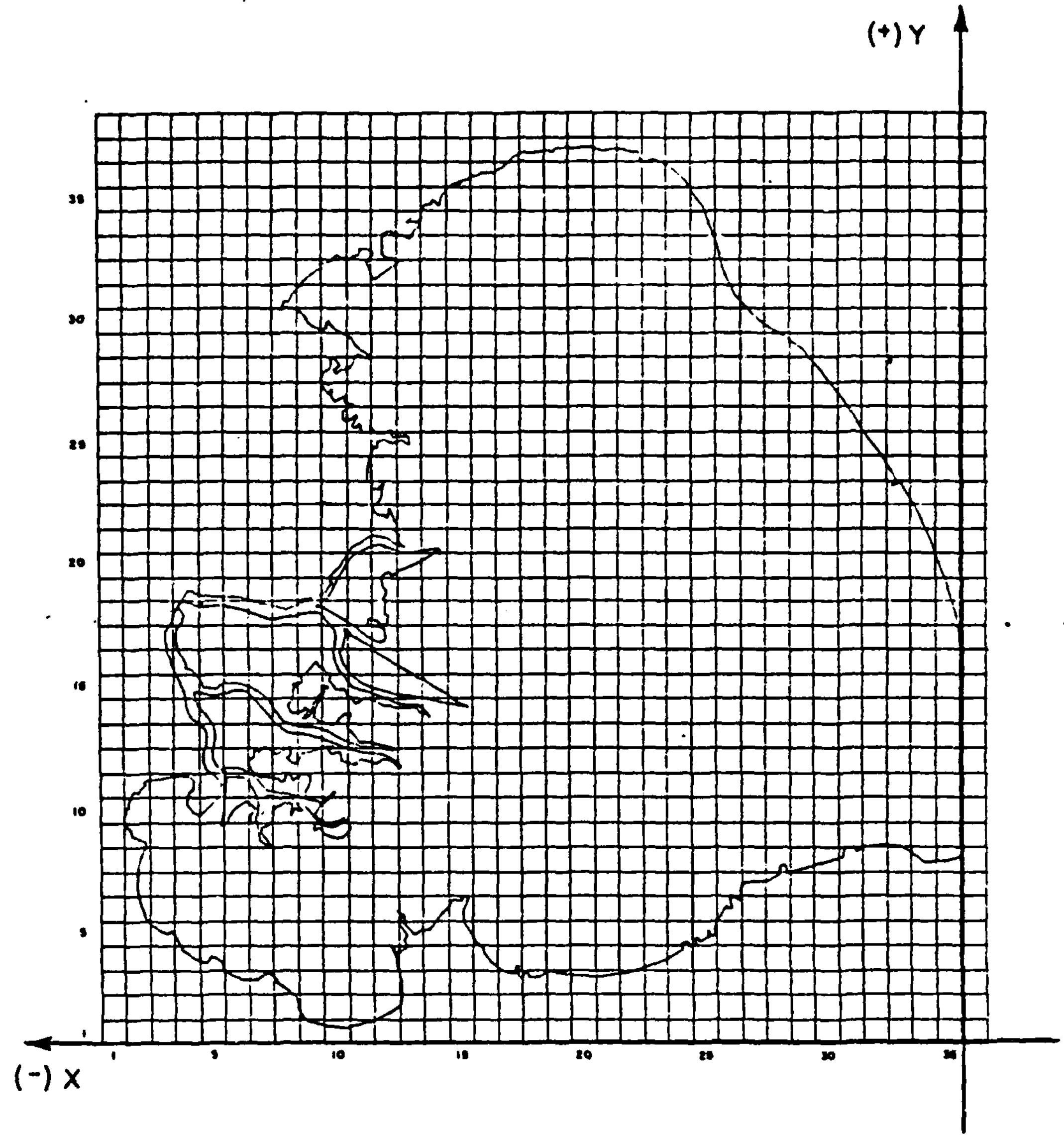


Figure 6 Lake grid boxes relative to x and y axes

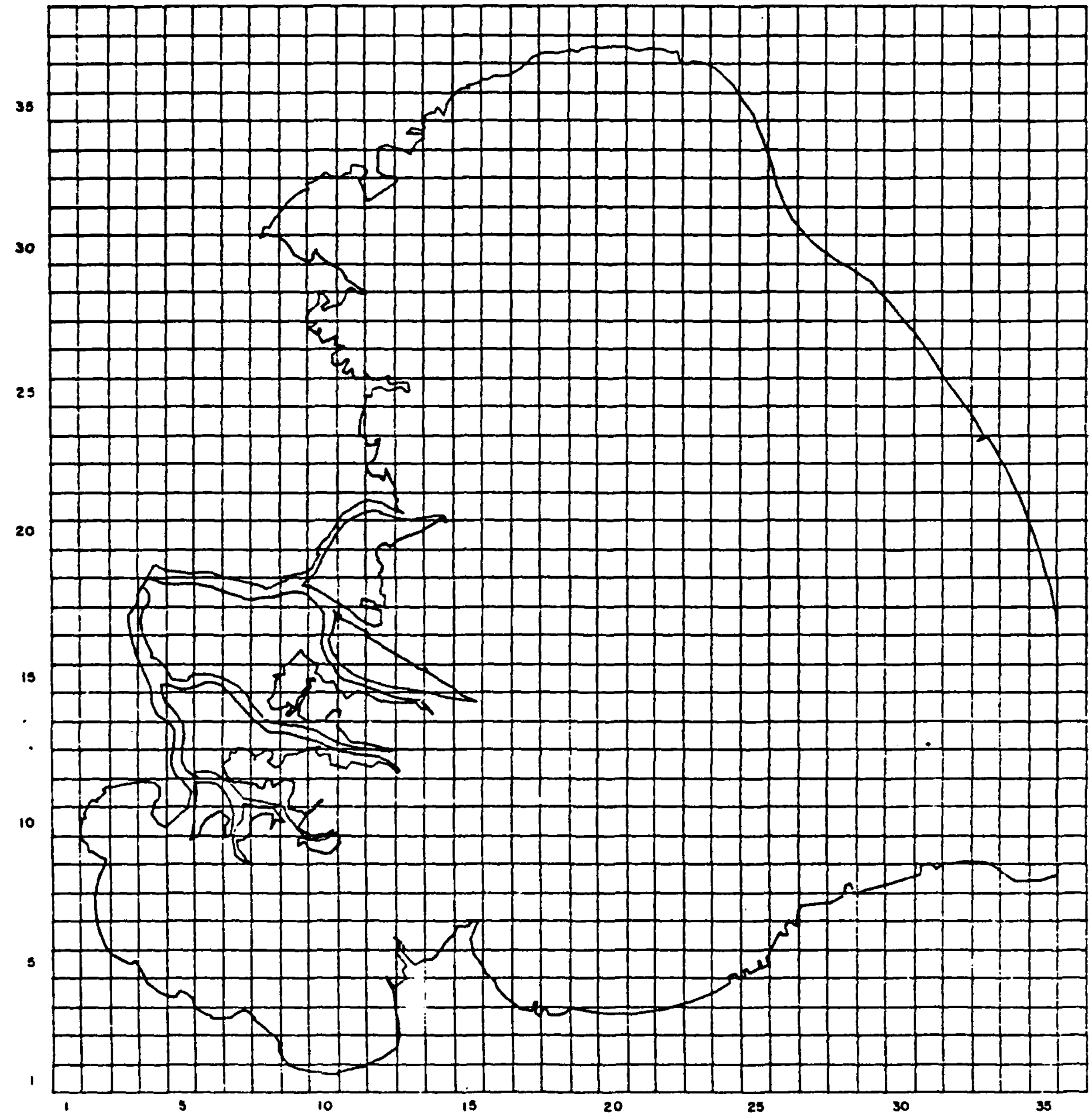
grid can be different in sizes as long as the lake grid is equal or larger in dimension than the river grid. The qualification of grid sizes is needed due to the logic used for determining whether a grid is located in the lake or the river and due to the fact that the lake grids are typically larger due to the greater surface area covered.

II.3 Grid Indices

Generally, both the lake and the river are described by grids in the same manner. Once the shoreline configuration for each water body is established and the grid is superimposed over them, simple counting is all that is required to identify any particular grid. Counting the x grids starts from one column to the left of the lake and continues until the end of the river is reached. All y grids are counted upwards from the x axis regardless of whether they fall in the river or lake. The grid index figures used in the Lake St. Clair-Detroit River area are supplied in Figures 7 through 14, for references. Figure 7 defines the grid system for the lake, Figure 8 is the index to Figures 9 through 14, which in turn defines the grid system for the river. These figures shall be used when attempting to locate the site of the oil slick as well as locations of oil contaminated from the output.

II.4 Shoreline Boundaries

The shorelines are schematized according to grid boxes described above. For every grid in the x direction, there are two corresponding y grids on the water side; one establishes the upper shoreline and the other establishes the lower shoreline as shown in Figure 15. Island shore grids are counted on the land side using the same method to denote upper and lower limits. All grid



Grid Size 4000' x 4000'

Figure 7 Grid index for Lake St. Clair (x grid range 1 to 35, y grid range 1 to 38)

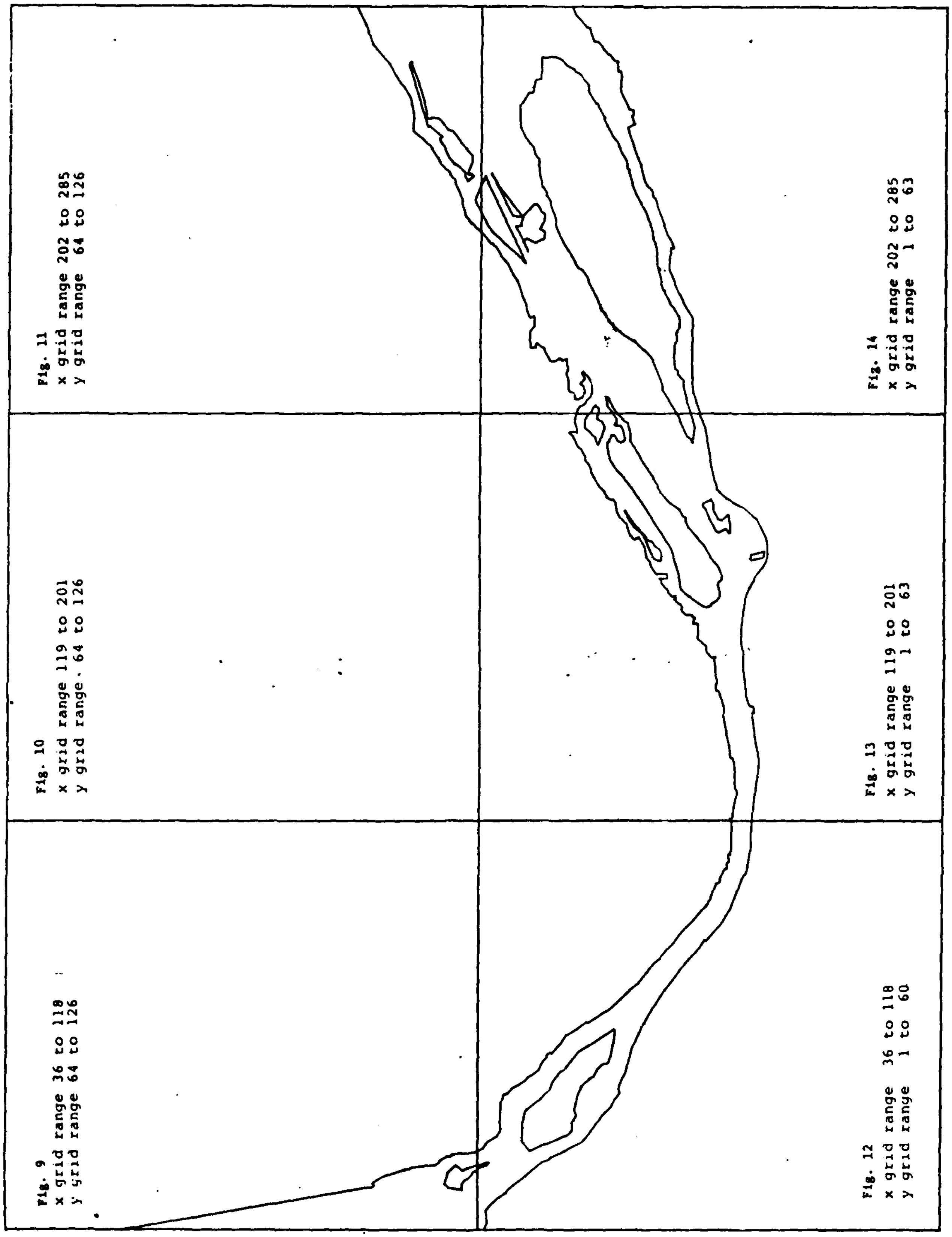


Figure 8 Index of Figures 9 through 14

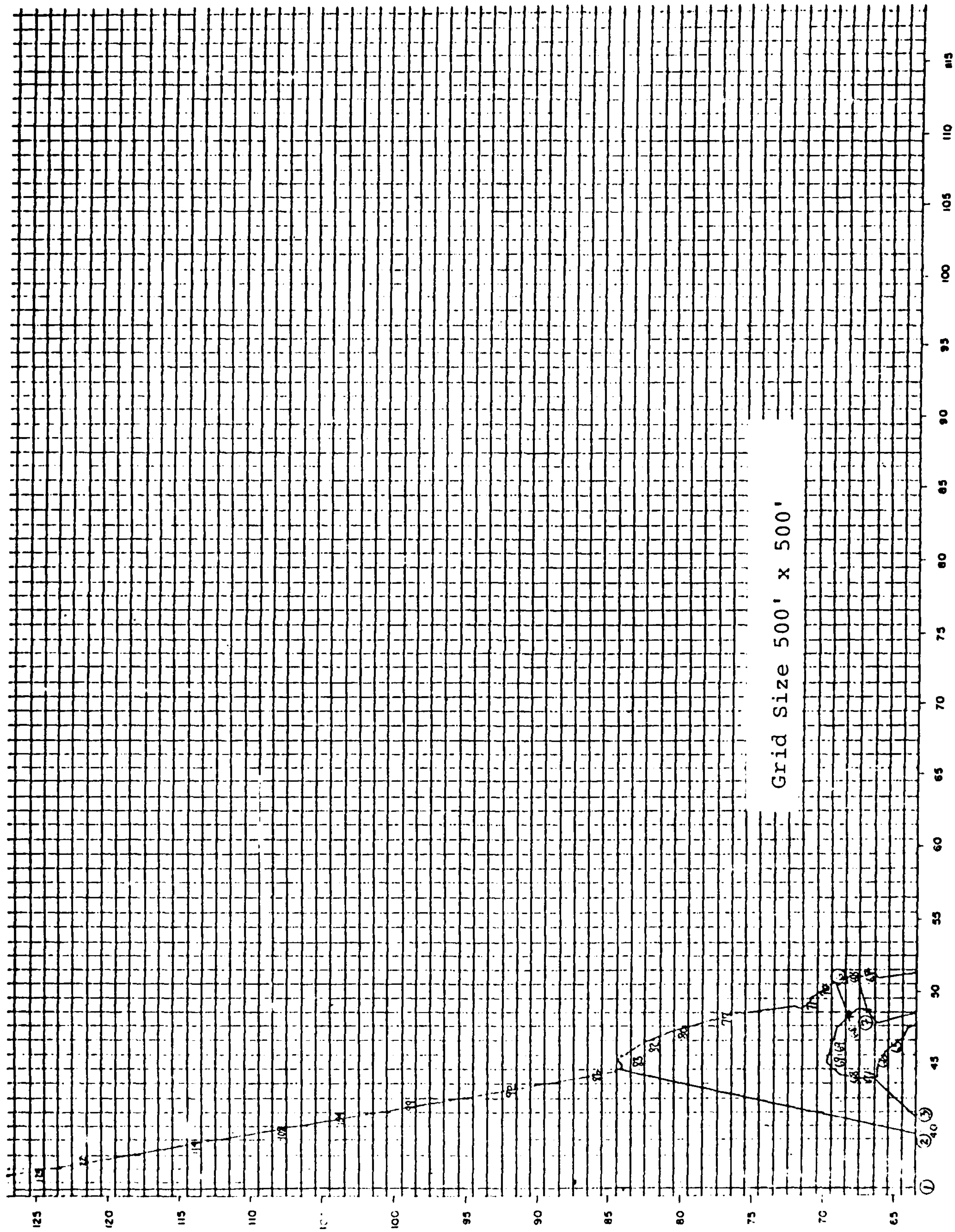


Figure 9 Grid Index for Detroit River (x grid range 36 to 118,
y grid range 1 to 63)

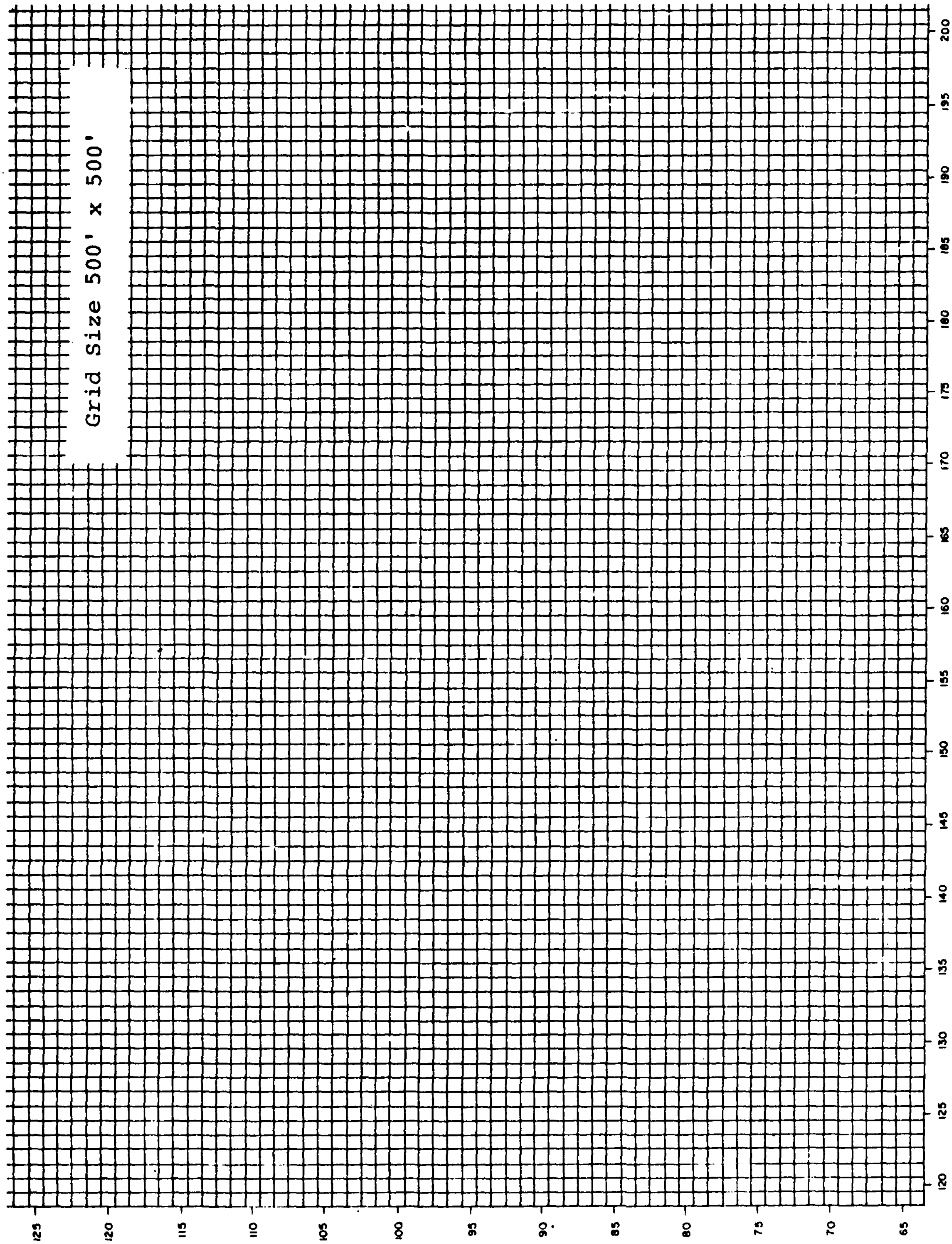


Figure 10 Grid Index for Detroit River (x grid range 119 to 201,
y grid range 1 to 63)

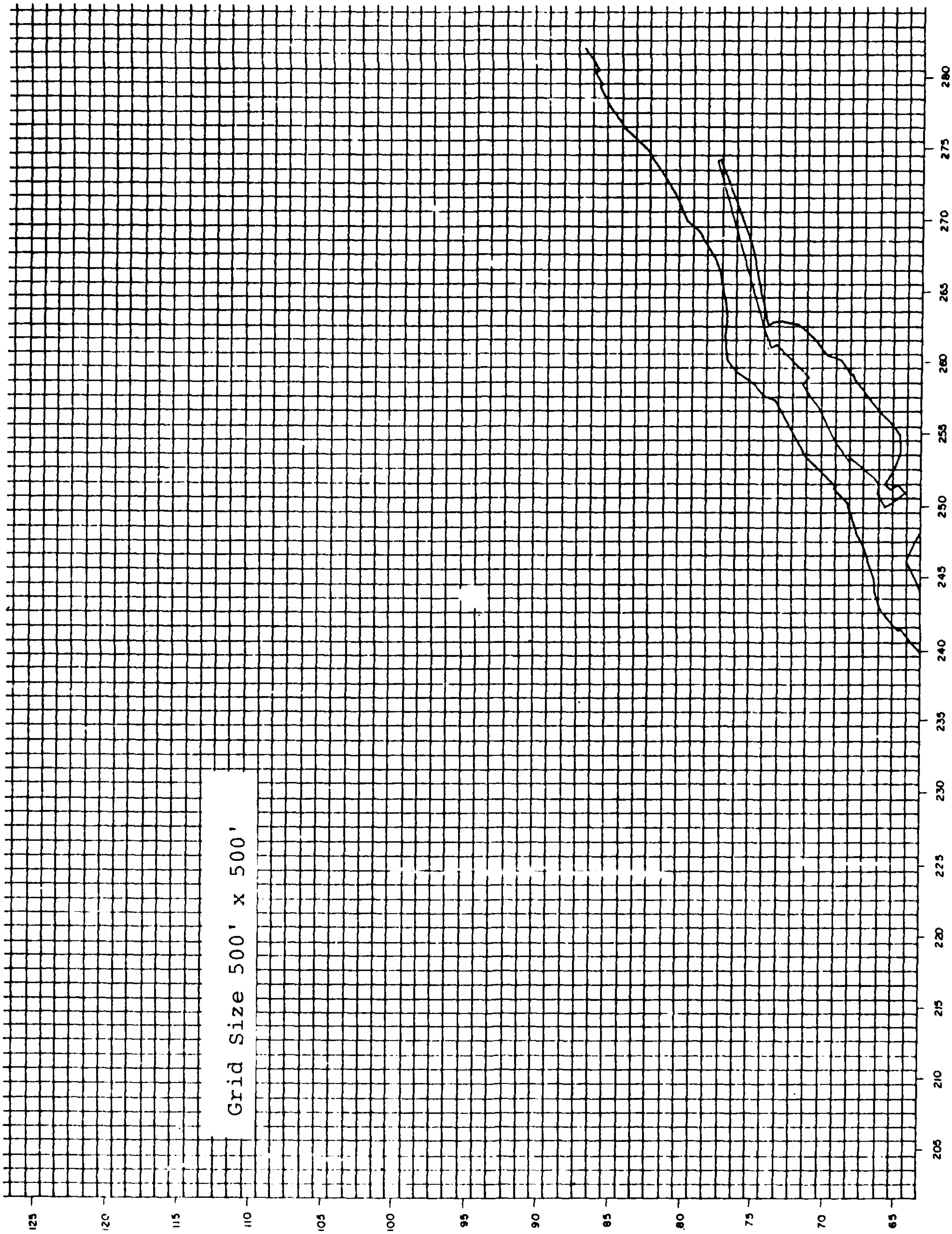


Figure 11 Grid Index for Detroit River (x grid range 202 to 285,
y grid range 1 to 63)

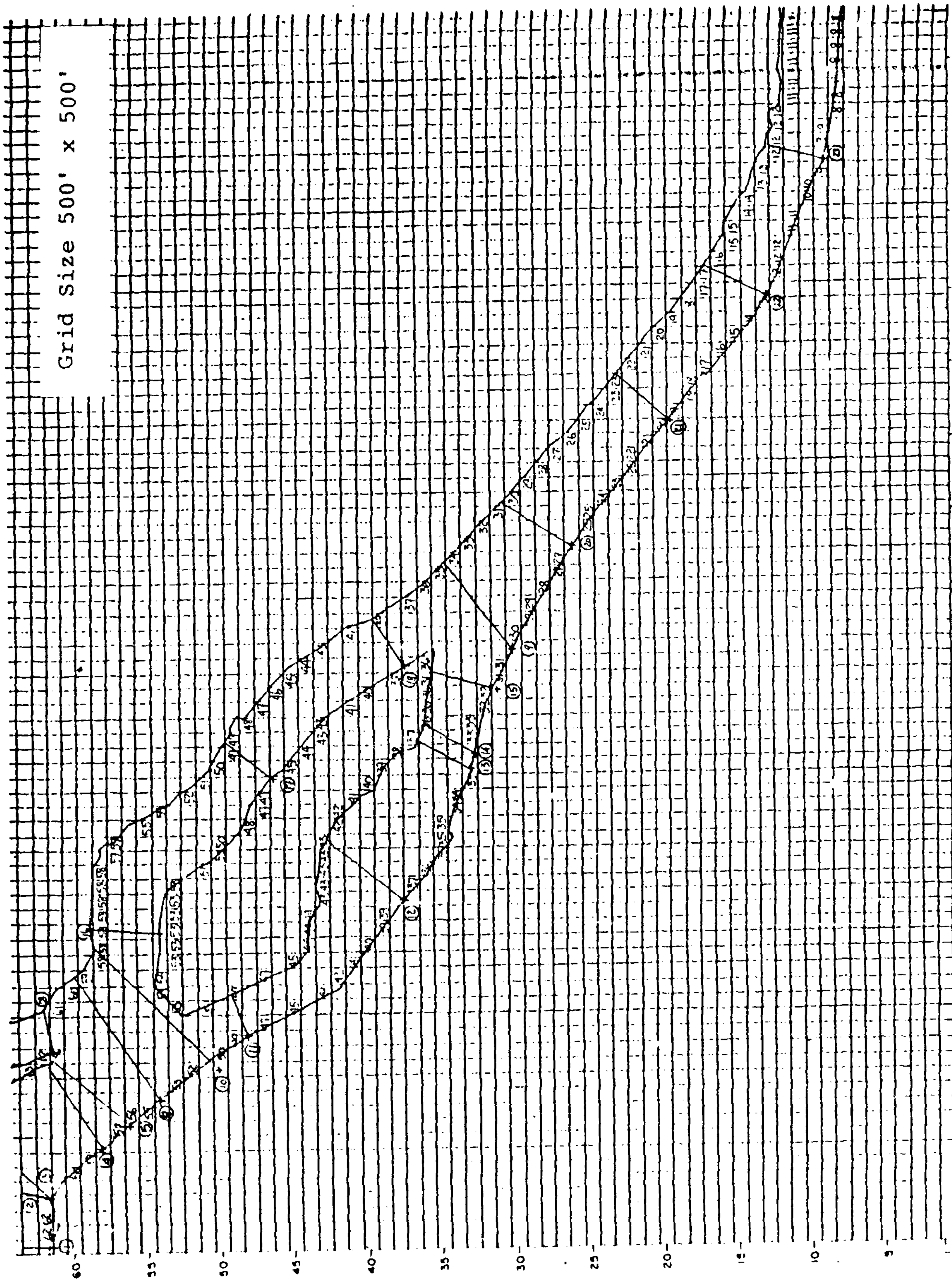


Figure 12 Grid Index for Detroit River (x grid range 36 to 118, y grid range 64 to 126)

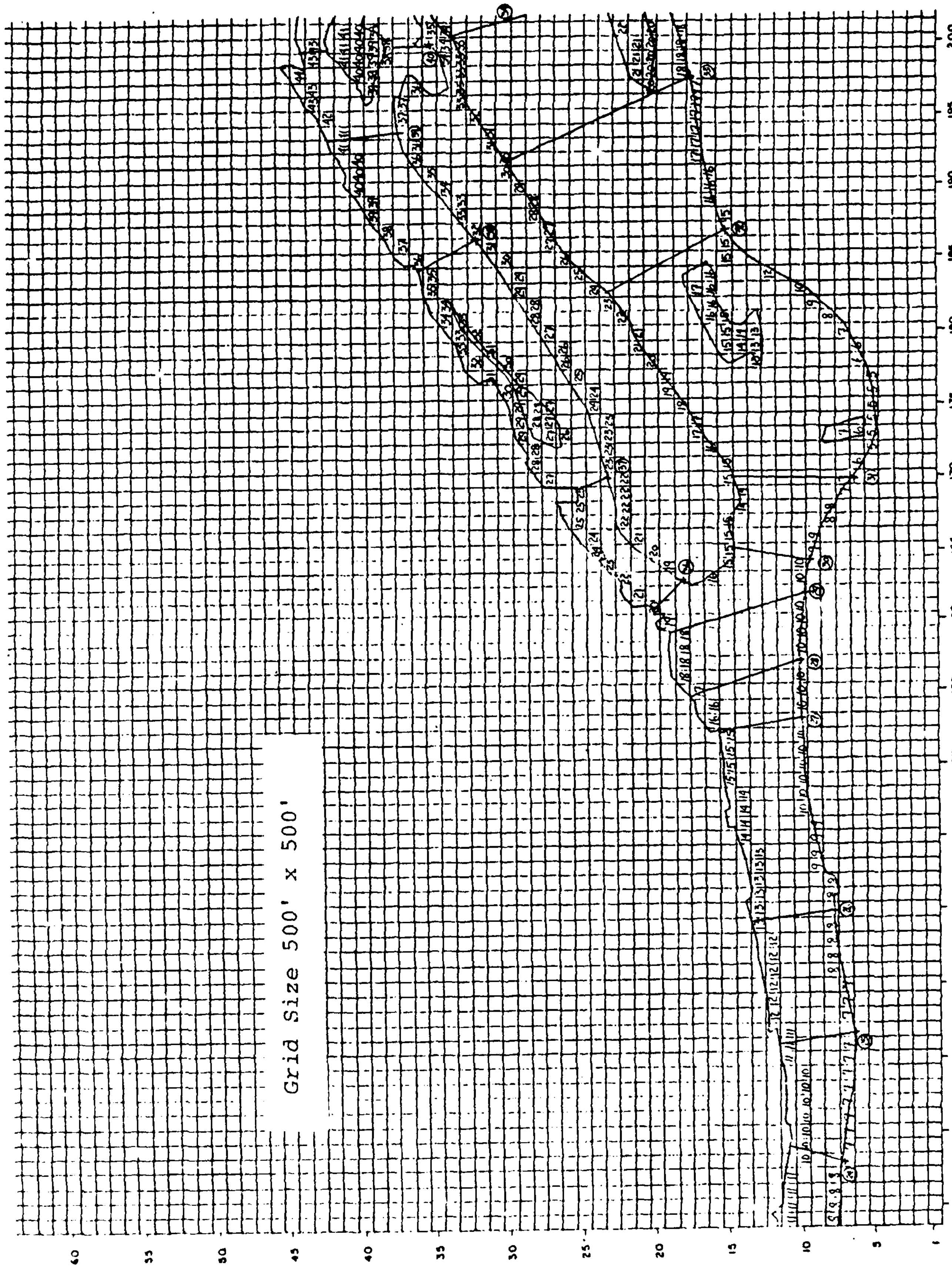


Figure 13 Grid Index for Detroit River (x grid range 119 to 201, y grid range 64 to 126)

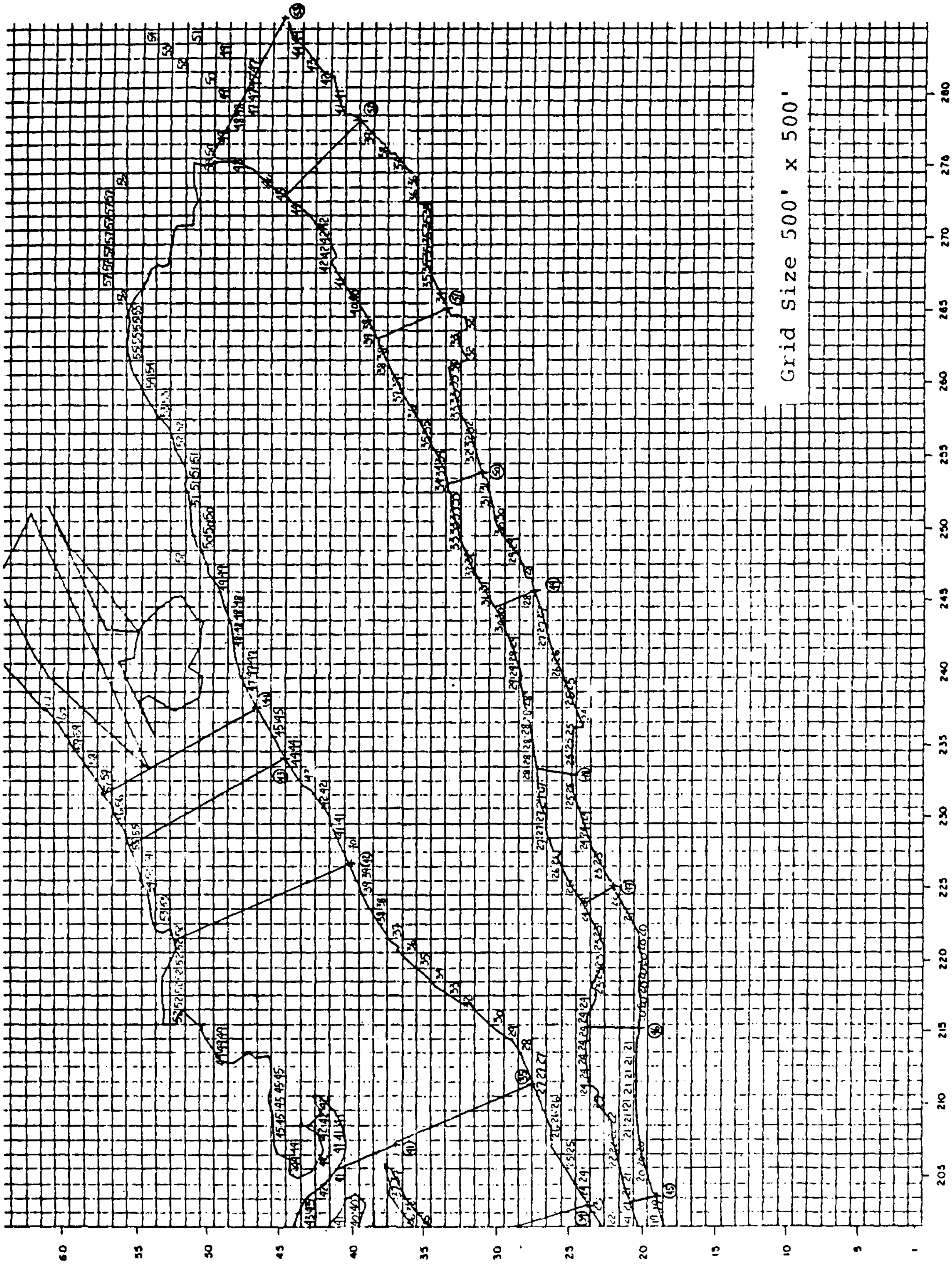


Figure 14 Grid Index for Detroit River (x grid range 202 to 285, y grid range 64 to 126)

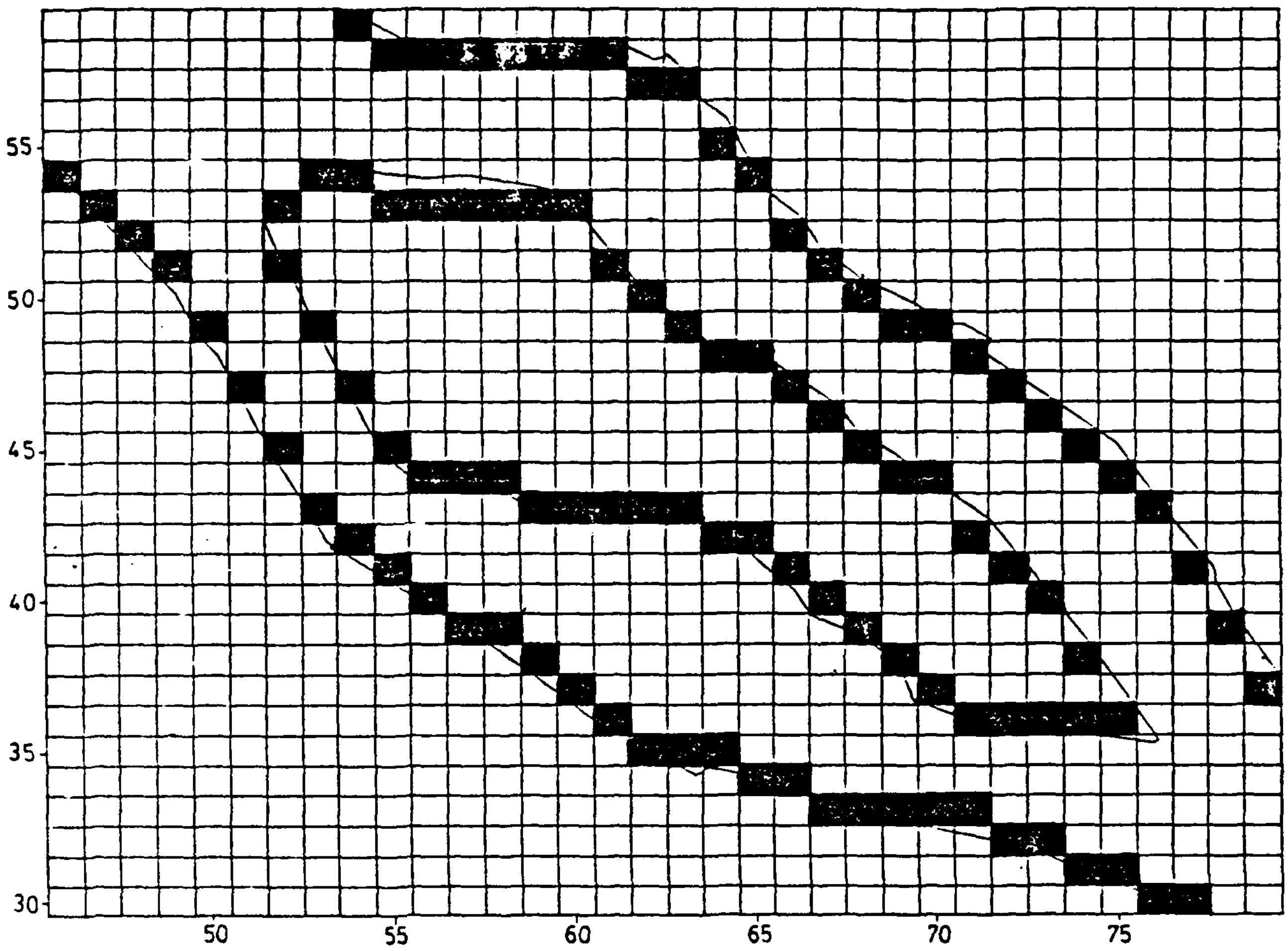


Figure 15 Portion of Figure 12 illustrating boxes selected as shoreline grid boxes

boxes contained between these limits, excluding those between island grids, constitute the lake or river water surface area. The data files set up for Lake St. Clair and the Detroit River are further detailed in Chapter III.

Indexing the shorelines as described requires some preliminary work. Once the axes are established on appropriate charts (National Oceanic and Atmospheric Administration, National Ocean Survey, Chart 14853, 8th Ed., April 14, 1979 for Detroit River and Chart 14850, 41st Ed., December 24, 1983 for Lake St. Clair), a scaled grid sheet can be placed over the charts and grids can be counted accordingly to locate the shoreline. This is sufficient for acquisition of the boundary grids, yet for graphical purposes and to facilitate easier interpretation of the oil spill model output, grid index figures of the type shown in figure 8 and figures 9 through 14 should be created. These figures are produced using a software (Petroski and Glebas, 1985) developed to record shoreline coordinates directly from the charts. The axes must still be established on the charts, although such problems as scaling, multiple charts (Chart 14853, No.'s 2-14), and translation and rotation from one chart to the next, can easily be handled by the computer. Furthermore, permanent indexing figures can be drawn by plotting the shoreline coordinates and the grid system together.

CHAPTER III

INPUT DATA FILES

There are three categories of input; the first is data for the computation of lake currents, the second is data for the computation of river currents and the third is miscellaneous data describing oil slick characteristics. Some of these categories overlap to a certain degree, as will be pointed out, although for the most part they are rather distinct. The river data is more explicitly explained in Volume II so it is only broken down into its components in conjunction with a complete sample data set in this chapter. The lake data will briefly be discussed here in Chapter IV, since most details are covered in the report by Schwab and Sellars (1980). Miscellaneous data will simply be defined as the need arises.

A data file may or may not contain a mixture of some river, some lake, and some miscellaneous data. The resason for this is to break the data up into fixed data which the user need never touch and data which must be adjusted from one spill to the next. The following is a list of the required input files with their contents.

/ Filename /	Type /	Unit /	Contents /
LDETR.GEO	FIXED	1	River geometry, cross sections and branch geometry
LDETR.ICE	ADJUST	5	Ice parameters, ice regions and lake ice thickness.
LDETR.FLW	ADJUST	7	Water level and discharge from unsteady flow model.
LDETR.BND	BOTH	8	Half life assignments to shore grids.

LDETR.SPL	ADJUST	12	Oil parameters, spill location and wind component of advection.
LAKEWIND.DAT	ADJUST	10	Meteorological data for lake.
LAKEBATH.DAT	FIXED	13	Lake bathymetry and parameters.
LAKEINIT.PSI	FIXED	14	Initial streamfunction values in lake.

The files are generally broken up into blocks and cards. A block covers a broad classification of data which may contain one or more card types. A card type is one line of specific data which is sometimes repeated. (Example: Block 5 in LDETR.GEO has Cards 1 and 2 where Card 2 is repeated as many times as needed.) By inspecting the example of a card and comparing it to the complete sample data set at the end of this chapter it is easy to see how the entire file comes together.

Most of the data read into the model is in list directed I/O (free format). If column numbers are shown, the data must be formatted accordingly, otherwise it is necessary to put only one space or comma between each number in a card.

III.1 River Data Files

LDETR.GEO

The LDETR.GEO file contains the complete geometric description of the river. This also includes shoreline grid boxes and grid control parameters for the lake. The file consists of five blocks of information. All blocks are listed below with descriptions and corresponding components.

None of the data in this file needs to be adjusted from one spill to the next. The user may choose to add additional cross sections, change the number of branches, relocate shore grids, etc. However, the user is cautioned to consult Volume II before making the attempt.

LDETR.GEO: Block 1 (Branch and grid information)

Card 1 (Identification)

Example:

DETR Lake St. Clair and Detroit River

Variable Name	Type and Length	Column Number	Definition
WORD	A4	1-4	key word identifying river
TEXT	19A4	5-80	any text to describe the purpose of computer run

Card 2 (Grid control paramters for lake and river)

Example:

16 35 285 4000 500 7 -1.4E+05

Variable Name	Type and Length	Column Number	Definition
NBRNCH	Integer	--	number of branches
LGRIDX	Integer	--	number of x grids along lake
NGRIDX	Integer	--	total number of x grid boxes
DXL	Real	--	size of lake grid (ft)
DXR	Real	--	size of river grid (ft)
KINTM	Integer	--	number of velocity interpolations between cross sections in a streamtube
BEGLK	Real	--	x coordinate of lake grid origin (ft)

Card 3 (Division of cross sections into branches)

Example:

2 5 8 10 15 18 22 27 29 31 33 35 41 44 50 52

Variable Name	Type and Length	Column Number	Definition
LCSTSQ(I)	Integer	--	last cross section in each branch. Last branch - use second last cross section. There must be NBRNCH numbers (one for each branch.) If line is not long enough, continue on another card.

LDETR.GEO; Block 2 (Cross section location and connection information)

Card 1 (1 card for each cross section)

Example:

1 (250.,30500) 1.57079630 11 11 2 0

Variable Name	Type and Length	Column Number	Definition
J	Integer	--	cross section number (for checking)
CORDLB(I)	Complex	--	complex variable giving x and y coordinates locating cross section on reference shore (ft,ft)
SCTANG(I)	Real	--	angle (radians) cross section makes with positive x-axis
NSTUBE(I)	Integer	--	number of streamtubes at current cross section
NUMCON(I)	Integer	--	if all streamtubes continue to next cross section undivided = 11, if streamtubes divide into two channels from main channel = 12, if streamtubes from divided channel connect back to main channel = 21.
NFIRCO(I)	Integer	--	next cross section connecting to current cross section. For a divided channel around an island, this represents the first cross section connected to in the lower division from the main channel cross section.

NSECO(I)	Integer	--	for a divided channel around an island, this represents the first section connected to in the upper division from the main channel cross section (if no island = 0, if lower division complete and returning to upper division = 888, if both divisions complete and resuming main channel = 999.)

LDETR.GEO; Block 3 (Cross section geometry)

Card 1 (1 card for each cross section)

Example:

1 17 571.71

Variable Name	Type and Length	Column Number	Definition
J	Integer	--	cross section number (for checking)
NSLSCT(J)	Integer	--	number of sounding depths used to describe the cross section geometry
ZD(J)	Real	--	reference datum for cross section J from which the sounding depth is evaluated (ft)

Card 2 (as many cards as required to input NSLSCT(J) sets of YWID,Z)

Example:

1375.0 6.0 1675.0 24.0 3000.0 21.0 3575.0 16.0 4000.0 23.0

Variable Name	Type and Length	Column Number	Definition
YWID(I,J)	F8.2	--	distance from the reference shore to the J^{th} sounding depth in the I^{th} cross section (ft)
Z(I,J)	F8.2	--	J^{th} sounding depth for the I^{th} section (ft)

NOTE: Block 3 must be repeated LCSTSQ (NBRNCH) times (i.e. = no. of cross sections defined)

LDETR.GEO; Block 4 (Boundary grid boxes in lake and river)

Card 1 (1 card for each grid in x-direction)

Example:

7 4 12 10 11

Variable Name	Type and Length	Column Number	Definition
J	Integer	--	x-grid box number
IGRILB(J)	Integer	--	y-direction grid box number of lower river boundary for J^{th} x-grid (water side grid box)
IGRIUB(J)	Integer	--	y-direction grid box number of upper river boundary for J^{th} x-grid (water side grid box)
IGRILB1(J)	Integer	--	y-direction grid box number of lower island boundary for J^{th} x-grid (land side grid box)
IRGIUB1(J)	Integer	--	y-direction grid box number of upper island boundary for J^{th} x-grid (land side grid box)

LDETR.GEO; Block 5 (define grids having zero velocity in lake and river)

Card 1

Example:

76

Variable Name	Type and Length	Column Number	Definition
NZRVB	Integer	--	number of boxes to be assigned zero velocity

Card 2

Example:

9 10

/ Variable Name	/ Type and Length	/ Column Number /	Definition /
IZRBX(I)	Integer	--	x grid number of I^{th} box to have zero velocity
IZRBY(I)	Integer	--	y grid number of I^{th} box to have zero velocity

There must be NZRVB pairs of IZRBX(I) and IZRBY(I). Data may be continued to as many lines as needed.

LDETR.ICE

The LDETR.ICE file contains information identifying ice regions which the user will have to adjust as ice conditions develop. An ice region is a range of grid boxes containing ice. Ice regions in the lake must be specified first. An example is shown in Fig. 16 where an ice regions may be identified as extending from grid (15,7) to grid (18,12). The ice region then covers every grid from (15,7) to the upper shoreline of x column (15), all grids in x columns (16) and (17), and from the lower shoreline in x column (18) up to and including grid (18,7). An ice region may also be identified as grid (21,7) to (21,9). Then, the ice region will only extend between y grids (7) and (9) inclusive in x grid column (21). This information is used when determining if spreading and advection takes place under ice or on open water. For the lake model, the ice region data locates where wind stress is zero, where the frictional stress due to the ice cover must be considered, and where the lake depths must be adjusted for the thickness of the ice.

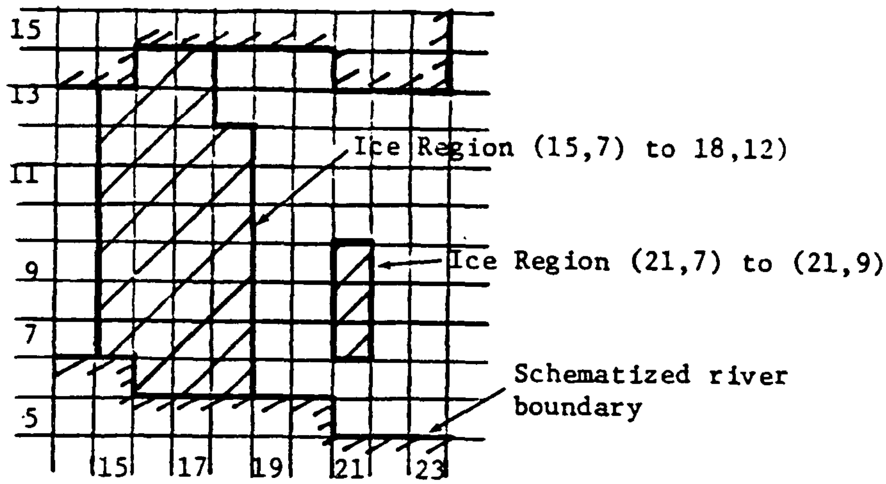


Figure 16. Defining Ice Regions

LDETR.ICE; Block 1 (Ice regions in lake and river)

Card 1

Example:

0.035 0.84

Variable Name	Type and Length	Column Number	Definition
ANICE	Real	--	Manning's n for ice roughness
AMIUO	Real	--	viscosity of oil (lb sec/ft^2)

Card 2

Example:

1 1

Variable Name	Type and Length	Column Number	Definition
NICERG	Ingeter	--	total number of ice regions
LICERG	Integer	--	number of ice regions in lake

Card 3 (1 card for all NICERG ice regions. If line is not long enough continue on next line)

Example:

15 7 18 9

Variable Name	Type and Length	Column Number	Defintion
NICEX1(I)	Integer	--	x grid at the beginning of ice region
NICEY1(I)	Integer	--	y grid at the beginning of ice region
NICEX2(I)	Integer	--	x grid at the end of ice region
NICEY2(I)	Integer	--	y grid at the end of ice region

Card 4 (1 card for all LICERG ice regions)

Example:

1.0

Variable Name	Type and Length	Column Number	Definition
ZLKICE(I)	Real	--	ice thickness in lake ice region (ft). Thickness must be defined for each lake ice region (use one line, and then continue to next)

NOTE: Card 4 will only appear after Card 3 for lake ice regions. Ice thickness in river is accounted for through input in file LDETR.FLW.

NOTE: Cards 2, 3 and 4 must be repeated for each unsteady flow model time step.

LDETR.FLW

The LDETR.FLW file contains the water level and discharge boundary conditions for each node in the river as defined by the one dimensional flow model (Thomas, 1984). Also included are the ice conditions for each cross section in the river. This data is separate from the ice region data in LDETR.ICE. The oil spill simulation model converts this information into boundary conditions for each river branch. The lake model, RLID, uses the water level and discharge at the beginning of branch No. 1 (lake-river interface) to adjust the bathymetric and streamfunction files to reflect the current flow conditions.

This file consists of three blocks of information. All blocks are listed below with descriptions and corresponding components. Blocks 2 and 3 must be repeated every time the velocities are updated in the model, i.e. every time step of the one dimensional flow model. Therefore, the data in this file needs to be adjusted on a more regular basis.

LDETR.FLW; Block 1 (Time step for updating flow conditions)

Card 1

Example:

3.0

Variable Name	Type and Length	Column Number	Definition
UFDT	Integer	--	time step in one dimensional flow model (hrs)

LDETR.FLW; Block 2 (Discharge and water level)

Card 1 (1 card for each node in the one dimensional model)

Example:

```
    573.72     149190.
```

Variable Name	Type and Length	Column Number	Definition
WL(I)	F6.2	--	water level at I^{th} node (ft)
Q(I)	F10.0	--	discharge at I^{th} node (cfs)

LDETR.FLW; Block 3 (Ice thickness information)

Card 1

Example:

```
    1
```

Variable Name	Type and Length	Column Number	Definition
ICINFO	Integer	--	number of cross sections with ice covered conditions. If there are no ice covered sections, set ICINFO=1 and then in Card 2, define an arbitrary section number to be OPEN.

Card 2 (1 card for each cross section with ice conditions)

Example:

```
    2     OPEN
```

Variable Name	Type and Length	Column Number	Definition
IS	I4	1-4	cross section number with ice cover condition
WORD	A4	6-9	cross section ice cover condition, "FULL"=fully covered, "PART"=partially covered, "OPEN"=open water

NOTE: Each Card 2 is followd by a Card 3 and the Card 2, Card 3 combination is repeated ICINFO(IS) times if WORD does not equal "OPEN". If the entire river is open water, the data for this time step ends here.

Card 3 (For fully covered cross section only, i.e. if WORD="FULL")

Example:

1.0

Variable Name	Type and Length	Column Number	Definition
FULLTI	Real	--	ice thickness of fully covered cross section (ft). Only one value read and assigned to entire cross section

Card 3 (For partially covered cross section only, i.e. if WORD="PART")

Example: (If NSLSCT(IS) = 9)

.0 1.0 0.9 0.8 0.5 0.0 0.0 0.7 0.8 1.1

Variable Name	Type and Length	Column Number	Defintion
TICE(I,J)	Real	--	ice thickness of partially covered cross section (ft). There must be one value for each NSLSCT(IS) sounding depth location plus one additional. Measurements start on the reference shoreline and proceed from one sounding point to the next until all points have some ice thickness value

NOTE: The one additional thickness is required since the sounding depth on the reference shoreline is taken as zero and is not input through data, yet an ice thickness may still be required at that point.

DETR.BND

The LDETR.BND file contains one block of data to identify the oil retention/rejection characteristics of shorelines. The user can set any shore grid box with one of the ten predetermined half life values defined internally to the computer model. The possible values are given later in section III.4.

Once the half life values are assigned, this file shouldn't require any further changes. This, however, is left up to the user's discretion.

LDETR.BND; Block 2 (Half life data for lake and river)

Card 1 (1 card for each range of grid boxes)

Example:

```
     1     1     285     10
```

Variable Name	Type and Length	Column Number	Definition
K	Integer	--	shore number; see Figure 4
LFROM	Integer	--	beginning limit (grid box number) for half life designation to shore
LTO	Integer	--	ending limit (grid box number) for half life designation to shore
ICODE	Integer	--	integer identifying which of the ten half life values to be assigned to a grid

NOTE: The last card must be a set of four zeros 0 0 0 0 which are used to identify the end of the data block.

DETR.SPL

The LDETR.SPL file contains two blocks of information controlling the oil characteristics and the general spill simulation. From the viewpoint of modeling actual spills, most of the data in this file will change for each spill. If only oil spill scenarios are to be conducted, most of the parameters describing a particular type of oil may remain untouched, although such information as the initial spill location would have to be changed. For these reasons, guidelines are given later (section III.4) as far as choosing appropriate numbers for the variables described here.

DETR.SPL; Block 1 (Simulation parameters and coefficients)

Card 1 (Type of oil - Identification only)

Example:

Fuel oil No. 2

Variable Name	Type and Length	Column Number	Definition
FUELTP	Character	1-16	text for identifying the oil type

Card 2 (Simulation time steps and printed output control parameters)

Example:

6.0 1 0 1 0 0 450. -1.0 -1.0

Variable Name	Type and Length	Column Number	Definition
TOTIME	Real	--	total time of oil spill simulation (hrs). This value must equal or exceed the time step in unsteady flow model, i.e. in FLW file.
IEVERY	Integer	--	frequency of obtaining output from **PLOTNU** and other subroutines i.e. value of two (2) gives output every other time step
IOPT1	Integer	--	two possible values: one (1) results in output of fixed data such as cross section geometry and shore conditions, zero (0) cancels this output
IOPT2	Integer	--	two possible values: one (1) results in output of computed velocities to be used for plotting, zero (0) cancels this output
IOPT3	Integer	--	two possible values: one (1) results in output of particle locations to be used in plotting, zero (0) cancels this output
IOPT4	Integer	--	two possible values: one (1) results in number plot of particle distribution (see **PLOTNU**), zero (0) cancels this output
SPLTIM	Real	--	duration of oil spill (sec). For a spill released over 7.5 minutes SPLTIM=450. Value of zero is allowed.

DIFFUL	Real	--	horizontal diffusion coefficient (ft^2/s) for lake. If the default formulation as described in Vol. I is desired, set this value to -1.0
DIFFUR	Real	--	horizontal diffusion coefficient (ft^2/s) for river. If the default formulation as described in Vol. I is desired set this value to -1.0

Card 3 (Spill description and spreading equation coefficients)

Example:

500 10000. 900. 0.84 1.411E-5 2.06E-3 1.14 0.98 1.6 1.39 1.39 1.43

Variable Name	Type and Length	Column Number	Definition
NTOTAL	Integer	--	total number of particles defined in the system (current maximum is 1000)
SPVOL	Real	--	total volume of oil spill (U.S. gal)
SPILDT	Real	--	magnitude of time step for spill simulation (sec)
SPGOIL	Real	--	specific gravity of oil
ANIU	Real	--	kinematic viscosity of water (sq ft/sec)
SIGMA	Real	--	surface tension of oil (lbs/ft)
AK2I	Real	--	gravity-inertia phase spreading coefficient (axisymmetrical)
AK2V	Real	--	gravity-viscous phase spreading coefficient (axisymmetrical)
AK2T	Real	--	surface tension-viscous phase spreading coefficient (axisymmetrical)
AKC10	Real	--	gravity-inertia spreading phase coefficient (one-dimensional)
AKC20	Real	--	gravity-viscous phase spreading coefficient (one-dimensional)
AKC30	Real	--	surface tension-viscous phase spreading coefficient (one-dimensional)

Card 4 (Spill location and additional oil properties)

Example:

-5000. 35000. .7063E-02 .1873E-02 7.88 465.0

Variable Name	Type and Length	Column Number	Definition
SPX	Real	--	x-coordinate of initial spill site (ft), negative if in lake
SPY	Real	--	y-coordinate of initial spill site (ft)
VMUNI	Real	--	molar volume of oil (cu ft/mol)
SOLUNI	Real	--	solubility of fresh oil (lbs/cu ft)
CEVP	Real	--	coefficient C of evaporation characteristics of oil
TOEVP	Real	--	boiling point temperature of oil ^{o}K

NOTE: If you define a value of less than 1.0 for boiling point temperature the program defines the evaporation characteristics, using fitted curves. Therefore the input values of CEVP and TOEVP have no influence on computations although they are read.

