R 82-45

MIT

USER'S MANUAL FOR THE MIT LAKE CIRCULATION MODELS (MITLAKE)

by
PETER SHANAHAN
and
DONALD R.F. HARLEMAN

Ralph M. Parsons Laboratory
Aquatic Science and Environmental Engineering

Report No. 285

Prepared under the support of the
National Science Foundation
Water Resources and Environmental Engineering Program
Division of Civil and Environmental Engineering
and
Eastern European Program
Division of International Programs

JUNE 1982

DEPARTMENT
OF
CIVIL
ENGINEERING

SCHOOL OF ENGINEERING
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Cambridge, Massachusetts 02139

R 82-45

USER'S MANUAL FOR THE
MIT LAKE CIRCULATION MODELS
(MITLAKE)

by

Peter Shanahan

and

Donald R.F. Harleman

Ralph M. Parsons Laboratory

Aquatic Science and Environmental Engineering

Report No. 285

Prepared under the support of the

National Science Foundation

Water Resources and Environmental Engineering Program

Division of Civil and Environmental Engineering

and

Eastern European Program

Division of International Programs

JUNE 1982

INTRODUCTION

During the course of research into water quality and hydrodynamic interactions in Lake Balaton in Hungary (Shanahan et al., 1981 and Shanahan and Harleman, 1982) two models of wind-driven lake circulation were employed. The models, which were developed from an earlier model by Nelson (1979), are two-dimensional and three-dimensional versions.

The 3-D model employs the Galerkin method to solve for the vertical velocity distribution as a function of time and horizontal space. A linear bottom friction relation is used, thus making the model inappropriate to very shallow lakes.

The 2-D model was developed in response to this limitation. The 2-D model determines the depth integrated velocity as a function of time and horizontal space. A non-linear bottom friction law is employed, broadening the model applicability to include very shallow lakes.

These notes are a brief description of the computer programs developed to perform the 2-D and 3-D model simulations and to plot the model output. The notes are intended as an aid to potential users; however, no effort has been made to be exhaustive in explaining the implementation of the programs.

ACKNOWLEDGMENTS

Support for the model development was provided by the National Science Foundation under Grant No. CEE-7906125, administered by Dr. Arthur A. Ezra, Program Director for Water Resources and Environmental Programs of the Division of Civil Engineering, and by N.S.F. Grant No. INT-8112452, administered by Ms. Deborah L. Wince, Program Officer of the Division of International Programs.

Dr. Keith D. Stolzenbach and Dr. Ole S. Madsen of M.I.T., and Dr. Laszlo Somlyody of IIASA, supplied assistance and advice during the course of model development and testing. The text of this note was expertly prepared by Ms. Carole J. Solomon, Ms. Laureen Luszcz, and Ms. Jacqueline A. Winton.

TABLE OF CONTENTS

1. 3-D CIRCULATION MODEL

1.1 Introduction

The computer program to simulate 3-D currents in lakes was first developed by Nelson (1979) drawing on earlier work by Cooper and Pearce (1977). The program was substantially modified for application to Lake Balaton in Hungary by Shanahan, Harleman and Somlyódy (1981). Subsequently, errors were found in the program, as reported by Shanahan and Harleman (1982) and have been corrected.

The primary technical reference for the model development is Nelson (1979). Additional technical background is found in Shanahan et al. (1981). Appendix A of this note indicates a number of typographical errors which should be corrected before delving into the detailed mathematical derivations in Nelson (1979) and Shanahan et al. (1981).

As noted in Appendix C of Shanahan and Harleman (1982), the application by Shanahan et al. (1981) contains errors. The calibration reported for the Lake Balaton application is largely fortuitous and the use of this program is not recommended for shallow lakes. The model is, nevertheless, applicable to deep lakes where bottom friction is not a dominant influence.

The model does not explicitly consider the dependence of density on depth as occurs in stratified lakes. This dependence may be captured indirectly through the depth-varying eddy viscosity function, as suggested by Nelson (1979). Nelsons' application to Lake Mendota, a stratified lake, showed fair agreement with field results without any consideration of density effects.

1.2 Major Programs

The 3-D model package includes a simulation program and two programs for the plotting of model predictions:

WC3D - FORTRAN source code program for computation of transient 3D velocity in a lake

PLOT3D - FORTRAN source code program to plot velocity vectors and other output of WC3D on a CRT display terminal employing Tektronix PLOT-10 software

WSPLOT - FORTRAN source code program to plot water surface elevation and mass flux output of WC3D on an off-line plotter using CalComp software

The programs were developed for use on the MIT Information Processing Service IBM 370/168. Limited software conversion may be necessary to transport the the programs to other computer systems.

1.3 Input/Output for WC3D

1.3.1 Input/Output Units

5 - main point unit for NAMELIST program input (input data file)

6 - printed output unit (line printer, terminal or file)

8 - terminal input-output unit

9 - output unit for velocity output for later input to PLOT3D (output data file)

10 - output unit for water surface elevation and flux output for later input to WSPLOT (output data file)

IOWIND - another unit may be defined for wind speed and direction data in the NAMELIST input---see below (input data file)

Characteristics of the I/O units using IBM notation are defined in the following table:

Table 1.1

I/O Unit	RECFM (Record Format)	LRECL (Record Length)	BLKSIZE (Block Size)	Carriage control Character
5	FB	80	80	no
6	FB	133	133	yes
8	FB	132	132	no
9	VS	404	800	no
10	VS	160	800	no
IOWIND	FB	80	800	no

1.3.2 Program Input (Unit 5)

The program utilizes the IBM NAMELIST utility for input. See IBM FORTRAN documentation for information on NAMELIST. The data are given in the following, record by record. In addition, Figure 1.1 is a sample data file.

RECORD 1 - TITLE CARD (fixed format)

80 Character Run Title

RECORD 2 - NAMELIST &MISC

JP - The number of cosine functions to be included in velocity trial function series expansion

KP - The number of points used to define the vertical eddy viscosity function (equal to the number of linear segments plus one) (≤ 4)

LP - The number of grid squares in the x-direction (for closed boundaries - shorelines - a zero depth grid must be included to define the boundary)

MP - The number of grid squares in the y-direction (including zero depth grids for boundaries)

CPHI - The latitude of the site in degrees from the equator

NS - The number of trapezoids to be used in numerical integrations - usually 20

GANGLE - The angle between the grid positive y axis and true north, measured in degrees clockwise from the y axis. Default value is 0.0

NOTE:
$JP*LP*MP \leq 7500$
$LP*MP \leq 1500$
$KP*NP \leq 150$
$JP*NP \leq 180$
$JP*JP*NP \leq 1080$

RECORD 3 - NAMELIST &TIME

TLAST - Simulation end time [seconds]

DTT - Simulation time step, Δt [seconds]

DL - Grid size, $\Delta x = \Delta y$ [meters]

RECORD 4 - NAMELIST &EDYVIS

CK - Non-dimensional depth values, z/h, defining piecewise-linear vertical eddy viscosity function in order from surface to bottom - KP values

EV - Eddy viscosity values defining vertical eddy viscosity function [m^2/sec] in order from surface to bottom - KP values

RECORD 5 - NAMELIST &XREF

NP - Number of grid property sets (≤ 30)

NXR - Cross reference array giving grid property set number for each grid square: enter by rows (from west to east in each row) with rows in south to north order - LP*MP values

RECORD 6 - NAMELIST &DEPTH

H - Array of depth values for grid property sets [meters] - NP values

RECORD 7 - NAMELIST &FRICT

CB - Array of bottom friction coefficient values for grid property sets [m/sec] - NP values

RECORD 8 - NAMELIST &ICS

ETA - Array of water surface elevation displacement values [meters] to be used as initial condition in each grid square - LP*MP values.
Default value is 0.0.

