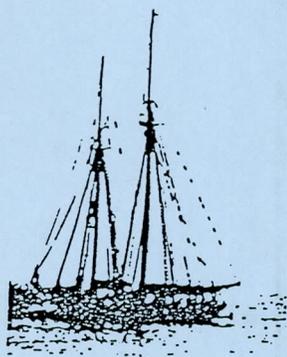


FINITE ELEMENT SURFACE-WATER MODELING SYSTEM:
TWO-DIMENSIONAL FLOW IN A HORIZONTAL PLANE

Property of
Flood Control District of MC Libran
Please Return to
2801 W. Du...
Phoenix, AZ 85009

FESWMS-2DH USERS MANUAL

Developed by
David C. Froehlich, Ph.D., P.E.



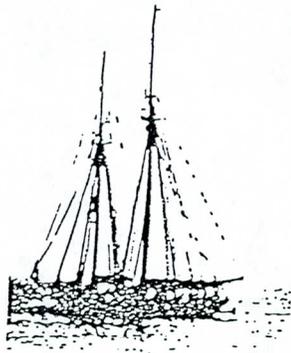
Distributed by
WEST Consultants, Inc.
2111 Palomar Airport Rd., Suite 180
Carlsbad, California 92009-1419

June 1991

FINITE ELEMENT SURFACE-WATER MODELING SYSTEM:
TWO-DIMENSIONAL FLOW IN A HORIZONTAL PLANE

FESWMS-2DH USERS MANUAL

Developed by
David C. Froehlich, Ph.D., P.E.



Distributed by
WEST Consultants, Inc.
2111 Palomar Airport Rd., Suite 180
Carlsbad, California 92009-1419

June 1991

SI* (MODERN METRIC) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
--------	---------------	-------------	---------	--------

LENGTH

in	inches	25.4	millimetres	mm
ft	feet	0.305	metres	m
yd	yards	0.914	metres	m
mi	miles	1.61	kilometres	km

AREA

in ²	square inches	645.2	millimetres squared	mm ²
ft ²	square feet	0.093	metres squared	m ²
yd ²	square yards	0.836	metres squared	m ²
ac	acres	0.405	hectares	ha
mi ²	square miles	2.59	kilometres squared	km ²

VOLUME

fl oz	fluid ounces	29.57	millilitres	mL
gal	gallons	3.785	litres	L
ft ³	cubic feet	0.028	metres cubed	m ³
yd ³	cubic yards	0.765	metres cubed	m ³

NOTE: Volumes greater than 1000 L shall be shown in m³.

MASS

oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2000 lb)	0.907	megagrams	Mg

TEMPERATURE (exact)

°F	Fahrenheit temperature	$5(F-32)/9$	Celsius temperature	°C
----	------------------------	-------------	---------------------	----

APPROXIMATE CONVERSIONS FROM SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
--------	---------------	-------------	---------	--------

LENGTH

mm	millimetres	0.039	inches	in
m	metres	3.28	feet	ft
m	metres	1.09	yards	yd
km	kilometres	0.621	miles	mi

AREA

mm ²	millimetres squared	0.0016	square inches	in ²
m ²	metres squared	10.764	square feet	ft ²
ha	hectares	2.47	acres	ac
km ²	kilometres squared	0.386	square miles	mi ²

VOLUME

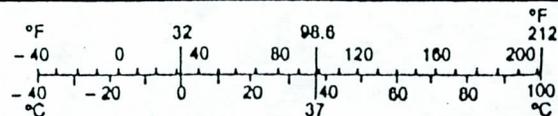
mL	millilitres	0.034	fluid ounces	fl oz
L	litres	0.264	gallons	gal
m ³	metres cubed	35.315	cubic feet	ft ³
m ³	metres cubed	1.308	cubic yards	yd ³

MASS

g	grams	0.035	ounces	oz
kg	kilograms	2.205	pounds	lb
Mg	megagrams	1.102	short tons (2000 lb)	T

TEMPERATURE (exact)

°C	Celsius temperature	$1.8C + 32$	Fahrenheit temperature	°F
----	---------------------	-------------	------------------------	----



* SI is the symbol for the International System of Measurement

TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
1. INTRODUCTION.....	1-1
2. OVERVIEW OF THE MODELING SYSTEM.....	2-1
Modeling System Identification.....	2-2
Physical System Highlights.....	2-3
Modeling System Applications.....	2-3
Methodology.....	2-5
Input and Output Data.....	2-6
Graphic Output.....	2-7
3. BASIC CONCEPTS OF THE FINITE ELEMENT METHOD.....	3-1
Method of Weighted Residuals.....	3-2
Elements and Interpolation Functions.....	3-4
Numerical Integration.....	3-15
4. GOVERNING EQUATIONS.....	4-1
Depth-Averaged Flow Equations.....	4-1
Momentum Correction Coefficients.....	4-5
Coriolis Parameter.....	4-6
Bed Shear Stresses.....	4-7
Surface Shear Stresses.....	4-8
Stresses Caused by Turbulence.....	4-9
Weir Flow and Roadway Overtopping.....	4-11
Bridge and Culvert Flow.....	4-12
One-Dimensional Bridge/Culvert Flow.....	4-13
Type 4 flow discharge coefficients.....	4-15
Type 5 flow discharge coefficients.....	4-17
Two-Dimensional Bridge/Culvert Flow.....	4-20
Initial and Boundary Conditions.....	4-22
Initial Conditions.....	4-22
Boundary Conditions.....	4-23
Solid boundary.....	4-23
Open boundary.....	4-24
5. FINITE ELEMENT EQUATIONS.....	5-1
Residual Expressions.....	5-1
Time Derivatives.....	5-3
Derivative Expressions.....	5-5
Application of Boundary and Special Conditions.....	5-7
Open Boundaries.....	5-7
Solid Boundaries.....	5-10
Total Flow Across a Boundary.....	5-11
Depth-Averaged Pressure Flow.....	5-15
Residual Expressions.....	5-15
Derivative Expressions.....	5-16

TABLE OF CONTENTS (continued)

<u>Section</u>	<u>Page</u>
6. MODELING SYSTEM OPERATION.....	6-1
Data Collection.....	6-1
Network Design.....	6-3
General Network Layout.....	6-4
One-Dimensional Weirs and Culverts.....	6-8
Two-Dimensional Bridges.....	6-9
Calibration.....	6-12
Validation.....	6-15
Application.....	6-15
7. PROGRAM LOGIC AND DATA FLOW.....	7-1
Data Input Module: DINMOD.....	7-1
Error Checking.....	7-5
Automatic Grid Generation.....	7-5
Element Resequencing.....	7-11
Minimum frontgrowth method.....	7-13
Level structure method.....	7-14
Network Refinement.....	7-14
Depth-Averaged Flow Module: FLOMOD.....	7-15
Error Checking.....	7-17
Equation Solution.....	7-19
Solution strategy.....	7-20
Frontal solution scheme.....	7-22
Continuity Norm.....	7-24
Automatic Boundary Adjustment.....	7-25
Initial Conditions and Convergence.....	7-27
Cold starts.....	7-27
Hot starts.....	7-28
Convergence problems.....	7-28
Specifying Boundary Conditions.....	7-29
Total flow at a cross section.....	7-34
Water-surface elevation at a cross section.....	7-34
Output Analysis Module: ANOMOD.....	7-35
8. INPUT DATA OVERVIEW.....	8-1
Input Data files.....	8-3
Network Data File.....	8-3
Initial Condition (Flow) Data File.....	8-4
Boundary Condition Data File.....	8-5
Wind Data File.....	8-6
Interactive Data File Editors.....	8-7

TABLE OF CONTENTS (continued)

<u>Section</u>	<u>Page</u>
9.	INPUT DATA DESCRIPTION: DINMOD..... 9-1
	Program Control Data Set..... 9-2
	Plot Data Set..... 9-5
	Element Data Set..... 9-8
	Node Data Set..... 9-9
	Node Interpolation Data Set..... 9-11
	Automatic Grid Generation Data Set..... 9-12
	Resequencing Data Set..... 9-13
10.	INPUT DATA DESCRIPTION: FLOMOD..... 10-1
	Program Control Data Set..... 10-2
	Element Data Set..... 10-12
	Node Data Set..... 10-13
	Property Data Set..... 10-15
	Initial Condition Data Set..... 10-17
	Boundary Condition Data Set..... 10-18
	Total Flow Cross-Section Data Set..... 10-21
	Water-Surface Elevation Cross-Section Data Set..... 10-22
	Wind Data Set..... 10-25
	Flow Check Data Set..... 10-26
	Weir Data Set..... 10-27
	Culvert Data Set..... 10-28
	Time-Dependent Data Set..... 10-30
11.	INPUT DATA DESCRIPTION: ANOMOD..... 11-1
	Program Control Data Set..... 11-2
	Element Data Set..... 11-5
	Node Data Set..... 11-6
	Flow Data Set(s)..... 11-8
	Network Plot Data Set..... 11-9
	Vector Plot Data Set..... 11-12
	Contour Plot Data Set..... 11-16
	Flow Check Report and Plot Data Set..... 11-20
	Time-History Report and Plot Data Set..... 11-23
12.	OUTPUT DATA DESCRIPTION..... 12-1
	Data Input Module: DINMOD..... 12-1
	Printed Output..... 12-1
	Output Data Files..... 12-3
	Graphic Output..... 12-3
	Depth-Averaged Flow Module: FLOMOD..... 12-3
	Printed Output..... 12-4
	Output Data Files..... 12-7
	Restart/recovery data file..... 12-7
	Flow data file..... 12-7

TABLE OF CONTENTS (continued)

<u>Section</u>	<u>Page</u>
Analysis of Output Module: ANOMOD.....	12-8
Printed Output.....	12-8
Graphic Output.....	12-8
Plotfile Format.....	12-9
Instruction Description.....	12-9
13. RUN-PREPARATION INSTRUCTIONS.....	13-1
Run-Stream Description.....	13-1
Resource Requirements.....	13-1
Restart/Recovery Procedure.....	13-8
Input/Output File Name Assignment.....	13-8
14. TROUBLE SHOOTING.....	14-1
Warning Messages.....	14-1
Error Messages.....	14-3
15. REFERENCES.....	15-1
Appendix A: INPUT DATA WORKSHEETS.....	A-1
Appendix B: MICROCOMPUTER IMPLEMENTATION.....	B-1
Appendix C: EXAMPLE INPUT DATA FILES.....	C-1

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
3-1. Examples of the three types of two-dimensional elements used in FESWMS-2DH: (A) A 6-node triangle; (B) an 8-node "serendipity" quadrilateral; and (C) a 9-node "Lagrangian" quadrilateral.....	3-7
3-2. Illustration of coordinate mapping of a triangular and a quadrangular element.....	3-8
3-3. Natural coordinate system and interpolation functions for a triangular parent element.....	3-10
3-4. Natural coordinate system and interpolation functions for a "serendipity" quadrangular parent element.....	3-11
3-5. Natural coordinate system and interpolation functions for a "Lagrangian" quadrangular parent element.....	3-12
3-6. Numerical integration point coordinates and weighting factors for triangular and quadrangular parent elements.....	3-17
4-1. Illustration of the coordinate system and variables used to derive the depth-averaged surface-water flow equations.....	4-3
4-2. Illustration of depth-averaged velocity.....	4-4
6-1. Some rules for one-to-one mapping of two-dimensional isoparametric finite elements.....	6-6
6-2. A finite element network at a roadway embankment that contains a culvert and is divided into weir segments.....	6-10
6-3. A finite element network at a bridge where pressure flow within the bridge opening is modeled.....	6-11
7-1. Schematic diagram of the Data Input Module (DINMOD) showing the logical flow of data through the module and the major functions that are performed.....	7-3
7-2. Illustration of (A) a region inside which a finite element network is to be generated automatically, and (B) a simply connected initial subdivision of the region.....	7-7

LIST OF FIGURES (continued)

<u>Figure</u>	<u>Page</u>
7-3. Illustration of two triangular elements that are created by adding a corner node (node 4), the location of which is based on the coordinates of the two corner nodes (nodes 1 and 3) adjacent to the vertex.....	7-9
7-4. Neighborhood of node i in element k used in Laplacian smoothing of a network that has been generated automatically.....	7-10
7-5. An example of a finite element network that has been generated automatically: (A) An initial subdivision that is defined by a series of connected corner nodes; and (B) the network that was generated inside the initial subdivision.....	7-12
7-6. Elements are refined by dividing them into four similar elements: (A) A 6-node triangular element; (B) an 8-node quadrangular element; and (C) a 9-node quadrangular element.....	7-16
7-7. Schematic diagram of the Depth-Averaged Flow Module (FLOMOD) showing the logical flow of data through the module and the major functions that are performed.....	7-18
7-8. Example of boundary condition specifications where a solid boundary and an open boundary meet and unit flows are specified.....	7-32
7-9. Example of boundary condition specifications where a solid boundary and an open boundary meet and water-surface elevations are specified.....	7-33
7-10. Schematic diagram of the Analysis of Output Module (ANOMOD) showing the logical flow of data through the module and the major functions that are performed.....	7-38

LIST OF TABLES

<u>Table</u>	<u>Page</u>
4-1. Submergence factor, C_{sub} , for weir flow over a roadway embankment for various ratios of submergence.....	4-13
4-2. Type 4 flow discharge coefficient, C_c , for box or pipe culverts placed flush in a vertical headwall for various ratios of entrance rounding or beveling.....	4-15
4-3. Type 4 flow discharge coefficient, C_c , for box culverts that have wingwalls and a square, rounded, or beveled entrance.....	4-16
4-4. Discharge coefficient adjustment factor, k_r , that accounts for entrance rounding of pipe or box culverts placed flush in a vertical headwall.....	4-16
4-5. Discharge coefficient adjustment factor, k_w , that accounts for entrance beveling of pipe or box culverts placed flush in a vertical headwall.....	4-17
4-6. Discharge coefficient adjustment factor, k_L , for pipe or pipe-arch culverts that extend beyond a headwall or embankment.....	4-18
4-7. Type 5 flow discharge coefficient, C_c , for box or pipe culverts placed flush in a vertical headwall for various ratios of entrance submergence and entrance rounding or beveling.....	4-19
4-8. Type 5 flow discharge coefficient, C_c , for box culverts that have wingwalls for various ratios of entrance submergence and various wingwall angles..	4-23
6-1. Summary of data that may be needed to apply the modeling system.....	6-2
13-1. Run-stream representation for the Data Input Module: DINMOD.....	13-2
13-2. Run-stream representation for the Depth-Averaged Flow Module: FLOMOD.....	13-3

LIST OF TABLES (continued)

<u>Table</u>		<u>Page</u>
13-3.	Run-stream representation for the Analysis of Output Module: ANOMOD.....	13-6

LIST OF SYMBOLS

<u>Symbol</u>	<u>Definition</u>
a_i	a vector of unknown nodal values at the i th iteration
Δa_i	change of the solution vector a_i at the i th iteration
a_i^o	coefficient used to prescribe flow normal to an open boundary at node i
b_i^o	coefficient used to prescribe flow normal to an open boundary at node i
a_i^s	coefficient used to prescribe flow normal to a solid boundary at node i
b_i^s	coefficient used to prescribe flow normal to a solid boundary at node i
A_c	cross-section area of a culvert
A_e	area of an element
c_f	bottom shear stress coefficient (dimensionless)
c_β	momentum correction coefficient model coefficient
c_s	wind stress model coefficient
c_{s1}	wind stress model coefficient
c_{s2}	wind stress model coefficient
c_μ	kinematic eddy viscosity model coefficient
C	Chézy discharge coefficient
C_c	culvert discharge coefficient (dimensionless)
C_i	an unknown parameter at node i
C_{sub}	weir submergence coefficient
C_w	weir discharge coefficient (dimensionless)

LIST OF SYMBOLS (continued)

<u>Symbol</u>	<u>Definition</u>
D	inside diameter of a pipe culvert or the maximum inside height of a pipe-arch culvert
e_*	dimensionless diffusivity
f	a known function
g	gravitational acceleration
H	water depth
\bar{H}	average water depth along an element side
H_i^*	value of H specified at node i
[J]	Jacobian matrix
J	determinant of a Jacobian matrix
k_r	culvert discharge coefficient adjustment factor that accounts for entrance rounding
k_w	culvert discharge coefficient adjustment factor that accounts for entrance beveling
k_L	culvert discharge coefficient adjustment factor that accounts for extension of the culvert beyond a headwall or embankment
K	cross-section conveyance; a matrix of assembled element coefficients
K_c	culvert coefficient
K_w	weir coefficient
ℓ_x	direction cosine between the outward normal to a network boundary and the x-coordinate direction
ℓ_y	direction cosine between the outward normal to a network boundary and the y-coordinate direction
\mathcal{L}	differential operator
L_c	length of a culvert

LIST OF SYMBOLS (continued)

<u>Symbol</u>	<u>Definition</u>
L_i	number of elements connected to node i
L_w	length of a weir segment
L_p	distance a culvert extends beyond a headwall or embankment
M_i	a linear interpolation function associated with node i in Manning roughness coefficient
n	Manning roughness coefficient
n_c	Manning roughness coefficient for a culvert
N_e	element continuity norm
N_i	an interpolation function associated with node i
N'	a coordinate interpolation function associated with node i
P	pressure head
Q	source/sink (inflow/withdrawal) at a node point.
Q_c	flow through a culvert
Q_{ci}	total culvert flow across a solid boundary at node i
Q_i^o	flow across an open boundary resulting from flow at node i
Q_i^s	flow across a solid boundary resulting from flow at node i
Q_i^{o*}	flow normal to an open boundary at node i that is directly specified
Q_i^{s*}	flow normal to a solid boundary at node i that is directly specified
Q_w	flow over a weir segment

LIST OF SYMBOLS (continued)

<u>Symbol</u>	<u>Definition</u>
Q_{wi}	total weir flow across a solid boundary at node i
Q_{xi}	amount of the total flow through a cross section that is assigned to node i
r_i	updating vector used in a quasi-Newton solution
R	a solution domain; a residual load vector
R_c	hydraulic radius of a culvert flowing full
R_H	continuity equation residual
S_e	element boundary
S_e^O	element boundary that is part of the open boundary of a network
S_e^S	element boundary that is part of the solid boundary of a network
t	time
u	an exact value of a dependent variable; point velocity in the x direction
\tilde{u}	an approximation of a dependent variable
U	depth-averaged velocity in the x-coordinate direction
U_i^*	value of U specified for node i
U_n	depth-averaged velocity normal to a boundary
U_t	velocity tangent to a boundary
U_*	bed shear velocity
v	point velocity in the y direction
V	depth-averaged velocity in the y-coordinate direction
V_i^*	value of V specified for node i

LIST OF SYMBOLS (continued)

<u>Symbol</u>	<u>Definition</u>
w	point velocity in the z direction
w_i	numerical integration weighting factor at the i th integration point
W	wind velocity
W_i	a weighting function corresponding to node point i
W_{min}	minimum wind velocity used to compute surface wind stress coefficient
x	a Cartesian coordinate (a horizontal direction)
y	a Cartesian coordinate (a horizontal direction)
z	a Cartesian coordinate (the vertical direction)
z_b	bed elevation
z_c	crest elevation of a weir segment, ceiling elevation
z_e^h	energy elevation at the upstream (headwater) side of a weir segment
z_{inv}	invert elevation at a culvert entrance
z_s^t	water-surface elevation at the downstream (tailwater) side of a weir segment or the downstream end of a culvert
α	coefficient used to compute time derivatives
β	isotropic momentum correction coefficient
β_{uv}	directional component of the momentum correction coefficient
β_o	momentum correction coefficient model constant
δ	angle between the positive x direction and a tangent to the boundary at node i
δ_i	updating vector used in a quasi-Newton solution

LIST OF SYMBOLS (continued)

<u>Symbol</u>	<u>Definition</u>
$\tilde{\Gamma}$	kinematic eddy diffusivity a residual
ζ	factor used to distribute cross section conveyance among node points that form the cross section
η	natural coordinate
θ	time-integration weighting coefficient
κ	von Karman's constant
$\tilde{\nu}$	isotropic depth-averaged kinematic eddy viscosity
$\tilde{\nu}_{xy}$	directional component of depth-averaged kinematic eddy viscosity
$\tilde{\nu}_0$	kinematic eddy viscosity model constant
ξ	natural coordinate
ρ	density of water
ρ_a	density of air
σ_t	turbulent Prandtl or Schmidt number
τ_x^b, τ_y^b	bottom shear stresses in the x and y direction, respectively
τ_x^c, τ_y^c	ceiling shear stresses in the x and y directions, respectively
τ_x^s, τ_y^s	surface shear stresses caused by wind in the x and y directions, respectively
τ_{xy}	shear stress caused by turbulence
τ_w	wall shear stress
ϕ	angle of latitude
ω	angular velocity; Laplacian smoothing algorithm coefficient
ω_r	iterative solution over/under-relaxation parameter

LIST OF SYMBOLS (continued)

<u>Symbol</u>	<u>Definition</u>
Ω	Coriolis parameter
∂	partial differential symbol

This page is blank.

Section 1

INTRODUCTION

The purpose of this manual is to provide nonprogramming users of the Finite Element Surface Water Modeling System: Two-Dimensional Flow in a Horizontal Plane (hereafter referred to as FESWMS-2DH) the information needed to use the modeling system effectively. The manual provides sufficient description of the programs that comprise the modeling system to allow users to determine when and how the system can be used, and will serve as a reference document for preparation of input data and interpretation of results.

A user is assumed to be interested mainly in obtaining results from the modeling system for specific applications. To apply the modeling system and interpret results effectively, a user needs to be aware of the logical structure of the modeling system, the general simulation approach, and the assumptions and limitations that affect use of the system. A user does not need to be interested in the details of programming beyond the required formats of input data and the presentation of results. The rest of this manual enables a nonprogramming user of FESWMS-2DH to understand the basic logic of the modeling system, the input data requirements, the flow of data through the modeling system, the output generated by the modeling system, and all limitations that affect application of the modeling system.

FESWMS-2DH applies the finite element method to solve the system of equations that govern two-dimensional flow in a horizontal plane. An overview of the modeling system is provided in section two to help a user determine the applicability of the modeling system for specific needs. To understand the general ideas of the solution procedure, some basic concepts of the Finite element method are described in section three. The

governing equations are presented in section four so that a user will understand how results are obtained and how empirical coefficients are used. Finite element equations that are formed by applying the finite element method to the governing equations are described in section five so that a user will understand how boundary conditions and other special conditions are prescribed, and how the equations that are formed are solved. Section six describes how the modeling system can be used to solve a surface-water flow problem, and includes discussions of the following: (1) Data collection, (2) finite element network design, (3) calibration of a model, (4) validation of a model, and (5) application of a model to evaluate the effects of natural or manmade influences. Section seven describes the logical flow of data through the modeling system, from the entry of input data to the generation of output data. Section eight through eleven describe in detail all the input data needed to run FESWMS-2DH. The material in these sections will serve as a reference for anyone who runs the modeling system. Section 12 describes in detail all the output produced by the modeling system, including their meaning and use. Section 13 describes procedures for organizing input data to be submitted for a computer run. Section 14 is a tabulation of warning and error messages produced by the modeling system. Suggested corrective actions are described for each error or warning. Section 15 is a list of references cited in the manual. The appendix contains worksheets that simplify entry of input data. Only one copy of each input data worksheet is included in the appendix. Copies of the original worksheets can be made as needed to assist in entering most of the input data required by FESWMS-2DH.

Section 2

OVERVIEW OF THE MODELING SYSTEM

FESWMS-2DH is a modular set of computer programs developed to simulate surface-water flow where the flow is essentially two-dimensional in a horizontal plane. The programs that comprise the modeling system have been designed specifically to analyze flow at bridge crossings where complicated hydraulic conditions exist. However, the modeling system can be applied to many other types of steady and unsteady surface-water flows. Three separate, but interrelated, programs form the core of the modeling system: (1) The Data Input Module (DINMOD), (2) the Depth-Averaged Flow Module (FLOMOD), and (3) the Analysis of Output Module (ANOMOD).

DINMOD acts as a data pre-processor in the modeling system. The primary purpose of DINMOD is to generate a two-dimensional finite element network (also called a finite element grid) that is error free. Functions performed by DINMOD include editing of input data, automatic generation of all or part of the finite element network, refinement of an existing network, ordering of elements to enable an efficient equation solution, and graphic display of the finite element network. Processed network data can be stored in a data file for use by other FESWMS-2DH programs.