DETR.SPL; Block 2 (Components of wind speed and environmental temperature)

Card 1 (1 card for each time step in simulation)

Example:

10.0 270.0 50.0

Variable Name	Type and Length	Column Number	Definition
VWMAG	Real	--	wind speed (ft/s)
THETA	Real	--	wind direction, clockwise angle from north degrees (ex. wind out of west=270^{o})
TENVF	Real	--	air temperature (F^{o})

III.2 Lake Data

LAKEWIND.DAT

The file LAKEWIND.DAT contains the meteorological data for the lake circulation model. Description of this data can be found in listings of subroutines RLID, PARTIC and TAU (Appendix II), and in the report by Schwab, et al. (1981). The lake model uses this data to compute the surface wind stress in both time and space. If only one wind observation is available, the time of observation must be given as zero (0). The most important detail to be aware of when assembling this data file is that the time interval between subsequent wind data must be the same as the time step (UFDT) for the one dimensional river model.

This file consists of only one block of data. Since wind stations and the elevation at which data is recorded isn't likely to change, the user need merely adjust the wind magnitude and direction for each execution of the program. Of course, if the interval (UFDT) changes, so must the times for this data.

LAKEWIND.DAT; Block 1 (Lake meteorological data)

Card 1 (1 card for each wind station, maximum 25 wind stations per time interval (UFDT))

Example:

```
0.    42.42    82.42    30.    64.    54.    15.0    180.
```

/ Variable Name	/ Type and Length	/ Column Number	/ Definition /
TLAST	G10.4	1-10	time at which wind observation is made (hrs from initial spill)
RLAT	G10.4	11-20	latitude of wind observation point (degrees north)
RLON	G10.4	21-30	longitude of wind observation point (degrees west)
Z	G10.4	31-40	height of instruments (ft)
TA	G10.4	41-50	temperature of air (^{o}F)
TW	G10.4	51-60	temperature of water (^{o}F)
WS	G10.4	61-70	wind speed (ft/sec)
WD	G10.4	71-76	wind direction (degrees clockwise form north)

NOTE: The last card must have a vlaue for TLAST equal to -1.0. This denotes that no more wind information is available. All data for the same time are grouped together.

LAKEBATH.DAT

The file LAKEBATH.DAT consists of three blocks of data which contains the bathymetric data and various grid control parameters for the lake circulation model. Description of this data can be found in listings of subroutines RLID, PARTIC and RGRID (Appendix II), and in the report by Schwab and Sellars (1980). The lake model uses the depth when solving the vertically integrated shallow water equations written in terms of the stream function (Volume I). The user need never change any data in this input file. In the event that changes are desired, the user should consult Chapter IV or the report by Schwab and Sellars (1980).

LAKEBATH.DAT; Block 1 (Title and grid control parameters)

Card 1

Example:

LAKE ST. CLAIR BATHYMETRY

Variable Name	Type and Length	Column Number	Definition
IPARM(5-54)	A1	1-50	title of lake

Card 2

Example:

36, 38, 42.3041534, 82.9315796, 4000, 19, 1, -2.24, 39.677, 10.087, -120.06

Variable Name	Type and Length	Column Number	Definition
IPARM(1)	I5	1-5	number of grids in x direction
IPARM(2)	I5	6-10	number of grids in y direction
RPARM(1)	F12.7	11-22	base latitude
RPARM(2)	F12.7	23-34	base longitude
RPARM(3)	F5.0	35-39	grid dize (ft)
RPARM(4)	F5.0	40-44	maximum depth (ft)
RPARM(5)	F5.0	45-49	minimum depth (ft)
RPARM(6)	F6.2	50-55	base rotation (counterclockwise is negative)
ZPARM(1)	F7.3	56-62	I-Displacement, the number of new grid squares in the x direction from the new grid origin to the old grid origin
ZPARM(2)	F7.3	63-69	J-Displacement, the number of new grid squares in the y direction from the new grid origin to the old grid origin
RPARM(7)	F7.2	70-76	rotation from base (counterclockwise is negative)

Card 3

Example:

0.822690E+02 -0.418687E+01 -0.892958E+00 0.549244E+00

Variable / Name	Type and / Length	Column / Number	Definition /
RPARM()	E15.6	1-60	geographic to map or map to geographic coordinate conversion coefficients

NOTE: Card 3 is repeated until all 16 coefficients RPARM(8) through RPARM(23) are read.

LAKEBATH.DAT; Block 2 (Grid depths)

Card 1 (Each card contains 19 grid depths)

Example:

9 9 6 4 0 0 0 3 10 10 10 12 11 11 9 7 5 0 0

/ Variable / Name	Type and / Length	Column / Number	Definition /
D(I,J)	F4.0	I-76	grid depths

NOTE: Card 1 repeated until all grid depths are read.

LAKEBATH.DAT; Block 3 (Time step of lake circulation model)

Card 1

Example:

1.

/ Variable / Name	Type and / Length	Column / Number	Definition /
DT	G8.2	1-8	time step for lake circulation model

LAKEINIT.PSI

The file LAKEINIT.PSI contains only one block of information which consists of initial stream function values for every grid in the lake. This file requires no adjustment once it is set for a specified discharge. Details of setting up this input file and how the stream functions are used to establish boundary conditions are covered in Chapter IV. If no initial stream function file is available the default will set all stream functions to zero.

LAKEINIT.PSI; Block 1 (Initial stream function values)

Card 1

Example:

0.10000E+21 0.10066E+06 0.97943E+05 0.96472E+05

/ Variable Number /	Type and Length /	Column Number /	Definition /
S(I,J)	E12.5	1-72	initial streamfunction for lake grids

NOTE: The example only shows typical values. Each card in the actual file contains six (6) streamfunction values.

III.3 Input Adjustments

For a lake-river system (i.e. Lake St. Clair-Detroit River area) which already has the necessary input files, very little has to be modified to run the model for a variety of spill scenarios. The cards most likely to require modification are cited below. This is followed up with some guidelines and suggested values for input. No at ·mpt is made to explain the formatting of the data changes here. The user is expected to refer to sections III.1 and III.2 for specific formattting procedures. Cards most likely to require up-to-date information include:

File Name	Block Number	Card Number	Variables
LDETR.ICE	1	1	ANICE, AMIUO
LDETR.ICE	1	2	NICERG, LICERG
LDETR.ICE	1	3	NICEX1(), NICEY1(), NICEX2(), NICEY2()
LDETR.ICE	1	4	ZLKICE
LDETR.FLW	1	1	UFDT
LDETR.FLW	2	1	WL(), Q()
LDETR.FLW	3	1	ICINFO
LDETR.FLW	3	2	IS, WORD
LDETR.FLW	3	3	FULLTI or TICE(,)
LDETR.BND	1	1	ICODE
LDETR.SPL	1	2	TOTIME, SPLTIM
LDETR.SPL	1	3	NPTCL, SPVOL, SPILDT, SPGOIL, ANIU, SIGMA, DIFFUL, DIFFUR
DETR.SPL	2	1	VWMAG, THETA, TENVF
LAKEWIND.DAT	1	1	TLAST, TA, TW, WS, WD

Some of the variables listed here may not change at all and other additional parameters not listed may need some revision.

DETR.ICE

The Manning's n of the undersurface of the ice cover and the viscosity of oil must be specified in this file as ANICE and AMIUO, respectively.

In general, Manning's n can range from 0.020 to 0.065 for the underside of an ice cover. Oil viscosity is a property specific to the type of oil which has been spilled.

The user must locate ice regions in the water body and convert that information to the appropriate grids. Suggestions for handling the acquisition of this data is as follows:

1.) Set up typical files for the stages of ice cover progression in both the lake and river. In this way, seasonal data files corresponding to the state of ice conditions, can be selected without spending too much time assembling data.

2.) Locate ice regions on the grid maps in Chapter II, which have sufficient detail on the shoreline geometry and landmarks.

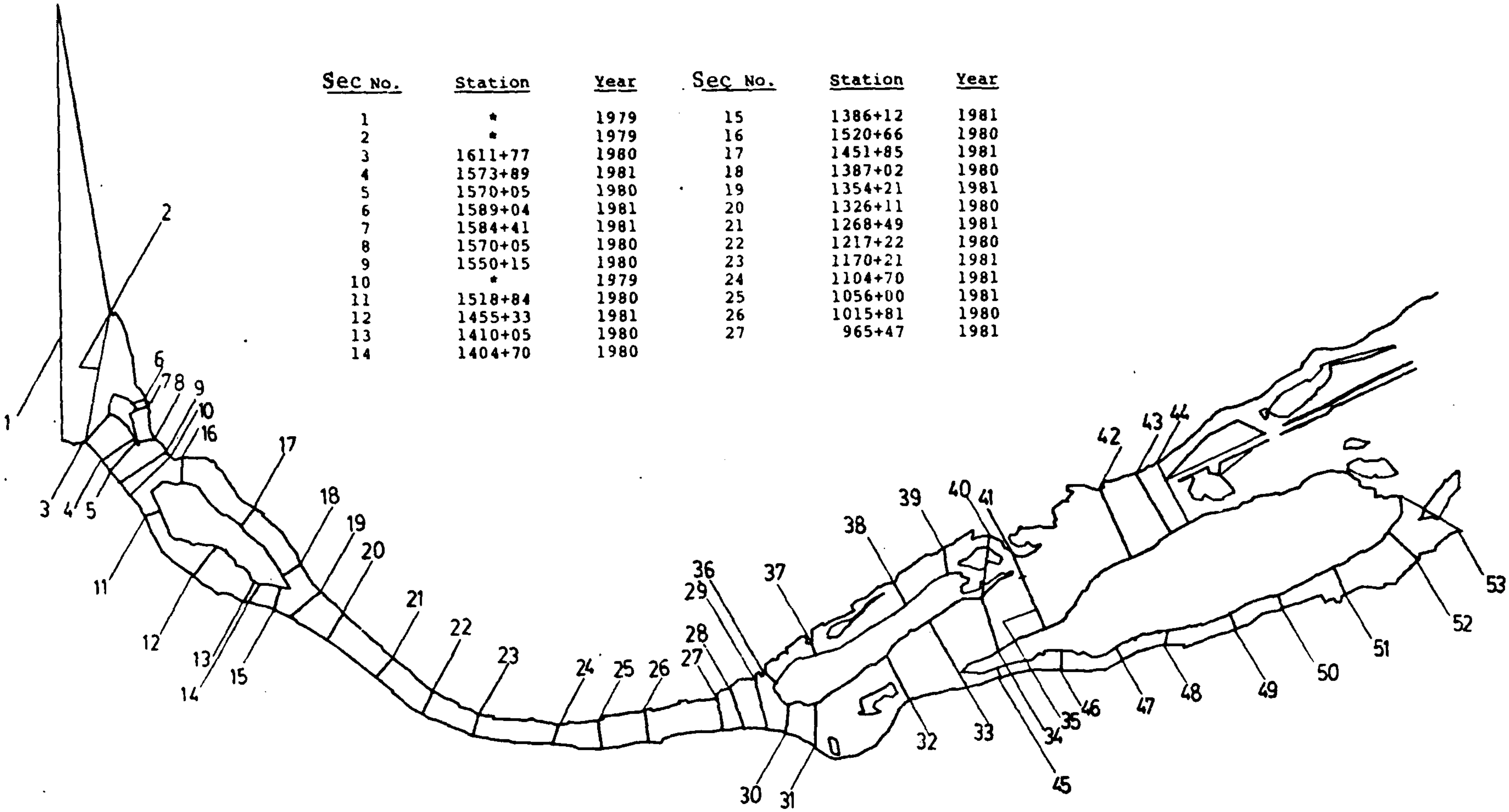
Specifying the ice region area and the number of ice regions was described earlier in Section III.2. Caution is given to keep ice regions in the lake separate from those in the river. An ice region that extends from lake to the river must be considered as two regions. When entering the data, like ice regions must be specified first followed by river ice regions. Also, note that LICERG only refers to the number of lake ice regions whereas NICERG is the total number of all ice regions. The model is only set up to handle a maximum of 20 ice regions. If more are desired, the size of arrays (ZLKICE(), NICEX1(), NICEY1(), NICEX2(), and NICEY2() must be increased.

The reason for specifying lake ice regions first is tied to the ice thickness data. Whenever an ice region is specified for the lake, the next information read is the corresponding lake ice thickness. The thickness, ZLKICE, is considered to be uniform for the entire region and is used to adjust the lake depths for the presence of ice.

DETR.FLW

Stage and discharges, Q() and WL() at nodes in the river are unlikely to remain constant with respect to time. Therefore this information must be entered to reflect the correct flow conditions. The one-dimensional river unsteady flow model is used to obtain this information. To set up the file, LDETR.FLW, simply enter the unsteady flow model time step, UFDT, and then the discharge, Q(), and water level, WL(), which appear in the one-dimensional models output.

To account for ice conditions when computing the river velocities, additional data locating ice in each cross section is required in file LDETR.FLW. Specifying this ice information for partially ice covered river cross sections needs some clarification. Remember that each cross section is described by sounding depths and distances from a reference shore at which those sounding depths were taken. The data is assembled in Block 3, Cards 1 & 2 of file LDETR.GEO with the reference shoreline for the Detroit River as the lower shoreline in the x-y grid system of Chapter II. The number of sounding depths and the locations from the reference shore are not the same for each cross section. This information is presented in tabular form in Appendix I. The above mentioned table is prepared based on the data in Block 3 of LDETR.GEO. In the case of a partially ice ocvered section an ice thickness needs to be defined at each sounding depth location. Figure 17 shows the Detroit River with the designated reference shoreline and cross section locations. Cross section numbers are the same as those found in LDETR.GEO.



Sec No.	Station	Year	Sec No.	Station	Year
1	*	1979	15	1386+12	1981
2	*	1979	16	1520+66	1980
3	1611+77	1980	17	1451+85	1981
4	1573+89	1981	18	1387+02	1980
5	1570+05	1980	19	1354+21	1981
6	1589+04	1981	20	1326+11	1980
7	1584+41	1981	21	1268+49	1981
8	1570+05	1980	22	1217+22	1980
9	1550+15	1980	23	1170+21	1981
10	*	1979	24	1104+70	1981
11	1518+84	1980	25	1056+00	1981
12	1455+33	1981	26	1015+81	1980
13	1410+05	1980	27	965+47	1981
14	1404+70	1980			

Sec No.	Station	Year	Sec No.	Station	Year
28	930+25	1980	41	*	1979
29	*	1979	42	548+10	1980
30	893+23	1980	43	504+17	1981
31	868+63	1981	44	*	1979
32	785+31	1981	45	693+07	1980
33	732+34	1981	46	635+94	1981
	728+51	1981			
34	700+00	1980	47	583+47	1980
35	650+68	1981	48	537+94	1981
36	893+25	1980	49	478+93	1980
37	853+93	1981	50	427+63	1981
38	767+74	1981	51	374+08	1980
39	715+05	1980	52	302+46	1980
40	*	1979	53	*	1979

* Cross sections measured from National Oceanic and Atmospheric Administration, National Ocean Survey, Chart 14853 of Detroit River, 8th Ed., April 14, 1979.

Figure 17 Cross section locations in Detroit River

Therefore, by using Appendix I, Figure 17, and the extent of ice cover at the cross sections, the user should be able to correctly input the ice thickness data for the array TICE(,).

LDETR.BND

The half-life code, ICODE, must be assigned to the shoreline grids along the lake and river to enable the computation of rejection/retention of oil. Ten possible half-life codes are currently available. They should be broad enough to cover any situation and are given according to the type of shoreline as:

ICODE	Half-Life (hrs)	Shore Characteristics
1	0.033	sheet piling
2	0.5	commercial docks
3	1	private docks
4	6	-------------
5	12	embankments
6	18	-------------
7	24	sand/gravel beach
8	48	marsh
9	48	shallow water
10	8760	bays, sheltered areas

No information is available to more accurately correlate the half-life values to the shoreline characteristics other than what is given here. Torgrimson's (1984) suggested half-life values were used to arrive at those given above. The logic used to obtain these values is that the smaller the half-life, the less likely the oil will remain on the shore.

LDETR.SPL

The oil spill parameters in LDETR.SPL will require the most modification of any file from spill to spill. Any particular spill will have its own total simulation time, TOTIME, simulation time step, SPILDT, duration of spill, SPLTIM, spill volume, SPVOL, number of particles, NPTCL (maximum 1000), spill location, SPX and SPY, wind speed and direction, VWMAG and THETA, air temperature, TENVF, and oil and water properties, so, it will be necessary to adjust these data each time a new spill is simulated. Most of these are self-explanatory and have already been described in Section III.1. Those which require particular attention are the spill location, wind components and oil and water properties.

The spill location can be determined using Figures 8 through 14. Knowing the grid sizes of both the lake (currently 4000 ft) and river (currently 500 ft) and the location of the x and y axes, a particular location can be pinpointed anywhere. Note that the lake domain has negative x coordinates and the river domain has positive x coordinates.

The wind speed and direction which is used to compute the wind component of the drift velocity must be input for every SPILDT time interval. The unit of wind speed is ft/s. The wind direction is the clockwise angle measured from magnetic north. As an example, wind blowing out of the west has a direction of 270°.

For the type of oil being used in the simulation, the specific gravity (SPGOIL), molar volume (VMUNI) and solubility in water (SOLUNI), must be specified. Also, the oil water interfacial tension (SIGMA), and the viscosity of water (ANIU), are required input.

For determining evaporation rates of oil, coefficient C (CEVP) and boiling point temperature (TOEVP) are needed. The user has an option here. If TOEVP is given a value less than 1.0, the model automatically computes CEVP and TOEVP, using the curves described in an earlier section. The curves used are for crude oils with API value ranging from 10 to 45. The data for CEVP and TOEVP must be present in the datafile although they serve no purpose in this case. The other option is to enter the correct CEVP and TOEVP values. In this case values of CEVP and TOEVP will be used for computing evaporation rates. It should be noted that an oil having an API value between 10 and 45 is not necessarily a crude oil. For non-crude oils, CEVP and TOEVP should be given as input.

LAKEWIND.DAT

The wind data required by the lake model may come from the same wind stations where the wind speed and direction were given as input to the oil spill model. However, in order to protect the integrity of both the lake and river models, the information must be read from two separate files. In the file LAKEWIND.DAT, forecasted wind speed, WS, wind direction (degrees clockwise from north), WD, and air and water temperatures can be directly placed in the file without any conversion or rotation. The time interval for data specification is independent of any other time step in the model. the user may simply input one line of wind data at T = 0 hours and the model will use this wind data for the entire simulation. The lake model will always use the last data read in if the simulation continues longer than the last time specified in LAKEWIND.DAT. The sample file given at the end of this chapter has ficticious weather station locations.

III.4 Sample Data Files (for Lake St. Clair-Detroit River Study Area)

Input

Datafile Name	Unit No.
LDETR.GEO	1
LDETR.ICE	5
LDETR.FLW	7
LDETR.BND	8
LDETR.SPL	12
LAKEWIND.DAT	10
LAKEBATH.DAT	13

Output

Datafile Name	Unit No.
OILPRT.OUT	2
VELSTR.OUT	3
VELCAR.OUT	4
LDETRSP.OUT	11
LAKETEMP.PSI	15

LDETR.GEO (UNIT 1)

```
DETR  LAKE ST. CLAIR AND DETROIT RIVER
16  35  285  4000.  500.  7  -1.4E+05
2  5  8  10  15  18  22  27  29  31  33  35  41  44  50  52
  1  (250.,30500)           1.57079630  11  11   2    0
  2  (1895.4,30474.9)       1.35387330  11  12   3    6
  3  (1895.4,30474.9)       0.80316770   9  11   4    0
  4  (3572.5,28827.7)       0.57252250   9  11   5    0
  5  (4329.2,27835.3)       0.85065230   9  21   9    0
  6  (6348.3,33851.1)       0.32890491   2  11   7  888
  7  (6573.7,33287.7)       0.27566774   2  11   8    0
  8  (6880.6,30383.6)       0.19895402   2  21   9  999
  9  (5218.9,26793.0)       0.60033042  11  11  10    0
 10  (6206.0,24830.4)       0.78546520  11  12  11   16
 11  (7367.3,23785.9)       0.38429453   4  11  12    0
 12  (11648.1,18516.6)      0.89099240   4  11  13    0
 13  (16161.9,16194.4)      1.06953870   4  11  14    0
 14  (16630.9,16089.5)      1.03244550   4  11  15    0
 15  (18885.3,15317.3)      1.29700210   4  21  19    0
 16  (10645.8,26730.4)      1.50359930   7  11  17  888
 17  (15909.5,22917.5)      0.89085370   7  11  18    0
 18  (19678.2,18445.3)      0.56874187   7  21  19  999
 19  (20184.2,14800.2)      0.65250602  11  11  20    0
 20  (23601.6,12740.2)      1.01761670  11  11  21    0
 21  (27986.0,9434.1)       0.84731720  11  11  22    0
 22  (32096.4,6111.7)       1.11644090  11  11  23    0
 23  (36555.3,4153.7)       1.31789400  11  11  24    0
 24  (43737.9,3322.7)       1.29165360  11  11  25    0
 25  (48111.9,2967.3)       1.73312770  11  11  26    0
 26  (52258.0,3574.1)       1.70696930  11  11  27    0
 27  (58891.5,4663.2)       1.74159230  11  11  28    0
 28  (60821.2,4700.4)       1.89850420  11  11  29    0
 29  (63193.4,4508.6)       1.85897050  11  12  30   36
 30  (64446.6,4279.9)       1.40498040   9  11  31    0
 31  (67199.6,2832.9)       1.55881210   9  11  32    0
 32  (75744.0,7230.3)       2.07408620   9  11  33    0
 33  (81028.9,8357.1)       2.01051160   9  12  45   34
 34  (83712.4,11584.0)      1.85562200   7  11  35  868
 35  (87888.6,13541.0)      1.97159570   7  21  42    0
 36  (63732.4,8702.1)       2.37748450   2  11  37  888
 37  (67304.8,11205.3)      1.95566550   2  11  38    0
 38  (75490.2,15658.0)      2.08554860   2  11  39    0
 39  (79227.9,18354.3)      1.69594960   2  21  40    0
 40  (82248.9,17195.8)      1.63638540   2  11  41    0
 41  (85910.8,18208.0)      1.96796920   2  21  42  999
 42  (95603.0,19752.6)      1.98250300   9  11  43    0
 43  (99230.8,22002.3)      2.09447510   9  11  44    0
 44  (101000.3,23059.3)     2.10296690   3  11  43    0
 45  (84015.6,9218.3)       1.97403760   2  11  46    0
 46  (89415.8,9892.4)       1.55666850   2  11  47    0
 47  (94836.8,10752.4)      2.10658900   2  11  48    0
 48  (98678.7,12200.2)      1.41654700   2  11  49    0
 49  (105015.3,13480.9)     1.97177820   2  11  50    0
 50  (109190.4,15253.8)     1.89456780   2  11  51    0
 51  (114885.3,16482.1)     2.00199130   2  11  52    0
 52  (121352.6,19517.5)     2.37169800   2  11  53    0
 53  (125494.9,22087.9)     2.67019810   2  11  52    0
      1      17    571.71
  1375.00     6.00  1675.00  24.00   3000.00  21.00   3575.00  16.00   4000.00  23.0
  4250.00    18.00  5000.00  10.00   8500.00   9.00   9625.00  10.00  10875.00   8.0
 13000.00     8.00 13250.00   6.00  15500.00   4.00  21000.00   6.00  26000.00   4.0
 29000.00     3.00 37000.00   0.00
```

2	16	571.71							
125.00	12.00	200.00	21.00	1500.00	27.00	2000.00	28.00	2500.00	12.0
2550.00	9.00	3125.00	6.00	4375.00	2.00	6000.00	6.00	6450.00	12.0
6625.00	19.00	7000.00	25.00	7750.00	7.00	8325.00	6.00	10500.00	2.0
11783.00	0.000								
3	10	574.87							
25.00	14.00	125.00	28.00	500.00	36.00	1000.00	37.00	1200.00	33.0
1750.00	32.00	2000.00	30.00	2250.00	10.00	3000.00	8.50	3425.00	0.0
4	10	574.94							
155.00	25.00	900.00	27.50	1025.00	36.00	1275.00	40.00	1400.00	35.5
1700.00	43.00	2125.00	30.00	2375.00	10.00	3250.00	5.00	3375.00	0.0
5	9	574.91							
50.00	12.00	125.00	20.00	750.00	21.00	1250.00	22.00	1350.00	28.0
2425.00	28.00	2525.00	5.00	3250.00	4.00	3425.00	0.00		
6	5	574.45							
75.00	12.00	175.00	22.00	250.00	27.00	825.00	27.00	1400.00	0.0
7	5	574.50							
75.00	12.00	175.00	22.00	250.00	27.00	825.00	27.00	1250.00	0.0
8	10	574.91							
80.80	11.00	141.60	26.30	197.30	28.80	373.00	29.50	627.40	30.8
759.60	30.00	1123.80	24.10	1183.60	21.10	1233.10	9.50	1322.00	0.0
9	15	574.92							
125.00	12.00	250.00	25.00	500.00	21.00	800.00	19.50	1000.00	27.5
1250.00	23.00	2050.00	22.50	2850.00	38.00	3500.00	38.00	4000.00	29.0
4250.00	43.00	4650.00	45.00	4825.00	6.50	5000.00	6.50	5100.00	0.0
10	11	574.87							
75.00	4.00	275.00	18.00	575.00	24.00	1250.00	18.00	2075.00	12.0
2825.00	27.00	3825.00	29.00	4450.00	39.00	4600.00	18.00	4675.00	2.0
5075.00	0.00								
11	7	574.82							
101.10	25.00	266.10	24.80	506.50	29.10	756.20	33.00	1222.90	31.0
1360.00	25.80	1461.00	0.00						
12	11	574.54							
25.00	12.00	75.00	25.00	475.00	23.00	575.00	12.00	800.00	6.0
1250.00	2.00	1875.00	5.00	2050.00	22.00	2550.00	27.00	2750.00	5.0
3400.00	0.00								
13	7	574.27							
87.60	22.20	159.00	26.90	1106.90	20.90	1092.90	26.40	1997.50	23.4
2073.60	11.70	2110.70	0.00						
14	10	574.57							
1.90	9.50	135.00	25.40	550.10	30.50	695.30	24.30	882.50	29.0
1000.00	20.00	1373.70	22.30	1691.80	26.30	1843.60	21.10	1923.20	0.0
15	11	574.16							
24.20	19.70	607.00	21.90	1004.50	28.50	1207.70	10.00	1382.10	32.7
1598.30	27.00	1684.20	19.90	1862.20	12.90	1891.40	7.30	2194.30	5.9
2203.80	0.00								
16	12	574.85							
125.00	25.50	250.00	35.00	350.00	33.00	680.00	47.50	825.00	42.0
1250.00	46.00	1525.00	35.00	1625.00	35.00	1825.00	9.00	2125.00	7.5
2533.00	9.00	2550.00	0.00						
17	8	574.18							
108.80	31.40	317.30	39.10	529.20	35.30	754.70	39.60	1003.90	36.6
1660.30	42.20	1831.30	24.10	1920.80	0.00				
18	12	574.47							
30.00	4.50	63.00	13.50	250.00	14.50	475.00	41.50	563.00	32.0
1125.00	32.50	1375.00	40.50	1625.00	42.50	1750.00	34.50	1825.00	38.0
1950.00	23.00	2000.00	0.00						
19	15	574.53							
125.00	12.00	375.00	18.00	425.00	25.00	1000.00	35.00	1250.00	37.0
1375.00	37.00	1500.00	9.00	1725.00	9.00	1850.00	25.00	2000.00	37.0
2375.00	29.00	2750.00	28.00	3125.00	32.00	3500.00	35.00	3750.00	0.0

20	9	574.59							
10.00	27.00	550.00	28.00	625.00	37.00	1375.00	35.50	1750.00	36.0
2125.00	48.00	2310.00	43.00	2625.00	38.00	2910.00	0.00		
21	8	574.02							
117.60	26.20	368.50	21.70	634.20	31.80	811.10	50.80	1337.70	38.6
1657.20	48.00	2223.70	40.50	2372.80	0.00				
22	13	574.40							
25.00	23.00	125.00	27.50	200.00	23.80	475.00	34.00	600.00	29.5
750.00	44.50	1000.00	49.80	1250.00	43.00	1600.00	47.40	1750.00	36.0
2000.00	45.00	2125.00	42.00	2250.00	0.00				
23	9	574.50							
125.00	18.00	175.00	29.00	375.00	41.00	1375.00	45.00	1500.00	47.0
1625.00	32.00	1800.00	18.00	1875.00	6.00	2050.00	0.00		
24	7	573.60							
197.00	37.80	451.10	46.20	717.30	46.90	1186.50	39.30	1593.60	42.8
1673.60	39.30	1903.10	0.00						
25	7	573.90							
25.00	28.00	125.00	38.00	1000.00	38.00	1500.00	31.00	2250.00	32.0
2500.00	27.00	2550.00	0.00						
26	8	574.19							
25.00	8.50	510.00	45.00	1120.00	37.50	2075.00	34.75	2375.00	37.5
2600.00	33.75	2710.00	8.00	2825.00	0.00				
27	9	574.14							
25.00	19.00	125.00	25.00	300.00	40.00	1000.00	39.00	1500.00	32.0
1750.00	36.00	2375.00	36.00	2625.00	24.00	2650.00	0.00		
28	11	574.12							
250.00	38.30	680.00	43.20	1125.00	36.00	1485.00	35.00	1750.00	40.8
2090.00	37.20	2500.00	37.50	2850.00	10.00	3320.00	5.20	3750.00	8.0
4250.00	0.00								
29	9	573.97							
175.00	34.00	1050.00	38.00	2175.00	28.00	2500.00	18.00	2675.00	25.0
3300.00	27.00	3425.00	4.00	4425.00	3.00	4675.00	0.00		
30	13	573.82							
185.00	28.50	375.00	40.00	500.00	41.00	625.00	40.20	850.00	35.5
1150.00	37.00	1500.00	35.80	1650.00	38.20	2000.00	29.50	2125.00	8.5
2310.00	4.80	2450.00	6.80	2510.00	0.00				
31	11	573.40							
27.10	3.20	230.90	6.00	450.10	20.00	835.60	30.00	1548.30	34.0
1842.40	45.50	2330.90	31.00	2586.60	34.40	2864.60	8.10	3354.00	6.5
3400.00	0.00								
32	14	573.39							
117.30	35.20	222.90	37.20	803.50	30.00	996.90	10.90	1503.20	6.0
1829.30	10.00	1975.90	31.00	2550.50	39.20	3196.90	35.00	3392.00	16.8
4179.00	4.80	4207.60	9.00	4276.50	8.20	4776.60	0.00		
33	19	573.51							
32.80	17.80	115.10	34.60	327.40	34.30	593.00	36.00	864.00	9.4
1148.70	6.40	1198.80	9.50	1254.40	0.00	1300.00	0.00	1337.40	7.9
2000.90	32.30	2389.90	7.20	2443.4	6.20	2983.60	13.70	3234.50	38.0
4076.00	38.00	4409.10	9.30	6203.8	6.00	6715.40	6.40	6760.70	0.0
34	12	573.75							
300.00	31.00	500.00	31.00	780.00	4.50	1310.00	4.60	1790.00	35.0
2090.00	37.00	2900.00	35.00	3250.00	36.00	3680.00	7.20	5600.00	6.8
5650.00	9.00	5750.00	0.00						
35	10	572.40							
450.00	6.00	575.00	28.00	1250.00	36.00	1750.00	33.00	2700.00	28.0
2725.00	18.00	2825.00	5.00	4500.00	3.00	5000.00	2.00	5025.00	0.0
36	8	573.73							
120.00	9.10	174.00	19.60	461.30	34.90	851.00	32.00	986.20	21.9
1040.00	13.40	1106.00	5.30	1175.30	0.00				
37	5	573.30							
24.40	14.40	573.00	33.40	722.30	31.00	896.10	16.00	975.80	0.0

LDETR.GEO (UNIT 1)

```
      38         8  573.40
  198.80     23.60  501.90     25.50  853.90     24.10  952.30      8.60 1887.40     7.0
 1980.10     25.80 2265.30     21.40 2428.00      0.00
      39         9  573.68
   92.20     24.90  301.20     31.00  445.60     28.10  680.60     35.20  969.40    22.0
 1051.70      8.10 1707.00      7.20 2242.90      4.30 2375.20      0.00
      40        11  573.76
 1500.00      0.01 1625.00     25.00 2000.00     25.00 2100.00      2.00 2300.00     0.0
 3750.00      0.00 3825.00      5.00 3900.00     32.00 4225.00     32.00 4375.00     2.0
 4675.00      0.00
      41        10  573.83
   25.00      2.00  625.00      1.00  750.00     27.00 1125.00     27.00 1175.00    22.0
 1250.00     30.00 1750.00     32.00 1925.00      6.00 2125.00      2.00 2150.00     0.0
      42        17  573.91
   50.00      3.00  475.00      3.00 1000.00     26.50 2000.00     27.20 2100.00    35.0
 2800.00     35.00 3100.00     24.80 3300.00     32.00 3750.00     30.00 4250.00    33.0
 4500.00     27.80 5100.00     28.00 5500.00      9.00 5900.00      7.50 6900.00     8.0
 7000.00     12.00 7100.00      0.00
      43        15  572.64
   72.80      3.30  292.90      6.40 2564.00     10.70 2709.50     26.60 3745.50    24.4
 3771.70     28.70 4424.00     36.20 4509.30     27.30 4996.00     25.00 5150.60    22.2
 5327.90     27.80 5482.00     28.30 5796.10     11.00 6110.80     13.10 6345.80     0.0
      44        12  572.70
  575.00      6.00  750.00      6.00 1000.00      5.00 2750.00      3.00 3175.00     6.0
 3375.00     18.00 4300.00     18.00 4375.00     27.00 4950.00     27.00 5400.00    18.0
 5900.00      6.00 6050.00      0.00
      45         8  573.76
   16.70      4.70  129.10     30.60  283.70     36.10  474.10     36.90  657.20    30.0
  812.00      8.70  866.90      7.40  898.10      0.00
      46        10  573.23
    4.00      3.00   44.90      3.50  243.80     32.30  536.30     33.00  779.30    31.4
  992.50      6.70 1152.00      4.30 1849.90      3.30 1940.20      1.90 2002.80     0.0
      47        11  573.45
   55.40     19.00  277.50     18.00  433.60     23.00  460.00     15.10  501.60    32.6
  772.50     34.60  822.90     21.00  848.20     25.30  978.40     10.50 1095.20     7.5
 1246.00      0.00
      48        10  572.93
   15.70      2.20   70.40      2.50  239.90     21.90  439.70      9.50  590.60     9.7
  876.40     27.30 1237.90     24.30 1376.90      7.70 1496.30      4.50 1523.20     0.0
      49        10  573.12
   47.90      3.60   99.50     23.90  636.30     16.10  756.00     19.90  837.70    17.6
  981.80     29.60 1143.90     26.50 1165.10     20.00 1379.10     14.70 1425.00     0.0
      50         4  572.83
   31.30     25.20  209.10     29.00  990.20     27.10 1145.90      0.00
      51        10  572.82
  226.10      7.50  274.60      6.50  766.00      7.80 1245.50     21.80 1345.30    19.1
 1805.80     15.70 1921.90     20.40 2395.50     15.50 2471.90      9.80 2638.50     0.0
      52        18  572.68
  319.50     19.60  405.70      8.20  559.00      9.30  881.80     22.80 1033.40     7.9
 1508.20      5.40 1676.10     21.90 2128.40     15.00 2194.10      9.00 2466.20     9.7
 2542.30     17.20 2646.40     13.60 2708.50      7.80 2770.80      7.10 3015.40     1.9
 3208.80      5.90 3431.90      7.20 3674.50      0.00
      53        11  572.60
  875.00      4.00 1000.00     18.00 2000.00     18.00 2175.00      5.00 2550.00     0.0
 4125.00      0.00 4625.00      2.00 4875.00     14.00 5125.00     14.00 5500.00     4.0
 6500.00      0.00
 1    10    10    10    10
 2     7    11     0     0
 3     5    11     0     0
 4     5    11     0     0
 5     4    10     0     0
```

LDETR.GEO (UNIT 1)

6	4	9	0	0
7	4	12	10	11
8	3	12	10	11
9	2	31	13	29
10	2	32	15	26
11	2	32	15	25
12	2	32	20	22
13	6	33	20	20
14	6	34	15	15
15	7	35	0	0
16	5	36	0	0
17	4	36	0	0
18	4	37	0	0
19	4	37	0	0
20	4	37	0	0
21	4	37	0	0
22	4	37	0	0
23	4	36	0	0
24	5	36	0	0
25	5	34	0	0
26	6	31	0	0
27	7	30	0	0
28	8	29	0	0
29	8	28	0	0
30	8	27	0	0
31	9	26	0	0
32	9	25	0	0
33	9	23	0	0
34	8	21	0	0
35	8	19	0	0
36	62	132	0	0
37	62	125	0	0
38	62	122	0	0
39	62	114	0	0
40	61	108	0	0
41	60	104	0	0
42	59	99	0	0
43	58	92	0	0
44	57	86	67	68
45	56	83	66	69
46	54	82	66	69
47	53	80	65	69
48	52	77	63	68
49	51	71	61	62
50	49	69	0	0
51	47	67	0	0
52	45	61	51	53
53	43	60	49	54
54	42	59	47	54
55	41	58	45	53
56	40	58	44	53
57	39	58	44	53
58	39	58	44	53
59	38	58	43	53
60	37	58	43	53
61	36	58	43	51
62	35	57	43	50
63	35	57	43	50
64	35	55	42	48
65	34	54	42	47
66	34	52	41	47

LDETR.GEO (UNIT 1)

67	33	51	40	46
68	33	50	39	45
69	33	49	38	44
70	33	49	37	43
71	33	48	36	43
72	32	47	36	41
73	32	46	36	40
74	31	45	36	38
75	31	44	36	36
76	30	43	0	0
77	30	41	0	0
78	29	39	0	0
79	29	37	0	0
80	28	36	0	0
81	27	35	0	0
82	27	34	0	0
83	26	33	0	0
84	25	32	0	0
85	25	31	0	0
86	24	30	0	0
87	23	29	0	0
88	22	28	0	0
89	22	27	0	0
90	21	26	0	0
91	20	25	0	0
92	19	24	0	0
93	18	23	0	0
94	18	23	0	0
95	17	22	0	0
96	16	21	0	0
97	15	20	0	0
98	14	19	0	0
99	13	18	0	0
100	13	17	0	0
101	12	17	0	0
102	12	16	0	0
103	12	15	0	0
104	11	15	0	0
105	11	14	0	0
106	10	14	0	0
107	10	13	0	0
108	9	13	0	0
109	9	12	0	0
110	9	12	0	0
111	9	12	0	0
112	8	12	0	0
113	8	11	0	0
114	8	11	0	0
115	8	11	0	0
116	8	11	0	0
117	8	11	0	0
118	8	11	0	0
119	8	11	0	0
120	8	11	0	0
121	8	11	0	0
122	8	11	0	0
123	7	10	0	0
124	7	10	0	0
125	7	10	0	0
126	7	10	0	0
127	7	10	0	0

LDETR.GEO (UNIT 1)

128	7	10	0	0
129	7	10	0	0
130	7	11	0	0
131	7	11	0	0
132	7	11	0	0
133	7	12	0	0
134	7	12	0	0
135	7	12	0	0
136	8	12	0	0
137	8	12	0	0
138	8	12	0	0
139	8	13	0	0
140	8	13	0	0
141	8	13	0	0
142	8	13	0	0
143	9	13	0	0
144	9	13	0	0
145	9	14	0	0
146	9	14	0	0
147	10	14	0	0
148	10	14	0	0
149	10	15	0	0
150	10	15	0	0
151	10	15	0	0
152	10	15	0	0
153	10	16	0	0
154	10	16	0	0
155	10	17	0	0
156	10	18	0	0
157	10	18	0	0
158	10	18	0	0
159	10	18	0	0
160	10	19	0	0
161	10	20	0	0
162	10	21	0	0
163	10	22	16	18
164	10	23	15	19
165	9	24	15	20
166	9	24	15	21
167	8	25	15	22
168	8	25	14	22
169	7	25	14	22
170	7	27	15	22
171	6	28	15	23
172	5	28	16	23
173	5	29	17	23
174	5	29	17	23
175	5	29	18	24
176	5	30	19	24
177	5	31	19	25
178	6	32	20	26
179	6	33	21	26
180	7	33	21	27
181	8	34	22	28
182	9	34	23	28
183	10	35	24	29
184	12	35	25	29
185	15	36	26	30
186	15	37	27	31
187	15	38	27	32
188	15	39	28	33

LDETR.GEO (UNIT 1)

189	16	39	28	33
190	16	40	29	34
191	16	40	30	35
192	17	40	30	36
193	17	41	31	36
194	17	41	31	37
195	17	42	32	37
196	17	43	33	37
197	17	43	20	20
198	18	44	20	21
199	18	43	20	21
200	18	43	20	21
201	18	43	20	22
202	19	43	21	22
203	19	43	21	23
204	19	42	21	24
205	20	41	21	24
206	20	44	22	25
207	20	44	22	25
208	21	45	22	26
209	21	45	22	26
210	21	45	23	26
211	21	45	24	27
212	21	45	24	27
213	21	49	24	27
214	21	49	24	28
215	20	49	24	29
216	20	52	24	30
217	20	52	24	32
218	20	52	23	33
219	20	52	23	34
220	20	52	23	35
221	20	52	23	36
222	20	52	23	37
223	21	53	24	38
224	22	53	24	38
225	22	54	25	39
226	23	54	26	39
227	23	54	26	40
228	24	55	27	40
229	24	55	27	41
230	24	56	27	41
231	25	56	27	42
232	25	57	27	42
233	25	57	28	43
234	25	58	28	44
235	25	59	28	44
236	25	59	28	45
237	24	60	28	45
238	25	61	28	46
239	25	47	29	47
240	26	47	29	47
241	26	47	29	47
242	27	48	29	48
243	27	48	30	48
244	27	48	30	48
245	28	48	31	48
246	28	49	31	49
247	28	49	32	49
248	29	52	32	52
249	29	50	33	50

```
250   30   50   33   50
251   30   50   33   50
252   31   51   33   51
253   31   51   34   51
254   31   51   34   51
255   32   51   34   51
256   32   52   35   52
257   32   52   35   52
258   33   53   36   53
259   33   53   37   53
260   33   54   37   54
261   33   54   38   54
262   32   55   38   55
263   33   55   39   55
264   32   55   39   55
265   34   55   40   55
266   34   56   40   56
267   35   57   41   57
268   35   57   42   57
269   35   57   42   57
270   35   57   42   57
271   35   57   42   57
272   35   57   44   57
273   36   57   45   57
274   36   56   46   56
275   37   50   48   50
276   38   50    0    0
277   39   49    0    0
278   40   43    0    0
279   41   48   47   48
280   41   49   47   49
281   42   50   47   50
282   43   52   47   52
283   44   53   49   53
284   44   54   51   54
285   45   55   52   55
76
9  10
11  12
11  13
10  13
10  29
12  15
12  16
12  25
13  15
173  6
173  7
173  26
173  27
174  27
174  28
175  27
175  28
176  29
177  29
178  13
178  30
179  13
179  14
179  15
```

LDETR.GEO (UNIT 1)

```
179 31
180 13
180 14
180 15
180 32
181 15
181 16
181 33
182 16
183 16
183 17
184 16
197 33
197 36
197 39
198 33
198 39
198 40
199 33
199 34
199 38
199 39
199 40
199 41
200 33
200 34
200 35
200 38
200 39
200 40
200 41
201 34
201 35
201 39
201 40
201 41
202 35
202 36
202 40
202 41
203 36
203 40
204 37
205 37
206 41
206 42
207 41
208 41
208 42
209 41
209 42
210 42
0
```

LDETR.ICE (UNIT 5)

```
0.035      12.5
1   1
 2   7   35   19
0.5
```

DETR.FLW (UNIT 7)

```
  6.0
573.72    149190.
573.61    149180.
573.61    184150.
573.61    120810.
573.46    120810.
573.46    184140.
573.14    184140.
573.01    184140.
573.01    153050.
572.67    153050.
572.67    115400.
572.31    115400.
572.31    146450.
573.69     34970.
573.61     34970.
573.61     63340.
573.46     63340.
573.01     31070.
572.31     31050.
572.67     37640.
571.97     37640.
571.35     37630.
    1
    2 OPEN
573.72    149190.
573.61    149180.
573.61    184150.
573.61    120810.
573.46    120810.
573.46    184140.
573.14    184140.
573.01    184140.
573.01    153050.
572.67    153050.
572.67    115400.
572.31    115400.
572.31    146450.
573.69     34970.
573.61     34970.
573.61     63340.
573.46     63340.
573.01     31070.
572.31     31050.
572.67     37640.
571.97     37640.
571.35     37630.
    1
    2 OPEN
```

LDETR.BND (UNIT 8)

(These are approximate values based on engineering judgement to illustrate the model capability)

```
1   1   285  10
2   1   285  10
3   7    14  10
3  44    49  10
3  52    75  10
3 163   275  10
3 279   285  10
4   7    14  10
4  44    49  10
4  52    75  10
4 163   275  10
4 279   285  10
0   0     0   0
```

LDETR.SPL (UNIT 12)

```
Fuel Oil No. 2
6.0   3   0   0   1   0   10.   -1.0   -1.0
500  10000.  900.  0.84  1.411E-5  2.06E-3 1.14 .98  1.6  1.39,1.39,1.43
-7999.    42000.   .7063E-02  .1873E-02  7.88   465.0
2.933  0.0   50.0
2.933  0.0   50.0
2.933  0.0   50.0
2.933  0.0   50.0
2.933  0.0   50.0
2.933  0.0   50.0
2.933  0.0   50.0
2.933  0.0   50.0
2.933  0.0   50.0
```

LAKEWIND.DAT (UNIT 10)

```
  0.    42.42    82.42    30.    64.    54.    15.0  180.
  3.    42.42    82.42    30.    64.    54.    15.0  180.
  6.    42.42    82.42    10.    65.    54.    15.0  150.
  9.    42.50    82.58    10.    66.    54.    12.0  180.
 12.    42.50    82.58    10.    66.    55.    10.0  270.
 -1.
```

LAKEBATH.DAT (UNIT 13)

```
LAKE ST. CLAIR BATHYMETRY
  36    38  42.3041534  82.9315796 4000    19      1 -2.24 39.677 10.087-120.06
   0.822690E+02   -0.418687E+01  -0.892958E+00   0.549244E+00
   0.284351E+01    0.110232E+03   0.182918E+00   0.235336E+00
   0.121387E-01    0.462637E-03   0.110856E-05  -0.102938E-05
  -0.313049E-03    0.905963E-02  -0.238882E-06  -0.279918E-06
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    3    5    4    3    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    3    5    6    5    4    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    5    7
   7    8    7    7    5    3    0    0    0    0    6    8    6    6    5    5    4    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    2    4    8    8   10   10
   9    9    6    4    0    0    0    3   10   10   10   12   11   11    9    7    5    0    0
   0    0    0    0    0    0    0    0    0    0    0    6    8    9    9   10   10   10    8
   9    6    3    4    0    5   11   12   11   10   12   11   11   10    9    8    0    0    0
   0    0    0    0    0    0    0    0    3    9    9    6    4    5    8    9    9   10    9
   6    4    5    5    9   12   13   13   13   13   13   13   12    8    2    0    0    0    0
   0    0    0    0    0    0    3    9    6    1    2    3    2    2    3    6    8    7    6
   7    9   12   13   13   14   14   14   14   14   14   13   14   12    6    2    0    0    0
   2    3    0    0    1    6    1    2    2    1    2    1    1    1    3    6    9   10   11
  13   14   14   15   15   15   15   14   14   14   14   13   13   12   12   11   10    8   15
   0    0    3    5    1    2    0    0    0    0    2    1    2    4    6   10   12   13   15
  14   14   16   15   14   15   15   15   15   14   14   14   13   13   13   12   12    0    0
   2    2    1    0    0    0    0    1    3    3    3    2    1    7   14   15   14   17   16
  17   17   16   15   15   15   16   15   12   13   10    8   10   10   10    0    0    0    0
   0    0    0    1    1    1    2    0 [illegible] 1    2    3   13   15   14   14   17   16   15
  15   16   18   16   16   14   12   13   10   12   11   10    7    0    0    0    0    0    0
   0    0    0    0    0    0    1    2    2    3   12   14   15   16   16   17   17   16   17
  16   16   15   14   15   13   12   13   12   10    6    0    0    0    0    0    0    0    0
   0    0    1    1    2    2    2    2   14   14   15   14   16   15   16   17   17   18   16
  14   17   13   15   15   11   11   10    7    0    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    1   10   15   16   16   17   16   17   19   17   19   18   17   12
  14   15   14   13   12    9    7    0    0    0    0    0    0    0    0    0    0    0    0
   0    1    2    6   13   15   17   17   17   17   17   17   19   17   18   17   16   16   15
  14   11   11    8    5    0    0    0    0    0    0    0    0    0    0    0    0    1    1
   7   12   14   16   17   17   17   18   18   19   18   18   17   17   17   16   15   14   13
  12   10    5    0    0    0    0    0    0    0    0    0    0    0    0    1    1    6   12
  13   15   16   17   18   18   18   18   18   19   18   16   17   15   15   15   13   10    8
   3    0    0    0    0    0    0    0    0    0    0    0    0    1    1    3    7   12   13
  16   17   18   18   18   18   17   17   17   17   16   16   16   15   13   12   10    3    0
   0    0    0    0    0    0    0    0    0    0    0    0    0    2    2   10   12   17   18
  18   17   17   17   16   17   17   15   16   16   16   15   13   11    5    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    2    1    1    6   13   17   18   17   17
  17   16   16   16   16   16   17   16   15   13   12   10    4    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    2    2    2    6   13   17   17   17   17   17   17
  17   15   16   15   15   15   15   14   11    9    0    0    0    0    0    0    0    0    0
   0    0    0    0    0    1    3    4    2    6   14   17   17   17   18   17   16   15   16
  15   12   15   15   13   11    9    2    0    0    0    0    0    0    0    0    0    0    0
   0    0    0    2    3    3    4    6   14   16   16   16   17   17   16   15   16   15   15
  14   11   12   10    5    0    0    0    0    0    0    0    0    0    0    0    0    0    0
   0    0    1    3    2    9   13   14   15   17   17   17   15   15   15   15   15   13   12
   9    6    2    0    0    0    0    0    0    0    0    0    0    0    0    0    0    1    1
   2    3    4   11   12   14   15   15   17   17   16   15   15   15   14   13   11    9    2
   0    0    0    0    0    0    0    0    0    0    0    0    0    0    1    1    3    2    4
   3   11   11   13   15   17   17   16   15   15   15   14   12   12    9    6    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    0    1    1    3    2    3    6   11
  10   15   15   16   17   17   16   15   14   13   11    9    6    0    0    0    0    0    0
   0    0    0    0    0    0    0    0    0    0    0    2    3    3    3    5   10   10   14
  14   15   16   16   15   13   12   10    8    3    0    0    0    0    0    0    0    0    0
```

LAKEBATH.DAT (UNIT 13)

0	0	0	0	0	0	0	2	3	4	5	5	3	6	10	11	13	15	17
16	15	13	10	9	7	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	2	1	2	6	7	9	10	10	11	17	16	15	15
13	10	7	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	6	10	10	10	12	12	16	17	15	13	12	9
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	6	11	11	12	17	17	13	13	11	8	2	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	2	6	13	11	11	17	13	12	12	11	6	2	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	2	5	12	12	12	11	11	9	4	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	10	10	10	9	5	4	2	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	2	2	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1.																	

CHAPTER IV

STREAM FUNCTION AND BATHYMETRY OF LAKE

IV.1 Introduction

The information in this chapter is needed only if lake bathymetry data needs to be changed. The input files LAKEBATH.DAT and LAKEINIT.PSI for Lake St. Clair can be revised if a smaller grid size or new boundary conditions are desired. The work is rather tedious and requires attention to details covered in this chapter, but it can be done. The report by Schwab and Sellars (1981) was used to direct the collection of the current Lake St. Clair files and may be consulted.

Figures 18 and 19 illustrate what the depth and stream function files look like respectively when printed out as two dimensional arrays. Areas in the lake which represent land are masked with a special value (SPVAL). This causes the stars (*) to be printed. The stars in the depth array (Figure 18) represent a depth of zero (0) feet. The stars in the stream function array (Figure 19) represent a very large number (i.e. 1.0E+21 currently used.) The SPVAL for the lake is used as an indicator for locating a boundary and is not part of the actual calculation of stream functions.

Close examination of the positions occupied by depths and stream functions quickly reveals that not every array location that contains a stream function value has a depth. The reason is partly because the stream functions not only serve to give the discharge between grid points, but they are also used to establish both the no flux boundary condition at the lake shore and the discharge conditions at river mouths. Furthermore, the additional boxes

```
0 VALUES MULTIPLIED BY 10**  1
***************************************************************************************************************
****************************************************************** 20  20  10  20  10******************************************
************************************************************* 10  10 100 100 100  90  50  40  20**************************************
********************************************************** 10  20  50 120 120 120 110 110  90  40**************************************
**************************************************** 20  60 130 110 110 170 130 120 120 110  60  20*************************************
************************************************** 10  10  60 110 110 120 170 170 130 130 110  80  20*********************************
*************************************** 10  10  10  60 100 100 100 120 120 160 170 150 130 120  90  40*********************************
*********************************** 10  20  10  20  60  70  90 100 100 110 170 160 150 150 130 100  70  30****************************
*********************************** 20  30  40  50  50  30  60 100 110 130 150 170 160 150 130 100  90  70  10************************
******************************************** 20  30  30  30  50 100 100 140 140 150 160 160 150 130 120 100  80  30********************
*************************************** 10  10  30  20  30  60 110 100 150 150 160 170 170 160 150 140 130 110  90  60*****************
*************************************** 10  10  30  20  40  30 110 110 130 150 170 170 160 150 150 150 140 120 120  90  60*************
******************************************* 10  10  20  30  40 110 120 140 150 150 170 170 160 150 150 150 140 130 110  90  20**********
************************************************** 10  30  20  90 130 140 150 170 170 170 150 150 150 150 150 130 120  90  60  20********
*********************************************** 20  20  30  40  60 140 160 160 160 170 170 160 150 160 150 150 140 110 120 100  50*******
*********************************************** 10  30  40  20  60 140 170 170 170 180 170 160 150 160 150 120 150 150 130 110  90  20****
*************************************************** 20  20  20  60 130 170 170 170 170 170 170 170 150 160 150 150 150 150 140 110  90****
*************************************************** 20  10  10  60 130 170 180 170 170 170 160 160 160 160 160 170 160 150 130 120 100  40***
****************************************************** 20  20 100 120 170 180 180 170 170 170 160 170 170 150 160 160 160 150 130 110  50***
*********************************************** 10  10  30  70 120 130 160 170 180 180 180 180 170 170 170 170 160 160 160 150 130 120 100  30****
*********************************************** 10  10  60 120 130 150 160 170 180 180 180 180 180 190 180 160 170 150 150 150 130 100  80  30****
*********************************************** 10  10  70 120 140 160 170 170 170 180 180 190 180 180 170 170 170 160 150 140 130 120 100  50****
*************************************************** 10  20  60 130 150 170 170 170 170 170 170 190 170 180 170 160 160 150 140 110 110  80  50****
********************************************************* 10 100 150 160 160 170 160 170 190 170 190 180 170 120 140 150 140 130 120  90  70****
*************************************** 10  10  20  20  20  20 140 140 150 140 160 150 160 170 170 180 160 140 170 130 150 150 130 110 100  70****
*********************************************** 10  20  20  30 120 140 150 160 160 170 170 160 170 160 160 150 140 150 130 120 130 120 100  60****
*************************** 10  10  10  20**** 20  10  20  30 130 150 140 140 170 160 150 150 160 180 160 160 140 120 130 100 120 110 100  70****
****  20  20  10*************** 10  30  30  30  20  10  70 140 150 140 170 160 170 170 160 150 150 150 160 150 120 130 100  80 100 100 100****
****  30  50  10  20*************** 20  10  20  40  60 100 120 130 150 140 140 160 150 140 150 150 150 150 140 140 140 130 130 130 120 120****
****  10  60  10  20  20  10  20  10  10  10  30  60  90 100 110 130 140 140 150 150 150 150 140 140 140 140 130 130 120 120 110 100  80 150****
****  30  90  60  10  20  30  20  20  30  60  80  70  60  70  90 120 130 130 140 140 140 140 140 140 130 140 120  60  20************  20  30****
****  30  90  90  60  40  50  80  90  90 100  90  60  40  50  50  90 120 130 130 130 130 130 130 120  80  20*****************************
********  60  80  90  90 100 100 100  80  90  60  30  40****  50 110 120 110 100 120 110 110 100  90  80*******************************
********  20  40  80  80 100 100  90  90  60  40************  30 100 100 100 120 110 110  90  70  50**********************************
***************  50  70  70  80  70  70  50  30***************  60  80  60  60  50  50  40*************************************
***************************  30  50  60  50  40*****************************************************************************
*****************************  30  50  40  30***************************************************************************
***************************************************************************************************************
```

Figure 18 Depth array for Lake St. Clair

0 VALUES MULTIPLIED BY 10** -3

```
************************************************************************************************************************************************
****************************************************************-101-101-101-101-101-101********************************************************
********************************************************-101-101-101-100-100-100-100-101-101-101************************************************
****************************************************-101-101-100-100 -99 -99 -99 -99-100-100-101************************************************
************************************************-101-101-100-100 -99 -98 -96 -97 -98 -98-100-101-101********************************************
********************************************-101-101-100-100 -99 -99 -97 -95 -95 -96 -97 -99-100-101********************************************
********************************-101-101-101-101-100-100-100 -99 -98 -97 -95 -94 -94 -96 -98-100-101********************************************
****************************-101-101-101-101-100-100-100-100 -99 -98 -97 -95 -93 -93 -94 -97 -99-101-101****************************************
****************************-101-101-101-100-100-100-100-100 -99 -98 -97 -95 -93 -91 -93 -95 -98-100-101-101************************************
****************************-101-101-101-101-100-101-100-100 -99 -98 -97 -95 -92 -90 -91 -93 -96 -98-100-101-101********************************
********************************-101-101-101-101-101-100-100 -99 -98 -96 -94 -91 -88 -88 -90 -92 -95 -97-100-101-101****************************
********************************-101-101-101-101-101-100-100 -99 -98 -95 -92 -89 -86 -84 -86 -88 -91 -94 -97 -99-101-101************************
********************************-101-101-101-101-101-100-100 -99 -97 -94 -91 -87 -83 -80 -81 -83 -86 -89 -92 -96 -99-101-101********************
************************************-101-101-101-101-101-100 -99 -95 -92 -88 -84 -80 -77 -77 -79 -81 -84 -87 -92 -96-100-101********************
****************************************-101-101-100-100-100 -99 -94 -90 -85 -81 -76 -72 -72 -74 -76 -78 -83 -88 -93 -97-101-101****************
****************************************-101-100 -99 -99 -99 -98 -94 -88 -82 -77 -72 -68 -67 -58 -70 -73 -78 -84 -89 -94 -98-101-101************
****************************************-101-101 -98 -96 -97 -98 -94 -87 -80 -74 -68 -64 -62 -62 -64 -68 -73 -78 -83 -88 -94 -99-101************
********************************************-101 -96 -93 -95 -97 -93 -85 -77 -70 -64 -59 -56 -56 -58 -61 -65 -69 -75 -81 -89 -97-101-101********
******************************************** -92 -92 -89 -91 -95 -91 -84 -75 -66 -60 -54 -50 -49 -50 -52 -55 -60 -66 -73 -82 -91 -99-101********
**************************************** -92 -92 -69 -79 -87 -92 -88 -81 -72 -63 -55 -48 -44 -42 -41 -42 -45 -49 -56 -64 -74 -85 -95-101-101****
**************************************** -92 -87 -78 -80 -86 -88 -83 -77 -68 -59 -51 -43 -37 -33 -31 -31 -34 -38 -45 -54 -65 -77 -89 -98-101****
**************************************** -92 -88 -85 -82 -84 -83 -78 -71 -63 -55 -46 -38 -30 -24 -20 -20 -22 -26 -34 -43 -56 -70 -84 -96-101****
**************************************** -92 -92 -89 -88 -87 -82 -75 -66 -57 -48 -40 -31 -23 -15 -10  -8  -9 -13 -22 -33 -46 -62 -78 -93-101****
******************************************** -92 -92 -92 -89 -83 -71 -60 -50 -40 -32 -23 -14  -6   1   5   4  -1  -9 -21 -35 -52 -70 -87 -98****
******************************** -20 -20 -20 -20 -20 -92 -86 -78 -64 -52 -41 -31 -22 -14  -5   4  12  17  17  12   5  -6 -20 -38 -5' -83 -93****
******************************** -20 -20 -20  -9 -28 -57 -63 -63 -51 -40 -30 -20 -10  -2   6  14  23  28  29  26  21  12  -1 -19 -39 -58 -79****
********************  38  38  38  38  38  38  11 -12 -31 -40 -43 -34 -26 -17  -7   2  10  18  26  34  40  43  42  38  31  20   4 -16 -36 -64****
 101 101 101 101****  38  38  38  38  38  38  18   0 -13 -21 -21 -15  -9  -2   7  16  24  30  37  45  52  56  57  54  49  40  28  11 -12 -58****
 101 100  99 101 101********  38  38  36  32  22  10   3   1   4   7  10  15  21  29  37  43  50  57  64  70  71  71  70  64  56  45  28 -34****
 101  99  97  98 101 101  38  38  38  39  36  32  28  28  30  30  29  30  32  36  42  49  55  62  69  76  83  86  89  94  92  90  86  65  12****
 101  98  94  93  91  81  56  46  43  44  45  48  53  59  61  56  51  48  48  51  56  61  67  74  81  89  96  97  99 101 101 101 101  94  94****
 101  99  94  89  84  75  62  53  51  54  61  70  78  86  88  78  69  64  64  65  68  73  78  84  91  98 101 101 101 101******** 101 101 101****
 101 101  97  91  83  76  68  64  65  71  81  90  94 101 101  93  83  78  77  78  80  84  88  93  99 101 101************************************
**** 101  99  97  90  85  81  79  79  85  94 101 101 101 101  98  92  89  89  89  90  93  97  99 101 101****************************************
**** 101 101 101  98  96  95  91  90  93  98 101******** 101 101  98  98  98  98  98  99 101 101 101********************************************
************ 101 101 101 101  98  96  97  99 101************ 101 101 101 101 101 101 101 101****************************************************
************************ 101 101  99  99 100 101************************************************************************************************
**************************** 101 101 101 101 101************************************************************************************************
```

Figure 19 Initial streamfunction array for Lake St. Clair with assigned stream functions are required by the second order differencing technique used to solve the stream function equation. The no flux condition along the lake boundary is accomplished by setting stream function values in adjacent grids equal. The difference in stream function values across the river mouth corresponds to the river discharge into or out of the lake. Figure 20 shows the discharge points, percentages and directions used in the current Lake St. Clair stream function file. Relating Figure 20 to Figure 19, all numbers bordering the stars across the bottom and top of the lake are constant and numbers bordering the stars on the left and right side change as discharge points are encountered.