RECORD 9 - NAMELIST &WIND

DTWIND - Time increment of wind input data [sec]

NWIND - Number of values in wind data arrays (≤ 100)

IOWIND - I/O unit number for wind input (use IOWIND = 5 if wind data is included with NAMELIST data)

NWFORM - Wind shear formula used:
1 - Wu (1969)
2 - Hick's et. al. (1974) shallow water
Default value is 1.

SMOOTH - If SMOOTH = .FALSE. (the default value), wind speed and direction are determined by linear interpolation between input values. If SMOOTH = .TRUE., continuous parabolic interpolation is used.

OPTIONAL RECORDS - WIND DATA - INCLUDE IF IOWIND = 5

WSPEED - Array of windspeed values [m/sec] - NWIND values in 8F10 fixed format records

DIREC - Array of wind direction values [direction from which wind blows in degrees clockwise from true north - not grid north] - NWIND values in 8F10 fixed format records

RECORD 10 - NAMELIST &FIELD

WNDFLD - Logical variable to indicate if a spatially non-uniform wind field is used. If WNDFLD = .TRUE., modify the wind speed by the WNDFAC value in each grid.

WNDFAC - Array of multiplicative factors to be applied to input wind speed in each grid square - LP*MP values. Factors are applied only if WNDFLD = .TRUE.

WCFAC - Multiplicative factor to be applied to windspeed values (i.e., for all grid sqaures). Default value is 1.0; factor is applied whether WNDFLD equals .TRUE. or .FALSE.

RECORD 11 - NAMELIST &PRINT

ISTPR1 - Starting time for tabular outputs [sec]. Tabular outputs are those specified on NAMELIST'S &PRINT, &PROFIL and &LAYER.

IDTPR1 - Time increment for tabular output [sec].

PRTDAT - Logical variable to request print-out of tables of input data. Default value is .TRUE. The input data tables are printed only once, at the run start.

PRTETA - Logical variables to request print-outs of water surface elevation table.

PRTFLX - Logical variable to request print-outs of the mass flux streamfunction table.

PRTAVG - Logical variable to request print-outs of depth averaged velocity table.

RECORD 12 - NAMELIST &PROFIL

PRTPRO - Logical variable to request print-out of vertical velocity profiles

NPRO - Number of grids for which velocity profiles are to be printed ($\leq$ 10)

LOCATE - Array of grid coordinates (in the form (L,M) where L is the column number (x coordinate) and M is the row number (y coordinate)) - NPRO coordinate pairs.

OPTIONAL RECORDS - PROFILE DEPTH DATA - INCLUDE IF NPRO > 0

ZPRO - Arrays of depths at which velocities are to be printed. NPRO records - one record per profile. Fixed-format records - 16F5 - no more than 16 values per profile.

RECORD 13 - NAMELIST &LAYER

PRTLAY - Logical variable to request tables of horizontal velocities.

PLTLAY - Logical variable to request storage of horizontal velocities in output file for later plotting.

NLAY - Number of layers for printing and plotting ($\leq$ 16)

OPTIONAL RECORD - LAYER DEPTH DATA - INCLUDE IF NLAY > 0

ZLAY - Array of layer depths [m] for printing and plotting - NLAY values in 16F5 fixed format record.

RECORD 14 - NAMELIST &LIST

ISTPR2 - Starting time for list outputs [sec]. List outputs are those specified on NAMELIST'S &LIST and &FLUXES.

IDTPR2 - Time increment for list output [sec].

LSTETA - Logical variable to request water surface elevation lists.

PLTETA - Logical variable to request storage of water surface elevation in output file for later plotting.

NETA - Number of locations at which water surface elevations are to be listed. ($\leq$ 10)

LOCATE - Array of grid coordinates (in the form (L,M) where L is the column number (x coordinate) and M is the row number (y coordinate)) - NETA values.

RECORD 15 - NAMELIST & FLUXES

LSTFLX - Logical variable to request laterally averaged longitudinal flux lists.

PLTFLX - Logical variable to request storage of fluxes in output file for later plotting.

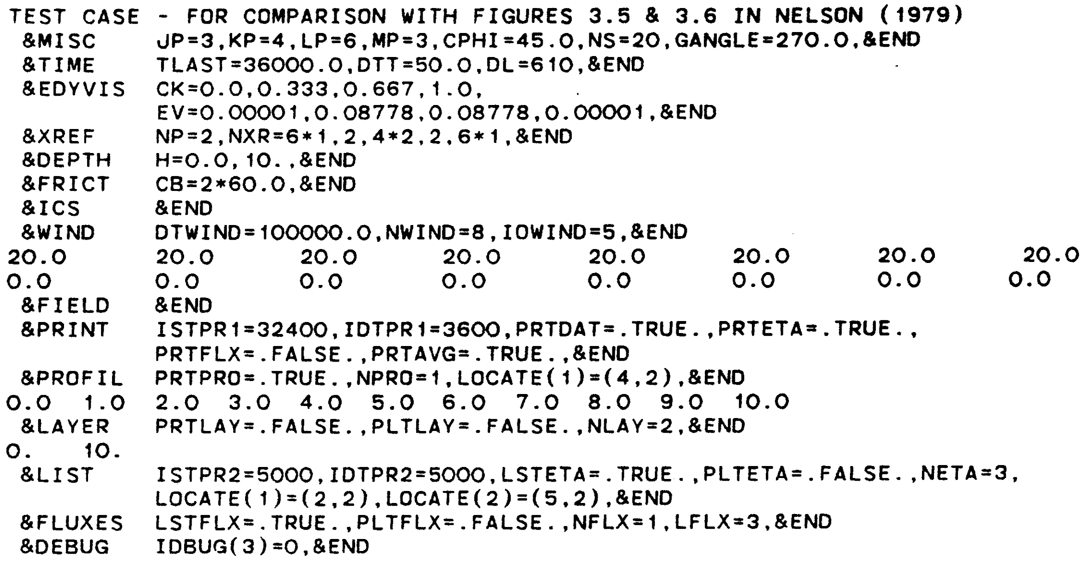
NFLX - Number of lateral sections at which fluxes are to be listed ($\leq$ 16).

LFLX - Array of section coordinates (as L, the grid column number (x coordinate)).

RECORD 16 - NAMELIST & DEBUG

IDBUG - Array of flags to request additional output from various program subroutines. These outputs show the internal program computations and are generally useful for program debugging only. There are 16 variables in the IDBUG array, each assigned to a different subroutine. IDBUG = 0, the default value, specifies no output. Increasing values of IDBUG produces greater quantities of output. This option should be used with care and on short simulations only due to the large quantity of printed output produced. The subroutine correspondence is:

IDBUG (1)	unassigned
(2)	BNDRY
(3)	CONST
(4)	HEADER
(5)	BSHEAR
(6)	OUTDAT
(7)	OUTPUT
(8)	PRINTS
(9)	SETETA
(10)	SETVEL
(11)	VELOCY
(12)	TRANS
(13)	NEWPRM
(14)	CDSOLV
(15)	SHEAR
(16)	unassigned



```
TEST CASE - FOR COMPARISON WITH FIGURES 3.5 & 3.6 IN NELSON (1979)
 &MISC     JP=3,KP=4,LP=6,MP=3,CPHI=45.0,NS=20,GANGLE=270.0,&END
 &TIME     TLAST=36000.0,DTT=50.0,DL=610,&END
 &EDYVIS   CK=0.0,0.333,0.667,1.0,
           EV=0.00001,0.08778,0.08778,0.00001,&END
 &XREF     NP=2,NXR=6*1,2,4*2,2,6*1,&END
 &DEPTH    H=0.0,10.,&END
 &FRICT    CB=2*60.0,&END
 &ICS      &END
 &WIND     DTWIND=100000.0,NWIND=8,IOWIND=5,&END
20.0       20.0      20.0      20.0      20.0      20.0      20.0      20.0
0.0        0.0       0.0       0.0       0.0       0.0       0.0       0.0
 &FIELD    &END
 &PRINT    ISTPR1=32400,IDTPR1=3600,PRTDAT=.TRUE.,PRTETA=.TRUE.,
           PRTFLX=.FALSE.,PRTAVG=.TRUE.,&END
 &PROFIL   PRTPRO=.TRUE.,NPRO=1,LOCATE(1)=(4,2),&END
0.0  1.0   2.0  3.0  4.0  5.0  6.0  7.0  8.0  9.0  10.0
 &LAYER    PRTLAY=.FALSE.,PLTLAY=.FALSE.,NLAY=2,&END
0.   10.
 &LIST     ISTPR2=5000,IDTPR2=5000,LSTETA=.TRUE.,PLTETA=.FALSE.,NETA=3,
           LOCATE(1)=(2,2),LOCATE(2)=(5,2),&END
 &FLUXES   LSTFLX=.TRUE.,PLTFLX=.FALSE.,NFLX=1,LFLX=3,&END
 &DEBUG    IDBUG(3)=0,&END
```