FLOMOD applies the finite element method to solve the governing system of equations using the defined network. FLOMOD can simulate both steady and unsteady (time-dependent) two-dimensional (in a horizontal plane) surface-water flow to obtain depth-averaged velocities and flow depths. The effects of bed friction and turbulent stresses are considered, as are, optionally, surface wind stresses and the Coriolis force. Pressure flow through bridges is considered if the water is in

contact with the bottom of the bridge deck which is defined by a "ceiling" elevation at a node point. Flow over weirs, or weir-type structures (such as highway embankments), and flow through culverts can also be modeled. The computed two-dimensional flow data can be written to a data file and stored for future use.

Results of flow simulations are presented graphically and in the form of reports by ANOMOD. Plots of velocity and unit-flow vectors; ground-surface and water-surface elevation contours; and time-history graphs of velocity, unit flow, or stage (water-surface elevation) at a computation point can be produced. Thus, ANOMOD acts as a post-processor in the modeling system.

Modeling System Identification

FESWMS-2DH is the result of an effort to provide a means of simulating flow at highway crossings where natural processes and manmade structures have created complicated hydraulic conditions that are difficult to evaluate using conventional methods. The first version of FESWMS-2DH was developed for the Federal Highway Administration by the author while with the U.S. Geological Survey, Water Resources Division. This enhanced version of the modeling system extends its simulation capabilities and provides improved data entry and graphic output features.

The computational modules of FESWMS-2DH are written in the Fortran 77 programming language as defined by the American National Standards Institute (1978). The controlling shell program, graphic display programs, and all other auxiliary programs are written in Turbo Pascal¹.

¹Turbo Pascal is a trademark of Borland International.

Physical System Highlights

In many surface-water flow problems of practical engineering concern, the three-dimensional nature of the flow is of secondary importance, particularly when the width-to-depth ratio of the water body is large. In such a case, the horizontal distribution of flow quantities may be the main interest, and two-dimensional flow approximations can be used to great economic advantage. In fact, the present state-of-the-art, and lack of suitable data, do not justify more complex three-dimensional solutions to most flow problems. Shallow rivers, flood plains, estuaries, harbors, and even coastal seas are examples of surface-water bodies where flows may be essentially two dimensional in character.

Throughout this manual, flow is assumed to be strictly two dimensional, except for the special cases of weir and culvert flow. A two-dimensional flow description is obtained by integrating the governing three-dimensional flow equations with respect to the depth of flow. Velocity in the vertical direction is assumed to be negligible, thus pressure in a column of water is considered to be hydrostatic. Flow depth and the resulting depth-averaged velocities are variable in a horizontal plane.

Modeling System Applications

FESWMS-2DH calculates depth-averaged horizontal velocities and water depth, and the time-derivatives of these quantities if a time-dependent flow is modeled. The equations that govern depth-averaged surface-water flow account for the effects of bed friction, wind-induced stress at the water surface, fluid stresses caused by turbulence, and the effect of the Earth's rotation. Because velocity in the vertical direction is not modeled, evaluation of phenomena such as stratified flow is beyond the scope of the modeling system. Also, because water density is assumed constant, flows resulting from horizontal density gradients cannot be evaluated.

FESWMS-2DH can be used to simulate flow in water bodies that have irregular topography and geometrical features, such as islands and highway embankments. Flow over dams, weirs, and highway embankments, and through bridges, culverts, and gated openings, also can be modeled. Boundary stresses (bed friction and surface stresses caused by wind) and stresses caused by turbulence are determined using empirical relations.

Flow through bridges and culverts can be modeled as either one-dimensional or two-dimensional flow. One-dimensional flow is described by an empirical equation that determines the flow rate through a bridge or culvert on the basis of the water-surface elevations at the upstream and downstream sides of the structure. When two-dimensional flow through a bridge is modeled, additional flow resistance that results from contact between the bridge deck and water surface is considered. Although it usually is not practical to model bridge piers directly, the effect of bridge piers can be accounted for indirectly by increasing resistance coefficients within a bridge opening.

Flow over highway embankments can be modeled as either one-dimensional or two-dimensional flow. However, for reasons that are described later, modeling flow over highway embankments as one-dimensional flow using empirical weir-flow equations is usually more accurate.

When water flows over a bridge deck and pressure flow exists within the bridge opening (combined weir and pressure flow), flow over the bridge needs to be modeled as one-dimensional weir flow. However, flow through the bridge can be modeled as either one- or two-dimensional flow.

The effect of changes to the system can be forecast by modifying the input data that describe an existing physical system. Thus, FESWMS-2DH can be used to study the consequences of designed works and operations.

Methodology

A fundamental requirement of any numerical model is a satisfactory quantitative description of the physical processes that affect the system that is being modeled. The partial differential equations that govern two-dimensional surface-water flow in a horizontal plane are derived from equations that govern three-dimensional flow by considering fluid velocity in the vertical direction to be negligible. Hence, pressure within the fluid is considered to be the same as in a hydrostatic condition.

The numerical technique used to solve the governing equations is based on the Galerkin finite element method. Application of the finite element method requires the water body being modeled to be divided into smaller regions called elements. An element can be either triangular or quadrangular in shape; shapes that can easily be easily arranged to fit complex boundaries. The elements are defined by a series of node points located at the element vertices, mid-side points, and, in the case of nine-node quadrilateral elements, at their centers. Values of dependent variables are approximated within each element using the nodal values and a set of interpolation functions (also called shape functions).

Approximations of the dependent variables are substituted into the governing equations, which generally will not be satisfied exactly, thus forming a residual. The residual is weighted over the entire solution region. The weighted residuals, which are defined by equations, are set to zero, and the resulting equations are solved for the dependent variables. In Galerkin's method, the weighting functions are chosen to be the same as those used to interpolate values of the dependent variables within each element.

The Galerkin finite element method requires the governing equations to be weighted over the entire solution domain. The

weighting process requires integration, which is performed numerically using Gaussian quadrature on a single element. Repetition of the integration for all elements that comprise a solution region produces a system of nonlinear algebraic equations when the time derivatives are discretized. Because the system of equations is nonlinear, an iterative solution procedure is needed. Newton iteration, or a variation thereof, is applied, and the resulting system of equations is solved using an efficient frontal solution scheme.

Input and Output Data

Input data can be classified broadly as one of the following categories: (1) Program control data, (2) network data, or (3) initial and boundary condition data.

Program control data govern the overall operation of a program. These data include codes that define functions to be performed by the modeling system, and constant values that are used as coefficients in equations and apply to the entire finite element network.

Network data describe the finite element network (grid). These data include element connectivity lists, element property type codes, node point coordinates, and node point ground-surface elevations. Also included as network data are sets of empirical coefficients that apply to a particular element property type.

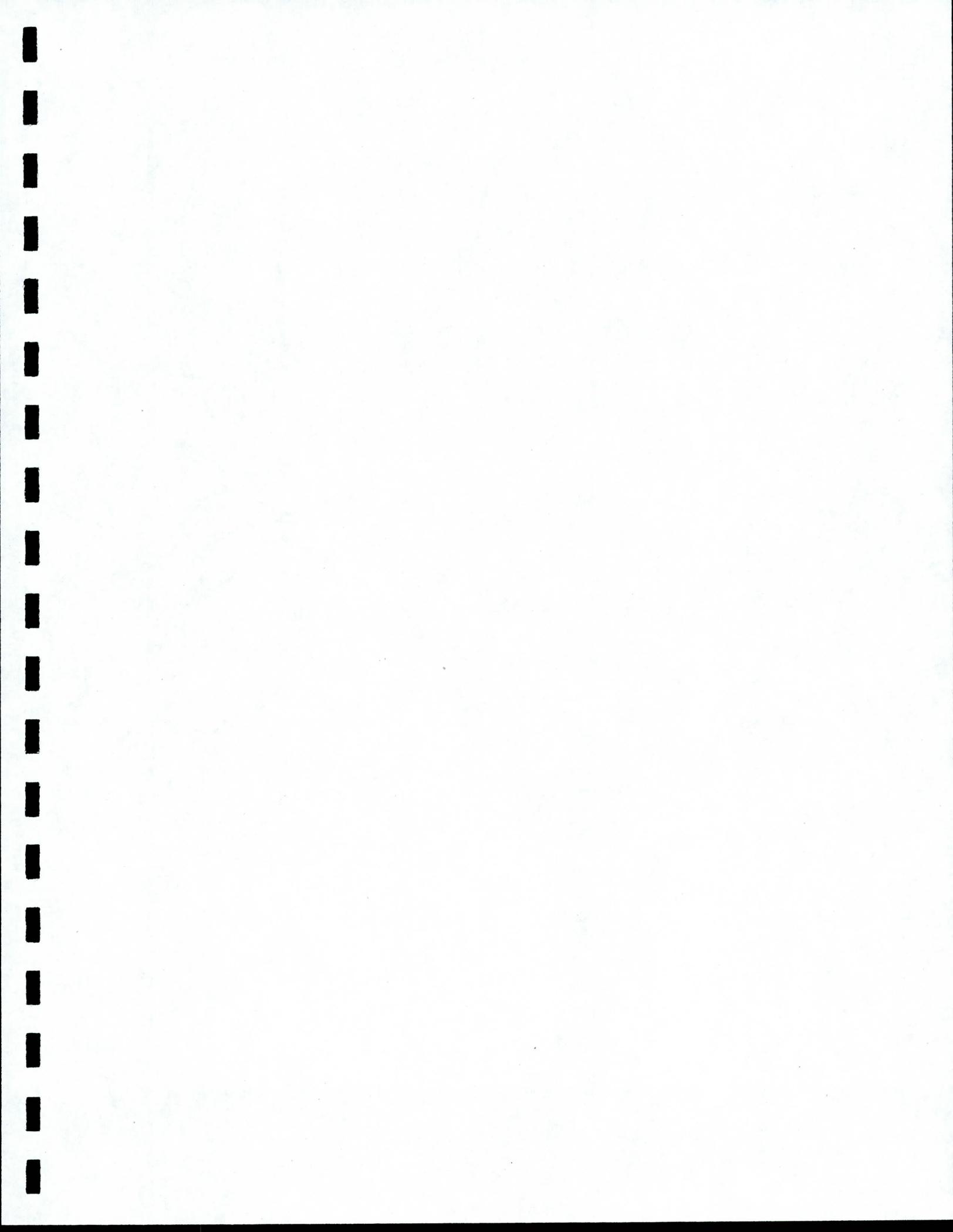
Initial condition data are starting values of the dependent variables and their time derivatives at each node point in the finite element network. Boundary condition data are values of dependent variables that are prescribed at particular node points along the boundary of the network.

Output from the modeling system consists of processed network data, computed flow data (depth-averaged velocities and

water depth at each node point, and the derivatives of these quantities with respect to time for unsteady flow simulations), and plots of both network data and flow data.

Graphic Output

For the purpose of transportation and long-term storage of graphical information, graphic output from FESWMS-2DH is written in a specified format to a data file that is called a plotfile. A plotfile can be read by a utility program that displays the graphic output on a specific hardware device. Graphic output stored in a plotfile can be processed afterward as often as necessary, stored for future use, or transported from one place to another.



This page is blank.

Section 3

BASIC CONCEPTS OF THE FINITE ELEMENT METHOD

The finite element method is a numerical procedure for solving differential equations encountered in problems of physics and engineering. Originally devised to analyze structural systems, the finite element method has developed into an effective tool for evaluating a wide variety of problems in the field of continuum mechanics. Development of the finite element method has been encouraged primarily by the continued advancement of high-speed digital computers, which provide a means of rapidly performing the many calculations that are needed to obtain a solution. Only in recent years has the finite element method been used to solve surface-water flow problems. Nevertheless, a large amount of literature on the subject has already emerged. Lee and Froehlich (1986) provide a detailed review of literature on the finite element solution of the equations of two-dimensional surface-water flow in a horizontal plane.

FESWMS-2DH uses the Galerkin finite element method to solve the governing system of differential equations. The solution begins by dividing the physical region of interest into a number of subregions, which are called elements. An element can be either triangular or quadrangular in shape, and is defined by a finite number of node points situated along its boundary or in its interior. A list of nodes connected to each element is easily recorded for identification and use. Values of a dependent variable are approximated within each element using values defined at the element's node points, and a set of interpolation (shape) functions. Mixed interpolation is used in FESWMS-2DH; that is, quadratic interpolation functions are used to interpolate depth-averaged velocities and linear functions are used to interpolate flow depth.

The method of weighted residuals is applied to the governing differential equations next to form a set of equations for each element. Approximations of the dependent variables are substituted into the governing equations, which generally are not satisfied exactly, to form residuals. The residuals are required to vanish, in an average sense, when they are multiplied by a weighting function and summed at every point in the solution domain. In Galerkin's method, the weighting functions are chosen to be the same as the interpolation functions. By requiring the summation of the weighted residuals to equal zero, the finite element equations take on an integral form. Coefficients of the equations are integrated numerically, and all the element (local) equations are assembled to obtain the complete (global) system of equations. The global set of algebraic equations is solved simultaneously.

Method of Weighted Residuals

The method of weighted residuals is a technique for approximating solutions to partial differential equations. Although the technique provides a means of forming the element equations, it is not directly related to the finite element method. Applying the method of weighted residuals involves two basic steps. The first step is to assume a general functional behavior of a dependent variable so that the governing differential equation and boundary condition equations can be satisfied approximately. Substitution of the assumed value of the dependent variable into the governing equations usually results in some error, called a residual. The residual is required to vanish, in an average sense, within the solution region. The second step of the method is to solve the residual equation for the parameters of the functional representation of the dependent variable.

To be more specific, let the differential equation for a problem be written as

$$\mathcal{L}u - f = 0 , \quad (3-1)$$

where \mathcal{L} is a differential operator, u is the dependent variable, and f is a known function. The dependent variable is assumed to be represented by \hat{u} , which is defined in terms of some unknown parameters, C , and a set of functions, N , as follows:

$$u \approx \tilde{u} = \sum_{i=1}^n N_i C_i . \quad (3-2)$$

When \tilde{u} is substituted for u in equation 3-1, it is unlikely the equation will be satisfied exactly. In fact, a trial solution is defined as

$$\mathcal{L}\tilde{u} - f = \varepsilon , \quad (3-3)$$

where ε is the residual (error) of the approximate solution. The method of weighted residuals attempts to determine the m unknown parameters, C , so that the error, ε , is as small as possible within the solution region. One way of minimizing ε is to form a weighted average of the error and to require the average to vanish when integrated with respect to the entire solution region. The weighted average is computed as

$$\int_R W_i \varepsilon \, dR = 0; \text{ for } i = 1, 2, \dots, m; \quad (3-4)$$

where R is the solution domain, and W are the m linearly independent weighting functions. After the weighting functions have been specified, a set of m simultaneous equations remain to be solved for the unknown parameters C . The second step in applying the method of weighted residuals is to solve for C , thus obtaining an approximate representation of the unknown dependent variable u , using equation 3-2.

There are several weighted residual methods that can be used. Each method is defined by the choice of weighting functions. The method used most often in finite element analysis is known as Galerkin's method. Application of Galerkin's method requires that the weighting functions be the same as those used to approximate u (that is, $W_i = N_i$, for $i = 1, 2, \dots, m$). Thus, Galerkin's method requires that

$$\int_R N_i (\mathcal{L}\tilde{u} - f) dR = 0; \text{ for } i = 1, 2, \dots, m. \quad (3-5)$$

After the approximating functions N are specified, the equations can be evaluated explicitly, and the solution found in a routine manner.

Elements and Interpolation Functions

The basic idea of the finite element method is to divide a solution region into a finite number of subregions, called elements. Within each element, it is assumed that the value of a continuous quantity can be approximated by a set of piecewise smooth functions using the values of that quantity at a finite number of points. The piecewise smooth functions are known as interpolation or shape functions, and are analogous to the functions N described in the previous section. The points at which the continuous quantity is defined are called node points, and the values of the quantity at the node points are analogous to the undetermined parameters C described in the previous section.

The approximation of a continuous quantity within an element is written as

$$\tilde{u}^{(e)} \approx \sum_{i=1}^n N_i^{(e)} u_i^{(e)}, \quad (3-6)$$

where $N_i^{(e)}$ are interpolation functions defined for an element, and $u_i^{(e)}$ are unknown values of u at the n node points in the element. Equation 3-6 applies to a single point in the solution region, or to any collection of points, such as those comprising an element. When Galerkin's method is applied, the left-hand side of equation 3-5 is computed as the sum of expressions of the form

$$\int_{R^{(e)}} N_i^{(e)} (\mathcal{L}\tilde{u}^{(e)} - f^{(e)}) dR^{(e)}; \text{ for } i = 1, 2, \dots, n, \quad (3-7)$$

where $R^{(e)}$ is an element domain, and $f^{(e)}$ is a function defined for an element.

A set of expressions like 3-7 is developed for each element that comprises a system. The element (local) expressions are assembled to form the complete set of system (global) equations. In a finite element solution, the values of a quantity at the node points are the unknowns. The behavior of the solution within the entire assemblage of elements is described by the element interpolation functions and the node point values, when they have been determined.

Before element equations can be assembled, the particular types of elements that will be used to model a region, and the associated interpolation functions, need to be specified; that is, the functions N need to be chosen. The interpolation functions also need to satisfy certain criteria so that convergence of the numerical solution to an exact solution of the governing differential equations can be achieved. Interpolation functions depend on the shape of an element and the order of approximation that is desired. Because the fundamental premise of the finite element method is that a region of arbitrary shape can be modeled accurately by an assemblage of elements, most finite element solutions use elements that are geometrically simple. The most commonly used two-dimensional elements are triangles and quadrilaterals. Although it is conceivable that

many types of functions could be used as interpolation functions, almost all finite element solutions use polynomials because of their relative simplicity.

If polynomial interpolation functions are used, linear variation of a quantity within an element can be determined by the values provided at the corners (vertices) of a triangular or quadrangular element. For quadratic variation of a quantity, additional values need to be defined along the sides, and possibly in the interior, of an element. FESWMS-2DH uses three types of two-dimensional elements: (1) 6-node triangles, (2) 8-node "serendipity" quadrilaterals, and (3) 9-node "Lagrangian" quadrilaterals. Both types of quadrilateral elements use identical linear interpolation functions, but their quadratic functions differ because of the presence of an additional node at the center of the 9-node quadrilateral element. The three types of elements used in FESWMS-2DH are illustrated in figure 3-1.

It may be desirable to model some complex geometric features using elements that have curved sides rather than straight sides. The basic idea behind development of curved-side elements is mapping (transformation) of a simple "parent" element, defined in a natural coordinate system, to the desired curved shape, defined in a global Cartesian coordinate system. Coordinate mappings for triangular and quadrangular elements are illustrated in figure 3-2. The transformation from straight to curved sides is accomplished by expressing the global coordinates (x,y) in terms of the natural coordinates (ξ,η) using interpolation functions in just the same way that a solution variable is interpolated within an element. Thus, the global coordinates are computed as

$$x = \sum_{i=1}^n N_i'(e) x_i^{(e)} \quad (3-8a)$$

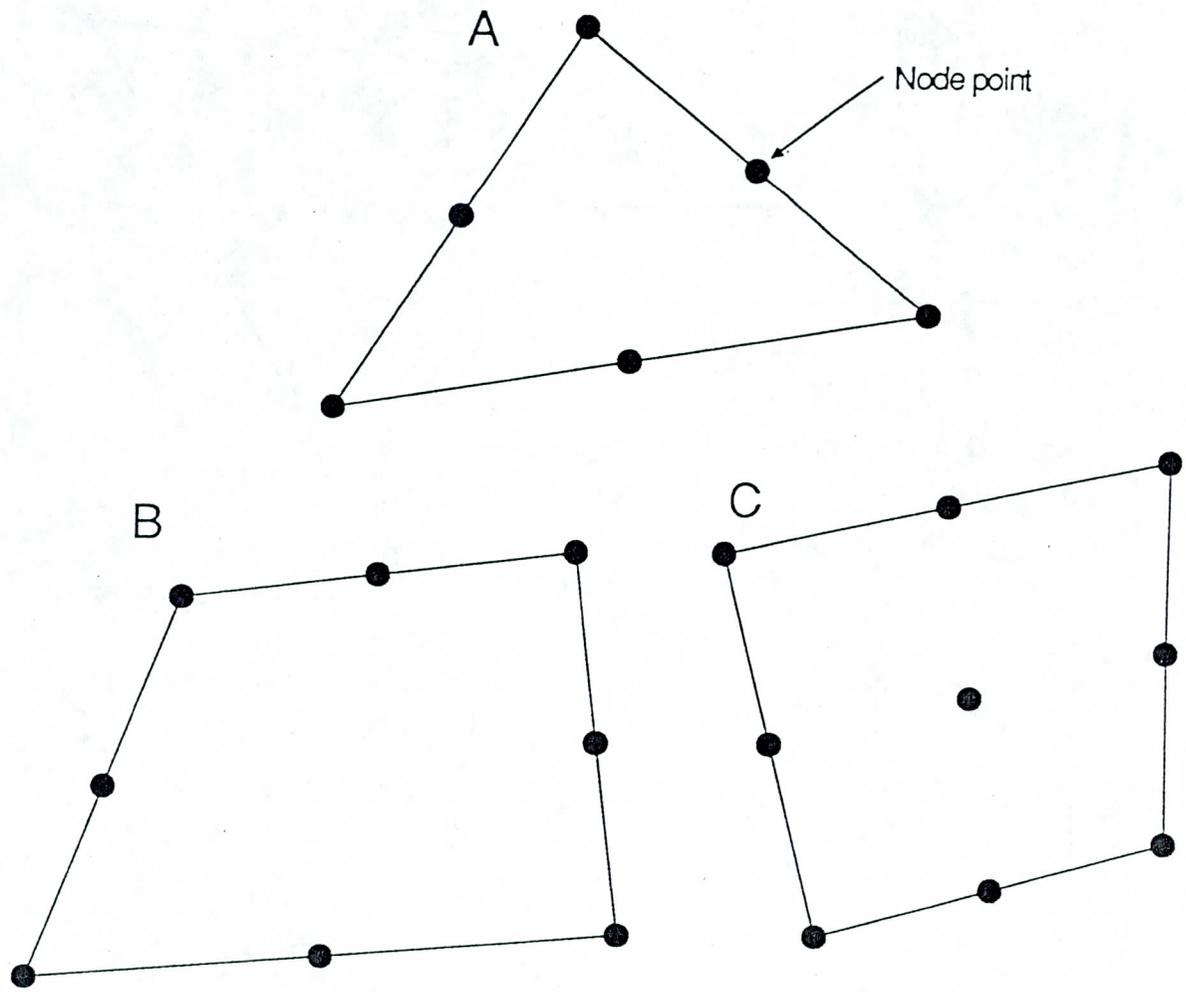
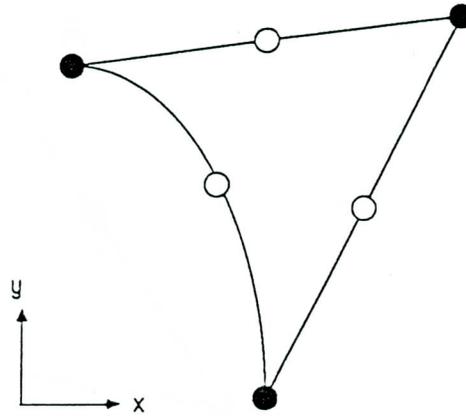
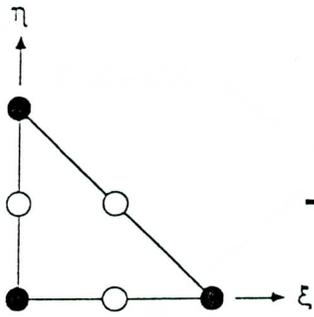


Figure 3-1. Examples of the three types of two-dimensional elements used in FESWMS-2DH: (A) A 6-node triangle; (B) an 8-node "serendipity" quadrilateral; and (C) a 9-node "Lagrangian" quadrilateral.