IV.2 Lake Bathymetry Data

This section will focus on the procedure for setting up a new depth array for Lake St. Clair. First, it is necessary to draw a new grid over the lake chart (National Ocean Survey Chart 14850) with the new grid size. To avoid changing any more additional parameters than absolutely necessary, grid (1,1) should originate from the same point as grid (1,1) does in Figure 7 and the current grid size (4000 ft) must be divisible to a whole number by the new grid size. Then, the geographic to map and map to geographic conversion coefficients will not be affected. These coefficients should never be changed unless a new base origin is used and the coefficients recalculated. The current model uses the original base computed by Schwab and Sellars (1981).

Second, parameters IPARM(1) and IPARM(2) are computed by counting the number of grids from grid (1,1) to the lake river interface plus one (1) and the number of grids from the x axis to at least one grid past the upper shoreline. Parameter RPARM(3) is changed to the new grid size. Then,

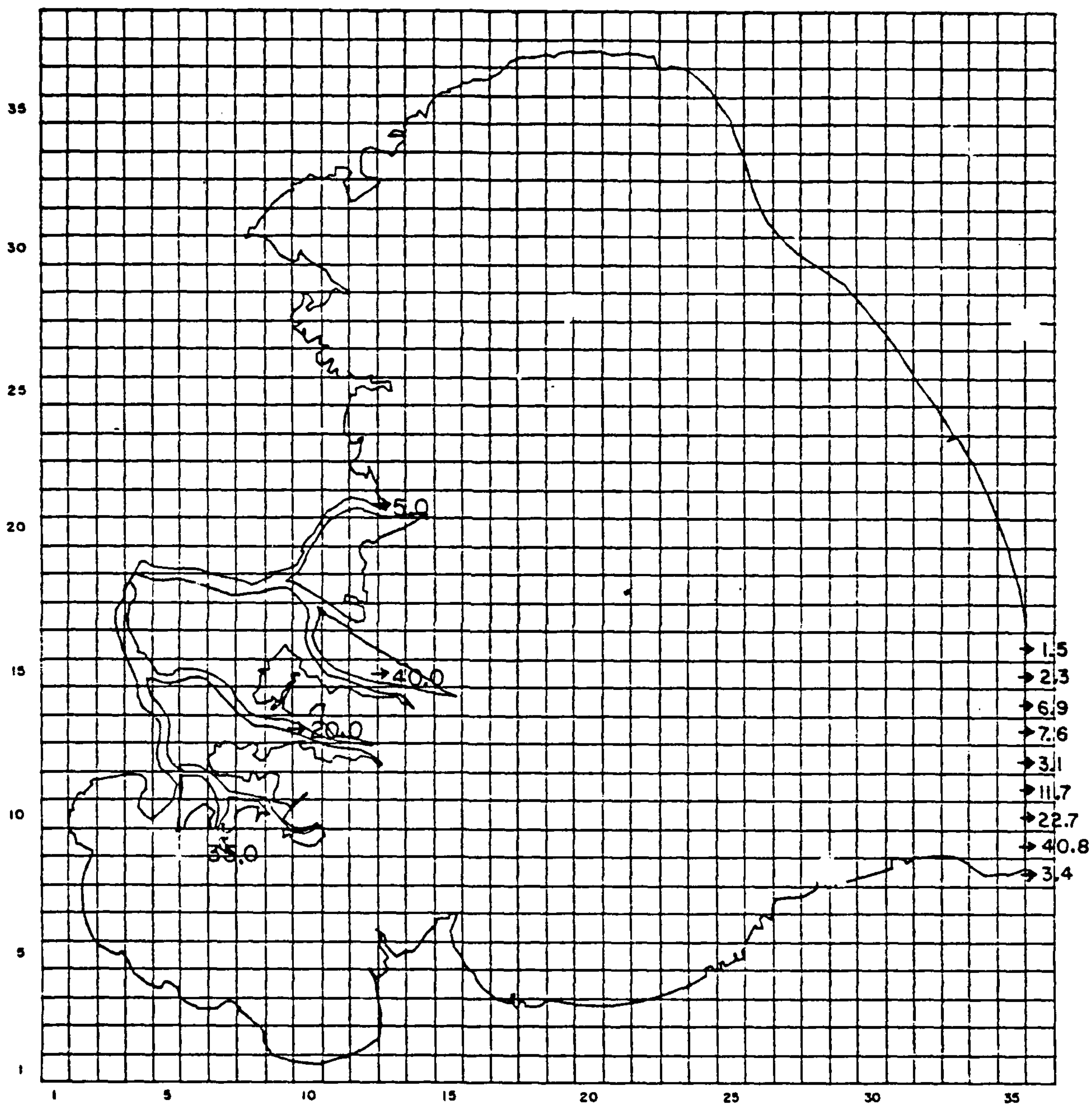


Figure 20 Discharge points, percentages of total discharge at the points, and discharge directions used in the current LAKEINIT.PSI file

ZPARM(1) and ZPARM(2) are multiplied by the ratio of the old grid size to the new grid size.

Third, the grids which fall within the lake shore will be assigned an average of the sounding depths shown on the chart. Islands are currently assigned the minimum depth of the lake since the model does not take into account the proper boundary conditions around them. So, if the minimum or maximum lake depths change, RPARM(4) (and the island depths) and RPARM(5) must reflect these changes. At this point, the new LAKEBATH.DAT file can be assembled. Note that lake depths are read from the file starting with the bottom row (left to right) in the lake and going up, so they must be set up in that order within the file.

Finally, the shore limits (IGRIUB(), IGRILB(), IGRUB1(), IGRLB1()) must be established based upon the grids having assigned depths. This means that the new shore limits, LGRIDX, NGRIDX and DXL must be changed in LDETR.GEO.

IV.3 Stream Function File (LAKEINIT.PSI)

Now that the grids containing the depths are known, the stream functions can be assigned. If the grid size was not changed, the procedure given here could be used to set up new boundary conditions in the lake.

The first step is to select the total discharge at which the file will be set (Figure 19 was set for 201,323 cfs, numbers shown are rounded off) and locate the grids containing the discharge points. Then, every grid on the boundary of the lake containing a depth other than zero (0) will be assigned a stream function value. The easiest way to choose what number and where to start is to take half the total discharge magnitude (100661.5 cfs in Figure 19), give it a positive sign, and assign that number at the lower side of the mouth of the Detroit River. Then subtract discharges from this number from the subsequent discharge points above when going from one grid to the next and assign the result to the corresponding grid. When the top of the mouth to the Detroit River is reached, the stream function value should be the same magnitude as at the lower side of the mouth but negative in sign (-100,661.5 cfs in Figure 19). Proceeding counterclockwise around the lake, the stream function will stay constant until the next discharge point is encountered and the appropriate value recorded for that grid. Finally, all grids within the lake are assigned interpolated values between the boundaries.

According to the definiton of a grid i,j as shown in Figure 3, it is now possible to complete the assignment of stream functions to the additional grids mentioned earlier. Every grid i,j in the lake must have a stream function at every corner to apply the second order differencing technique. The only way for all four corners to have stream function values is for adjacent grid boxes to have stream functions. By starting at any boundary grid and proceeding around the lake, each grid with an assigned depth is checked to see if all four corners will have assigned stream functions. If not, the additional boxes required to fill the need are assigned a stream function (of the same numerical value as the box which was checked.) Upon completion of this step, all grids which require a stream function value should have one except the land grids. These are assigned the number SPVAL.

The LAKEINIT.PSI file can now be set up. Again, the numbers are read in starting from the bottom row (left to right) in the lake array and proceed up to the highest row. Extreme caution is advised when typing numbers since an error could cause instability in the computations.

IV.4 Calculating Stream Function Values

Simple hand calculations do not give accurate interior stream function values. A program was set up to accurately determine the stream functions at the desired total discharge. The program listing (PSISET.F) for calculating stream function values is given in Appendix III. This program computes the steady state stream function values for the given total discharge and boundary condition using the unsteady finite difference scheme. The input files required to run the program are LAKEBATH.DAT, LAKEINIT.PSI and LAKEWIND.DAT. LAKEINIT.PSI is the file just created. LAKEWIND.DAT is the meteorological data file with all wind speeds set to zero (0). LAKEBATH.DAT is the same as that detailed in Chapter III and described (if revised for new grid size) in this chapter with the addition of five (5) more variables added to Card 3 after the time step DT. A typical LAKEBATH.DAT and LAKEWIND.DAT file is shown in Figure 21. The 5 added variables are:

Variable Name	Type and Length	Column Number	Definition
TT	G8.2	9-16	total time to run program, suggest using 1500 hours
RWD	G8.2	17-24	reference water level for depths, 571.71 if depths taken off of Chart 14850
CLWL	G8.2	25-32	current lake water level corresponding to current discharge TLKQ
TLKQ	G8.0	33-40	total lake discharge at CLWL
RLKQ	G8.0	41-48	reference discharge which was used when setting up the boundary stream function values

```
LAKE ST. CLAIR BATHYMETRY
  36   38  42.3041534  82.9315796 4000   19    1 -2.24 39.677 10.087-120.06
  0.822690E+02  -0.418687E+01  -0.892958E+00   0.549244E+00
  0.284351E+01   0.110232E+03   0.182918E+00   0.235336E+00
  0.121387E-01   0.462637E-03   0.110856E-05  -0.102938E-05
 -0.313049E-03   0.905963E-02  -0.238882E-06  -0.279918E-06
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   3   5   4   3   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   3   5   6   5   4   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   5   7
   7   8   7   7   5   3   0   0   0   0   6   8   6   6   5   5   4   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   2   4   8   8  10  10
   9   9   6   4   0   0   0   3  10  10  10  12  11  11   9   7   5   0   0
   0   0   0   0   0   0   0   0   0   0   0   6   8   9   9  10  10  10   8
   9   6   3   4   0   5  11  12  11  10  12  11  11  10   9   8   0   0   0
   0   0   0   0   0   0   0   0   3   9   9   6   4   5   8   9   9  10   9
   6   4   5   5   9  12  13  13  13  13  13  13  12   8   2   0   0   0   0
   0   0   0   0   0   0   3   9   6   1   2   3   2   2   3   6   8   7   6
   7   9  12  13  13  14  14  14  14  14  14  13  14  12   6   2   0   0   0
   2   3   0   0   1   6   1   2   2   1   2   1   1   1   3   6   9  10  11
  13  14  14  15  15  15  15  14  14  14  14  13  13  12  12  11  10   8  15
   0   0   3   5   1   2   0   0   0   0   2   1   2   4   6  10  12  13  15
  14  14  16  15  14  15  15  15  15  14  14  14  13  13  13  12  12   0   0
   2   2   1   0   0   0   0   1   3   3   3   2   1   7  14  15  14  17  16
  17  17  16  15  15  15  16  15  12  13  10   8  10  10  10   0   0   0   0
   0   0   0   1   1   1   2   0   2   1   2   3  13  15  14  14  17  16  [illegible]
  15  16  18  16  16  14  12  13  10  12  11  10   7   0   0   0   0   0   0
   0   0   0   0   0   0   1   2   2   3  12  14  15  16  16  17  17  16  17
  16  16  15  14  15  13  12  13  12  10   6   0   0   0   0   0   0   0   0
   0   0   1   1   2   2   2   2  14  14  15  14  16  15  16  17  17  18  16
  14  17  13  15  15  13  11  10   7   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   1  10  15  16  16  17  16  17  19  17  19  18  17  12
  14  15  14  13  12   9   7   0   0   0   0   0   0   0   0   0   0   0   0
   0   1   2   6  13  15  17  17  17  17  17  17  19  17  18  17  16  16  15
  14  11  11   8   5   0   0   0   0   0   0   0   0   0   0   0   0   1   1
   7  12  14  16  17  17  17  18  18  19  18  18  17  17  17  16  15  14  13
  12  10   5   0   0   0   0   0   0   0   0   0   0   0   0   1   1   6  12
  13  15  16  17  18  18  18  18  18  19  18  16  17  15  15  15  13  10   8
   3   0   0   0   0   0   0   0   0   0   0   0   0   1   1   3   7  12  13
  16  17  18  18  18  18  17  17  17  17  16  16  16  15  13  12  10   3   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   2   2  10  12  17  18
  18  17  17  17  16  17  17  15  16  16  16  15  13  11   5   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   2   1   1   6  13  17  18  17  17
  17  16  16  16  16  16  17  16  15  13  12  10   4   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   2   2   2   6  13  17  17  17  17  17  17
  17  15  16  15  15  15  15  14  11   9   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   1   3   4   2   6  14  17  17  17  18  17  16  15  16
  15  12  15  15  13  11   9   2   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   2   3   3   4   6  14  16  16  16  17  17  16  15  16  15  15
  14  11  12  10   5   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   1   3   2   9  13  14  15  17  17  17  15  15  15  15  15  13  12
   9   6   2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   1
   2   3   4  11  12  14  15  15  17  17  16  15  15  15  14  13  11   9   2
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   1   3   2   4
   3  11  11  13  15  17  17  16  15  15  15  14  12  12   9   6   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   1   1   3   2   3   6  11
  10  15  15  16  17  17  16  15  14  13  11   9   6   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   C   0   2   3   3   3   5  10  10  14
  14  15  16  16  15  13  12  10   8   3   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   2   3   4   5   5   3   6  10  11  13  15  17
  16  15  13  10   9   7   1   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   1   2   1   2   6   7   9  10  10  11  17  16  15  15
  13  10   7   3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   1   1   1   6  10  10  10  12  12  16  17  15  13  12   9
   4   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   1   1   6  11  11  12  17  17  13  13  11   8   2   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   2   6  13  11  11  17  13  12  12  11   6   2   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   1   2   5  12  12  12  11  11   9   4   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   1   1  10  10  10   9   5   4   2   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   2   2   1   2   1   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
       1.      50.  571.71   574.00 190000. 201323.
```

```
       0.        42.5      82.75        30.         64.         54.         00.         270.
      -1.
```

Figure 21 Typical LAKEBATH.DAT and LAKEWIND.DAT files for creating an initial stream function file

NOTE: TLKQ will become RLKQ at the end of program execution. This value and RWD must be inserted into subroutines RLID and PARTIC (DATA statement for RLKQ and RWD) prior to using the new files in the oil spill simulation. The output from the program will be a stabilized stream function array stored in a file named STREAM.DAT. This is the file which will be used as input to the oil spill simulation model but under the file name LAKEINIT.PSI.

CHAPTER V

MODEL OUTPUT

The amount of output generated is governed by several parameters and subroutines. The option number assigned to the parameter determines whether or not a specific portion of output will be generated. The following table summarizes these parameters and their functions:

Parameter	Yes	No	Function
IOPT1	1	0	Call subroutine PRINT to write fixed geometry of river to file OILPRT.OUT
IOPT2	1	0	Write location and magnitude of streamtube velocities to file VELSTR.OUT and depth-averaged grid velocities to VELCAR.OUT
IOPT3	1	0	Write locations of oil particles to LDETRSP.OUT
IOPT4	1	0	Call subroutine PLOTNU to write number plot of particle locations
INDPRN	1	0	Call subroutine PGPARM to write lake model input data and subroutine PRNT to write lake depths, lake stream functions, and lake ice region locations

V.1 Sample Output

The following figures represent selected output and graphics for two sample runs of the lake-to-river oil spill model. The first sample run (Figures 22 through 27) is for an instantaneous spill near the lake river interface. Figure 27 shows the graphical output that corresponds to results in Figs. 24 to 26. The second case (Figures 28 through 39) is on the opposite side of Lake St. Clair where the St. Clair River discharges into the lake. The graphical plots are not a direct form of output from the oil spill

Lake St. Clair and Detroit River

```
*************************
*  INSTANTANEOUS SPILL  *
*          AT           *
*    -7999., 42000.     *
*************************
```

SIMULATION PERIOD = 6.0 Hrs

Characteristics of spill

```
No. of particles           :  500
Oil spilled                :  10000. gals of Fuel Oil No. 2
DT for spill simulation    :    900. Secs.
Specific gravity of oil    :    0.84 (API index = 37.0)
Kinematic Visco. of Water  :0.1411E-04 sq ft/sec
Surafce Tension            :0.2060E-02 lbs/ft
```

Spreading Coefficients

K2i	K2v	K2t	c10	c20	c30
1.14	0.98	1.60	1.39	1.39	1.43

```
Molar volume               :0.7063E-02 cu ft/mol
Solubility of fresh oil    :0.1873E-02 lbs/cu ft
Viscosity of Oil           :    0.84 lbs/ft-sec
Manning s Roughness of Ice :   0.035
```

Surface Diffusion

LAKE - Default formulation is used

RIVER- Default formulation is used

API option is not selected . Evap. constants are C = 7.88   T0 = 465.0

Time step for river flow computation = 6.00Hrs

Figure 22 Description of spill type and location, oil properties, and various coefficients

Open Water Conditions exist in the river

Flow and Discharge Conditions in the River

Branch	Q (cfs)	Stage (ft)
1	184160.	573.72
2	149190.	573.72
3	34970.	573.69
4	184150.	573.61
5	63340.	573.61
6	120810.	573.61
7	184140.	573.46
8	184140.	573.14
9	184140.	573.01
10	153050.	573.01
11	153050.	572.67
12	115400.	572.67
13	31070.	573.01
14	146470.	572.31
15	37640.	572.67
16	37640.	571.97

Open Water conditions exist in the lake

Meteorological Station Data Used in Lake Circulation Model

Time hrs	Lat. deg	Long. deg	Height ft	T-air F	T-H2O F	Wind mph	deg
0.0	42.42	82.42	30.0	50.0	54.0	2.0	0.0
3.0	42.42	82.42	30.0	50.0	54.0	2.0	0.0

Figure 23 Stage and discharges at river branches and meteorological data

```
--------------------------------------------------------------------------------
Time =    0.25 Hrs -- Wind :mag=  2.0 mph, dir =  0.0 deg    -- Air Temp= 50.0 F
Spill center after advection=  -7378., 41536. (ft)
Volume per particle          =  20.000 gals

      Slick condition during this time step

Slick information by pie / strip
Pie  No. of particles  Mean radius(ft)
 1         59             90.
 2         77             93.
 3         60             90.
 4         53             89.
 5         57             90.
 6         44             88.
 7         61             91.
 8         49             90.


      Slick condition at the end of this time step

Fraction Evaporated = .60670E-03
Amount Dissolved (gals)       : This Step = .27559E-01 Total = .27559E-01
```

Figure 24   Spill information at t = 15 mins

```
Time =   1.00 Hrs -- Wind :mag=  2.0 mph, dir =  0.0 deg   -- Air Temp= 50.0 F
Spill center after advection=  -5518., 40153. (ft)
Volume per particle          =  19.878 gals

     Slick condition during this time step

Slick information by pie / strip
Pie  No. of particles  Mean radius(ft)
 1         64              291.
 2         66              281.
 3         58              262.
 4         62              280.
 5         59              276.
 6         56              263.
 7         71              274.
 8         53              288.


      Slick condition at the end of this time step

Fraction Evaporated = .14277E-01
Amount Dissolved (gals)      : This Step = .37287     Total = .64506
```

Figure 25 Spill Information at t = 1 hr

```
-----------------------------------------------------------------------------
Time =   4.00 Hrs -- Wind :mag=  2.0 mph, dir =  0.0 deg   -- Air Temp= 50.0 F
Spill center after advection=   3168., 32298. (ft)
Volume per particle          =  15.189 gals

      Slick condition during this time step

Slick information by pie / strip
 Strip  Particles  -Le(ft)  X-mean  Le(ft)
 -124       3        -237.     0.      0.
 -123       6        -111.     0.     79.
 -122      12         -64.     0.     94.
 -121      42        -182.     0.     65.
 -120      18        -108.     0.     76.
 -119      34         -86.     0.    120.
 -118      19         -76.     0.     94.
 -117      34        -134.     0.     80.
 -116      57         -73.     0.     97.
 -115     106         -88.     0.     88.
 -114      34         -57.     0.    164.
 -113      38        -142.     0.    102.
 -112      47        -154.     0.     89.
 -111      13         -99.     0.     95.
 -110      25         -48.     0.    229.
 -109      12         -72.     0.    227.

       Slick condition at the end of this time step

Fraction Evaporated = .24958
Amount Dissolved (gals)      : This Step = 2.4525     Total = 27.238
```

Figure 26   Spill information at t = 4 hrs

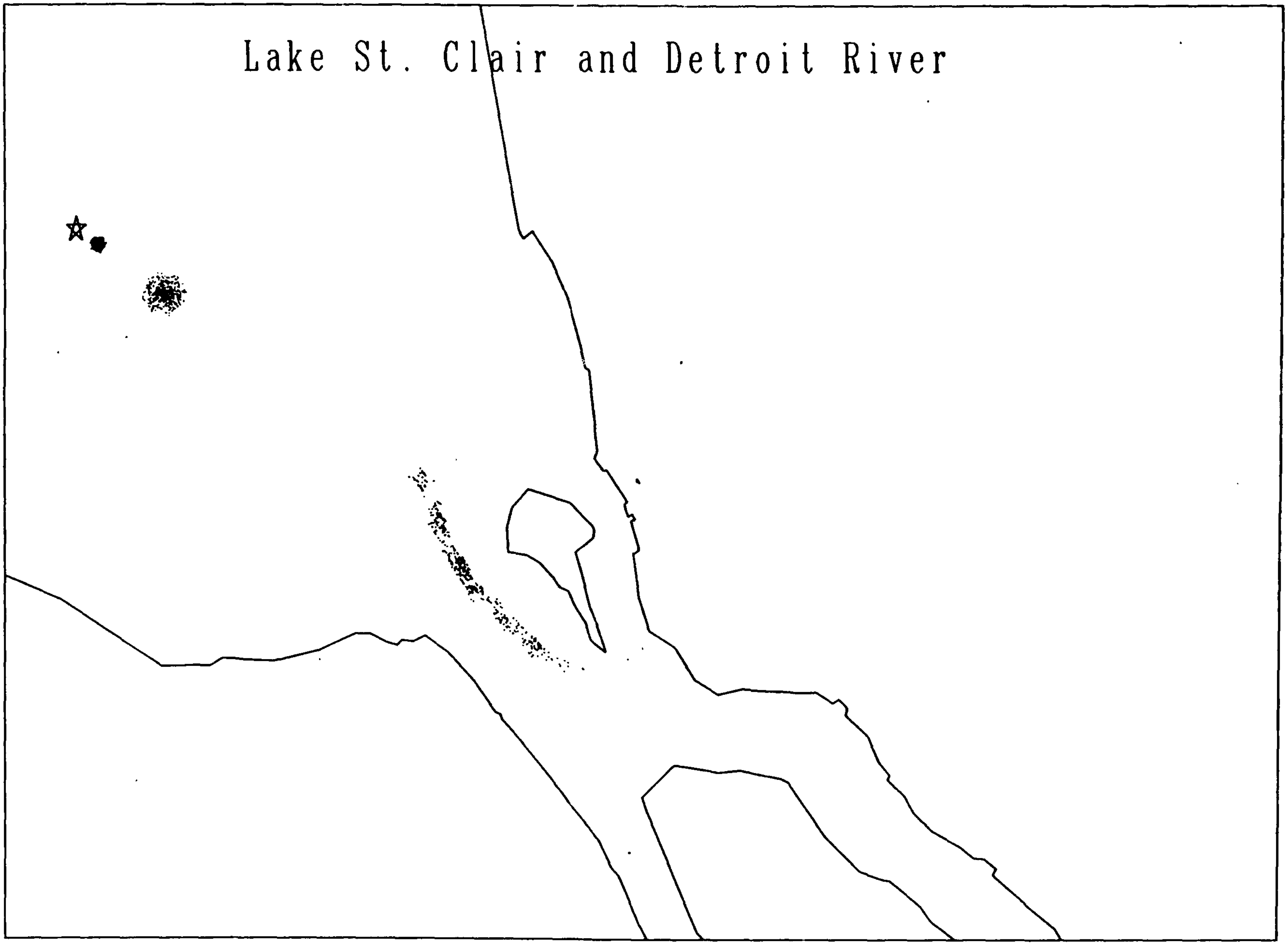


Figure 27 Plot of slick locations at

Lake St. Clair and Detroit River

```
**************************
*   CONTINUOUS SPILL     *
*          AT            *
*   -91000., 51900.      *
*    FOR   60. min.      *
**************************
```

SIMULATION PERIOD = 6.0 Hrs

Characteristics of spill

No. of particles	:	500
Oil spilled	:	10000. gals of Fuel Oil No. 2
DT for spill simulation	:	900. Secs.
Specific gravity of oil	:	0.84 (API index = 37.0)
Kinematic Visco. of Water	:	0.1411E-04 sq ft/sec
Surafce Tension	:	0.2060E-02 lbs/ft

Spreading Coefficients

K2i	K2v	K2t	c10	c20	c30
1.14	0.98	1.60	1.39	1.39	1.43

Molar volume	:	0.7063E-02 cu ft/mol
Solubility of fresh oil	:	0.1873E-02 lbs/cu ft
Viscosity of Oil	:	0.84 lbs/ft-sec
Manning s Roughness of Ice	:	0.035

Surface Diffusion

LAKE - Default formulation is used
RIVER- Default formulation is used

API option is not selected . Evap. constants are C = 7.88 TO = 465.0

Time step for river flow computation = 6.00Hrs

Figure 28 Description of spill type and location, oil properties and various coefficients

Open Water Conditions exist in the river

Flow and Discharge Conditions in the River

Branch	Q (cfs)	Stage (ft)
1	169740.	573.26
2	132690.	573.26
3	37050.	573.20
4	169740.	573.10
5	51850.	573.10
6	117890.	573.10
7	169750.	573.00
8	169750.	572.78
9	169760.	572.63
10	142660.	572.63
11	142660.	572.51
12	108140.	572.51
13	27100.	572.62
14	135240.	572.29
15	34520.	572.51
16	34530.	571.98

No. of Ice Covered Regions in the Lake = 1

Region	from X,Y Grid	to X,Y Grid	Ice Thic(ft)
1	15, 7	15, 20	0.10

Meteorological Station Data Used in Lake Circulation Model

Time hrs	Lat. deg	Long. deg	Height ft	T-air F	T-H20 F	Wind mph	Wind deg
0.0	42.42	82.42	30.0	40.0	32.0	4.0	0.0
3.0	42.42	82.42	30.0	40.0	32.0	4.0	0.0

Figure 29 Stage and Discharge at river branches, and meteorological data

```
--------------------------------------------------------------------------------
Time =   1.00 Hrs -- Wind :mag=  4.0 mph, dir =  0.0 deg   -- Air Temp= 40.0 F
Spill center after advection= -90336., 50086. (ft)
Volume per particle          =  19.145 gals

      Slick condition during this time step

Slick information by pie / strip
Strip  Particles  -Le(ft)  X-mean  Le(ft)
 56        23       -36.      0.    126.
 57        64       -78.      0.    132.
 58        67      -101.      0.     98.
 59        63       -95.      0.    100.
 60        63       -93.      0.     63.
 61        67       -76.      0.     70.
 62        64      -130.      0.      0.
 63        65       -50.      0.     41.
 64        24       -37.      0.     32.

       Slick condition at the end of this time step

Fraction Evaporated = .64919E-01
Amount Dissolved (gals)      : This Step = 1.1728     Total = 2.9714
```

Figure 30 Spill information at t = 1 hr

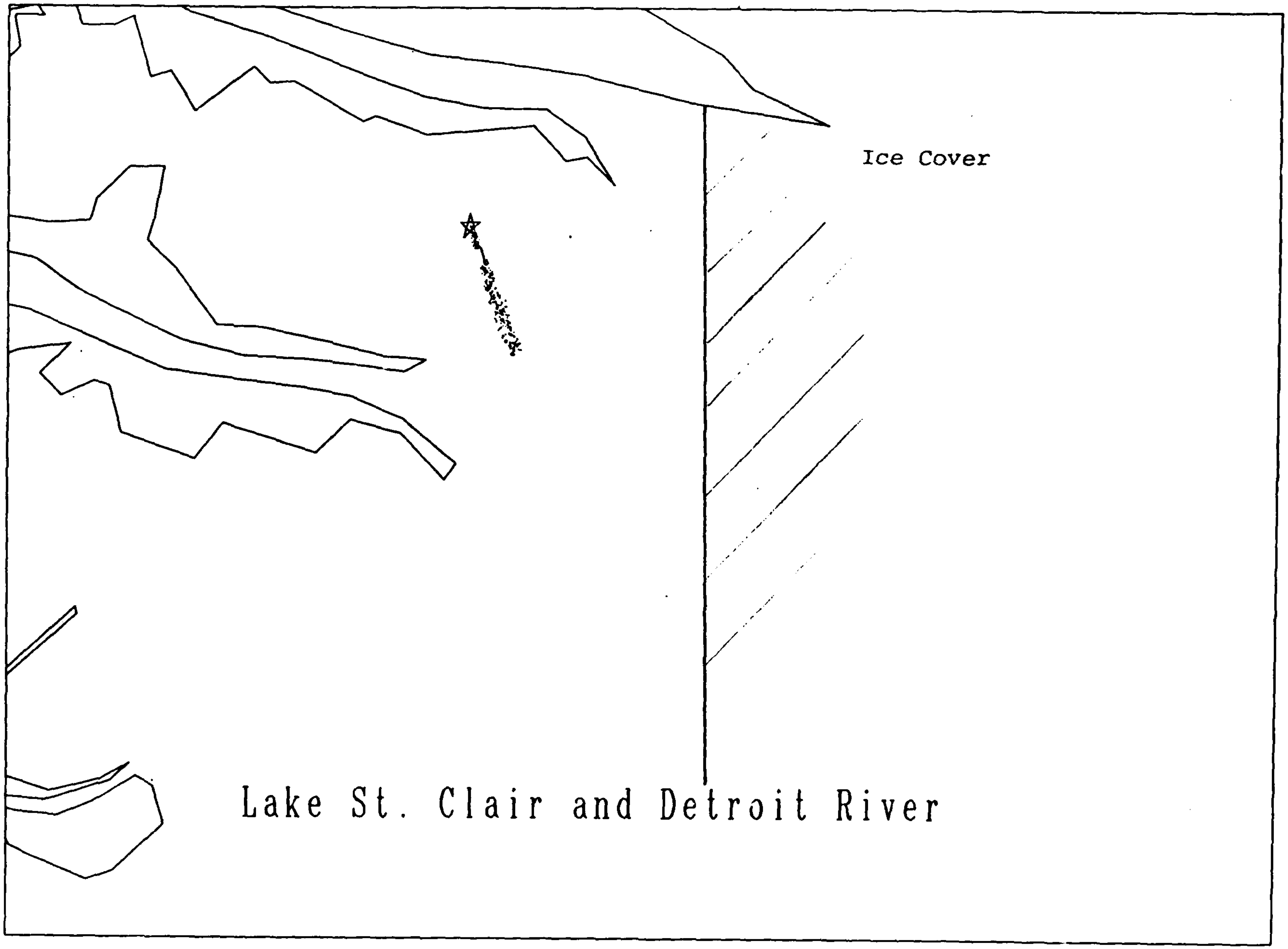


Figure 31 Plot of slick location corresponding to Figure 30. Figure ranges from -104.000 to -68,000 in the x-direction and 31.820 to 58.000 in the y-direction

```
-----------------------------------------------------------------------------
Time =   2.00 Hrs -- Wind :mag=  4.0 mph, dir =  0.0 deg   -- Air Temp= 40.0 F
Spill center after advection= -88782., 46501. (ft)
Volume per particle          =  17.378 gals

      Slick condition during this time step

Slick information by pie / strip
Strip  Particles  -Le(ft)  X-mean  Le(ft)
 27        66        -42.      0.    121.
 28        51        -80.      0.    116.
 29        65        -89.      0.    119.
 30        54       -176.      0.     52.
 31        68       -101.      0.    116.
 32        56       -103.      0.    101.
 33        68        -84.      0.     95.
 34        66        -99.      0.     82.
 35         5       -101.      0.     43.

       Slick condition at the end of this time step

Fraction Evaporated = .14706
Amount Dissolved (gals)      : This Step = 1.5688     Total = 9.2324
```

Figure 32   Spill information at t = 2 hrs

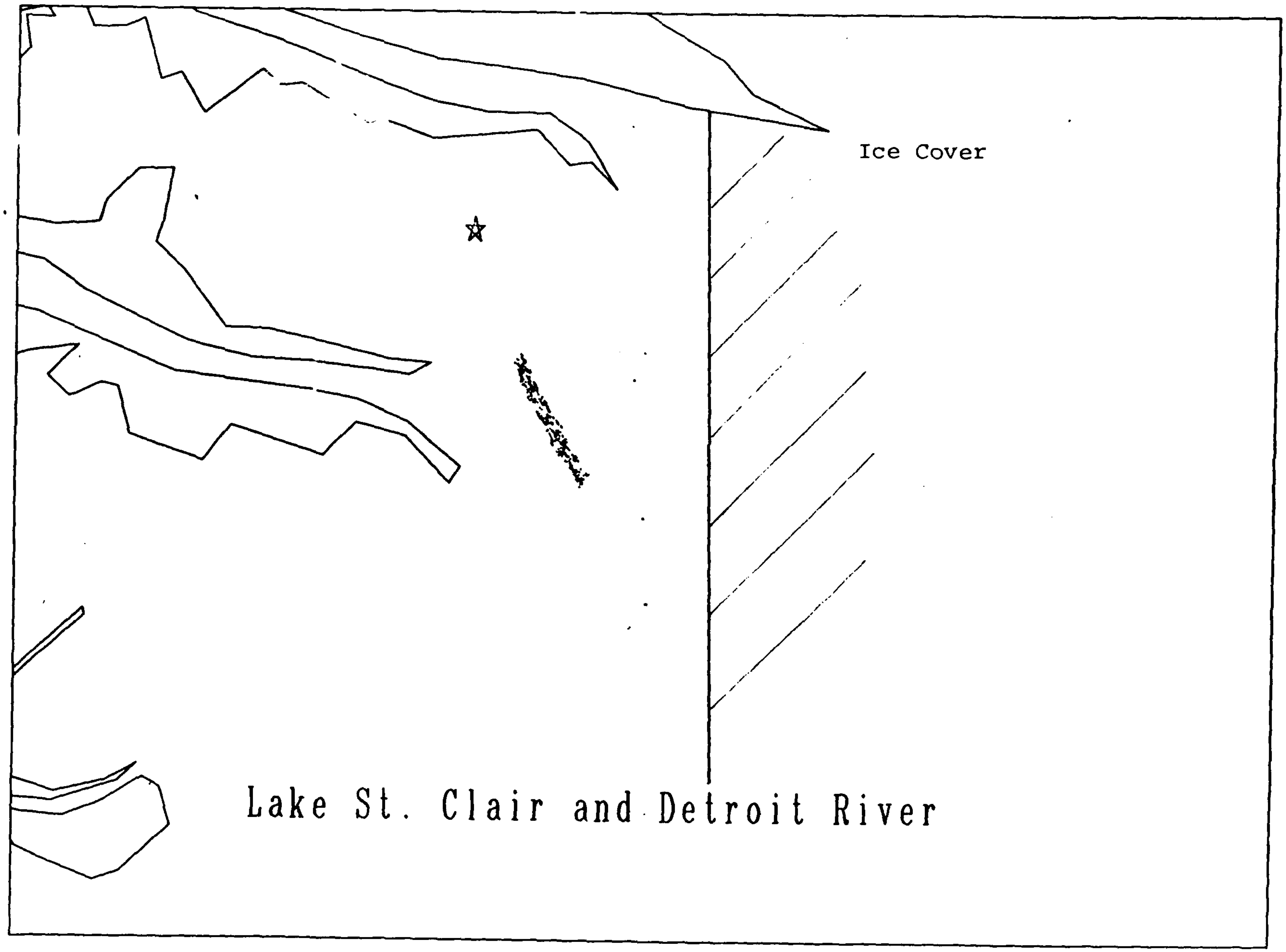


Figure 33 Plot of slick location corresponding to Figure 32
Figure ranges from -104,000 to -68,000 in x-direction
and 31,820 to 58,000 in the y-direction

```
-------------------------------------------------------------------------------
Time =   3.00 Hrs -- Wind :mag=  4.0 mph, dir =  0.0 deg   -- Air Temp= 40.0 F
Spill center after advection= -86307., 43639. (ft)
Volume per particle          =  16.185 gals

     Slick condition during this time step

Slick information by pie / strip
Strip  Particles  -Le(ft)  Y-mean  Le(ft)
-47        21       -123.     0.     80.
-46        72       -124.     0.     73.
-45        77        -97.     0.     93.
-44        90       -117.     0.     99.
-43        70       -128.     0.     92.
-42        75       -118.     0.    196.
-41        58       -113.     0.    202.
-40        37        -66.     0.     96.

     Slick condition at the end of this time step

Fraction Evaporated = .20031
Amount Dissolved (gals)      : This Step = 1.4632     Total = 15.505
```

Figure 34   Spill information at t = 3 hrs

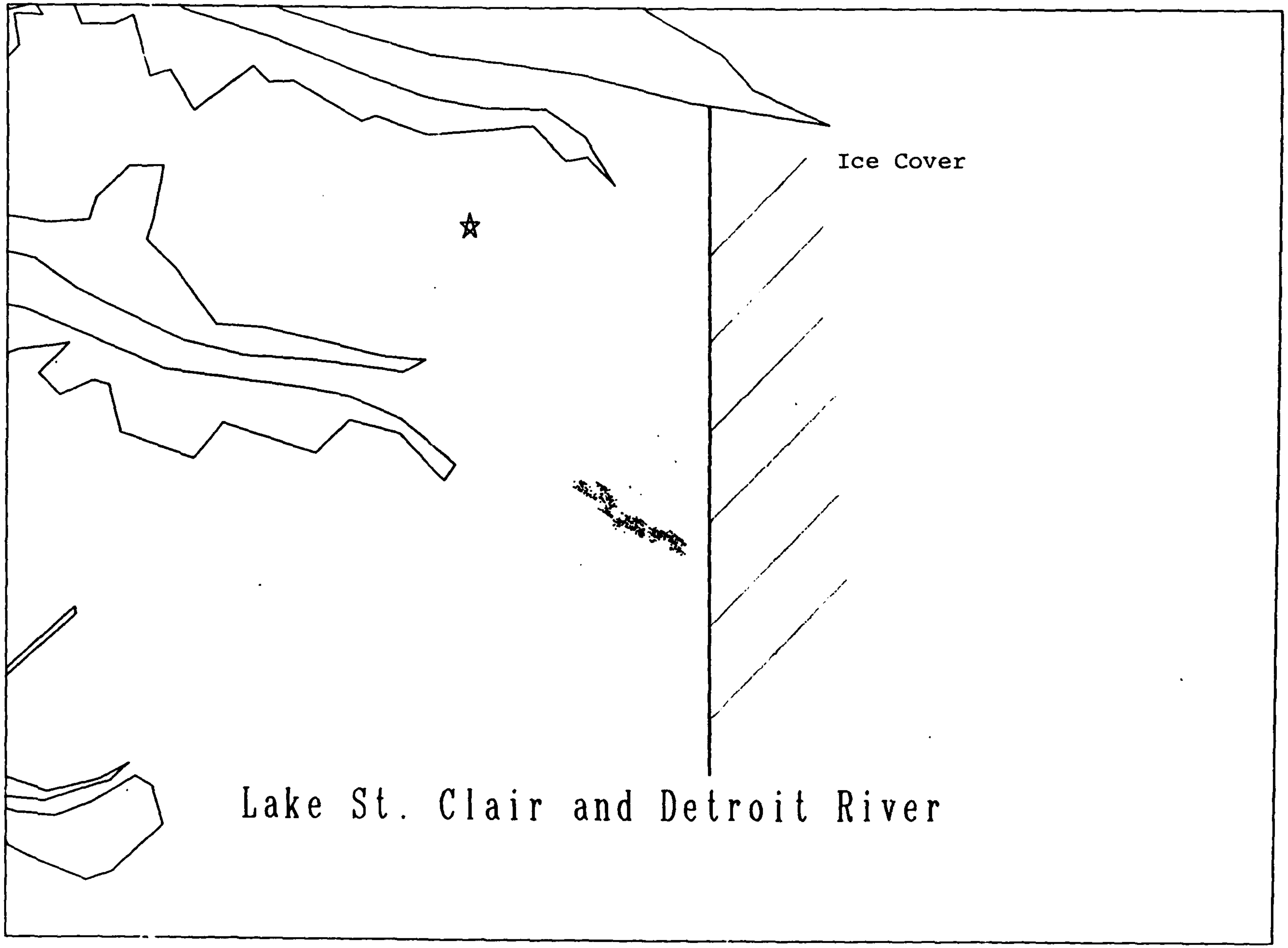


Figure 35 Plot of slick location corresponding to Figure 34. Figure ranges from -104,000 to -68,000 in x-direction and 31,820 to 58,000 in the y-direction.

```
Time =   4.00 Hrs -- Wind :mag=  4.0 mph, dir =  0.0 deg   -- Air Temp= 40.0 F
Spill center after advection= -83927., 42445. (ft)
Volume per particle         =  15.510 gals

     Slick condition during this time step

Slick information by pie / strip
Pie  No. of particles  Mean radius(ft)
 1          42              278.
 2          16              176.
 3          35              395.
 4         102              578.
 5          13              237.
 6          24              161.
 7          59              286.
 8         179              437.


      Slick condition at the end of this time step

Fraction Evaporated = .22655
Amount Dissolved (gals)       : This Step = .88386     Total = 20.118
```

Figure 36   Spill information at t = 4 hrs

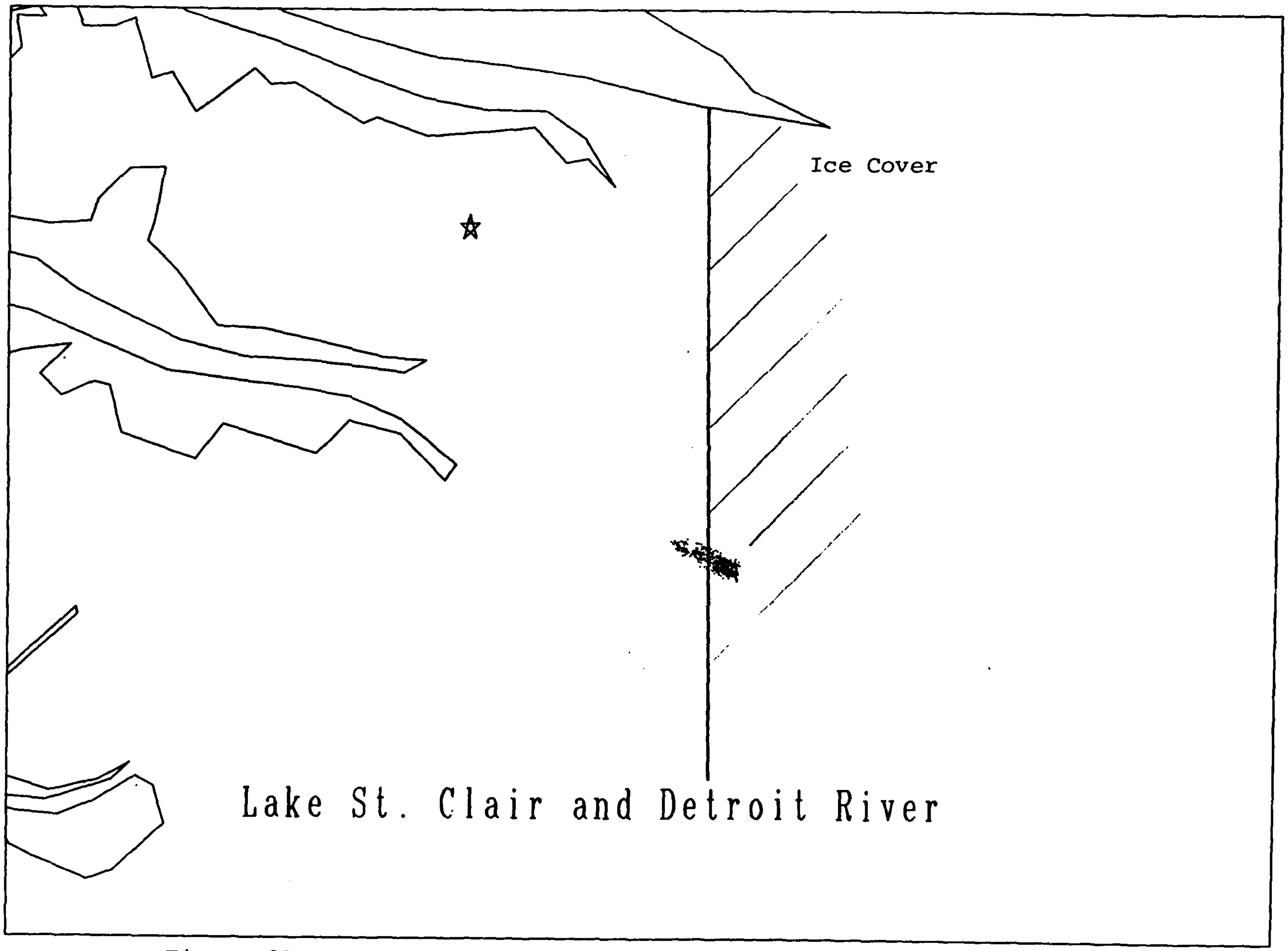


Figure 37 Plot of slick location corresponding to Figure 36. Figure ranges from -104,000 to -68,000 in x-direction and 31,820 to 58,000 in the y-direction

----

Time =   5.00 Hrs -- Wind :mag= 4.0 mph, dir = 0.0 deg   -- Air Temp= 40.0 F
Spill center after advection= -83634., 42312. (ft)
Volume per particle   = 15.407 gals

Slick condition during this time step

Slick information by pie / strip

Pie	No. of particles	Mean radius(ft)
1	61	225.
2	39	167.
3	63	226.
4	92	253.
5	58	208.
6	30	153.
7	78	235.
8	75	249.

Slick condition at the end of this time step

Fraction Evaporated = .22757
Amount Dissolved (gals)   : This Step = .19960   Total = 21.146

Figure 38   Spill information at t = 5 hrs

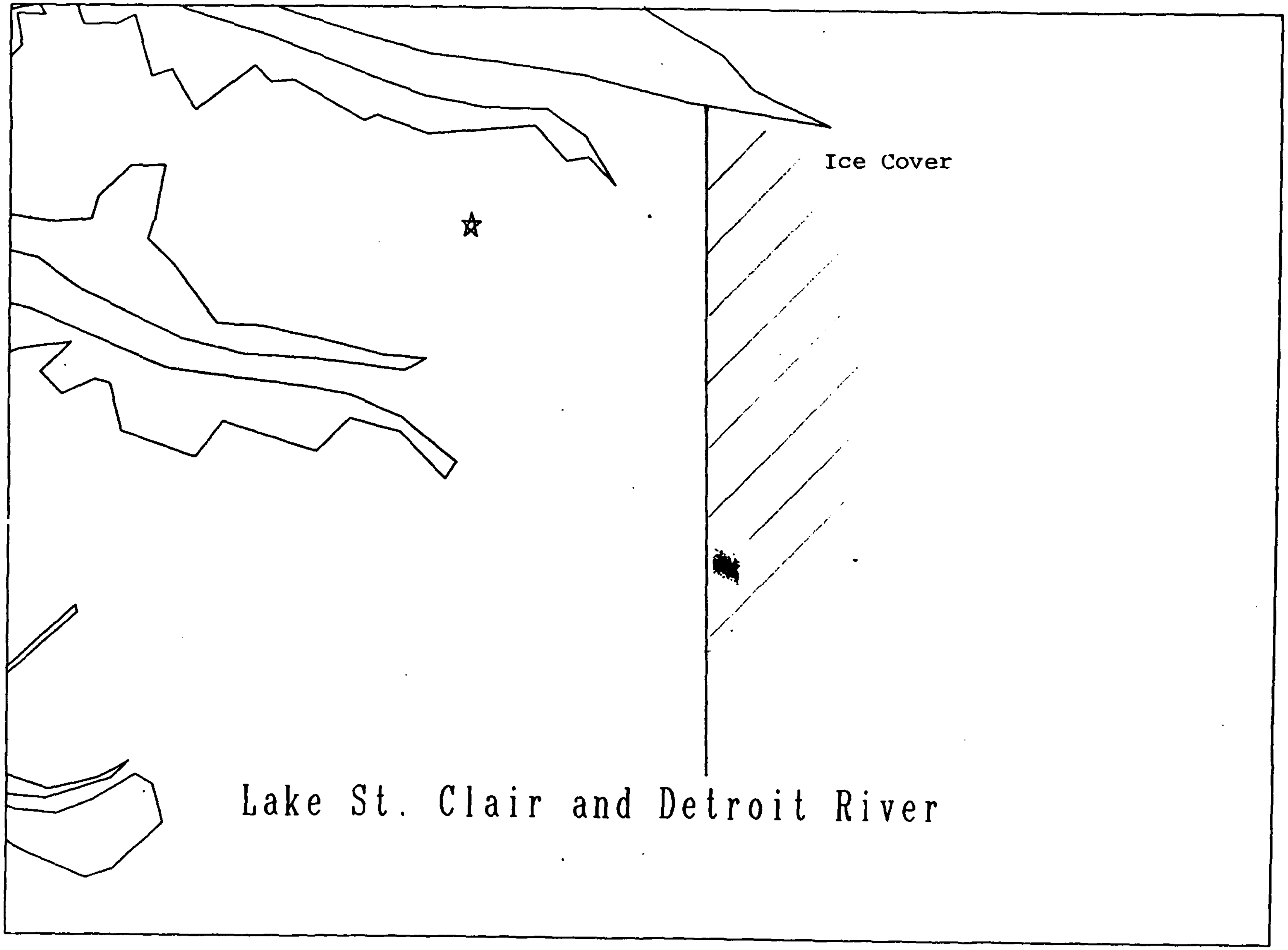


Figure 39 Plot of slick location corresponding to Figure 38. Figure ranges from -104,000 to -68,000 in x-direction and 31,820 to 58,000 in the y-direction

model. Rather, they are indirectly generated from the shoreline data in file LDETR.SHO and the particle locations in file LDETRSP.OUT. The output should be self-explanatory with the aid of the figure captions.

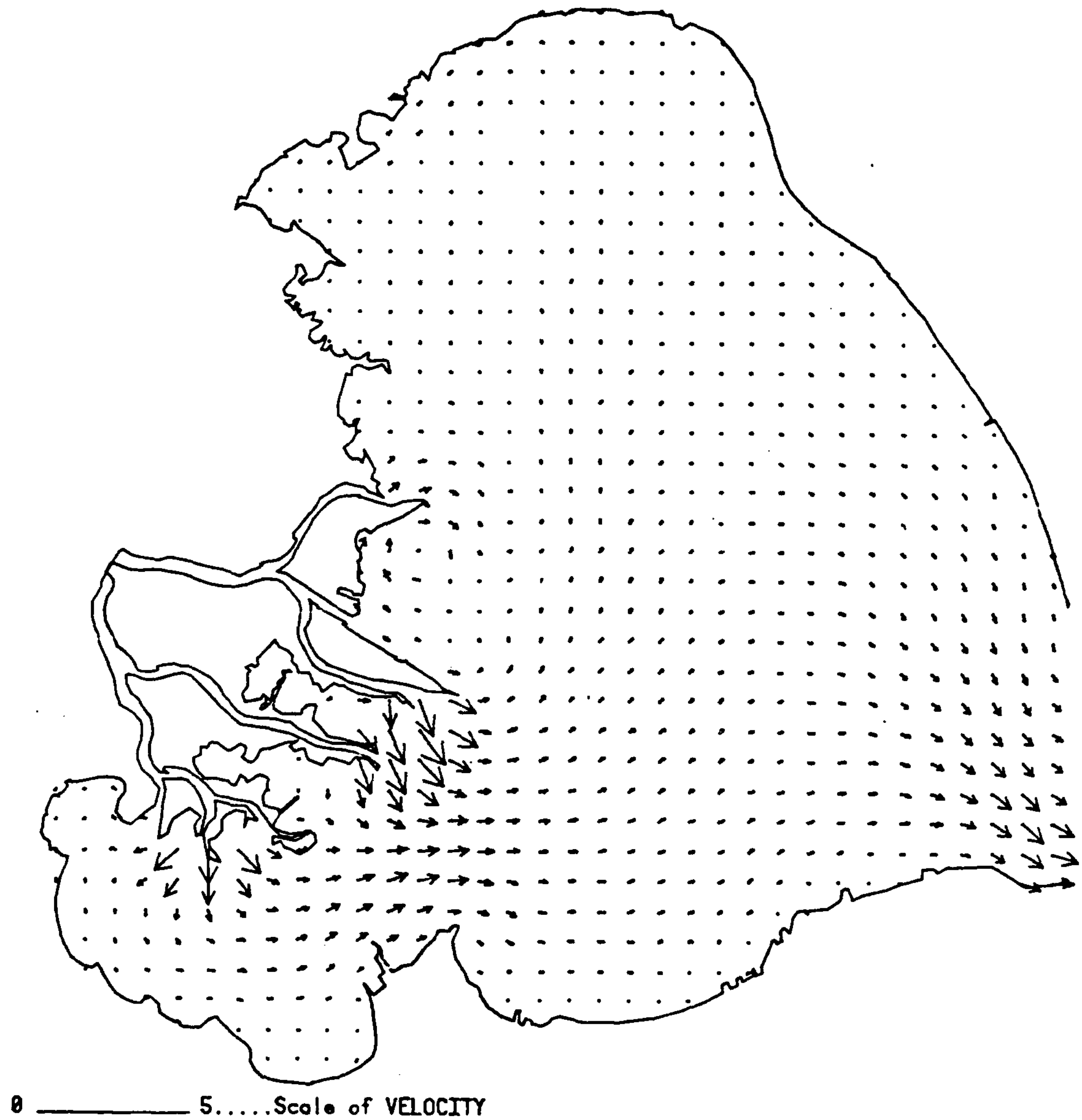


Figure 40 Velocity distribution in Lake St. Clair corresponding to stage/discharge in Figure 23.