Figure 1.1

Sample Data File, I/O Unit 5
for Program WC3D

1.3.3 Program Input (other units)

Unit 8 - various free fromat input data are entered at the terminal in response to program prompts

Unit IOWIND - if IOWIND $\neq$ 5, wind input data must be available on the units specified as IOWIND. These are fixed-format records of length 80 defined as:

Wind speed data (8F10.0)

WSPEED(I),I=1,NWIND	wind speed [m/s]

Wind direction data (8F10.0)

WDIREC(I),I=1,NWIND	direction from which wind blows [degrees measured counterclock-wise from grid x-axis]

1.4 Programming Notes

1.4.1 Subroutine Structure

The program functions are separated into modules handled by individual subroutines. The various routines and their functions are listed below. The calling sequence of the routines is shown in Figure 1.2.

MAIN - Main program. Calls input and computation routines.

AVGTAB - Prints depth averaged velocity output tables.

BLOCK DATA - Variable initialization and default value setting.

BNDRY - Sets water height at open ocean (head) boundaries. This routine is not used for lake simulations.

CDSOLV - Solves for c_j and d_j in each grid square

CONST - Calculates model parameters and constants for each grid property set.

DATIME - Determines run date and time from computer system.

DATOUT - Fills an output table for a two-dimensional array.

DATO2 - Fills an output table for a three-dimensional array.

FLXTAB - Prints mass flux stream function output tables.

HEADER - Prints output header page and input data tables.

INDAT - Array input routine for unformatted records.

INDATI - Entry point to INDAT. Array input routine for formatted integer records - 16I5 format.

INDATR - Entry point to INDAT. Array input routine for formatted real records - 8F10 format.

MINV - Matrix inversion routine from the IBM SSP library.

MPRINT - Prints a square matrix.

MVPRT - Entry point to MPRINT. Prints a square matrix and vector of same dimension.

NEWPRM - Calculates calculation parameters for each grid property set.

OOPTS - Reads output options input data.

OUTDAT - Array output routine for unformatted records.

OUTDTI - Entry point to OUTDAT. Array output routine for formatted integer records - 16I5 format.

OUTDTR - Entry point to OUTDAT. Array output routine for formatted real records - 8F10 format.

OUTPUT - Prints output table for two-dimensional arrays.

OUTP2 - Prints output table for three-dimensional arrays.

PRINTS - Prints transient computation results.

READIN - Reads program input.

SETVEL - Sets velocities on open ocean (head) boundaries. This routine is not used for lake simulations.

SHEAR - Calculates wind shear stress.

SWWS - Calculates wind shear stress for shallow water using Hicks et al. (1974) formula.

TRANS - Transient simulation monitoring routine.

VELOCY - Computes current velocity at a specified location.

VELPRO - Prints velocity profile output table.

VELTAB - Prints horizontal velocity output tables and storage file.

WSELEV - Solves continuity equation for water surface displacement.

WSTAB - Prints water surface elevation displacement output table.

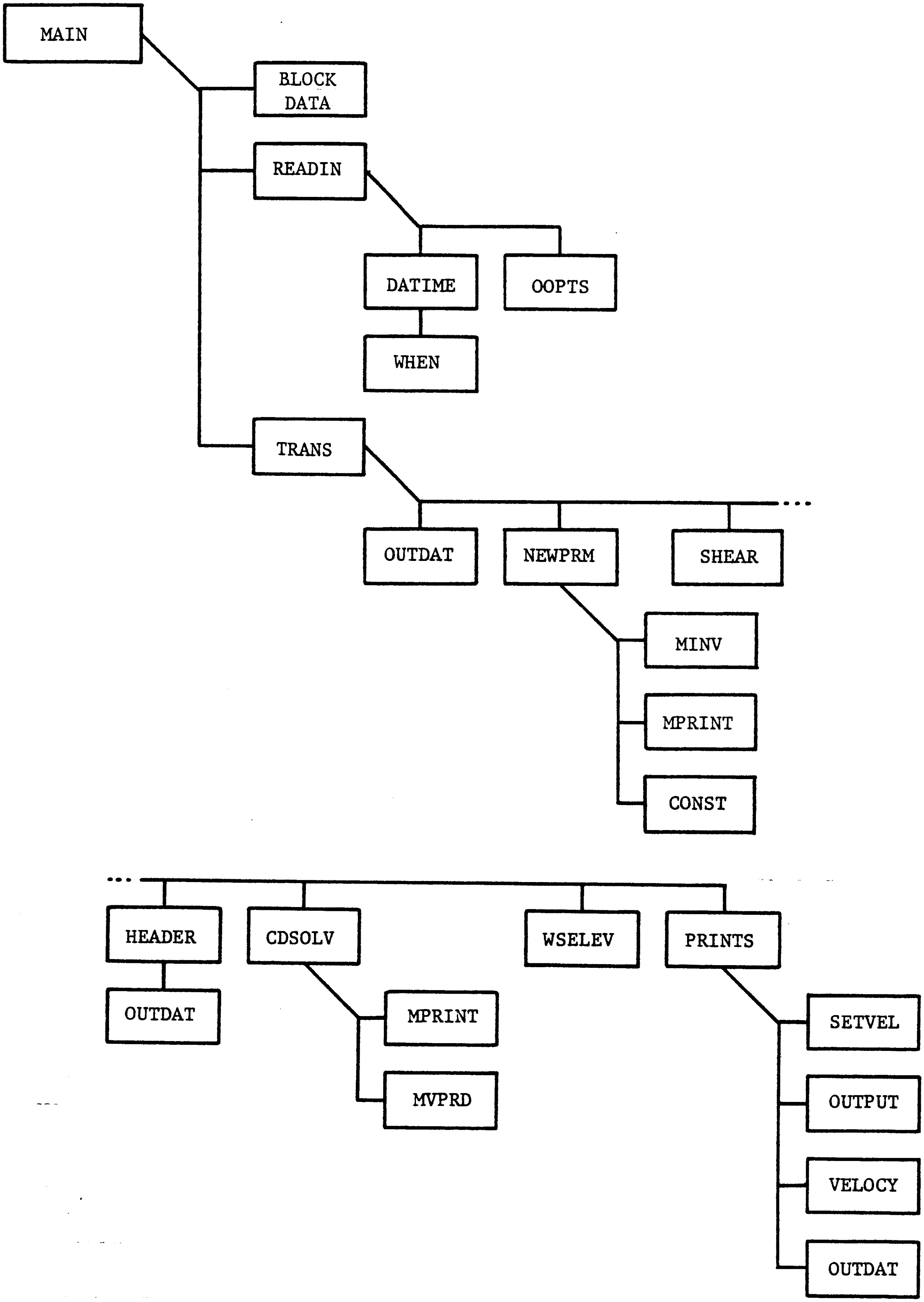


Figure 1.2
WC3D Subroutine Calling Sequence

1.4.2 Program Variables

The program employs a dual set of hydraulic property arrays as a storage reduction device. The basic set of hydraulic properties (still water depth, bottom friction coefficient, eddy viscosity functions, and derived properties) are defined for a set of "grid property sets" which can number no more than 30. These grid property sets are in turn assigned to the actual 2-D horizontal grid which defines the horizontal geometry of the lake. The 2-D horizontal grid is a square grid LP squares "wide" (in the x-direction) and MP squares "tall" (in the y-direction). The product LP*MP can be no more than 1500.