Parent Element in
Natural Coordinates

Global Elements in
Cartesian Coordinates



● Corner node

○ Midside node

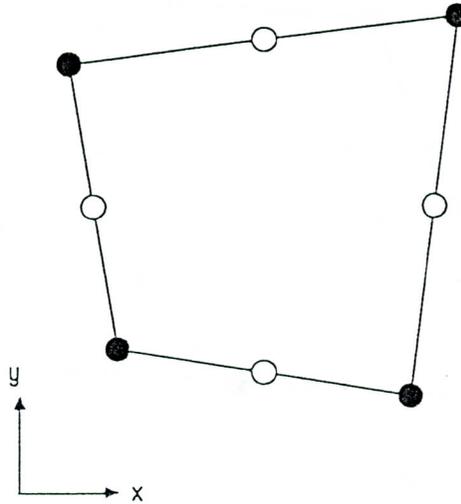
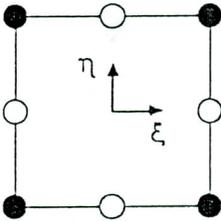


Figure 3-2. Illustration of coordinate mapping of a triangular and a quadrangular element.

and

$$y = \sum_{i=1}^n N_i'(e) y_i^{(e)}, \quad (3-8b)$$

where $N_i'(e) = N_i'(\xi, \eta)$. Quadratic interpolation functions are used in FESWMS-2DH to transform natural coordinates to global coordinates.

The natural coordinates (ξ and η) depend on the shape of an element (that is, triangular or quadrangular). The natural coordinate system and interpolation functions for parent elements of triangular and quadrangular global elements are illustrated in figures 3-3 to 3-5. Both linear and quadratic interpolation functions are given for each element shape because mixed interpolation is used to solve the governing differential equations (that is, linear functions are used to interpolate depth, and quadratic functions are used to interpolate depth-averaged velocities).

If a finite element equation contains derivatives of dependent variables with respect to the global coordinates x and y , then the derivatives of interpolation functions with respect to x and y also need to be defined because, for example,

$$\frac{\partial u}{\partial x} = \frac{\partial}{\partial x} \left(\sum N_i^{(e)} u_i^{(e)} \right) = \sum \frac{\partial N_i}{\partial x} u_i^{(e)}. \quad (3-9)$$

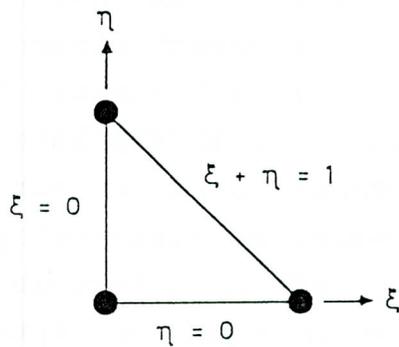
Because the interpolation functions are given in terms of natural coordinates, it is necessary to transform the derivatives with respect to natural coordinates to derivatives with respect to global coordinates. By the general rules of partial differentiation,

PARENT ELEMENT

NATURAL COORDINATE INTERPOLATION FUNCTIONS

- Corner node
- Midside node

$$\xi_o = \xi \xi_i, \quad \eta_o = \eta \eta_i$$



Linear Interpolation

Corner nodes

$$N_i = \xi_o + \eta_o + (1 - \xi - \eta)(1 - \xi_i)(1 - \eta_i)$$

Quadratic Interpolation

Corner nodes

$$N_i = \xi_o (2\xi - 1) + \eta_o (2\eta - 1) + (1 - \xi - \eta)(1 - 2\xi - 2\eta)(1 - \xi_i)(1 - \eta_i)$$

Midside nodes

$$N_i = 16\xi_o \eta_o + 8(\eta_o + \xi_o)(1 - \xi - \eta)$$

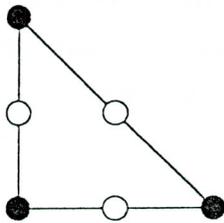


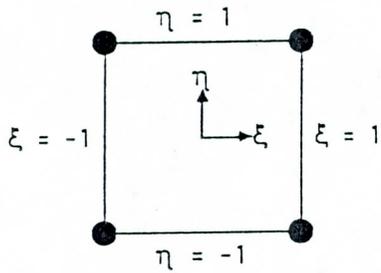
Figure 3-3. Natural coordinate system and interpolation functions for a triangular parent element.

PARENT ELEMENT

NATURAL COORDINATE INTERPOLATION FUNCTIONS

- Corner node
- Midside node

$$\xi_o = \xi \xi_i, \quad \eta_o = \eta \eta_i$$



Linear Interpolation

Corner nodes

$$N_i = 0.25(1 + \xi_o)(1 + \eta_o)$$

Quadratic Interpolation

Corner nodes

$$N_i = 0.25(1 + \xi_o)(1 + \eta_o)(\xi_o + \eta_o - 1)$$

Midside nodes

$$N_i = 0.5(1 - \xi^2)(1 + \eta_o); \quad \xi_i = 0$$

$$N_i = 0.5(1 - \eta^2)(1 + \xi_o); \quad \eta_i = 0$$

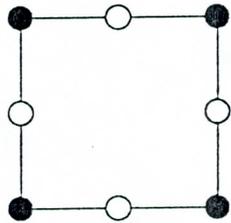


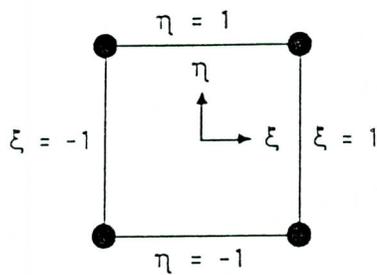
Figure 3-4. Natural coordinate system and interpolation functions for a "serendipity" quadrangular parent element.

PARENT ELEMENT

NATURAL COORDINATE INTERPOLATION FUNCTIONS

- Corner node
- Midside node
- ⊙ Center node

$$\xi_o = \xi \xi_i, \quad \eta_o = \eta \eta_i$$



Linear Interpolation

Corner nodes

$$N_i = 0.25(1 + \xi_o)(1 + \eta_o)$$

Quadratic Interpolation

Corner nodes

$$N_i = 0.25 \xi_o \eta_o (1 + \xi_o)(1 + \eta_o)(\xi_o + \eta_o - 1)$$

Midside nodes

$$N_i = 0.5 \eta_o (1 - \xi_o^2)(1 + \eta_o); \quad \xi_i = 0$$

$$N_i = 0.5 \xi_o (1 - \eta_o^2)(1 + \xi_o); \quad \eta_i = 0$$

Center node

$$N_i = (1 - \xi_o^2)(1 - \eta_o^2)$$

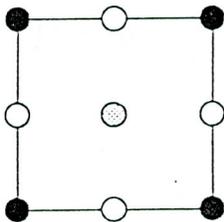


Figure 3-5. Natural coordinate system and interpolation functions for a "Lagrangian" quadrangular parent element.

$$\frac{\partial N_i}{\partial x} = \frac{\partial N_i}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial N_i}{\partial \eta} \frac{\partial \eta}{\partial x} \quad (3-10a)$$

and

$$\frac{\partial N_i}{\partial y} = \frac{\partial N_i}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial N_i}{\partial \eta} \frac{\partial \eta}{\partial y} , \quad (3-10b)$$

where the superscript (e) has been dropped for convenience. However, ξ and η usually cannot be expressed explicitly in terms of x and y . It is first necessary to consider N_i to be a function of x and y . Writing the derivatives of N_i with respect to ξ and η yields, in matrix form,

$$\begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{Bmatrix} , \quad (3-11)$$

where

$$[J] \equiv \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix}$$

is known as the Jacobian matrix. Using equation 3-8, the Jacobian matrix can be computed explicitly in terms of the natural coordinates as

$$[J] = \begin{bmatrix} \sum_{i=1}^n \frac{\partial N_i'}{\partial \xi} x_i & \sum_{i=1}^n \frac{\partial N_i'}{\partial \xi} y_i \\ \sum_{i=1}^n \frac{\partial N_i'}{\partial \eta} x_i & \sum_{i=1}^n \frac{\partial N_i'}{\partial \eta} y_i \end{bmatrix} \quad (3-12)$$

where N_i is the function that defines the coordinate transformation. The global derivatives are then computed as

$$\begin{pmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{pmatrix} = [J]^{-1} \begin{pmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{pmatrix}, \quad (3-13)$$

or

$$\frac{\partial N_i}{\partial x} = |J|^{-1} \left(\frac{\partial y}{\partial \eta} \frac{\partial N_i}{\partial \xi} - \frac{\partial y}{\partial \xi} \frac{\partial N_i}{\partial \eta} \right) \quad (3-14a)$$

and

$$\frac{\partial N_i}{\partial y} = |J|^{-1} \left(\frac{\partial x}{\partial \xi} \frac{\partial N_i}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial N_i}{\partial \xi} \right), \quad (3-14b)$$

where

$$|J| = \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi} \quad (3-15)$$

is the determinant of $[J]$. The operations indicated in equations 3-13 and 3-14 depend on the existence of $[J]$ everywhere in each element. In addition, the coordinate mapping provided by equation 3-8 is one-to-one only if $|J|$ does not vanish within an element.

The area of an element also needs to be expressed in terms of the natural coordinates ξ and η . It can be shown (Sokolnikoff and Redheffer, 1966, p. 355) that

$$dx dy = |J| d\xi d\eta. \quad (3-16)$$

Using equation 3-16, a function can be integrated numerically with respect to the area of a two-dimensional triangular or quadrangular element that has straight or curved sides.

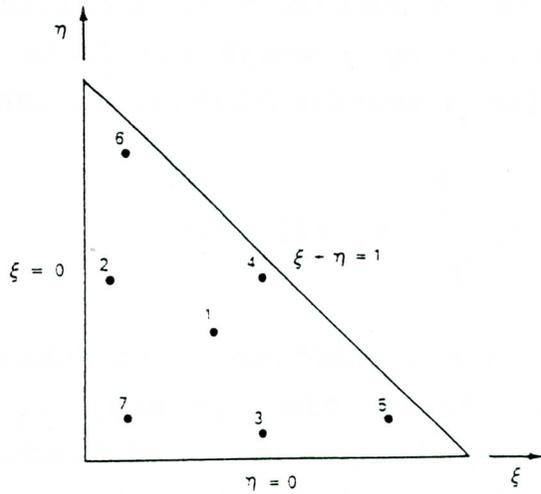
Numerical Integration

Numerical integration is used to evaluate the integrals that appear in the finite element equations. To integrate numerically, the function being integrated is evaluated at specific locations within an element, multiplied by a weighting factor, then summed. The summation process for a two-dimensional element is

$$\iint_{A_e} f(\xi, \eta) \, d\xi \, d\eta \approx A_e \sum_{i=1}^k w_i f(\xi_i, \eta_i) , \quad (3-17)$$

where A_e is the element area; f is the function being integrated; k is the number of numerical integration points; w_i is a weighting factor for the i th integration point; and ξ and η are natural coordinates of the i th integration point. The natural coordinates, ξ_i and η_i , are invariant with respect to the shape of the element in the global coordinate system.

A numerical integration scheme needs to be of sufficient accuracy to assure convergence of a finite element solution. Strang and Fix (1973) suggest that convergence will occur if a numerical integration scheme is accurate enough to compute exactly the area of an element. An exact integration of element area requires that a formula that provides at least third-order accuracy be used to integrate curve-sided quadratic elements. At least 2 by 2 Gaussian integration is needed for parabolic quadrilaterals. Parabolic triangles require at least a 3-point formula for third-order accuracy. However, it has been found that although an exact integration of element area may guarantee convergence as the size of an element approaches zero, an integration formula that has greater accuracy may be needed to integrate accurately some terms in an equation. For this reason, numerical integration formulas that provide sixth-order accuracy are used. The locations of numerical integration points and the associated weighting factors are shown in figure 3-6.

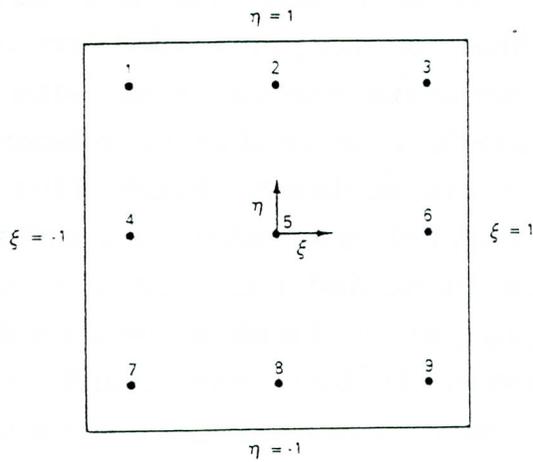


Point No.	Coordinates ξ_i, η_i	Weight w_i
1	$\frac{1}{3}, \frac{1}{3}$	0.22500000
2	α_1, β_1	0.13239415
3	β_1, α_1	
4	β_1, β_1	
5	α_2, β_2	0.12593918
6	β_2, α_2	
7	β_2, β_2	

where $\alpha_1 = 0.05971587$
 $\beta_1 = 0.47014206$
 $\alpha_2 = 0.79742698$
 $\beta_2 = 0.10123551$

EXPLANATION

• INTEGRATION POINT



Point No.	Coordinates ξ_i, η_i	Weight w_i
1	$-\alpha, \alpha$	Y_1
2	$0, \alpha$	Y_2
3	α, α	Y_1
4	$-\alpha, 0$	Y_2
5	$0, 0$	0.19753086
6	$\alpha, 0$	Y_2
7	$-\alpha, -\alpha$	Y_1
8	$0, -\alpha$	Y_2
9	$\alpha, -\alpha$	Y_1

where $\alpha = 0.77459667$
 $Y_1 = 0.07716049$
 $Y_2 = 0.12345679$

Figure 3-6. Numerical integration point coordinates and weighting factors for triangular and quadrangular parent elements.

Section 4

GOVERNING EQUATIONS

The equations that govern the hydrodynamic behavior of an incompressible fluid are based on the classical concepts of conservation of mass and momentum. For many practical surface-water flow applications, knowledge of the full three-dimensional flow structure is not required, and it is sufficient to use mean-flow quantities in two perpendicular horizontal directions. Equations that describe depth-averaged two-dimensional flow are presented in this section. Additional equations that are used to model special cases of one-dimensional flow through bridges and culverts and one-dimensional flow over weirs and highway embankments are described. Initial and boundary conditions needed to solve the set of governing equations are also discussed.

Depth-Averaged Flow Equations

The depth-averaged velocity components in the horizontal x and y coordinate directions, respectively, are defined as

$$U = \frac{1}{H} \int_{z_b}^{z_s} u \, dz \quad (4-1a)$$

and

$$V = \frac{1}{H} \int_{z_b}^{z_s} v \, dz \quad (4-1b)$$

where H is the water depth; z is the vertical direction; z_b is the bed elevation; $z_s = z_b + H$ is the water-surface elevation; u is the horizontal velocity in the x direction at a point along the vertical coordinate; and v is the horizontal velocity in the y direction at a point along the vertical coordinate. The coord-

inate system and variables used are illustrated in figure 4-1. Depth-averaged velocity is illustrated in figure 4-2. The depth-averaged surface-water flow equations are derived by integrating the three-dimensional mass and momentum balance equations with respect to the vertical coordinate from the bed to the water surface, assuming that vertical velocities and accelerations are negligible (see Jansen and others, 1979, p. 41 for a thorough derivation). The vertically-integrated momentum equations are

$$\begin{aligned} \frac{\partial}{\partial t}(HU) + \frac{\partial}{\partial x}(\beta_{uu}HUU + \frac{1}{2}gH^2) + \frac{\partial}{\partial y}(\beta_{uv}HUV) + gH\frac{\partial z_b}{\partial x} - \Omega HV \\ + \frac{1}{\rho} \left[\tau_x^b - \tau_x^s - \frac{\partial}{\partial x}(H\tau_{xx}) - \frac{\partial}{\partial y}(H\tau_{xy}) \right] = 0 \end{aligned} \quad (4-2)$$

for flow in the x direction, and

$$\begin{aligned} \frac{\partial}{\partial t}(HV) + \frac{\partial}{\partial x}(\beta_{vu}HVU) + \frac{\partial}{\partial y}(\beta_{vv}HVV + \frac{1}{2}gH^2) + gH\frac{\partial z_b}{\partial y} + \Omega HU \\ + \frac{1}{\rho} \left[\tau_y^b - \tau_y^s - \frac{\partial}{\partial x}(H\tau_{yx}) - \frac{\partial}{\partial y}(H\tau_{yy}) \right] = 0 \end{aligned} \quad (4-3)$$

for flow in the in the y direction, where β_{uu} , β_{uv} , β_{vu} , and β_{vv} are momentum correction coefficients that account for the variation of velocity in the vertical direction; g is gravitational acceleration; Ω is the Coriolis parameter; ρ is the density of water, which is considered to be constant; τ_x^b and τ_y^b are bed shear stresses acting in the x and y directions, respectively; τ_x^s and τ_y^s are surface shear stresses acting in the x and y directions, respectively; and τ_{xx} , τ_{xy} , τ_{yx} , and τ_{yy} are shear stresses caused by turbulence where, for example, τ_{xy} is the shear stress acting in the x direction on a plane that is perpendicular to the y direction. The vertically-integrated mass balance equation (that is, the continuity equation) is

$$\frac{\partial H}{\partial t} + \frac{\partial}{\partial x}(HU) + \frac{\partial}{\partial y}(HV) = q, \quad (4-4)$$

where q is a unit source (inflow) or a unit sink (outflow) term.

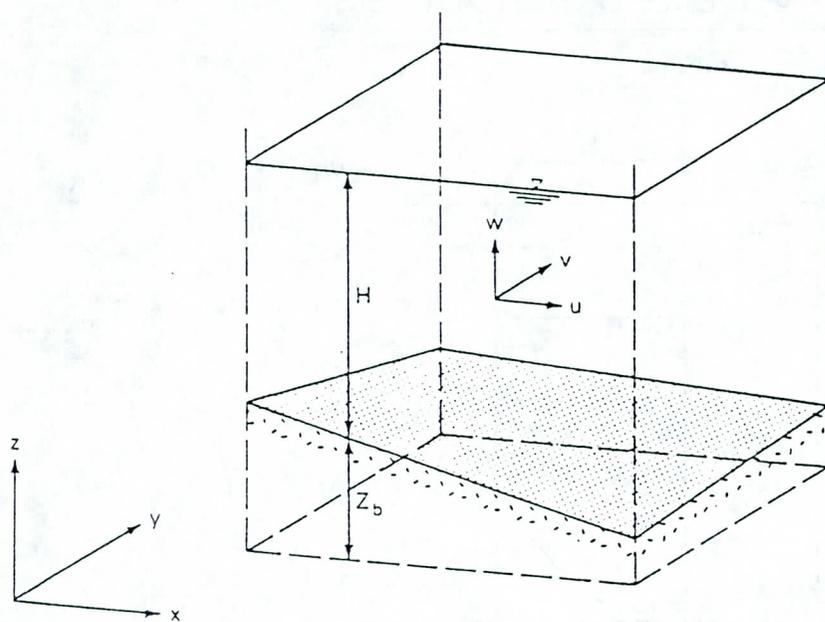


Figure 4-1. Illustration of the coordinate system and variables used to derive the depth-averaged surface-water flow equations.

Depth-Averaged Velocities

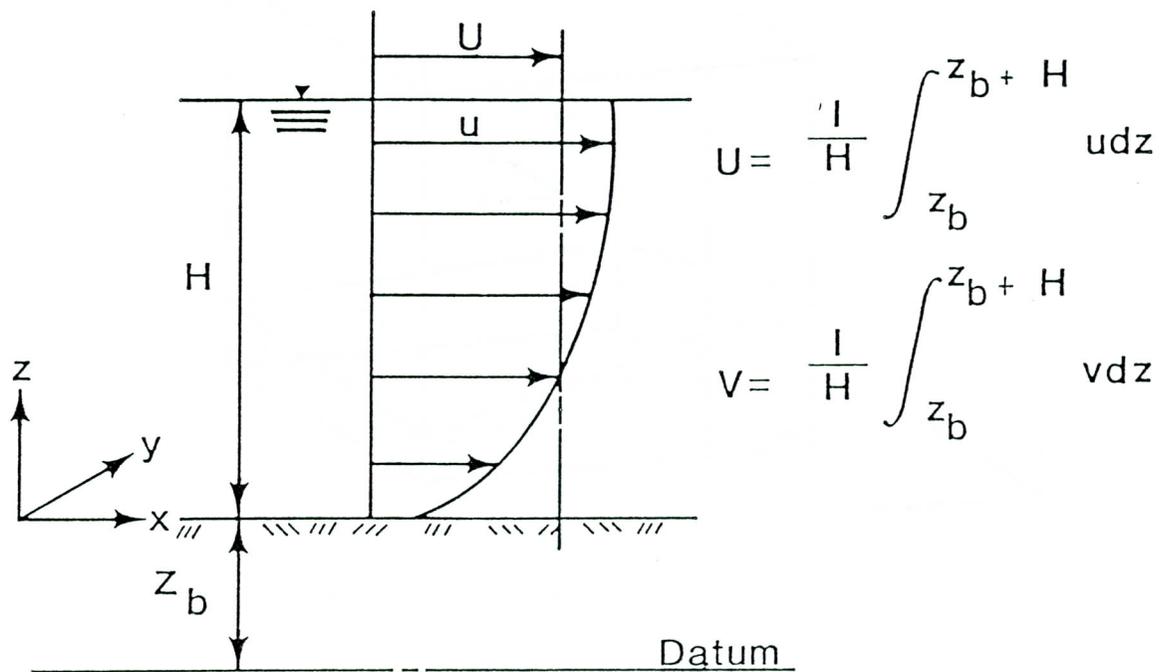


Figure 4-2. Illustration of depth-averaged velocity.

Momentum Correction Coefficients

The momentum correction coefficients β_{uu} , β_{uv} , β_{vu} , and β_{vv} result from the vertical integration of the momentum balance equations and account for vertical variations of u and v . The momentum correction coefficients are used to multiply the advective momentum flux terms in equations 4-2 and 4-3, and are computed as

$$\beta_{uu} = \frac{1}{HUU} \int_{z_b}^{z_s} uu \, dz , \quad (4-5a)$$

$$\beta_{uv} = \beta_{vu} = \frac{1}{HUV} \int_{z_b}^{z_s} uv \, dz , \quad (4-5b)$$

and

$$\beta_{vv} = \frac{1}{HVV} \int_{z_b}^{z_s} vv \, dz . \quad (4-5c)$$

The momentum correction coefficients depend on the vertical velocity distribution, and often are assumed to equal unity (that is, a uniform vertical velocity distribution is assumed). If the velocity in the vertical direction can be computed as

$$u = \frac{U_*}{\kappa} \log \left(\frac{z - z_b}{k} \right) , \quad (4-6)$$

where $U_* = \sqrt{c_f} U$ is bed shear velocity; c_f is a dimensionless bed shear-stress coefficient (to be discussed later); κ is von Karman's constant; and k is a constant that has a dimension of length; then the resulting momentum correction coefficients are all equal and are computed as

$$\beta = 1 + \frac{c_f}{\kappa^2} . \quad (4-7)$$

The momentum correction coefficient in FLOMOD is computed using the expression

$$\beta = \beta_0 + c_\beta c_f . \quad (4-8)$$

Equations 4-7 and 4-8 are equivalent when $\beta_0 = 1.0$ and $c_\beta = 1/\kappa^2$. The coefficient κ has been found to equal approximately 0.4, from which c_β equals 6.25. A constant momentum correction coefficient can be specified by setting β_0 equal to the desired value, and setting c_β equal to zero. The default values in FLOMOD for β_0 and c_β are 1.0 and 0.0, respectively. Acceptance of these default values means that vertical variations in velocity are considered to be negligible.

Coriolis Parameter

The Coriolis parameter, Ω , is equal to $2\omega \sin \phi$, where ω is the angular velocity of the rotating Earth (7.27×10^{-5} radians per second), and ϕ is the mean angle of latitude of the area being modeled. The terms in the momentum equations that contain Ω account for the effect of the Earth's rotation on water movement. The sign of ϕ is positive in the northern hemisphere and negative in the southern hemisphere. A constant value of the Coriolis parameter is used in FLOMOD (that is, the variation of Ω within the area covered by a finite element network is considered to be negligible). For most shallow flows where the width to depth ratio is large (for example, flows in rivers and flood plains), the Coriolis effect will be small and can be safely ignored.