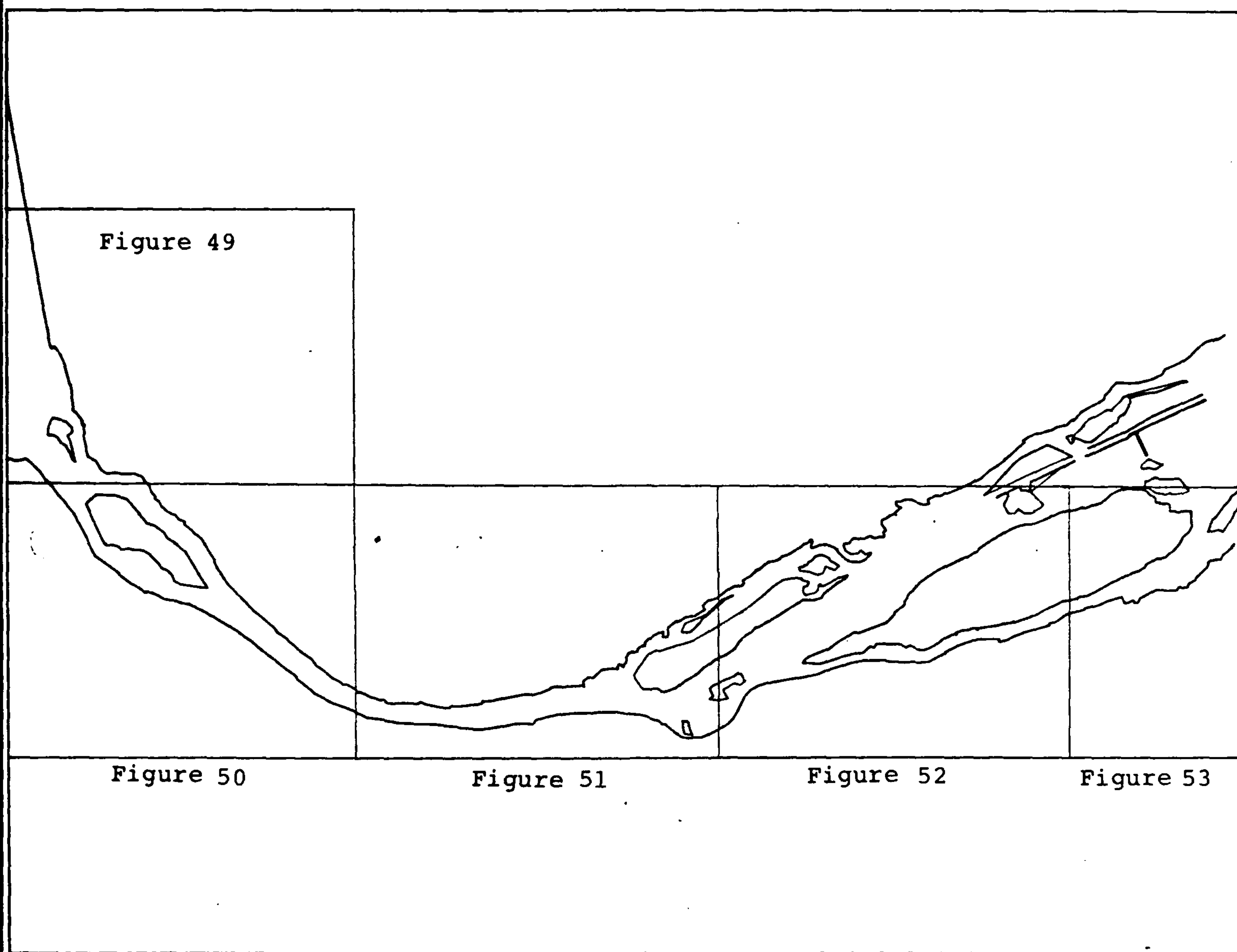


Figure 41 Index to Figures 42 through 46

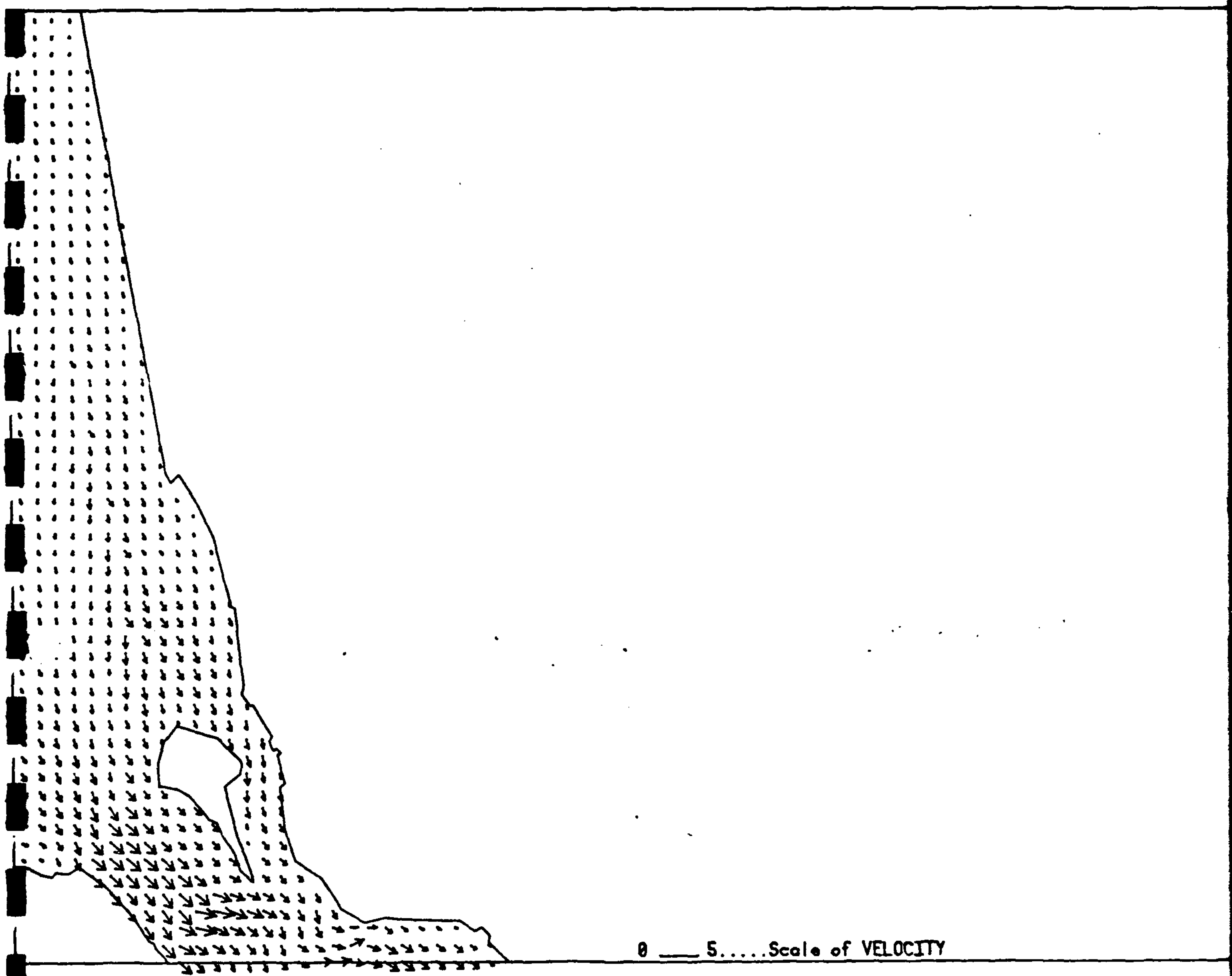


Figure 42 Velocity distribution in Detroit River corresponding to stage/discharge in Figure 23

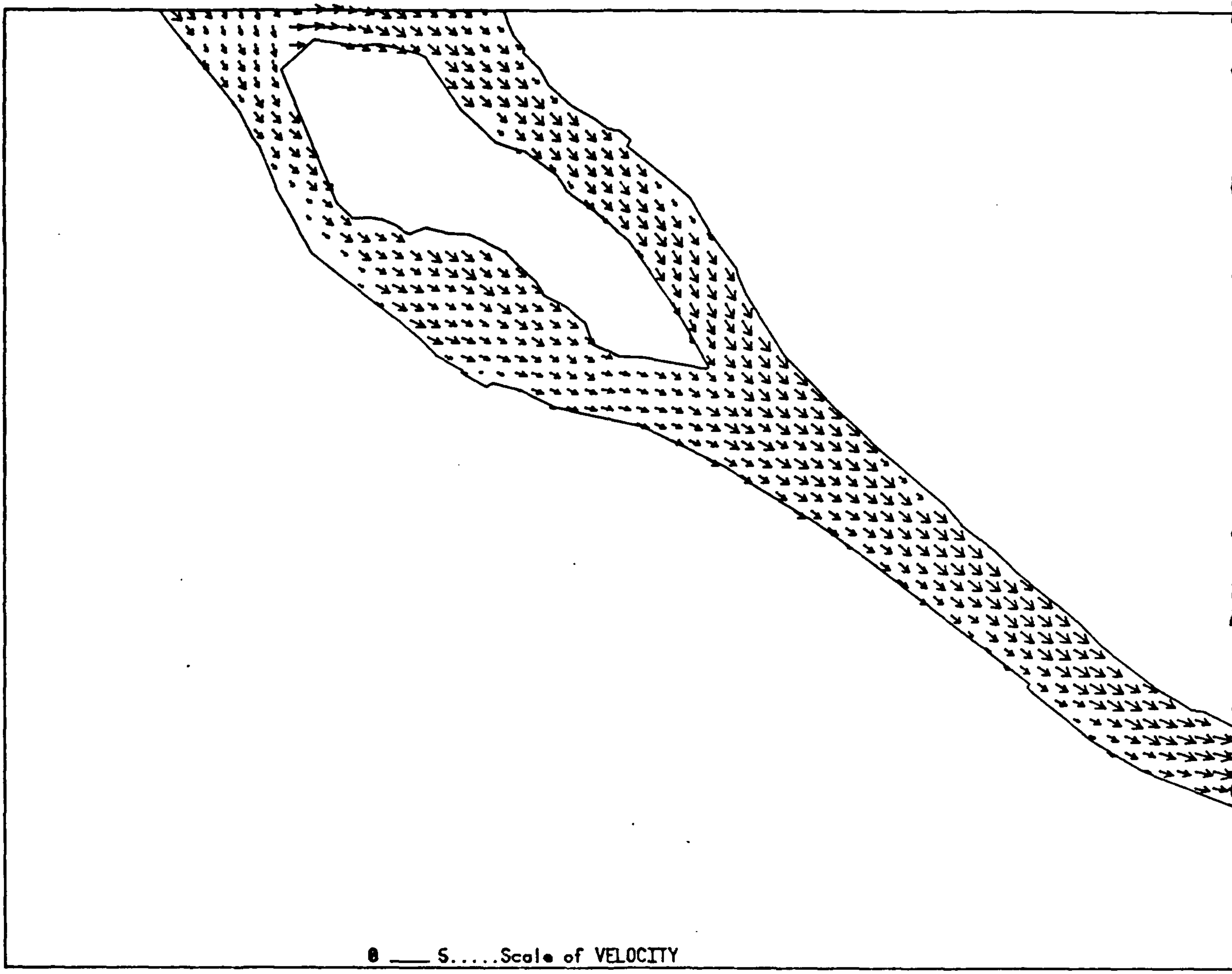


Figure 43 Velocity distribution in Detroit River corresponding to stage/discharge in Figure 23

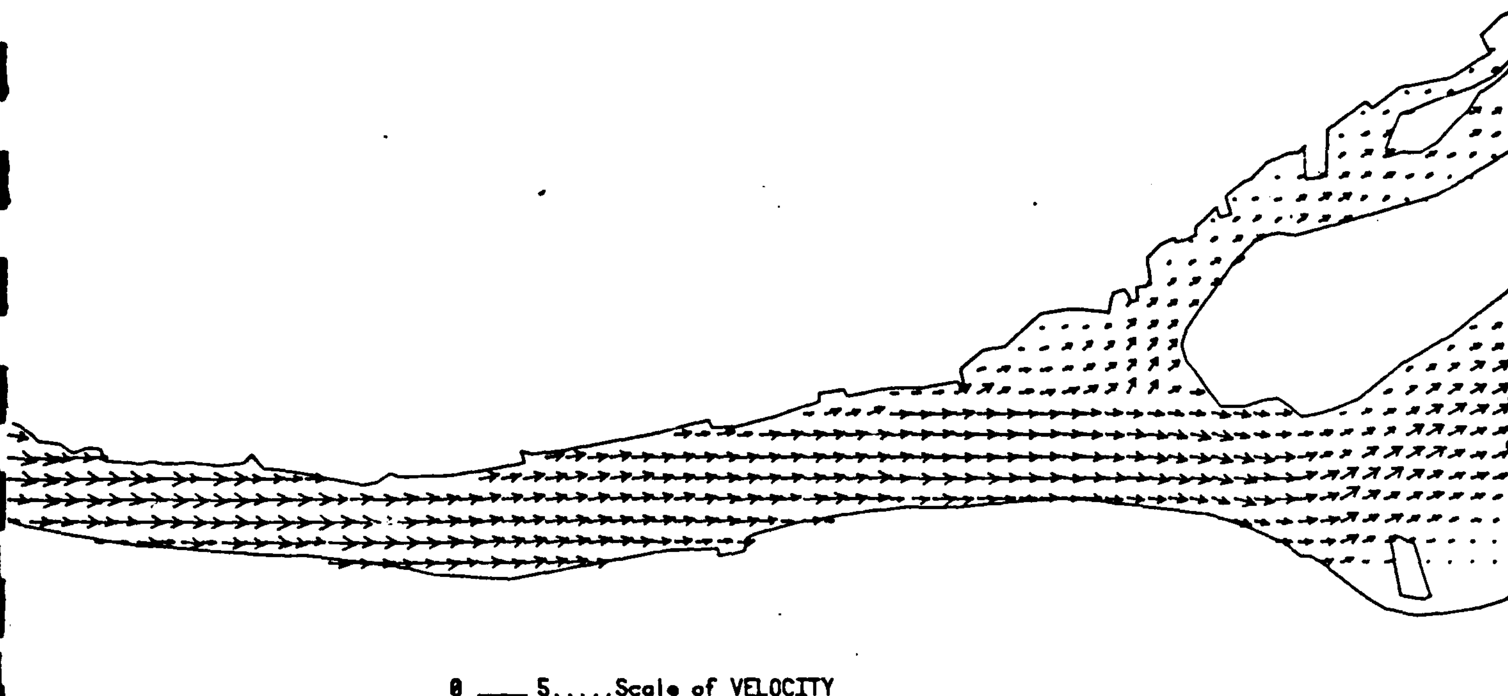


Figure 44 Velocity distribution in Detroit River corresponding to stage/discharge in Figure 23

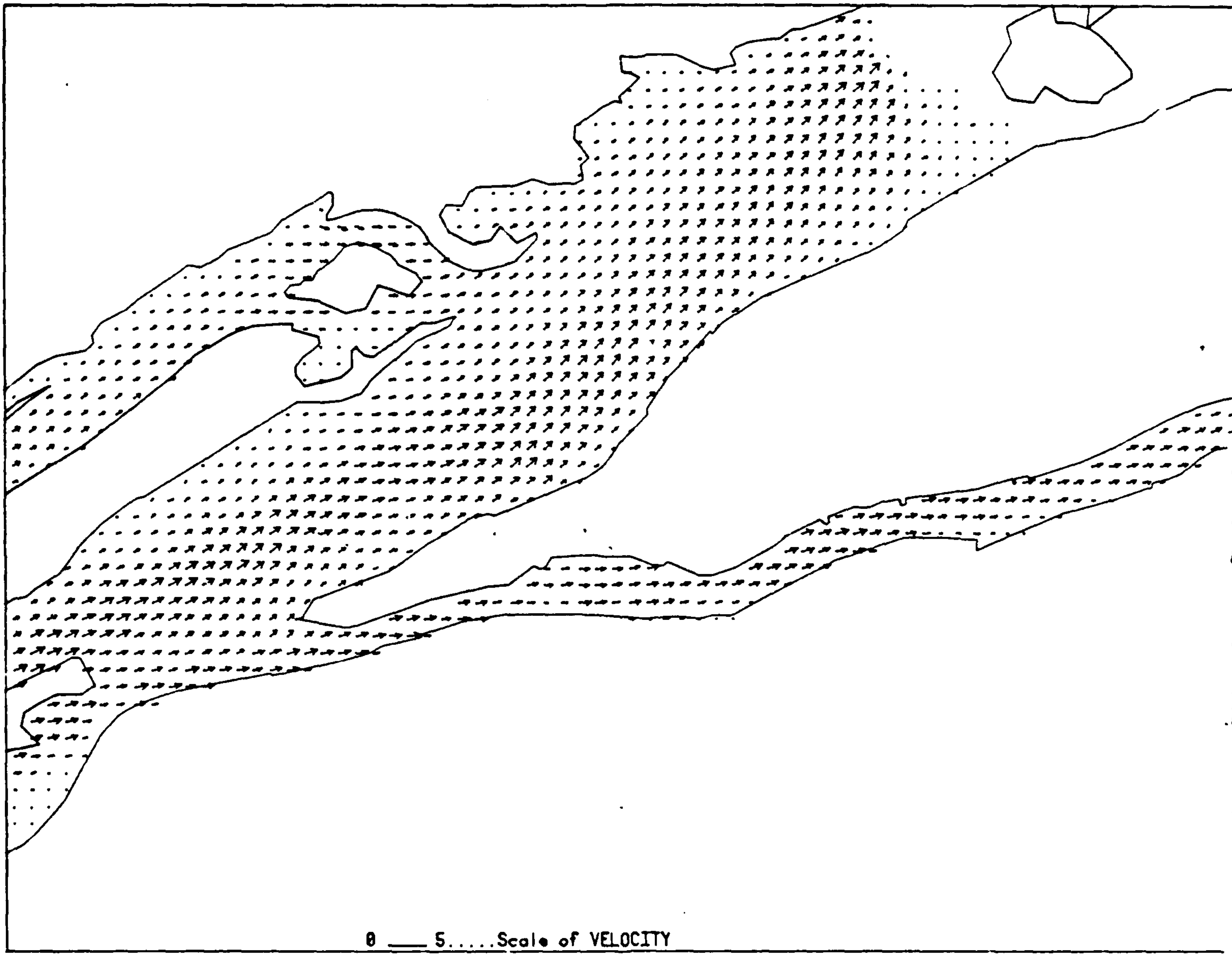


Figure 45 Velocity distribution in Detroit River corresponding to stage/discharge in Figure 23

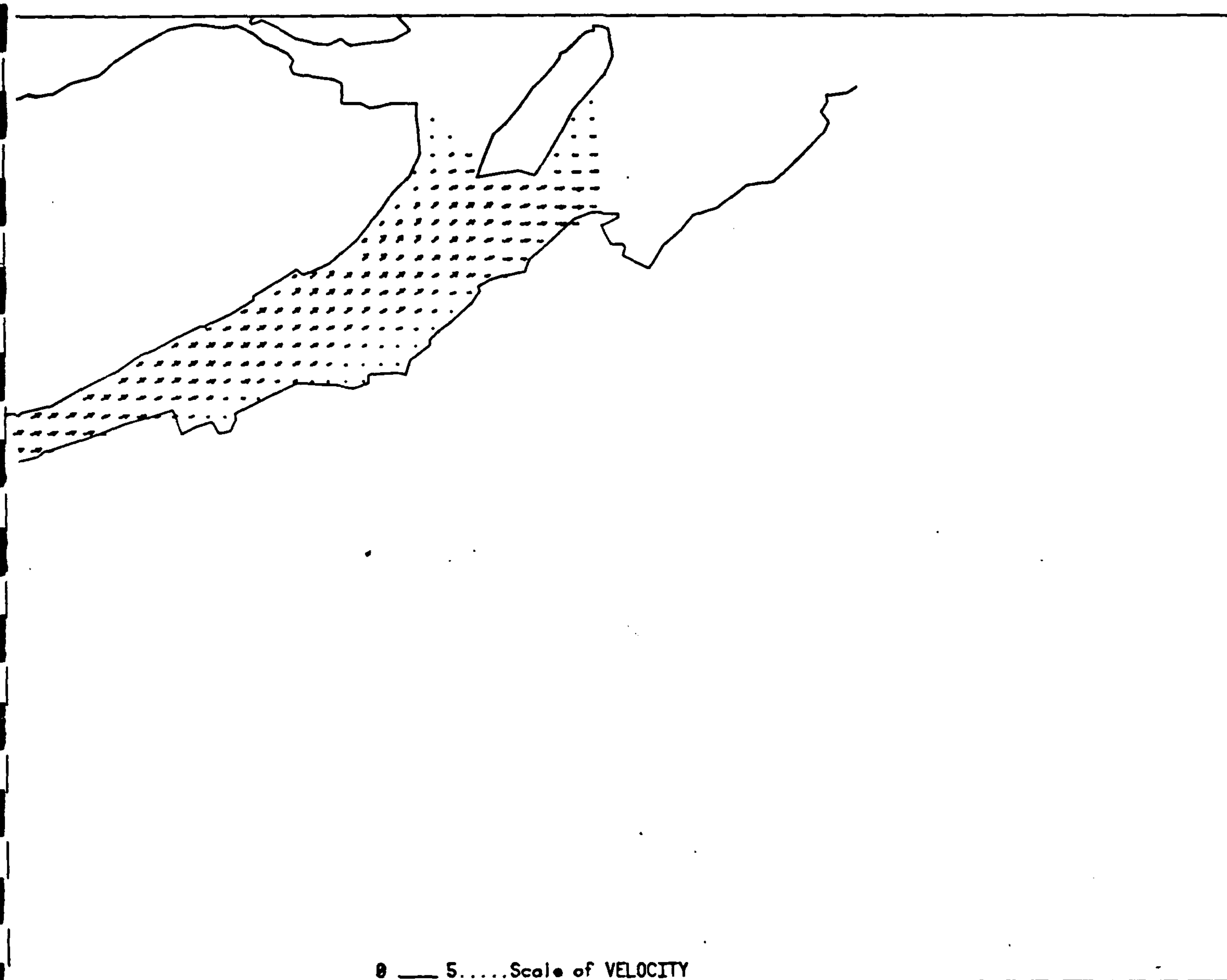


Figure 46 Velocity distribution in Detroit River corresponding to stage/discharge in Figure 23

APPENDIX I

DETROIT RIVER CROSS SECTIONS

This appendix contains the detailed geometrical data for Detroit River cross sections used in the model LROSS.

Lake St. Clair and Detroit River

X-Sect. No.		Vertical Line No. 2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	Distance(ft)	1375	1675	3000	3575	4000	4250	5000	8500	9625	10875	13000	13250	15500	21000	26000	29000	37000		
	Depth (ft)	6	24	21	16	23	18	10	9	10	8	8	6	4	6	4	3	0		
2	Distance(ft)	125	200	1500	2000	2500	2550	3125	4375	6000	6450	6625	7000	7750	8325	10500	11783			
	Depth (ft)	12	21	27	28	12	9	6	2	6	12	19	25	7	6	2	0			
3	Distance(ft)	25	125	500	1000	1200	1750	2000	2250	3000	3425									
	Depth (ft)	14	28	36	37	33	32	30	10	9	0									
4	Distance(ft)	155	900	1025	1275	1400	1700	2125	2375	3250	3375									
	Depth (ft)	25	28	36	40	36	43	30	10	5	0									
5	Distance(ft)	50	125	750	1250	1350	2425	2525	3250	3425										
	Depth (ft)	12	20	21	22	28	28	5	4	0										
6	Distance(ft)	75	175	250	825	1400														
	Depth (ft)	12	22	27	27	0														
7	Distance(ft)	75	175	250	825	1250														
	Depth (ft)	12	22	27	27	0														
8	Distance(ft)	81	142	198	373	627	760	1124	1184	1233	1322									
	Depth (ft)	11	26	29	30	31	30	24	21	10	0									
9	Distance(ft)	125	250	500	800	1000	1250	2050	2850	3500	4000	4250	4650	4825	5000	5100				
	Depth (ft)	12	25	21	20	28	23	23	38	38	29	43	45	7	7	0				
10	Distance(ft)	75	275	575	1250	2075	2825	3825	4450	4600	4675	5075								
	Depth (ft)	4	18	24	18	12	27	29	39	18	2	0								
11	Distance(ft)	101	266	507	756	1223	1360	1461												
	Depth (ft)	25	25	29	33	31	26	0												
12	Distance(ft)	25	75	475	575	800	1250	1875	2050	2550	2750	3400								
	Depth (ft)	12	25	23	12	6	2	5	22	27	5	0								
13	Distance(ft)	88	159	1107	1093	1998	2074	2111												
	Depth (ft)	22	27	21	26	23	12	0												
14	Distance(ft)	2	135	550	695	883	1000	1374	1692	1844	1923									
	Depth (ft)	10	25	31	24	29	20	22	26	21	0									

Distance is measured from Lower Left Bank.

15	Distance(ft)	24	607	1005	1208	1388	1598	1684	1862	1891	2194	2204				
	Depth (ft)	20	22	29	10	33	27	20	13	7	6	0				
16	Distance(ft)	125	250	350	680	825	1250	1525	1625	1825	2125	2533	2550			
	Depth (ft)	26	35	33	48	42	46	35	35	9	8	9	0			
17	Distance(ft)	109	317	529	755	1004	1660	1831	1921							
	Depth (ft)	31	39	35	40	37	42	24	0							
18	Distance(ft)	30	63	250	475	563	1125	1375	1625	1750	1825	1950	2000			
	Depth (ft)	5	14	15	42	32	33	41	43	35	38	23	0			
19	Distance(ft)	125	375	425	1000	1250	1375	1500	1725	1850	2000	2375	2750	3125	3500	3750
	Depth (ft)	12	18	25	35	37	37	9	9	25	37	29	28	32	35	0
20	Distance(ft)	10	550	625	1375	1750	2125	2310	2625	2910						
	Depth (ft)	27	28	37	36	36	48	43	38	0						
21	Distance(ft)	118	369	634	811	1338	1657	2224	2373							
	Depth (ft)	26	22	32	51	39	48	41	0							
22	Distance(ft)	25	125	200	475	600	750	1000	1250	1600	1750	2000	2125	2250		
	Depth (ft)	23	28	24	34	30	45	50	43	47	36	45	42	0		
23	Distance(ft)	125	175	375	1375	1500	1625	1800	1875	2050						
	Depth (ft)	18	29	41	45	47	32	18	6	0						
24	Distance(ft)	197	451	717	1187	1594	1674	1903								
	Depth (ft)	38	46	47	40	43	39	0								
25	Distance(ft)	25	125	1000	1500	2250	2500	2550								
	Depth (ft)	28	38	38	31	32	27	0								
26	Distance(ft)	25	510	1120	2075	2375	2600	2710	2825							
	Depth (ft)	9	45	38	35	38	34	8	0							
27	Distance(ft)	25	125	300	1000	1500	1750	2375	2625	2650						
	Depth (ft)	19	25	40	39	32	36	36	24	0						
28	Distance(ft)	250	680	1125	1485	1750	2090	2500	2850	3320	3750	4250				
	Depth (ft)	38	43	36	35	41	37	38	10	5	8	0				
29	Distance(ft)	175	1050	2175	2500	2675	3300	3425	4425	4675						
	Depth (ft)	34	38	28	18	25	27	4	3	0						
30	Distance(ft)	185	375	500	625	850	1150	1500	1650	2000	2125	2310	2450	2510		
	Depth (ft)	29	40	41	40	36	37	36	38	30	9	5	7	0		

31	Distance(ft)	27	231	450	836	1548	1842	2331	2587	2865	3354	3400								
	Depth (ft)	3	6	20	30	34	46	31	34	8	7	0								
32	Distance(ft)	117	223	804	997	1503	1829	1976	2551	3197	3392	4179	4208	4277	4777					
	Depth (ft)	35	37	30	11	6	10	31	39	35	17	5	9	8	0					
33	Distance(ft)	33	115	327	593	864	1149	1199	1254	1300	1337	2001	2390	2443	2984	3235	4076	4409	6204	6715
	Depth (ft)	18	35	34	36	9	6	10	0	0	8	33	7	6	14	38	38	9	6	6
34	Distance(ft)	300	500	780	1310	1790	2090	2900	3250	3680	5600	5650	5750							
	Depth (ft)	31	31	5	5	35	37	35	36	7	7	9	0							
35	Distance(ft)	450	575	1250	1750	2700	2725	2825	4500	5000	5025									
	Depth (ft)	6	28	36	33	28	18	5	3	2	0									
36	Distance(ft)	120	174	461	851	986	1040	1106	1175											
	Depth (ft)	9	20	35	32	22	13	5	0											
37	Distance(ft)	24	573	722	896	976														
	Depth (ft)	14	33	31	16	0														
38	Distance(ft)	199	502	854	952	1887	1980	2265	2428											
	Depth (ft)	24	26	24	9	7	26	21	0											
39	Distance(ft)	92	301	446	681	969	1052	1707	2243	2375										
	Depth (ft)	25	31	28	35	22	8	7	4	0										
40	Distance(ft)	1500	1625	2000	2100	2300	3750	3825	3900	4225	4375	4675								
	Depth (ft)	0	25	25	2	0	0	5	32	32	2	0								
41	Distance(ft)	25	625	750	1125	1175	1250	1750	1925	2125	2150									
	Depth (ft)	2	1	27	27	22	30	32	6	2	0									
42	Distance(ft)	50	475	1000	2000	2100	2800	3100	3300	3750	4250	4500	5100	5500	5900	6900	7000	7100		
	Depth (ft)	3	3	27	27	35	35	25	32	30	33	28	28	9	8	8	12	0		
43	Distance(ft)	73	293	2564	2710	3746	3772	4424	4509	4996	5151	5328	5482	5796	6111	6346				
	Depth (ft)	3	6	11	27	24	29	36	27	25	22	28	28	11	13	0				
44	Distance(ft)	575	750	1000	2750	3175	3375	4300	4375	4950	5400	5900	6050							
	Depth (ft)	6	6	5	3	6	18	18	27	27	18	6	0							
45	Distance(ft)	17	129	284	474	657	812	867	898											
	Depth (ft)	5	31	36	37	30	9	7	0											
46	Distance(ft)	4	45	244	536	779	993	1152	1850	1940	2003									
	Depth (ft)	3	4	32	33	31	7	4	3	2	0									

47	Distance(ft)	55	278	434	460	502	773	823	848	978	1095	1246							
	Depth (ft)	19	18	23	15	33	35	21	25	11	8	0							
48	Distance(ft)	16	70	240	440	591	876	1238	1377	1496	1523								
	Depth (ft)	2	3	22	10	10	27	24	8	5	0								
49	Distance(ft)	48	100	636	756	838	982	1144	1165	1379	1425								
	Depth (ft)	4	24	16	20	18	30	27	20	15	0								
50	Distance(ft)	31	209	990	1146														
	Depth (ft)	25	29	27	0														
51	Distance(ft)	226	275	766	1246	1345	1806	1922	2396	2472	2639								
	Depth (ft)	8	7	8	22	19	16	20	16	10	0								
52	Distance(ft)	320	406	559	882	1033	1508	1676	2128	2194	2468	2542	2646	2709	2771	3015	3209	3432	3675
	Depth (ft)	20	8	9	23	8	5	22	15	9	10	17	14	8	7	2	6	7	0
53	Distance(ft)	875	1000	2000	2175	2550	4125	4625	4875	5125	5500	6500							
	Depth (ft)	4	18	18	5	0	0	2	14	14	4	0							

APPENDIX II

PROGRAM LISTING

The program listing for LROSS and subprograms are presented in this appendix. The program listing is arranged in the following sequence.

<u>Main Program</u>

LROSS[1]

<u>LROSS Subroutines</u>[1]	<u>GLERL Subroutines</u>[3]
ADVECT	INIT
BOUNDR	OUTP
DISOLU	PARTIC
EVAPOR	PGPARM
NDCONV	PRNT
ORIENT	RGRID
PLOTNU	RLID
PRELSE	UPDATE
PRINT1	UZL
SPRDAX	
SPRD1D	<u>GLERL Function Subprograms</u>[3]
VELDIS	RLAT
	RAU
<u>Systems Subroutines</u>[2]	UZ
GAUSS	XDIST
RANDU	YDIST

---

1 - These programs were developed for the oilspill model at Clarkson University.

2 - These programs were available at Clarkson University computing system. The source code is provided here for completeness. Other systems may substitute these with appropriate subprograms.

3 - These subroutines/functions were originally developed at GLERL, Ann Arbor, MI. They were slightly modified to match with needs of the oilspill programs.

Main program LROSS

```
C
C     **********************************************************************
C     *
C     *     Lake & River Oil Slick Simulation Model          ..L-ROSS..
C     *
C     *     Last Date of Revision October 14, 1986
C     *
C     *    Developed by the Department of Civil and Environmental Engineering
C     *              Clarkson University, Potsdam, New York 13676
C     *     under the support of the Detroit District, U. S. Army Corps of
C     *     Engineers, through the Cold Regions Research and Engineering
C     *     Laboratory, Hanover, N.H.
C     *-----------------------------------------------------------------------
C     *  This program is furnished by the Government and is accepted and used
C     *  by the recipient upon the express understanding that
C     *  the United States Government           makes no warranties, express
C     *  or implied, concerning the accuracy, completeness, reliability,
C     *  usability, or suitability for any particular purpose of the
C     *  information and data contained in this program or furnished in
C     *  connection therewith, and the United States Government shall be under
C     *  no liability whatsoever to any person by reason of any use made
C     *  thereof.  The program herein belongs to the Government.  Therefore,
C     *  the recipient further agrees not to assert any proprietary rights
C     *  therein or to represent this program to anyone as other than a
C     *  Government program.
C     *-----------------------------------------------------------------------
C     *
C     **********************************************************************
C
      DIMENSION IPARTX(5),IPARTY(5),HLIFE(10),IDUM(20)
      COMPLEX VSTRM(99,16),CORDV(99,16),VCAR(8000),CORDLB(99)
      COMPLEX SPCEN,PARTCL(1000),VWIND,VDRIFT
      COMPLEX SPCENO
      COMMON /VEL/VSTRM,CORDV,CORDLB,Q(30),WL(30),TICE(99,20),
     $ YWID(99,20),Z(99,20),ZD(99),NSLSCT(99),SCTANG(99),
     $ LCSTSQ(30),NSTUBE(99),NUMCON(99),NFIRCO(99),NSECO(99),KINTM
      COMMON /VA/ VCAR,VWIND,VDRIFT
      COMMON /VASB/IGRILB(300),IGRIUB(300),IGRLB1(300),IGRUB1(300)
      COMMON /ASB/SPCEN,PARTCL,NPTCL,NHITB,IHITB(1000),TYPBND(4,300)
      COMMON /BLOCK7/SPGOIL,ANIU,SIGMA,AK2I,AK2V,AK2T,
     $ VOLPAR,VOLPIE(8),SLICKR(8)
      COMMON /BLOCK8/AKC10,AKC20,AKC30
      COMMON /SE/FEVP1,FEVP2,CEVP,T0EVP
      COMMON /V/IZRBX(100),IZRBY(100),NZRVB
      COMMON /ICE/ZWND(40,40),ZLKICE(20),NICEX1(20),NICEY1(20),
     $ NICEX2(20),NICEY2(20),IPOS1(20),IPOS2(20),AMIUO,ANICE,
     $ SPAICE,NICERG,LICERG
      COMMON /SO/ IMOVIN(1000),YSHIFT(1000),NMOVIN,SSHIFT
      COMMON /GRIDS/ DXL, DXR, LGRIDX, NGRIDX, IRGRID, BEGLK
      COMMON /LKMD/ D(40,40), S(40,40), U(40,40), V(40,40)
      INTEGER UFSTPS,OSTPS
      CHARACTER *4 FULL,PART,OPEN,STCL,DETR,STMA,WORD
      CHARACTER *12 FINAME
      CHARACTER *4 TEXT(11),FUELTP(4)
      CHARACTER *42 SLINFO(3)
      DATA FULL,PART,OPEN/'FULL','PART','OPEN'/
```

```
      DATA STCL,DETR,STMA/'STCL','DETR','STMA'/
      DATA HLIFE/0.033,.5,1.,6.,12.,18.,24.,48.,48.,8760./
      DATA SLINFO(1)/' Pie   No. of particles   Mean radius(ft)'/
      DATA SLINFO(2)/' Strip   Particles   -Le(ft)  Y-mean   Le(ft)'/
      DATA SLINFO(3)/' Strip   Particles   -Le(ft)  X-mean   Le(ft)'/
         OPEN(15,FILE='lross.fnm')
         REWIND 15
         DO 2222 IFILES=1,13
         READ(15,1111)IUNIT,FINAME
         OPEN(IUNIT,FILE=FINAME)
         REWIND IUNIT
2222     CONTINUE
1111     FORMAT(I3,A12)
C
C  Explanation of Variables
C
C -- Single Varibles
C    The next four variables are for controlling output.  They can
C    have the values  0-NO , 1-YES
C
C      IOPT1  - Fixed data like Geometry and Bank (shore) conditions
C      IOPT2  - Computed Vleocities for plotting
C      IOPT3  - Location of particles for plotting
C      IOPT4  - Number Plot(particle distribution) on print
C
C      ISPTYP - Spill type  0-Instantaneous,  1-Continuous.  Computed by
C               model based on : if SPLTIM > 0.5*SPILDT  ISPTYP=1 else=0
C
C      FEVP1  - Fraction evaporated at previous time step
C      FEVP2  - Fraction evaporated at present time step
C
C      NBRNCH - No. of Branches in the 1-D Flow Model
C      NGRIDX - Total No. of grid boxes in X-direction
C
C      TOTDIS - Total amouunt of Dissolved Oil (gms)
C
C      UFDT   - 1-D Model time step(hrs)
C      UFSTPS - No. of 1-D model steps
C      OSTPS  - No. of Oilspill steps per UFDT
C
C -- Some important variable names used for intermediate computations
C      AIY      - area of the IYth Trapezoid
C      PERI     - wetted perimeter by IYth Trapezoid
C      HR       - hydraulic radius of IYth trapezoid
C
C -- One Dimensional Arrays
C      IGRILB(I) - y-dir grid box number of lower river boundary column I
C      IGRIUB(I) - y-dir grid box number of upper river boundary column I
C      LCSTSQ(I) - last section number of branch I
C      NSLSCT(I) - No. of slices of data for section I
C      NSTUBE(I) - No. of streamtubes for section I
C      NUMCON(I) - Condition Number (see text) for section I
C      NFIRCO(I) - Next section first connecting to section I
C      Q(I)      - Discharge in the Ith Branch
C      SCTANG(I) - angle Ith section makes with X-direction
C      WL(I)     - water level of upstream of branch I
```

```
C      ZD(I)     - reference level from datum for section I at which Z's
C                  are evaluated
C
C -- Two Dimensional Arrays
C      TICE(I,J) - Equivalent ice thickness of Jth vertical in Ith section
C      YWID(I,J) - Distance from lower bank of river to the Jth vertical
C                  in Ith section
C      Z(I,J)    - Height of Jth vertical in Ith section
C
C -- Complex Variables (these store X-component as real part and
C                           Y-component as imaginary part)
C      CORDLB(I) - lower bank co-ords of the Ith section
C      CORDV(I,J)- co-ords at which VSTRM(I,J) is acting
C      VSTRM(I,J)- stream velocity of the Ith section and Jth streamtube
C      VCAR(I)   - velocity in the cartesian box grid system of box I
C
C -- Variables derived from addition of lake
C      D         - depths in lake grid boxes
C      S         - stream function in lake grid boxes
C      U         - x-velocity in lake grid boxes
C      V         - y-velocity in lake grid boxes
C      DXL       - size of lake grid
C      DXR       - size of river grid
C      LGRIDX    - number of x-grids along lake
C      IRGRID    - number of x-grids along river
C      NGRIDX    - total number of combined river and lake x-grid boxes
C      BEGLK     - x-coordinate of left-most side of lake (negative value)
C      ILVCAR    - total number of lake grids containing velocities
C      LICERG    - number of ice regions only in lake
C      ZLKICE(I) - ice thickness in corresponding lake ice region
C    -----------------------------------------------------------------
C
       READ(1,650)WORD,TEXT
C
C    Read the lake and river grid control parameters
C
       READ(1,*)NBRNCH,LGRIDX,NGRIDX,DXL,DXR,KINTM,BEGLK
       READ(1,*)(LCSTSQ(I),I=1,NBRNCH)
       IRGRID = NGRIDX - LGRIDX
       IS2 = LCSTSQ(NBRNCH)+ 1
C
C    Read river cross section and geometry data
C
       DO 100 I=1,IS2
          READ(1,*)J,CORDLB(I),SCTANG(I),NSTUBE(I),NUMCON(I),NFIRCO(I)
     $    ,NSECO(I)
          IF(J.NE.I)WRITE(*,700)
C         SCTANG(I) = SCTANG(I)*3.1415/180.
100       CONTINUE
       DO 110 I=1,IS2
          READ(1,*)J,NSLSCT(I),ZD(I)
          IF(J.NE.I)WRITE(*,710)
          NNN=NSLSCT(I)+1
          READ(1,*)(YWID(I,J),Z(I,J),J=2,NNN)
110       CONTINUE
       DO 120 I=1,NGRIDX
```

```
        READ(1,*)J,IGRILB(I),IGRIUB(I),IGRLB1(I),IGRUB1(I)
        IF(I.NE.J)WRITE(*,720)
120     CONTINUE
C
C   Read the I,J values of Grid boxes in which velocity =0.0
C
      READ(1,*)NZRVB
      IF(NZRVB.EQ.0)GOTO 140
      IF(NZRVB.GT.100)WRITE(*,730)
      IF(NZRVB.GT.100)NZRVB=100
      READ(1,*)(IZRBX(I),IZRBY(I),I=1,NZRVB)
C
C   Read the spill volume and spill location
C
140   READ(12,650) FUELTP
      READ(12,*)TOTIME,IEVERY,IOPT1,IOPT2,IOPT3,IOPT4,SPLTIM,
     $ DIFFUL,DIFFUR
      READ(12,*)NTOTAL,SPVOL,SPILDT,SPGOIL,ANIU,SIGMA,AK2I,AK2V,AK2T
     $ ,AKC10,AKC20,AKC30
C
C   Check for instantaneous or continuous spill
C
       ISPTYP = 0
       IF(SPLTIM.GT.0.5*SPILDT)ISPTYP=1
       READ(5,*)ANICE,AMUNI
C
C   SPVOL is U.S. gallons.  VOLPAR is cu ft. of volume per particle
C
      SPLRAT = 0.13368*SPVOL/SPLTIM
      VOLPAR = 0.13368*SPVOL/NTOTAL
      VZERO = SPVOL*3.7850E-03
      API = 0.0
      READ(12,*)SPX,SPY,VMUNI,SOLUNI,CEVP,T0EVP
      SOLBLT = SOLUNI*16018.453
      VMOL = VMUNI*0.02831682
      AMIUO = AMUNI*14.88162
      TOUNI = T0EVP*9./5.0
      IF(T0EVP.LT.1.0)API = 141.5/SPGOIL - 131.5
      APITEM = 141.5/SPGOIL -131.5
      SPCEN0 = CMPLX(SPX,SPY)
C
C   Check if the spill co-ordinates are in land
C    (This is a check for input error)
C
      IF (SPX .GT. 0.0E0) GOTO 145
      L = (SPX-BEGLK)/DXL + 1.0
      M = SPY/DXL + 1.0
      GOTO 155
145   L = SPX/DXR + 1.0 + LGRIDX
      M = SPY/DXR + 1.0
155   IF(M.LT.IGRILB(L).OR.M.GT.IGRIUB(L))GOTO 150
      IF(IGRLB1(L).EQ.0)GOTO 160
      IF(M.GE.IGRLB1(L).AND.M.LE.IGRUB1(L))GOTO 150
      GOTO 160
150   WRITE(*,800)L,M
      STOP
```

```
160   CONTINUE
      TTTT=SPLTIM/60.
      IF(ISPTYP.EQ.1)WRITE(*,810)TEXT,SPCENO,TTTT
      IF(ISPTYP.EQ.0)WRITE(*,820)TEXT,SPCENO
      WRITE(*,830)TOTIME,NTOTAL,SPVOL,FUELTP,SPILDT,SPGOIL,APITEM
     $ ,ANIU,SIGMA
      WRITE(*,840)AK2I,AK2V,AK2T,AKC10,AKC20,AKC30,VMUNI,SOLUNI
      WRITE(*,842)AMUNI,ANICE
      WRITE(*,*) '          Surface Diffusion'
      IF(DIFFUL.LT.0.0)WRITE(*,*)' LAKE - Default formulation is used'
      IF(DIFFUL.GE.0.0)WRITE(*,846)DIFFUL
      IF(DIFFUR.LT.0.0)WRITE(*,*)' RIVER- Default formulation is used'
      IF(DIFFUR.GE.0.0)WRITE(*,847)DIFFUR
846   FORMAT(' LAKE - coeff. =',F6.2,' sq ft/sec'//)
847   FORMAT(' RIVER- coeff. =',F6.2,' sq ft/sec'//)
      IF(API.LT.1.) WRITE(*,844)CEVP,T0EVP
843   FORMAT(///' Meteorological Station Data Used in Lake Circulation'
     $ ,' Model'
     $/6X,'Time    Lat.   Long. Height   T-air   T-H2O     Wind'/
     $7X, 'hrs    deg     deg     ft       F        F     mph  deg'/)
844   FORMAT(//' API option is not selected . Evap. constants are ',
     $ ' C= ',F6.2,'  T0= ',F7.1)
C
C   Read boundary type information and calculate rejection rates
C
170   READ(8,*)K,LFROM,LTO,ICODE
      IF(K.EQ.0)GOTO 190
      AN = HLIFE(ICODE)*3600./SPILDT
      REJRAT = 1 - 0.5**(1./AN)
      DO 180 L=LFROM,LTO
180   TYPBND(K,L)=REJRAT
      GOTO 170
190   CONTINUE
C
C   If continuous spill, find number of particles released over time step
C
      NSPILS=(SPLTIM+1.0)/SPILDT
      IF(ISPTYP.EQ.1)NPERDT = NTOTAL/NSPILS
      IF(ISPTYP.EQ.1)GOTO 210
      NPERDT=NTOTAL
      DO 200 I=1,NTOTAL
200   PARTCL(I) = SPCENO
210   CONTINUE
C
C   First set Vol of each pie=8*one eighth of vol released in SPILDT
C
      DO 220 I=1,8
220   VOLPIE(I) = VOLPAR*NPERDT
C
C   Set random number generation seed IX
C
      IX=13
      WRITE(11,650)TEXT,FUELTP
      WRITE(11,651)NTOTAL,SPVOL,SPILDT,SPGOIL,ANIU,SIGMA,VMUNI,SOLUNI
     $ ,AMUNI
      TIMET = 0.
```

```
      IPARTX(1)=REAL(SPCEN0)
      IPARTY(1)=AIMAG(SPCEN0)
      WRITE(11,850)IPARTX(1),IPARTY(1)
      INDX1D = 0
      NST =1
      NPTCL = NPERDT
      FEVP1=0.
      FEVP2=0.
      TOTDIS=0.
      NBRP1 = NBRNCH + 1
C
C   Read flow condition update interval
C
      READ(7,*)UFDT
      WRITE(*,845)UFDT
      UFSTPS = TOTIME/UFDT
      OSTPS = (UFDT*3600.+1.0)/SPILDT
      DO 340 IUF=1,UFSTPS
      REWIND 3
      REWIND 4
C
C   Read Data Created by unsteady Flow Model
C
        IF(WORD.EQ.STCL)IRCODE=1
        IF(WORD.EQ.DETR)IRCODE=2
        IF(WORD.EQ.STMA)IRCODE=3
C
C   If the numbering sequence of branches in oilspill model is the
C   same as that of 1-D model the following 3 statements can be used to
C   Read the Q & WL data. In this case subroutine NDCONV is not needed.
C   Subroutine NDCONV is specifically written for reading Q & WL from
C   the three 1-D River models St. Clair, Detroit and St. Mary's
C
C       DO 230 I=1,NBRP1
C          READ(7,*)WL(I),Q(I)
C230       CONTINUE
        CALL NDCONV(NBRP1,IRCODE)
C
C   Read ice thickness information for cross sections
C
      READ(7,*)ICINFO
      DO 270 I=1,ICINFO
         READ(7,660)IS,WORD
         NNN=NSLSCT(IS)+1
         IF(ICINFO.EQ.1.AND.WORD.EQ.OPEN) THEN
         WRITE(*,234)
         ELSE
         IF(I.EQ.1)WRITE(*,235)
         ENDIF
         IF(WORD.NE.FULL)GOTO 250
         READ(7,*)FULLTI
         DO 240 K=1,NNN
            TICE(IS,K)=FULLTI
240         CONTINUE
         WRITE(*,236)IS,FULLTI
250         IF(WORD.EQ.PART) THEN
```

```
              READ(7,*)(TICE(IS,J),J=1,NNN)
              DO 252 J=1,NNN
              IDUM(J) = YWID(IS,J)
252           CONTINUE
              WRITE(*,237) IS,(IDUM(J),J=1,NNN)
              WRITE(*,238) (TICE(IS,J),J=1,NNN)
              ENDIF
           IF(WORD.NE.OPEN)GOTO 270
           DO 260 K=1,NNN
              TICE(IS,K)=0.0
260           CONTINUE
270     CONTINUE
C
C    Read ice region information for spreading and advection
C
        READ(5,*)NICERG, LICERG
        IF(NICERG.NE.0)THEN
        DO 275 I=1,NICERG
          READ(5,*)NICEX1(I),NICEY1(I),NICEX2(I),NICEY2(I)
          IF (I .LE. LICERG) READ(5,*) ZLKICE(I)
275       CONTINUE
        ENDIF
        ICERGR = NICERG - LICERG
        LR1 = LICERG + 1
        IF(ICERGR.GT.0)WRITE(*,932)ICERGR,(I,NICEX1(I),NICEY1(I),NICEX2(I)
       $ ,NICEY2(I),I=LR1,NICERG)
930     FORMAT(///10X,'No. of Ice Covered Regions in the Lake=',I3/
       $ 5X,' Region    from X,Y Grid to X,Y Grid  Ice Thic(ft)'/
       $(5X,I4,11X,I3,',',I3,6X,I3,',',I3,5X,F6.2))
932     FORMAT(/10X,'No. of Ice Covered Regions in the river=',I3/' ( for
       $river ice cover thickness refer to Ice Conditions at X-sections)'
       $//5X,' Region    from X,Y Grid to X,Y Grid'/
       $ (5X,I4,11X,I3,',',I3,6X,I3,',',I3))
C
C     write flow and discharge info
C
        WRITE(*,860)
        DO 280 I=1,NBRNCH
280     WRITE(*,870)I,Q(I),WL(I)
C
        IF(LICERG.GT.0)THEN
        WRITE(*,930)LICERG,(I,NICEX1(I),NICEY1(I),NICEX2(I),NICEY2(I),
       $ ZLKICE(I),I=1,LICERG)
        ENDIF
C
C    Set up the one dimensional array locations that define Ice Regions
C
        IF(NICERG.EQ.0)GOTO 278
        DO 40 K=1,NICERG
          LK1=NICEX1(K)-1
          MK1=NICEY1(K)-1
          IPOS1(K) = 0
          IF(LK1.EQ.0)GOTO 45
          DO 44 L1=1,LK1
            IPOS1(K) = IPOS1(K)+IGRIUB(L1)-IGRILB(L1)+3
44          CONTINUE
```

```
45        IPOS1(K) = IPOS1(K)+MK1-IGRILB(LK1+1)+3
          LK2=NICEX2(K)-1
          MK2=NICEY2(K)-1
          IPOS2(K) = 0
          IF(LK2.EQ.0)GOTO 48
          DO 47 L1=1,LK2
            IPOS2(K) = IPOS2(K)+IGRIUB(L1)-IGRILB(L1)+3
47          CONTINUE
48        IPOS2(K) = IPOS2(K)+MK2-IGRILB(LK2+1)+3
40      CONTINUE
278     CONTINUE
C     Call Lake Circulation Model (RLID) to calculate and stabilize stream
C     function values for lake grids.  RLID runs for time step of 1 hour for
C     a total of 20 hours.  Stream function values at the end of this time
C     period will be written to a temporary file.  Rlid will then be called and
C     run for every UFDT amount of time to update the lake stream functions.
C     The stage and discharge at the river entrance is used for the lake.
        INDPRN = 0
        IF(LICERG.EQ.0)WRITE(*,'(//A40)')' Open Water conditions exist in
       $the lake'
        CALL RLID(UFDT, Q(1), WL(1), TIMET, INDPRN)
C     Call subroutine PARTIC to calculate velocities in the lake grid
C     boxes at initial spill time.  PARTIC is called each UFDT amount of
C     time to update the lake velocities.  ILVCAR counts how many
C     velocities are written to VCAR(I).
        CALL PARTIC(WL(1), ILVCAR, IOPT2)
C
C     Now call VELDIS to find the 2-D vel distribution in the river
C
        IF(IOPT1.EQ.1)CALL PRINT1(2, NBRNCH)
        CALL VELDIS(IOPT2, NBRNCH, ILVCAR)
        NST1=(IUF-1)*OSTPS+1
        NST2= NST1 + OSTPS -1
        DO 330 I=NST1,NST2
        READ(12,*)VWMAG,THETA,TENVF
        THET = (127.1-THETA)*3.141592/180.
        VWX=VWMAG*SIN(THET)
        VWY= - VWMAG*COS(THET)
        VWIND = CMPLX(VWX,VWY)
        WNDSPD = VWMAG/3.28
        VWMPH = VWMAG*0.6818
        TENV = (TENVF-32)*5./9. + 273.
        INDPRN = 0
        IF(MOD(I-1,IEVERY).EQ.0)INDPRN = 1
        TIMET = TIMET + SPILDT
        IF(ISPTYP.NE.1)GOTO 290
        IF(I.LE.NSPILS)CALL PRELSE(SPILDT,IX,NST,NPTCL,SPCEN0,DIFFUL,
       $  DIFFUR)
        IF(I.GT.1)CALL ADVECT(SPILDT, IX, 1, NST-1, DIFFUL, DIFFUR ).
290     IF(ISPTYP.EQ.0)CALL ADVECT(SPILDT, IX, 1, NTOTAL,DIFFUL,DIFFUR)
        CALL ORIENT(INDX1D,ANGLE)
        IF(INDX1D.LT.3)GOTO 293
        IF(NICERG.EQ.0)INDX1D=INDX1D-3
        IF(NICERG.EQ.0)GOTO 293
        NPTICE=0
        DO 292 KK=1,NMOVIN
```

```
        J= IMOVIN(KK)
        SPX = REAL(PARTCL(J))
        IF (SPX .GT. 0.0E0) GOTO 144
        L = (SPX - BEGLK)/DXL
        M = (AIMAG(PARTCL(J))+YSHIFT(J))/DXL
        GOTO 154
144     L = SPX/DXR + LGRIDX
        M = (AIMAG(PARTCL(J))+YSHIFT(J))/DXR
154     CONTINUE
        IPOS = 0
        IF(L.EQ.0)GOTO 117
        DO 115 L1=1,L
          IPOS = IPOS+IGRIUB(L1)-IGRILB(L1)+3
115     CONTINUE
117     IPOS = IPOS+M-IGRILB(L+1)+3
        DO 118 K=1,NICERG
          IF(IPOS.GE.IPOS1(K).AND.IPOS.LE.IPOS2(K))NPTICE=NPTICE+1
118       CONTINUE
292     CONTINUE
        RATICE=FLOAT(NPTICE)/FLOAT(NMOVIN)
        IF(RATICE.LT.0.5)INDX1D=INDX1D-3
        IF(RATICE.GE.0.5)INDX1D=0
293   TTTT=TIMET/3600.
      GALPAR = VOLPAR/0.13368
      IF(INDPRN.EQ.1)WRITE(*,880)TTTT,VWMPH,THETA,TENVF,SPCEN,GALPAR,
     $  SLINFO(INDX1D+1)
      IF(INDX1D.EQ.0)CALL SPRDAX(SPILDT, TIMET, INDPRN, SPAREA
     $ ,SPLTIM, SPLRAT)
      IF(INDX1D.EQ.1.OR.INDX1D.EQ.2)CALL SPRD1D(SPILDT, TIMET, INDPRN,
     $ SPAREA , ANGLE)
      FEVP1=FEVP2
      CALL EVAPOR(API,TENV,WNDSPD,VMOL,VZERO,SPAREA,SPILDT,I)
      CALL DISOLU(SPAREA,SOLBLT,TIMET,SPILDT,API,DELDIS,TOTDIS)
      DELUNI = DELDIS*264.172E-06/SPGOIL
      TOTUNI = TOTDIS*264.172E-06/SPGOIL
      VOLPAR= 0.13368*(SPVOL*(1-FEVP2)-TOTUNI)/NTOTAL
      IF(NHITB.GT.0)CALL BOUNDR(INDPRN)
      IF(I.GE.NSPILS)NST = NPTCL +1
      IF(I.GE.NSPILS)GOTO 300
      NST=NPTCL+1
      NPTCL=NST+NPERDT-1
300   IF(INDPRN.NE.1)GOTO 330
      WRITE(*,900)FEVP2,DELUNI,TOTUNI
      IF(IOPT4.EQ.1)CALL PLOTNU(6)
      IF(IOPT3.NE.1)GOTO 330
      IPARTX(1)=REAL(SPCEN)
      IPARTY(1)=AIMAG(SPCEN)
      WRITE(11,910)IPARTX(1),IPARTY(1),TTTT,VWX,VWY,GALPAR
      DO 320 J=1,NTOTAL,5
        DO 310 K=1,5
          IPARTX(K) = REAL(PARTCL(J+K-1))
          IPARTY(K) = AIMAG(PARTCL(J+K-1))
310       CONTINUE
        WRITE(11,850)(IPARTX(K),IPARTY(K),K=1,5)
320     CONTINUE
330   CONTINUE
```

```
340  CONTINUE
     STOP
234  FORMAT(///' Open Water Conditions exist in the river ')
235  FORMAT(//' Ice conditions at cross sections',/1H+,32('_')//
    $' X-sect',9X,'Condition (A thickness of 0.0 implies open water)')
236  FORMAT(/I4,5X,'Ice cover of uniform thickness =',F5.2,
    $ ' ft across the river')
237  FORMAT(/I4,5X,'Partial or non-uniform ice cover across the river.'
    $,' Distances from'/9X,'the lower bank and corresponding ice'
    $ ' thickness is given below'/9X,'Dist(ft) ',20I5)
238  FORMAT(9X,'Thic(ft) ',20F5.2)
720  FORMAT(' ** ERROR ** READING GRID INFO.')
730  FORMAT(' NZRVB is GT 100 and is reset to 100')
710  FORMAT(' ** ERROR ** READING X SECTION - DATA ')
700  FORMAT(' ** ERROR ** READING LOWER BOUNDARY - DATA ')
660  FORMAT(I4,1X,A4)
650  FORMAT(20A4)
651  FORMAT(I4,F8.0,F7.0,F6.2,5E11.3)
810  FORMAT(8X,11A4,//5X,25('*')/5X,'*    CONTINUOUS SPILL     *'/5X,
    $ '*',11X,'AT',10X,'*'/5X,'*',4X,F7.0,',',F7.0,4X,'*'/
    $ 5X,'*',5X,'FOR ',F5.0,' min.',4X,'*'/5X,25('*'))
820     FORMAT(8X,11A4,//5X,25('*')/5X,'*  INSTANTANEOUS SPILL  *'/5X,
    $ '*',11X,'AT',10X,'*'/5X,'*',4X,F7.0,',',F7.0,4X,'*'/5X,25('*'))
830  FORMAT(//' SIMULATION PERIOD  = ',F5.1,' Hrs'///
    $ ' Characteristics of spill'/1H+,24('_')//
    $ ' No. of particles            :',I5,/
    $ ' Oil spilled                 :',F8.0,' gals of ',4A4,/
    $ ' DT for spill simulation     :',F8.0,' Secs.',/
    $ ' Specific gravity of oil     :',F8.2,' (API index =',F5.1,')'/
    $ ' Kinematic Visco. of Water   :',E10.4,' sq ft/sec',/
    $ ' Surafce Tension             :',E10.4,' lbs/ft',/)
840  FORMAT (/9X,'Spreading Coefficients' /
    $ '    K2i    K2v    K2t    c10    c20    c30',/,6F6.2//
    $ ' Molar volume                :',E10.4,' cu ft/mol'/
    $ ' Solubility of fresh oil     :',E10.4,' lbs/cu ft')
842  FORMAT (' Viscosity of Oil            :',F8.2,' lbs/ft-sec'/
    $ ' Manning s Roughness of Ice :',F8.3/)
845  FORMAT(/' Time step for river flow computation = ',F6.2,'Hrs')
910  FORMAT(I8,I7,F9.4,2F6.1,F8.2)
850  FORMAT(5(I8,I7))
880  FORMAT(//1X,78('-')/' Time = ',F6.2,' Hrs -- Wind :mag=',F5.1,
    $ ' mph, dir =',F5.1,' deg    -- Air Temp=',F5.1,' F'/
    $ ' Spill center after advection= ',F7.0,',',F7.0,' (ft)'/
    $ ' Volume per particle            = ',F7.3,' gals'//
    $ '        Slick condition during this time step'//
    $ ' Slick information by pie / strip',/A42)
900  FORMAT(//8X,'Slick condition at the end of this time step'
    $ //' Fraction Evaporated = ',G10.5/' Amount Dissolved (gals)
    $ : This Step = ',G10.5,' Total = ',G10.5/)
860  FORMAT(///' Flow and Discharge Conditions in the River'/
    $ '  Branch    Q (cfs)    Stage (ft)    ')
870  FORMAT(4X,I2,5X,F7.0,5X,F6.2)
800  FORMAT(//' Spill location co-ords are in land X & Y GRID box no.s
    $ are ',I4,' &',I4,//'        Execution terminated        ')
     END
```

```
      SUBROUTINE ADVECT(SPILDT, IX, N1, N2, DIFFUL, DIFFUR)
C
C  This subroutine handles the slick advection in each time step
C  Each particle is advected according to current & wind velociites
C  (see text for details)
C      -- This routine advects moving particles in the range N1 to N2
C      -- This version includes advection under ice covers
C      -- Modifications for addition of lake were made.
C
C      -- Last Date of Revision :  Sept 19, 1986
C
      COMPLEX VCAR(8000),SPCEN,PARTCL(1000),VWIND,VDRIFT
      COMMON /VA/ VCAR,VWIND,VDRIFT
      COMMON /VASB/IGRILB(300),IGRIUB(300),IGRLB1(300),IGRUB1(300)
      COMMON /ASB/SPCEN,PARTCL,NPTCL,NHITB,IHITB(1000),TYPBND(4,300)
      COMMON /GRIDS/ DXL, DXR, LGRIDX, NGRIDX, IRGRID, BEGLK
      COMMON /BLOCK7/SPGOIL,ANIU,SIGMA,AK2I,AK2V,AK2T,
     $ VOLPAR,VOLPIE(8),SLICKR(8)
      COMMON /ICE/ZWND(40,40),ZLKICE(20),NICEX1(20),NICEY1(20),
     $ NICEX2(20),NICEY2(20),IPOS1(20),IPOS2(20),AMIUO,ANICE,
     $ SPAICE,NICERG,LICERG
C
C  Input : .. Location of each particle
C           .. Velocity distribution in river
C  Output: .. New loaction of each particle
C
      DATA PI/3.141592/, STPTIM/0.0/
      STPTIM = STPTIM + SPILDT
      IF(NICERG.EQ.0)GOTO 25
C
C  DELEQ - Equilibrium thickness (ft)
C  UTH   - Threshold current speed for slick movement ( ft/sec)
C  UFAIL - Failure velocity under rough ice cover ( ft/sec)
C  FRAMFA- FRiction AMplification FActor denoted by 'K' in text
C  AMIUO - Viscosity of Oil in g/cm-sec
C
      DELEQ = (1.67 - 8.5*(1.0-SPGOIL))/30.48
      UTH = (305.79/(88.68-AMIUO))/30.48
      FRAMFA = 35.55*ANICE + 1.0
      IF(ANICE.GT.0.045)FRAMFA = 2.6
      TERM1=SQRT(SIGMA*(32.2)**2*62.4*(1.-SPGOIL))
      UFAIL=1.5*SQRT(2.*(1.+SPGOIL)*TERM1/(62.4*SPGOIL))
      ROUGH = (ANICE/0.034)**6
C
C  Loop 60 operates for each moving particle in the system
C
25    DO 60 I=N1,N2
         SUMDT = 0.
         IPASS = 1
C   SUMDT - Sum of the small Dt's  (DTSMAL)
C   IPASS - pass number in this loop. A particle may move from its
C   previous position to present position through only one pass or
C   several passes depending on the magnitude of velocities in the region
40          DO 30 J=1,NHITB
            IF(I.EQ.IHITB(J))GOTO 60
```

```
30        CONTINUE
          SPX = REAL(PARTCL(I))
          IF (SPX .GT. 0.0D0) GOTO 144
          L = (SPX - BEGLK)/DXL
          M = AIMAG(PARTCL(I))/DXL
          GOTO 154
144       L = SPX/DXR + LGRIDX
          M = AIMAG(PARTCL(I))/DXR
154       CONTINUE
          IPOS = 0
          IF(L.EQ.0)GOTO 117
          DO 115 L1=1,L
             IPOS = IPOS+IGRIUB(L1)-IGRILB(L1)+3
115          CONTINUE
117       IPOS = IPOS+M-IGRILB(L+1)+3
          IF(NICERG.EQ.0)GOTO 125
C
C  Determine whether the particle is under ice or not
C
          ICOND=0
          DO 120 K=1,NICERG
             IF(IPOS.GE.IPOS1(K).AND.IPOS.LE.IPOS2(K))ICOND=1
             IF(ICOND.EQ.1)GOTO 180
120          CONTINUE
C
C  Advection velocity in free-surface conditions
C
125     VDRIFT = 0.03*VWIND + 1.1*VCAR(IPOS)
C
C  Add in turbulent fluctuation
C
        GOTO 210
C
C  Advection Velocity under Ice
C
180     UWATER = CABS(VCAR(IPOS))
        IF(ROUGH.GT.DELEQ)GOTO 190
        IF(UWATER.LT.UTH)GOTO 60
        GOTO 200
190     IF(UWATER.LT.UFAIL)GOTO 60
200     VDRIFT = VCAR(IPOS)
        FDELTA = UWATER/SQRT((1.0-SPGOIL)*32.2*DELEQ)
        FK = SQRT(FRAMFA/(0.115*FDELTA**2 + 1.105))
        VDRIFT = VDRIFT*(1-FK)
210     DTSMAL = 86400.
C       86400 is just a large number in this case equal to secs in a day
C
        VELUPV =  ABS(REAL(VDRIFT)) + ABS(AIMAG(VDRIFT))
        IF(VELUPV.GT.0.01) THEN
           IF(SPX.GE.0.0) DTSMAL = DXR/VELUPV
           IF(SPX.LT.0.0) DTSMAL = DXL/VELUPV
           ENDIF
        IF(DTSMAL.GT.SPILDT)THEN
           IF(IPASS.EQ.1)DTSMAL = SPILDT
           IF(IPASS.GT.1)DTSMAL = SPILDT - SUMDT
           ENDIF
```

```
      IF(DTSMAL.LT.0.0)DTSMAL = 0.
      IF((SUMDT+DTSMAL).GE.SPILDT) IPASS = 9999
      SUMDT = SUMDT + DTSMAL
      CALL RANDU(IX,IY,YFL)
      IX = IY
      ANG = PI*YFL
      CALL GAUSS(IX,1.0,0.0,VRAND)
      IF(SPX.LT.0.0) THEN
         TELAPS = STPTIM+SUMDT-DTSMAL/2.0
         IF(DIFFUL.LT.0.0) VPRIME = 3.407E-03*TELAPS**0.67/SQRT(DTSMAL)
         IF(DIFFUL.GE.0.0) VPRIME = SQRT(4*DIFFUL/DTSMAL)
         ENDIF
      IF(SPX.GE.0.0)THEN
         IF(DIFFUR.LT.0.0) DDD = 2.88*CABS(VDRIFT)
         IF(DIFFUR.GE.0.0) DDD = 4*DIFFUR
         VRPIME =SQRT(DDD/DTSMAL)
         ENDIF
      VRAND = VPRIME*VRAND
      VX = VRAND*COS(ANG)
      VY = VRAND*SIN(ANG)
      VMAG = CABS(VDRIFT)
      VDRIFT = VDRIFT+ CMPLX(VX,VY)
      PARTCL(I) = PARTCL(I) + DTSMAL*VDRIFT
      IF(IPASS.NE.9999) GOTO 40
C
C  Check for particle hitting the boundaries
C
      SPX = REAL(PARTCL(I))
      IF (SPX .GT. 0.0E0) GOTO 145
      L = (SPX - BEGLK)/DXL + 1
      M = AIMAG(PARTCL(I))/DXL + 1
      GOTO 155
145   L = SPX/DXR + 1 + LGRIDX
      M = AIMAG(PARTCL(I))/DXR + 1
155   CONTINUE
      IF(M.GE.IGRILB(L).AND.M.LE.IGRIUB(L))GOTO 55
      NHITB = NHITB + 1
      IHITB(NHITB) = I
55    IF(IGRLB1(L).EQ.0)GOTO 59
      IF(M.LE.IGRUB1(L).AND.M.GE.IGRLB1(L))GOTO 58
      GOTO 59
58    NHITB = NHITB+1
      IHITB(NHITB) = I
59    CONTINUE
      IF(IPASS.NE.9999)GOTO 40
60    CONTINUE
      RETURN
      END
```