The cross reference array, XREF, is used to assign the grid property sets to the actual 2-D grid. Each of the LP*MP grid squares must be identified to possess properties defined by one of the grid property sets. Since the number of grid property sets is much less than the possible number of grid squares, there is an implicit assumption that many of the grid squares will have identical depths, bottom friction coefficients and other properties. This has not proved a restrictive condition and it may be eased by changing program dimensions.

Study of the example data set in Figure 1.1 should make the dual array scheme clear. In the example, two grid property sets (one with zero depth, the other with 10 m depth) are used to define a 3 by 6 two-dimensional grid. The grid defines an open-ended 1-D channel.

Arrays with multiple subscripts are not employed in the program in order to reduce computation expense. Instead, single subscript arrays (vectors) are employed where two or more subscripts might normally be used. The procedure followed to substitute singly dimensioned arrays for multiply dimensioned arrays is described by Cooper and Pearce (1977).

1.4.3 Other Usage Notes

The 3-D lake circulation program is appropriate for deep lakes where a linear bottom friction law may be used with good approximation. The linear bottom friction leads to a bottom "slip velocity" - a small but finite velocity predicted at the lake bottom. Such a bottom friction condition must necessarily be accompanied by an eddy viscosity function which is constant with depth near the bottom. It would be erroneous to employ an eddy viscosity function which decreases or increases with depth near the bottom in this model.

The stability criterion given by Shanahan et al. (1981) is applicable to this model. The stability limitation due to vertical eddy diffusion which is given by Nelson (1979) may be disregarded since an implicit solution scheme solves for the vertical dimension in this program. See Shanahan et al. (1981), Appendix B for details.

The program calls the IBM system routine WHEN to determine the date and time of each simulation as a unique identifier. The routine calling arguments are defined in Appendix B. Equivalent routines are available on most systems.

1.5 Current Vector Plotting with PLOT3D

At the user's specification, WC3D will create an output file on I/O Unit 9 which contains data for later plotting by PLOT3D. The user executes PLOT3D at an interactive graphics terminal to produce layer by layer vector plots of the 2-D current patterns. Although listings of the same data may be printed when WC3D is executed, the plots are far more useful and easily interpreted. Example plots may be seen in Figures 8.15, 8.16, 8.17 and 8.20 of Shanahan et al. (1981).

The velocity output data file from WC3D is an unformatted (machine code) file which is unintelligible if printed or listed by the user. An unformatted file is much less expensive to write and read by programs than user-readable formatted files, however. Unformatted files also require less storage space.

PLOT3D was programmed for use on a Tektronix 4096 Graphics terminal operating at 1200 band. It employs the Tektronix PLOT-10 library for its basic plotting functions. If the program is to be executed from different hardware, a number of initializing functions performed in subroutine STRPLT should be modifed. If PLOT-10 software is not available, substantial program modifications would be required. In this situation, the similar PLOT2D program should be used as a reference point for reprogramming. PLOT2D uses the more widely available CalComp plotting routines.

Execution of the program is straightforward. The plotting is controlled by the user's response to self-explanatory program prompts. The user must make the PLOT-10 library available for program execution and define the I/O Units given in Table 1.2.

Table 1.2

I/O Unit	RECFM (Record Format)	LRECL (Record Length)	BLKSIZE (Block size)	Definition
5	FB	132	132	Terminal
6	FB	132	132	Terminal
9	VS	404	800	WC3D output file

1.6 Water Surface Elevation and Mass Flux Plots with WSPLOT

1.6.1 Program Operation

In addition to the current vector plot files described in Section 1.5, WC3D can be programmed by the user to create a file of water surface elevation and mass fluxes for later plotting. The water surface elevations are recorded at particular locations (grid squares) specified by the user, while mass flux (discharge) is determined across a specified section of the lake and stored. The information is stored in an output file designated as I/O Unit 10. Printed listings of the same information are also possible. Examples of the plots created by WSPLOT may be seen in Figures 5.10 to 5.12 of Shanahan and Harleman (1982).

The program WSPLOT is executed to create computer plots from the data stored in the files. The program is executed interactively, the user responding to program questions to specify the plots to be made and the plot formats.

WSPLOT may be operated in either of two modes. The program calls standard CalComp plotting routines to construct the plots, and it may be used to create off-line plots on CalComp hardware. It is also programmed to permit interactive plotting on a graphics terminal using the Tektronix CalComp Prescreening software library. In the interactive mode, the proper Tektronix library must be made available for program execution.

WSPLOT accepts program results from up to 10 data files in order to permit comparison plots. It can also accommodate a data file containing field data for comparison of program predictions with field observations. Program operation differs slightly depending upon the number of program output files processed simultaneously. If a single program output data file is used, up to 10 separate variables may be plotted, comparing with field data if desired. (By separate variable we mean the water surface elevation at a given point or mass flux at a given section.) These 10 plots may be combined in one graph each for water surface elevation and mass flux or plotted in individual graphs. When the results of 2 or more simulations are compared by simultaneous processing of multiple output files, only a single variable can be plotted in an execution of WSPLOT. The multiple WC3D output files should be in identical format (the output variables in the same order) for these comparison runs.

1.6.2 Field Data Files

If the user desires to compare field data with the plotted program results, he must prepare a file of field data points for input to WSPLOT. This is a free format file containing sequences of data for each water surface elevation station or mass flux file. The data must be arranged in the order the histories will be plotted: water surface elevation plots first, then all flux plots. The order within each category must further agree with the order the plots are specified in the WSPLOT request sequence.

The free-format files are prepared as shown below. The free-format records require that the data be in the correct order, separated by commas or blanks.

For each data set: (W.S. Elevation Station or Flux Section)

Record 1, Data specifications - ND,ZØ,TØ

ND - number of data points (pairs)

ZØ - zero point of Z values (W.S. elevation or flux)
This is the value of Z in the data file defined to forrespond with Z=0 in the simulation and plot. [cm or m^3/s]

TØ - zero point of T values (time)
This is the time in the data file which corresponds to the start of the simulation (T=0). [hours]

Record Type 2, Time data - (TC(K),K=1,ND)
(may be more than 1 record)

TC(K) - The time corresponding to the K'th observation, ZC(K). [hours]

Record Type 3, Observation data - (ZC(K),K=1,ND)
(may be more than 1 record)

ZC(K) - The water surface elevation [cm] or mass flux [m^3/s] observed at time TC(K).

1.6.3 I/O Units

WSPLOT requires the I/O units shown in Table 1.3

Table 1.3

	RECFM	LRECL	BLKSIZE	
	(Record	(Record	(Block	
I/O Unit	Format)	Length)	Size)	Definition
5	FB	132	132	Terminal
6	FB	132	132	Terminal
12	FB	80	800	Field data file
21	VS	160	800	WC3D output file
22-30	VS	160	800	Additional WC3D output files for multiple plots
For Tektronix Pre-Screening:				
8	FB	132	132	Terminal for Tektronix prescreening messages
9	FB	132	132	Terminal for plot commands
10	VBS	1204	12044	File for replay plot storage
For CalComp Plotting:				
16	VS	504	508	CalComp output device

Note: The specifications for Units 8, 10 and 16 may be system specific. Determine the conventions for CalComp plotting or Tektronix pre-screening on your system before using.