Bed Shear Stresses

The directional components of the bed shear stress are computed as

$$\tau_x^b = \rho c_f m_b U \sqrt{U^2 + V^2} \quad (4-9a)$$

and

$$\tau_y^b = \rho c_f m_b U \sqrt{U^2 + V^2} \quad (4-9b)$$

where c_f is a dimensionless bed-friction coefficient, and

$$m_b = \sqrt{1 + \left(\frac{\partial z_b}{\partial x}\right)^2 + \left(\frac{\partial z_b}{\partial y}\right)^2} \quad (4-10)$$

is a coefficient that accounts for increased shear stress caused by a sloping bed.

The bed friction coefficient c_f can be computed either as

$$c_f = \frac{g}{C^2} \quad (4-11)$$

or

$$c_f = \frac{gn^2}{\phi H^{1/3}} \quad (4-12)$$

where C is the Chézy discharge coefficient; n is the Manning roughness coefficient; and ϕ is a factor that equals 2.208 when U.S. Customary units are used, or 1.0 when S.I. units are used.

FLOMOD allows Manning roughness coefficients to be varied as a function of flow depth. Vertical variation of roughness coefficients can be used to model flow through areas where the

surface roughness either increases or decreases with the depth of flow, depending on the ground cover and the type and density of vegetation. FLOMOD does not allow Chézy coefficients to be specified as a function of flow depth.

Values of Chézy discharge coefficients and Manning roughness coefficients for natural and man-made channels, as well as flood plains, can be estimated using references such as Chow (1959), Barnes (1967), and Arcement and Schneider (1984). However, coefficients in these references have been determined on the basis of assumed one-dimensional flow, and implicitly account for the effects of turbulence and deviation from a uniform velocity in a cross section. Because the depth-averaged flow equations directly account for horizontal variations of velocity and the effect of turbulence, values of c_f computed using coefficients based on a one-dimensional flow assumption may be slightly greater than necessary. Little information is available to help select coefficients for two-dimensional depth-averaged flow computations. For the time being, it is suggested that Chézy or Manning coefficients be estimated on the basis of available references and experience.

Surface Shear Stresses

The directional components of surface shear stress caused by wind are computed as

$$\tau_x^S = c_s \rho_a W^2 \quad (4-13a)$$

and

$$\tau_y^S = c_s \rho_a W^2 \sin \psi \quad (4-13b)$$

where c_s is a dimensionless surface stress coefficient; ρ_a is the density of air; W is a characteristic wind velocity near the

water surface; and ψ is the angle between the wind direction and the positive x-axis.

The surface stress coefficient has been found to be a function of wind speed, and is computed as

$$c_s = \begin{cases} c_{s1} \times 10^{-3}; & \text{if } W \leq W_{\min} \\ [c_{s1} + c_{s2}(W - W_{\min})] \times 10^{-3}; & \text{if } W > W_{\min} \end{cases} \quad (4-14)$$

For wind speed in meters per second, measured 10 meters above the water surface, Garratt (1977) reports that $c_{s1} = 1.0$, $c_{s2} = .067$, and $W_{\min} = 4$ m/s. Wang and Connor (1975) compare several relations for c_s and conclude that $c_{s1} = 1.1$, $c_{s2} = 0.0536$, and $W_{\min} = 0$ m/s. Hicks (1972) reports that $c_{s1} = 1.0$, $c_{s2} = 0.05$, and $W_{\min} = 5.0$ m/s.

Factors other than wind velocity can influence the value of the surface stress coefficient c_s . For example, Hicks and others (1974) show that as water becomes shallow (less than 2.5 m deep) long period waves are not able to develop fully. As a result, the water surface will be smoother and the value of c_s remains close to 1.0×10^{-3} for all wind speeds.

Equation 4-15 is used in FLOMOD to compute the surface stress coefficient. The coefficients c_{s1} and c_{s2} , and the minimum wind velocity W_{\min} , can be specified. Default values of c_{s1} and c_{s2} are 1.0 and 0.0, respectively. The default value of W_{\min} is 0.0 m/s.

Stresses Caused by Turbulence

The depth-averaged stresses caused by turbulence are computed using Boussinesq's eddy viscosity concept whereby the turbulent stresses, like viscous stresses, are assumed to be proportional to gradients of the depth-averaged velocities. The

turbulent stresses are computed as follows:

$$\tau_{xx} = \rho \tilde{\nu}_{xx} \left(\frac{\partial U}{\partial x} + \frac{\partial U}{\partial x} \right) \quad (4-15a)$$

$$\tau_{xy} = \tau_{yx} = \rho \tilde{\nu}_{xy} \left(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x} \right) , \quad (4-15b)$$

$$\tau_{yy} = \rho \tilde{\nu}_{yy} \left(\frac{\partial V}{\partial y} + \frac{\partial V}{\partial y} \right) , \quad (4-15c)$$

where $\tilde{\nu}_{xx}$, $\tilde{\nu}_{xy}$, $\tilde{\nu}_{yx}$, and $\tilde{\nu}_{yy}$ are directional values of the depth-averaged kinematic eddy viscosity or turbulent exchange coefficient. In FLOMOD, the depth-averaged kinematic eddy viscosity is considered to be isotropic (that is, $\tilde{\nu}_{xx} = \tilde{\nu}_{xy} = \tilde{\nu}_{yx} = \tilde{\nu}_{yy}$), and is denoted by $\tilde{\nu}$.

Eddy viscosity is related to eddy diffusivity for heat or mass transfer, $\tilde{\Gamma}$, as

$$\tilde{\Gamma} = \frac{\tilde{\nu}}{\sigma_t} , \quad (4-16)$$

where σ_t is an empirical constant called the Prandtl number (for diffusion of heat) or Schmidt number (for diffusion of mass). Many experiments on spreading of dye in open channels (Fischer and others, 1979) have shown that values of dimensionless diffusivity, $e_* = \tilde{\Gamma}/U_*H$, usually are between 0.1 and 0.2 in straight uniform open channels, and that channel curves and sidewall irregularities increase e_* . Values of e_* in natural streams hardly ever are less than 0.4. The turbulent Prandtl/Schmidt number has been found in heat and mass transfer experiments to vary from 0.5, in free shear flows, to 0.9, in flow regions near walls (Rodi, 1982, p. 587). Assuming that the turbulent exchange of mass and momentum are similar (that is, $\sigma_t = 1.0$), eddy viscosity in natural open channels can be related to the bed shear velocity and depth by

$$\tilde{\nu} = (0.6 \pm 0.3) U_*H , \quad (4-17)$$

where larger values are likely to occur if a channel has sharp curves or rapid changes in geometry.

Eddy viscosity is computed using the formula

$$\tilde{\nu} = \tilde{\nu}_0 + c_\mu U_* H, \quad (4-18)$$

where $\tilde{\nu}_0$ is a base kinematic eddy viscosity, and c_μ is a dimensionless coefficient. Comparing equations 4-17 and 4-18, an approximate value for c in natural channels is 0.6. A constant eddy viscosity is assigned by specifying $c_\mu = 0.0$ and $\tilde{\nu}_0 > 0$.

Weir Flow and Roadway Overtopping

The depth-averaged flow equations were derived by assuming that velocity in the vertical direction is negligible. However, flow over weirs, or weir-like structures such as roadway embankments, can have a significant velocity component in the vertical direction and cannot always be simulated accurately using the depth-averaged flow equations. Flow over weirs (roadway embankments) is modeled more accurately using an empirical equation to calculate discharge over a horizontal weir.

One-dimensional flow over a weir or roadway embankment is modeled by dividing the weir or roadway into sections that are called weir segments. Each weir segment is described by either one or two *boundary nodes*, a discharge coefficient, and the length and crest elevation of the segment. Two boundary nodes are needed, one on each side of the segment, if the areas on both the upstream and downstream sides of a weir segment are included in the finite element network. Water that flows over a weir segment defined by two nodes is considered to leave the network at the upstream node (the node with the highest water surface) and to enter the network at the downstream node. If the area on only one side of a weir segment is included in a network, only one boundary node is needed for each weir segment. Water flowing

over a weir segment defined by only one node is considered to leave the network at the node and not return to the network.

Flow over a weir segment, Q_w , is computed as

$$Q_w = K_w (z_e^h - z_c)^{3/2}, \quad (4-19)$$

where K_w is a weir coefficient; z_e^h is the elevation of the energy head at the upstream node; and z_c is the crest elevation of the weir segment. The weir coefficient, K_w , is computed as

$$K_w = C_{sub} C_w L_w \sqrt{g}, \quad (4-20)$$

where C_{sub} is a coefficient that adjusts K_w for submergence of a weir segment by tailwater; C_w is a dimensionless discharge coefficient for free (that is, not submerged) weir flow (usually about 0.53); and L_w is the length of the weir segment. The submergence coefficient, C_{sub} , is determined automatically using a relation taken from Bradley (1978) which is presented in table 4-1. If only one boundary node is used to define a weir segment, free flow is assumed (that is, C_{sub} is set equal to 1.0).

Bridge and Culvert Flow

Flow through bridges and culverts can be modeled as either one-dimensional or two-dimensional flow. If the width of a bridge or culvert is small in relation to the width of the channel or flood plain on which it is located, it is probably best to model the flow as one-dimensional flow. If the width of a bridge or culvert is large in relation to the width of the channel or flood plain, two-dimensional flow probably needs to be modeled.

Table 4-1. Submergence factor, C_{sub} , for weir flow over a roadway embankment for various ratios of submergence.

Submergence ratio ^a	Submergence factor, C_{sub}
less than or equal to 0.75	1.000
0.80	0.995
0.84	0.987
0.86	0.975
0.88	0.960
0.90	0.930
0.92	0.885
0.94	0.885
0.96	0.710
0.98	0.575
0.99	0.450
1.00	0.000

^aSubmergence ratio = $(z_s^t - z_c) / (z_e^h - z_c)$, where z_s^t is the water-surface elevation at the downstream side of the roadway; z_c is the roadway crest elevation; and z_e^h is the energy head elevation at the upstream side of the roadway.

One-Dimensional Bridge/Culvert Flow

One-dimensional flow through a small bridge or a culvert is calculated using an equation developed for flow through culverts. Each bridge/culvert is described by either one or two boundary nodes, a discharge coefficient, and the physical characteristics of the bridge or culvert. If the areas at both ends of a bridge/culvert are included in a finite element network, two boundary nodes are needed. Water flowing through a one-dimensional bridge or culvert is considered to leave the network at the upstream node (the node with the highest water surface) and to enter the network at the downstream node. If the area on only one end of a bridge/culvert is included in a network, only one boundary node is needed for the culvert. Water flowing

through the culvert is considered to leave the network at the node and not return to the network.

Flow through a culvert is calculated as either type 4 flow or type 5 flow as described by Bodhaine (1968). For type 4 flow (fully submerged), a culvert is submerged by both headwater and tailwater. For type 5 flow (inlet control), the top edge of a culvert entrance contracts the flow in a manner similar to a sluice gate, and the culvert barrel flows partly full, at a depth less than critical depth, and culvert flow rate is computed as

$$Q_C = \begin{cases} K_C \sqrt{z_S^h - z_S^t} & ; \text{ Type 4 flow} \\ K_C \sqrt{z_S^h - z_{inv}} & ; \text{ Type 5 flow} \end{cases} \quad (4-21)$$

where K_C is a coefficient that depends on the type of flow in the culvert; z_S^h is the water-surface elevation at the upstream end of a culvert (that is, the headwater elevation); z_S^t is the water-surface elevation at the downstream end of the culvert (that is, tailwater elevation); and z_{inv} is the invert elevation at the culvert entrance. For type 4 flow, the culvert coefficient is computed as

$$K_C = C_C A_C \sqrt{\frac{2g}{1 + \frac{29 C_C^2 n_C^2 L_C}{R_C^{4/3}}}} \quad (4-22)$$

where C_C is a dimensionless discharge coefficient; A_C is the cross section area of the culvert; n_C is the Manning roughness coefficient of the culvert barrel; L_C is the length of the culvert barrel; and R_C is the hydraulic radius of the culvert barrel flowing full. For type 5 flow, the culvert coefficient is computed as

$$K_C = C_C A_C \sqrt{2g} \quad (4-23)$$

Type 4 flow discharge coefficients

The discharge coefficients, C_c , for type 4 flow conditions that are described in the following subsections are taken from Bodhaine (1968).

Flush setting in vertical headwall.—Discharge coefficients for box or pipe culverts placed flush in a vertical headwall are presented in table 4-2. The coefficients in table 4-2 apply to square-end pipe or box culverts, corrugated-metal culverts, concrete-pipe culverts that have beveled or bell-mouthed ends, and box culverts that have rounded or beveled sides. The discharge coefficient for pipe culverts with flared ends is 0.90 for all culvert diameters.

Table 4-2. Type 4 flow discharge coefficient, C_c , for box or pipe culverts placed flush in a vertical headwall for various ratios of entrance rounding or beveling.

Entrance rounding or beveling ratio ^a r/b, w/b, r/D, or w/D	Discharge coefficient, C_c
0.00	0.84
0.02	0.88
0.04	0.91
0.06	0.94
0.08	0.96
0.10	0.97
0.12	0.98

^ar = radius of entrance rounding;
b = width of box culvert;
w = length of a chamfer;
D = minimum inside diameter of pipe culvert.

Wingwall Entrance.—The addition of wingwalls to the entrance of a pipe culvert placed flush in a vertical headwall does not affect discharge, so coefficients given in table 4-2 also apply to pipe culverts that have wingwalls. Discharge coefficients for box culverts that have wingwalls are given in table 4-3. If a box culvert has a wingwall angle equal to 90

degrees, and a rounded or beveled entrance, the discharge coefficient needs to be adjusted. Adjustment coefficients k_r and k_w that account for rounded and beveled entrance edges, respectively, are given in tables 4-4 and 4-5, respectively.

Table 4-3. Type 4 flow discharge coefficient, C_c , for box culverts that have wingwalls and a square, rounded, or beveled entrance.

Wingwall angle in degrees	Discharge coefficient for entrance that is	
	Square	Rounded or beveled
30 to 75	0.87	Value from table 4-1 but no less than 0.87
90	0.75	$0.75 k_r$ or $0.75 k_w$

Table 4-4. Discharge coefficient adjustment factor, k_r , that accounts for entrance rounding of pipe or box culverts placed flush in a vertical headwall.

Entrance rounding ratio r/b or r/D^a	Discharge coefficient adjustment factor, k_r
0.00	1.000
0.02	1.042
0.04	1.082
0.06	1.120
0.08	1.155
0.10	1.180
0.12	1.195
greater than or equal to 0.14	1.200

^a r = radius of entrance rounding;
 b = width of box culvert;
 D = minimum inside diameter of pipe culvert.

Table 4-5. Discharge coefficient adjustment factor, k_w , that accounts for entrance beveling of pipe or box culverts placed flush in a vertical headwall.

Entrance beveling ratio w/b or w/D ^a	Discharge coefficient adjustment factor k_w for a bevel angle, in degrees, of		
	30	45	60
0.00	1.000	1.000	1.000
.01	1.014	1.033	1.045
.02	1.027	1.063	1.088
.03	1.039	1.087	1.128
.04	1.500	1.107	1.162
.05	1.060	1.123	1.194
.06	1.068	1.135	1.220
.08	1.080	1.150	1.260
.10	1.088	1.150	1.280

^ar = radius of entrance rounding;
w = length of a chamfer;
D = minimum inside diameter of pipe culvert.

Projecting Entrance.—The discharge coefficient for corrugated-metal pipe and pipe-arch culverts that extend past a headwall or embankment is computed by multiplying the appropriate coefficient from table 4-2 by the adjustment factor k_L given in table 4-6. The discharge coefficient for concrete-pipe culverts that have a beveled entrance and that have a projecting entrance are the same as those given in table 4-2.

Mitered pipe set flush with sloping embankment.—The discharge coefficient for pipes mitered and placed flush with a sloping embankment is 0.74. For corrugated-metal pipe culverts and pipe-arch culverts that project beyond an embankment, the base coefficient 0.74 is multiplied by the adjustment factor k_L given in table 4-6.

Type 5 flow discharge coefficients

The discharge coefficients, C_c , for type 5 culvert flow conditions described in the following subsections also are taken from Bodhaine (1968).

Table 4-6. Discharge coefficient adjustment factor, k_L , for pipe and pipe-arch culverts that extend beyond a headwall or embankment.

Value of L_p/D^a	Adjustment factor, k_L	Value of L_p/D	Adjustment factor, k_L
0.00	1.00	0.00	1.00
.01	.99	.1	.92
.02	.98	.2	.92
.03	.98	.3	.92
.04	.97	.4	.91
.05	.96	.5	.91
.06	.95	.6	.91
.07	.94	.7	.91
.08	.94	.8	.90
.09	.93	.9	.90
.10	.92	1.0	.90

^a L_p = distance a culvert barrel projects beyond a headwall or embankment; and
 D = the inside diameter of a pipe culvert or the maximum inside height of a pipe-arch culvert.

Flush setting in vertical headwall.—The discharge coefficient for box or pipe culverts placed flush in a vertical headwall is given in table 4-7. The coefficients in table 4-7 apply to square-end pipe and box culverts, corrugated-metal pipe and pipe-arch culverts, concrete pipe culverts that have a beveled entrance, and box culverts that have rounded or beveled sides. Type 5 flow usually will not occur in a pipe culvert that has flared ends.

Wingwall entrance.—For pipes placed flush in a vertical headway, the addition of wingwalls does not affect the discharge coefficient given in table 4-7. The discharge coefficient for box culverts that have wingwalls and a square entrance is given in table 4-8. If the entrance is rounded or beveled, the value of $\frac{r}{D}$ or $\frac{W}{D}$ for the entrance is used to select a discharge coefficient from table 4-4. However, the discharge coefficient obtained from table 4-8 is used as a lower limit.

Table 4-7. Type 5 flow discharge coefficient, C_c , for box or pipe culverts placed flush in a vertical headwall for various ratios of entrance submergence and entrance rounding or beveling.

Entrance submergence ratio ^a	Discharge coefficient C_c for an entrance rounding or beveling ratio r/b , w/b , r/D , or w/D of						
	0.00	0.02	0.04	0.06	0.08	0.10	0.14
1.4	0.44	0.46	0.49	0.50	0.50	0.51	0.51
1.5	.46	.49	.52	.53	.53	.54	.54
1.6	.47	.51	.54	.55	.55	.56	.56
1.7	.48	.52	.55	.57	.57	.57	.57
1.8	.49	.54	.57	.58	.58	.58	.58
1.9	.50	.55	.58	.59	.60	.60	.60
2.0	.51	.56	.59	.60	.61	.61	.62
2.5	.54	.59	.62	.64	.64	.65	.66
3.0	.55	.61	.64	.66	.67	.69	.70
3.5	.57	.62	.65	.67	.69	.70	.71
4.0	.58	.63	.66	.68	.70	.71	.72
5.0	.59	.64	.67	.69	.71	.72	.73

^aEntrance submergence ratio = $(z_s^h - z_{inv})/D$, where z_s^h is the water surface elevation at the culvert entrance; z_{inv} is the invert elevation at the culvert entrance; and D is the inside height of a box culvert or the inside diameter of a pipe culvert.

Projecting entrance.—The discharge coefficient for pipe or pipe-arch culverts that extend past a headwall or embankment is computed by multiplying the coefficient obtained from table 4-7 by the adjustment factor, k , given in table 4-6. The discharge coefficient for projecting concrete pipe culverts is obtained from table 4-7 but is not adjusted for a projecting entrance.

Mitered pipe placed flush in an embankment.—The discharge coefficient for mitered pipe culverts placed flush in a sloping embankment is computed by multiplying the coefficient obtained from table 4-7 by 0.92. If the mitered pipe is thin-walled (for example, a corrugated metal culvert) and projects beyond the embankment, the discharge coefficient also is multiplied by the adjustment factor k_L given in table 4-6.

Table 4-8. Type 5 flow discharge coefficient, C_c , for box culverts that have wingwalls for various ratios of entrance submergence and various wingwall angles.

Entrance submergence ratio ^a	Discharge coefficient C_c for a wingwall angle in degrees of				
	30	45	60	75	90
1.3	0.44	0.44	0.43	0.42	0.39
1.4	.46	.46	.45	.43	.41
1.5	.47	.47	.46	.45	.42
1.6	.49	.49	.48	.46	.43
1.7	.50	.50	.48	.47	.44
1.8	.51	.51	.50	.48	.45
1.9	.52	.52	.51	.49	.46
2.0	.53	.53	.52	.49	.46
2.5	.56	.56	.54	.52	.49
3.0	.58	.58	.56	.54	.50
3.5	.60	.60	.58	.55	.52
4.0	.61	.61	.59	.56	.53
5.0	.62	.62	.60	.58	.54

^a Entrance submergence ratio = $(z_s^h - z_{inv})/D$, where z_s^h is the water surface elevation at the culvert entrance; z_{inv} is the invert elevation at the culvert entrance; and D is the inside height of the culvert.

Two-Dimensional Bridge/Culvert Flow

Two-dimensional flow through a bridge or culvert is modeled exactly as ordinary free-surface flow when the water surface is not in contact with the top of the bridge or culvert opening (unconfined flow). When the water surface is in contact with the top of the opening (hereafter referred to as the "ceiling") confined, or pressure, flow conditions exist. The depth-averaged flow equations are modified at node points where pressure flow occurs, and pressure head rather than depth is computed. Although it usually is not practical to include them in a network, the effect of piers and piles on flow can be taken into account by increasing bed friction coefficients within a bridge opening.

Depth-averaged pressure flow through a bridge or culvert is modeled by specifying a "ceiling" elevation at node points within the opening. When the water surface is in contact with the ceiling, pressure flow exists and the governing depth-averaged flow equations are modified. The momentum equations become

$$H \frac{\partial U}{\partial t} + \frac{\partial}{\partial x} (\beta H U U + g H P - \frac{1}{2} g H^2) + \frac{\partial}{\partial y} (\beta H U V) + g P \frac{\partial z_b}{\partial x} - g (P - H) \frac{\partial z_c}{\partial x} - \Omega H V + \frac{1}{\rho} \left[\tau_x^b + \tau_x^c - \frac{\partial}{\partial x} (H \tau_{xx}) - \frac{\partial}{\partial y} (H \tau_{xy}) \right] = 0 \quad (4-24)$$

in the x direction, and

$$H \frac{\partial V}{\partial t} + \frac{\partial}{\partial x} (\beta H V U) + \frac{\partial}{\partial y} (\beta H V V + g H P - \frac{1}{2} g H^2) + g P \frac{\partial z_b}{\partial y} - g (P - H) \frac{\partial z_c}{\partial y} + \Omega H U + \frac{1}{\rho} \left[\tau_y^b + \tau_y^c - \frac{\partial}{\partial x} (H \tau_{yx}) - \frac{\partial}{\partial y} (H \tau_{yy}) \right] = 0 \quad (4-25)$$

in the y direction, and the continuity equation becomes

$$\frac{\partial}{\partial x} (H U) + \frac{\partial}{\partial y} (H V) = q, \quad (4-26)$$

where P is pressure head; z_c is the ceiling elevation; τ_x^c and τ_y^c are directional components of shear stress at the ceiling; and $H = z_c - z_b$. The dependent variables in the confined flow case are U, V, and P. The effect of increased frictional resistance created by contact between water and the ceiling is described by the surface shear stress term. The directional components of ceiling shear stress are computed as

$$\tau_x^c = \rho c_f m_c U \sqrt{U^2 + V^2} \quad (4-27)$$

and

$$\tau_y^c = \rho c_f m_c V \sqrt{U^2 + V^2} \quad (4-28)$$

where

$$m_c = \sqrt{1 + \left(\frac{\partial z_c}{\partial x}\right)^2 + \left(\frac{\partial z_c}{\partial y}\right)^2} \quad (4-29)$$

is a factor that accounts for increased resistance caused by a sloping ceiling, and c_f is considered to be the same dimensionless friction coefficient used to model the bed shear stress. When pressure flow occurs, surface stresses caused by wind are not considered.

Initial and Boundary Conditions

Initial conditions and boundary conditions need to be specified to solve the system of depth-averaged flow equations. From the mathematical point of view, the initial conditions and the number and kind of boundary conditions that are specified need to make the problem well-posed (that is, stable). A well-posed problem is one in which increasingly smaller changes to boundary conditions produce increasingly smaller changes in the solution at points not located on the boundary. When an incorrect number of boundary conditions or boundary conditions of the wrong type are prescribed, small changes to the boundary conditions may result in large changes in the solution on the interior of the modeled region. A system of equations that exhibits this kind of unstable behavior is said to be ill-posed.