Subroutine BOUNDR

```
      SUBROUTINE BOUNDR(INDPRN)
      COMPLEX SPCEN,PARTCL(1000)
      COMMON /VASB/IGRILB(300),IGRIUB(300),IGRLB1(300),IGRUB1(300)
      COMMON /ASB/SPCEN,PARTCL,NPTCL,NHITB,IHITB(1000),TYPBND(4,300)
      COMMON /GRIDS/ DXL, DXR, LGRIDX, NGRIDX, IRGRID, BEGLK
      DIMENSION NPTBND(4,325),IDUM1(300),IDUM2(300)
C
C  This subroutine handles adsorption and rejection at the boundaries
C  -- Modifications for addition of lake were made.
C
C  -- Last Date of Revision October 29,1985
C
      DO 10 I=1,NGRIDX
        DO 10 K=1,4
10      NPTBND(K,I)=0
      DO 80 I=1,NHITB
      J = IHITB(I)
      SPX = REAL(PARTCL(J))
      IF (SPX .GT. 0.0D0) GOTO 144
      L = (SPX - BEGLK)/DXL + 1
      M = AIMAG(PARTCL(J))/DXL + 1
      DX = DXL
      GOTO 154
144   L = SPX/DXR + 1 + LGRIDX
      M = AIMAG(PARTCL(J))/DXR +1
      DX = DXR
154   CONTINUE
C
C  Check if the particle is below the lower boundary, If so assign
C  it to the boundary grid and count
C
      IF(M.GE.IGRILB(L))GOTO 20
      IF(M.EQ.(IGRILB(L)-1))GOTO 15
      X1 = REAL(PARTCL(J))
      Y1 = IGRILB(L)*DX - 1.5*DX
      PARTCL(J) = CMPLX(X1,Y1)
15    NPTBND(1,L) = NPTBND(1,L) + 1
      GOTO 80
C
C  Check if the particle is above the upper boundary, If so assign
C  it to the boundary grid and count
C
20    IF(M.LE.IGRIUB(L))GOTO 40
      IF(M.EQ.(IGRIUB(L)+1))GOTO 30
      X1 = REAL(PARTCL(J))
      Y1 = IGRIUB(L)*DX + DX/2.
      PARTCL(J) = CMPLX(X1,Y1)
30    NPTBND(2,L) = NPTBND(2,L) + 1
      GOTO 80
C
C  If it didn't belong to the two categories above, it must be in the
C  Island, therefore assign to the nearest boundary and count
C
40    Y2 = AIMAG(PARTCL(J))-IGRLB1(L)*DX + 0.75*DX
      Y3 = IGRUB1(L)*DX - 0.25*DX - AIMAG(PARTCL(J))
      IF(Y2.LT.Y3)PARTCL(J) = PARTCL(J) - CMPLX(0.,Y2)
```

```
      IF(Y2.LT.Y3)NPTBND(3,L) = NPTBND(3,L) + 1
      IF(Y3.LT.Y2)PARTCL(J) = PARTCL(J) + CMPLX(0.,Y3)
      IF(Y3.LT.Y2)NPTBND(4,L) = NPTBND(4,L) + 1
80    CONTINUE
C
C  Number of particles in each boundary grid has been determined
C  Now check for the boundary type and re-entrain the excess particles
C
      IF(INDPRN.EQ.1)WRITE(*,300)
      DO 220 L1=1,NGRIDX
      NBNDR = 2
      IF(IGRLB1(L1).NE.0)NBNDR=4
      DO 210 K=1,NBNDR
      IF(NPTBND(K,L1).EQ.0)GOTO 210
      IALOWD = 0.5 +  (1.-TYPBND(K,L1))*NPTBND(K,L1)
      KOUNT = 0
      I = 0
90    I = I + 1
      IF(I.GT.NHITB)GOTO 205
      J = IHITB(I)
      SPX = REAL(PARTCL(J))
      IF (SPX .GT. 0.0D0) GOTO 145
      L = (SPX - BEGLK)/DXL + 1
      M = AIMAG(PARTCL(J))/DXL + 1
      DX = DXL
      GOTO 155
145   L = SPX/DXR + 1 + LGRIDX
      M = AIMAG(PARTCL(J))/DXR +1
      DX = DXR
155   CONTINUE
      IF(L.NE.L1)GOTO 90
      IF(K.NE.1)GOTO 110
      IF(M.NE.IGRILB(L)-1)GOTO 90
      KOUNT = KOUNT + 1
      IF(KOUNT.LE.IALOWD)GOTO 90
      IF(DX.EQ.DXL) XCO=L*DX + BEGLK - 0.5*DX
      IF(DX.EQ.DXR) XCO=(L-LGRIDX)*DX - 0.5*DX
      YCO=M*DX + 0.5*DX
      PARTCL(J) = CMPLX(XCO,YCO)
      NHITB = NHITB - 1
      DO 105 II =I,NHITB
105   IHITB(II) = IHITB(II+1)
      IHITB(NHITB+1)=0
         I=I-1
         GOTO 90
C
110   IF(K.NE.2)GOTO 130
      IF(M.NE.IGRIUB(L)+1)GOTO 90
      KOUNT = KOUNT + 1
      IF(KOUNT.LE.IALOWD)GOTO 90
      IF(DX.EQ.DXL) XCO=L*DX + BEGLK - 0.5*DX
      IF(DX.EQ.DXR) XCO=(L-LGRIDX)*DX - 0.5*DX
      YCO=M*DX - 1.5*DX
      PARTCL(J) = CMPLX(XCO,YCO)
      NHITB = NHITB - 1
      DO 115 II =I,NHITB
```

```
115   IHITB(II) = IHITB(II+1)
      IHITB(NHITB+1)=0
        I=I-1
        GOTO 90
C
130   IF(NBNDR.EQ.2)GOTO 90
      IF(K.NE.3)GOTO 150
      IF(M.NE.IGRLB1(L))GOTO 90
      IF(IGRUB1(L).NE.IGRLB1(L))GOTO 140
      XXX = AIMAG(PARTCL(J))-(IGRLB1(L)-1)*DX
      IF(XXX.GT.0.5*DX)GOTO 90
140   KOUNT = KOUNT + 1
      IF(KOUNT.LE.IALOWD)GOTO 90
      IF(DX.EQ.DXL) XCO=L*DX + BEGLK - 0.5*DX
      IF(DX.EQ.DXR) XCO=(L-LGRIDX)*DX - 0.5*DX
      YCO=M*DX - 1.5*DX
      PARTCL(J) = CMPLX(XCO,YCO)
      NHITB = NHITB - 1
      DO 125 II =I,NHITB
125   IHITB(II) = IHITB(II+1)
      IHITB(NHITB+1)=0
        I=I-1
        GOTO 90
C
150   IF(K.NE.4)GOTO 90
      IF(M.NE.IGRUB1(L))GOTO 90
      IF(IGRUB1(L).NE.IGRLB1(L))GOTO 160
      XXX = IGRUB1(L)*DX - AIMAG(PARTCL(J))
      IF(XXX.GT.0.5*DX)GOTO 90
160   KOUNT = KOUNT + 1
      IF(KOUNT.LE.IALOWD)GOTO 90
      IF(DX.EQ.DXL) XCO=L*DX + BEGLK - 0.5*DX
      IF(DX.EQ.DXR) XCO=(L-LGRIDX)*DX - 0.5*DX
      YCO=M*DX + 0.5*DX
      PARTCL(J) = CMPLX(XCO,YCO)
      NHITB = NHITB - 1
      DO 135 II =I,NHITB
135   IHITB(II) = IHITB(II+1)
      IHITB(NHITB+1)=0
        I=I-1
        GOTO 90
205     CONTINUE
210   CONTINUE
220   CONTINUE
      IF(INDPRN.EQ.0)RETURN
      DO 430 K=1,4
      KOUNT=0
      DO 410 L=1,NGRIDX
      IF(NPTBND(K,L).EQ.0)GOTO 410
      KOUNT = KOUNT+1
      IDUM1(KOUNT)=L
      IDUM2(KOUNT)=NPTBND(K,L)
      IF(KOUNT.GT.250)WRITE(420)
410   CONTINUE
      IF(KOUNT.EQ.0)GOTO 430
      K1=1
```

```
      K2=KOUNT
      IF(KOUNT.GT.28)K2=28
415   WRITE(*,440)K,(IDUM1(I),I=K1,K2)
      WRITE(*,450)(IDUM2(I),I=K1,K2)
      IF(K2.GE.KOUNT)GOTO 430
      K1=K2+1
      K2=K1+27
      GOTO 415
430   CONTINUE
      RETURN
300   FORMAT(/25X,'Oil in Lake and/or River Banks')
440   FORMAT(/' Bank',I2,'; X-Grid',28I4)
450   FORMAT(6X,'Particles',28I4)
310   FORMAT(5X,3(I3,3X))
      END
```

```
      SUBROUTINE DISOLU(SPAREA,SOLBLT,TIMET,SPILDT,API,DELDIS,TOTDIS)
C
C  This subroutine computes the amount of oil dissolved in water
C  The solubility of oil is so low that it does not affect the
C  trajectory (spreading), but is important for environmental impact
C  assessment.  --- The working units of this subroutine are METRIC
C
C  Explanation of variables
C  DISOLK   - Dissolution mass transfer coefficient   1 cm/hr
C                                                  or 2.7777E-06 m/sec
C  SOLBLT   - Solubility of fresh oil (g/cu m)
C  ARBAR    - mean area of slick during the time step (sq. m)
C  DELDIS   - amount of oil dissolved during time step (grams)
C  SPAREA   - Free surface area of spill (sq. ft)
C  SPAICE   - Area of spill under ice (sq. ft)
C
C  -- Last Date of Revision : October 29, 1985
C
      COMMON /ICE/ZWND(40,40),ZLKICE(20),NICEX1(20),NICEY1(20),
     $ NICEX2(20),NICEY2(20),IPOS1(20),IPOS2(20),AMIUO,ANICE,
     $ SPAICE,NICERG,LICERG
      DATA DISOLK/2.7777E-06/
      SPAR2=(SPAREA+SPAICE)/10.76
      ARBAR=(SPAR1+SPAR2)/2.0
      SPAR1=SPAR2
      T1=(TIMET-SPILDT)/36000
      T2=TIMET/36000.
      DELDIS=-DISOLK*ARBAR*SOLBLT*36E3*(EXP(-T2)-EXP(-T1))
      TOTDIS=TOTDIS+DELDIS
C     WRITE(*,*)SPAREA,SPAICE,ARBAR
      RETURN
      END
```

```
      SUBROUTINE EVAPOR(API,TENV,WNDSPD,VMOL,VZERO,SPAREA,SPILDT,JSTEP)
C
C     This subroutine computes evaporatioon rates based on
C     Mackay, Patterson and Nadeau ' s theory.
C     In this subroutine metric unit system is used.  The reason for
C     for using units differnt units from other subroutine is to make
C     cross reference with theory easier.
C
C     Last date of revision October 29,1985
C
C     Explanation of variables used in EVAPOR
C     AKM     - mass transfer coeff. Km   (m/s)
C     P0      - vapor pressure at TENV    (atm)
C     C       - coefficient C at TENV
C     FEVP2       - fraction evaporated
C     JSTEP   - current time step
C     RGAS    - gas constant:the values of RGAS in differnt units are
C             -  1.98726   cal/deg mole
C             -  8.31470   joules/deg mole
C             - 82.0597    cc-atm/deg mole
C     SPAREA - Free surface area of spill (sq. ft)
C     SPAICE - Area of spill under ice (sq. ft)
C
      COMMON /BLOCK7/SPGOIL,ANIU,SIGMA,AK2I,AK2V,AK2T,
     $ VOLPAR,VOLPIE(8),SLICKR(8)
      COMMON /SE/FEVP1,FEVP2,CEVP,T0EVP
      DATA RGAS/8.3147/
      DATA SCKET,SUMC,SUMP0/3*0./
      SPAR2=SPAREA/10.76
      ARBAR=(SPAR2+SPAR1)/2.0
      SPAR1=SPAR2
      IF(WNDSPD.LT.0.0001)WNDSPD=0.001
      AKM = 0.0025*WNDSPD**0.78
      C=CEVP
      T0=T0EVP
      IF(API.LT.1.0)GOTO 50
      C = 1158.9*API**(-1.1435)
      T0=542.6-30.27*API+1.565*API**2-3.439E-02*API**3+2.604E-04*API**4
50    P0= EXP(10.6*(1-T0/TENV))
      CC= C*283./TENV
      SUMP0 = SUMP0 + P0
      SUMC  = SUMC + CC
      P0BAR = SUMP0/JSTEP
      CBAR  = SUMC /JSTEP
      TOIL = TENV
      AKE = AKM*ARBAR*VMOL/(RGAS*TOIL*VZERO)
      SCKET = SCKET + CBAR*AKE*SPILDT*10.137E04
      SUME = SUME + AKE*SPILDT*10.137E04
      FEVP2 = (ALOG(P0BAR) + ALOG(SCKET+1./P0BAR))/CBAR
      IF(FEVP2.GT.0.6)FEVP2 = 0.6
      RETURN
      END
```

```
      SUBROUTINE NDCONV(NBRP1, IRCODE)
C
C  This subroutine reads nodal water level and discharge according
C  to the sequence from Detroit's 1-D flow model and then converts
C  to the branch B.C. required by Oilspill model.  If both have the
C  same sequence of numbering this subroutune is not required.
C
C  -- This subroutine is for the three rivers given below which currently
C     run on Detroit's one dimensional unsteady flow model.
C  -- This version has one additional branch added to Detroit River
C     causing a change in Q(14) & Q(1).
C  -- Modifications for addition of lake were made
C
C  -- Last Date of Revision : October 29, 1985
C
      DIMENSION DWL(22),DQ(22),NPTS(3)
      INTEGER RIV1(16),RIV2(17),RIV3(16)
      COMPLEX COMPXY,VSTRM(99,16),CORDV(99,16),VCAR(8000),CORDLB(99)
      COMPLEX SPCEN,PARTCL(1000),VWIND,VDRIFT
      COMMON /VEL/VSTRM,CORDV,CORDLB,Q(30),WL(30),TICE(99,20),
     $ YWID(99,20),Z(99,20),ZD(99),NSLSCT(99),SCTANG(99),
     $ LCSTSQ(30),NSTUBE(99),NUMCON(99),NFIRCO(99),NSECO(99),KINTM
      COMMON /VA/ VCAR,VWIND,VDRIFT
        DATA RIV1/1,2,3,4,5,5,7,6,7,8,9,5*1/
        DATA RIV2/1,1,14,3,16,4,6,7,8,9,10,11,18,13,20,21,22/
        DATA RIV3/1,9,10,3,4,5,6,7,11,13,13,5*1/
        DATA NPTS/9,22,13/
        N=NPTS(IRCODE)
        DO 10 I=1,N
          READ(7,*)DWL(I),DQ(I)
10     CONTINUE
        DO 20 I=1,NBRP1
        IF(IRCODE.EQ.1)K=RIV1(I)
        IF(IRCODE.EQ.2)K=RIV2(I)
        IF(IRCODE.EQ.3)K=RIV3(I)
        WL(I)=DWL(K)
        Q(I)=DQ(K)
20      CONTINUE
C
C  For Detroit River only
C
        IF(IRCODE.EQ.2)Q(14)=Q(13)+Q(12)
        IF(IRCODE.EQ.2)Q(1)=Q(2)+Q(3)
C
C  For St.Clair River only
C
        IF(IRCODE.NE.1)RETURN
        WL(7)=DWL(6)+ (DWL(5)-DWL(6))*20630./35680.
        Q(5) = DQ(6)*0.7
        Q(6) = DQ(6)*0.3
        RETURN
        END
```

```
      SUBROUTINE ORIENT(INDX1D, ANGLE)
C
C  This program computes the Orientation
C  and Aspect Ratio of the oil slick.
C
C  If Aspect Ratio >3, the slick will be treated as one dimensional.
C
C     INDX1D=0        :  Axisymmetrical spreading
C     INDX1D=1 or 2   :  One Dim. spreading
C     INDX1D=3        :  Axisymmetrical spreading(Short slick)
C     INDX1D=4 or 5   :  One Dim. spreading  (Short slick)
C
C  -- Last Date of Revision : September 19, 1986
C
      COMPLEX SPCEN,PARTCL(1000)
      COMMON /SO/IMOVIN(1000),YSHIFT(1000),NMOVIN,SSHIFT
      COMMON /ASB/SPCEN,PARTCL,NPTCL,NHITB,IHITB(1000),TYPBND(4,300)
      COMMON /VASB/IGRILB(300),IGRIUB(300),IGRLB1(300),IGRUB1(300)
      COMMON /GRIDS/ DXL, DXR, LGRIDX, NGRIDX, IRGRID, BEGLK
      PI = ATAN(1.) *4.
      NMOVIN=0
      INDX1D = 0
      COUNT = 0.
      SPCEN = (0.,0.)
C
C  Find the indeces of moving particles and assign them to
C  array IMOVIN(). Also compute the spill center (SPCEN)
C
      DO 30 I=1,NPTCL
        DO 15 J=1,NHITB
          IF(I.EQ.IHITB(J))GOTO 30
15        CONTINUE
        NMOVIN=NMOVIN+1
        IMOVIN(NMOVIN)=I
        SPCEN = SPCEN + PARTCL(I)
        COUNT = COUNT + 1.
30      CONTINUE
      SPCEN = SPCEN/COUNT
C
C  If there is an island, any particles in the upper channel are
C  shifted in (-)y-dir by a distance = width of island at that point.
C  IMPORTANT : The particles are moved back by equal amounts in
C  SPRDAX, SPRD1X or SPRD1Y subroutine.
C
      SSHIFT = 0.
      DO 430 I=1,NMOVIN
        J=IMOVIN(I)
        YSHIFT(J)=0.
        SPX = REAL(PARTCL(J))
        IF (SPX .GT. 0.0E0) GOTO 145
        L = (SPX - BEGLK)/DXL + 1
        M = AIMAG(PARTCL(J))/DXL + 1
        DX = DXL
        GOTO 155
145     L = SPX/DXR + 1 + LGRIDX
```

```
      M = AIMAG(PARTCL(J))/DXR +1
      DX = DXR
155   CONTINUE
      IF(IGRLB1(L).EQ.0)GOTO 430
      IF(M.LE.IGRUB1(L))GOTO 430
      YSHIFT(J)=(IGRUB1(L)-IGRLB1(L)+1)*DX
      PARTCL(J)=PARTCL(J)-CMPLX(0.,YSHIFT(J))
      SSHIFT=SSHIFT+YSHIFT(J)
430   CONTINUE
C
C  If particles are shifted, re-compute the Spill-Center
C
      SPX=REAL(SPCEN)
      SPY=AIMAG(SPCEN)
      IF(SSHIFT.LT.DXR)GOTO 450
      SPY=SPY - SSHIFT/NMOVIN
      SPCEN = CMPLX(SPX,SPY)
450   SUMIX=0.
      SUMIY=0.
      SUMIXY = 0.
      AVGRAD=0.0
      SPX=REAL(SPCEN)
      SPY=AIMAG(SPCEN)
      DO 50 I=1,NMOVIN
        J=IMOVIN(I)
        XX=REAL(PARTCL(J))-SPX
        YY=AIMAG(PARTCL(J))-SPY
        AVGRAD = AVGRAD + SQRT(XX*XX+YY*YY)
        SUMIXY = SUMIXY + XX*YY
        SUMIY=SUMIY+ XX*XX
        SUMIX=SUMIX+ YY*YY
50      CONTINUE
      AVGRAD = AVGRAD/NMOVIN
      TOP= -2*SUMIXY
      BOT= SUMIX-SUMIY
      THETA=ATAN2(TOP,BOT)
      THETA=THETA/2.0
      IF(THETA.LT.0.0)THETA=THETA+2*PI
      CTHETA = COS(THETA)
      STHETA = SIN(THETA)
      SALONG=0.
      SNORML=0.
      DO 60 I=1,NMOVIN
        J=IMOVIN(I)
        XX=REAL(PARTCL(J))-SPX
        YY=AIMAG(PARTCL(J))-SPY
        SALONG = SALONG + ABS(XX*CTHETA+YY*STHETA)
        SNORML = SNORML + ABS(YY*CTHETA-XX*STHETA)
60      CONTINUE
      SALONG = SALONG/NMOVIN
      SNORML = SNORML/NMOVIN
      ASPECT = SALONG/SNORML
      IF(ASPECT.LT.1.0)THETA = THETA + 0.5*PI
      IF(ASPECT.LT.1.0)ASPECT = SNORML/SALONG
      IF(THETA.GT.2*PI)THETA=THETA - 2*PI
      IF(ASPECT.LT.3.0)GOTO 80
```

```
      INDX1D = 1
      IF(THETA.GT.0.25*PI.AND.THETA.LT.0.75*PI)INDX1D=2
      IF(THETA.GT.1.25*PI.AND.THETA.LT.1.75*PI)INDX1D=2
80    DEG= THETA*180./PI
      IF(AVGRAD.LT.0.5*DX)INDX1D = INDX1D+3
      ANGLE = THETA
      IF(THETA.GT.270.)ANGLE = THETA - 360.
      IF(THETA.LE.270.0.AND.THETA.GE.90.0) ANGLE = THETA - 180.0
      RETURN
200   FORMAT(' Aspect Ratio=',F5.2/' Major Axis =',F8.5,' rad',
     $ 5X,F8.3,' deg',5X,' INDX1D =',I3,F9.2)
      END
```

```
      SUBROUTINE PLOTNU(IUT)
C
C  This subroutine plots oil concentrations as the no. of particles
C  in each grid of a grid system superimposed over the river-lake grid.
C  The grid size of the superimposed grid is equal to DXR. This is
C  the only subroutine requiring DXL to be exactly divisible by DXR.
C
C  -- Last Date of Revision : April 04, 1986
C
      COMPLEX SPCEN,PARTCL(1000)
      COMMON /VASB/IGRILB(300),IGRIUB(300),IGRLB1(300),IGRUB1(300)
      COMMON /ASB/SPCEN,PARTCL,NPTCL,NHITB,IHITB(1000),TYPBND(4,300)
      COMMON /SO/IMOVIN(1000),YSHIFT(1000),NMOVIN,SSHIFT
      COMMON /GRIDS/ DXL, DXR, LGRIDX, NGRIDX, IRGRID, BEGLK
      DIMENSION KOUNT(20,20)
      XXX = REAL(SPCEN)
      YYY = AIMAG(SPCEN)+SSHIFT/NMOVIN
C
C  Shifted spill center.
C
      SPX = XXX - BEGLK
      SPY = YYY
C
C  First set all array elements to print stars as output.
C
      DO 40 I=1,20
         DO 40 J=1,20,2
            KOUNT(I,J)=1001
            KOUNT(I,J+1)=1001
40    CONTINUE
      DD = DXL/DXR
      DDM1 = DD - 1
C
C  Calculate max and min boxes for the superimposed grid
C  using a shifted spill center.
C
      IMIN = (SPX/DXR+1) - 9
      IMAX = IMIN + 19
      JMIN = (SPY/DXR+1) - 9
      JMAX= JMIN + 19
C
C  Calculate the actual max and min positions of the
C  superimposed grid over the river-lake coordinate system.
C
      XMIN = (IMIN - 1)*DXR + BEGLK
      XMAX = XMIN + 10000.
      YMIN = JMIN*DXR
      YMAX = YMIN + 10000.
      DO 80 I=1,20
C
C  Determine 1.) if superimposed grid is in river or lake
C  2.) the appropriate x-grid box of actual grid and
C  3.) the range of boxes between the river boundaries.
C
         XLOC = (IMIN + I - 1)*DXR + BEGLK - DXR/2.0D0
```

```
         IF (XLOC .GT. 0.0D0) GOTO 41
         L = (XLOC - BEGLK)/DXL + 1
         GOTO 50
41       L = XLOC/DXR + LGRIDX + 1
50       IF (L .LT. 1) GOTO 80
         M1 = IGRILB(L)
         M2 = IGRIUB(L)
         IF (L .LE. LGRIDX) GOTO 55
         IM1 = M1 - JMIN - 1
         IM2 = M2 - JMIN + 1
         GOTO 56
55       IM1 = M1*DD - DDM1 - JMIN - 4
         IM2 = M2*DD - JMIN + 4
56       IF (IM1 .LT. 1) IM1=1
         IF (IM2 .GT. 20) IM2=20
C
C  Set array elements representing the combined
C  river and boundary boxes to zero.
C
         DO 70 J=IM1,IM2
            KOUNT(I,J)=0
70          CONTINUE
         IF(IGRLB1(L).EQ.0)GOTO 80
C
C  Set all boxes in superimposed grid representing islands to stars.
C
         M1 = IGRLB1(L)
         M2 = IGRUB1(L)
         IF (L .LE. LGRIDX) GOTO 72
         IM1 = M1 - JMIN + 1
         IM2 = M2 - JMIN - 1
         GOTO 73
72       IM1 = M1*DD - DDM1 - JMIN + 4
         IM2 = M2*DD - JMIN - 4
73       IF (IM1 .LT. 1) IM1=1
         IF (IM2 .GT. 20) IM2=20
         IF (IM1 .GT. IM2) GOTO 80
         DO 75 J=IM1,IM2
            KOUNT(I,J)=1001
75       CONTINUE
80    CONTINUE
C
C  Count the number of particles in the superimposed grid boxes.
C
      DO 450 I=1,NPTCL
         L = (REAL(PARTCL(I)) - BEGLK)/DXR + 2 - IMIN
         M = AIMAG(PARTCL(I))/DXR + 1 - JMIN
         IF (L .LT. 1 .OR. L .GT. 20) GOTO 450
         IF (M .LT. 1 .OR. M .GT. 20) GOTO 450
         KOUNT(L,M)=KOUNT(L,M)+1
450      CONTINUE
      WRITE(IUT,620) YMIN, YMAX
      WRITE(IUT,610) (KOUNT(1,M),M=1,20),XMIN
      DO 580 L=2,19
         WRITE(IUT,600) (KOUNT(L,M),M=1,20)
580      CONTINUE
```

```
      WRITE(IUT,610) (KOUNT(20,M),M=1,20),XMAX
      RETURN
600   FORMAT(1X,20I3)
610   FORMAT(1X,20I3,' - X =',F8.0,' ft')
620   FORMAT(/ 'Y =',F8.0,' ft',T47,F8.0,' ft = Y'/'   I',T60,'I')
      END
```

```
      SUBROUTINE PRELSE(SPILDT, IX, N1, N2, SPCENO, DIFFUL, DIFFUR)
C
C  This subroutine, to be used for continuous spills, releases
C  particles (No.s N1 to N2) at SPCENO.  Note that the number of
C  particles released in SPILDT is NPERDT. Therefore NPERDT=N2-N1+1
C  -- The release will be at equal intervals of time.
C  -- This version30 has a modified advection term
C
C  -- Last Date of Revision :  July 03 ,1986
C
      COMPLEX VCAR(8000),SPCEN,PARTCL(1000),VWIND,VDRIFT
      COMPLEX SPCENO,VDR1
      COMMON /VA/ VCAR,VWIND,VDRIFT
      COMMON /VASB/IGRILB(300),IGRIUB(300),IGRLB1(300),IGRUB1(300)
      COMMON /ASB/SPCEN,PARTCL,NPTCL,NHITB,IHITB(1000),TYPBND(4,300)
      COMMON /BLOCK7/SPGOIL,ANIU,SIGMA,AK2I,AK2V,AK2T,
     $ VOLPAR,VOLPIE(8),SLICKR(8)
      COMMON /ICE/ZWND(40,40),ZLKICE(20),NICEX1(20),NICEY1(20),
     $ NICEX2(20),NICEY2(20),IPOS1(20),IPOS2(20),AMIUO,ANICE,
     $ SPAICE,NICERG,LICERG
      COMMON /GRIDS/ DXL, DXR, LGRIDX, NGRIDX, IRGRID, BEGLK
C
C  Input : .. Location of spill center
C          .. Velocity distribution in river
C  Output: .. New location of each particle
C
      DATA PI /3.141592/
      IF(NICERG.EQ.0)GOTO 25
C
C  DELEQ  - Equilibrium thickness (ft)
C  UTH    - Threshold current speed for slick movement ( ft/sec)
C  UFAIL  - Failure velocity under rough ice cover ( ft/sec)
C  FRAMFA - FRiction AMplification FActor denoted by 'K' in text
C  AMIUO  - Viscosity of Oil in g/cm-sec
C
      DELEQ = (1.67 - 8.5*(1.0-SPGOIL))/30.48
      UTH = (305.79/(88.68-AMIUO))/30.48
      FRAMFA = 35.55*ANICE + 1.0
      IF(ANICE.GT.0.045)FRAMFA = 2.6
      TERM1=SQRT(SIGMA*(32.2)**2*62.4*(1.-SPGOIL))
      UFAIL=1.5*SQRT(2.*(1.+SPGOIL)*TERM1/(62.4*SPGOIL))
      ROUGH = (ANICE/0.034)**6
25    SPX = REAL(SPCENO)
        IF (SPX .LT. 0.0)THEN
          L = (SPX - BEGLK)/DXL
          M = AIMAG(SPCENO)/DXL
            ELSE
          L = SPX/DXR + LGRIDX
          M = AIMAG(SPCENO)/DXR
          ENDIF
        IPOS = 0
        IF(L.EQ.0)GOTO 117
        DO 115 L1=1,L
          IPOS = IPOS+IGRIUB(L1)-IGRILB(L1)+3
115       CONTINUE
117     IPOS = IPOS+M-IGRILB(L+1)+3
```

```
      IF(NICERG.EQ.0)GOTO 125
C
C  Determine whether the spill center is under ice or not
C
      ICOND=0
      DO 120 K=1,NICERG
        IF(IPOS.GE.IPOS1(K).AND.IPOS.LE.IPOS2(K))ICOND=1
        IF(ICOND.EQ.1)GOTO 180
120     CONTINUE
C
C  Advection velocity in free-surface conditions
C
125   VDRIFT = 0.03*VWIND + 1.1*VCAR(IPOS)
      GOTO 220
C
C  Advection Velocity under Ice
C
180   VDRIFT = (0.0,0.0)
      UWATER = CABS(VCAR(IPOS))
      IF(ROUGH.GT.DELEQ)GOTO 190
      IF(UWATER.LT.UTH)GOTO 220
      GOTO 200
190   IF(UWATER.LT.UFAIL)GOTO 220
200   VDRIFT = VCAR(IPOS)
      FDELTA = UWATER/SQRT((1.0-SPGOIL)*32.2*DELEQ)
      FK = SQRT(FRAMFA/(0.115*FDELTA**2 + 1.105))
      VDRIFT = VDRIFT*(1-FK)
220   VDR1 = VDRIFT
      DO 60 I=N1,N2
        DTPTCL = SPILDT*(I-N1+1)/(N2-N1+1)
        SUMDT = 0.
        IPASS = 1
        PARTCL(I) = SPCEN0
        VDRIFT = VDR1
40      IF(IPASS.EQ.1)GOTO 28
        IF(NHITB.EQ.0)GOTO 35
        DO 30 J=1,NHITB
        IF(I.EQ.IHITB(J))GOTO 60
30       CONTINUE
35    SPX = REAL(PARTCL(I))
        IF (SPX .LT. 0.0)THEN
          L = (SPX - BEGLK)/DXL
          M = AIMAG(PARTCL(I))/DXL
            ELSE
          L = SPX/DXR + LGRIDX
          M = AIMAG(PARTCL(I))/DXR
          ENDIF
      IPOS = 0
      IF(L.EQ.0)GOTO 517
      DO 515 L1=1,L
        IPOS = IPOS+IGRIUB(L1)-IGRILB(L1)+3
515     CONTINUE
517   IPOS = IPOS+M-IGRILB(L+1)+3
      IF(NICERG.EQ.0)GOTO 525
C
C  Determine whether the spill center is under ice or not
```

```
C
        ICOND=0
        DO 520 K=1,NICERG
          IF(IPOS.GE.IPOS1(K).AND.IPOS.LE.IPOS2(K))ICOND=1
          IF(ICOND.EQ.1)GOTO 580
520       CONTINUE
C
C  Advection velocity in free-surface conditions
C
525   VDRIFT = 0.03*VWIND + 1.1*VCAR(IPOS)
      GOTO 620
C
C  Advection Velocity under Ice
C
580     VDRIFT = (0.0,0.0)
      UWATER = CABS(VCAR(IPOS))
      IF(ROUGH.GT.DELEQ)GOTO 590
      IF(UWATER.LT.UTH)GOTO 620
      GOTO 600
590   IF(UWATER.LT.UFAIL)GOTO 620
600   VDRIFT = VCAR(IPOS)
      FDELTA = UWATER/SQRT((1.0-SPGOIL)*32.2*DELEQ)
      FK = SQRT(FRAMFA/(0.115*FDELTA**2 + 1.105))
      VDRIFT = VDRIFT*(1-FK)
C
620   CONTINUE
C
C  Add in turbulent fluctuation
C
28    VELUPV = ABS(REAL(VDRIFT)) + ABS(AIMAG(VDRIFT))
C
C   The next two statements prevent division by zero.  86400 is just a
C   large number in this case = no. secs in a day.
C
      DTSMAL =86400.
        IF(VELUPV.GT.0.01)THEN
        SPX=REAL(PARTCL(I))
        IF(SPX.GT.0.0)DTSMAL = DXR/VELUPV
        IF(SPX.LE.0.0)DTSMAL = DXL/VELUPV
        ENDIF
      IF((DTSMAL+SUMDT).GT.DTPTCL)DTSMAL = DTPTCL - SUMDT
      IPASS = IPASS + 1
      IF((SUMDT+DTSMAL).GE.DTPTCL)IPASS = 9999
      SUMDT = SUMDT + DTSMAL
      CALL RANDU(IX,IY,YFL)
      IX = IY
      ANG = PI*YFL
      CALL GAUSS(IX,1.0,0.0,VRAND)
      IF(SPX.LT.0.0) THEN
        TELAPS = SUMDT - DTSMAL/2.0
        IF(DIFFUL.LT.0.0)VPRIME=3.407E-03*TELAPS**0.67/SQRT(DTSMAL)
        IF(DIFFUL.GE.0.0) VPRIME = SQRT(4*DIFFUL/DTSMAL)
        ENDIF
      IF(SPX.GE.0.0)THEN
        IF(DIFFUR.LT.0.0) DDD = 2.88*CABS(VDRIFT)
        IF(DIFFUR.GE.0.0) DDD = 4*DIFFUD
```

```
         VRPIME =SQRT(DDD/DTSMAL)
         ENDIF
      VRAND = VPRIME*VRAND
      VX = VRAND*COS(ANG)
      VY = VRAND*SIN(ANG)
      VMAG = CABS(VDRIFT)
      VDRIFT = VDRIFT + CMPLX(VX,VY)
      PARTCL(I) = PARTCL(I) + DTSMAL*VDRIFT
C
C  Check for spill hitting the boundaries
C
      SPX = REAL(PARTCL(I))
      IF (SPX .GT. 0.0E0) GOTO 145
      L = (SPX - BEGLK)/DXL + 1
      M = AIMAG(PARTCL(I))/DXL + 1
      GOTO 155
145   L = SPX/DXR + 1 + LGRIDX
      M = AIMAG(PARTCL(I))/DXR +1
155   CONTINUE
      IF(M.GE.IGRILB(L).AND.M.LE.IGRIUB(L))GOTO 55
      NHITB = NHITB + 1
      IHITB(NHITB) = I
55    IF(IGRLB1(L).EQ.0)GOTO 59
      IF(M.LE.IGRUB1(L).AND.M.GE.IGRLB1(L))GOTO 58
      GOTO 59
58    NHITB = NHITB+1
      IHITB(NHITB) = I
59    CONTINUE
      IF(IPASS.NE.9999)GOTO 40
60    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PRINT1(IUT, NBRNCH)
C
C  This subroutine prints heading and river configuration data
C  -- IUT defines the unit No. to which the info will be written
C  -- Modifications for addition of lake were made.
C
C  -- Last Date of Revision :  October 29, 1985
C
      REAL *8 DATRUN(2),TIMRUN
      COMPLEX VSTRM(99,16),CORDV(99,16),VCAR(8000),CORDLB(99)
      COMPLEX SPCEN,PARTCL(1000),VWIND,VDRIFT
      COMMON /VA/ VCAR,VWIND,VDRIFT
      COMMON /VEL/VSTRM,CORDV,CORDLB,Q(30),WL(30),TICE(99,20),
     $ YWID(99,20),Z(99,20),ZD(99),NSLSCT(99),SCTANG(99),
     $ LCSTSQ(30),NSTUBE(99),NUMCON(99),NFIRCO(99),NSECO(99),KINTM
      COMMON /VASB/IGRILB(300),IGRIUB(300),IGRLB1(300),IGRUB1(300)
      COMMON /ASB/SPCEN,PARTCL,NPTCL,NHITB,IHITB(1000),TYPBND(4,300)
      COMMON /GRIDS/ DXL, DXR, LGRIDX, NGRIDX, IRGRID, BEGLK
C     CALL TSTIME(1,TIMRUN)
C     CALL TSDATE(1,DATRUN)
      WRITE(IUT,10)DATRUN,TIMRUN
      WRITE(IUT,20)NBRNCH,IRGRID,DXR,KINTM
      IS2=0
      DO 100 I=1,NBRNCH
         IS1=IS2+1
         IS2 = LCSTSQ(I)
         WRITE(IUT,30)I,IS1,IS2
100      CONTINUE
      WRITE(IUT,40)
      IS2 = IS2 +1
      DO 110 I=1,IS2
         J = NSLSCT(I)+1
         IWIDTH = YWID(I,J)
         WRITE(IUT,50)I,CORDLB(I),SCTANG(I),IWIDTH,ZD(I),NSTUBE(I),
     $   NUMCON(I),NFIRCO(I)
110      CONTINUE
      WRITE(IUT,70)
      DO 120 I=1,IS2
         IN=NSLSCT(I)+1
         WRITE(IUT,80)I,(YWID(I,J),Z(I,J),J=1,IN)
120      CONTINUE
      WRITE(IUT,60)
      DO 130 I=1,NGRIDX
         KNUM=2
         IF(IGRLB1(I).NE.0)KNUM=4
         WRITE(IUT,65)I,IGRILB(I),IGRIUB(I),IGRLB1(I),IGRUB1(I),
     $   (TYPBND(K,I),K=1,KNUM)
130      CONTINUE
10    FORMAT(1H1,///2X,75('*')/2X,75('*')/'  **',71X,'**'/'  **',12X,
     $ 'SIMULATION MODEL FOR OIL SPILLS IN RIVERS AND LAKES',
     $ 8X,'**'/'  **',71X,
     $ '**'/'  **',7X,'DEVELOPED AT - CIVIL & ENVIR. ENG. DEPT., ',
     $ 'CLARKSON UNIVERSITY',3X,'**'/'  **',7X,'SPONSERED BY - U.S. ARMY
     $ CORPS OF ENGINEERS, DETROIT DISTRICT',3X,'**'
```

```
     $ /'  **',71X,'**'/'  **',9X,'DATE AND TIME OF RUN : ',2A8,2X,A8
     $ ,13X,'**',/'  **',71X,'**'/2X,75('*')/2X,75('*'))
20    FORMAT(///'  GEOMETRIC PROPERTIES OF RIVER'/1H+,1X,29('_')//
     $ 5X,'NO. OF BRANCHES IN UNSTEADY FLOW MODEL -',I5/
     $ 5X,'NO. OF GRIDS IN X-DIRECTION OF RIVER    -',I5/
     $ 5X,'RIVER GRID SIZE IN ft.',17X,'-',F6.0/
     $ 5X,'NO. OF INTERPOLATIONS BETWN SECTIONS    -',I5//
     $ 5X,'SECTIONS IN EACH BRANCH'/1H+,4X,23('_')//
     $ 5X,'BRANCH SECTIONS INVOLVED'/15X,'FROM      TO')
30    FORMAT(3(7X,I2))
40    FORMAT(1H1,//11X,'INFORMATION ON RIVER SECTIONS'/1H+,10X,29('_'),
     $ //2X,'SECTION Lower bank intersection  Angle  Width  Ref datum ',
     $ ' No str  Cond.  Connect'/12X,
     $ 'X-CORD       Y-CORD        (rad)  (ft.)  for depth  tubes   No.',
     $ '     next 1st')
50    FORMAT(4X,I2,5X,F8.1,2X,F8.1,6X,F5.3,I7,F9.2,3I8)
60    FORMAT(1H1///10X,'GRID CONFIGURATION and BOUNDARY TYPES ',
     $ 'OF SCHEMATIZED RIVER AND LAKE'/1H+,9X   ,58('_')//
     $ '  X',15X,'Y GRID OF',17X,'REJECTION RATE PER TIME STEP'/
     $ ,' GRID',2('   Bank 1   Bank 2   Bank 3   Bank 4  '))
65    FORMAT(I4,2X,4(3X,I3,2X),5X,4(1X,F5.4,2X))
70    FORMAT(///,10X,'Geometry of X-Sections'/1H+,9X,22('_')//
     $ ' SCTN',10X,'Distance and Depth (ft.) in pairs of data')
80    FORMAT(/I3,1X,9(F6.0,':',F5.1,2X)/4X,9(F6.0,':',F5.1,2X))
      RETURN
      END
```

```
      SUBROUTINE SPRDAX(SPILDT,TIMET,INDPRN,SPAREA,SPLTIM,SPLRAT)
C
C This subroutine handles the Axi-symmetrical spreading of spill
C due to gravity, viscous and surface tension forces
C -- This version includes spreading under ice cover.
C -- Modifications for addition of lake were made.
C -- Last Date of Revision : October 29, 1985
C
      COMPLEX SPCEN,PARTCL(1000)
      COMMON /VASB/IGRILB(300),IGRIUB(300),IGRLB1(300),IGRUB1(300)
      COMMON /ASB/SPCEN,PARTCL,NPTCL,NHITB,IHITB(1000),TYPBND(4,300)
      COMMON /BLOCK7/SPGOIL,ANIU,SIGMA,AK2I,AK2V,AK2T,
     $ VOLPAR,VOLPIE(8),SLICKR(8)
      COMMON /SO/IMOVIN(1000),YSHIFT(1000),NMOVIN,SSHIFT
      COMMON /SE/FEVP1,FEVP2,CEVP,TOEVP
      COMMON /ICE/ZWND(40,40),ZLKICE(20),NICEX1(20),NICEY1(20),
     $ NICEX2(20),NICEY2(20),IPOS1(20),IPOS2(20),AMIUO,ANICE,
     $ SPAICE,NICERG,LICERG
      COMMON /GRIDS/ DXL, DXR, LGRIDX, NGRIDX, IRGRID, BEGLK
      COMMON /SPREAD/ RADIUS(1000),NTRACK(1000)
C
C The spill area is divided into 8 pie segments around the spill
C center.  Particles in each pie spread according to modified
C Fay's law's for axi-symmetrical case. (see text for details)
C Input : .. Spill center
C         .. Location of each particle
C         .. oil properties
C Output: .. New loaction of each particle
C
C Explanation of variables used only in this subroutine
C VOLPIE(I)  - an array containing the volume of oil in each pie
C              segment at previous time step NOTE: volume stored is
C              8 times the volume in pie
C RADIUS(I)  - distance to particles in pie from spillcenter. It is
C              assumed that no more than 500 particle are in a pie any time
C SPAREA     - Free surface area of spill (sq. ft)
C SPAICE     - Area of spill under ice (sq. ft)
C ICOND  = 0 - Oil in the pie has free surface conditions
C        = 1 - Oil in the pie is under ice
C
      DATA ROWAT, G /1.92, 32.2/
      PI = ATAN(1.) * 4.
C
C Evaluate some constants to be used in subsequent computations
C
      DELTA =  1.0 - SPGOIL
      AKINER = 0.25*AK2I*(DELTA*G)**0.25
      AKVISC = AK2V*(DELTA*G/SQRT(ANIU))**0.166
      AKSURF = 0.75*AK2T*(SIGMA**2/(ROWAT**2*ANIU))**0.25
      AKICE  = 0.0056666*((1-SPGOIL)*32.2*SPLRAT**2)**0.16666/ANICE
C
C Compute the mean radius for all moving particles
C
      TOTRAD=0.
      DO 7 I =1,NMOVIN
```

```
          J=IMOVIN(I)
          TOTRAD = TOTRAD+CABS(PARTCL(J)-SPCEN)
7         CONTINUE
        TOTRAD = TOTRAD/NMOVIN
        SPXCEN = REAL(SPCEN)
        SPYCEN = AIMAG(SPCEN)
        SPAREA = 0.0
         SPAICE = 0.0
C
C  Loop 500 is working for one pie at a time
C
        DO 500 IPIE=1,8
        ANG1 = (IPIE-1)*PI/4.
        ANG2 = ANG1 + PI/4.
        NPTPIE = 0
        NPTICE = 0
        ICOND  = 0
        DO 10 I=1,NPTCL
10      NTRACK(I)=0
C
C  Loop 20 is for finding the ID no's of particles belonging
C  to the pie. Radial dist. to particle from center is also computed
C  and stored in RADIUS(). NTRACK() stores the ID's of particles.
C
        DO 20 I=1,NMOVIN
          J=IMOVIN(I)
          ATX2 = REAL(PARTCL(J))-SPXCEN
          ATX1 = AIMAG(PARTCL(J))-SPYCEN
          ANG=ATAN2(ATX1,ATX2)
          IF(ANG.LT.0.0)ANG = ANG + 2.*PI
          ANGDEG = ANG*180./PI
          IF(ANG.LT.ANG1.OR.ANG.GE.ANG2)GOTO 20
          RAD = CABS(PARTCL(J)-SPCEN)
          IF(RAD.GT.2.20*TOTRAD)GOTO 20
          NPTPIE = NPTPIE+1
          RADIUS(NPTPIE) = RAD
          NTRACK(NPTPIE) = J
          IF(NICERG.EQ.0)GOTO 20
          SPX = REAL(PARTCL(J))
          IF (SPX .GT. 0.0E0) GOTO 143
          L = (SPX - BEGLK)/DXL
          M = (AIMAG(PARTCL(J))+YSHIFT(J))/DXL
          GOTO 153
143       L = SPX/DXR + LGRIDX
          M = (AIMAG(PARTCL(J))+YSHIFT(J))/DXR
153       CONTINUE
          IPOS = 0
          IF(L.EQ.0)GOTO 117
          DO 115 L1=1,L
           IPOS = IPOS+IGRIUB(L1)-IGRILB(L1)+3
115       CONTINUE
117       IPOS = IPOS+M-IGRILB(L+1)+3
          DO 120 K=1,NICERG
            IF(IPOS.GE.IPOS1(K).AND.IPOS.LE.IPOS2(K))NPTICE=NPTICE+1
120         CONTINUE
20        CONTINUE
```

```
C
C  NO PARTICLES- NO SPREADING
C
      IF(NPTPIE.LT.1)GOTO 500
      RMEAN=0.
      DO 40 I=1,NPTPIE
40      RMEAN=RMEAN+RADIUS(I)
      RMEAN =RMEAN/NPTPIE
C
C  Check if this pie should spread for free-surface or ice conditions.
C  If it is ice conditions, is the spilling still continuing.
C
C     WRITE(*,*)NPTICE,NPTPIE,RMEAN,ICOND
      IF(FLOAT(NPTICE)/FLOAT(NPTPIE).GT.0.5)ICOND=1
      IF(ICOND.EQ.1.AND.TIMET.GT.SPLTIM)GOTO 170
C
C  Determine the rate of spread at pie radius
C
      VOLNOW = VOLPAR*NPTPIE*8
      TIMBAR = TIMET - SPILDT/2.0
      VOLBAR =(VOLNOW+VOLPIE(IPIE))/2.0
      IF(ICOND.EQ.1)GOTO 47
      TVISC=(AK2V/AK2I)**4*(VOLBAR/(DELTA*G*ANIU))**0.333
      TERMIN=823.5*(ROWAT/SIGMA)**0.6666*SQRT(VOLBAR)*ANIU**0.3333
     $ /AK2T**1.3333
      IF(TIMBAR.GT.TERMIN)GOTO 500
      TSURFT = (AK2V/AK2T)**2*(DELTA*G*ANIU)**0.3333
     $ *(ROWAT/SIGMA)*VOLBAR**0.6666
      IF(TIMBAR.GT.TSURFT) GOTO 45
      DVDT = VOLNOW*(FEVP1-FEVP2)/SPILDT
      IF(TIMBAR.LE.TVISC) DRDT =
     $ AKINER*(DVDT+2*VOLBAR/TIMBAR)*SQRT(TIMBAR)/(VOLBAR**0.75)
      IF(TIMBAR.GT.TVISC)DRDT =
     $ AKVISC*(DVDT/3.+VOLBAR/(TIMBAR*4))*TIMBAR**0.25/VOLBAR**0.666
      GOTO 48
45    DRDT = AKSURF/(TIMBAR**0.25)
47    IF(ICOND.EQ.1)DRDT = AKICE/(TIMBAR**0.33333)
48    VOLPIE(IPIE) = VOLNOW
      SPRATE = DRDT*SPILDT/RMEAN
C
C  Rate of spreading at mean pie radius has been computed. Now spread
C  the particles in the pie proportionately.
C
      DO 140 I=1,NPTPIE
        J=NTRACK(I)
        RADOLD = CABS(PARTCL(J)-SPCEN)
        RADNEW = RADOLD*(SPRATE+1)
        IF(RADNEW.LT.0.0)RADNEW = 0.
        RADIUS(I) = RADNEW
        X = REAL(PARTCL(J)-SPCEN)
        Y = AIMAG(PARTCL(J)-SPCEN)
        X = X*RADNEW/RADOLD
        Y = Y*RADNEW/RADOLD
        PARTCL(J) = SPCEN + CMPLX(X,Y)
140     CONTINUE
      RMEAN=0.
```

```
      DO 160 I=1,NPTPIE
160      RMEAN=RMEAN+RADIUS(I)
      RMEAN =RMEAN/NPTPIE
170   SLICKR(IPIE) = RMEAN
      IF(ICOND.EQ.0)SPAREA = SPAREA + PI*RMEAN**2/8.
      IF(ICOND.EQ.1)SPAICE = SPAICE + PI*RMEAN**2/8.
      IF(INDPRN.EQ.1)WRITE(*,220)IPIE,NPTPIE,RMEAN
500   CONTINUE
C
C  Check for spill hitting the boundaries
C
      DO 60 I=1,NMOVIN
        J=IMOVIN(I)
        IF(YSHIFT(J).LT.DXR)GOTO 54
        PARTCL(J)=PARTCL(J)+CMPLX(0.,YSHIFT(J))
        SPX = REAL(PARTCL(J))
        IF (SPX .GT. 0.0E0) GOTO 144
        L = (SPX - BEGLK)/DXL + 1
        M = AIMAG(PARTCL(J))/DXL + 1
        DX = DXL
        GOTO 154
144     L = SPX/DXR + 1 + LGRIDX
        M = AIMAG(PARTCL(J))/DXR + 1
        DX = DXR
154     CONTINUE
        IF(M.GT.IGRUB1(L))GOTO 54
        X=REAL(PARTCL(J))
        Y=IGRUB1(L)*DX-0.25*DX
        PARTCL(J)=CMPLX(X,Y)
        NHITB= NHITB+1
        IHITB(NHITB)=J
        GOTO 60
54      SPX = REAL(PARTCL(J))
        IF (SPX .GT. 0.0E0) GOTO 145
        L = (SPX - BEGLK)/DXL + 1
        M = AIMAG(PARTCL(J))/DXL + 1
        GOTO 155
145     L = SPX/DXR + 1 + LGRIDX
        M = AIMAG(PARTCL(J))/DXR + 1
155     CONTINUE
        IF(M.GE.IGRILB(L).AND.M.LE.IGRIUB(L))GOTO 55
        NHITB = NHITB + 1
        IHITB(NHITB) = J
55      IF(IGRLB1(L).EQ.0)GOTO 60
        IF(M.LE.IGRUB1(L).AND.M.GE.IGRLB1(L))GOTO 58
        GOTO 60
58      NHITB = NHITB+1
        IHITB(NHITB) = J
60      CONTINUE
      RETURN
210   FORMAT(' WARNING * MAY CAUSE ERRORS PARTICLES IN PIE EXCEED 500')
220   FORMAT(I3,8X,I3,10X,F7.0)
      END
```

```
      SUBROUTINE SPRD1D(SPILDT, TIMET, INDPRN, SPAREA , ANGLE)
      COMPLEX SPCEN,PARTCL(1000), XYP(1000)
      COMMON /VASB/IGRILB(300),IGRIUB(300),IGRLB1(300),IGRUB1(300)
      COMMON /ASB/SPCEN,PARTCL,NPTCL,NHITB,IHITB(1000),TYPBND(4,300)
      COMMON /BLOCK7/SPGOIL,ANIU,SIGMA,AK2I,AK2V,AK2T,
     $ VOLPAR,VOLPIE(8),SLICKR(8)
      COMMON /BLOCK8/AKC10,AKC20,AKC30
      COMMON /SO/IMOVIN(1000),YSHIFT(1000),NMOVIN,SSHIFT
      COMMON /SE/FEVP1,FEVP2
      COMMON /ICE/ZWND(40,40),ZLKICE(20),NICEX1(20),NICEY1(20),
     $ NICEX2(20),NICEY2(20),IPOS1(20),IPOS2(20),AMIUO,ANICE,
     $ SPAICE,NICERG,LICERG
      COMMON /GRIDS/ DXL, DXR, LGRIDX, NGRIDX, IRGRID, BEGLK
      COMMON /SPREAD/ RADIUS(1000),NTRACK(1000)
      DIMENSION SPRATE(2),NPT(2),XLE(2),XSQ(2)
C
C This Subroutine handles one dimensional spreading in
C The spill area is divided into strips. Particles in each strip
C spreads according to spreading law  for one-dimensional case.
C (see text for details)
C Input : .. Spill center
C         .. Location of each particle
C         .. oil properties
C Output: .. New loaction of each particle
C
C Explanation of variables used only in this subroutine
C RADIUS(I) - distance to particles in a strip from strip center.
C             A maximum of 500 particles can be in a strip at any time
C IMOVIN(I) - Index in array PARTCL of moving particles
C             ex. 1,3,4,5,7,11,12,13...... etc.
C NMOVIN     - Number of Moving Particles
C SPAREA     - Free surface area of spill (sq. ft)
C SPAICE     - Area of spill under ice (sq. ft)
C ICOND = 0 - Oil in the strip has free surface conditions
C       = 1 - Oil in the strip is under ice
C XSQ        - sum of square of the distances from center line to mean edge
C                of the slick. The same variable later store the standard
C                deviationx2.
C -- Last Date of Revision : September 19, 1986
C
      DATA ROWAT, G , PI  /1.92, 32.2, 3.141592/
C
C Evaluate some constants to be used in subsequent computations
C
      DELTA = 1.0 - SPGOIL
      AKINER = AKC10*(G*DELTA/DXR)**0.3333
      AKVISC = AKC20*(G*DELTA)**0.25/(SQRT(DXR)*ANIU**0.125)
      AKSURF = AKC30*SQRT(SIGMA/ROWAT)/(ANIU**0.25)
      CC1 = 0.6666*AKC10*(G*DELTA)**0.3333
      CC2 = AKC20*(G*DELTA/SQRT(ANIU))**0.25
C
C To minimize some later computing, determine XP-grid boxes of
C extreme particles.
C
      COST = COS(ANGLE)
      SINT = SIN(ANGLE)
```

```
      LMAX=0
      LMIN=10000
      DO 40 I=1,NMOVIN
        J=IMOVIN(I)
        SPX = REAL(PARTCL(J)) - BEGLK
        SPY = AIMAG(PARTCL(J))
        XP = SPX*COST + SPY*SINT
        YP = SPY*COST - SPX*SINT
        XYP(J) = CMPLX(XP,YP)
        L = XP /DXR + 1
        IF(L.GT.LMAX)LMAX=L
        IF(L.LT.LMIN)LMIN=L
40      CONTINUE
      SPAREA = 0.0
      SPAICE = 0.0
C
C  Loop 500 : One strip at a time
C
      DO 500 ISTRIP=LMIN,LMAX
      YPBAR=0.
      NPTSTR = 0
      NPTICE = 0
      ICOND  = 0
      DO 50 I=1,NMOVIN
        J=IMOVIN(I)
        XP = REAL(XYP(J))
        L = XP/DXR + 1
        IF(ISTRIP.NE.L)GOTO 50
        NPTSTR = NPTSTR+1
        NTRACK(NPTSTR) = J
        YPBAR = YPBAR+AIMAG(XYP(J))
        IF(NICERG.EQ.0)GOTO 50
        SPX = REAL(PARTCL(J))
        IF (SPX .LE. 0.0) THEN
           L = (SPX - BEGLK)/DXL
           M = (AIMAG(PARTCL(J))+YSHIFT(J))/DXL
           ENDIF
        IF (SPX.GT.0.0) THEN
           L = SPX/DXR + LGRIDX
           M = (AIMAG(PARTCL(J))+YSHIFT(J))/DXR
           ENDIF
        IPOS = 0
        IF(L.EQ.0)GOTO 117
        DO 115 L1=1,L
          IPOS = IPOS+IGRIUB(L1)-IGRILB(L1)+3
115       CONTINUE
117     IPOS = IPOS+M-IGRILB(L+1)+3
        DO 120 K=1,NICERG
          IF(IPOS.GE.IPOS1(K).AND.IPOS.LE.IPOS2(K))NPTICE=NPTICE+1
120       CONTINUE
50      CONTINUE
C
C  Must have at least two particles in the strip for spreading
C
      IF(NPTSTR.LT.2)GOTO 500
      YPBAR=YPBAR/NPTSTR
```

```
      DO 60 I=1,NPTSTR
        J=NTRACK(I)
        RADIUS(I)= AIMAG(XYP(J))-YPBAR
60      CONTINUE
      IF(FLOAT(NPTICE)/FLOAT(NPTSTR).GT.0.5)ICOND=1
C
C  XLE are the distances to the spreading edge of slick in the strip
C  computed based on the mean distance to particles from strip center.
C  index=1 for + dir from YBAR and 2 for - dir from YBAR
C
      XLE(1) = 0.
      XLE(2) = 0.
      XSQ(1) = 0.
      XSQ(2) = 0.
      NPT(2) = 0
      DO 80 I=1,NPTSTR
        IF(RADIUS(I).GE.0.0) THEN
           XLE(1)=XLE(1)+RADIUS(I)
           XSQ(1) = XSQ(1) + RADIUS(I)*RADIUS(I)
        ELSE
           XLE(2)=XLE(2)+RADIUS(I)
           XSQ(2)=XSQ(2) + RADIUS(I)*RADIUS(I)
           NPT(2)=NPT(2)+1
        ENDIF
80      CONTINUE
      NPT(1) =NPTSTR-NPT(2)
      DO 85 K = 1,2
        XSQ(K)=(NPT(K)*XSQ(K)-XLE(K)**2)/(NPT(K)*(NPT(K)-1.))
        XSQ(K) = 2. * SQRT(XSQ(K))
        XLE(K) =XLE(K)/NPT(K)
85      CONTINUE
      IF(ICOND.EQ.1)GOTO 170
C
C  If slick thickness (STHICK) is less than ultimate thickness
C  for spreading (UTHICK), then no spreading
C  NOTE that XLE(2) is always negative
C

      STHICK = VOLPAR*(NPT(1)+NPT(2))/(DXR*(XLE(1)-XLE(2)))
      UTHICK = 1.3458E-5 * (VOLPAR*NMOVIN)**0.25
      IF(STHICK.LT.UTHICK)GOTO 500
C
C  Determine the rate of spread at mean radius (leading edge)
C
      DO 130 K=1,2
      VOLNOW = VOLPAR*NPT(K)
      TIMBAR = TIMET - SPILDT/2.0
      VOLBAR=VOLNOW
      DLDT = SQRT(VOLNOW/DXR)*(SQRT(1-FEVP2)-SQRT(1-FEVP1))/SPILDT
      VOLPDX = VOLNOW/DXR
C
C  TVISC  - Time in secs for transition from Inertia to Viscous
C  TSURFT - Time in secs for transition from Viscous to Surf Tension
C  TERMIN - Time in secs at spreading termiation
C  DRDT   - Spreading rate at leading edge (ft/sec)
C
```

```
      TSURFT = (AKVISC/AKSURF)**2.6666*VOLBAR**1.3333
      TVISC=(AKVISC/AKINER)**3.4285*VOLBAR**0.5714
      IF(TIMBAR.LE.TVISC) DRDT =
     $ CC1*(TIMBAR**0.6666*DLDT/VOLPDX**.1666 + (VOLPDX/TIMBAR)**0.3333)
      IF(TIMBAR.GT.TVISC.AND.TIMBAR.LE.TSURFT)DRDT =
     $ CC2*(.375*SQRT(VOLPDX)*TIMBAR**(-0.675) + DLDT*TIMBAR**0.375)
      IF(TIMBAR.GT.TSURFT)DRDT = 0.75*AKSURF/(TIMBAR**0.25)
      SPRATE(K) = DRDT*SPILDT/ABS(XLE(K))
      IF(SPRATE(K).LT.-1.0)SPRATE(K)=-1.0
130   CONTINUE
C
C  Spreading rates for mean leading edges on either side has been
C  computed. Now spread the particles proportiontely.
C
      DO 140 I=1,NPTSTR
        J=NTRACK(I)
        IF(RADIUS(I).GE.0.0.AND.ABS(RADIUS(I)).LT.XSQ(1))
     $      YPNEW=RADIUS(I)*(SPRATE(1)+1)
        IF(RADIUS(I).LT.0.0.AND.ABS(RADIUS(I)).LT.XSQ(2))
     $      YPNEW=RADIUS(I)*(SPRATE(2)+1)
        RADIUS(I) = YPNEW
        XP = REAL(XYP(J))
        YP = YPBAR + YPNEW
        XYP(J) = CMPLX(XP,YP)
        X = XP*COST - YP*SINT + BEGLK
        Y = YP*COST + XP*SINT
        PARTCL(J) = CMPLX(X,Y)
140     CONTINUE
C
C  Compute the mean distances to leading edges after spreading.
C
      XLE(1)=0.
      XLE(2)=0.
      DO 160 I=1,NPTSTR
        IF(RADIUS(I).GE.0.0)XLE(1)=XLE(1)+RADIUS(I)
        IF(RADIUS(I).LT.0.0)XLE(2)=XLE(2)+RADIUS(I)
160     CONTINUE
      XLE(1) =XLE(1)/NPT(1)
      XLE(2) =XLE(2)/NPT(2)
      SPAREA = SPAREA + DXR*(XLE(1)+ ABS(XLE(2)))
170   IF(INDPRN.EQ.1)WRITE(*,220)ISTRIP,NPTSTR,XLE(2),YBAR,XLE(1)
      IF(ICOND.EQ.1) SPAICE = SPAICE + DXR*(XLE(1)+ ABS(XLE(2)))
      IF(ICOND.EQ.1)WRITE(*,230)
500   CONTINUE
C
C  Move the particles back into the upper channel which were
C  shifted by ORIENT routine.  Also check for the particles hitting
C  the boundaries
C
      DO 460 I=1,NMOVIN
        J=IMOVIN(I)
        IF(YSHIFT(J).LT.DXR)GOTO 54
        PARTCL(J)=PARTCL(J)+CMPLX(0.,YSHIFT(J))
        SPX = REAL(PARTCL(J))
        IF (SPX .GT. 0.0E0) GOTO 145
        L = (SPX - BEGLK)/DXL + 1
```

```
          M = AIMAG(PARTCL(J))/DXL + 1
          DX = DXL
          GOTO 155
145       L = SPX/DXR + LGRIDX + 1
          M = AIMAG(PARTCL(J))/DXR + 1
          DX = DXR
155       CONTINUE
C
C  Check for spill hitting the boundaries
C
          IF(M.GT.IGRUB1(L))GOTO 54
          NHITB=NHITB+1
          IHITB(NHITB)=J
          X=REAL(PARTCL(J))
          Y=IGRUB1(L)*DX-0.25*DX
          PARTCL(J)=CMPLX(X,Y)
          GOTO 460
54        SPX = REAL(PARTCL(J))
          IF (SPX .GT. 0.0E0) GOTO 146
          L = (SPX - BEGLK)/DXL + 1
          M = AIMAG(PARTCL(J))/DXL + 1
          DX = DXL
          GOTO 156
146       L = SPX/DXR + LGRIDX + 1
          M = AIMAG(PARTCL(J))/DXR + 1
          DX = DXR
156       CONTINUE
          IF(M.GE.IGRILB(L).AND.M.LE.IGRIUB(L))GOTO 55
          NHITB = NHITB + 1
          IHITB(NHITB) = J
          GOTO 460
55        IF(IGRLB1(L).EQ.0)GOTO 460
          IF(M.LE.IGRUB1(L).AND.M.GE.IGRLB1(L))GOTO 58
          GOTO 460
58        NHITB = NHITB+1
          IHITB(NHITB) = J
460       CONTINUE
        RETURN
220     FORMAT(I4,7X,I3,5X,F7.0,F7.0,F7.0)
230     FORMAT(1H+,50X,' ICE')
        END
```

```
      SUBROUTINE VELDIS(IPROPT, NBRNCH, ILVCAR)
C
C  This program computes the Velocity along and across the river
C     (Two-Dimensional velocity distribution)
C  -- Modifications for addition of lake were made.
C
C  -- Last Date of Revision: October 29, 1985
C
      COMPLEX COMPXY,VSTRM(99,16),CORDV(99,16),VCAR(8000),CORDLB(99)
      COMPLEX VWIND,VDRIFT
      COMMON /VEL/VSTRM,CORDV,CORDLB,Q(30),WL(30),TICE(99,20),
     $ YWID(99,20),Z(99,20),ZD(99),NSLSCT(99),SCTANG(99),
     $ LCSTSQ(30),NSTUBE(99),NUMCON(99),NFIRCO(99),NSECO(99),KINTM
      COMMON /VA/ VCAR,VWIND,VDRIFT
      COMMON /VASB/IGRILB(300),IGRIUB(300),IGRLB1(300),IGRUB1(300)
      COMMON /V/IZRBX(100),IZRBY(100),NZRVB
      COMMON /GRIDS/ DXL, DXR, LGRIDX, NGRIDX, IRGRID, BEGLK
C
C   Input  : Q & WL both of which are arrays of size at least NBRNCH
C            and NBRNCH+1 respec.  ILVCAR and BEGLK are used as indices
C            for determining where to write velocities in VCAR and if
C            grid boxes are in river or lake.
C
C   Output : x & y components of velocity in the river for each grid box
C            Also computed are velocities at sections perpendicular to
C            stream thalweg and co-ordinates of the position at which
C            they are acting
C
C   This program computes velocities in the following manner:
C    1) Go from branch to branch - (Branch here refers to branches in
C       Unsteady flow model
C    2) Then do for each section in a branch
C    3) Finally scans across the river, streamtube by streamtube
C       The above numbers also show the looping sequence where 1 is the
C       outermost and 3 is the innermost
C
      IS2=0
      DO 80 IB=1,NBRNCH
       IS1=IS2+1
       IS2=LCSTSQ(IB)
       TBRLEN=0.
       DO 30 IS=IS1,IS2
        ISCON = NFIRCO(IS)
        TBRLEN=TBRLEN+CABS(CORDLB(IS)-CORDLB(ISCON))
30      CONTINUE
       IF(IB.EQ.NBRNCH)IS2=IS2+1
       SCTLEN=0.
       DO 80 IS=IS1,IS2
        QSTUBE=Q(IB)/NSTUBE(IS)
        ATUBE1=0.
        YSTB1 = 0.
        ISCON = NFIRCO(IS)
        SCTLEN=SCTLEN+CABS(CORDLB(IS)-CORDLB(ISCON))
        IBCON = IB+1
        IF(NUMCON(IS2).NE.21)GOTO 38
        IF(NSECO(IS2).EQ.999)GOTO 38
```

```
36      IBCON = IBCON+1
        LASTSC = LCSTSQ(IBCON-1)
        IF(NUMCON(LASTSC).NE.21)GOTO 36
38      IF(IS2.EQ.NFIRCO(IS2-1).AND.(IS2-1).EQ.NFIRCO(IS2))IBCON=IB
        WLSCT=WL(IB)-(WL(IB)-WL(IBCON))*SCTLEN/TBRLEN
        SARIY=0.
        NIY=NSLSCT(IS)+1
        SXAREA = 0.
        DO 40 IY=2,NIY
         DYRS=YWID(IS,IY)-YWID(IS,IY-1)
         PERI=SQRT(DYRS**2 + (Z(IS,IY)-Z(IS,IY-1))**2)
         ICEIND=0
         IF(TICE(IS,IY).GT.0.001.AND.TICE(IS,IY-1).GT.0.001)ICEIND=1
         TISUM = TICE(IS,IY)+TICE(IS,IY-1)
         IF(ICEIND.EQ.1)PERI=PERI+DYRS
         IF(ICEIND.EQ.0)TISUM=0.0
         AIY=DYRS*((Z(IS,IY)+Z(IS,IY-1)-TISUM)/2.+WLSCT-ZD(IS))
         IF(AIY.LT.0.0)AIY = 0.0
         HR=AIY/PERI
         SARIY=SARIY+AIY*HR**0.6666
         SXAREA = SXAREA + AIY
40       CONTINUE
        NSTUB1 = NSTUBE(IS)-1
        DO 70 ITB=1,NSTUB1
         QSET=QSTUBE*ITB
         PSARIY=0.
         SPERI =0.
         SAIY  =0.
         DO 60 IY=2,NIY
          DYRS=YWID(IS,IY)-YWID(IS,IY-1)
          PERI=SQRT(DYRS**2 + (Z(IS,IY)-Z(IS,IY-1))**2)
          ICEIND=0
          IF(TICE(IS,IY).GT.0.001.AND.TICE(IS,IY-1).GT.0.001)ICEIND=1
          TISUM = TICE(IS,IY)+TICE(IS,IY-1)
          IF(ICEIND.EQ.1)PERI=PERI+DYRS
          IF(ICEIND.EQ.0)TISUM=0.0
          AIY =DYRS*((Z(IS,IY)+Z(IS,IY-1)-TISUM)/2. + WLSCT - ZD(IS))
          IF(AIY.LT.0.0)AIY = 0.0
          HR   = AIY/PERI
          ARIY=AIY*HR**0.6666
          PSARIY= PSARIY + ARIY
          SPERI = SPERI + PERI
          SAIY  = SAIY + AIY
          QIY   = Q(IB)*PSARIY/SARIY
          IF(QIY.LT.QSET)GOTO 60
          QIY1 = Q(IB)*(PSARIY-ARIY)/SARIY
          YSTB2 = YWID(IS,IY-1)+DYRS*(QSET-QIY1)/(QIY-QIY1)
          YSTB = (YSTB1+YSTB2)/2.
          YSTB1 = YSTB2
          ATUBE = SAIY-AIY+AIY*(YSTB2-YWID(IS,IY-1))/DYRS
          VSTRM(IS,ITB) = CMPLX(QSTUBE/(ATUBE-ATUBE1),0.)
          ATUBE1 = ATUBE
          ANGL= SCTANG(IS)
          CORDV(IS,ITB)=CORDLB(IS)+CMPLX(YSTB*COS(ANGL),YSTB*SIN(ANGL))
          GOTO 65
60        CONTINUE
```

```
65         CONTINUE
70        CONTINUE
          NSTB=NSTUBE(IS)
          VSTRM(IS,NSTB)=CMPLX(QSTUBE/(SXAREA-ATUBE1),0.)
          YSTB = (YWID(IS,NIY)+YSTB1)/2.
          CORDV(IS,NSTB)=CORDLB(IS)+CMPLX(YSTB*COS(ANGL),YSTB*SIN(ANGL))
80        CONTINUE
C
C   At this point 2-D stream velocity (Along the river section by section
C   and across the river streamtube by streamtube) is assigned to
C   VSTRM's x-component. Therefore it has the correct magnitude but not
C   the correct direction. Later this magnitude will be correctly
C   distributed into x & y components with the correct direction.
C   CORDV stores the location at which VSTRM is acting
C   NOTE : CORDV and VSTRM are both 2-D COMPLEX arrays
C
C   Now assign the correct direction to velocities
C
        IS2=LCSTSQ(NBRNCH)
        DO 100 IS=1,IS2
          NSTB = NSTUBE(IS)
          DO 100 ITB=1,NSTB
            NFIRST=NFIRCO(IS)
            ISCON =NFIRST
            IF(ITB.GT.NSTUBE(NFIRST))ISCON=NSECO(IS)
            ITBCON=ITB
            IF(NUMCON(IS).EQ.11)GOTO 97
            IF(NUMCON(IS).EQ.12.AND.ITB.GT.NSTUBE(NFIRST))ITBCON=
     $      ITB-NSTUBE(NFIRST)
            IF(NUMCON(IS).NE.21)GOTO 97
            IF(NSECO(IS).NE.999)GOTO 97
            DO 93 I =1,999
              J = IS -I
              IF(NSECO(J).NE.0)GOTO 95
93            CONTINUE
95          ITBCON = ITB + NSTUBE(J-1)
97          VMAG=REAL(VSTRM(IS,ITB))
            COMPXY = CORDV(ISCON,ITBCON)-CORDV(IS,ITB)
            RAD = CABS(COMPXY)
            VVX = VMAG*REAL(COMPXY)/RAD
            VVY = VMAG*AIMAG(COMPXY)/RAD
            VSTRM(IS,ITB) = CMPLX(VVX,VVY)
100       CONTINUE
C
C   The next segment writes velocities and co-ords to a file if IPROPT=1
C   This information can be used by program DIRPLOT to plot velocities
C
        IF(IPROPT.EQ.0)GOTO 415
        DO 110 IS=1,IS2
          NSTB = NSTUBE(IS)
          DO 110 ITB=1,NSTB
            WRITE(3,2100)CORDV(IS,ITB),VSTRM(IS,ITB)
110         CONTINUE
C
C   From the velocities computed at stream cross sections, now assign the
C   velocities to each grid center in the Cartesian System.
```

```
C   First assign the velocity to a grid box if co-ords are within the box.
C
415   DO 120 IS =1,IS2
        NSTB = NSTUBE(IS)
        DO 120 ITB = 1,NSTB
          L = REAL(CORDV(IS,ITB))/DXR
          M =AIMAG(CORDV(IS,ITB))/DXR
          IPOS = ILVCAR
          IF(L.EQ.0)GOTO 117
          DO 115 L1=1,L
            IPOS = IPOS+IGRIUB(L1+LGRIDX)-IGRILB(L1+LGRIDX)+3
115         CONTINUE
117       IPOS = IPOS+M-IGRILB(L+LGRIDX+1)+3
          VMAG = CABS(VCAR(IPOS))
          IF(VMAG.LE.0.001)VCAR(IPOS) = VSTRM(IS,ITB)
          IF(VMAG.GT.0.001)VCAR(IPOS) = (VCAR(IPOS)+VSTRM(IS,ITB))/2.
120   CONTINUE
C
C   Now check for the boxes with no assigned velocity yet;
C   For KINTM intermediate Sections interpolate in streamtube between
C   Two adjacent X-scetions and assign a weighted mean velocity
C
      DO 130 IS=1,IS2
        NSTB = NSTUBE(IS)
        DO 130 ITB = 1,NSTB
          NFIRST=NFIRCO(IS)
          ISCON =NFIRST
          IF(ITB.GT.NSTUBE(NFIRST))ISCON=NSECO(IS)
          ITBCON=ITB
          IF(NUMCON(IS).EQ.11)GOTO 197
          IF(NUMCON(IS).EQ.12.AND.ITB.GT.NSTUBE(NFIRST))ITBCON=
     $    ITB-NSTUBE(NFIRST)
          IF(NUMCON(IS).NE.21)GOTO 197
          IF(NSECO(IS).NE.999)GOTO 197
          DO 193 I =1,999
            J = IS -I
            IF(NSECO(J).NE.0)GOTO 195
193         CONTINUE
195       ITBCON = ITB + NSTUBE(J-1)
197       CONTINUE
        DO 130 K=1,KINTM
          COMPXY= ((KINTM+1-K)*CORDV(IS,ITB)+K*CORDV(ISCON,ITBCON))
     $    /(KINTM+1)
          L = REAL(COMPXY)/DXR
          M =AIMAG(COMPXY)/DXR
          IPOS = ILVCAR
          IF(L.EQ.0)GOTO 127
          DO 125 L1=1,L
            IPOS = IPOS+IGRIUB(L1+LGRIDX)-IGRILB(L1+LGRIDX)+3
125         CONTINUE
127       IPOS = IPOS+M-IGRILB(L+LGRIDX+1)+3
          VMAG = CABS(VCAR(IPOS))
          IF(VMAG.LE.0.001)VCAR(IPOS)=
     $    ((KINTM+1-K)*VSTRM(IS,ITB)+K*VSTRM(ISCON,ITBCON))/(KINTM+1)
130     CONTINUE
C
```

```
C   There may still be boxes without any assigned velocities
C   Now velocities will be assigned based on the average value of the
C   surrounding boxes
C   Start from column LGRIDX+2 and then move to subsequent ones. The first
C   column is neglected. Before the process begins a value of 0.0011 is
C   assigned to the grids just outside the boundary (a technique used
C   purely to simplify computations).
C
      IY1=1
      DO 133 I=1,NGRIDX
        IY2=IGRIUB(I)-IGRILB(I)+2+IY1
        VCAR(IY1)=0.0011
        VCAR(IY2)=0.0011
        IY1 = IY2+1
133     CONTINUE
      IY2 = ILVCAR - 1
      LP1 = LGRIDX + 1
      DO 150 L=LP1,NGRIDX
        IY1 = IY2+3
        IY2 = IGRIUB(L) - IGRILB(L)+IY1
        DO 150 M = IY1,IY2
          COMPXY = (0.,0.)
          COUNT  = 0.
          IROW = M-IY1+IGRILB(L)
          IF(IGRLB1(L).EQ.0)GOTO 141
          IF(IROW.GE.IGRLB1(L).AND.IROW.LE.IGRUB1(L))VCAR(M)=0.0011
141       IF(CABS(VCAR(M)).GT.0.001)GOTO 150
C
C  If first column in river, don't use lake grids to left
C
          IF(L .EQ. 1) GOTO 142
          IF((IGRILB(L-1)-IROW) .GT. 2) GOTO 142
          IF((IROW-IGRIUB(L-1)) .GT. 2) GOTO 142
          MM = M+IGRILB(L)-IGRIUB(L-1) - 3
          IF(CABS(VCAR(MM)).LE.0.001)GOTO 142
          COMPXY=COMPXY+VCAR(MM)
          COUNT = COUNT+1
142       MM = M+IGRIUB(L)-IGRILB(L+1)+3
          IF((IGRILB(L+1)-IROW) .GT. 2) GOTO 144
          IF((IROW-IGRIUB(L+1)) .GT. 2) GOTO 144
          IF(CABS(VCAR(MM)).LE.0.001)GOTO 144
          COMPXY=COMPXY+VCAR(MM)
          COUNT = COUNT+1
144       IF(CABS(VCAR(M-1)).LE.0.001)GOTO 146
          COMPXY=COMPXY+VCAR(M-1)
          COUNT = COUNT+1
146       IF(CABS(VCAR(M+1)).LE.0.001)GOTO 148
          COMPXY=COMPXY+VCAR(M+1)
          COUNT = COUNT+1
148       VCAR(M)=COMPXY/COUNT
150     CONTINUE
C
C   For the boxes defined thru input data, set VCAR=0.0
C
      DO 164 IBOX=1,NZRVB
        IF(NZRVB.GT.100)GOTO 164
```

```
          L = IZRBX(IBOX) - 1
          M = IZRBY(IBOX) - 1
          IPOS = 0
          IF(L.EQ.0)GOTO 163
          DO 160 L1=1,L
           IPOS = IPOS+IGRIUB(L1)-IGRILB(L1)+3
160       CONTINUE
163       IPOS = IPOS+M-IGRILB(L+1)+3
          VCAR(IPOS) = 0.0
164     CONTINUE
        IF(IPROPT.EQ.0)RETURN
C
C    Write the river grid velocities to a file.
C
        J1=ILVCAR+2
        IRGRID = NGRIDX-LGRIDX
        DO 170 I=1,IRGRID
        X = I*DXR - 0.5*DXR
        II = I+LGRIDX
        J2 = IGRIUB(II) - IGRILB(II)+J1
          DO 165 J=J1,J2
          Y = (IGRILB(II)+J-J1)*DXR-0.5*DXR
          WRITE(4,2100)X,Y,VCAR(J)
165     CONTINUE
        J1=J2+3
170     CONTINUE
        RETURN
123     FORMAT(3I5,3F8.2,2F10.0)
2100    FORMAT(2F9.0,2F7.2)
        END
```