2. 2-D CIRCULATION MODEL

2.1 Introduction

The 2-D circulation model was developed to model the environment of very shallow lakes where bottom friction is an important factor. It was determined that the linear bottom friction law of the 3-D model inadequately represented frictional forces in such lakes. Moreover, in shallow lakes vertical velocity gradients can be reasonably expected to be minor. These and other considerations prompted the development of the 2-D model by Shanahan and Harleman (1982).

The computer program for the 2-D model was developed by modifying the 3-D program described above. The 2-D model thus shares many common features with the 3-D model and we will simplify our explanations in this chapter by referring to the previous discussion of the 3-D program.

2.2 Major Programs

The 2-D model includes a simulation model and two plotting programs in direct correspondence to the 3-D model:

WC2D - FORTRAN source code program for simulation of transient 2-D depth-integrated currents and water surface elevations in a lake

PLOT2D - FORTRAN source code program to plot velocity vectors and other WC2D output on a CalComp or similar plotter

The second plotting program is WSPLOT, which can process the output from either WC3D or WC2D.

2.3 Input/Output for WC2D

2.3.1 Input/Output Units

Units 5, 6, 8, 10 and IOWIND and identically as defined in Section 1.3.1. In addition, three other I/O units are employed:

9 - output unit for mass flux output for later input to PLOT2D (output data file)

15 - output unit for laterally averaged flux for linkage to the Lake Balaton water quality model (output data file)

The characteristics of the I/O Units are tabulated below.

Table 2.1

I/O Unit	RECFM (Record Format)	LRECL (Record Length)	BLKSIZE (Block Size)	Carriage Control Character
5	FB	80	80	no
6	FB	133	133	yes
8	FB	132	132	no
9	VS	404	800	no
10	VS	160	800	no
IOWIND	FB	80	800	no
15	VS	160	800	no

2.3.2 Program Input (Unit 5)

The program input is similar to that described in Section 1.3.2. There are some differences due to the change from a 3-D to 2-D program and some changes in output data. The format of the data is given record-by-record in the following. In addition, an example data file is included as Figure 2.1.

RECORD 1 - TITLE CARD (fixed format)

80 Character Run Title

RECORD 2 - NAMELIST &MISC

LP - The number of grid squares in the x-direction (for closed boundaries - shorelines - a zero depth grid must be included to define the boundary)

MP - The number of grid squares in the y-direction (including zero depth grids for boundaries)

CPHI - The latitude of the site in degrees from the equator

GANGLE - The angle between the grid positive y axis and true north, measured in degrees clockwise from the y axis. Default value is 0.0.

RECORD 3 - NAMELIST &TIME

TLAST - Simulation end time [seconds].

DTT - Simulation time step, Δt [seconds].

DL - Grid size, $\Delta x = \Delta y$ [meters].

RECORD 4 - NAMELIST &REF

NP - Number of grid property sets ($\leq$ 30).

NXR - Cross reference array giving grid property set number for each grid square: enter by rows (from west to east in each row) with rows in south to north order - LP*MP values.

RECORD 5 - NAMELIST &DEPTH

H - Array of depth values for grid property sets [meters] - NP values.

RECORD 6 - NAMELIST &FRICT

CHEZY - Array of Chezy bottom friction coefficient values for grid property sets [$\mathrm{m/sec}^{1/2}$] - NP values.

RECORD 7 - NAMELIST &ICS

ETA - Array of water surface elevation displacement values [meters] to be used as initial condition in each grid square - LP*MP values. Default value is 0.0.

RECORD 8 - NAMELIST &WIND

DTWIND - Time increment of wind input data [sec].

NWIND - Number of values in wind data arrays ($\leq$ 100).

IOWIND - I/O unit number for wind input (use IOWIND = 5 if wind data is included with NAMELIST data).

NWFORM - Wind shear formula used:
1 - Wu (1969)
2 - Hicks et al. (1974) shallow water

3 - Van Dorn (1953)
4 - Fixed wind shear drag coefficient specified as CD in input
Default value is 1.

CD - If NWFORM = 4, CD is read as the value of the wind shear drag coefficient.

OPTIONAL RECORDS - WIND DATA - INCLUDE IF IOWIND = 5

WSPEED - Array of windspeed values [m/sec] - NWIND values in 8F10 fixed format records.

DIREC - Array of wind direction values [direction from which wind blows in degrees clockwise from true north - not grid north] - NWIND values in 8F10 fixed format records.

RECORD 9 - NAMELIST &FIELD

WNDFLD - Logical variable to indicate if a spatially non-uniform wind field is used. If WNDFLD =.TRUE., modify the wind speed by the WNDFAC value in each grid.

WNDFAC - Array of multiplicative factors to be applied to input wind speed in each grid square - LP*MP values. Factors are applied only if WNDFLD = .TRUE.

WCFAC - Multiplicative factor to be applied to windspeed values (i.e. for all grid squares). Default value is 1.0; factor is applied whether WNDFLD equals .TRUE. or .FALSE.

WNDTRN - The default value (WNDTRN = .FALSE.) is recommended. If WNDTRN = .TRUE., a special wind direction transform developed by Somlyódy for Lake Balaton is applied.

RECORD 10 - NAMELIST &PRINT

ISTPR1 - Starting time for tabular outputs [sec]. Tabular outputs are those specified on NAMELIST's &PRINT, &PROFIL and & LAYER.

IDTPR1 - Time increment for tabular output [sec].

PRTDAT - Logical variable to request print-out of tables of input data. Default value is .TRUE. The input data tables are printed only once, at the run start.

PRTETA - Logical variable to request print-outs of water surface elevation table.

PRTFLX - Logical variable to request print-outs of the mass flux streamfunction table

PRTAVG - Logical variable to request print-outs of depth averaged velocity table.

PLTAVG - Logical variable to request storage of depth averaged velocity in output file for later plotting.

RECORD 11 - NAMELIST &LIST

ISTPR2 - Starting time for list outputs [sec]. List outputs are those specified on NAMELIST's &LIST and &FLUXES.

IDTPR2 - Time increment for list output [sec].

LSTETA - Logical variable to request water surface elevation lists.

PLTETA - Logical variable to request storage of water surface elevation in output file for later plotting.

NETA - Number of locations at which water surface elevations are to be listed. ($\leq$ 10)

LOCATE - Array of grid coordinates (in the form (L,M) where L is the column number (x coordinate) and M is the row number (y coordinate)) - NETA values.

RECORD 12 - NAMELIST &FLUXES

LSTFLX - Logical variable to request laterally averaged longitudinal flux lists.

PLTFLX - Logical variable to request storage of fluxes in output file for later plotting.

NFLX - Number of lateral sections at which fluxes are to be listed ($\leq$ 16).

LFLX - Array of section coordinates (as L, the grid column number (x coordinate)).

LSTDIS - Logical variable to request computation and output of longitudinal dispersion due to lateral and vertical velocity non-uniformities.

EDDYD - Horizontal eddy diffusivity for dispersion calculation [cm^2/sec]. Default value is 1.0.

RECORD 13 - NAMELIST &WQ

IDTPR3 - Time increment for printout of flux data to transfer file for Balaton Water Quality Model [sec].

WQFILE - Logical variable to request printout to transfer file for Balaton Water Quality Model.