Initial Conditions

To obtain a solution, both the water depth and the depth-averaged x and y velocity components need to be specified as initial conditions of the problem throughout the entire solution region. When initial conditions are unknown, a cold-start procedure is used. During a cold-start procedure, the same water-surface elevation is assigned to every node point in a finite element network, and velocities are set to zero every-

where. When results from a previous run are available, they can be used as initial conditions for a subsequent run. The use of results from a previous run as initial conditions is referred to as a hot start.

Boundary Conditions

Boundary conditions are specified around the entire boundary of a network for the duration of a simulation. Boundary condition specifications consist of either the normal mass flux (normal flow) or the normal force (normal stress), in addition to either the tangential mass flux (tangential flow) or the tangential force (shear stress) at all points on the boundary of a network.

The required boundary information depends on the type of boundary and the flow condition. Physically, there are two types of boundaries that are encountered in surface-water flow problems: (1) A solid, or no-flux, boundary; and (2) an open boundary.

Solid boundary

A solid boundary defines a geometric feature such as a natural shoreline, a highway embankment, a jetty, or a seawall. The flow across a solid boundary generally equals zero. In addition, either the tangential velocity or tangential stress needs to be specified on a solid boundary. Three types of conditions can be prescribed on a solid boundary: (1) A "slip" condition, (2) a "no-slip" condition, and (3) a "semi-slip" condition.

Slip condition.—A slip condition allows flow in a direction that is tangent to the boundary at a node point and imposes zero tangential shear stress at the boundary. The tangential direction at a boundary node is determined by requiring that the net flow across the solid boundary resulting from velocities at

the node be zero. Slip conditions are usually applied when the solid boundary represents an imaginary vertical wall where flow depths are shallow and lateral shear stresses are negligible.

No-slip condition.—A no-slip condition is prescribed at a solid boundary node by setting the velocities equal to zero; hence the requirement of zero net flow across the boundary is automatically satisfied. No-slip conditions are usually applied when velocities along a boundary are known to be very small and a network of closely-spaced node points is constructed to resolve any large velocity gradients that may exist near the boundary.

Semi-slip condition.—A semi-slip condition is imposed on a solid boundary by allowing flow in a direction that is tangent to the boundary just as for a slip condition, and by prescribing a non-zero tangential shear stress caused by friction generated by flow against a vertical wall. Vertical wall friction τ_w is computed as

$$\tau_w = c_f \rho U_t^2 ,$$

where c_f is the same dimensionless friction coefficient used to calculate bed shear stresses within an element, and U_t is the velocity tangent to the wall (that is, at the boundary node where a vertical wall is assumed to exist). Semi-slip conditions are usually applied when the solid boundary represents an actual physical boundary such as a wall that is vertical or nearly vertical. Increased frictional resistance caused by the wall will then be considered.

Open boundary

An open boundary, which is exactly what the name implies, defines an area where flow is allowed to enter (an inflow boundary) or leave (an outflow boundary) a finite element network. The values that need to be specified at an open boundary depend on the type of boundary (inflow or outflow) and the type of flow (subcritical or supercritical).

Inflow boundary.—If the flow at an inflow boundary node is subcritical, either (1) unit flow normal to the boundary and unit flow tangential to the boundary, or (2) water-surface elevation and tangential shear stress need to be prescribed.

If the flow at an inflow boundary node is supercritical, unit flow normal to the boundary, unit flow tangential to the boundary, and water-surface elevation need to be prescribed at the node.

Tangential shear stresses acting on an open boundary are automatically set to zero if unit flow tangent to the boundary is not specified. Velocity rather than unit flow can be specified at an open boundary node. However, the ability to prescribe velocity directly at a node point seems to offer no practical advantages.

Usually unit flow in both the x and y directions will be specified at inflow boundary nodes, and water-surface elevation (from which depth is determined by subtracting the ground elevation) is specified at outflow boundary nodes of a channel/flood plain model. Total flow at a cross section composed of nodes lying on the network boundary can also be specified. Assigning open boundary inflows using this feature of FLOMOD greatly simplifies the specification of unit flows at upstream boundaries of channel/flood plain models (outflows can be specified as well). Water-surface elevations along a cross section composed of boundary nodes can also be specified. Water-surface elevations may be constant across the section, or slope from one side of the cross section to the other.

Outflow boundary.—If flow at an outflow boundary node is subcritical, water-surface elevation and tangential shear stress need to be prescribed. Tangential shear stresses are set to zero automatically, so only water-surface elevation needs to be specified.

If flow at an outflow boundary node is supercritical, only tangential shear stresses are prescribed, and this is done automatically. However, the fact that a node is a supercritical outflow boundary node still needs to be specified.

Section 5

FINITE ELEMENT EQUATIONS

The method of weighted residuals using Galerkin weighting is applied to the governing depth-averaged flow equations to form the finite element equations. Because the system of equations is nonlinear, Newton's iterative method is used to solve them (see, for example, Zienkiewicz, 1977, p. 452). At each iteration of the solution, the finite element equations express a residual; hence, these equations are referred to as residual expressions. In addition, a matrix of derivatives with respect to each dependent variable for each residual expression is required. This matrix is called the Jacobian matrix and each of its members is defined by a derivative expression. The finite element formulations of the residual and derivative expressions at the i th node point are presented in the following sections. Application of boundary and other "special" conditions also is described.

Residual Expressions

Finite element expressions for the residuals of the depth-averaged momentum equations weighted with respect to a function defined at node i are written as follows:

$$\begin{aligned}
 f_{1i} = & \sum_e \int_{A_e} \left\{ N_i \left[H \frac{\partial U}{\partial t} + U \frac{\partial H}{\partial t} + gH \frac{\partial z_b}{\partial x} - \Omega HV + \frac{1}{\rho} (\tau_x^b - \tau_x^s) \right] \right. \\
 & + \frac{\partial N_i}{\partial x} \left[-\beta HUU - \frac{1}{2} gH^2 + 2\tilde{v}H \frac{\partial U}{\partial x} \right] + \frac{\partial N_i}{\partial y} \left[-\beta HUV + \tilde{v}H \left(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x} \right) \right] \left. \right\} dA_e \\
 & + \sum_e \int_{S_e} N_i \left[(\beta HUU + \frac{1}{2} gH^2) \ell_x + \beta HUV \ell_y \right] dS_e
 \end{aligned}$$

$$- \sum_e \int_{S_e} N_i \left[2\tilde{\nu}H \frac{\partial U}{\partial x} \ell_x + \tilde{\nu}H \left(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x} \right) \ell_y \right] dS_e \quad (5-1)$$

for flow in the the x-direction, and

$$\begin{aligned} f_{2i} = & \sum_e \int_{A_e} \left\{ N_i \left[H \frac{\partial V}{\partial t} + v \frac{\partial H}{\partial t} + gH \frac{\partial z_b}{\partial y} + \Omega HU + \frac{1}{\rho} (\tau_y^b - \tau_y^s) \right] \right. \\ & + \frac{\partial N_i}{\partial x} \left[-\beta HUV + \tilde{\nu}H \left(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x} \right) \right] + \frac{\partial N_i}{\partial y} \left[-\beta HVV - \frac{1}{2}gH^2 + 2\tilde{\nu}H \frac{\partial V}{\partial y} \right] \left. \right\} dA_e \\ & + \sum_e \int_{S_e} N_i \left[\beta HUV \ell_x + \left(\beta HVV + \frac{1}{2}gH^2 \right) \ell_y \right] dS_e \\ & - \sum_e \int_{S_e} N_i \left[\tilde{\nu}H \left(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x} \right) \ell_x + 2\tilde{\nu}H \frac{\partial V}{\partial y} \ell_y \right] dS_e \quad (5-2) \end{aligned}$$

for flow in the the y-direction, where \sum_e indicates a summation with respect to all elements, A_e indicates an element surface, S_e indicates an element boundary, and ℓ_x and ℓ_y are the direction cosines between the outward normal to the boundary and the x and y directions, respectively. All second-order derivatives in the momentum expressions have been integrated by parts using the Green-Gauss theorem. Reduction of the order of the expressions in this way allows use of quadratic functions to interpolate velocities. The advection and pressure terms in the momentum equations also have been integrated by parts. Integration by parts of the advection terms simplifies the finite element equation formulation, and integration by parts of the pressure terms facilitates application of normal-stress boundary conditions. The last boundary integral in the two momentum residual expressions represents the lateral stress resulting from the transport of momentum by turbulence.

The expression for the weighted residual of the continuity equation is

$$f_{3i} = \sum_e \int_{A_e} M_i \left[\frac{\partial H}{\partial t} + H \frac{\partial U}{\partial x} + U \frac{\partial H}{\partial x} + H \frac{\partial V}{\partial y} + V \frac{\partial H}{\partial y} \right] dA_e - Q_i \quad (5-3)$$

where

$$Q_i = \sum_e \int_{A_e} M_i q dA_e$$

is the total source/sink flow attributed to node i.

Time Derivatives

Expressions 5-1, 5-2, and 5-3 apply to a particular instant in time. For a steady-state solution all the time derivatives are equal to zero and do not need to be evaluated. However, if the solution is time-dependent the residuals need to be integrated with respect to time as well as with respect to space. The temporal derivative of a dependent variable, U for example, is integrated as follows:

$$U^{j+1} = U^j + \Delta t \left[(1-\theta) \frac{\partial U^j}{\partial t} + \theta \frac{\partial U^{j+1}}{\partial t} \right], \quad (5-4)$$

where Δt is the length of time (that is the time step) between time levels j and j+1, and the factor θ controls the degree of implicitness of the time integration. For $\theta = 0$, the the integration scheme is explicit (forward Euler), for $\theta = 1$, the integration scheme is implicit (backward Euler), and for $\theta = 0.5$, a trapezoidal (Crank-Nicholson) integration scheme results. Setting θ equal to 0.67 has been found to produce an accurate and stable solution even for relatively large time steps (King and Norton, 1978). An approximation of $\partial U / \partial t$ at the advanced time level is obtained from equation 5-4:

$$\frac{\partial U^{j+1}}{\partial t} = \frac{1}{\theta} \left(\frac{U^{j+1} - U^j}{\Delta t} \right) - \left(\frac{1-\theta}{\theta} \right) \frac{\partial U^j}{\partial t} . \quad (5-5)$$

The expression for $\partial U/\partial t$ at the advanced time level can be rewritten as

$$\frac{\partial U^{j+1}}{\partial t} = \alpha U^{j+1} + \beta_1 , \quad (5-6)$$

where $\alpha = \frac{1}{\theta \Delta t}$, and

$$\beta_1 = \alpha U^j + \left(\frac{1-\theta}{\theta} \right) \frac{\partial U^j}{\partial t} \quad (5-7)$$

is a factor that contains values known from the solution at the previous time step.

In a similar way, time derivatives of V and H are approximated as

$$\frac{\partial V^{j+1}}{\partial t} = \alpha V^{j+1} + \beta_2 , \quad (5-8)$$

and

$$\frac{\partial H^{j+1}}{\partial t} = \alpha H^{j+1} + \beta_3 , \quad (5-9)$$

where

$$\beta_2 = \alpha V^j + \left(\frac{1-\theta}{\theta} \right) \frac{\partial V^j}{\partial t} ; \quad (5-10)$$

and

$$\beta_3 = \alpha H^j + \left(\frac{1-\theta}{\theta} \right) \frac{\partial H^j}{\partial t} . \quad (5-11)$$

Derivative Expressions

The finite element formulations of the the expressions for derivatives of residuals at node i with respect to variables at node j as defined as follows:

$$\begin{aligned}
 \frac{\partial f_{1i}}{\partial U_j} &\equiv \sum_e \int_{A_e} \left\{ N_i N_j \left[\alpha H + \frac{\partial H}{\partial t} + \frac{1}{\rho} \tau_x^b \frac{(2U^2 + V^2)}{U(U^2 + V^2)} \right] \right. \\
 &+ \frac{\partial N_i}{\partial x} N_j [-2\beta H U] + \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} [2\tilde{v} H] + \frac{\partial N_i}{\partial y} N_j [-\beta H V] + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} [\tilde{v} H] \left. \right\} dA_e \\
 &+ \sum_e \int_{A_e} \left\{ N_i N_j [2\beta H U \ell_x + \beta H V \ell_y] - N_i \frac{\partial N_i}{\partial x} [2\tilde{v} H \ell_x] - N_i \frac{\partial N_i}{\partial y} [\tilde{v} H \ell_y] \right\} dS_e
 \end{aligned} \tag{5-12}$$

$$\begin{aligned}
 \frac{\partial f_{1i}}{\partial V_j} &\equiv \sum_e \int_{A_e} \left\{ N_i N_j \left[-\Omega H + \frac{1}{\rho} \tau_x^b \frac{V}{(U^2 + V^2)} \right] \right. \\
 &+ \frac{\partial N_i}{\partial y} N_i [-\beta H U] + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial x} [\tilde{v} H] \left. \right\} dA_e \\
 &+ \sum_e \int_{S_e} \left\{ N_i N_j [\beta H U \ell_y] - N_i \frac{\partial N_j}{\partial x} [\tilde{v} H \ell_y] \right\} dS_e
 \end{aligned} \tag{5-13}$$

$$\begin{aligned}
 \frac{\partial f_{1i}}{\partial H_j} &\equiv \sum_e \int_{A_e} \left\{ N_i M_j \left[\frac{\partial U}{\partial t} + \alpha U - \Omega V + g \frac{\partial z_b}{\partial x} + \frac{1}{\rho} \tau_x^b \frac{1}{c_f} \frac{\partial c_f}{\partial H} \right] \right. \\
 &+ \frac{\partial N_i}{\partial x} M_j [-\beta U U - g H + 2\tilde{v} \frac{\partial U}{\partial x}] + \frac{\partial N_i}{\partial y} M_j [-\beta U V + \tilde{v} (\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x})] \left. \right\} dA_e \\
 &+ \sum_e \int_{S_e} N_i M_j \left\{ \left[\beta U U + g H - 2\tilde{v} \frac{\partial U}{\partial x} \right] \ell_x + \left[\beta U V - \tilde{v} (\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x}) \right] \ell_y \right\} dS_e
 \end{aligned} \tag{5-14}$$

$$\begin{aligned}
\frac{\partial f_{2i}}{\partial U_j} &\equiv \sum_e \int_{A_e} \left\{ N_i N_j \left[\Omega H + \frac{1}{\rho} \tau_Y^b \frac{U}{(U^2 + V^2)} \right] \right. \\
&\quad \left. + \frac{\partial N_i}{\partial x} N_i [-\beta H V] + \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial Y} [\tilde{v} H] \right\} dA_e \\
&\quad + \sum_e \int_{S_e} \left\{ N_i N_j [\beta H V l_x] - N_i \frac{\partial N_j}{\partial Y} [\tilde{v} H l_x] \right\} dS_e \quad (5-15)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial f_{2i}}{\partial V_j} &\equiv \sum_e \int_{A_e} \left\{ N_i N_j \left[\alpha H + \frac{\partial H}{\partial t} + \frac{1}{\rho} \tau_Y^b \frac{(U^2 + 2V^2)}{V(U^2 + V^2)} \right] \right. \\
&\quad \left. + \frac{\partial N_i}{\partial x} N_j [-\beta H U] + \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} [\tilde{v} H] + \frac{\partial N_i}{\partial Y} N_j [-2\beta H V] + \frac{\partial N_i}{\partial Y} \frac{\partial N_j}{\partial Y} [2\tilde{v} H] \right\} dA_e \\
&\quad + \sum_e \int_{S_e} N_i N_j [\beta H U l_x + 2\beta H V l_y] - N_i \frac{\partial N_i}{\partial x} [\tilde{v} H l_x] - N_i \frac{\partial N_i}{\partial Y} [2\tilde{v} H l_y] dS_e \quad (5-16)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial f_{2i}}{\partial H_j} &\equiv \sum_e \int_{A_e} \left\{ N_i M_j \left[\frac{\partial V}{\partial t} + \alpha V - \Omega U + g \frac{\partial z_b}{\partial x} + \frac{1}{\rho} \tau_x^b \frac{1}{c_f} \frac{\partial c_f}{\partial H} \right] \right. \\
&\quad \left. + \frac{\partial N_i}{\partial x} M_j \left[-\beta U V + \tilde{v} \left(\frac{\partial U}{\partial Y} + \frac{\partial V}{\partial x} \right) \right] + \frac{\partial N_i}{\partial Y} M_j \left[-\beta V V - g H + 2\tilde{v} \frac{\partial V}{\partial Y} \right] \right\} dA_e \\
&\quad + \sum_e \int_{S_e} N_i M_j \left\{ \left[\beta U V - \tilde{v} \left(\frac{\partial U}{\partial Y} + \frac{\partial V}{\partial x} \right) \right] l_x + \left[\beta V V + g H - 2\tilde{v} \frac{\partial V}{\partial Y} \right] l_y \right\} dS_e \quad (5-17)
\end{aligned}$$

$$\frac{\partial f_{3i}}{\partial U_j} \equiv \sum_e \int_{A_e} \left\{ M_i \frac{\partial N_j}{\partial x} [H] + M_i N_i \left[\frac{\partial H}{\partial x} \right] \right\} dA_e - \frac{\partial Q_i}{\partial U_j} \quad (5-18)$$

$$\frac{\partial f_{3i}}{\partial V_j} \equiv \sum_e \int_{A_e} \left\{ M_i \frac{\partial N_j}{\partial Y} [H] + M_i N_i \left[\frac{\partial H}{\partial Y} \right] \right\} dA_e - \frac{\partial Q_i}{\partial V_j} \quad (5-19)$$

$$\frac{\partial f_{3i}}{\partial H_j} = \sum_e \int_{A_e} \left\{ M_i M_j \left[\alpha + \frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} \right] + M_i \frac{\partial M_j}{\partial x} [U] + M_i \frac{\partial M_j}{\partial y} [V] \right\} dA_e - \frac{\partial Q_i}{\partial H_j} \quad (5-20)$$

where

$$\frac{\partial c_f}{\partial H} = \begin{cases} 0, & \text{if Chézy discharge coefficients are used} \\ \frac{\phi g n^2}{H^{4/3}}, & \text{if Manning roughness coefficients are used} \end{cases} ;$$

and

$$\phi = \begin{cases} 0.151 & \text{for U.S. Customary units} \\ 0.333 & \text{for S.I. units} \end{cases} .$$

Application of Boundary and Special Conditions

The Galerkin finite element formulation allows complicated boundary conditions to be automatically satisfied as natural conditions of the problem. Natural boundary conditions are implicitly imposed in the problem statement and require no further treatment. Boundary conditions that are imposed explicitly are known as forced or essential conditions. Essential boundary conditions are prescribed by modifying the finite element equation governing that variable. In addition, special boundary conditions imposed by one-dimensional flow at culverts and weirs can be easily applied.

Open Boundaries

Velocities and depth can be applied as essential boundary conditions at any node point on an open boundary as long as the system of equations does not become overconstrained. Velocities and depth are prescribed at node point i by redefining the residual expressions as

$$f_{1i} \equiv U_i^* , \quad (5-21)$$

$$f_{2i} \equiv V_i^* , \quad (5-22)$$

and

$$f_{3i} \equiv H_i^* ; \quad (5-23)$$

where U_i^* , V_i^* , and H_i^* are the specified values; and redefining the derivatives as

$$\frac{\partial f_{1i}}{\partial U_j} \equiv \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}; \quad \frac{\partial f_{1i}}{\partial V_j} \equiv 0; \quad \frac{\partial f_{1i}}{\partial H_j} \equiv 0; \quad (5-24a,b,c)$$

$$\frac{\partial f_{2i}}{\partial U_j} \equiv 0; \quad \frac{\partial f_{2i}}{\partial V_j} \equiv \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}; \quad \frac{\partial f_{2i}}{\partial H_j} \equiv 0; \quad (5-25a,b,c)$$

$$\frac{\partial f_{3i}}{\partial U_j} \equiv 0; \quad \frac{\partial f_{3i}}{\partial V_j} \equiv 0; \quad \frac{\partial f_{3i}}{\partial H_j} \equiv \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}; \quad (5-26a,b,c)$$

Unit flow rates are applied at node i in a similar manner by redefining the momentum equation residuals as

$$f_{1i} = U_i H_i - q_{xi}^* \quad (5-27)$$

and

$$f_{2i} = V_i H_i - q_{yi}^*; \quad (5-28)$$

where q_{xi}^* and q_{yi}^* are specified unit flow rates in the x and y directions, respectively, at node i ; and by redefining the conservation of momentum residual expression derivatives as

$$\frac{\partial f_{1i}}{\partial U_j} \equiv \begin{cases} H_i, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}; \quad \frac{\partial f_{1i}}{\partial V_j} \equiv 0; \quad \frac{\partial f_{1i}}{\partial H_j} \equiv \begin{cases} U_i, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}; \quad (5-29a,b,c)$$

and

$$\frac{\partial f_{2i}}{\partial U_j} \equiv 0; \quad \frac{\partial f_{2i}}{\partial V_j} \equiv \begin{cases} H_i, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}; \quad \frac{\partial f_{2i}}{\partial H_j} \equiv \begin{cases} U_i, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases};$$

(5-30a,b,c)

Depth also can be applied as a natural boundary condition by using the specified value of depth at node i , H_i^* , to evaluate the boundary integral terms in the conservation of momentum residual expressions 5-1 and 5-2. Contributions from the boundary-integral terms are taken as zero when derivatives of the momentum equation residuals with respect to H_i^* are computed.

When water depth is specified as a natural boundary condition, global mass conservation is insured and total inflow will equal total outflow in steady-state simulations. However, water depths computed at nodes where the water-surface elevation is applied as a natural boundary condition may differ slightly from the specified values. When water depth is specified as an essential boundary condition, the computed depth will equal the specified depth, but the total outflow may differ slightly from the total inflow in steady-state simulations because the mass conservation equations at node points along the boundary have been replaced.

If total flow through a cross section that forms part of the open boundary of a finite element network is specified, a constant friction slope along the section is assumed and the total flow is divided among the node points on the basis of conveyance. The cross section is defined by a list of node points that form a connected series of element sides. Each element side is composed of three nodes (1, 2, and 3) where nodes 1 and 3 are corner nodes, and node 2 is a midside node. Conveyance through each element side is defined as

$$K = A\sqrt{gR/c_f} , \quad (5-31)$$

where R is the hydraulic radius (area divided by wetted perimeter) of the element side; and A is the area of the element side below the water surface. Total conveyance for the cross section is computed as the sum of the conveyance of each element side that is contained in the section.

Conveyance through each element side is distributed among the three nodes that form the side as follows:

$$K_1 = \frac{1}{6} K(1 - \zeta); \quad K_2 = \frac{2}{3} K; \quad K_3 = \frac{1}{6} K(1 + \zeta); \quad (5-32a,b,c)$$

where $\zeta = 5\Delta H/12\bar{H}$; $\Delta H = H_3 - H_1$; $\bar{H} = (H_1 + H_3)/2$; H_1 is the depth at node 1; and H_3 is the depth at node 3. Total flow normal to the open boundary at each cross section node point is computed on the basis of the ratio of conveyance assigned to each node to the total conveyance computed for the cross section. The velocities and depth computed at each node are required to satisfy the condition that the net flow across the open boundary resulting from flow at the node will equal the assigned portion of the total cross section flow. The procedure used to specify net flow across a boundary that results from a single node point is described in a subsequent section.

Solid Boundaries

Solid boundaries define features such as natural shorelines, jetties, or seawalls. For viscous fluids, the velocity at a solid boundary is actually zero. This is commonly referred to as a "no-slip" boundary condition. A no-slip condition can be specified by applying x and y velocities of zero as essential boundary conditions. To accurately model the flow near a boundary at which a no-slip condition has been imposed, a network

composed of relatively small elements is needed. However, for practical purposes a "slip" condition usually is applied at a solid boundary whereby flow is allowed to move in a direction tangent to the boundary. Imposing a slip condition at solid boundaries reduces the total number of elements needed in a network and thus decreases the number of equations that need to be solved. Slip conditions are applied at a solid boundary node by first transforming the x and y momentum equations that are associated with that node into equations that express conservation of momentum in directions that are tangent and normal to the boundary. The conservation of momentum equation for flow in the normal direction is then replaced by a constraint equation that requires the net flow across the solid boundary that results from flow at the node point to equal zero. This procedure is described in the following section.