Subroutines GAUSS and RANDU

```
      SUBROUTINE GAUSS(IX,S,AM,V)
      A=0.0
      DO 50 I=1,12
      CALL RANDU(IX,IY,Y)
      IX=IY
50    A=A+Y
      V=(A-6.0)*S+AM
      RETURN
      END


      SUBROUTINE RANDU(IX,IY,YFL)
      IY = IX*65539
      IF(IY)5,6,6
5       IY = IY + 2147483647+1
6       YFL = IY
      YFL = YFL*0.4656613E-9
      YFL=RND(-1)
      RETURN
      END
```

```
      SUBROUTINE INIT(S, IDIM, TLKQ, RLKQ, INDPRN)
C
C     This subroutine reads in a reference stream function file
C     supplied in (LAKEINIT.PSI) logical unit 14.  If another discharge
C     besides the reference discharge is required, appropriate adjustments
C     to the stream function values are made.  These values are only
C     used as a first best guess.
C
C     Last Date of Revision : September 10, 1985
C
C ***********************************************************************
      DIMENSION S(40,40)
      COMMON /GPARM/ RPARM(23), IPARM(54), ZPARM(2)
      DATA SPVAL, IFIRST /1.0E20, 0/
      IF (IFIRST .NE. 0) GOTO 55
      IFIRST = 1
C
C  FILE 14 LAKEINIT.PSI OPENED IN OILEX18
C
      REWIND 14
      IM = IPARM(1)
      JM = IPARM(2)
      READ(14,30,END=10) ((S(I,J),I=1,IM),J=1,JM)
C
C  CONVERT CFS TO CMS
C
      DO 50 I=1,IM
      DO 50 J=1,JM
      IF(S(I,J) .EQ. SPVAL) GOTO 50
      S(I,J) = S(I,J)/35.3198
   50 CONTINUE
C
C  UPDATE STREAM FUNCTION TO CURRENT VALUES
C
      GOTO 65
   55 REWIND 15
      READ(15,35,END=10) ((S(I,J),I=1,IM),J=1,JM)
   65 IF (TLKQ .EQ. RLKQ) RETURN
      ADDQ = (TLKQ-RLKQ)/(2.0*35.3198)
      DO 40 I=1,IM
         DO 40 J=1,JM
         IF (S(I,J) .EQ. SPVAL) GOTO 40
         IF (S(I,J) .GT. 0.0D0) S(I,J) = S(I,J)+ADDQ
         IF (S(I,J) .LT. 0.0D0) S(I,J) = S(I,J)-ADDQ
   40 CONTINUE
C
C  PRINT CURRENT STREAM FUNCTION VALUES (CMS)
C
   45 IF(INDPRN .EQ. 1) WRITE(*,60)
      IF(INDPRN .EQ. 1) CALL PRNT(6, S, IDIM, IM, JM, SPVAL)
      RLKQ = TLKQ
      RETURN
C
C  NO INITIAL CONDITION FILE, SET STREAMFUNCTION TO ZERO
C
```

```
10 CONTINUE
   DO 20 I = 1, IM
      DO 20 J = 1, JM
20       S(I,J) = 0.
30 FORMAT(6E12.5)
35 FORMAT(E12.5)
60 FORMAT(/'INITIAL STREAM FUNCTION FILE FOR LAKE'/)
   RETURN
   END
```

```
      SUBROUTINE OUTP(TIME, TTS, IDIM)
C
C     This subroutine writes stream function values to file (LAKETEMP.PSI)
C     logical unit 15 for later use when determining velocities in lake grid
C     boxes.  Only values calculated after an initial period of 20 hours are
C     written to the file.  After that, values are written according
C     to the update interval (UFDT) of the unsteady river model.
C
C     Last Date of Revision : September 24, 1985
C
C ******************************************************************
      COMMON /LKMD/ D(40,40), S(40,40), U(40,40), V(40,40)
      COMMON /GPARM/ RPARM(23), IPARM(54), ZPARM(2)
C
C  Run for 20 hours to establish steady state.
C
      IF (TIME .NE. TTS*3600.) RETURN
       REWIND 15
      IM=IPARM(1)
      JM=IPARM(2)
C
C  Write stream function field for time 0 and then
C  write stream function field for UFDT amount of time .
C
   10 WRITE(15,20) ((S(I,J),I=1,IM),J=1,JM)
   20 FORMAT(E12.5)
      RETURN
      END
```

```
      SUBROUTINE PARTIC(CLWL, ILVCAR, IOPT2)
C     PROGRAM PARTIC
C
C  To avoid general confusion, the documentation supplied by GLERL
C  has been modified from the original version contained in the
C  PATHFINDER Particle Trajectory Program.  The modifications
C  merely remove unnecessary comments and definitions and add any
C  changes to the program.  These include:
C   1.) The March 1983 comment by D. J. Schwab was removed.
C   2.) The control parameters and bathymetric data is read in from
C         logical unit 13 but the meteorlogical data is read from
C         logical unit 10.  The data is initially in English (fps)
C         units but is converted into Metric (mks) units for use in
C         calculating the lake circulation. However, the final lake
C         velocities are reconverted into English units.
C         Currently, the wind information is not used in PARTIC.
C   3.) Stream function values are read from a temporary file via
C         logical unit 15.
C   4.) Row IM and column JM are set to SPVAL so that stream functions,
C         depths, and velocities will print correctly using subroutine PRNT.
C   5.) Velocities for grid boxes are stored in array VCAR()
C         and are written to a file (for plotting purposes) if IOPT2=1.
C         velocities are computed using a 4-point average of four
C         surrounding grid corner velocities and then assigned to the
C         grid center.
C   6.) Common Blocks GRIDS & LKMD were added.
C   7.) Subroutines UPPART & POUTP were not used.
C
C  Last Date of Revision : September 24, 1985
C
C****************************PROGRAM PARTIC**************************
C PURPOSE :
C                   THE PURPOSE OF THIS PROGRAM I[illegible]   HELP IN SIMULATING THE
C                   MOVEMENT OF TRACER PARTIC[illegible]    A GRID MODEL OF A LAKE.
C                   THE LAKE IS REPRESENTED AS AN ARRAY OF SQUARE GRID
C                   BOXES.  THE USER MUST SUPPLY A DESCRIPTION OF THE LAKE
C                   BATHYMETRY, THE WIND FIELD, THE VOLUME TRANSPORTS BETWEEN
C                   BOXES, CONTROL PARAMETERS (TIME STEP, DURATION OF RUN,
C                   WATER LEVEL INCREMENT, AND HEIGHT OF ANEMOMETER), AND
C                   TRACER PARTICLE INITIAL POSITIONS.  VELOCITIES IN EACH
C                   GRID BOX WILL ULTIMATELY BE CALCULATED FOR USE IN THE
C                   OIL SPILL SIMULATION.
C
C  INPUT :
C
C      NOTE: CLWL SUPPLIED THROUGH SMOSIR AND NDCONV FROM 1-D MODEL
C              CLWL- CURRENT LAKE WATER LEVEL (FT)
C
C   LOGICAL UNIT 13 : LAKEBATH.DAT
C     RECORD 1 - CONTROL PARAMETER RECORD :
C                                                   FORMAT   CARD COLUMNS
C              DT   - TIME STEP IN HOURS             G8.2       9 - 16
C
C     BATHYMETRIC DATA FILE :
C                 THE FORMAT OF THE BATHYMETRIC DATA FILE IS DESCRIBED
```

```
C               IN SCHWAB AND SELLERS (1980): 'COMPUTERIZED BATHY-
C               METRY AND SHORELINES OF THE GREAT LAKES', NOAA DATA
C               REPORT ERL-GLERL-16, AS DESCRIBED IN SCHWAB, BENNETT,
C               AND JESSUP (1981) : 'A TWO-DIMENSIONAL LAKE CIRCULATION
C               MODELING SYSTEM', NOAA TECHNICAL MEMORANDUM ERL-GLERL-38.
C               FIVE ADDITIONAL FIELDS ARE PRESENT ON BATHYMETRIC
C               DATA HEADER RECORD NUMBER 2.  THESE ARE:
C                                                  FORMAT  CARD COLUMNS
C               MINIMUM DEPTH (FT)                  I5       45-49
C               BASE GRID ROTATION FROM E-W         F6.2     50-55
C               I DISPLACEMENT                      F7.3     56-62
C               J DISPLACEMENT                      F7.3     63-69
C               ROTATION ANGLE  FROM BASE           F7.2     70-76
C                (ANGLES MEASURED IN DEGREES COUNTERCLOCKWISE)
C
C               THE ADDITIONAL FIELDS ARE REQUIRED FOR CONVERSIONS
C               BETWEEN LATITUDE, LONGITUDE PAIRS AND GRID DISTANCES.
C               ONLY THE BATHYMETRIC PART OF THE FILE IS USED, SHORE-
C               LINE INFORMATION NEED NOT BE INCLUD.
C
C   LOGICAL UNIT 10 : LAKEWIND.DAT
C     METEOROLOGICAL DATA FILE :
C                                                  FORMAT  CARD COLUMNS
C               TLAST - TIME FROM BEGINNING         G10.4     1 - 10
C                        OF RUN (H)
C               PLAT - LATITUDE IN DEGREES NORTH    G10.4    11 - 20
C               PLON - LONGITUDE IN DEGREES WEST    G10.4    21 - 30
C               Z - HEIGHT OF INSTRUMENT (FT)       G10.4    31 - 40
C               TA - TEMPERATURE OF AIR (F)         G10.4    41 - 50
C               TW - TEMPERATURE OF WATER (F)       G10.4    51 - 60
C               WS - WIND SPEED (FT/S)              G10.4    61 - 70
C               WD - WIND DIRECTION (DEG)           G6.0     71 - 76
C
C         ALL DATA FOR THE SAME TIME ARE GROUPED TOGETHER, WITH A
C         MAXIMUM OF 25 STATIONS IN A GROUP.
C
C       * NOTE: END-OF-FILE IS INDICATED BY A RECORD WITH A
C               NEGATIVE TIME.
C
C  OUTPUT :
C    LOGICAL UNIT 6 :
C      CONTROL PARAMETERS, BATHYMETRY. AND A LIST OF THE
C      METEOROLOGICAL DATA RECORDS.
C
C  COMMON BLOCKS :
C               /GRIDS/ DXL, DXR, LGRIDX, NGRIDX, IRGRID, BEGLK
C                  SEE SUBROUTINE SMOSIR FOR DETAILS.
C               /LKMD/ D(40,40), S(40,40), U(40,40), V(40,40)
C               /GPARM/ RPARM(23), IPARM(54), ZPARM(2) - REAL AND INTEGER
C                  PARAMETERS DESCRIBING THE BATHYMETRIC GRID.
C                  SEE SUBROUTINE RGRID FOR DETAILS.
C
C  SUBROUTINES :
C
C        RGRID  - READS THE BATHYMETRIC DATA FILE
C        PGPARM - PRINTS GRID PARAMETERS
```

```
C         PRNT    - FORMATS AND PRINTS OUTPUT ON GRID
C         UZL     - CALCULATES DRAG COEFFICIENTS AND WIND PROFILE
C                   PARAMETERS USED BY FUNCTION WIND
C         FUNCTION WIND  - READS METEOROLOGICAL DATA AND CALCULATES
C                          X AND Y COMPONENTS OF WIND
C         FUNCTION XDIST - RETURNS X DISTANCE FROM GRID ORIGIN GIVEN
C                          LATITUDE AND LONGITUDE
C         FUNCTION YDIST - RETURNS Y DISTANCE FROM GRID ORIGIN GIVEN
C                          LATITUDE AND LONGITUDE
C         FUNCTION UZ    - CALCULATES WIND SPEED AT A DIFFERENT
C                          HEIGHT (ZWIND) THAN THE OBSERVATIONAL HEIGHT (Z)
C                          BASED ON WIND PROFILE PARAMETERS
C
C              THE USER MUST SUPPLY THREE SUBROUTINES TO UPDATE
C              TRANSPORTS AND GENERATE OUTPUT.  THEY ARE:
C
C         UPPART(T,PART,WFACTR,CFACTR,NPMAX) - CALLED AT THE BEGINNING
C              OF EACH TIMESTEP TO INITIALIZE PARTICLE POSITIONS.  T IS
C              IN SECONDS FROM BEGINNING OF RUN.  PART DEFINES THE PARTICLE
C              POSITIONS IN METERS RELATIVE TO THE GRID ORIGIN.  WFACTR IS
C              THE FRACTION OF THE WIND SPEED WITH WHICH PARTICLES MOVE IN
C              PURELY WIND DRIVEN MOTION. CFACTR IS THE FRACTION FOR
C              CURRENT. NPMAX IS THE MAXIMUM NUMBER OF PARTICLES THAT MAY BE
C              INTRODUCED IN ONE TIMESTEP.
C
C         UPDATE(IDIM) - SUPPLIES TRANSPORT FIELD AT EACH TIMESTEP.
C              D IS THE DEPTH ARRAY. U AND V ARE THE X AND Y COMPONENTS OF
C              TRANSPORT. U(I,J), THE X COMPONENT OF TRANSPORT, IS DEFINED AT
C              THE CENTER OF THE RIGHT SIDE OF GRID BOX I,J.  V(I,J), THE Y
C              COMPONENT OF TRANSPORT, IS DEFINED AT THE CENTER OF THE TOP OF
C              GRID BOX I,J.  S(I,J) IS A TEMPORARY STORAGE ARRAY CONTAIN-
C              ING STREAM FUNCTION FIELDS.  IDIM IS THE FIRST DIMENSION OF
C              D, U, AND V.
C
C         POUTP(T,D,PART,IDIM) - GENERATES USER-REQUIRED OUTPUT.
C              POUTP IS CALLED BY PARTIC EVERY TIME STEP (DT, SPECIFIED
C              BY THE USER).  T IS THE TIME IN SECONDS FROM THE BEGINNING
C              OF THE RUN.  D IS THE DEPTH ARRAY.  PART DEFINES THE
C              PARTICLE POSITIONS (IN METERS RELATIVE TO THE GRID ORIGIN).
C              IDIM IS THE FIRST DIMENSION OF D.
C
C  HISTORY :
C                WRITTEN BY J.R. BENNETT, 1982, GLERL, ANN ARBOR,MI.
C
C *************************************************************************
      COMMON /V/ IZRBX(100), IZRBY(100), NZRVB
      COMMON /VA/ VCAR,VWIND,VDRIFT
      COMMON /VASB/ IGRILB(300), IGRIUB(300), IGRLB1(300), IGRUB1(300)
      COMMON /LKMD/ D(40,40), S(40,40), U(40,40), V(40,40)
      COMMON /GPARM/ RPARM(23), IPARM(54), ZPARM(2)
      COMMON /GRIDS/ DXL, DXR, LGRIDX, NGRIDX, IRGRID, BEGLK
      COMPLEX VCAR(8000),VWIND,VDRIFT
      DATA NPMAX, SPVAL, LUNB, LUNM /1000, 1.0E20, 13, 10/
      DATA IDIM, JDIM /40,40/
C  RWD - reference water datum for bathymetric data (FT)
      DATA RWD /571.71/
```

```
C
C  Reread bathymetric grid information.
C
      CALL RGRID(LUNB, D, IDIM, JDIM)
C
C  Read control parameters.
C
      READ(LUNB,100) DT
C
C  DADD - mean water level relative to RWD.
C
      DADD = (CLWL-RWD) / 3.281
      IM = IPARM(1)
      JM = IPARM(2)
      DS = RPARM(3)
      DMAX = RPARM(4)
      DMIN = RPARM(5) + DADD
      IMM1 = IM - 1
      JMM1 = JM - 1
      DT = DT*3600.
C
C  Clear arrays and add water level increment.
C
      DO 20 I = 1,IM
         DO 20 J = 1,JM
            U(I,J) = 0.
            V(I,J) = 0.
            S(I,J) = 0.
            IF ( D(I,J) .LT. RPARM(5)) GOTO 20
            D(I,J) = D(I,J) + DADD
   20 CONTINUE
C
C  If water level increment results in a negative DMIN, stop.
C
      IF (DMIN .LE. 0.0) GOTO 90
C
C  Main iteration loop
C
C  Update transports
C
      CALL UPDATE(IDIM)
C
C  Convert transports to half currents by dividing by twice depth
C
      DO 40 I=1,IMM1
         DO 40 J=1,JMM1
            IF (D(I,J) .GE. DMIN .AND. D(I+1,J) .GE. DMIN)
     1         U(I,J) = U(I,J)/(D(I,J) + D(I+1,J))
            IF (D(I,J) .GE. DMIN .AND. D(I,J+1) .GE. DMIN)
     1         V(I,J) = V(I,J)/(D(I,J) + D(I,J+1))
   40 CONTINUE
C
C  Interpolate currents to stream function points, taking care
C  to use interior currents at the shore boundary.
C
      UMAX = 0.
```

```
      DO 51 I=1,IMM1
        IM1 = I-1
        IF(I .EQ. 1) IM1 = 1
        DO 51 J=1,JMM1
          IF(D(I,J)   .LT. DMIN .AND. D(I+1,J)   .LT. DMIN .AND.
     1       D(I,J+1) .LT. DMIN .AND. D(I+1,J+1) .LT. DMIN) GOTO 51
          UUP = U(I,J+1)
          IF (D(I,J+1) .LT. DMIN) UUP = U(I+1,J+1)
          IF (D(I+1,J+1) .LT. DMIN) UUP = U(IM1,J+1)
          UDN = U(I,J)
          IF (D(I,J) .LT. DMIN) UDN = U(I+1,J)
          IF (D(I+1,J) .LT. DMIN) UDN = U(IM1,J)
          IF (D(I,J) .LT. DMIN .AND. D(I+1,J) .LT. DMIN) UDN=UUP
          IF (D(I,J+1) .LT. DMIN .AND. D(I+1,J+1) .LT. DMIN) UUP=UDN
          S(I,J) = UUP + UDN
C
C Calculate maximum current speed.
C
          UMAX = AMAX1(UMAX,ABS(S(I,J)))
   51 CONTINUE
      DO 50 I=1,IMM1
        DO 50 J=1,JMM1
        U(I,J) = S(I,J)
C
C Interpolate currents to stream function points, taking care
C to use interior currents at the shore boundary.
C
   50 CONTINUE
      DO 53 J=1,JMM1
        JM1 = J-1
        IF (J .EQ. 1) JM1 = 1
        DO 53 I=1,IMM1
          IF(D(I,J)   .LT. DMIN .AND. D(I+1,J)   .LT. DMIN .AND.
     1       D(I,J+1) .LT. DMIN .AND. D(I+1,J+1) .LT. DMIN) GOTO 53
          VR = V(I+1,J)
          IF (D(I+1,J) .LT. DMIN) VR = V(I+1,J+1)
          IF (D(I+1,J+1) .LT. DMIN) VR = V(I+1,JM1)
          VL = V(I,J)
          IF (D(I,J) .LT. DMIN) VL = V(I,J+1)
          IF (D(I,J+1) .LT. DMIN) VL = V(I,JM1)
          IF (D(I,J) .LT. DMIN .AND. D(I,J+1) .LT. DMIN) VL=VR
          IF (D(I+1,J) .LT. DMIN .AND. D(I+1,J+1) .LT. DMIN) VR=VL
          S(I,J) = VL + VR
C
C Calculate maximum current speed.
C
          UMAX = AMAX1(UMAX,ABS(S(I,J)))
   53 CONTINUE
      DO 52 I=1,IMM1
        DO 52 J=1,JMM1
        V(I,J) = S(I,J)
   52 CONTINUE
C
C Set column IM and row JM equal to SPVAL.
C
      DO 58 IN=1,IM
```

```
      U(IN,JM)=SPVAL
      V(IN,JM)=SPVAL
      DO 58 JN=1,JM
         U(IM,JN)=SPVAL
         V(IM,JN)=SPVAL
   58 CONTINUE
C
C  Write lake velocities to VCAR(I).
C  Extrapolate to grid box center using 4-point average.
C
      J1 = 2
C     LM1 = LGRIDX - 1
      DO 21 LX=1,LGRIDX
         X = LX*DXL-0.5*DXL+BEGLK
         ISLU = IGRUB1(LX)
         ISLL = IGRLB1(LX)
         J2 = IGRIUB(LX)-IGRILB(LX)+J1
         LY = IGRILB(LX)
         DO 11 J=J1,J2
            Y = LY*DXL-0.5*DXL
             COUNT=0.0D0
             LX1 = LX -1
             IF(LX1.LE.0)LX1=1
             VVX=(U(LX1,LY) + U(LX,LY) + U(LX1,LY-1) + U(LX,LY-1))/4.0D0
             IF (U(LX1,LY) .EQ. SPVAL) COUNT=COUNT+1.0D0
             IF (U(LX,LY) .EQ. SPVAL) COUNT=COUNT+1.0D0
             IF (U(LX1,LY-1) .EQ. SPVAL) COUNT=COUNT+1.0D0
             IF (U(LX,LY-1) .EQ. SPVAL) COUNT=COUNT+1.0D0
             IF(COUNT.GT.0.0D0 .AND. COUNT.LT.4.0D0) VVX=(VVX*4.0D0-
     &       (COUNT*SPVAL))/(4.0D0-COUNT)
            VVX = VVX*3.281
             COUNT=0.0D0
             VVY=(V(LX1,LY) + V(LX,LY) + V(LX1,LY-1) + V(LX,LY-1))/4.0D0
             IF (V(LX1,LY) .EQ. SPVAL) COUNT=COUNT+1.0D0
             IF (V(LX,LY) .EQ. SPVAL) COUNT=COUNT+1.0D0
             IF (V(LX1,LY-1) .EQ. SPVAL) COUNT=COUNT+1.0D0
             IF (V(LX,LY-1) .EQ. SPVAL) COUNT=COUNT+1.0D0
             IF(COUNT.GT.0.0D0 .AND. COUNT.LT.4.0D0) VVY=(VVY*4.0D0-
     &       (COUNT*SPVAL))/(4.0D0-COUNT)
            VVY = VVY*3.281
            VCAR(J) = CMPLX(VVX,VVY)
            IF(LY.GE.ISLL.AND.LY.LE.ISLU) VCAR(J) = 0.0011
            DO 164 IBOX=1, NZRVB
               IF(LX.EQ.IZRBX(IBOX).AND.LY.EQ.IZRBY(IBOX))
     &         VCAR(J) = 0.0011
  164       CONTINUE
            IF(IOPT2 .EQ. 1) WRITE(4,2100) X, Y, VCAR(J)
            LY = LY+1
   11    CONTINUE
         J1 = J2+3
   21 CONTINUE
      ILVCAR = J2 + 1
C
C  Print out X & Y velocities.
C
C     CALL PRNT(6, U, IDIM, IM, JM, SPVAL)
```

```
C       CALL PRNT(6, V, IDIM, IM, JM, SPVAL)
C
C  End main loop.
C
   30 CONTINUE
      GOTO 9999
   90 WRITE(*,120) DADD
      GOTO 9999
  100 FORMAT(G8.2)
  120 FORMAT('1',20X,'THE WATER LEVEL INCREMENT',F8.2,'RESULTS IN A',
     1         'NEGATIVE MINIMUM DEPTH - PROGRAM TERMINATED')
 2100 FORMAT(2F9.0,2F7.2)
 9999 CONTINUE
      RETURN
      END
```