RECORD 14 - NAMELIST &DEBUG

IDBUG - Array of flags to request additional output from various program subroutines. These outputs show the internal program computations and are generally useful for program debugging only. There are 16 variables in the IDBUG array, each assigned to a different subroutine. IDBUG = 0, the default value, specifies no output. Increasing values of IDBUG produces greater quantities of output. The subroutine correspondence is:

IDBUG	(1)	Unassigned
	(2)	Unassigned
	(3)	Unassigned
	(4)	HEADER
	(5)	BSHEAR
	(6)	OUTDAT
	(7)	OUTPUT
	(8)	PRINTS
	(9)	Unassigned
	(10)	SETVEL
	(11)	Unassigned
	(12)	TRANS (*)
	(13)	Unassigned
	(14)	MSOLV
	(15)	SHEAR
	(16)	DISPC

*A special output option was programmed as a "temporary" feature of subroutine TRANS. If the TRANS debug flag IDBUG(12) = -1, a special plotting output file is written on I/O unit 16 (RECFM = VS, LRECL = 404, BLKSIZE = 800). The file is in the same format as the depth averaged velocity vector file invoked by specifying PLTAVG = .TRUE., however the file contains the net depth averaged velocity computed by continuously summing the velocities in each grid square during the entire run. The resulting net velocities, if computed over a sufficiently long simulation, will reflect the net mass transport, having eliminated most of the oscillatory component of motion in the summation procedure. This output file may be plotted by program PLOT2D as described in Section 2.5 below.

```
LAKE BALATON - 2-D MODEL INPUT DATA
 &MISC     LP=42,MP=9,
           CPHI=46.77,GANGLE=27.5,&END
 &TIME     TLAST=86400,DTT=180,DL=1900.0,&END
 &XREF     NP=17,
           NXR = 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
           1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
           1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
           1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
           1, 1, 1, 9, 9, 9, 9, 9, 9, 7, 5, 4, 1, 1, 1, 11,
           11, 11, 12, 13, 14, 13, 12, 1, 1, 1,
           1, 1, 1, 1, 1, 1, 3, 12, 12, 12, 12, 12, 12, 12, 13, 13,
           13, 13, 13, 13, 14, 14, 13, 14, 14, 13, 11, 11, 12, 11, 13, 13,
           14, 14, 15, 15, 16, 15, 14, 13, 1, 1,
           1, 1, 1, 1, 1, 11, 13, 12, 12, 13, 12, 11, 11, 11, 11, 12,
           12, 12, 12, 13, 13, 13, 12, 12, 12, 12, 10, 1, 1, 11, 13, 14,
           14, 14, 14, 15, 15, 15, 14, 13, 7, 1,
           1, 1, 1, 1, 10, 12, 12, 12, 11, 11, 9, 9, 9, 7, 7, 7,
           7, 7, 9, 11, 12, 11, 10, 9, 9, 9, 9, 1, 1, 11, 12, 13,
           13, 13, 13, 14, 14, 14, 14, 12, 7, 1,
           1, 5, 7, 9, 9, 11, 11, 11, 9, 7, 5, 1, 1, 1, 1, 1,
           1, 1, 1, 1, 9, 5, 1, 1, 1, 1, 1, 1, 1, 9, 9, 9,
           9, 9, 11, 12, 13, 14, 13, 7, 1, 1,
           1, 8, 8, 6, 3, 1, 7, 9, 7, 1, 1, 1, 1, 1, 1, 1,
           1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
           1, 1, 1, 9, 11, 12, 11, 1, 1, 1,
           1, 6, 6, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
           1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
           1, 1, 1, 1, 5, 9, 1, 1, 1, 1,
           1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
           1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
           1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  &END
 &DEPTH    H=0.0,0.5,1.0,1.5,2.0,2.25,2.5,2.75,3.0,3.25,3.5,3.75,
           4.0,4.25,4.5,4.75,8.0,&END
 &FRICT    CHEZY=17*60.0,&END
 &ICS      &END
 &WIND     DTWIND=7200,NWIND=13,IOWIND=11,NWFORM=4,CD=0.0013,&END
 &FIELD    WNDFLD=.FALSE.,WNDTRN=.FALSE.,
           &END
 &PRINT    ISTPR1=720,IDTPR1=86500,PRTDAT=.FALSE.,PRTETA=.FALSE.,
           PRTFLX=.FALSE.,PRTAVG=.FALSE.,PLTAVG=.FALSE.,&END
 &LIST     ISTPR2=720,IDTPR2=720,LSTETA=.TRUE.,PLTETA=.TRUE.,
           NETA=4,LOCATE=(2,8),(22,2),(23,5),(40,6),&END
 &FLUXES   LSTFLX=.TRUE.,PLTFLX=.TRUE.,NFLX=3,LFLX=6,18,29,
           LSTDIS=.TRUE.,&END
 &WQ       &END
 &DEBUG    IDBUG(3)=0,IDBUG(7)=0,&END
```

Figure 2.1 Sample data file, I/O Unit 5 for Program WC2D

2.3.3 Program Inputs (other units)

All other program input data files and termainal inputs are as described in Section 1.3.3.

2.4 Programming Notes

2.4.1 Subroutine Structure

As in the 3-D model, program functions are modularized by using subroutines. Many of the routines in WC2D are identical to those of WC3D; others share the same subroutine name, but differ slightly. Changes in the WC2D subroutine structure from that shown in Section 1.4.1 are indicated below:

MSOLV - Replaces CDSOLV, MPRINT and MVPRD. MSOLV solves for U and V in each grid square

NEWPRM, MINV, MPRINT and CONST are no longer called.

BSHEAR - Called by MSOLV. Calculates bottom shear stress.

VELOCY and VELPRO are no longer called.

2.4.2 Program Variables

The variables employed by WC2D are either defined in Section 2.3.2 or agree with the WC3D variables defined in Section 1.4.2. The introductory remarks of Section 1.4.2 apply to WC2D as well.

2.4.3 Other Usage Notes

This program is applicable to lakes where significant vertical stratification is absent. Most usually, such lakes are shallow. WC2D employs a non-linear bottom friction relation which is appropriate for shallow lakes where bottom friction is significant. More information on the program formulation may be found in Shanahan and Harleman (1982).

The finite difference computation is limited by the Courant stability condition (Shanahan and Harleman, 1982) and requires:

$$\Delta t < \frac{\Delta x}{\sqrt{gH}}$$

where Δt is the computation time step,
Δx is the grid size,
g is gravitational acceleration, and
H is water depth

The program calls the IBM system routines WHEN and TIMING. WHEN returns the date and time of the simulation, TIMING returns elapsed CPU time. More information on the routines is given in Appendix B.

2.5 Current Vector Plotting with PLOT2D

If specified on input, WC3D will create an output file on I/0 Unit 9 containing data for later plotting by PLOT2D. PLOT2D is executed from an interactive terminal to create off-line plots on a CalComp or similar plotting machine. It may also be used to prepare run-time plots on a graphics terminal if appropriate CalComp prescreening software is used (for example, Tektronix has such software). Plots are made of the depth-averaged current throughout the lake as a function of time. Examples may be seen in Figures 5.9 and 6.5 in Shanahan and Harleman (1982).

PLOT2D is similar to PLOT3D in most respects. It differs in that it has been modified to plot depth average velocity only and to call CalCOMP rather Tektronix plotting routines. The CalComp routines are standard with the exception of routine NUMBR1. NUMBR1 is a system routine available at MIT. The calling sequence for NUMBR1 is included in Appendix B. Routines with similar characteristics may be available on other systems; if not, the standard routine NUMBER may be substituted.

The user executes PLOT2D by replying to a series of program questions. To execute, the user must have available the necessary system plotting library and have defined the necessary I/0 units. I/0 units are shown in Table 2.2.

Table 2.2

I/0 Unit	RECFM (Record Format)	LRECL (Record Length)	BLKSIZE (Block Size)	Definition
5	FB	132	132	Terminal
6	FB	132	132	Terminal
11	VS	404	800	WC2D output file
12	VS	504	508	CalComp plotting instructions output file

2.6 Water Surface Elevation and Mass Flux Plots

These plots are produced from the output files saved by WC2D using the plotting program WSPLOT. Execution of WSPLOT is described in Section 1.6; execution and other information is identical for output from WC2D and WC3D.

REFERENCES

Cooper, C.K. and B.R. Pearce, "A Three-Dimensional Model to Calculate Currents in Coastal Waters Utilizing a Depth Varying Vertical Eddy Viscosity," Report Number 226, Ralph M. Parsons Laboratory for Water Resources and Hydrodynamics, Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, August, 1977.