Total Flow Across a Boundary

Total flow across a boundary (normal flow) at a node point comes from several sources. Flow across an open boundary is defined as

$$Q_i^o \equiv Q_{si}^o + Q_{xi} , \quad (5-33)$$

where Q_{si}^o is the flow normal to the boundary at node i that is specified directly; and Q_{xi} is the amount of the total flow through a cross section that is assigned to node i by the procedure discussed in the subsection on open boundaries. Flow across a solid boundary is defined as

$$Q_i^s \equiv Q_{si}^s + Q_{wi} + Q_{ci} , \quad (5-34)$$

where Q_{si}^s is the flow normal to the solid boundary at node i that is specified directly; Q_{wi} is the computed flow over a weir (roadway embankment) segment at node i; and Q_{ci} is the computed flow through a culvert at node i.

Along a boundary (either open or solid) where flow normal to the boundary is to be prescribed, the conservation of momentum residual expressions for flows in the x and y directions first are transformed into conservation of momentum residual expressions for flows in directions that are tangent and normal to the boundary. At node point i, the transformation is accomplished as follows:

$$f'_{1i} = f_{1i} \cos \delta + f_{2i} \sin \delta , \quad (5-35)$$

$$f'_{2i} = -f_{1i} \sin \delta + f_{2i} \cos \delta , \quad (5-36)$$

where f'_{1i} and f'_{2i} are the transformed residual expressions in the tangential and normal directions, respectively; and δ is the angle between the positive x direction and a tangent to the boundary at node i.

If flow normal to an open boundary at node i is specified, the residual expression for flow tangent to the boundary is redefined as

$$f_{1i} \equiv a_i^O U_i + b_i^O V_i - Q_i^O . \quad (5-37)$$

If flow normal to a solid boundary at node i is specified, the conservation of momentum equation for flow normal to the boundary is redefined as

$$f_{2i} \equiv a_i^S U_i + b_i^S V_i - Q_i^S . \quad (5-38)$$

The coefficients a_i^O , b_i^O , a_i^S , and b_i^S in expressions 5-37 and 5-38 are determined by requiring the computed flow across an open or solid boundary at node i to equal the specified flow, that is

$$U_i \sum_e \int_{S_e^O} N_i H \ell_x \, dS_e^O + V_i \sum_e \int_{S_e^O} N_i H \ell_y \, dS_e^O = Q_i^O \quad (5-39)$$

and

$$U_i \sum_e \int_{S_e^O} N_i H l_x ds_e^S + V_i \sum_e \int_{S_e^O} N_i H l_y ds_e^S = Q_i^S, \quad (5-40)$$

where N_i is the interpolation function for velocity at node i ; S_e^O is the part of the network boundary that is open; and S_e^S is the part of the network boundary that is solid. Comparing expression 5-37 to equation 5-39, and expression 5-38 to equation 5-40, it is readily seen that

$$a_i^O \equiv \sum_e \int_{S_e^O} N_i H l_x ds_e^O; \quad (5-41)$$

$$b_i^O \equiv \sum_e \int_{S_e^O} N_i H l_y ds_e^O; \quad (5-42)$$

$$a_i^S \equiv \sum_e \int_{S_e^S} N_i H l_x ds_e^S; \quad (5-43)$$

and

$$b_i^S \equiv \sum_e \int_{S_e^S} N_i H l_y ds_e^S; \quad (5-44)$$

Derivatives of the residual expression for total flow across an open boundary are defined as follows:

$$\frac{\partial f_{1i}}{\partial U_j} \equiv \begin{cases} a_i^O, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}; \quad (5-45)$$

$$\frac{\partial f_{1i}}{\partial V_j} \equiv \begin{cases} b_i^O, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}; \quad (5-46)$$

$$\frac{\partial f_{1i}}{\partial H_j} \equiv \frac{\partial a_i^O}{\partial H_j} U_i + \frac{\partial b_i^O}{\partial H_j} V_i; \quad (5-47)$$

where

$$\frac{\partial a_i^o}{\partial H_j} \equiv \sum_e \int_{S_e^o} N_i M_j^l x \, ds_e^o \quad (5-48)$$

and

$$\frac{\partial b_i^o}{\partial H_j} \equiv \sum_e \int_{S_e^o} N_i M_j^l y \, ds_e^o . \quad (5-49)$$

Derivatives of the residual expression for total flow across a solid boundary are defined as follows:

$$\frac{\partial f_{2i}}{\partial U_j} = \begin{cases} a_i^s, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases} ; \quad (5-50)$$

$$\frac{\partial f_{2i}}{\partial V_j} = \begin{cases} b_i^s, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases} ; \quad (5-51)$$

$$\frac{\partial f_{2i}}{\partial H_j} \equiv \frac{\partial a_i^s}{\partial H_j} U_i + \frac{\partial b_i^s}{\partial H_j} V_i - \frac{\partial Q_{wi}}{\partial H_j} - \frac{\partial Q_{ci}}{\partial H_j} ; \quad (5-52)$$

where

$$\frac{\partial a_i^s}{\partial H_j} \equiv \sum_e \int_{S_e^s} N_i M_j^l x \, ds_e^s , \quad (5-53)$$

$$\frac{\partial b_i^s}{\partial H_j} \equiv \sum_e \int_{S_e^s} N_i M_j^l y \, ds_e^s , \quad (5-54)$$

$$\frac{\partial Q_{wi}}{\partial H_j} \equiv \frac{3}{2} \left(\frac{Q_{wi}}{z_e^h - z_c} \right) , \quad (5-55)$$

and

$$\frac{\partial Q_{ci}}{\partial H_j} \equiv \frac{1}{2} \left(\frac{Q_{ci}}{z_s^h - z_s^t} \right). \quad (5-56)$$

Depth-Averaged Pressure Flow

When two-dimensional flow through a bridge is in contact with the ceiling, pressure flow exists and the pressure, P , at node points replaces the flow depth, H , as the solution variable.

Residual Expressions

In the case of pressure flow, the residual expressions at the i th node point are written as follows:

$$\begin{aligned} f_{1i} = & \sum_e \int_{A_e} \left\{ N_i \left[H \frac{\partial U}{\partial t} + gP \frac{\partial z_b}{\partial x} - g(P - H) \frac{\partial z_c}{\partial x} - \Omega HV + \frac{1}{\rho} (\tau_x^b + \tau_x^c) \right] \right. \\ & + \frac{\partial N_i}{\partial x} \left[-\beta HUU - gH(P - \frac{1}{2}H) + 2\tilde{v}H \frac{\partial U}{\partial x} \right] + \frac{\partial N_i}{\partial x} \left[-\beta HUV + \tilde{v}H \left(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x} \right) \right] \left. \right\} dA_e \\ & + \sum_e \int_{S_e} N_i \left[\beta HUU l_x + gH(P - \frac{1}{2}H) l_x + \beta HUV l_y \right] dS_e \\ & - \sum_e \int_{S_e} N_i \left[2\tilde{v}H \frac{\partial U}{\partial x} l_x + \tilde{v}H \left(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x} \right) l_y \right] dS_e, \quad (5-58) \end{aligned}$$

$$\begin{aligned} f_{2i} = & \sum_e \int_{A_e} \left\{ N_i \left[H \frac{\partial V}{\partial t} + gP \frac{\partial z_b}{\partial y} - g(P - H) \frac{\partial z_c}{\partial y} - \Omega HU + \frac{1}{\rho} (\tau_y^b + \tau_y^c) \right] \right. \\ & + \frac{\partial N_i}{\partial x} \left[-\beta HUV + \tilde{v}H \left(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x} \right) \right] + \frac{\partial N_i}{\partial x} \left[-\beta HUV - gH(P - \frac{1}{2}H) + 2\tilde{v}H \frac{\partial V}{\partial y} \right] \left. \right\} dA_e \\ & + \sum_e \int_{S_e} N_i \left[\beta HUV l_x + \beta HUV l_y + gH(P - \frac{1}{2}H) l_y \right] dS_e \end{aligned}$$

$$- \sum_e \int_{S_e} N_i \left[\tilde{v}H \left(\frac{\partial U}{\partial Y} + \frac{\partial V}{\partial X} \right) \ell_x + 2\tilde{v}H \frac{\partial U}{\partial X} \ell_y \right] dS_e, \quad (5-59)$$

$$f_{3i} = \sum_e \int_{S_e} M_i \left[H \frac{\partial U}{\partial X} + U \frac{\partial H}{\partial X} + H \frac{\partial V}{\partial Y} + V \frac{\partial H}{\partial Y} \right] dA, \quad (5-60)$$

where $H = z_s - z_c$ is the height of the full flowing opening (that is, the distance between the bed and the ceiling).

Derivative Expressions

Derivatives of the depth-averaged pressure-flow residuals are written for variables at node i with respect to variables at node j as follows:

$$\begin{aligned} \frac{\partial f_{1i}}{\partial U_j} &= \sum_e \int_{A_e} \left\{ N_i N_j \left[\alpha H + \frac{1}{\rho} (\tau_x^b + \tau_x^c) \frac{(2U^2 + V^2)}{U(U^2 + V^2)} \right] \right. \\ &+ \frac{\partial N_i}{\partial X} N_j \left[-2\beta H U \right] + \frac{\partial N_i}{\partial X} \frac{\partial N_j}{\partial X} \left[2\tilde{v}H \right] + \frac{\partial N_i}{\partial Y} N_j \left[-\beta H U \right] + \frac{\partial N_i}{\partial Y} \frac{\partial N_j}{\partial Y} \left[\tilde{v}H \right] \left. \right\} dA_e \\ &+ \sum_e \int_{S_e} \left\{ N_i N_j \left[2\beta H U \ell_x + \beta H V \ell_y \right] \right\} dS_e \\ &- \sum_e \int_{S_e} \left\{ N_i \frac{\partial N_j}{\partial X} \left[2\tilde{v}H \ell_x \right] + N_i \frac{\partial N_j}{\partial Y} \left[\tilde{v}H \ell_y \right] \right\} dS_e \end{aligned} \quad (5-61)$$

$$\begin{aligned} \frac{\partial f_{1i}}{\partial V_j} &= \sum_e \int_{A_e} \left\{ N_i N_j \left[-\Omega H + \frac{1}{\rho} (\tau_x^b + \tau_x^c) \frac{V}{(U^2 + V^2)} \right] \right. \\ &+ \frac{\partial N_i}{\partial Y} N_j \left[-\beta H U \right] + \frac{\partial N_i}{\partial Y} \frac{\partial N_j}{\partial X} \left[\tilde{v}H \right] \left. \right\} dA_e \\ &+ \sum_e \int_{S_e} \left\{ N_i N_j \left[\beta H U \ell_y \right] \right\} dS_e - \sum_e \int_{S_e} \left\{ N_i \frac{\partial N_j}{\partial X} \left[\tilde{v}H \ell_y \right] \right\} dS_e \end{aligned} \quad (5-62)$$

$$\begin{aligned} \frac{\partial f_{1i}}{\partial P_j} = & \sum_e \int_{A_e} \left\{ N_i M_j \left[g \left(\frac{\partial z_b}{\partial x} - \frac{\partial z_c}{\partial x} \right) + \frac{\partial N_i}{\partial x} N_j \left[-gH \right] \right\} dA_e \\ & + \sum_e \int_{S_e} N_i M_j \left[gH \ell_x \right] dS_e . \end{aligned} \quad (5-63)$$

$$\begin{aligned} \frac{\partial f_{2i}}{\partial U_j} = & \sum_e \int_{A_e} \left\{ N_i N_j \left[\Omega H + \frac{1}{\rho} (\tau_Y^b + \tau_Y^c) \frac{U}{(U^2 + V^2)} \right] \right. \\ & \left. + \frac{\partial N_i}{\partial x} N_j \left[-\beta H V \right] + \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial Y} \left[\tilde{\nu} H \right] \right\} dA_e \\ & + \sum_e \int_{S_e} \left\{ N_i N_j \left[\beta H V \ell_x \right] \right\} dS_e - \sum_e \int_{S_e} \left\{ N_i \frac{\partial N_j}{\partial Y} \left[\tilde{\nu} H \ell_x \right] \right\} dS_e \end{aligned} \quad (5-64)$$

$$\begin{aligned} \frac{\partial f_{2i}}{\partial V_j} = & \sum_e \int_{A_e} \left\{ N_i N_j \left[\alpha H + \frac{1}{\rho} (\tau_Y^b + \tau_Y^c) \frac{(U^2 + 2V^2)}{U(U^2 + V^2)} \right] \right. \\ & \left. + \frac{\partial N_i}{\partial x} N_j \left[-\beta H U \right] + \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} \left[\tilde{\nu} H \right] + \frac{\partial N_i}{\partial Y} N_j \left[-2\beta H U \right] + \frac{\partial N_i}{\partial Y} \frac{\partial N_j}{\partial Y} \left[2\tilde{\nu} H \right] \right\} dA_e \\ & + \sum_e \int_{S_e} \left\{ N_i N_j \left[\beta H U \ell_x + 2\beta H V \ell_y \right] \right\} dS_e \\ & - \sum_e \int_{S_e} \left\{ N_i \frac{\partial N_j}{\partial x} \left[\tilde{\nu} H \ell_x \right] + N_i \frac{\partial N_j}{\partial Y} \left[2\tilde{\nu} H \ell_y \right] \right\} dS_e \end{aligned} \quad (5-65)$$

$$\begin{aligned} \frac{\partial f_{2i}}{\partial P_j} = & \sum_e \int_{A_e} \left\{ N_i M_j \left[g \left(\frac{\partial z_b}{\partial Y} - \frac{\partial z_c}{\partial Y} \right) + \frac{\partial N_i}{\partial Y} N_j \left[-gH \right] \right\} dA_e \\ & + \sum_e \int_{S_e} N_i M_j \left[gH \ell_y \right] dS_e . \end{aligned} \quad (5-66)$$

$$\frac{\partial f_{3i}}{\partial U_j} = \sum_{\bar{e}} \int_{A_e} \left\{ M_i N_j \left[\frac{\partial H}{\partial x} \right] + M_i \frac{\partial N_j}{\partial x} [H] \right\} dA_e - \frac{\partial Q_i}{\partial U_j}, \quad (5-67)$$

$$\frac{\partial f_{3i}}{\partial V_j} = \sum_{\bar{e}} \int_{A_e} \left\{ M_i N_j \left[\frac{\partial H}{\partial Y} \right] + M_i \frac{\partial N_j}{\partial Y} [H] \right\} dA_e - \frac{\partial Q_i}{\partial H_j}, \quad (5-68)$$

$$\frac{\partial f_{3i}}{\partial P_j} = 0. \quad (5-69)$$

Section 6

MODELING SYSTEM OPERATION

The steps generally taken to operate FESWMS-2DH are: (1) Data collection, (2) network design, (3) calibration, (4) validation, and (5) application. These five steps are common to the operation of almost any type of numerical model and are described in this section.

Data Collection

After a surface-water flow problem has been defined, the first step in the operation of the modeling system consists of gathering data. The required data are classified as either topographic or hydraulic data. Topographic data describe the geometry of the physical system and include an evaluation of surface roughness to be used in estimating bed friction coefficients. Hydraulic data include measurements of stage and flow hydrographs; spot measurements of stage, flow, and velocity; high-water marks left by floods; rating curves; limits of flooding; and wind measurements. Hydraulic data are used to establish model boundary conditions, and to calibrate and validate a model. Data requirements are summarized in table 6-1.

The type and amount of data that are needed to design a network properly and to apply a model mainly depend on the purpose of the model. The more data that can be obtained the better, and all of it can be used to improve the quality of a model's output. Theoretically, any surface-water flow can be simulated as accurately as desired provided the important physical processes are represented adequately by the governing equations. However, the purpose of a model needs to be considered when deciding what and how much data is needed to provide results of the desired accuracy. For example, a finite

Table 6-1. Summary of data that may be needed to apply the modeling system, their use, and possible sources.

Data item	Use of data	Source of data
Ground-surface elevations	Assignment of ground surface elevations at each node, and layout of a network	Hydrographic and topographic charts and field surveys
Bridge and culvert dimensions	Layout of a network, assignment of 1-D bridge and culvert parameters	Design drawings and field surveys
Channel and overbank surface roughness	Evaluation of bed friction coefficient and eddy viscosity	Aerial and ground photographs, field inspection
Water-surface elevations	Establishment of boundary conditions, model calibration, and model validation	Field measurements, gauge records
Current velocity or flow rate	Establishment of boundary conditions, model calibration, and model validation	Field measurements, gauge records
Wind velocity	Computation of water-surface stresses	Field measurements, weather station records
Water temperature	Determination of water density	Field measurement, gauge records
Latitude	Computation of Coriolis force	Map

model's output. Theoretically, any surface-water flow can be simulated as accurately as desired provided the important physical processes are represented adequately by the governing equations. However, the purpose of a model needs to be considered when deciding what and how much data is needed to provide results of the desired accuracy. For example, a finite element model of flow in a laboratory flume might require a computational resolution of inches (or less) to provide the

desired results. On the other hand, a model of a tidal estuary might require a computational resolution of a mile or more.

It is difficult to determine the minimum data requirements for a particular application. Model construction (that is, network design, calibration, and validation) and subsequent application require consideration of the objective of the study and the available time, manpower, and funding. Because time, manpower, and funding always have finite limits, decisions need to be made regarding the degree of detail to be represented by the model and the extent of calibration and validation to be performed. If a high level of detail is provided by a network, the risk of not representing a physical system properly will be reduced, but the difficulty (in time and expense) of obtaining a solution will be increased. On the other hand, if a simple network is designed, the risk of not accurately representing a physical system will be increased, but the difficulty of obtaining a solution will be reduced. A knowledge of the important physical processes that govern the response of a system under study is needed to evaluate the trade-off between risk of not accurately representing the system and difficulty of obtaining a solution. Sometimes constraints on time, manpower, or funding will predetermine the level of discretization to be used and/or the amount of calibration and validation to be performed, thus requiring acceptance of a larger amount of risk than would otherwise be desired.

Network Design

The next step in applying FESWMS-2DH is to design a finite element network. Network design can be defined simply as the process whereby the surface-water body being modeled is subdivided into an assemblage of finite elements. The basic goal of network design is to create a representation of the water body that provides an adequate approximation of the true solution of the governing equations at a reasonable cost. There are no set

rules for achieving this goal because of the many different conditions encountered from one problem to the next. The design of a satisfactory finite element network depends largely on the use of sound engineering judgment gained from previous modeling experience. However, some helpful guidelines are presented in this section.

General Network Layout

Design of a finite element network requires decisions as to the number, size, shape, and configuration of elements used to provide an adequate representation of the water body that is to be modeled. As long as the elements obey some basic requirements for a convergent solution, the accuracy of the solution will improve as the size of the elements in a network is reduced. However, increasing the number of elements in a network also increases computational expenses. Elements need to be made small enough to provide a solution of sufficient detail and accuracy, yet large enough to obtain the solution at a reasonable cost.

The first step in the design process is to obtain a map of the surface water body to be modeled. The map scale and detail that are required depend on the degree of solution accuracy that is desired. Because some trial-and-error probably will be needed during network design, it is best to overlay the map with a clear, gridded mylar sheet that has a matted surface that can be drawn on with a pencil. It will be much easier to erase and redesign on the mylar sheet than on paper. A gridded mylar sheet also provides an easy means of determining coordinates of node points. Node point coordinates can be recorded in any system of units and then converted to the desired units (feet or meters) by the FESWMS-2DH programs.

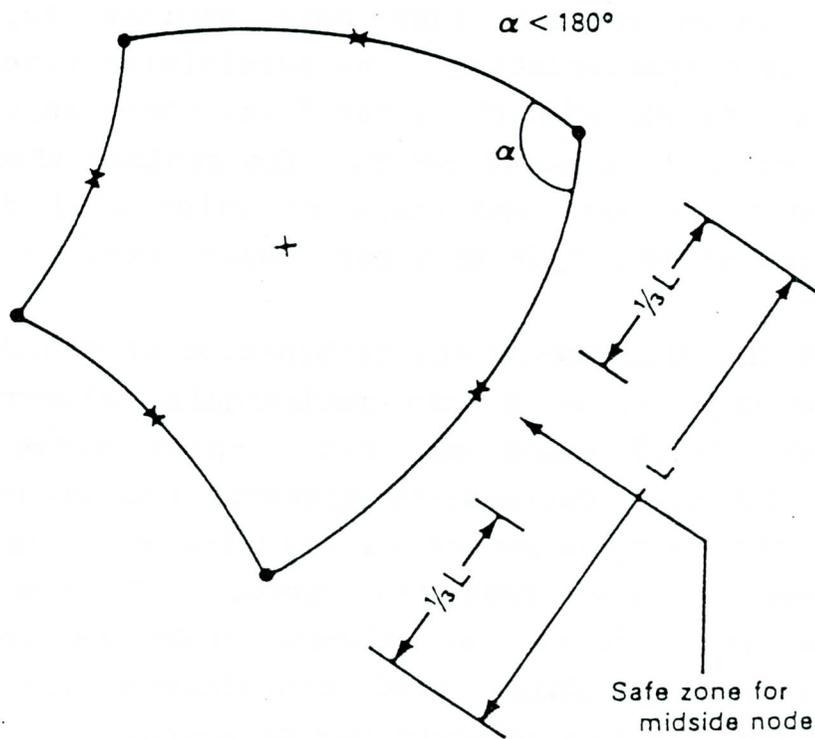
Next, the limits of the area to be modeled are defined. As a general rule, model boundaries are located where water-surface elevations and flows can be specified accurately. The effect

that errors in boundary conditions will have on a solution needs to be considered. If the accuracy of boundary conditions is not certain, the limits of a model can be placed as far away as possible from areas of primary interest so that any errors introduced at the boundaries will have little influence at the points of interest.

After boundaries have been defined, subdivision of the solution domain proceeds by dividing the area to be modeled into relatively large regions that have similar topographic and surface cover characteristics. The subdivision lines between the regions are located, as much as possible, where abrupt changes in topography or surface cover occur. The regions then are divided into elements the size and shape of which will depend on the desired level of detail in that particular area.

FESWMS-2DH will accept any combination of 6-node triangular, 8-node quadrangular, or 9-node quadrangular elements that have straight or curved sides so that complex geometries can be modeled in detail. Curve-sided elements are created simply by specifying the coordinates of the midside node as well as the corner nodes of sides that are curved. If the midside-node coordinates are omitted, an element side is assumed to be straight and the midside node coordinates are interpolated halfway between the two adjacent corner nodes.

Some conditions regarding the shape of an element need to be satisfied so that the determinant of the Jacobian matrix will not vanish within the element (that is, the isoparametric mapping between a global element and its parent element needs to be one-to-one). It is a good idea to make sure that a midside node is located within the middle third of the curved element side that it defines as shown in figure 6-1. Also, it is suggested that internal angles of all elements be kept much less than 180 degrees as shown in figure 6-1. For quadrilaterals, it is suggested that internal angles not approach zero degrees.



EXPLANATION

- CORNER NODE
- × MIDSIDE OR CENTER NODE

Figure 6-1. Some rules to insure one-to-one mapping of two-dimensional isoparametric finite elements.

A uniform network in which all the elements have about the same size and shape throughout may be easy to construct but may not always be practical. The ability to vary the size and shape of elements within a single network is a major advantage of the finite element method. In regions where the gradients of dependent variables are expected to be large, small elements will provide a more accurate solution than large elements. Locations where gradients of velocities and depth can be large include stream channels, constrictions, and areas near large inflows or outflows. Small elements also need to be used to model boundaries that have irregular shapes. In regions where the solution variables are expected to change very slowly, or in areas of the model that are of minor interest, relatively large elements may provide a solution of sufficient accuracy. The transition between a section of a network that is composed of large elements and a section of a network that is composed of much smaller elements needs to be gradual; that is, very large elements should not be connected to very small elements. Also, it is a good idea to position nodes at locations where point inflows or outflows are to be applied.