A213710

There are no pages 178 + 179

There are no pages numbered 178 + 179

```
      SUBROUTINE PGPARM(LUN)
C
C PURPOSE:
C                       TO PRINT THE GRID DESCRIPTION PARAMETERS (RPARM
C                       AND IPARM IN COMMON BLOCK /GPARM/)
C
C ARGUMENTS:
C                       LUN - LOGICAL UNIT NUMBER ON WHICH TO PRINT
C
C COMMON BLOCK:
C                       /GPARM/ RPARM(23),IPARM(54),ZPARM(2)
C
      COMMON /GPARM/ RPARM(23), IPARM(54), ZPARM(2)
      WRITE (6,10) (IPARM(I),I=5,54)
      WRITE (6,20) IPARM(1), IPARM(2), ZPARM(1), ZPARM(2)
      WRITE (6,30) (RPARM(I),I=1,7)
      WRITE (6,40) (RPARM(I),I=8,23)
   10 FORMAT ('0BATHYMETRIC DATA FILE HEADER FOR ', 50A1)
   20 FORMAT ('0IDIMENSION : IPARM(1) =    ', I5, 6X, 'JDIMENSION : ',
     1        'IPARM(2) = ', 8X, I5/'0', 14X,
     2        'DISPLACEMENT OF ORIGIN IN ', 'NUMBER OF GRID SQUARES'/
     3        '0IDISPLACEMENT : ZPARM(1) =', F7.3, 4X,
     4        'JDISPLACEMENT : ZPARM(2) =', 4X, F7.3)
   30 FORMAT ('0BASE LATITUDE : RPARM(1) = ', F12.7/'0BASE LONGITUDE ',
     1        ': RPARM(2) =', F12.7/'0GRID SIZE (M) : RPARM(3) = ', F6.0,
     2        5X, 'MAX DEPTH (M) : RPARM(4) = ', 5X, F6.2/
     3        '0MIN DEPTH (M) : ', 'RPARM(5) = ', F6.2, 4X,
     4        'BASE ROTATION : RPARM(6) = ', 8X, F5.2/
     5        '0ANGLE ROTATED : RPARM(7) = ', F7.2)
   40 FORMAT ('0', 11X, 'GEOGRAPHIC-TO-MAP COORDINATE CONVERSION ',
     1        'COEFFICENTS FOR X'/'0RPARM(8) = ', 6X, E15.6, 5X,
     2        'RPARM(9) = ', 12X, E15.6/'0RPARM(10) = ', 5X, E15.6, 5X,
     3        'RPARM(11) = ', 11X, E15.6/'0', 11X,
     4        'GEOGRAPHIC-TO-MAP COORDINATE CONVERSION COEFFICIENTS',
     5        ' FOR Y'/'0RPARM(12) = ', 5X, E15.6, 5X, 'RPARM(13) = ',
     6        11X, E15.6/'0RPARM(14) = ', 5X, E15.6, 5X, 'RPARM(15) = ',
     7        11X, E15.6/'0', 7X,
     8        'MAP-TO-GEOGRAPHIC COORDINATE CONVERSION COEFFICIENTS',
     9        ' FOR LONGITUDE'/'0RPARM(16) = ', 5X, E15.6, 5X,
     *        'RPARM(17) = ', 11X, E15.6/'0RPARM(18) = ', 5X, E15.6, 5X,
     1        'RPARM(19) = ', 11X, E15.6/'0', 7X,
     2        'MAP-TO-GEOGRAPHIC COORDINATE CONVERSION COEFFICIENTS',
     3        ' FOR LATITUDE'/'0RPARM(20) = ', 5X, E15.6, 5X,
     4        'RPARM(21) = ', 11X, E15.6/'0RPARM(22) = ', 5X, E15.6, 5X,
     5        'RPARM(23) = ', 11X, E15.6)
      RETURN
      END
```

```
      SUBROUTINE PRNT(LUN, A, IDIM, IMAX, JMAX, SPVAL)
C PURPOSE:
C                    TO NORMALIZE AND PRINT THE TWO DIMENSIONAL ARRAY A
C
C ALGORITHM:
C                    PRNT FIRST DETERMINES THE MAXIMUM ABSOLUTE VALUE OF
C                    DATA IN ARRAY A.  ONLY POINTS FOR WHICH A(I,J) IS
C                    NOT EQUAL TO SPVAL ARE CONSIDERED.  IT THEN FINDS
C                    THE POWER OF TEN BY WHICH AMAX MUST BE MULTILPIED
C                    FOR IT TO LIE BETWEEN 100 AND 1000.  THE POWER OF
C                    TEN IS PRINTED, FOLLOWED BY THE NORMALIZED DATA,
C                    FORMATTED AS 3 DIGIT INTEGERS.  POINTS AT WHICH
C                    A(I,J) IS EQUAL TO SPVAL ARE MASKED BY ASTERISKS.
C                    DATA ARE OUTPUT IN BLOCKS WITH J DECREASING DOWN
C                    AND I INCREASING ACROSS THE PAGE.  THE NUMBER OF
C                    VALUES PRINTED ACROSS A PAGE IS AN INTERNAL PARA-
C                    METER IN THE SUBROUTINE.
C ARGUMENTS:
C                    LUN - LOGICAL UNIT NUMBER ON WHICH TO PRINT
C                    A - TWO-DIMENSIONAL ARRAY TO BE NORMALIZED AND
C                       PRINTED. UNCHANGED BY PRNT.
C                    IDIM - FIRST DIMENSION OF A AND D IN DIMENSION
C                       STATEMENT OF CALLING PROGRAM
C                    IMAX - MAXIMUM I VALUE ACTUALLY USED IN A AND D
C                    JMAX - MAXIMUM J VALUE ACTUALLY USED IN A AND D
C                    SPVAL - SPECIAL MASKING VALUE: ONLY THE A(I,J)
C                             WHICH ARE NOT EQUAL TO SPVAL ARE PRINTED
C
C **************************************************************************
      DIMENSION INTEG(30), A(40,40)
C
C  NCOL IS THE NUMBER OF VALUES TO PRINT ACROSS A PAGE
C
      NCOL = 19
C
C  AMAX=MAXIMUM ABSOLUTE VALUE OF ARRAY A
C
      AMAX = 0.0
      DO 10 I = 1, IMAX
        DO 10 J = 1, JMAX
          IF (A(I,J) .EQ. SPVAL) GOTO 10
          AMAX = AMAX1(AMAX,ABS(A(I,J)))
   10 CONTINUE
C
C  NOW FIND THE POWER OF TEN BY WHICH WE MUST MULTIPLY AMAX
C  SO THAT IT FALLS BETWEEN 100 AND 1000.
C
C  INITIALLY THE POWER IS ZERO
C
      MP = 0
      IF (AMAX .EQ. 0) GOTO 20
C
C  TAKE BASE 10 LOGARITHM OF AMAX
C
      AP = ALOG10(AMAX)
```

```
C
C  IF AMAX IS GREATER THAN 1000, MP IS NEGATIVE
C
      IF (AP .GT. 3.) MP = -IFIX(AP - 2.)
C
C  IF AMAX IS LESS THAN 100, MP IS POSITIVE
C
      IF (AP .LT. 2.) MP = IFIX(3. - AP)
   20 CONTINUE
C
C  PRINT THE GRID
C
      I1 = 1
      II = (IMAX - 1) / NCOL + 1
      IRMDR = IMAX - NCOL * (II - 1)
      DO 50 L = 1, II
C
C  WHEN L=II ONLY PRINT IRMDR VALUES
C
        IF (L .EQ. II) NCOL = IRMDR
C
C  PRINT THE POWER
C
        WRITE(LUN,60) MP
        I2 = I1 + NCOL - 1
        DO 40 JJ = 1, JMAX
          J = JMAX - JJ + 1
          DO 30 I = I1, I2
            I3 = 1 + I - I1
            INTEG(I3) = -9999
            IF (A(I,J) .NE. SPVAL) INTEG(I3) = INT(A(I,J)*10.**MP +
     1      SIGN(0.5,A(I,J)*10.**MP))
   30     CONTINUE
          WRITE(LUN,70) (INTEG(I),I=1,NCOL)
   40   CONTINUE
        I1 = I2 + 1
   50 CONTINUE
   60 FORMAT ('0 VALUES MULTIPLIED BY 10**', I3)
   70 FORMAT (' ', 30I4)
      RETURN
      END
```

```
      SUBROUTINE RGRID(LUN, D, IDIM, JDIM)
C PURPOSE:
C                TO READ A STANDARD BATHYMETRIC GRID DATA FILE
C                AND RETURN GRID PARAMETERS AND DEPTHS.
C ARGUMENTS:
C ON INPUT:
C                LUN - LOGICAL UNIT NUMBER OF BATHYMETRIC DATA FILE
C                IDIM - FIRST DIMENSION OF ARRAY D IN
C                       DIMENSION STATEMENT OF CALLING PROGRAM
C                JDIM - SECOND DIMENSION OF ARRAY D IN
C                       DIMENSION STATEMENT OF CALLING PROGRAM
C ON OUTPUT:
C                D - DEPTH ARRAY. ZERO FOR LAND, AVERAGE DEPTH
C                    OF GRID BOX IN METERS FOR WATER.
C                RPARM - ARRAY CONTAINING REAL-VALUED BATHYMETRIC
C                        GRID PARAMETERS AS FOLLOWS:
C                    1.  BASE LATITUDE
C                    2.  BASE LONGITUDE
C                    3.  GRID SIZE (FT)
C                    4.  MAXIMUM DEPTH (FT)
C                    5.  MINIMUM DEPTH (FT)
C                    6.  BASE ROTATION (COUNTERCLOCKWISE NEGATIVE)
C                    7.  ROTATION FROM BASE (COUNTERCLOCKWISE NEGATIVE)
C                    8-11. GEOGRAPHIC-TO-MAP COORDINATE CONVERSION
C                          COEFFICIENTS FOR X
C                    12-15.  GEOGRAPHIC-TO MAP COORDINATE CONVERSION
C                            COEFFICIENTS FOR Y
C                    16-19.  MAP-TO-GEOGRAPHIC COORDINATE CONVERSION
C                            COEFFICIENTS FOR LONGITUDE
C                    20-23.  MAP-TO-GEOGRAPHIC COORDINATE CONVERSION
C                            COEFFICIENTS FOR LATITUDE
C                IPARM - ARRAY CONTAINING INTEGER-VALUED BATHYMETRIC
C                        GRID PARAMETERS AS FOLLOWS:
C                    1.  NUMBER OF GRID BOXES IN X DIRECTION
C                    2.  NUMBER OF GRID BOXES IN Y DIRECION
C                    3.  NOT USED
C                    4.  NOT USED
C                    5-54.  LAKE NAME (50A1)
C                NOTE: IF GRID IS TOO LARGE FOR DIMENSIONS OF D,
C                      THE IPARM ARRAY IS SET TO ZERO
C
C                ZPARM - ARRAY CONTAINING REAL-VALUED BATHYMETRIC
C                        GRID PARAMETERS AS FOLLOWS:
C                    1.  I DISPLACEMENT - THE NUMBER OF NEW GRID
C                        SQUARES IN THE X-DIRECTION FROM THE NEW
C                        GRID ORIGIN TO THE OLD GRID ORIGIN
C                    2.  J DISPLACEMENT - THE NUMBER OF NEW GRID
C                        SQUARES IN THE Y-DIRECTION FROM THE NEW
C                        GRID ORIGIN TO THE OLD GRID ORIGIN
C
C COMMON BLOCK:
C                /GPARM/RPARM(23),IPARM(54),ZPARM(2)
C
C   Last Date of Revision : September 11, 1985
C
```

```
C ****************************************************************************
      DIMENSION D(IDIM,JDIM)
      COMMON /GPARM/ RPARM(23), IPARM(54), ZPARM(2)
      REWIND LUN
      READ (LUN,30) (IPARM(I),I=5,54), IPARM(1), IPARM(2),
     1(RPARM(I),I=1,6), ZPARM(1), ZPARM(2), RPARM(7)
      READ (LUN,60) (RPARM(I),I=8,23)
      IM = IPARM(1)
      JM = IPARM(2)
      RPARM(3) = RPARM(3) / 3.281
      RPARM(4) = RPARM(4) / 3.281
      RPARM(5) = RPARM(5) / 3.281
      IF (IPARM(1) .GT. IDIM .OR. IPARM(2) .GT. JDIM) GOTO 10
      READ (LUN,40) ((D(I,J),I=1,IM),J=1,JM)
      DO 80 I=1,IM
         DO 80 J=1,JM
            D(I,J) = D(I,J) / 3.281
   80 CONTINUE
      RETURN
   10 DO 20 I = 1, 54
   20 IPARM(I) = 0
      WRITE(*,50)
   30 FORMAT (50(A1)/2I5, 2F12.7, 3F5.0, F6.2, 2F7.3, F7.2)
   40 FORMAT (19F4.0, 4X)
   50 FORMAT (' BATHYMETRIC GRID TOO LARGE - INCREASE DIMENSIONS OF',
     1        ' NDEPTH AND DEPTH IN MAIN PROGRAM')
   60 FORMAT (4E15.6, 20X)
   70 RETURN
      END
```

```
      SUBROUTINE RLID(UFDT, TLKQ, CLWL, TIMET, INDPRN)
C
C  To avoid general confusion, the documentation supplied by GLERL
C  has been modified from the original version contained in the
C  PATHFINDER Particle Trajectory Program.  The modifications
C  merely remove unnecessary comments and definitions and add any
C  changes to the program.  These include:
C   1.) The February 1983 comment by D. J. Schwab was removed.
C   2.) The control parameters and bathymetric data is read in on
C        logical unit 13 but the meteorlogical data is read from
C        logical unit 10.  The data is initially in English (fps)
C        units but is converted into Metric (mks) units for use in
C        calculating the lake circulation. However, the final lake
C        velocities are reconverted into English units.
C   3.) Initial stream function values are read in from logical
C        unit 14 using subroutine INIT.
C   4.) Stream function values, for use in calculating lake grid
C        velocities, are written to logical unit 15 (a temporary file)
C        as controlled by subroutine OUTP.
C   5.) Common Blocks GRIDS, ICE, VASB & LKMD were added.
C
C  Last Date of Revision : September 24, 1985
C
C**************************PROGRAM RLID*******************************
C
C                        RIGID LID CIRCULATION MODEL
C
C PURPOSE:
C               THE PURPOSE OF THIS PROGRAM IS TO COMPUTE THE TIME-DEP-
C               ENDENT CIRCULATION FOR A GRID MODEL OF A LAKE.  THE
C               CIRCULATION IS ASSUMED TO BE NON-DIVERGENT SO THAT IT
C               CAN BE REPRESENTED IN TERMS OF A STREAM FUNCTION.  THE
C               LAKE IS REPRESENTED AS AN ARRAY OF SQUARE GRID BOXES
C               AND A FINITE DIFFERENCE FORM OF THE VORTICITY
C               EQUATION IS APPLIED TO THE GRID.  THE STREAM FUNCTION
C               S(I,J) IS DEFINED AT THE TOP RIGHT CORNER OF GRID SQUARE
C               I, J.  THE USER IS REQUIRED TO SUPPLY A DESCRIPTION OF
C               THE LAKE BATHYMETRY, THE METEOROLOGICAL FORCING CONDIT-
C               IONS, A SUBROUTINE THAT SETS THE INITIAL CONDITION FOR
C               THE STREAM FUNCTION FIELD (WHICH MAY INCLUDE INFLOW AND
C               AND OUTFLOW CONDITIONS), CONTROL PARAMETERS (TIMESTEP,
C               DURATION OF RUN, AND WATER LEVEL INCREMENT IF RE-
C               QUIRED), AND A SUBROUTINE TO HANDLE OUTPUT FUNCTIONS.
C  INPUT :
C
C         NOTE: CLWL AND TLKQ SUPPLIED THROUGH SMOSIR & NDCONV
C
C                CLWL- CURENT LAKE WATER LEVEL (FT)
C                TLKQ- PRESENT TOTAL LAKE DISCHARGE
C
C   LOGICAL UNIT 13 : LAKEBATH.DAT
C     CONTROL PARAMETER RECORD :
C                                                  FORMAT   CARD COLUMNS
C                DT  - TIME STEP (HOURS)           G8.2        1 -  8
C
```

```
C     BATHYMETRIC DATA FILE :
C               THE FORMAT OF THE BATHYMETRIC DATA FILE IS DESCRIBED
C               IN SCHWAB AND SELLERS (1980): 'COMPUTERIZED BATHY-
C               METRY AND SHORELINES OF THE GREAT LAKES', NOAA DATA
C               REPORT ERL-GLERL-16. FIVE ADDITIONAL FIELDS ARE
C               PRESENT ON BATHYMETRIC DATA HEADER RECORD 2. THESE
C               ARE:
C                                                   FORMAT  CARD COLUMNS
C               MINIMUM DEPTH (FT)                   I5       45-49
C               BASE GRID ROTATION FROM E-W          F6.2     50-55
C               I DISPLACEMENT                       F7.3     56-62
C               J DISPLACEMENT                       F7.3     63-69
C               ROTATION ANGLE  FROM BASE            F7.2     70-76
C                (ANGLES MEASURED IN DEGREES COUNTERCLOCKWISE)
C
C               THE ADDITIONAL FIELDS ARE REQUIRED FOR CONVERSIONS
C               BETWEEN LATITUDE, LONGITUDE PAIRS AND GRID DISTANCES.
C               ONLY THE BATHYMETRIC PART OF THE FILE IS USED, SHORE-
C               LINE INFORMATION NEED NOT BE INCLUDED.
C
C    LOGICAL UNIT 10 : LAKEWIND.DAT
C     METEOROLOGICAL DATA FILE :
C                                                   FORMAT  CARD COLUMNS
C               TLAST - TIME FROM BEGINNING          G10.4     1 - 10
C                       OF RUN (H)
C               RLAT - LATITUDE IN DEGREES NORTH     G10.4    11 - 20
C               RLON - LONGITUDE IN DEGREES WEST     G10.4    21 - 30
C               Z - HEIGHT OF INSTRUMENT (FT)        G10.4    31 - 40
C               TA - TEMPERATURE OF AIR (F)          G10.4    41 - 50
C               TW - TEMPERATURE OF WATER (F)        G10.4    51 - 60
C               WS - WIND SPEED (FT/S)               G10.4    61 - 70
C               WD - WIND DIRECTION (DEG)            G6.0     71 - 76
C
C          ALL DATA FOR THE SAME TIME ARE GROUPED TOGETHER, WITH A
C          MAXIMUM OF 25 STATIONS IN A GROUP.
C
C    *NOTE END-OF-FILE IS INDICATED BY A RECORD WITH A NEGATIVE TIME
C
C   OUTPUT :
C    LOGICAL UNIT 6 :
C      CONTROL PARAMETERS, BATHYMETRY, A LIST OF THE
C      METEOROLOGICAL DATA RECORDS, U; TRANSPORT IN X-DIRECTION,
C      V; TRANSPORT IN Y-DIRECTION, S; STREAM TRANSPORT, D; DEPTHS
C
C   COMMON BLOCKS :
C                /ICE/ ZWND(40,40),ZLKICE(20),NICEX1(20),NICEY1(20),
C                 NICEX2(20),NICEY2(20),IPOS1(20),IPOS2(20),AMIUO,ANICE,
C                 SPAICE,NICERG,LICERG
C               /LKMD/ D(40,40), S(40,40), U(40,40), V(40,40)
C                  SEE SUBROUTINE SMOSIR FOR DETAILS.
C               /VASB/ IGRILB(300),IGRIUB(300),IGRLB1(300),IGRUB1(300)
C               /GPARM/ RPARM(23), IPARM(54), ZPARM(2) - REAL AND INTEGER
C                  PARAMETERS DESCRIBING THE BATHYMETRIC GRID.
C                  SEE SUBROUTINE RGRID FOR DETAILS.
C               /ITPARM/ DSTMAX, STMIN, STMAX, FRAC, ITS
C                  RELAXATION PARAMETERS:
```

```
C                         DSTMAX - MAXIMUM CHANGE IN ABSOLUTE VALUE OF
C                                  STREAM FUNCTION AT LAST ITERATION
C                         STMIN - MINIMUM VALUE OF STREAM FUNCTION
C                         STMAX - MAXIMUM VALUE OF STREAM FUNCTION
C                         FRAC - DSTMAX / (STMAX   STMIN)
C                         ITS - NUMBER OF ITERATIONS TAKEN TO CONVERGE
C
C  SUBROUTINES:
C        RGRID - READS THE BATHYMETRIC DATA FILE
C        PGPARM - PRINTS GRID PARAMETERS
C        PRNT - FORMATS AND PRINTS DATA FROM GRID I.E. STREAM FUNCTION,DEPTH
C        UZL - CALCULATES DRAG COEFFICIENT FOR METEOROLOGICAL DATA
C        FUNCTION TAU - READS METEOROLOGICAL DATA AND CALCULATES
C             WIND STRESS
C        FUNCTION XDIST - RETURNS X DISTANCE FROM GRID ORIGIN
C             GIVEN LATITUDE AND LONGITUDE
C        FUNCTION YDIST - RETURNS Y DISTANCE FROM GRID ORIGIN
C             GIVEN LATITUDE AND LONGITUDE
C        FUNCTION RLAT - RETURNS LATITUDE GIVEN X AND Y DISTANCE
C             FROM GRID ORIGIN
C
C  THE USER MUST SUPPLY TWO SUBROUTINES TO HANDLE INITIAL
C  CONDITIONS AND OUTPUT.  THEY ARE:
C
C        INIT(D, S, TLKQ, RLKQ, IDIM) - SETS INITIAL CONDITION FOR STREAM
C          FUNCTION FIELD (S).  D IS THE DEPTH ARRAY (INTERPOLATED TO
C          STREAM FUNCTION POINTS) AND IDIM IS THE FIRST DIMENSION OF
C          D AND S.  TLKQ IS THE CURRENT TOTAL LAKE DISCHARGE.  RLKQ IS
C          THE REFERENCE DISCHARGE FOR THE INITIAL STREAM FUNCTION FILE.
C
C        OUTP(TIME, TTS, IDIM) - GENERATES USER-REQUIRED
C          OUTPUT. OUTP IS CALLED BY RLID EACH TIMESTEP WITH THE CURRENT
C          TIME, TIME, IN SECONDS. THE TIME AT WHICH THE STREAM FUNCTION
C          WILL BE SAVED, TTS, AND THE DIMENSION OF THE STREAM FUNCTION
C          ARRAY, IDIM.
C
C  HISTORY: WRITTEN BY J.  R. BENNETT AND D. J. SCHWAB, 1981,
C  GLERL, ANN ARBOR, MI
C
C  MODIFIED 3/83 TO REFLECT CERTAIN IMPROVEMENTS IN THE
C  RELAXATION SCHEME AND THE WAY DMIN AND DADD ARE HANDLED
C
C ************************************************************************
      DIMENSION RHS(40,40), SPD(40,40), FR(40,40)
      COMMON /ICE/ZWND(40,40),ZLKICE(20),NICEX1(20),NICEY1(20),
     $ NICEX2(20),NICEY2(20),IPOS1(20),IPOS2(20),AMIUO,ANICE,
     $ SPAICE,NICERG,LICERG
      COMMON /LKMD/ D(40,40), S(40,40), U(40,40), V(40,40)
      COMMON /VASB/ IGRILB(300),IGRIUB(300),IGRLB1(300),IGRUB1(300)
      COMMON /GPARM/ RPARM(23), IPARM(54), ZPARM(2)
      COMMON /ITPARM/ DSTMAX, STMIN, STMAX, FRAC, ITS
      COMMON /GRIDS/ DXL, DXR, LGRIDX, NGRIDX, IRGRID, BEGLK
      DATA IDIM, JDIM /40,40/
C
C  IDIM AND JDIM ARE THE FIRST AND SECOND DIMENSIONS OF THE ARRAYS
C  D, S, RHS, AND SPD
```

```
C
      DATA LUNB, LUNM, LUNS /13, 10, 6/
C
C  LUNB IS THE LOGICAL UNIT NUMBER FOR THE BATHYMETRIC DATA
C  LUNM IS THE LOGICAL UNIT NUMBER FOR THE METEOROLOGICAL DATA
C
      DATA OM, FRO, FRI /7.29E-5, 2.E-3, 3.E-3/
C
C  PHYSICAL CONSTANTS:
C    OM - ANGULAR SPEED OF ROTATION OF THE EARTH (RAD / S)
C    FRO - FRICTIONAL DRAG COEFFICIENT FOR BOTTOM STRESS
C    FRI - FRICTIONAL DRAG COEFFICIENT FOR ICE STRESS
C
      DATA RELAX, ITMAX, CONV /1.6, 50, 1.E-3/
C
C  CONTROL PARMETERS FOR STREAM FUNCTION RELAXATION SCHEME
C    RELAX - OVERRELAXATION FACTOR FOR ITERATIVESOLUTION OF STREAM
C            FUNCTION EQUATION AT EACH TIMESTEP
C    ITMAX - MAXIMUM NUMBER OF ITERATIONS FOR RELAXATION SCHEME AT
C            EACH TIMESTEP
C    CONV -  RELATIVE CONVERGENCE CRITERION FOR RELAXATION SCHEME
C
      DATA RLKQ, RWD, IFIRST /201323., 571.71, 0/
C
C  RLKQ - REFERENCE LAKE DIS. FOR STREAM FUNCTION INITIAL CONDITION FILE
C  RWD - REFERENCE WATER LEVEL FOR BATHYMETRIC DATA
C
      PI = ATAN(1.) * 4.
C
C  FIRST TIME THROUGH, MODEL RUNS 20 HOURS TO STEADY STATE.  THEN, RUNS
C  FOR UFDT AMOUNT OF TIME TO UPDATE DISCHARGES AND VELOCITIES. TTS
C  IS USED TO KEEP TRACK OF DURATION OF RLID EXECUTION. TIMET IS INCREASED
C  BY A NEGLIGIBLE AMOUNT OF TIME TO FACILITATE ACCURATE READING OF
C  METEOROLOGICAL DATA AND DECREASED BY SAME AMOUNT BEFORE RETURN TO SMOSIR
C
      TIMSAV = TIMET
      TIMET = TIMET + 0.000001
      TTS = UFDT
      IF (IFIRST .NE. 0) GOTO 5
      TTS = 20.
    5 CONTINUE
C
C  READ BATHYMETRIC GRID INFORMATION
C
      CALL RGRID(LUNB, D, IDIM, JDIM)
C
C  READ CONTROL PARAMETERS
C
      READ (LUNB,100) DT
      NSTEPS = TTS / DT
C
C  DADD - MEAN WATER LEVEL RELATIVE TO RWD
C
      DADD = (CLWL-RWD) / 3.281
      IM = IPARM(1)
      JM = IPARM(2)
```

```
      DS = RPARM(3)
      DMAX = RPARM(4)
      DMIN = RPARM(5) + DADD
C
C  CALCULATE CORIOLIS PARAMETER AT CENTER OF GRID
C
      F = 2. * OM * SIN(RLAT(IM*DS/2.,JM*DS/2.)*PI/180.)
      IF(INDPRN .EQ. 1) WRITE(LUNS,110) (IPARM(I),I=5,54), DT, TTS,
     1 DADD, NSTEPS, F
      IMM1 = IM - 1
      JMM1 = JM - 1
      IMM2 = IM - 2
      JMM2 = JM - 2
      DT = DT * 3600.
      FDT24 = F * DT / 24.
C
C  ADJUST RELAXATION FACTOR FOR GRID SIZE
C
      IF(IFIRST .NE. 0) GOTO 6
      RELAX = RELAX / (1. + SIN(ACOS(0.5*(COS(PI/IMM2) + COS(PI/JMM2)))
     1))
C
C  INTERPOLATE DEPTH TO STREAM FUNCTION POINTS
C  USING SPEED ARRAY FOR TEMPORARY STORAGE
C
    6 DO 10 I = 1, IM
        DO 10 J = 1, JM
          SPD(I,J) = 0.0
          S(I,J) = 0.
          FR(I,J) = FRO
          ZWND(I,J) = 1.0E0
          IF (D(I,J) .LT. RPARM(5)) GOTO 10
          D(I,J) = D(I,J) + DADD
   10 RHS(I,J) = 0.
C
C  SET FRICTION FACTOR AND ZERO WIND ARRAY IN ICE REGIONS
C  ALSO ADJUST DEPTHS FOR ICE THICKNESS
C
      IF(LICERG.EQ.0) GOTO 2
      DO 1 N=1,LICERG
         LBEG = NICEX1(N)
         LEND = NICEX2(N)
         DO 111 L=LBEG,LEND
            IF(L .GT. LGRIDX) GOTO 111
            IF(L .EQ. LBEG) M1=NICEY1(N)
            IF(L .EQ. LBEG .OR. L .EQ. LEND) M2=NICEY2(N)
            IF(L .EQ. LBEG .AND. L .NE. LEND) M2=IGRIUB(L)
            IF(L .EQ. LEND .AND. L .NE. LBEG) M1=IGRILB(L)
            IF(L .NE. LBEG .AND. L .NE. LEND) M1=IGRILB(L)
            IF(L .NE. LBEG .AND. L .NE. LEND) M2=IGRIUB(L)
            DO 222 M=M1,M2
               IF(M.LE.IGRUB1(L) .AND. M.GE.IGRLB1(L))GOTO 222
               FR(L,M) = FR(L,M) + FRI
               ZWND(L,M) = 0.0E0
               D(L,M) = D(L,M) - 0.9*(ZLKICE(N)/3.281)
  222 CONTINUE
```

```
  111 CONTINUE
   1  CONTINUE
      IF(INDPRN .EQ. 1) WRITE(*,160)
      IF(INDPRN .EQ. 1) CALL PRNT(6, FR, IDIM, IM, JM, FRO)
    2 CONTINUE
C
C  IF WATER LEVEL INCREMENT RESULTS IN A NEGATIVE DMIN, STOP.
C
      IF (DMIN .LE. 0.0) GOTO 90
      DO 20 I = 1, IM
        DO 20 J = 1, JM
          SPD(I,J) = 0.999999 * DMIN
          IF (I .EQ. IM .OR. J .EQ. JM) GOTO 20
          IF (D(I,J) .LT. DMIN) GOTO 20
          IF (D(I + 1,J) .LT. DMIN) GOTO 20
          IF (D(I,J + 1) .LT. DMIN) GOTO 20
          IF (D(I + 1,J + 1) .LT. DMIN) GOTO 20
          SPD(I,J) = 0.25 * (D(I,J) + D(I + 1,J) + D(I,J + 1) + D(I + 1,
     1    J + 1))
   20 CONTINUE
C
C  PRINT BATHYMETRIC GRID PARAMETERS
C
      IF(IFIRST .NE. 0) GOTO 25
      IF(INDPRN .EQ. 1) CALL PGPARM(LUNS)
C
C  PRINT BATHYMETRIC GRID INFORMATION
C
      IF(INDPRN .EQ. 1) WRITE(LUNS,140)
      IF(INDPRN .EQ. 1) CALL PRNT(6, D, IDIM, IM, JM, 0.)
C
C  STORE INVERSE DEPTH BACK IN D
C
   25 DMINI = 1. / DMIN
      DO 30 I = 1, IM
        DO 30 J = 1, JM
          D(I,J) = 1. / SPD(I,J)
   30 SPD(I,J) = 0.0
C
C  GET INITIAL CONDITIONS
C
      CALL INIT(S, IDIM, TLKQ, RLKQ, INDPRN)
C
C  MAIN ITERATION LOOP
C
      TIME=0.
C     WRITE(*,150)
      DO 80 N = 1, NSTEPS
        TIME = TIME+DT/2.
C
C  CALCULATE CURRENT SPEED AT CENTER OF GRID BOX I, J
C
        DO 40 I = 2, IMM1
          DO 40 J = 2, JMM1
            IF(D(I,J) .GT. DMINI .AND. D(I-1,J) .GT. DMINI .AND.
     1      D(I,J-1) .GT. DMINI .AND. D(I-1,J-1) .GT. DMINI) GOTO 40
```

```
            DU = 0.5 * (D(I,J) + D(I,J - 1))
            DV = 0.5 * (D(I,J) + D(I - 1,J))
            DUM = 0.5 * (D(I - 1,J) + D(I - 1,J - 1))
            DVM = 0.5 * (D(I,J - 1) + D(I - 1,J - 1))
            SPD(I,J) = (0.5/DS) * SQRT(((((S(I,J) - S(I,J - 1))*DU) + ((
     1      S(I - 1,J) - S(I - 1,J - 1))*DUM))**2 + (((S(I,J) - S(I - 1,
     2      J))*DV) + ((S(I,J - 1) - S(I - 1,J - 1))*DVM))**2))
   40    CONTINUE
C
C  ITERATE TO CALCULATE STREAM FUNCTION AT NEXT TIME STEP WITH ALTER-
C  NATING  SWEEP DIRECTIONS
C
         DO 60 K = 1, ITMAX
C          WRITE(*,*) K
           KK=K+N
           DSTMAX = 0.
           STMIN = 0.
           STMAX = 0.
           ITS = K
           DO 50 II = 1, IM
             I = II
             IF (MOD(KK,2) .EQ. 0) I = IM - II + 1
             DO 50 JJ = 1, JM
               J = JJ
               IF (MOD(KK,2) .EQ. 0) J = JM - JJ + 1
               IF (D(I,J) .GT. DMIND) GOTO 50
               DUP = 0.5 * (D(I,J + 1) + D(I,J))
               DVP = 0.5 * (D(I + 1,J) + D(I,J))
               SPDUP = 0.5 * (SPD(I + 1,J + 1) + SPD(I,J + 1))
               SPDVP = 0.5 * (SPD(I + 1,J + 1) + SPD(I + 1,J))
               DU = 0.5 * (D(I,J) + D(I,J - 1))
*               DV = 0.5 * (D(I,J) + D(I - 1,J))
               SPDU = 0.5 * (SPD(I + 1,J) + SPD(I,J))
               SPDV = 0.5 * (SPD(I,J + 1) + SPD(I,J))
               DCENT = DVP + DV + DUP + DU
C
C  LAPLACIAN TERM
C
               TERM1 = DVP * S(I + 1,J) + DV * S(I - 1,J) + DUP * S(I,J +
     1          1) + DU * S(I,J - 1) - DCENT * S(I,J)
C
C  ARAKAWA'S JACOBIAN
C
               TERM2 = S(I + 1,J) * (D(I,J + 1) + D(I + 1,J + 1) - D(I,J
     1         - 1) - D(I + 1,J - 1)) + S(I - 1,J) * (-D(I,J + 1) - D(I -
     2          1,J + 1) + D(I,J - 1) + D(I - 1,J - 1)) + S(I,J + 1) * (-
     3         D(I + 1,J) - D(I + 1,J + 1) + D(I - 1,J) + D(I - 1,J + 1))
     4         +S(I,J - 1) * (D(I + 1,J) + D(I + 1,J - 1) - D(I - 1,J) -
     5         D(I - 1,J - 1)) + S(I + 1,J + 1) * (-D(I + 1,J) + D(I,J +
     6         1)) + S(I + 1,J - 1) * (D(I + 1,J) - D(I,J - 1)) + S(I -
     7         1,J + 1) * (D(I - 1,J) - D(I,J + 1)) + S(I - 1,J - 1) * (-
     8         D(I - 1,J) + D(I,J - 1))
C
C  FRICTION TERM
C
               TYP = -DVP * (S(I + 1,J) - S(I,J)) * FR(I+1,J) * SPDVP
```

```
              TYM = -DV * (S(I,J) - S(I - 1,J)) * FR(I-1,J) * SPDV
              TXP = DUP * (S(I,J + 1) - S(I,J)) * FR(I,J+1) * SPDUP
              TXM = DU * (S(I,J) - S(I,J - 1)) * FR(I,J-1) * SPDU
              TERM3 = (DVP*TYP - DV*TYM - DUP*TXP + DU*TXM)
C
C   SET RIGHT HAND SIDE THE FIRST TIME THROUGH
C
              IF (K .EQ. 1) RHS(I,J) = TERM1 - FDT24 * TERM2 + DT * 0.5
     1        * TERM3+DT * DS * (DVP*ZWND(I+1,J)*TAU(TIMET,I+1,J,2) -
     2        DV*ZWND(I,J)*TAU(TIMET,I,J,2) - DUP*ZWND(I,J+1)*
     3        TAU(TIMET,I,J+1,1)+DU*ZWND(I,J)*TAU(TIMET,I,J,1))
              IF (K .EQ. 1) GOTO 50
C
C   CALCULATE NEW STREAM FUNCTION
C
              D4 = DCENT + FR(I,J) * 0.5 * DT * (DVP**2*SPDVP + DV**2*
     1        SPDV + DUP**2*SPDUP + DU**2*SPDU)
              DST = (TERM1 + FDT24*TERM2 - 0.5*DT*TERM3 - RHS(I,J)) / D4
              S(I,J) = S(I,J) + RELAX * DST
              DSTMAX = AMAX1(DSTMAX,ABS(DST))
              STMIN = AMIN1(STMIN,S(I,J))
              STMAX = AMAX1(STMAX,S(I,J))
   50      CONTINUE
C
C   CALCULATE RELATIVE CHANGE IN STREAM FUNCTION FOR ALL ITERATIONS BUT
C   THE FIRST
C
           IF (K .EQ. 1) GOTO 60
           IF (STMAX .EQ. STMIN) GOTO 70
           FRAC = DSTMAX / (STMAX - STMIN)
           IF (FRAC .LE. CONV) GOTO 70
   60    CONTINUE
   70    CONTINUE
C
C   UPDATE TIME
C
      TIME=N*DT
C
C   CALL OUTPUT ROUTINE
C
      CALL OUTP(TIME, TTS, IDIM)
C     IF(INDPRN .EQ. 1) WRITE(LUNS,*) N
C
C   END MAIN ITERATION LOOP
C
   80 CONTINUE
      GOTO 130
   90 WRITE(LUNS,120) DADD
      GOTO 130
  100 FORMAT (G8.2)
  110 FORMAT ('1RIGID LID CIRCULATION MODEL FOR ',
     1        50A1/' TIME STEP(H): DT= ', F10.2/
     1       ' DURATION OF RUN(H): TT= ', F7.2/
     2       ' MEAN WATER LEVEL(M) (RELATIVE TO L.W.D.): DADD= ', F5.2/
     3       ' NUMBER OF TIME STEPS: NSTEPS=', I6/
     4       ' CORIOLIS PARAMETER (S**-1): F= ', E10.3)
```

```
 120 FORMAT (' THE WATER LEVEL INCREMENT', F8.2, ' RESULTS IN A',
    1         ' NEGATIVE MINUMUM DEPTH - PROGRAM TERMINATED')
 140 FORMAT (1H1,/'Present Lake Grid Depths in Meters'/)
C 150 FORMAT (1H1,/'Time Step # for Stream Function Calculation'/)
 160 FORMAT (1H1,/'Areas with Ice Cover and NO Wind Stress,',
    1' Numbers Represent Drag Coefficient'/)
 130 CONTINUE
     IFIRST = 1
     TIMET = TIMSAV
     RETURN
     END
```

Subroutine UPDATE

```
      SUBROUTINE UPDATE (IDIM)
C
C PURPOSE :
C                    TO UPDATE THE TRANSPORTS BY READING
C                    THE STREAM FUNCTION FIELDS AT INTERVALS
C                    SPECIFIED BY THE USER.
C
C ARGUMENTS :
C                    D - DEPTH ARRAY (METERS)
C                    U - COMPONENT OF TRANSPORT IN X-DIRECTION
C                    V - COMPONENT OF TRANSPORT IN Y-DIRECTION
C                    S - STORAGE ARRAY CONTAINING STREAM FUNCTION FIELDS
C                    IDIM - FIRST DIMENSION OF D,U, AND V IN DIMENSION STATEMENT
C                           OF CALLING PROGRAM
C
C COMMON BLOCK :
C                    /GPARM/ RPARM(23), IPARM(54),ZPARM(2)
C   Last Date of Revision : September 24, 1985
C
C ************************************************************************
      COMMON /LKMD/ D(40,40), S(40,40), U(40,40), V(40,40)
      COMMON /GPARM/ RPARM(23), IPARM(54), ZPARM(2)
      IM = IPARM(1)
      JM = IPARM(2)
      DS = RPARM(3)
C
C  REWIND STREAM FUNCTION FILE
C
      REWIND 15
C
C  GET STREAM FUNCTION FROM CURRENT TIMESTEP
C
      READ(15,40) ((S(I,J),I=1,IM),J=1,JM)
C
C  CALCULATE X COMPONENT OF TRANSPORT
C
      DO 20 I=1,IM
         DO 20 J=2,JM
            U(I,J)=(S(I,J-1) - S(I,J))/DS
   20 CONTINUE
C
C  CALCULATE Y COMPONENT OF TRANSPORT
C
      DO 30 I=2,IM
         DO 30 J=1,JM
            V(I,J)=(S(I,J) - S(I-1,J))/DS
   30 CONTINUE
   40 FORMAT(E12.5)
      RETURN
      END
```

```
      SUBROUTINE UZL(UM, ZM, TD, ZTM, CD, CH, Z0, FL)
C
C PURPOSE:
C               TO CALCULATE THE BULK AERODYNAMIC COEFFICIENTS FOR
C               MOMENTUM AND HEAT OVER A LAKE SURFACE AS FUNCTIONS
C               OF WIND SPEED AND AIR-SEA TEMPERATURE DIFFERENCE.
C
C ALGORITHM:
C               THERE IS AN OUTER ITERATION IN WHICH THE ROUGHNESS
C               LENGTH IS VARIED ACCORDING TO CHARNOCK'S FORMULA AND
C               AN INNER ITERATION IN WHICH THE STABILITY LENGTH
C               (MONIN-OBUKHOV LENGTH) IS VARIED ACCORDING TO THE
C               BUSINGER-DYER FORMULATION.  THE CONSTANT IN CHARMOCK'S
C               FORMULA IS CHOSEN SO THAT UNDER NEUTRAL CONDITIONS THE
C               10 M DRAG COEFFICIENT IS 0.0016.
C
C ARGUMENTS:
C  ON INPUT:
C               UM - WIND SPEED (M / S)
C               ZM - ANEMOMETER HEIGHT (M)
C               TD - AIR-SEA TEMPERATURE DIFFERENCE (DEG K)
C               ZTM - THERMOMETER HEIGHT
C                     (INITIALLY ASSUMED EQUAL TO ANEMOMETER HEIGHT)
C  ON OUTPUT:
C               CD - BULK AERODYNAMIC COEFFICIENT FOR MOMENTUM
C               CH - BULK AERODYNAMIC COEFFICIENT FOR HEAT
C               Z0 - ROUGHNESS LENGTH (M)
C               FL - STABILITY LENGTH (M)
C
C HISTORY:
C               WRITTEN BY J. R. BENNETT AND J. D. BOYD, 1979, GLERL,
C               ANN, ARBOR, MI;  BASED ON A CONSTANT ROUGHNESS VERSION
C               WRITTEN BY PAUL LONG AND WILL SHAFFER OF THE TECHNIQUES
C               DEVELOPMENT LABORATORY, NOAA, SILVER SPRINGS, MD.
C
C
C ***************************************************************************
      DATA C1, C2, C3 /.684E-4, 4.28E-3, -4.43E-4/
      DATA B1, B2, B3 /1.7989E-3, 4.865E-4, 3.9028E-5/
      EPS = .01
      IF (UM .LT. .001) UM = .001
      FK = .35
      TBAR = 278.
      ALPHA = 4.7
      BETA = .74
      GAMM = 15.
      GAMT = 9.
      UST1 = 0.04 * UM
      H = ZM
      DTHETA = TD
      IF (ABS(DTHETA) .LT. 1.E-7) DTHETA = SIGN(1.E-7,DTHETA)
C
C INITIAL GUESS FOR Z0
C
      Z0 = .00459 * UST1 * UST1
```

```
      S = UM * UM * TBAR / (9.8*DTHETA)
      IF (ABS(S) .GT. 1.E6) S = SIGN(1.E6,S)
      X = ALOG(H/Z0)
C
C       INITIAL GUESS FOR L
C
      FL = S / X
      DO 60 ITER = 1, 20
        X = ALOG(H/Z0)
        IF (ABS(FL) .GT. 3.E6) FL = SIGN(3.E6,FL)
        IF (FL .GT. 0.) GOTO 20
C
C UNSTABLE SECTION (L LT 0 OR DT LT 0)
C
        FLI = 1. / FL
C
C ASSUME 5 ITERATIONS SUFFICIENT
C
        DO 10 I = 1, 5
          X1 = GAMT * FLI
          ARG1 = SQRT(1. - X1*H)
          ARG2 = SQRT(1. - X1*Z0)
          A = BETA * ALOG((ARG1 - 1.)*(ARG2 + 1.)/((ARG1 + 1.)*(ARG2 -
     1    1.)))
          X1 = GAMM * FLI
          ARG1 = (1. - X1*H) ** (.25)
          ARG2 = (1. - X1*Z0) ** (.25)
          B = ALOG((ARG1 - 1.)*(ARG2 + 1.)/((ARG1 + 1.)*(ARG2 - 1.))) +
     1    2. * (ATAN(ARG1) - ATAN(ARG2))
          FL = S * A / (B*B)
          IF (ABS(FL) .GT. 3.E6) FL = SIGN(3.E6,FL)
          FLI = 1. / FL
   10   CONTINUE
        GOTO 50
C
C STABLE SECTION
C
C TRY MILDLY STABLE-
C
   20   CONTINUE
        AA = X * X
        X1 = H - Z0
        BB = 9.4 * X1 * X - .74 * S * X
        CC = 4.7 * X1
        CC = CC * CC - CC * S
        ROOT = BB * BB - 4. * AA * CC
        IF (ROOT .LT. 0.) GOTO 30
        FL = (-BB + SQRT(ROOT)) / (2.*AA)
        IF (FL .LE. H) GOTO 30
        B = X + 4.7 * X1 / FL
        A = BETA * X + 4.7 * X1 / FL
        GOTO 50
C
C STRONGLY STABLE-
C
   30   CONTINUE
```

```
      IF (FL .LE. Z0) FL = Z0 + 1.E-5
      DO 40 I = 1, 5
        ARG1 = FL / Z0
        X1 = ALOG(ARG1)
        X2 = ALOG(H/FL)
        ARG1 = 1. - 1. / ARG1
        A = .74 * X1 + 4.7 * ARG1 + 5.44 * X2
        B = X1 + 4.7 * ARG1 + 5.7 * X2
        FL = A * S / (B*B)
        IF (FL .LE. Z0) FL = Z0 + 1.E-5
        IF (FL .GT. H) FL = H
   40 CONTINUE
C
C CALCULATE USTAR AND Z0NEW
C
   50 CONTINUE
      TSTAR = FK * DTHETA / A
      USTAR = FK * UM / B
      Z0NEW = .00459 * USTAR * USTAR
      IF (ITER .GT. 5 .AND. ABS((USTAR - UST1)/UST1) .LT. EPS)
     1   GOTO 80
      UST1 = USTAR
      Z0 = Z0NEW
   60 CONTINUE
C
C IF COME HERE, TOO MANY ITERATIONS (UGH - UGH)
C
      WRITE(*,70)
   70 FORMAT ('0TOO MANY ITERATIONS ON Z0 IN SUBROUTINE UZL - CHECK ',
     1        'METEOROLOGICAL DATA - PROGRAM TERMINATED')
      STOP
   80 CONTINUE
      Z0 = Z0NEW
      CD = (USTAR/UM) ** 2
      CH = FK * FK / (A*B)
   90 RETURN
      END
```

```
      FUNCTION RLAT(X,Y)
C
C  PURPOSE:  TO RETURN LATITUDE OF A POINT ON THE GRID
C                      DESCRIBED BY THE COMMON BLOCK /GPARM/, GIVEN THE
C                      X AND Y DISPLACEMENTS FROM THE GRID ORIGIN
C  ARGUMENTS:
C                      X - X DISTANCE FROM THE GRID ORIGIN (M)
C                      Y - Y DISTANCE FROM THE GRID ORIGIN (M)
C                      RPARM, IPARM - ARRAYS CONTAINING BATHYMETRIC GRID
C                           PARAMETERS AS DESCRIBED IN SUBROUTINE RGRID
C
C  COMMON BLOCK:  /GPARM/RPARM(23),IPARM(54),ZPARM(2)
C
C  Last Date of Revision : September 11, 1985
C
C ************************************************************************
      COMMON /GPARM/ RPARM(23), IPARM(54), ZPARM(2)
      PI = ATAN2(0.,-1.)
      ALPHA = RPARM(7) * PI / 180.
C
C  TRANSFORM THE POINTS TO THE 'PRIMED' COORDINATE SYSTEM,
C  IE., THAT OF THE STANDARD BATHYMETRIC GRID
C
C  FIRST TRANSLATE
C
      XX = X - ZPARM(1) * RPARM(3)
      YY = Y - ZPARM(2) * RPARM(3)
C
C  NOW ROTATE
C
      XP=(XX*COS(ALPHA)-YY*SIN(ALPHA))/(1000.)
      YP=(YY*COS(ALPHA)+XX*SIN(ALPHA))/(1000.)
      DLAT = RPARM(20) * XP + RPARM(21) * YP + RPARM(22) * XP * YP +
     1       RPARM(23) * XP ** 2
      RLAT = RPARM(1) + DLAT
      RETURN
      END
```

```
C ******************************************************************

      FUNCTION TAU(T, I, J, K)
C
C PURPOSE :
C                TO RETURN WIND STRESS DIVIDED BY WATER DENSITY IN
C                METERS**2/SECOND**2 FOR GRID BOX I,J AT TIME T.
C                THE PARAMETER K INDICATES WHICH COMPONENT OF WIND
C                STRESS IS RETURNED.  THE X-COMPONENT OF WIND STRESS
C                IS CALCULATED AT THE MIDDLE OF THE RIGHT SIDE OF THE
C                GRID BOX.  THE Y-COMPONENT IS CALCULATED AT THE MIDDLE
C                OF THE TOP SIDE.  BOTH COMPONENTS ARE LINEAR FUNCTIONS
C                OF X AND Y.
C
C ALGORITHM:     THIS FUNCTION READS METEOROLOGICAL DATA FROM A FILE,
C                CONVERTS IT TO WIND STRESS, AND INTERPOLATES THE WIND
C                STRESS IN TIME AND SPACE.  ALL DATA FOR THE SAME TIME
C                ARE GROUPED TOGETHER, WITH A MAXIMUM OF 25 STATIONS IN
C                A GROUP.  AS DATA RECORDS ARE READ, SUBROUTINE UZL IS
C                USED TO CONVERT METEOROLOGICAL MEASUREMENTS TO SUR-
C                FACE STRESS FOR EACH STATION.  THE TWO COMPONENTS OF
C                THE WIND STRESS ARE ASSUMED TO BE LINEAR FUNCTIONS OF
C                X AND Y AND THE BEST-FIT LINEAR SURFACE FOR THE GIVEN
C                DATA POINTS IS CALCULATED.  EACH TIME TAU IS CALLED,
C                THE TIME PARAMETER IS CHECKED AGAINST THE TIME OF THE
C                LAST SET OF METOROLOGICAL DATA READ.  IF IT IS GREATER,
C                ANOTHER GROUP IS READ.  IF IT IS LESS, THE APPROPRIATE
C                WIND STRESS COMPONENT IS CALCULATED USING LINEAR INTER-
C                POLATION BETWEEN THE LAST TWO LINEAR SURFACES COM-
C                PUTED FOR THAT COMPONENT.  IF THE END-OF-FILE IS EN-
C                COUNTERED, WIND STRESS IS CALCULATED WITH THE LAST
C                LINEAR SURFACE COEFFICIENTS.
C
C                THE X-COMPONENT OF STRESS FOR GRID BOX I,J HAS CO-
C                ORDINATES (I*DELTA,(J-1/2)*DELTA) RELATIVE TO THE GRID
C                ORIGIN, AND THE Y-COMPONENT OF STRESS HAS COORDINATES
C                ((I-1/2)*DELTA,J*DELTA), WHERE DELTA IS THE GRID SIZE.
C
C                MANY OF THE VARIABLES USED IN THIS FUNCTION ARE ASSUM-
C                ED TO HAVE RETAINED THEIR VALUES FROM THE PREVIOUS
C                CALL.  IF YOUR FORTRAN SYSTEM DOES NOT DO THIS AUTO-
C                MATICALLY, PROVISION MUST BE MADE TO RETAIN VALUES
C                FOR THE VARIABLES:ISTA, AOLD1, BOLD1, COLD1, AOLD2,
C                BOLD2, COLD2, ANEW1, BNEW1, CNEW1, ANEW2, BNEW2, CNEW2,
C                TOLD, TNEW, TLAST, RLAT, RLON, Z, TA, TW, WS, WD.
C
C  ARGUMENTS:
C                T - TIME IN SECONDS FROM BEGINNING OF RUN AT WHICH
C                    STRESS IS TO BE CALCULATED
C                I - FIRST INDEX OF GRID BOX FOR WHICH STRESS IS TO
C                    BE CALCULATED
C                J - SECOND INDEX OF GRID BOX FOR WHICH STRESS IS TO
C                    BE CALCULATED
C                K - SPECIFIES WHETHER X-COMPONENT OR Y-COMPONENT
C                    OF STRESS IS RETURNED.  IF K=1, THE X-COMPONENT
C                    OF STRESS AT THE MIDDLE RIGHT SIDE OF GRID BOX
```

```
C                   I, J IS RETURNED.  IF K=2, THE Y-COMPONENT OF
C                   STRESS AT THE MIDDLE TOP SIDE OF GRID BOX I, J
C                   IS RETURNED.
C
C INPUT:
C  LOGICAL UNIT 13 :
C      METEOROLOGICAL DATA FILE
C      CARD COLUMN        FORMAT      PARAMETER
C          1-10           G10.4     TLAST - TIME (HRS) FROM BEGINNING
C                                             OF RUN
C         11-20           G10.4     RLAT - LATITUDE IN DEGREES NORTH
C         21-30           G10.4     RLON - LONGITUDE IN DEGREES WEST
C         31-40           G10.4     Z - HEIGHT OF INSTRUMENTS (FT)
C         41-50           G10.4     TA - TEMPERATURE OF AIR (F)
C         51-60           G10.4     TW - TEMPERATURE OF WATER (F)
C         61-70           G10.4     WS - WIND SPEED (FT/S)
C         71-76            G6.0     WD - WIND DIRECTION (DEG)
C
C         ALL DATA FOR THE SAME TIME ARE GROUPED TOGETHER, WITH A
C         MAXIMUM OF 25 STATIONS IN A GROUP.
C
C   *NOTE THAT END OF FILE IS INDICATED BY A RECORD WITH A NEGATIVE
C         TIME.
C
C OUTPUT :
C  LOGICAL UNIT 6:
C      PRESENTATION OF METEOROLOGICAL DATA STATION BY STATION
C
C COMMON BLOCKS:
C              /GPARM/ RPARM(23),IPARM(54),ZPARM(2)
C              /ICE/ZWND(40,40),ZLKICE(20),NICEX1(20),NICEY1(20),
C                  NICEX2(20),NICEY2(20),IPOS1(20),IPOS2(20),AMIUO,ANICE,
C                  SPAICE,NICERG,LICERG
C
C SUBPROGRAMS :
C      FUNCTION XDIST - RETURNS X DISTANCE FROM GRID ORIGIN
C                         GIVEN LATITUDE AND LONGITUDE
C      FUNCTION YDIST - RETURNS Y DISTANCE FROM GRID ORIGIN
C                         GIVEN LATITUDE AND LONGITUDE
C      SUBROUTINE UZL - RETURNS DRAG COEFFICIENT CD
C
C HISTORY :
C             WRITTEN BY A. T. JESSUP, 1981, GLERL, ANN ARBOR, MI
C
C            *MODIFIED AUGUST, 1983 TO PROPERLY HANDLE CO-LINEAR
C             METEOROLOGICAL DATA POINTS
C
C  Last Date of Revision : September 11, 1985
C
C *************************************************************************
      DIMENSION X(25), U(2), Y(25), ATAU(25,2)
      LOGICAL XEQ, YEQ
      COMMON /GPARM/ RPARM(23), IPARM(54), ZPARM(2)
      COMMON /ICE/ZWND(40,40),ZLKICE(20),NICEX1(20),NICEY1(20),
     $ NICEX2(20),NICEY2(20),IPOS1(20),IPOS2(20),AMIUO,ANICE,
     $ SPAICE,NICERG,LICERG
```

```
      DATA RHOAW, ISTA,PINDEX /1.25E-3, 0,1/
      DATA LUN /10/
      DATA AOLD1, BOLD1, COLD1, AOLD2, BOLD2, COLD2, TOLD /7*0./
C
C  STATEMENT FUNCTION TO CALCULATE DETERMINANT OF 3 BY 3 MATRIX
C
      DET(A11, A21, A31, A12, A22, A32, A13, A23, A33) =
     1  A11 * A22 * A33 - A11 * A23 * A32 +
     2  A12 * A23 * A31 - A12 * A21 * A33 +
     3  A13 * A21 * A32 - A13 * A22 * A31
C
C  IF THIS IS THE FIRST TIME THROUGH, READ A RECORD
C
      IF (ZWND(I,J) .EQ. 0.0D0) RETURN
      IF (ISTA .EQ. 0) GOTO 30
C
C  CHECK IF NECESSARY TO READ ANOTHER RECORD
C
      IF (T .LT. TNEW .OR. TNEW .LT. 0.) GOTO 90
   10 AOLD1 = ANEW1
      BOLD1 = BNEW1
      COLD1 = CNEW1
      AOLD2 = ANEW2
      BOLD2 = BNEW2
      COLD2 = CNEW2
      TOLD = TNEW
   20 TNEW = TLAST
      IF (TNEW .LT. 0) GOTO 90
C
C  FIND ATAU, THE WIND STRESS FOR THE CURRENT STATION
C
      TD = TA - TW
      X(ISTA) = XDIST(RLAT,RLON)
      Y(ISTA) = YDIST(RLAT,RLON)
      U(1) = WS * COS((270. - WD - RPARM(6) - RPARM(7))*ATAN(1.)/45.)
      U(2) = WS * SIN((270. - WD - RPARM(6) - RPARM(7))*ATAN(1.)/45.)
      CALL UZL(WS, Z, TD, Z, CD, CH, Z0, FL)
      CDD = CD * 1.E3
      TTLAST = TLAST / 3600.
      XKM = X(ISTA) / 1000.
      YKM = Y(ISTA) / 1000.
      ZUNI = Z*3.282
      TAUNI = TA*9./5. +32.
      TWUNI = TW*9./5. + 32.
      WSUNI = WS*2.238
      IF(PINDEX.EQ.1)WRITE(*,843)
      PINDEX=PINDEX+1
      WRITE(*,150) TTLAST, RLAT, RLON,ZUNI,TAUNI,TWUNI,WSUNI,WD
      ATAU(ISTA,1) = CD * RHOAW * U(1) * WS
      ATAU(ISTA,2) = CD * RHOAW * U(2) * WS
   30 READ (LUN,130) TLAST, RLAT, RLON, Z, TA, TW, WS, WD
      Z = Z / 3.281
      TA = 5./9.*(TA - 32.)
      TW = 5./9.*(TW - 32.)
      WS = WS / 3.281
      TLAST = TLAST * 3600.
```

```
      IF (T .LT. TLAST .AND. ISTA .EQ. 0) GOTO 120
      ISTA = ISTA + 1
C
C  IF FIRST TIME THROUGH, FIND ATAU
C
      IF (ISTA .EQ. 1) GOTO 20
C
C  CHECK IF LAST RECORD AT TIME TLAST HAS BEEN READ
C
      IF (TLAST .EQ. TNEW) GOTO 20
C
C  NOW FIND THE BEST-FIT LINEAR SURFACE
C
      SX = 0.
      SY = 0.
      SXY = 0.
      SX2 = 0.
      SY2 = 0.
      SXTAU1 = 0.
      SYTAU1 = 0.
      SATAU1 = 0.
      SXTAU2 = 0.
      SYTAU2 = 0.
      SATAU2 = 0.
      N = ISTA - 1
      AN = FLOAT(N)
      XEQ = .TRUE.
      YEQ = .TRUE.
      DO 40 IN = 1, N
        SX = SX + X(IN)
        SY = SY + Y(IN)
        SXY = SXY + X(IN) * Y(IN)
        SX2 = SX2 + X(IN) ** 2
        SY2 = SY2 + Y(IN) ** 2
        SXTAU1 = SXTAU1 + X(IN) * ATAU(IN,1)
        SYTAU1 = SYTAU1 + Y(IN) * ATAU(IN,1)
        SXTAU2 = SXTAU2 + X(IN) * ATAU(IN,2)
        SYTAU2 = SYTAU2 + Y(IN) * ATAU(IN,2)
        SATAU1 = SATAU1 + ATAU(IN,1)
        SATAU2 = SATAU2 + ATAU(IN,2)
        IF (X(IN) .NE. X(1)) XEQ = .FALSE.
        IF (Y(IN) .NE. Y(1)) YEQ = .FALSE.
   40 CONTINUE
C
C  CALCULATE COEFFICIENTS ANEW, BNEW, CNEW, WHERE
C                        TAU = ANEW * X + BNEW * Y + CNEW
C                        FOR EACH COMPONENT.
C
      ANEW1 = 0.
      ANEW2 = 0.
      BNEW1 = 0.
      BNEW2 = 0.
      CNEW1 = SATAU1 / AN
      CNEW2 = SATAU2 / AN
C
C  CHECK FOR ONE DATA POINT ONLY
```

```
C
      IF (XEQ .AND. YEQ) GOTO 80
C
C  CHECK IF DATA POINTS ARE CO-LINEAR
C
      IF(N .EQ. 2) GOTO 60
      DO 50 IN = 3, N
        A = SQRT((X(1) - X(IN))**2 + (Y(1) - Y(IN))**2)
        B = SQRT((X(1) - X(IN-1))**2 + (Y(1) - Y(IN-1))**2)
        C = SQRT((X(IN) - X(IN-1))**2 + (Y(IN) - Y(IN-1))**2)
        S = (A + B + C) / 2.
        D = SQRT(S * (S - A) * (S - B) * (S - C)) / AMAX1(A, B, C, 1.)
C
C  IF NOT CO-LINEAR, GOTO 75
C
        IF (D .GT. 100.) GOTO 75
   50 CONTINUE
C
C  CALCULATE COEFFICIENTS FOR CO-LINEAR DATA POINTS
C
C  60 WRITE(*,175)
   60 CONTINUE
      DO 70 IN=2,N
        IF (X(IN) .EQ. X(1) .AND. Y(IN) .EQ. Y(1)) GOTO 70
        JN=IN
   70 CONTINUE
      ALPHA = ATAN2(Y(JN) - Y(1), X(JN) - X(1))
      CN = COS(ALPHA)
      SN = SIN(ALPHA)
      SS = CN * SX + SN * SY
      SS2 = CN * CN * SX2 + 2. * CN * SN * SXY + SN * SN * SY2
      SSTAU1 = CN * SXTAU1 + SN * SYTAU1
      SSTAU2 = CN * SXTAU2 + SN * SYTAU2
      D = AN * SS2 - SS * SS
      ANEW1 = CN * (AN * SSTAU1 - SS * SATAU1) / D
      ANEW2 = CN * (AN * SSTAU2 - SS * SATAU2) / D
      BNEW1 = SN * (AN * SSTAU1 - SS * SATAU1) / D
      BNEW2 = SN * (AN * SSTAU2 - SS * SATAU2) / D
      CNEW1 = (SS2 * SATAU1 - SSTAU1 * SS) / D
      CNEW2 = (SS2 * SATAU2 - SSTAU2 * SS) / D
      GOTO 80
C
C  CALCULATE COEFFICIENTS FOR BEST-FIT LINEAR SURFACE
C
   75 D = DET(SX2, SXY, SX, SXY, SY2, SY, SX, SY, AN)
      ANEW1 = DET(SXTAU1, SYTAU1, SATAU1, SXY, SY2, SY, SX, SY, AN) / D
      ANEW2 = DET(SXTAU2, SYTAU2, SATAU2, SXY, SY2, SY, SX, SY, AN) / D
      BNEW1 = DET(SX2, SXY, SX, SXTAU1, SYTAU1, SATAU1, SX, SY. AN) / D
      BNEW2 = DET(SX2, SXY, SX, SXTAU2, SYTAU2, SATAU2, SX, SY, AN) / D
      CNEW1 = DET(SX2, SXY, SX, SXY, SY2, SY, SXTAU1, SYTAU1, SATAU1)/D
      CNEW2 = DET(SX2, SXY, SX, SXY, SY2, SY, SXTAU2, SYTAU2, SATAU2)/D
   80 CONTINUE
      ISTA = 1
      IF (T .GT. TNEW) GOTO 10
C
C  DETERMINE PROPER LOCATION XS,YS AND CALCULATE STRESS
```

```
C
      IF (XEQ .AND. YEQ) GOTO 80
C
C  CHECK IF DATA POINTS ARE CO-LINEAR
C
      IF(N .EQ. 2) GOTO 60
      DO 50 IN = 3, N
        A = SQRT((X(1) - X(IN))**2 + (Y(1) - Y(IN))**2)
        B = SQRT((X(1) - X(IN-1))**2 + (Y(1) - Y(IN-1))**2)
        C = SQRT((X(IN) - X(IN-1))**2 + (Y(IN) - Y(IN-1))**2)
        S = (A + B + C) / 2.
        D = SQRT(S * (S - A) * (S - B) * (S - C)) / AMAX1(A, B, C, 1.)
C
C  IF NOT CO-LINEAR, GOTO 75
C
        IF (D .GT. 100.) GOTO 75
   50 CONTINUE
C
C  CALCULATE COEFFICIENTS FOR CO-LINEAR DATA POINTS
C
C  60 WRITE(*,175)
   60 CONTINUE
      DO 70 IN=2,N
        IF (X(IN) .EQ. X(1) .AND. Y(IN) .EQ. Y(1)) GOTO 70
        JN=IN
   70 CONTINUE
      ALPHA = ATAN2(Y(JN) - Y(1), X(JN) - X(1))
      CN = COS(ALPHA)
      SN = SIN(ALPHA)
      SS = CN * SX + SN * SY
      SS2 = CN * CN * SX2 + 2. * CN * SN * SXY + SN * SN * SY2
      SSTAU1 = CN * SXTAU1 + SN * SYTAU1
      SSTAU2 = CN * SXTAU2 + SN * SYTAU2
      D = AN * SS2 - SS * SS
      ANEW1 = CN * (AN * SSTAU1 - SS * SATAU1) / D
      ANEW2 = CN * (AN * SSTAU2 - SS * SATAU2) / D
      BNEW1 = SN * (AN * SSTAU1 - SS * SATAU1) / D
      BNEW2 = SN * (AN * SSTAU2 - SS * SATAU2) / D
      CNEW1 = (SS2 * SATAU1 - SSTAU1 * SS) / D
      CNEW2 = (SS2 * SATAU2 - SSTAU2 * SS) / D
      GOTO 80
C
C  CALCULATE COEFFICIENTS FOR BEST-FIT LINEAR SURFACE
C
   75 D = DET(SX2, SXY, SX, SXY, SY2, SY, SX, SY, AN)
      ANEW1 = DET(SXTAU1, SYTAU1, SATAU1, SXY, SY2, SY, SX, SY, AN) / D
      ANEW2 = DET(SXTAU2, SYTAU2, SATAU2, SXY, SY2, SY, SX, SY, AN) / D
      BNEW1 = DET(SX2, SXY, SX, SXTAU1, SYTAU1, SATAU1, SX, SY, AN) / D
      BNEW2 = DET(SX2, SXY, SX, SXTAU2, SYTAU2, SATAU2, SX, SY, AN) / D
      CNEW1 = DET(SX2, SXY, SX, SXY, SY2, SY, SXTAU1, SYTAU1, SATAU1)/D
      CNEW2 = DET(SX2, SXY, SX, SXY, SY2, SY, SXTAU2, SYTAU2, SATAU2)/D
   80 CONTINUE
      ISTA = 1
      IF (T .GT. TNEW) GOTO 10
C
C  DETERMINE PROPER LOCATION XS,YS AND CALCULATE STRESS
```

```
C
   90 IF (K .EQ. 2) GOTO 100
      XS = I * RPARM(3)
      YS = (FLOAT(J) - .5) * RPARM(3)
      TAUNEW = ANEW1 * XS + BNEW1 * YS + CNEW1
      TAUOLD = AOLD1 * XS + BOLD1 * YS + COLD1
      GOTO 110
  100 XS = (FLOAT(I) - .5) * RPARM(3)
      YS = J * RPARM(3)
      TAUNEW = ANEW2 * XS + BNEW2 * YS + CNEW2
      TAUOLD = AOLD2 * XS + BOLD2 * YS + COLD2
C
C  INTERPOLATE IN TIME
C
  110 TAU = (TAUNEW - TAUOLD) / (TNEW - TOLD) * (T - TOLD) + TAUOLD
      GOTO 180
  120 WRITE(*,140) TLAST, T
      STOP
  130 FORMAT (7G10.4,G6.0)
  140 FORMAT (' TIME OF FIRST METEOROLOGICAL DATA', G10.4,
     1         ' SECONDS IS', ' GREATER THAN T = ', G10.4,
     2         ' - PROGRAM TERMINATED')
  150 FORMAT (4X, F6.1,2F7.2,F6.1,F8.1,3F6.1,F6.2)
  175 FORMAT(' THESE DATA POINTS ARE CO-LINEAR OR NEARLY CO-LINEAR')
 843  FORMAT(///' Meteorological Station Data Used in Lake Circulation'
     $ ,' Model'
     $/6X,'Time   Lat.  Long. Height  T-air  T-H2O     Wind'/
     $7X, 'hrs    deg   deg    ft      F      F     mph  deg'/)
  180 RETURN
      END
```