Hicks, B.B., R.L. Drinkrow, and G. Grauze, "Drag and Bulk Transfer Coefficients Associated with a Shallow Water Surface," Boundary Layer Meteorology, 6, 287, 1974.

Nelson, S.M., "Application of a Numerical Model for Three-Dimensional Sheared Flows," M.S. Thesis, Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, September, 1979.

Shanahan, P., D.R.F. Harleman and L. Somlyódy, "Modeling Wind-Driven Circulation in Lake Balaton," Collaborative Paper CP-81-7, International Institute of Applied Systems Analysis, Laxenburg, Austria, March, 1981.

Shanahan, P. and D.R.F. Harleman, "Linked Hydrodynamic and Biogeochemical Models of Water Quality in Shallow Lakes," Report Number 268, Ralph M. Parsons Laboratory for Water Resources and Hydrodynamics, Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, March, 1982.

Van Dorn, W.G., "Wind Stress on an Artificial Pond," Journal of Marine Research, 12:3, 249, 1953.

Wu, J. "Wind Stress and Surface Roughness at Air-Sea Interface," Journal of Geophysical Research, 74:2, 444, January 15, 1969.

Appendix A

Errata for

Nelson (1979) and

Shanahan et al. (1981)

APPLICATION OF A NUMERICAL MODEL FOR THREE-DIMENSIONAL SHEARED FLOWS
By Susan M. Nelson, M.S. Thesis, Department of Civil Engineering,
Massachusetts Institute of Technology, September, 1979.

Errata

Page 21 Equation (2.12) should be:

$$K = 0.6 \times 10^{-6}\, W^{1/2} \qquad W < 13.7 \text{ m/s}$$

$$K = 3.1 \times 10^{-6} \qquad W > 13.7 \text{ m/s}$$

Page 29 The term ρ is missing in:

or, after integration

$$-fH \left[\frac{\tau_{SY}H-}{(\alpha_1 H+\beta_1)\underset{\uparrow}{\rho}} (\psi_I-\gamma_I) + \frac{\tau_{SY}}{\rho}\Gamma_I + \frac{d_I}{2}\right]$$

change $\phi_I = \frac{\cos a_I}{a_I}\ldots$ to $\phi_I = \frac{\cos a_I}{a_I^2}\ldots$

Page 32 The second expression on the page should be:

$$\gamma_{KI} = \frac{2\zeta_{K+I}\cos a_I\zeta_{K+1}}{a_1^2} + \left(\frac{a_I^2\, \overset{\downarrow}{\zeta}{}^2_{K+1}-2}{a_I^3}\right)\sin a_I\zeta_{K+1}$$

$$- \frac{2\zeta_K\cos a_I\zeta}{a_I^2} - \left(\frac{a_I^2\overset{\downarrow}{\zeta}{}_K^2 - 2}{a_I^3}\right)\sin a_I\zeta_K$$

Replace the expression

$$\sum_{J=1}^{J'} c_J\delta_{JI} = \sum_{J=1}^{J'}\frac{a_Jc_J}{2}\sum_{K=1}^{K'-1}\alpha_K\lambda_{KJI}$$

with

$$\delta_{ji} = \frac{a_J}{2}\sum_{K=1}^{K'-1}\alpha_K\lambda_{KJI}$$

Page 36 Replace $B_I = \frac{2}{a_I} \sin a_I$ with $B_I = \frac{g}{a_I} \sin a_I$

The expression for A_{KI} should be:

$$A_{KI} = \frac{1}{(\alpha_1 H+\beta_1)}\left[\sum_{K=1}^{\overset{\prime}{K-1}}{}_{\uparrow} H\alpha_K(9\gamma_{KI} - 4\phi_{KI})\right.$$

$$\left. + \beta_K(6\phi_{KI}-2S_{KI})_{\uparrow}\right] + \nu_I^{\prime} - \nu_I$$

Page 37 In the second line of the expression for G change the sign to read:

$$\underset{\uparrow}{-}\frac{1}{\alpha_1}(\beta_1 \mathrm{Ln}\beta_1-\beta_1)]$$

Replace $S_{1J} = \frac{\sin(\zeta_2 a_J)}{a_J}$ with $S_{1J} = \frac{\sin a_J}{a_J}$

MODELING WIND-DRIVEN CIRCULATION IN LAKE BALATON
By Peter Shanahan, Donald R.F. Harleman and Lászlό Somlyόdy
IIASA Collaborative Paper CP-81-7 March, 1981

Errata

Page 3-1, Equation 3.2

Replace: $\frac{\partial u}{\partial t} + \{u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial x}\} =$

with: $\frac{\partial u}{\partial t} + \{ u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\underset{\uparrow}{\partial z}}\} =$

Page 3-3, Figure 3.1 and Equation 3.5

Replace: η with: $\underset{\uparrow}{-}\ \eta$ (3 corrections)

Page 3-4, Equation 3.6

Equation should read:

$$\underset{\uparrow}{-}\rho A_v \frac{\partial u}{\partial z} = \tau_s^x \quad \underset{\uparrow}{-}\rho A_v \frac{\partial v}{\partial z} = \tau_s^y \quad \text{at} \quad z = \underset{\uparrow}{-}\eta$$

Page 3-4, Equation 3.8

Equation should read:

$$\underset{\uparrow}{-}\rho A_v \frac{\partial u}{\partial z} = \tau_b^x \quad \underset{\uparrow}{-}\rho A_v \frac{\partial v}{\partial z} = \tau_b^y \quad \text{at } z = h$$

Page 3-6, Equations 3.15 and 3.16

Replace: $- \frac{\partial}{\partial z} [E_v \frac{\partial u}{\partial z}]$ with: $\underset{\uparrow}{+} \frac{\partial}{\partial z} [E_v \frac{\partial u}{\partial z}]$ in 3.15

and similarly in 3.16

Page 3-7, Equation 3.18

Replace: at $z = \eta \frac{Fr^2}{R_o}$ with: at $z = \underset{\uparrow}{-}\eta \frac{Fr^2}{R_o}$

Page 3-7, Equation 3.19

Equation should read:

$$\underset{\uparrow}{-}\,\frac{\partial u}{\partial z} = \tau_s^x \qquad \underset{\uparrow}{-}\,\frac{\partial v}{\partial z} = \underset{\uparrow}{\tau_s^y} \quad \text{at } z = \underset{\uparrow}{-}\eta\,\frac{Fr^2}{R_o}$$

Page 3.7, Equation 3.20

Equation should read:

$$\underset{\uparrow}{-}\,\frac{\partial u}{\partial z} = \underset{\uparrow}{\tau_b^x} \qquad \underset{\uparrow}{-}\,\frac{\partial v}{\partial z} = \underset{\uparrow}{\tau_b^y} \quad \text{at} \quad z = \underset{\uparrow}{h}$$

Page 3-8, Equation 3.21

Replace: η with: $\underset{\uparrow}{-}\eta$ as upper limit of integrals (2 corrections)

Page 4-4

Second paragraph, line 6 and last line

Change: η to: $-\eta$

Page 5-9, Equation 5.2

Replace: at $z = H$ with: at $z = \underset{\uparrow}{h}$

Page 5-9, Equation 5.3

Equation should read:

$$\underset{\uparrow}{-}\rho A_v \frac{\partial u}{\partial z} = \tau_b^x \qquad \underset{\uparrow}{-}\rho A_v \frac{\partial v}{\partial z} = \tau_b^y$$

Page 5-9, Line immediately before Eq. 5.4 should read:

$\underset{\uparrow}{h}$, such that:

Page 5-9, Equation 5.4

Equation should read:

$$\underset{\uparrow}{h} > 4\sqrt{\frac{2A_v}{f}} = \frac{4D}{\underset{\uparrow}{\pi}}$$