The question of which type of element to use to construct a network (that is, a 6-node triangle, an 8-node quadrangle, or a 9-node quadrangle) is not answered easily. The ease of approximating a two-dimensional region with an assemblage of arbitrary triangular elements has been demonstrated in many applications. The two kinds of quadrangular elements are similar except for the presence of an internal node in the 9-node Lagrangian element. The additional node in a 9-node quadrilateral element requires a little more computational effort, but provides a slightly more accurate solution than an 8-node quadrilateral element. For most networks, a mixture of 6-node triangular elements and 9-node quadrangular elements will provide the best representation of the water body that is being modeled.

Another characteristic of network design that affects a finite element solution is the aspect ratio of elements used in the network. The aspect ratio of a two-dimensional element is defined as the ratio of the longest element dimension to the shortest element dimension. The optimum aspect ratio for a particular element depends on the local gradients of the solution variables. If the gradients can be estimated in advance, it is best to align the longest element dimension to the direction of the smallest gradient, and to align the shortest element dimension to the direction of the largest gradient. Elements that have aspect ratios that are much greater than unity need to be designed cautiously. A well-designed network usually will be composed of elements that have a variety of shapes, sizes, and a wide range of aspect ratios.

One-Dimensional Weirs and Culverts

One-dimensional flow modeled at weirs, culverts, and small bridges is treated as either a point flow on the boundary of a finite element network when the nodes defining the structure are network boundary nodes, or as a sink/source term when the nodes defining the structure are interior nodes. A point flow is the total flow that crosses the network boundary because of flow at a single node point.

One-dimensional weirs and culverts are described by a set of parameters and two node points, one on either side of a weir or on either end of a culvert. Flow over a weir or through a culvert is computed on the basis of the water-surface elevations and velocities at the two node points, and the specified parameters. The following items need to be specified for each weir segment: (1) A discharge coefficient for free-flow conditions; (2) length of the weir segment; and (3) crest elevation of the weir segment. The following items need to be specified for each culvert: (1) A discharge coefficient; (2) cross-sectional area of the culvert barrel; (3) hydraulic

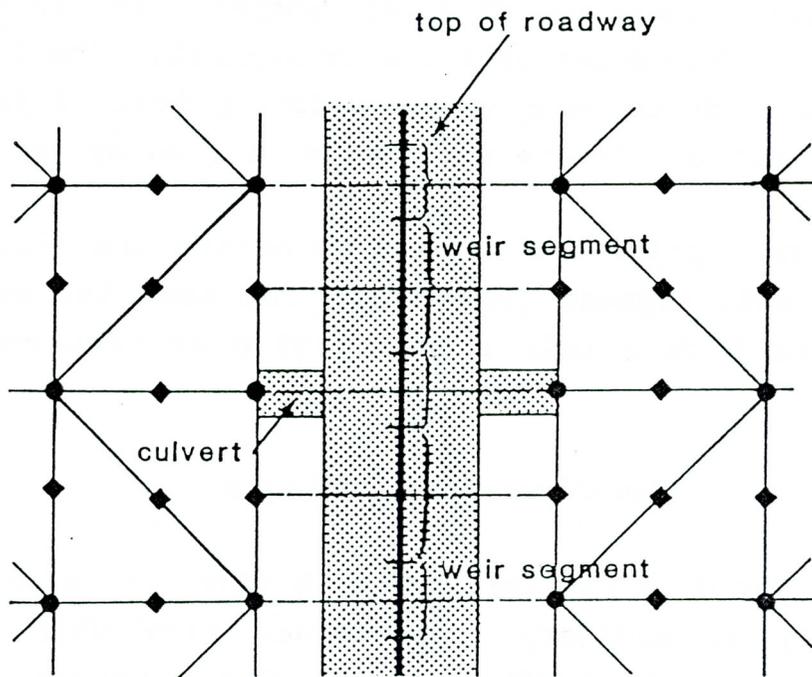
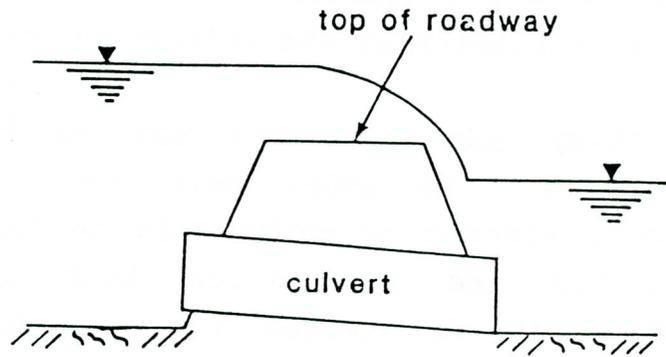
radius of the culvert barrel flowing full; (4) length of the culvert barrel; (5) Manning roughness coefficient of the culvert barrel; and (6) invert elevation at the culvert entrance.

Flow over roadway embankments is modeled best as one-dimensional weir flow. To model weir flow over roadway embankments, a finite element network needs to be designed so that solid boundaries are located on both sides of an embankment. The embankment is divided into a number of weir segments, and appropriate parameters assigned to each segment. The number of segments to use depends on the variation of the roadway elevation along the embankment and the spacing of node points on the solid boundaries that define the embankment. Node points that define the sides of a weir segment need to be located approximately at the center of the weir segment. The location of a weir segment needs to be considered during initial design of a finite element network in the vicinity of a roadway embankment.

A single node point can be used to define the side (end) of more than one weir segment (culvert). The same two node points can be used for both a weir segment and a culvert as shown in figure 6-2.

Two-Dimensional Bridges

Two-dimensional flow through a bridge or a culvert is modeled exactly as ordinary free-surface flow when the water surface is not in contact with the top of the bridge or culvert opening. However, when the water surface is in contact with the top of an opening pressure flow exists. When pressure flow conditions can occur at a bridge or culvert, special consideration needs to be given to the design of a finite element network in the vicinity of the structure.



- Corner node
- ◆ Midside node

Figure 6-2. A finite element network at a roadway embankment that contains a culvert and is divided into weir segments

If pressure flow within a bridge opening is to be modeled, elements need to be constructed to conform to the two-dimensional plan of the bridge deck as shown in figure 6-3. The elevation of the ceiling (that is, the underside of the bridge deck) also needs to be specified for each of the corner nodes contained in elements that conform to the bridge deck. More than two rows of elements within an opening may be needed to model pressure flow accurately. Increased resistance to flow caused by shear along the underside of the bridge deck is included when pressure flow occurs.

Combined pressure flow through a bridge opening and weir flow over the bridge deck can be modeled by specifying weir segments that define the top of the bridge deck. The bridge opening can be either completely or partially submerged. The two nodes on either side of the weir segment will be the nodes on the upstream and downstream sides of the bridge that correspond to the location of the weir segment. These nodes will always be internal nodes, except at the point where the bridge deck intersects a network boundary. However, weir flow over a bridge deck is treated as a source/sink term in the continuity equation, even when the upstream and downstream nodes are boundary nodes.

Model Calibration

A finite element model is a simplified, discrete representation of a complex and continuous physical flow system. Three-dimensional topographic features are represented by two-dimensional elements and the physics of flow are assumed to obey differential equations in which several empirical coefficients appear. As soon as a model produces useful results it needs to be calibrated if enough data are available. Model calibration is the process of adjusting the dimensions of simplified geometric elements and empirical hydraulic coefficients so that values computed by a model reproduce as closely as possible values measured on site.

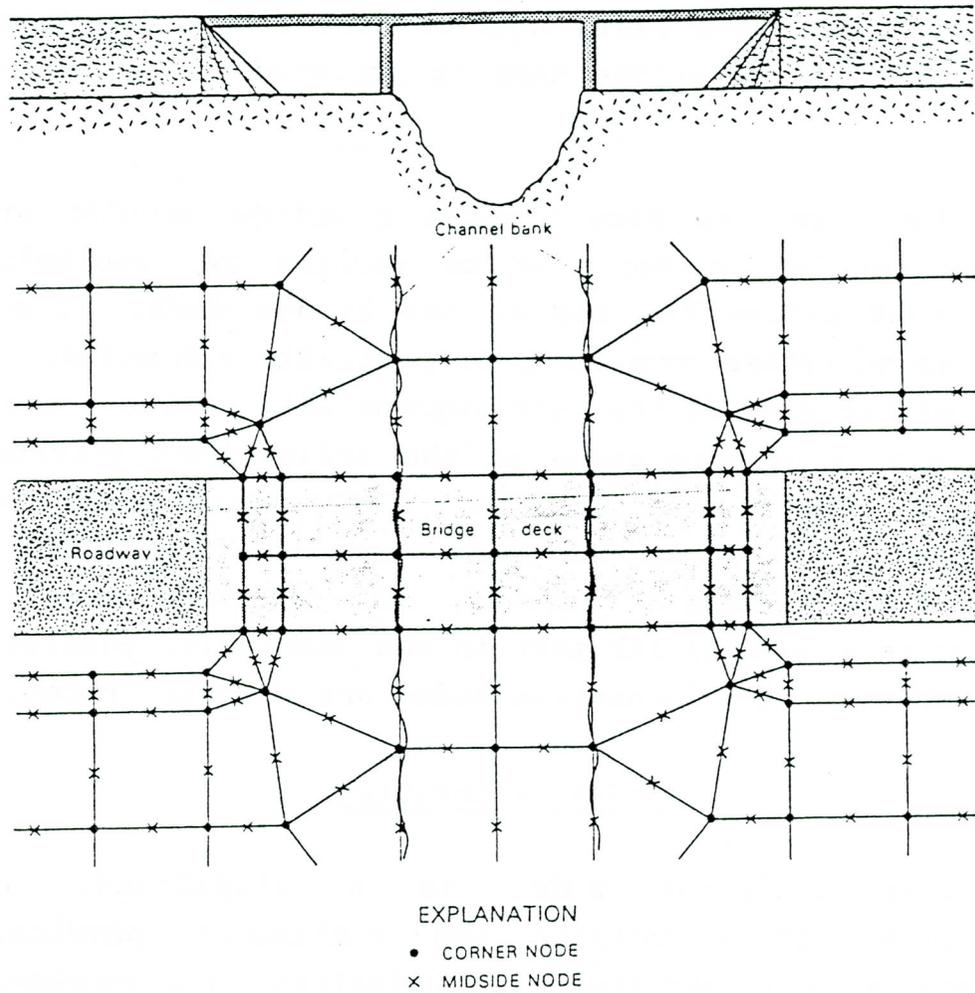


Figure 6-3. A finite element network at a bridge where pressure flow within the bridge opening is modeled.

The ability of a model to reproduce and predict measured values depends on the amount and quality of topographic, topology, and hydraulic data that have been collected. Although model parameters can be adjusted to obtain close agreement between computed and measured values, an adjustment may not be extended beyond physically reasonable values. For example, if good agreement can be obtained only by using Manning roughness coefficients three times as large as estimated initially, the finite element network probably is a poor representation of the physical region being modeled. The purpose of model calibration is to obtain an accurate mathematical representation of reality, not a force-fitting of a poorly constructed model.

Model calibration proceeds by adjusting parameters in a systematic fashion so that computed and measured values agree as closely as possible. Measured values of water-surface elevation, total flow rates, and velocities can be used to calibrate a model. An impression of the sensitivity of computed values to changes in model data can be obtained from initial adjustment runs. Sensitivity of results to changes in model data can indicate a need to measure more accurately those parameters for which small changes have a significant affect on model output.

Roughness (or discharge) coefficients are empirical coefficients that have the greatest effect on a solution. Roughness coefficients estimated during the initial design of a network will not have to be adjusted much if sufficient and accurate topographic data have been collected. Changes to roughness coefficients need to be made carefully so that adjusted values are appropriate for the bed material, channel slope, and vegetative cover that exist in the area covered by a particular element. For example, two channel reaches that have about the same bed material and cross section shape, or two flood plains that have about the same vegetative cover and topographic characteristics, also need to be assigned roughness coefficients that are about equal.

Eddy viscosity coefficients usually affect a solution much less than roughness coefficients. The largest influence of eddy viscosity occurs where velocity gradients are large. Increasing eddy viscosity coefficients will cause velocity gradients to be reduced, and the horizontal velocity distribution will become more uniform. Reducing eddy viscosity coefficients will cause velocity gradients to increase.

If close agreement between measured and computed water-surface elevations, flow rates, and velocities cannot be obtained using roughness and eddy viscosity coefficients that are within reasonable ranges, then model discretization and the accuracy of topographic and hydraulic data need to be examined.

If a model has been thoroughly calibrated and is still not capable of reproducing measured values satisfactorily, one or more of the following kinds of problems may exist:

- The time step or element sizes may be too large to resolve short wave components in unsteady flow simulations. The time step needs to be made small enough to model accurately time-dependent boundary conditions. The only definite way to determine whether or not the time step is too large is to simulate the same event using a successively smaller time step. If the size of the time step significantly affects computed values, it is too large and needs to be reduced.
- The data measurement techniques or frequency of observations may be inaccurate. Errors may be caused by inaccurate leveling, erroneous high-water marks, or faulty gauges.
- Measurements or estimates of tributary inflow may be needed.
- Phenomena that affect the flow significantly are not accounted for in the model. Possibly surface wind stresses, bed variation caused by erosion or sedimentation, or seasonal variation of roughness resulting from changes in vegetative cover need to be considered.

Model Validation

Model validation is subsequent testing of a calibrated model to see if computed values compare favorably to measured values that were not used to calibrate the model. If a model reproduces closely the additional measured values without any further adjustment of model parameters or redesign of the finite element network, the model can be used to simulate confidently conditions outside the range of calibration. Often it is impossible to validate a model because of insufficient data or because of time or funding constraints.

Model Application

After the preliminary steps have been completed, a model can be used to simulate a variety of flow conditions. A model still needs to be applied with care, especially if it is used to evaluate conditions far outside the range of calibration and validation. If a model has been calibrated and validated properly, it can be used to gain valuable insights to the response of a surface-water flow system to natural or manmade changes.

This page is blank.

Section 7

PROGRAM LOGIC AND DATA FLOW

This section describes the logical flow of data through the modeling system from the entry of input data to the generation of output data. The major software features of each system module and the flow of data through each module are illustrated in schematic diagrams. Features of the modeling system that require explanation are described in detail.

Data Input Module: DINMOD

DINMOD is not a simulation program, it is an input data preprocessor for the modeling system. The primary purpose of DINMOD is to assist a user in developing a finite element network that is free from errors. Specific functions performed by DINMOD include the following:

- Reads all data needed to define a finite element network. Input data are read from data records and, optionally, from a previously generated network data file.
- Checks all input data for compatibility with array dimensions; and, optionally, checks for strict geometric consistency and completeness, which is useful when a new network is being developed or when extensive revisions are being made to an existing network.
- Interpolates node point coordinates along straight-line segments of a finite element network.
- Generates automatically all or part of a finite element network including element connectivity lists and node point locations.
- Subdivides all elements in a network into four similar elements. This feature allows rapid refinement of an entire finite element network.

- Determines an element assembly sequence that will result in an efficient frontal solution of a system of finite element equations.
- Writes processed network data to a file which can be read by other FESWMS-2DH programs.
- Creates graphic output files that can be processed to display finite element network and ground-surface elevation contour plots.

DINMOD can be used to develop a new finite element network; or to refine, update, or modify an existing network. Review of a completed network is advised even though no errors may have been detected by the program checks. A visual inspection of printed program output for all but the smallest of networks is not adequate to insure an error-free network. Examination of graphic output from DINMOD is the best way to make a final check of a network.

The logical flow of data through DINMOD from the entry of input data to the generation of a network data file, and the major software features of the module are illustrated in figure 7-1. DINMOD begins by reading job control data from data records. An existing network data file is read next if requested. Remaining data records then are read. Any network data read from data records will override data previously read from an existing network data file.

Next, coordinates along straight-line segments of the network are interpolated between two specified end points. Either the y coordinate or the x coordinate or both coordinates can be computed. If only the y coordinate is to be determined, the x coordinate of each node needs to have been specified. If the only x coordinate is to be determined, the y coordinate of each node point needs to have been specified. If both coordinates are to be computed, nodes are assumed to be equally spaced between the two end points.

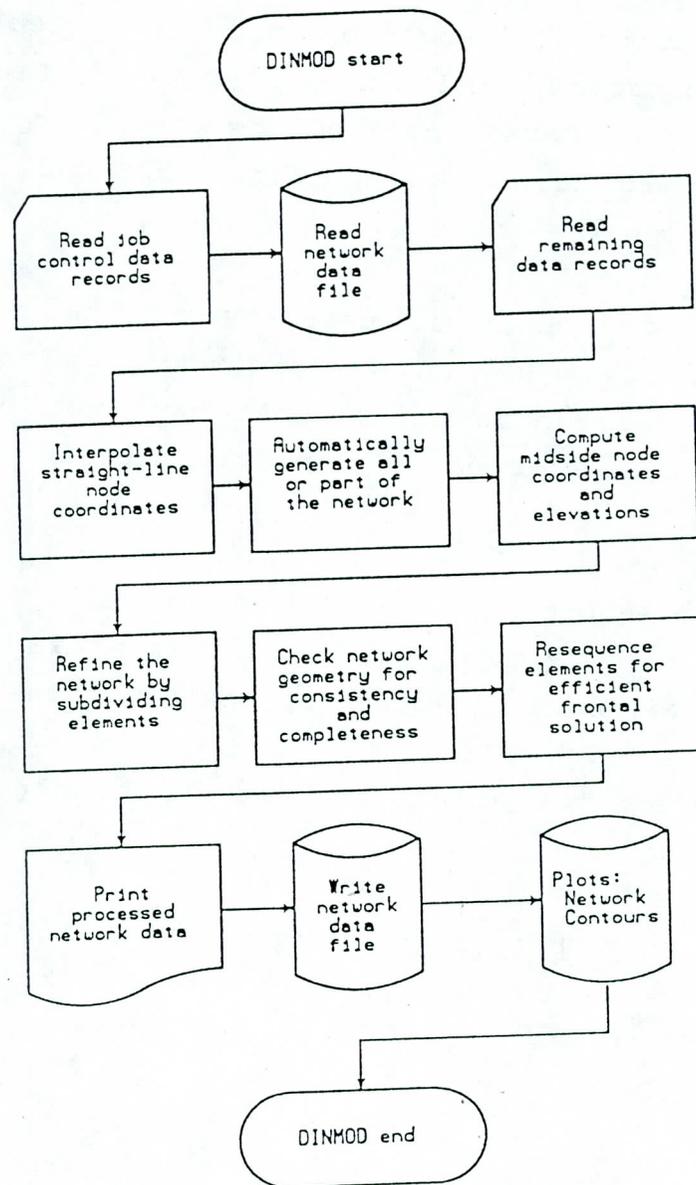


Figure 7-1. Schematic diagram of the Data Input Module (DINMOD) showing the logical flow of data through the module and the major functions that are performed.

Automatic grid generation is performed next if requested. All or part of the network can be filled in by 6-node triangular elements for which connectivity lists and node point coordinates will be computed. The area inside which a network is to be generated needs to be divided into simply connected regions that have similar topography and surface cover. Each region is defined by a list of corner node points that form its boundary. Coordinates need to have been specified for the boundary node points.

At this point, the basic geometry of the finite element network has been defined. The coordinates and ground-surface elevations of midside node points that have not been specified previously are then interpolated linearly between their adjacent corner nodes.

Element resequencing is performed next. The purpose of resequencing is to develop an efficient element assembly sequence that is used in a frontal solution of the equation system that is created by FLOMOD. An efficient assembly sequence will minimize the core memory and the amount of computer time needed to solve the system of equations.

The completely processed network data including element assembly sequences are then printed in report form, and the network data is written to a network data file that can be read by other FESWMS-2DH programs.

The last major function performed by DINMOD is creation of a graphic output file that can include plots of the finite element network and contour plots of ground-surface elevations. A careful examination of all plotted network data is advised before a network is considered to be free from errors.

Error Checking

Numerous error checks are contained in DINMOD to assist a user in developing or modifying a finite element network. Specifically, DINMOD does the following:

- Checks all node, element, element-sequence, and property-type numbers or compatibility with the appropriate array dimensions and other program limits.
- Checks nodal coordinates to make sure that they are within an allowable range.
- Checks each corner node to make sure that its coordinates have been specified.
- Compares the coordinates of every node in an element to the coordinates of every other node in that element to determine if two nodes are located at the same point.
- Checks consistency of element sides common to two elements.
- Checks each node to make sure that it is used only as a corner node, a midside node, or a center node.
- Checks values of the Jacobian determinant at numerical integration points to make sure that the determinant does not vanish.

Network data cannot be considered free of errors until a network plot and a contour plot of ground-surface elevations are inspected carefully.

Automatic Grid Generation

Automatic generation of all or part of a finite element network is begun by subdividing the area or areas for which elements are to be generated into one or more subareas of relatively simple shape. A second-level subdivision is then imposed on each of the initial regions to create an orderly assemblage of elements and node points. DINMOD uses a

triangulation technique in combination with a final smoothing procedure to generate 6-node triangular elements during the second-level subdivision.

Initial subdivisions typically define areas of similar topography and surface cover where solution gradients (that is, the spatial rates of change of water-surface elevation and velocities) are relatively small. An initial region is described by a list of corner nodes that form its boundary. The corner nodes are recorded starting at any corner node and proceeding around the boundary in a counterclockwise direction.

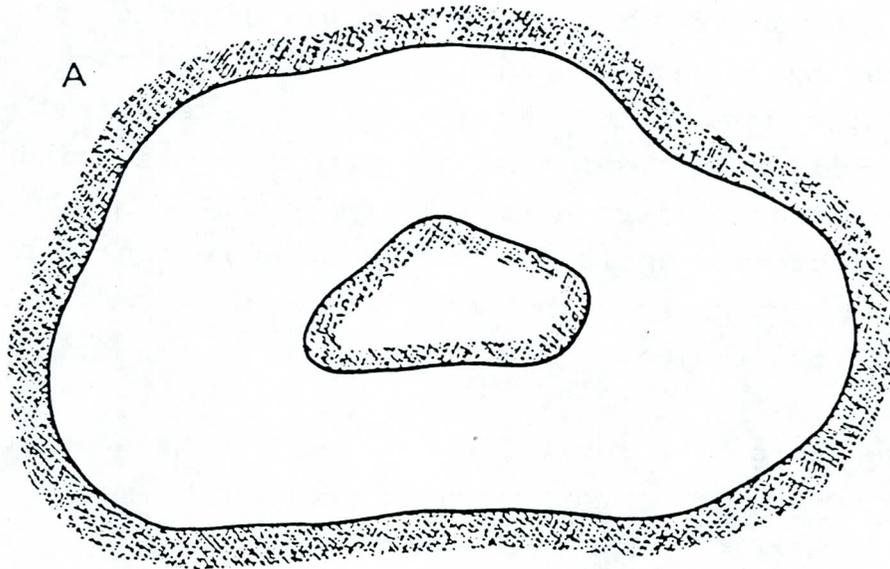
An initial region needs to be simply connected (that is, the entire boundary needs to be formed by a continuous line). If a grid is to be generated automatically inside a region such as the one shown in figure 7-2(A), the area needs to be defined by more than one initial subdivision as shown in figure 7-2(B).

The polygon formed by the list of corner node points that define an initial subdivision is divided into 6-node triangular elements. Elements are formed by cutting off sharp corners of the polygon and replacing selected node points on the boundary of the polygon by new node points inside the region.