Function UZ

```
      FUNCTION UZ(Z,UM,CD,Z0,FL)
C
C  PURPOSE:
C
C         TO DETERMINE WIND SPEED AT HEIGHT Z GIVEN WIND SPEED (UM)
C         AND DRAG COEFFICIENT (CD) FROM ANOTHER HEIGHT AND THE
C         ROUGHNESS LENGTH (Z0) AND STABILITY LENGTH (FL) OF THE
C         PROFILE. USUALLY FUNCTION UZ IS USED AFTER SUBROUTINE UZL
C         IN ORDER TO FIND WIND SPEED AT A DIFFERENT HEIGHT THAN
C         THE OBSERVATION HEIGHT.
C
C  ALGORITHM:
C
C         THE WIND PROFILE IS ASSUMED TO CONFORM TO THE BUSINGER-
C         DYER FORMULATION USED IN SUBROUTINE UZL.
C
C  ARGUMENTS:
C
C         Z - HEIGHT AT WHICH WIND SPEED IS REQUIRED (METERS)
C         UM - WIND SPEED AT OBSERVATION HEIGHT (METERS PER SECOND)
C         CD - DRAG COEFFICIENT CORRESPONDING TO OBSERVATION HEIGHT
C         Z0 - ROUGHNESS LENGTH (METERS)
C         FL - STABILITY LENGTH (METERS)
C
C  HISTORY:
C
C         WRITTEN BY D.J SCHWAB, 1983, GLERL, ANN ARBOR, MI.
C         SEE SUBROUTINE UZL IN SCHWAB, BENNETT, AND JESSUP (1981)
C
C ******************************************************************
      IF(FL.GT.0.) GOTO 10
C
C  UNSTABLE PROFILE
C
      X1=15./FL
      ARG1=(1.-X1*Z)**0.25
      ARG2=(1.-X1*Z0)**0.25
      B=ALOG((ARG1-1.)*(ARG2+1.)/((ARG1+1.)*(ARG2-1.)))+
     1   2.*(ATAN(ARG1)-ATAN(ARG2))
      GOTO 30
C
C  STABLE SECTION
C
   10 IF(FL.LE.Z) GOTO 20
C
C  MILDLY STABLE PROFILE
C
      B=ALOG(Z/Z0)+4.7*(Z-Z0)/FL
      GOTO 30
C
C  STRONGLY STABLE PROFILE
C
   20 CONTINUE
      ARG1=FL/Z0
      X1=ALOG(ARG1)
```

```
      X2=ALOG(Z/FL)
      ARG1=1.-1./ARG1
      B=X1+4.7*ARG1+5.7*X2
C
C  CALCULATE USTAR AND UZ
C
   30 CONTINUE
      USTAR=UM*SQRT(CD)
      UZ=USTAR*B/0.35
      RETURN
      END
```

```
      FUNCTION XDIST(RLAT, RLON)
C
C  PURPOSE :       TO RETURN X DISTANCE IN METERS FROM THE GRID ORIGIN
C                  DESCRIBED BY THE COMMON BLOCK / GPARM /, GIVEN
C                  LATITUDE AND LONGITUDE
C  ARGUMENTS:
C                  RLAT - LATITUDE OF POINT
C                  RLON - LONGITUDE (WEST) OF POINT
C                  RPARM, IPARM - ARRAYS CONTAINING BATHYMETRIC GRID
C                         PARAMETERS AS DESCRIBED IN SUBROUTINE RGRID
C
C  COMMON BLOCK:  /GPARM/ RPARM(23),IPARM(54),ZPARM(2)
C
C  Last Date of Revision : September 11, 1985
C
C *********************************************************************
      COMMON /GPARM/ RPARM(23), IPARM(54), ZPARM(2)
      PI = ATAN2(0.,-1.)
      ALPHA = RPARM(7) * PI / 180.
      DLAT = RLAT - RPARM(1)
      DLON = RPARM(2) - RLON
C
C  FIND XPRIME, YPRIME - DISTANCES FROM THE ORIGIN OF THE STANDARD
C  BATHYMETRIC GRID
C
      XP = RPARM(8) * DLON + RPARM(9) * DLAT + RPARM(10) * DLON * DLAT +
     1     RPARM(11) * DLON ** 2
      YP = RPARM(12) * DLON + RPARM(13) * DLAT + RPARM(14) * DLAT *
     1     DLON + RPARM(15) * DLON ** 2
C
C  TRANSFORM TO 'UNPRIMED' SYSTEM
C
C  FIRST ROTATE
C
      XDIST = (XP*COS(ALPHA) + YP*SIN(ALPHA)) * 1000.
C
C  NOW TRANSLATE
C
      XDIST = XDIST + ZPARM(1) * RPARM(3)
      RETURN
      END
```

```
      FUNCTION YDIST(RLAT, RLON)
C
C  PURPOSE :      TO RETURN Y DISTANCE IN METERS FROM THE GRID ORIGIN
C                 DESCRIBED BY THE COMMON BLOCK / GPARM /, GIVEN
C                 LATITUDE AND LONGITUDE
C  ARGUMENTS:
C                 RLAT - LATITUDE OF POINT
C                 RLON - LONGITUDE (WEST) OF POINT
C                 RPARM, IPARM - ARRAYS CONTAINING BATHYMETRIC GRID
C                      PARAMETERS AS DESCRIBED IN SUBROUTINE RGRID
C
C  COMMON BLOCK:  /GPARM/ RPARM(23),IPARM(54),ZPARM(2)
C
C  Last Date of Revision : September 11, 1985
C
C ***********************************************************************
      COMMON /GPARM/ RPARM(23), IPARM(54), ZPARM(2)
      PI = ATAN2(0.,-1.)
      ALPHA = RPARM(7) * PI / 180.
      DLAT = RLAT - RPARM(1)
      DLON = RPARM(2) - RLON
C
C  FIND XPRIME, YPRIME - DISTANCES FROM THE ORIGIN OF THE STANDARD
C  BATHYMETRIC GRID
C
      XP = RPARM(8) * DLON + RPARM(9) * DLAT + RPARM(10) * DLON * DLAT +
     1     RPARM(11) * DLON ** 2
      YP = RPARM(12) * DLON + RPARM(13) * DLAT + RPARM(14) * DLAT *
     1     DLON + RPARM(15) * DLON ** 2
C
C  TRANSFORM TO 'UNPRIMED' SYSTEM
C
C  FIRST ROTATE
C
      YDIST = (YP*COS(ALPHA) - XP*SIN(ALPHA)) * 1000.
C
C  NOW TRANSLATE
C
      YDIST = YDIST + ZPARM(2) * RPARM(3)
      RETURN
      END
```

APPENDIX III

PROGRAM PSISET.F

Program (PSISET.F), used to calculate stream function values (LAKEINIT.PSI), is given in this appendix.

```
C     PROGRAM RLID
C
      DIMENSION D(75,75), S(75,75), RHS(75,75), SPD(75,75),
     &          U(75,75), V(75,75)
      COMMON /GPARM/ RPARM(23), IPARM(54), ZPARM(2)
      COMMON /ITPARM/ DSTMAX, STMIN, STMAX, FRAC, ITS
      COMMON /APARM/ NSTEPS, IM, JM, DS, DMAX, DMIN, IMM1, JMM1
      DATA IDIM, JDIM /75,75/
      DATA LUNB, LUNM /7, 8/
      DATA OM, FR /7.292-5, 2.E-3/
      DATA RELAX, ITMAX, CONV /1.6, 50, 1.E-3/
      PI = ATAN(1.) * 4.
C
C  OPEN METEROLOGICAL DATA FILE
C
      OPEN(LUNB, FILE='lakebath.dat', STATUS='OLD')
      OPEN(LUNM, FILE='lakewind.dat', STATUS='OLD')
      REWIND LUNM
C
C  OPEN AND READ BATHYMETRIC GRID INFORMATION
C
      CALL RGRID(LUNB, D, IDIM, JDIM)
C
C  READ CONTROL PARAMETERS
C
      READ (LUNB,100) DT, TT, RWD, CLWL, TLKQ, RLKQ
      NSTEPS = TT / DT
      DADD = (CLWL-RWD) / 3.281
      Z = Z / 3.281
      IM = IPARM(1)
      JM = IPARM(2)
      DS = RPARM(3)
      DMAX = RPARM(4)
      DMIN = RPARM(5) + DADD
      CALL PRNT(6, D, IDIM, IM, JM, 0.)
C
C  CALCULATE CORIOLIS PARAMETER AT CENTER OF GRID
C
      F = 2. * OM * SIN(RLAT(IM*DS/2.,JM*DS/2.)*PI/180.)
C     WRITE(6,110) (IPARM(I),I=5,54), DT, TT, DADD, NSTEPS, F
      IMM1 = IM - 1
      JMM1 = JM - 1
      IMM2 = IM - 2
      JMM2 = JM - 2
      DT = DT * 3600.
      FDT24 = F * DT / 24.
C
C      ADJUST RELAXATION FACTOR FOR GRID SIZE
C
      RELAX = RELAX / (1. + SIN(ACOS(0.5*(COS(PI/IMM2) + COS(PI/JMM2))
     1))
C
C  INTERPOLATE DEPTH TO STREAM FUNCTION POINTS
C  USING SPEED ARRAY FOR TEMPORARY STORAGE
C
      DO 10 I = 1, IM
        DO 10 J = 1, JM
          SPD(I,J) = 0.0
          S(I,J) = 0.
          IF (D(I,J) .LT. RPARM(5)) GO TO 10
          D(I,J) = D(I,J) + DADD
```

```
   10 RHS(I,J) = 0.
C
C IF WATER LEVEL INCREMENT RESULTS IN A NEGATIVE DMIN, STOP.
C
      IF (DMIN .LE. 0.0) GO TO 90
      DO 20 I = 1, IM
        DO 20 J = 1, JM
          SPD(I,J) = 0.999999 * DMIN
          IF (I .EQ. IM .OR. J .EQ. JM) GO TO 20
          IF (D(I,J) .LT. DMIN) GO TO 20
          IF (D(I + 1,J) .LT. DMIN) GO TO 20
          IF (D(I,J + 1) .LT. DMIN) GO TO 20
          IF (D(I + 1,J + 1) .LT. DMIN) GO TO 20
          SPD(I,J) = 0.25 * (D(I,J) + D(I + 1,J) + D(I,J + 1) + D(I + 1,
     1    J + 1))
   20 CONTINUE
C
C  STORE INVERSE DEPTH BACK IN D
C
      DMINI = 1. / DMIN
      DO 30 I = 1, IM
        DO 30 J = 1, JM
          D(I,J) = 1. / SPD(I,J)
   30 SPD(I,J) = 0.0
C
C  GET INITIAL CONDITIONS
C
      CALL INIT(D, S, IDIM, ILKQ, RLKQ)
C
C  MAIN ITERATION LOOP
C
      TIME=0.
      DO 80 N = 1, NSTEPS
        TIME = TIME+DT/2.
C
C  CALCULATE CURRENT SPEED AT CENTER OF GRID BOX I, J
C
        DO 40 I = 2, IMM1
          DO 40 J = 2, JMM1
            IF(D(I,J) .GT. DMINI .AND. D(I-1,J) .GT. DMINI .AND.
     1      D(I,J-1) .GT. DMINI .AND. D(I-1,J-1) .GT. DMINI) GOTO 40
            DU = 0.5 * (D(I,J) + D(I,J - 1))
            DV = 0.5 * (D(I,J) + D(I - 1,J))
            DUM = 0.5 * (D(I - 1,J) + D(I - 1,J - 1))
            DVM = 0.5 * (D(I,J - 1) + D(I - 1,J - 1))
            SPD(I,J) = (0.5/DS) * SQRT((((S(I,J) - S(I,J - 1)).*DU) + ((
     1      S(I - 1,J) - S(I - 1,J - 1))*DUM))**2 + (((S(I,J) - S(I - 1,
     2      J))*DV) + ((S(I,J - 1) - S(I - 1,J - 1))*DVM)).**2))
   40   CONTINUE
C
C  ITERATE TO CALCULATE STREAM FUNCTION AT NEXT TIME STEP WITH ALTER-
C  NATING  SWEEP DIRECTIONS
C
        DO 60 K = 1, ITMAX
          KK=K+N
          DSTMAX = 0.
          STMIN = 0.
          STMAX = 0.
          ITS = K
          DO 50 II = 1, IM
            I = II
```

```
            IF (MOD(KK,2) .EQ. 0) I = IM - II + 1
            DO 50 JJ = 1, JM
              J = JJ
              IF (MOD(KK,2) .EQ. 0) J = JM - JJ + 1
              IF (D(I,J) .GT. DMINI) GO TO 50
              DUP = 0.5 * (D(I,J + 1) + D(I,J))
              DVP = 0.5 * (D(I + 1,J) + D(I,J))
              SPDUP = 0.5 * (SPD(I + 1,J + 1) + SPD(I,J + 1))
              SPDVP = 0.5 * (SPD(I + 1,J + 1) + SPD(I + 1,J))
              DU = 0.5 * (D(I,J) + D(I,J - 1))
              DV = 0.5 * (D(I,J) + D(I - 1,J))
              SPDU = 0.5 * (SPD(I + 1,J) + SPD(I,J))
              SPDV = 0.5 * (SPD(I,J + 1) + SPD(I,J))
              DCENT = DVP + DV + DUP + DU
C
C     LAPLACIAN TERM
C
              TERM1 = DVP * S(I + 1,J) + DV * S(I - 1,J) + DUP * S(I,J +
     1         1) + DU * S(I,J - 1) - DCENT * S(I,J)
C
C     ARAKAWA'S JACOBIAN
C
              TERM2 = S(I + 1,J) * (D(I,J + 1) + D(I + 1,J + 1) - D(I,J
     1        - 1) - D(I + 1,J - 1)) + S(I - 1,J) * (-D(I,J + 1) - D(I -
     2         1,J + 1) + D(I,J - 1) + D(I - 1,J - 1)) + S(I,J + 1) * (-
     3        D(I + 1,J) - D(I + 1,J + 1) + D(I - 1,J) + D(I - 1,J + 1))
     4        +S(I,J - 1) * (D(I + 1,J) + D(I + 1,J - 1) - D(I - 1,J) -
     5        D(I - 1,J - 1)) + S(I + 1,J + 1) * (-D(I + 1,J) + D(I,J +
     6        1)) + S(I + 1,J - 1) * (D(I + 1,J) - D(I,J - 1)) + S(I -
     7        1,J + 1) * (D(I - 1,J) - D(I,J + 1)) + S(I - 1,J - 1) * (-
     8        D(I - 1,J) + D(I,J - 1))
C
C     FRICTION TERM
C
              TYP = -DVP * (S(I + 1,J) - S(I,J)) * FR * SPDVP
              TYM = -DV * (S(I,J) - S(I - 1,J)) * FR * SPDV
              TXP = DUP * (S(I,J + 1) - S(I,J)) * FR * SPDUP
              TXM = DU * (S(I,J) - S(I,J - 1)) * FR * SPDU
              TERM3 = (DVP*TYP - DV*TYM - DUP*TXP + DU*TXM)
C
C     SET RIGHT HAND SIDE THE FIRST TIME THROUGH
C
              IF (K .EQ. 1) RHS(I,J) = TERM1 - FDT24 * TERM2 + DT * 0.5
     1        * TERM3 + DT * DS * (DVP*TAU(TIME,I + 1,J,2) - DV*TAU(
     2        TIME,I,J,2) - DUP*TAU(TIME,I,J + 1,1) + DU*TAU(TIME,I,J,1)
     3        )
              IF (K .EQ. 1) GO TO 50
C
C     CALCULATE NEW STREAM FUNCTION
C
              D4 = DCENT + FR * 0.5 * DT * (DVP**2*SPDVP + DV**2*SPDV +
     1        DUP**2*SPDUP + DU**2*SPDU)
              DST = (TERM1 + FDT24*TERM2 - 0.5*DT*TERM3 - RHS(I,J)) / D4
              S(I,J) = S(I,J) + RELAX * DST
              DSTMAX = AMAX1(DSTMAX,ABS(DST))
              STMIN = AMIN1(STMIN,S(I,J))
              STMAX = AMAX1(STMAX,S(I,J))
   50       CONTINUE
C
C     CALCULATE RELATIVE CHANGE IN STREAM FUNCTION FOR ALL ITERATIONS BUT
C     THE FIRST
```

```
C
          IF (K .EQ. 1) GO TO 60
          IF (STMAX .EQ. STMIN) GO TO 70
          FRAC = DSTMAX / (STMAX - STMIN)
          IF (FRAC .LE. CONV) GO TO 70
   60   CONTINUE
   70   CONTINUE
C
C  UPDATE TIME
C
        TIME=N*DT
C
C  CALL OUTPUT ROUTINE
C
          CALL OUTP(TIME, D, S, SPD, IDIM)
C
C  CHECK FOR STEADY STATE
C
 CALL COMPARE(TIME, D, S, SPD, IDIM, U, V)
C
C  END MAIN ITERATION LOOP
C
   80 CONTINUE
      GO TO 130
   90 WRITE(6,120) DADD
      GO TO 130
  100 FORMAT (4G8.2, 2G8.0)
C 110 FORMAT ('1RIGID LID CIRCULATION MODEL FOR ',
C    1        50A1/' TIME STEP(H): DT= ', F10.2/
C    1        ' DURATION OF RUN(H): TT= ', F7.2/
C    2        ' MEAN WATER LEVEL(M) (RELATIVE TO L.W.D.): DADD= ', F5.2/
C    3        ' NUMBER OF TIME STEPS: NSTEPS=', I6/
C    4        ' CORIOLIS PARAMETER (S**-1): F= ', E10.3)
  120 FORMAT (' THE WATER LEVEL INCREMENT', F8.2, ' RESULTS IN A',
     1        ' NEGATIVE MINUMUM DEPTH - PROGRAM TERMINATED')
  130 CONTINUE
      STOP
      END
C
      SUBROUTINE RGRID(LUN, D, IDIM, JDIM)
      DIMENSION D(IDIM,JDIM)
      COMMON /GPARM/ RPARM(23), IPARM(54), ZPARM(2)
C
C  REWIND BATHYMETRY FILE
C
      REWIND LUN
      READ (LUN,30) (IPARM(I),I=5,54), IPARM(1), IPARM(2),
     1(RPARM(I),I=1,6), ZPARM(1), ZPARM(2), RPARM(7)
      READ (LUN,60) (RPARM(I),I=8,23)
      IM = IPARM(1)
      JM = IPARM(2)
      RPARM(3) = RPARM(3) / 3.281
      RPARM(4) = RPARM(4) / 3.281
      RPARM(5) = RPARM(5) / 3.281
      IF (IPARM(1) .GT. IDIM .OR. IPARM(2) .GT. JDIM) GO TO 10
      READ (LUN,40) ((D(I,J),I=1,IM),J=1,JM)
      DO 80 I=1,IM
  DO 80 J=1,JM
     D(I,J) = D(I,J) / 3.281
   80 CONTINUE
      RETURN
```

```
   10 DO 20 I = 1, 54
   20 IPARM(I) = 0
      WRITE(6,50)
   30 FORMAT (50(A1)/2I5, 2F12.7, 3F5.0, F6.2, 3F7.3)
   40 FORMAT (19F4.0, 4X)
   50 FORMAT (' BATHYMETRIC GRID TOO LARGE - INCREASE DIMENSIONS OF',
     1        ' NDEPTH AND DEPTH IN MAIN PROGRAM')
   60 FORMAT (4E15.6, 20X)
   70 RETURN
      END
C
      SUBROUTINE PRNT(LUN, A, IDIM, IMAX, JMAX, SPVAL)
      DIMENSION INTEG(30), A(IDIM,1)
      NCOL = 19
C
C  AMAX=MAXIMUM ABSOLUTE VALUE OF ARRAY A
C
      AMAX = 0.0
      DO 10 I = 1, IMAX
        DO 10 J = 1, JMAX
          IF (A(I,J) .EQ. SPVAL) GO TO 10
          AMAX = AMAX1(AMAX,ABS(A(I,J)))
   10 CONTINUE
C
C  NOW FIND THE POWER OF TEN BY WHICH WE MUST MULTIPLY AMAX
C  SO THAT IT FALLS BETWEEN 100 AND 1000.
C
C  INITIALLY THE POWER IS ZERO
C
      NP = 0
      IF (AMAX .EQ. 0) GO TO 20
C
C  TAKE BASE 10 LOGARITHM OF AMAX
C
      AP = ALOG10(AMAX)
C
C  IF AMAX IS GREATER THAN 1000, NP IS NEGATIVE
C
      IF (AP .GT. 3.) NP = -IFIX(AP - 2.)
C
C  IF AMAX IS LESS THAN 100, NP IS POSITIVE
C
      IF (AP .LT. 2.) NP = IFIX(3. - AP)
   20 CONTINUE
C
C  PRINT THE GRID
C
      I1 = 1
      II = (IMAX - 1) / NCOL + 1
      IRMDR = IMAX - NCOL * (II - 1)
      DO 50 L = 1, II
C
C  WHEN L=II ONLY PRINT IRMDR VALUES
C
        IF (L .EQ. II) NCOL = IRMDR
C
C  PRINT THE POWER
C
        WRITE(LUN,60) NP
        I2 = I1 + NCOL - 1
        DO 40 JJ = 1, JMAX
```

```
          J = JMAX - JJ + 1
          DO 30 I = I1, I2
            I3 = 1 + I - I1
            INTEG(I3) = -9999
            IF (A(I,J) .NE. SPVAL) INTEG(I3) = INT(A(I,J)*10.**MP +
     1      SIGN(0.5,A(I,J)*10.**MP))
   30     CONTINUE
          WRITE(LUN,70) (INTEG(I),I=1,NCOL)
   40   CONTINUE
        I1 = I2 + 1
   50 CONTINUE
   60 FORMAT ('0 VALUES MULTIPLIED BY 10**', I3)
   70 FORMAT (' ', 30I4)
      RETURN
      END
C
      SUBROUTINE UZL(UM, ZM, TD, ZTM, CD, CH, Z0, FL)
      DATA C1, C2, C3 /.684E-4, 4.28E-3, -4.43E-4/
      DATA B1, B2, B3 /1.7989E-3, 4.865E-4, 3.9028E-5/
      EPS = .01
      IF (UM .LT. .001) UM = .001
      FK = .35
      TBAR = 278.
      ALPHA = 4.7
      BETA = .74
      GAMM = 15.
      GAMT = 9.
      UST1 = 0.04 * UM
      H = ZM
      DTHETA = TD
      IF (ABS(DTHETA) .LT. 1.E-7) DTHETA = SIGN(1.E-7,DTHETA)
C
C INITIAL GUESS FOR Z0
C
      Z0 = .00459 * UST1 * UST1
      S = UM * UM * TBAR / (9.8*DTHETA)
      IF (ABS(S) .GT. 1.E6) S = SIGN(1.E6,S)
      X = ALOG(H/Z0)
C
C        INITIAL GUESS FOR L
C
      FL = S / X
      DO 60 ITER = 1, 20
        X = ALOG(H/Z0)
        IF (ABS(FL) .GT. 3.E6) FL = SIGN(3.E6,FL)
        IF (FL .GT. 0.) GO TO 20
C
C UNSTABLE SECTION (L LT 0 OR DT LT 0)
C
        FLI = 1. / FL
C
C ASSUME 5 ITERATIONS SUFFICIENT
C
        DO 10 I = 1, 5
          X1 = GAMT * FLI
          ARG1 = SQRT(1. - X1*H)
          ARG2 = SQRT(1. - X1*Z0)
          A = BETA * ALOG((ARG1 - 1.)*(ARG2 + 1.)/((ARG1 + 1.)*(ARG2 -
     1    1.)))
          X1 = GAMM * FLI
          ARG1 = (1. - X1*H) ** (.25)
```

```
            ARG2 = (1. - X1*Z0) ** (.25)
            B = ALOG((ARG1 - 1.)*(ARG2 + 1.)/((ARG1 + 1.)*(ARG2 - 1.))) +
     1      2. * (ATAN(ARG1) - ATAN(ARG2))
            FL = S * A / (B*B)
            IF (ABS(FL) .GT. 3.E6) FL = SIGN(3.E6,FL)
            FLI = 1. / FL
   10     CONTINUE
          GO TO 50
C
C STABLE SECTION
C
C TRY MILDLY STABLE-
C
   20     CONTINUE
          AA = X * X
          X1 = H - Z0
          BB = 9.4 * X1 * X - .74 * S * X
          CC = 4.7 * X1
          CC = CC * CC - CC * S
          ROOT = BB * BB - 4. * AA * CC
          IF (ROOT .LT. 0.) GO TO 30
          FL = (-BB + SQRT(ROOT)) / (2.*AA)
          IF (FL .LE. H) GO TO 30
          B = X + 4.7 * X1 / FL
          A = BETA * X + 4.7 * X1 / FL
          GO TO 50
C
C STRONGLY STABLE-
C
   30     CONTINUE
          IF (FL .LE. Z0) FL = Z0 + 1.E-5
          DO 40 I = 1, 5
            ARG1 = FL / Z0
            X1 = ALOG(ARG1)
            X2 = ALOG(H/FL)
            ARG1 = 1. - 1. / ARG1
            A = .74 * X1 + 4.7 * ARG1 + 5.44 * X2
            B = X1 + 4.7 * ARG1 + 5.7 * X2
            FL = A * S / (B*B)
            IF (FL .LE. Z0) FL = Z0 + 1.E-5
            IF (FL .GT. H) FL = H
   40     CONTINUE
C
C CALCULATE USTAR AND ZONEW
C
   50     CONTINUE
          TSTAR = FK * DTHETA / A
          USTAR = FK * UM / B
          ZONEW = .00459 * USTAR * USTAR
          IF (ITER .GT. 5 .AND. ABS((USTAR - UST1)/UST1) .LT. EPS)
     1      GO TO 80
          UST1 = USTAR
          Z0 = ZONEW
   60 CONTINUE
C
C IF COME HERE, TOO MANY ITERATIONS (UGH - UGH)
C
      WRITE(6,70)
   70 FORMAT ('0TOO MANY ITERATIONS ON Z0 IN SUBROUTINE UZL - CHECK ',
     1        'METEOROLOGICAL DATA - PROGRAM TERMINATED')
      STOP
```

```
   80 CONTINUE
      Z0 = Z0NEW
      CD = (USTAR/UM) ** 2
      CH = FK * FK / (A*B)
   90 RETURN
      END
C
      FUNCTION UZ(Z,UM,CD,Z0,FL)
      IF(FL.GT.0.) GO TO 10
C
C  UNSTABLE PROFILE
C
      X1=15./FL
      ARG1=(1.-X1*Z)**0.25
      ARG2=(1.-X1*Z0)**0.25
      B=ALOG((ARG1-1.)*(ARG2+1.)/((ARG1+1.)*(ARG2-1.)))+
     1   2.*(ATAN(ARG1)-ATAN(ARG2))
      GO TO 30
C
C  STABLE SECTION
C
   10 IF(FL.LE.Z) GO TO 20
C
C  MILDLY STABLE PROFILE
C
      B=ALOG(Z/Z0)+4.7*(Z-Z0)/FL
      GO TO 30
C
C  STRONGLY STABLE PROFILE
C
   20 CONTINUE
      ARG1=FL/Z0
      X1=ALOG(ARG1)
      X2=ALOG(Z/FL)
      ARG1=1.-1./ARG1
      B=X1+4.7*ARG1+5.7*X2
C
C  CALCULATE USTAR AND UZ
C
   30 CONTINUE
      USTAR=UM*SQRT(CD)
      UZ=USTAR*B/0.35
      RETURN
      END
C
      FUNCTION TAU(T, I, J, K)
      DIMENSION X(25), U(2), Y(25), ATAU(25,2)
      LOGICAL XEQ, YEQ
      COMMON /GPARM/ RPARM(23), IPARM(54), ZPARM(2)
      DATA RHOAW, ISTA /1.25E-3, 0/
      DATA LUN /8/
      DATA AOLD1, BOLD1, COLD1, AOLD2, BOLD2, COLD2, TOLD /7*0./
C
C  STATEMENT FUNCTION TO CALCULATE DETERMINANT OF 3 BY 3 MATRIX
C
      DET(A11, A21, A31, A12, A22, A32, A13, A23, A33) =
     1  A11 * A22 * A33 - A11 * A23 * A32 +
     2  A12 * A23 * A31 - A12 * A21 * A33 +
     3  A13 * A21 * A32 - A13 * A22 * A31
C
C  IF THIS IS THE FIRST TIME THROUGH, READ A RECORD
```

```
C
      IF (ISTA .EQ. 0) GO TO 30
C
C  CHECK IF NECESSARY TO READ ANOTHER RECORD
C
      IF (T .LT. TNEW .OR. TNEW .LT. 0.) GO TO 90
   10 AOLD1 = ANEW1
      BOLD1 = BNEW1
      COLD1 = CNEW1
      AOLD2 = ANEW2
      BOLD2 = BNEW2
      COLD2 = CNEW2
      TOLD = TNEW
   20 TNEW = TLAST
      IF (TNEW .LT. 0) GO TO 90
C
C  FIND ATAU, THE WIND STRESS FOR THE CURRENT STATION
C
      TD = TA - TW
      X(ISTA) = XDIST(RLAT,RLON)
      Y(ISTA) = YDIST(RLAT,RLON)
      U(1) = WS * COS((270. - WD - RPARM(6) - RPARM(7))*ATAN(1.)/45.)
      U(2) = WS * SIN((270. - WD - RPARM(6) - RPARM(7))*ATAN(1.)/45.)
      CALL UZL(WS, Z, TD, Z, CD, CH, Z0, FL)
      CDD = CD * 1.E3
      TTLAST = TLAST / 3600.
C     IF (ISTA .LE. 1) WRITE(6,160)
      XKM = X(ISTA) / 1000.
      YKM = Y(ISTA) / 1000.
C     WRITE(6,150) TTLAST, RLAT, RLON, Z, TA, TW, WS, WD, CDD,
C    1 XKM, YKM
      ATAU(ISTA,1) = CD * RHOAW * U(1) * WS
      ATAU(ISTA,2) = CD * RHOAW * U(2) * WS
   30 READ (LUN,130) TLAST, RLAT, RLON, Z, TA, TW, WS, WD
      Z = Z / 3.281
      TA = 5./9.*(TA - 32.)
      TW = 5./9.*(TW - 32.)
      WS = WS / 3.281
      TLAST = TLAST * 3600.
      IF (T .LT. TLAST .AND. ISTA .EQ. 0) GO TO 120
      ISTA = ISTA + 1
C
C  IF FIRST TIME THROUGH, FIND ATAU
C
      IF (ISTA .EQ. 1) GO TO 20
C
C  CHECK IF LAST RECORD AT TIME TLAST HAS BEEN READ
C
      IF (TLAST .EQ. TNEW) GO TO 20
C
C  NOW FIND THE BEST-FIT LINEAR SURFACE
C
      SX = 0.
      SY = 0.
      SXY = 0.
      SX2 = 0.
      SY2 = 0.
      SXTAU1 = 0.
      SYTAU1 = 0.
      SATAU1 = 0.
      SXTAU2 = 0.
```

```
      SYTAU2 = 0.
      SATAU2 = 0.
      N = ISTA - 1
      AN = FLOAT(N)
      XEQ = .TRUE.
      YEQ = .TRUE.
      DO 40 IN = 1, N
        SX = SX + X(IN)
        SY = SY + Y(IN)
        SXY = SXY + X(IN) * Y(IN)
        SX2 = SX2 + X(IN) ** 2
        SY2 = SY2 + Y(IN) ** 2
        SXTAU1 = SXTAU1 + X(IN) * ATAU(IN,1)
        SYTAU1 = SYTAU1 + Y(IN) * ATAU(IN,1)
        SXTAU2 = SXTAU2 + X(IN) * ATAU(IN,2)
        SYTAU2 = SYTAU2 + Y(IN) * ATAU(IN,2)
        SATAU1 = SATAU1 + ATAU(IN,1)
        SATAU2 = SATAU2 + ATAU(IN,2)
        IF (X(IN) .NE. X(1)) XEQ = .FALSE.
        IF (Y(IN) .NE. Y(1)) YEQ = .FALSE.
   40 CONTINUE
C
C   CALCULATE COEFFICIENTS ANEW, BNEW, CNEW, WHERE
C                             TAU = ANEW * X + BNEW * Y + CNEW
C                             FOR EACH COMPONENT.
C
      ANEW1 = 0.
      ANEW2 = 0.
      BNEW1 = 0.
      BNEW2 = 0.
      CNEW1 = SATAU1 / AN
      CNEW2 = SATAU2 / AN
C
C   CHECK FOR ONE DATA POINT ONLY
C
      IF (XEQ .AND. YEQ) GO TO 80
C
C   CHECK IF DATA POINTS ARE CO-LINEAR
C
      IF(N .EQ. 2) GO TO 60
      DO 50 IN = 3, N
        A = SQRT((X(1) - X(IN))**2 + (Y(1) - Y(IN))**2)
        B = SQRT((X(1) - X(IN-1))**2 + (Y(1) - Y(IN-1))**2)
        C = SQRT((X(IN) - X(IN-1))**2 + (Y(IN) - Y(IN-1))**2)
        S = (A + B + C) / 2.
        D = SQRT(S * (S - A) * (S - B) * (S - C)) / AMAX1(A, B, C, 1.)
C
C   IF NOT CO-LINEAR, GO TO 75
C
        IF (D .GT. 100.) GO TO 75
   50 CONTINUE
C
C   CALCULATE COEFFICIENTS FOR CO-LINEAR DATA POINTS
C
C  60 WRITE(6,175)
   60 CONTINUE
      DO 70 IN=2,N
        IF (X(IN) .EQ. X(1) .AND. Y(IN) .EQ. Y(1)) GO TO 70
        JN=IN
   70 CONTINUE
      ALPHA = ATAN2(Y(JN) - Y(1), X(JN) - X(1))
```

```
      CN = COS(ALPHA)
      SN = SIN(ALPHA)
      SS = CN * SX + SN * SY
      SS2 = CN * CN * SX2 + 2. * CN * SN * SXY + SN * SN * SY2
      SSTAU1 = CN * SXTAU1 + SN * SYTAU1
      SSTAU2 = CN * SXTAU2 + SN * SYTAU2
      D = AN * SS2 - SS * SS
      ANEW1 = CN * (AN * SSTAU1 - SS * SATAU1) / D
      ANEW2 = CN * (AN * SSTAU2 - SS * SATAU2) / D
      BNEW1 = SN * (AN * SSTAU1 - SS * SATAU1) / D
      BNEW2 = SN * (AN * SSTAU2 - SS * SATAU2) / D
      CNEW1 = (SS2 * SATAU1 - SSTAU1 * SS) / D
      CNEW2 = (SS2 * SATAU2 - SSTAU2 * SS) / D
      GO TO 80
C
C  CALCULATE COEFFICIENTS FOR BEST-FIT LINEAR SURFACE
C
   75 D = DET(SX2, SXY, SX, SXY, SY2, SY, SX, SY, AN)
      ANEW1 = DET(SXTAU1, SYTAU1, SATAU1, SXY, SY2, SY, SX, SY, AN) / D
      ANEW2 = DET(SXTAU2, SYTAU2, SATAU2, SXY, SY2, SY, SX, SY, AN) / D
      BNEW1 = DET(SX2, SXY, SX, SXTAU1, SYTAU1, SATAU1, SX, SY, AN) / D
      BNEW2 = DET(SX2, SXY, SX, SXTAU2, SYTAU2, SATAU2, SX, SY, AN) / D
      CNEW1 = DET(SX2, SXY, SX, SXY, SY2, SY, SXTAU1, SYTAU1, SATAU1)/D
      CNEW2 = DET(SX2, SXY, SX, SXY, SY2, SY, SXTAU2, SYTAU2, SATAU2)/D
   80 CONTINUE
      ISTA = 1
      IF (T .GT. TNEW) GO TO 10
C     WRITE(6,170)
C
C  DETERMINE PROPER LOCATION XS,YS AND CALCULATE STRESS
C
   90 IF (K .EQ. 2) GO TO 100
      XS = I * RPARM(3)
      YS = (FLOAT(J) - .5) * RPARM(3)
      TAUNEW = ANEW1 * XS + BNEW1 * YS + CNEW1
      TAUOLD = AOLD1 * XS + BOLD1 * YS + COLD1
      GO TO 110
  100 XS = (FLOAT(I) - .5) * RPARM(3)
      YS = J * RPARM(3)
      TAUNEW = ANEW2 * XS + BNEW2 * YS + CNEW2
      TAUOLD = AOLD2 * XS + BOLD2 * YS + COLD2
C
C  INTERPOLATE IN TIME
C
  110 TAU = (TAUNEW - TAUOLD) / (TNEW - TOLD) * (T - TOLD) + TAUOLD
      GO TO 180
  120 WRITE(6,140) TLAST, T
      STOP
  130 FORMAT (8G10.4)
  140 FORMAT (' TIME OF FIRST METEOROLOGICAL DATA', G10.4,
     1        ' SECONDS IS', ' GREATER THAN T = ', G10.4,
     2        ' - PROGRAM TERMINATED')
  150 FORMAT (' ', F5.1, ' * ', F10.7, ' * ', F10.7, ' * ', F4.1, ' * ',
     1        F5.2, ' * ', F5.2, ' * ', F6.2, ' * ', F4.0, ' * ', F5.2,
     2        ' * ',F5.0,' * ',F5.0)
  160 FORMAT (' ', 95('*')/
     1        ' TIME  *  LATITUDE  *  LONGITUDE *  Z   * T-AIR * ',
     2        'T-H2O * W-SPD  * W-DIR * CDE3 *    X    *    Y'/' ', 95('*'))
  170 FORMAT (' ', 95('*'))
  175 FORMAT(' THESE DATA POINTS ARE CO-LINEAR OR NEARLY CO-LINEAR')
  180 RETURN
```

```
      END
C
      FUNCTION XDIST(RLAT, RLON)
      COMMON /GPARM/ RPARM(23), IPARM(54), ZPARM(2)
      PI = ATAN2(0.,-1.)
      ALPHA = RPARM(7) * PI / 180.
      DLAT = RLAT - RPARM(1)
      DLON = RPARM(2) - RLON
C
C  FIND XPRIME, YPRIME - DISTANCES FROM THE ORIGIN OF THE STANDARD
C  BATHYMETRIC GRID
C
      XP = RPARM(8)  * DLON + RPARM(9)  * DLAT + RPARM(10)  * DLON * DLAT +
     1     RPARM(11)  * DLON ** 2
      YP = RPARM(12)  * DLON + RPARM(13)  * DLAT + RPARM(14)  * DLAT *
     1     DLON + RPARM(15)  * DLON ** 2
C
C  TRANSFORM TO 'UNPRIMED' SYSTEM
C
C  FIRST ROTATE
C
      XDIST = (XP*COS(ALPHA) + YP*SIN(ALPHA)) * 1000.
C
C  NOW TRANSLATE
C
      XDIST = XDIST + ZPARM(1) * RPARM(3)
      RETURN
      END
C
      FUNCTION YDIST(RLAT, RLON)
      COMMON /GPARM/ RPARM(23), IPARM(54), ZPARM(2)
      PI = ATAN2(0.,-1.)
      ALPHA = RPARM(7) * PI / 180.
      DLAT = RLAT - RPARM(1)
      DLON = RPARM(2) - RLON
C
C  FIND XPRIME, YPRIME - DISTANCES FROM THE ORIGIN OF THE STANDARD
C  BATHYMETRIC GRID
C
      XP = RPARM(8)  * DLON + RPARM(9)  * DLAT + RPARM(10)  * DLON * DLAT +
     1     RPARM(11)  * DLON ** 2
      YP = RPARM(12)  * DLON + RPARM(13)  * DLAT + RPARM(14)  * DLAT *
     1     DLON + RPARM(15)  * DLON ** 2
C
C  TRANSFORM TO 'UNPRIMED' SYSTEM
C
C  FIRST ROTATE
C
      YDIST = (YP*COS(ALPHA) - XP*SIN(ALPHA)) * 1000.
C
C  NOW TRANSLATE
C
      YDIST = YDIST + ZPARM(2)  * RPARM(3)
      RETURN
      END
C
      FUNCTION RLAT(X,Y)
      COMMON /GPARM/ RPARM(23), IPARM(54), ZPARM(2)
      PI = ATAN2(0.,-1.)
      ALPHA = RPARM(7) * PI / 180.
C
```

```
C  TRANSFORM THE POINTS TO THE 'PRIMED' COORDINATE SYSTEM,
C  IE., THAT OF THE STANDARD BATHYMETRIC GRID
C
C  FIRST TRANSLATE
C
      XX = X - ZPARM(1)  * RPARM(3)
      YY = Y - ZPARM(2)  * RPARM(3)
C
C  NOW ROTATE
C
      XP=(XX*COS(ALPHA)-YY*SIN(ALPHA))/(1000.)
      YP=(YY*COS(ALPHA)+XX*SIN(ALPHA))/(1000.)
      DLAT = RPARM(20) * XP + RPARM(21) * YP + RPARM(22) * XP * YP +
     1        RPARM(23) * XP ** 2
      RLAT = RPARM(1) + DLAT
      RETURN
      END
C
      SUBROUTINE INIT(D, S, IDIM, TLKQ, RLKQ)
      DIMENSION S(IDIM,IDIM), D(IDIM,IDIM)
      COMMON /GPARM/ RPARM(23), IPARM(54), ZPARM(2)
      DATA SPVAL /1.0E20/
      OPEN(16, FILE='lakeinit.psi', STATUS='OLD')
      REWIND 16
      IM = IPARM(1)
      JM = IPARM(2)
      READ(16,30,END=10) ((S(I,J),I=1,IM),J=1,JM)
C  CONVERT CFS TO CMS
      DO 50 I=1,IM
  DO 50 J=1,JM
     IF(S(I,J) .EQ. SPVAL) GO TO 50
     S(I,J) = S(I,J)/3.53198E+01
   50 CONTINUE
      CALL PRNT(6, S, IDIM, IM, JM, SPVAL)
C  UPDATE STREAM FUNCTION TO CURRENT VALUES
      IF (TLKQ .EQ. RLKQ) RETURN
      ADDQ = (TLKQ-RLKQ)/(2.0D0*35.3198)
      DO 40 I=1,IM
  DO 40 J=1,JM
     IF (S(I,J) .EQ. SPVAL) GO TO 40
     IF (S(I,J) .GT. 0.0D0) S(I,J) = S(I,J)+ADDQ
     IF (S(I,J) .LT. 0.0D0) S(I,J) = S(I,J)-ADDQ
   40 CONTINUE
      CALL PRNT(6, S, IDIM, IM, JM, SPVAL)
      RETURN
C
C  NO INITIAL CONDITION FILE, SET STREAMFUNCTION TO ZERO
C
   10 CONTINUE
      DO 20 I = 1, IM
         DO 20 J = 1, JM
   20       S(I,J) = 0.
   30 FORMAT(6E12.5)
      RETURN
      END
C
      SUBROUTINE OUTP(TIME, D. S, SPD, IDIM)
      DIMENSION D(IDIM,IDIM),S(IDIM,IDIM),SPD(IDIM,IDIM)
      COMMON /GPARM/ RPARM(23), IPARM(54), ZPARM(2)
      DATA IREC /0/
      IF (TIME .LT. 0.) RETURN
```

```
      IM=IPARM(1)
      JM=IPARM(2)
C
C  FIRST TIME THROUGH, OPEN OUTPUT FILE
C
      IF(IREC .NE. 0) GO TO 30
      OPEN(11, FILE='stream.dat', STATUS='NEW')
C
C  WRITE STREAM FUNCTION FIELD
C
   30 IREC=IREC+1
      DO 70 I=1,IM
  DO 70 J=1,JM
  IF (S(I,J) .EQ. 1.0E+20) GO TO 70
  S(I,J) = S(I,J) * 35.3198
   70 CONTINUE
      REWIND 11
      WRITE(11,40) ((S(I,J),I=1,IM),J=1,JM)
      DO 80 I=1,IM
  DO 80 J=1,JM
  IF (S(I,J) .EQ. 1.0E+20) GO TO 80
  S(I,J) = S(I,J) / 35.3198
   80 CONTINUE
   40 FORMAT(6E12.5)
      RETURN
      END
C
      SUBROUTINE COMPARE(T, D, S, SPD, IDIM, U, V)
C
C PURPOSE:
C
C           CHECKS FOR THE FINAL STEADY STATE RESPONSE
C           FOR THE GIVEN INPUT TO RLID
C
C ALGORITHIM:
C
C           UP TO 10 GRID BOXES ARE SELECTED AT RANDOM FROM
C           WITHIN THE LAKE GRID SYSTEM. THE VALUES FOR S, U, V AND/OR
C           SPD ARE COMPARED BETWEEN THE CURRENT TIME STEP AND THE
C           TIME STEP IMMEDIATELY PRECEDING. THE DIFFERENCE BETWEEN
C           VALUES AT THE CONSECUTIVE TIME STEPS ARE COMPARED TO A
C           PREDETERMINED TOLERANCE. IF ALL DIFFERENCES ARE LESS THAN
C           THE TOLERANCE, THE FINAL STEADY STATE TRANSPORTS ARE PRINTED
C           OUT ALONG WITH THE COMPUTED (PRECEDING AND CURRENT) S, SPD,
C           U, AND/OR V DIFFERENCES FOR THE SELECTED POINTS.
C
      DIMENSION D(IDIM,1), S(IDIM,1), SPD(IDIM,1), CDIF(20), RMS(2)
     &    ,PS(10), PSPD(10), CS(10), CSPD(10), RDIF(20)
C    &            ,U(IDIM,1), V(IDIM,1)
C    &            ,CU(10), CV(10), PU(10), PV(10)
      COMMON /GPARM/ RPARM(23), IPARM(54), ZPARM(2)
      DATA NIDIF, TOL, NTSTEPS /10, .01, 1/
C
C FILE TO PLOT RELATIVE DIFFERENCES OPENED IN RLID.F
C
C CHECK IF FIRST TIME STEP
C
      WRITE(6,*) T,DT
C     if(t .lt. 1710000.) go to 99
      IF(T .EQ. DT) GO TO 99
      IM = IPARM(1)
```

```
      JM = IPARM(2)
C
C SET CURRENT S & SPD T,ARRAY CS & CSPD
C OR CHANGE TO U & V WITH CU &CV.
C
      CS(1) = S(4,7)
      CS(2) = S(10,6)
      CS(3) = S(15,12)
      CS(4) = S(16,22)
      CS(5) = S(12,28)
      CS(6) = S(22,10)
      CS(7) = S(33,13)
      CS(8) = S(26,26)
      CS(9) = S(22,33)
      CS(10) = S(20,19)
      CSPD(1) = SPD(4,7)
      CSPD(2) = SPD(10,6)
      CSPD(3) = SPD(15,12)
      CSPD(4) = SPD(16,22)
      CSPD(5) = SPD(12,28)
      CSPD(6) = SPD(22,10)
      CSPD(7) = SPD(33,13)
      CSPD(8) = SPD(26,26)
      CSPD(9) = SPD(22,33)
      CSPD(10) = SPD(20,19)
C
C CALCULATE DIFFERENCES
C
      DO 10 I=1,NIDIF
          CDIF(I) = ABS(PS(I)-CS(I))
          CDIF(I+NIDIF) = ABS(PSPD(I)-CSPD(I))
   10 CONTINUE
C
C CALCULATE RELATIVE DIFFERENCES
C
      DO 15 I=1,NIDIF
   RDIF(I) = ABS(PS(I)-CS(I))/ABS(CS(I))
   RDIF(I+NIDIF) = ABS(PSPD(I)-CSPD(I))/ABS(CSPD(I))
   15 CONTINUE
C
C CALCULATE ROOT MEAN SQUARE
C
      RMS(1) = 0.0D0
      RMS(2) = 0.0D0
      DO 20 I=1,NIDIF
           RMS(1) = RMS(1)+(CDIF(I)*CDIF(I))
           RMS(2) = RMS(2)+(CDIF(I+NIDIF)*CDIF(I+NIDIF))
   20 CONTINUE
      RMS(1) = SQRT(RMS(1)/FLOAT(NIDIF))
      RMS(2) = SQRT(RMS(2)/FLOAT(NIDIF))
C
C COUNT THIS TIME STEP
C
      NTSTEPS = NTSTEPS+1
C
C WRITE SELECTED POINT VALUES AND DIFFERENCES
C
      WRITE(6,100)
      WRITE(6,105)
      DO 30 I=1,NIDIF
   II = I+NIDIF
```

```
         WRITE(6,110) I, PS(I), CS(I), CDIF(I),
     &                II, PSPD(I), CSPD(I), CDIF(II)
         WRITE(3,115) NTSTEPS, RDIF(I)
      WRITE(3,115) NTSTEPS, RDIF(II)
   30 CONTINUE
      WRITE(6,120) RMS(1), RMS(2)
C
C CHECK TOLERANCE
C
      NDIF = 2*NIDIF
      J = 0
      DO 50 I=1,NDIF
         IF(CDIF(I) .GT. TOL) GO TO 55
         GO TO 50
   55    J = J+1
   50 CONTINUE
C
C HAS STEADY STATE BEEN REACHED? IF YES, PRINT
C STEADY STATE VALUES AND STOP.
C
      IF(J .GT. 0) GO TO 60
      WRITE(6,150) T
      WRITE(6,160)
      CALL PRNT(6, S, IDIM, IM, JM, 0.)
      WRITE(6,170)
      CALL PRNT(6, SPD, IDIM, IM, JM, 0.)
      STOP
   60 CONTINUE
      WRITE(6,180)
C
C IF NO, RESET PU & PV AND PROCEED TO NEXT TIME STEP.
C
   99 CONTINUE
      PS(1) = S(4,7)
      PS(2) = S(10,6)
      PS(3) = S(15,12)
      PS(4) = S(16,22)
      PS(5) = S(12,28)
      PS(6) = S(22,10)
      PS(7) = S(33,13)
      PS(8) = S(26,26)
      PS(9) = S(22,33)
      PS(10) = S(20,19)
      PSPD(1) = SPD(4,7)
      PSPD(2) = SPD(10,6)
      PSPD(3) = SPD(15,12)
      PSPD(4) = SPD(16,22)
      PSPD(5) = SPD(12,28)
      PSPD(6) = SPD(22,10)
      PSPD(7) = SPD(33,13)
      PSPD(8) = SPD(26,26)
      PSPD(9) = SPD(22,33)
      PSPD(10) = SPD(20,19)
C
  100 FORMAT(1X,/75(1H*)/14X,'COMPARISON OF COMPUTED ',
     &'TRANSPORTS'/20X,20HTO FIND STEADY STATE/)
  105 FORMAT(1X,2(5X,30HI   PREVIOUS     CURRENT   DIF)/12X,
     &15HS              S,20X,17HSPD             SPD)
  110 FORMAT(3X,2X,I2,F10.3,2X,F10.3,1X,F8.4,
     &3X,I2,2X,F8.4,4X,F8.4,3X,F7.4)
  115 FORMAT(1X,I3,E12.5)
```

```
120 FORMAT(75(1H*)/2X,2(20X,'RMS = ',F8.4)/75(1H*)/)
130 FORMAT(1X,75(1H*))
140 FORMAT(1X,32HTOLERANCE NOT SATISFIED AT DIF = ,I2)
150 FORMAT(1X,22HSTEADY STATE SATISFIED,' TIME = ',F13.4)
160 FORMAT(1X,'PLOT OF S')
170 FORMAT(1X,'PLOT OF SPD')
180 FORMAT(1X,'STEADY STATE HAS NOT YET BEEN REACHED')
190 FORMAT(1X,'J = ',I2)
    RETURN
    END
```