Page 5-9, following Equation 5.4, equation should read:

$$D = \pi \sqrt{\frac{2A_v}{f}}$$

Page 7-2, 4 lines from bottom

Replace: at $z = \eta$ with: at $z = \underset{\uparrow}{-}\eta$

Page 4-5, the last of the variable definitions should be for:

$E_{\underset{\uparrow}{v}}$ and E_H

Page 5-10, Equation 5.6

Replace: C_b (upper case) with : c_b (lower case) (2 corrections)

Page 5-11, Equation 5.7

In the denominator on the right hand side of both equations, C should be C^2 (2 corrections)

Page B-1, Equation B.1

Equations should read:

$$L_x(u,v) = \frac{\partial u}{\partial t} - \frac{\partial}{\partial z} (A_v \frac{\partial u}{\partial z}) - fv + g \frac{\partial \eta}{\partial x} = 0$$

$$\underset{\uparrow}{L_y}(u,v) = \frac{\partial v}{\partial t} - \frac{\partial}{\partial z} (A_v \frac{\partial v}{\partial z}) \underset{\uparrow}{+} fu \underset{\uparrow}{+} g \underset{\uparrow}{\frac{\partial \eta}{\partial y}} = 0$$

Page B-1, Equations B.2

Equations should read:

$$R_x = L_x(\hat{u},\hat{v}) - L_x(u,v) = L_x(\hat{u},\hat{v})$$

$$R_y = L_y(\hat{u}\underset{\uparrow}{,}\hat{v}) - L_y(u\underset{\uparrow}{,}v) = L_y(\hat{u}\underset{\uparrow}{,}\hat{v})$$

Page B-4, the line immediately after Equation B.7b should read:

where $\underset{\uparrow}{\alpha_1}$ and $\underset{\uparrow}{\beta_1}$ are ...

Page B-5, second line of second paragraph

Replace: ...for u and v into...

with: ...for $\underset{\uparrow}{\hat{u}}$ and $\underset{\uparrow}{\hat{v}}$ into...

Page B-5, immediately before Equation B.10

Replace: Substituting for R:

with: Substituting for $\underset{\uparrow}{R_x}$:

Page B-5, Equation B.10 on the bottom line

Replace: $-f\langle v,\Omega_i\rangle - g\frac{\partial\eta}{\partial x}\langle 1,\Omega_i\rangle$

with: $-f\langle \underset{\uparrow}{\hat{v}},\Omega_i\rangle \underset{\uparrow}{+} g\frac{\partial\eta}{\partial x}\langle 1,\Omega_i\rangle$

Page B-6, the equation at the top of the page should read:

$$\langle \hat{u},\Omega_i\rangle = \langle u_o,\Omega_i\rangle + c_i \langle\Omega_i,\underset{\uparrow}{\Omega_i}\rangle$$

Page B-7, the line immediately after Eq. 7.4

Replace: from u and v.

with: from $\underset{\uparrow}{\hat{u}}$ and $\underset{\uparrow}{\hat{v}}$.

Page B-7, Equations B.12

On the right hand sides of the equations, the term: k_i (lower case) should be: K_i (upper case) (2 corrections)

Page B-7, Equation B.13 and the immediately preceding equation, the term: $\underline{k}$ (lower case) should be: $\underline{K}$ (upper case) (4 corrections)

Page B-8, third equation (the continuity equation)

Replace the term $\underline{k}$ (lower case) with $\underline{K}$ (upper case) (2 corrections)

Page B-9, at the end of the first line,

Replace: The values of c

with: The values of $c_{\underset{\uparrow}{i}}$

Page B-9, Equation B.14a in the first term of the second line

Replace: τ^x

with: $\tau^x_{\underset{\uparrow}{s}}$

A similar change is required in B.14b.

Page B-9, near the bottom of the page

Replace: τ^x and τ^Y

with: $\tau^x_{\underset{\uparrow}{s}}$ and $\tau^Y_{\underset{\uparrow}{s}}$

Page B-5, Equation B.10, second line

Replace: $<A_v \frac{\partial^2 u}{\partial z^2}, \Omega_i>$

With: $<A_v \frac{\partial^2 \hat{u}}{\partial \underset{\uparrow}{z}^2}, \Omega_i>$

Appendix B

System Routines

NAME: TIMING, to obtain the central processor (CPU) time charged

DESCRIPTION: TIMING can be used to measure the amount of Central Processor Unit (CPU) time for which a computation within a job step has been charged. On the MVT version of the Operating System, this value can vary by 10% or more, depending on the total workload. Furthermore, because the value is rounded to hundredths of a second, its relative accuracy is poor for very short computations.

USAGE: A FORTRAN call is:

```
CALL TIMING (ICPU)
```

The value of the output variable is:

```
ICPU the CPU time in hundredths of a second
```

An appropriate declaration is:

```
INTEGER*4 ICPU
```

EXAMPLE: The following example illustrates the use of TIMING to store in ITIME the CPU time charged for a computation.

```
INTEGER*4 ISTART, ISTOP, ITIME
CALL TIMING (ISTART)
     . . .
     Computation to be timed
     . . .
CALL TIMING (ISTOP)
ITIME = ISTOP - ISTART
```

NAME: WHEN, to obtain the date and time of day

DESCRIPTION: WHEN provides the date and time of day in a form suitable for printing. The accuracy of the date and time is dependent on the accuracy with which the Operations staff has set them at system initialization.

USAGE: A FORTRAN call is:

```
     CALL WHEN(DAYTIM)
```

The output value is:

DAYTIM the date and time of day in alphanumeric form.

The appropriate declaration is:

```
     REAL*4 DAYTIM(5)
```

EXAMPLES: The following example illustrates the use of WHEN to print the date and time of day:

```
     REAL*4 DAYTIM(5)
     CALL WHEN (DAYTIM)
     WRITE(6,100) DAYTIM
100  FORMAT(1H,5A4)
```

The printed output will be in the form:

```
     MO¦DD¦YY HH*MI*SS*CC
```

where MO is the month, DD the day, YY the year, HH the hour, MI the minute, SS the second, and CC hundredths of a second.

If only the date is desired, the following example may be followed:

```
     REAL*4 DATE(5)
     CALL WHEN(DATE)
     WRITE(6,100)(DATE(I),I=1,2)
100  FORMAT(1H, 'DATE IS: ',2A4)
```

Similarly, if only the time of day is desired, the following example may be followed:

```
     REAL*4 TIME(5)
     CALL WHEN(TIME)
     WRITE(6,100)(TIME(I), I=3,5)
100  FORMAT(1H, 'TIME OF DAY IS:  ',3A4)
```

```
NUMBR1(XPAGE,YPAGE,± SIZE,  NUM  ,ANGLE,± NDIGIT)
                            FNUM
```

All arguments are input to this subroutine.

This subroutine differs from the standard CalComp routine NUMBER in that it will draw either an integer (NUM) or a floating point number (FNUM) at a point by converting it to an equivalent character representation and then calling the IPC-supplied subroutine SYNBL5. The value of NUM cannot exceed $2^{31}-1$ and FNUM can have up to seven digits to the left of the decimal point and 12 digits to the right.

XPAGE,YPAGE are the physical page coordinates (i.e., plotter space) in inches from the reference point where the number will be drawn.

SIZE is the size of the number in inches. If SIZE is positive, the number will be drawn using (XPAGE,YPAGE) as the lower left corner. If SIZE is negative, the number will be centered along its own X axis.

NUM or FNUM is the number to be drawn.

ANGLE is the rotation, in degrees from the +X direction, that the number will be rotated. Centered numbers will be rotated about their center point and uncentered numbers will be rotated about the lower-left corner of the first number.

NDIGIT If: NDIGIT is positive, it is the number of digits after the decimal point.
NDIGIT equals 0, no decimal point will be written.
NDIGIT is negative, the number is an integer.