Automatic triangulation begins by removing each vertex (corner node) of the polygon that has an internal angle less than 90 degrees by connecting the two adjacent corner nodes to form a triangle. Then, starting at any vertex that has an internal angle less than 180 degrees, two new triangular elements are created by adding a corner node to the interior of the polygon. The location of the new corner node is computed using the coordinates of the two corner nodes adjacent to the vertex. The x and y coordinates of the new node point are computed as follows:

$$x_4 = \frac{1}{2}(x_1 + x_3) + \omega(Y_1 - Y_4) \quad (7-1a)$$

$$Y_4 = \frac{1}{2}(Y_1 + Y_3) + \omega(x_3 - x_1) , \quad (7-1b)$$



EXPLANATION

● CORNER NODE

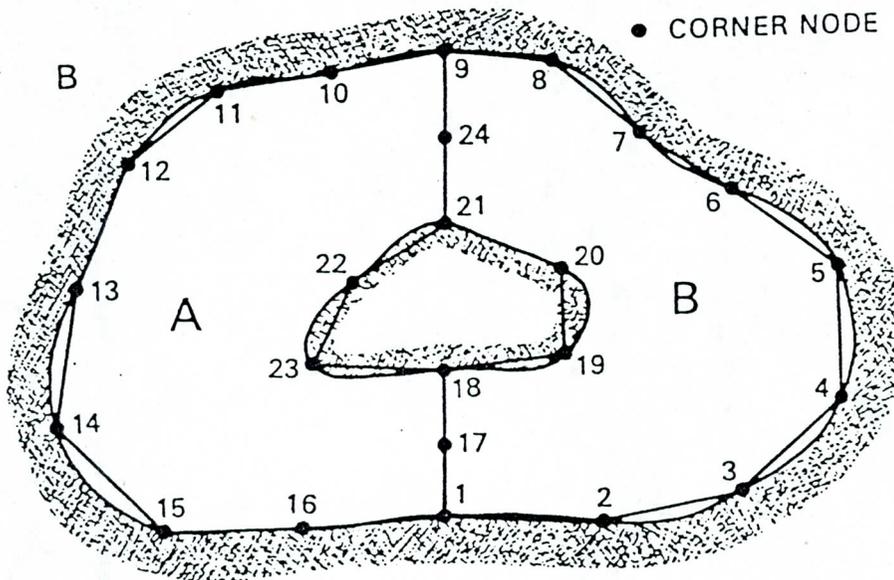


Figure 7-2. Illustration of (A) a region inside which a finite element network is to be generated automatically, and (B) a simply connected initial subdivision of the region.

where the subscripts refer to the numbered node points as shown in figure 7-3, and ω is a weighting factor. The default value of ω used in DINMOD is 1/3. Different values of ω will generate networks having different geometries. If a vertex that has an internal angle less than 90 degrees is created, that vertex is removed immediately from the polygon by connecting the two adjacent vertices to form a new element. New elements are added using this procedure until only three nodes remain in the polygon list. The last three nodes define the last element that is generated in the region.

Because there is a possibility of generating some overlapping elements that would cause computational problems, a smoothing procedure is used to adjust the shape of the elements created by the triangulation process. The smoothing procedure used is the Laplacian scheme described by Buell and Bush (1973). The Laplacian scheme requires the coordinates of newly created node points to satisfy the equations

$$x = \frac{1}{2L_i} \sum_{k=1}^{L_i} (x_{kj} + x_{kl}) \quad (7-2a)$$

and

$$y = \frac{1}{2L_i} \sum_{k=1}^{L_i} (y_{kj} + y_{kl}) , \quad (7-2b)$$

where L_i is the number of elements connected to node i ; and (x_{kj}, y_{kj}) and (x_{kl}, y_{kl}) are the coordinates of nodes in neighboring element k as shown in figure 7-4. Because equations 7-2 are nonlinear, they are solved using an indirect iterative technique. Convergence usually is achieved within five iterations.

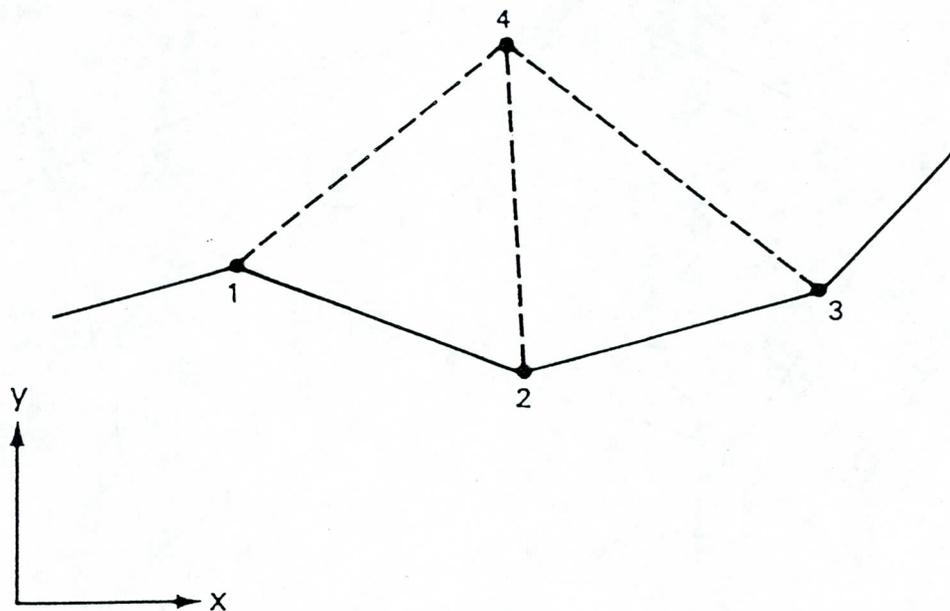


Figure 7-3. Illustration of two triangular elements that are created by adding a corner node (node 4) the location of which is based on the coordinates of the two corner nodes (nodes 1 and 3) adjacent to the vertex.

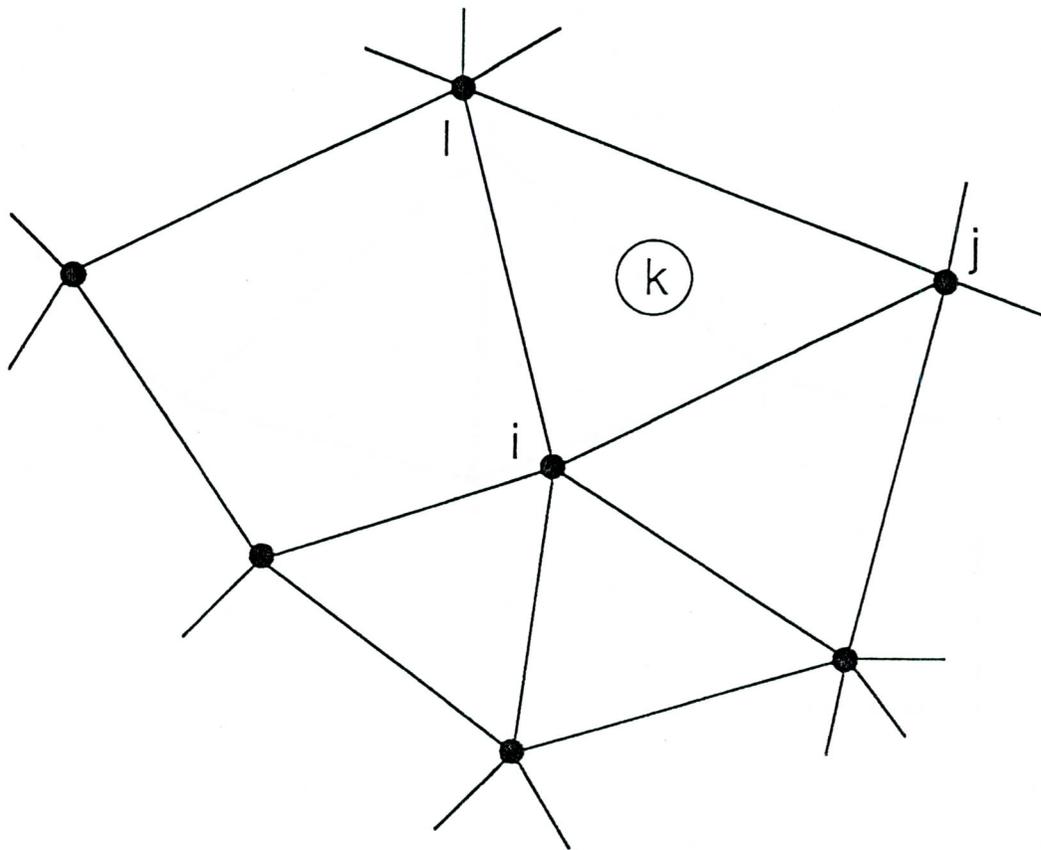


Figure 7-4. Neighborhood of node i in element k used in Laplacian smoothing of a network that has been generated automatically.

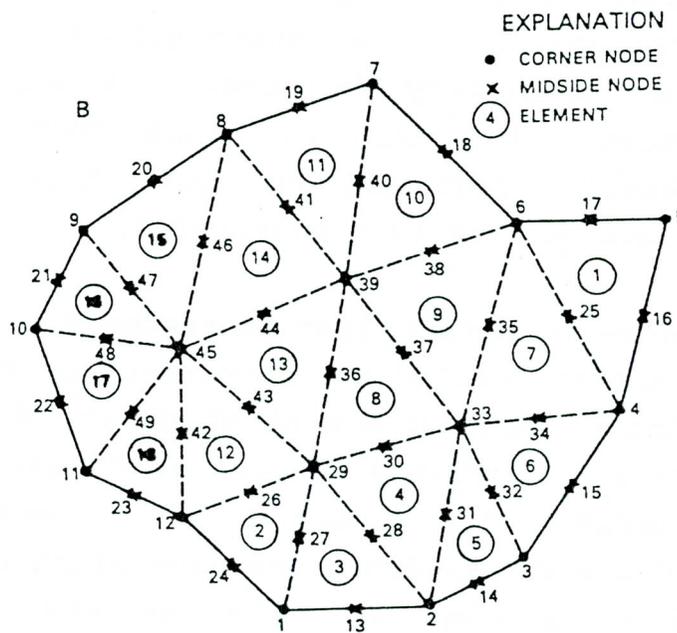
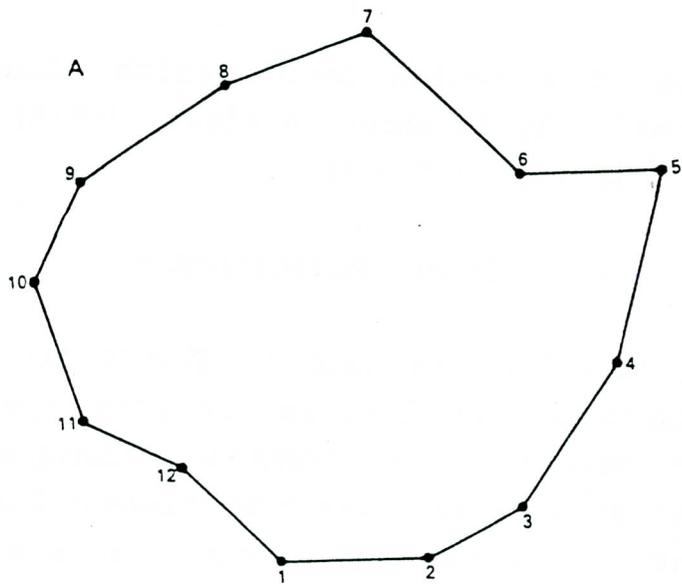
Element connectivity lists and the coordinates of newly created node points are computed automatically. Ground-surface elevations of the corner node points need to be entered manually.

An example of a region inside which elements are to be generated automatically is shown in figure 7-5(A). The generated network is shown in figure 7-5(B)

Element Resequencing

A frontal technique is used in FLOMOD to obtain a direct solution of the equations that result from application of the finite element method. The frontal technique assembles and reduces the equations on an element-by-element basis. As soon as the coefficients of a particular equation are assembled, the completed equation can be eliminated and stored out-of-core. During a frontal solution, the entire global coefficient matrix is never formed completely in core. At any given instant, equations contained in core are those that are either not complete (partially assembled) or those that have just been completed but have not yet been eliminated.

The degrees-of-freedom that correspond to the equations contained in core are called the wavefront, or simply the front, because the line of node points at which the degrees-of-freedom are located moves through the network as a wave as elements are processed in order. The number of degrees-of-freedom in the front is called the frontwidth. The frontwidth will vary in size during equation solution and the maximum frontwidth will determine how much core memory is required. The sum of frontwidth-squared as each equation is eliminated is proportional to the number of arithmetic operations performed during the solution. The sequence in which the elements are assembled determines the maximum frontwidth and the sum of frontwidth-squared, and thus determines the core memory requirements and the computer time needed to solve a system of equations. An element



EXPLANATION

- CORNER NODE
- × MIDSIDE NODE
- (4) ELEMENT

Figure 7-5. An example of a finite element network that has been generated automatically: (A) An initial subdivision that is defined by a series of connected corner nodes; and (B) the network that was generated inside the initial subdivision.

assembly sequence that keeps the maximum frontwidth and the sum of frontwidths-squared to a minimum also will keep core memory requirements and equation solution time to a minimum.

For small networks, a manual determination of an optimal element assembly sequence is possible, but for large networks the task quickly becomes tedious and uneconomical to perform by hand. Two methods are available in DINMOD to develop an efficient element assembly sequence automatically: (1) The *minimum frontgrowth method*, and (2) the *level structure method*. Because it is virtually impossible to investigate all the combinations of element sequences, the minimum frontgrowth method and the level structure method provide good, but not necessarily the best, assembly sequences.

The two element resequencing methods are based on different strategies, but both methods require an initial list of elements (at least one) to begin a resequencing. From a starting element list, assembly sequences are determined for the remaining elements in a network. For both methods, several different starting element lists probably need to be tried to find a good element assembly sequence. An initial starting list that usually will provide a good assembly sequence consists of all or just some of the elements that form the most narrow edge of a network.

Minimum frontgrowth method

The minimum frontgrowth method maintains the smallest possible frontwidth at all times. An initial wavefront is determined from the starting element list and is defined in terms of nodes rather than degrees-of-freedom. The nodes that form the wavefront form a boundary between elements that have been assembled and elements that have not been assembled. A list of unassembled elements that border the front is called the adjacent element list. The element in the adjacent element list that provides the smallest frontwidth upon its assembly is chosen to be the next element assembled. If more than one element provides

the same minimum frontwidth, various tie-breaking strategies are used to choose between them. After an element is assembled, the wavefront is modified and the adjacent element list is updated. This process continues until all elements have been resequenced.

Sometimes an element in an adjacent element list is passed over for assembly a great number of times, resulting in excessively large frontwidths. To avoid this occurrence, the length-of-stay of an element in an adjacent element list is limited. An appropriate value for the maximum length-of-stay needs to be determined by trial-and-error, but a value equal to about twice the expected maximum frontwidth (in terms of nodes) will be a good first try. The maximum frontwidth can be estimated as the number of nodes in a line that extends across the widest part of a network when the network is aligned lengthwise.

Level structure method

The level structure method uses a simple layer-by-layer resequencing strategy and is much faster than the minimum-frontwidth method, especially for large networks. From a starting element list, a wavefront and a list of elements adjacent to the wavefront are determined. Then, the first element in the adjacent element list is assembled and the unassembled elements adjacent to it are added to the adjacent element list. The adjacent element list is updated and the process is repeated until all elements have been assembled.

Network Refinement

The level of discretization that is needed in a finite element network to provide a solution of a desired accuracy is not always obvious. If this is the case, one way to proceed is to develop an initial network using the largest (fewest) possible elements. If the solution indicates that smaller (more) elements are needed to provide the desired level of accuracy and detail,

each element in the network can be divided quickly into smaller elements using the network refinement feature in DINMOD. A network is refined by dividing all the elements of the network into four similar elements as shown in figure 7-6. Elements that have curved sides are transformed into four similar curve-sided elements.

New element connectivity lists and node point data are generated automatically. However, a new element assembly sequence will have to be developed using the element resequencing capability of DINMOD.

Depth-averaged Flow Module: FLOMOD

The primary function of FLOMOD is to solve the equations that govern steady and unsteady (time-dependent) two-dimensional surface-water flow in a horizontal plane. Specifically, FLOMOD does the following:

- Inputs geometric, initial, boundary, wind, and element property data.
- Uses either U.S. customary (inch-pound) units or International System (metric) units.
- Checks input data for compatibility with array dimensions.
- Computes flow across specified cross sections.
- Computes element continuity norms.
- Adjusts the network boundary automatically.
- Prints solution results at selected iterations or time steps.
- Creates an output (flow) data file.

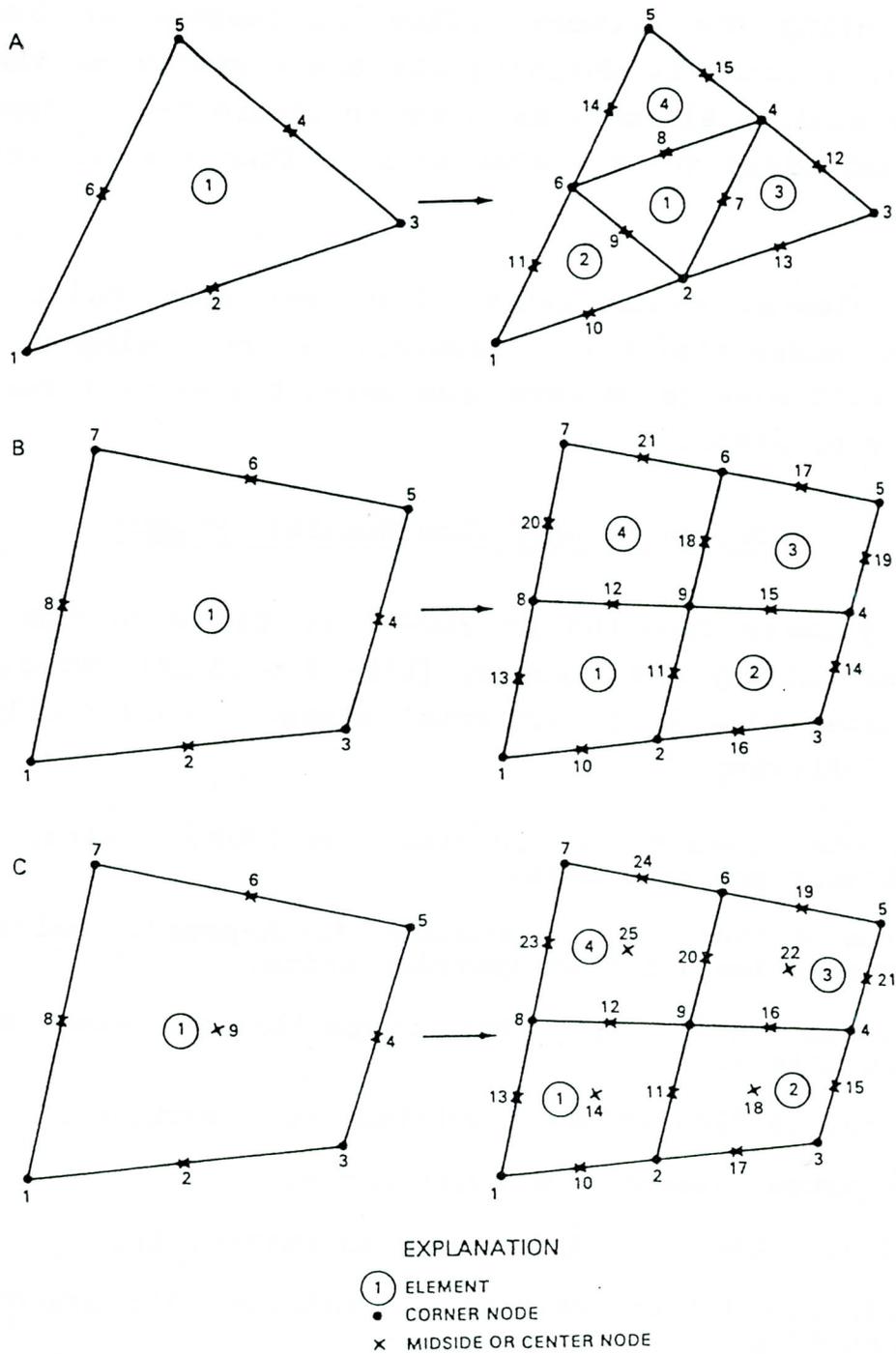


Figure 7-6. Elements are refined by dividing them into four similar elements: (A) A 6-node triangular element; (B) an 8-node quadrangular element; and (C) a 9-node quadrangular element.

Error Checking

To insure that input data are free from errors, FLOMOD does the following:

- Checks all node, element, element-sequence, and property type numbers for compatibility with the appropriate array dimensions and other program limits.
- Checks values of empirical coefficients to make sure they are within an allowable range.
- Checks specified water-surface elevations at boundaries to make sure that positive water depths result.
- Checks ceiling elevations to make sure they are greater than the ground-surface elevation at the node point.
- Checks weir and culvert nodes to make sure they are boundary nodes.
- Checks the maximum frontwidth and the maximum number of equations for consistency with array dimensions and other program limits.
- Checks equations coefficients that are used as pivotal elements to make sure they are non-zero.

The logical flow of data through FLOMOD from the entry of input data to the generation of output data and the major functions of the program are illustrated in figure 7-7.

FLOMOD begins by reading job control data records. Network data, initial condition data, boundary condition data, and wind data files are read next if requested. Any remaining input data records for an initial solution then are read. Any data entered on data records will override information that was previously read from data files.

Preliminary computations and an initial flow check are performed next. A degree-of-freedom array that relates every

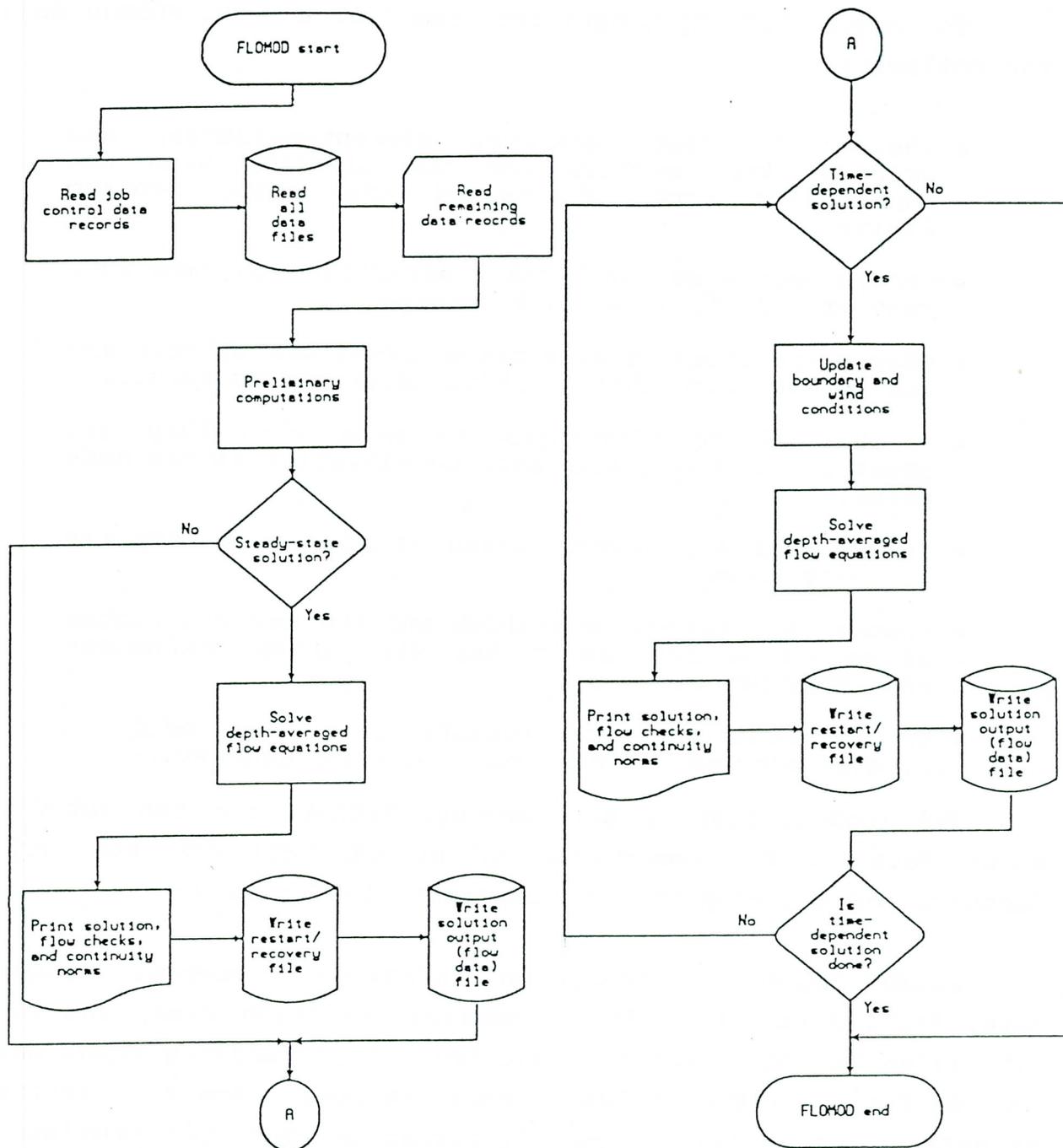


Figure 7-7. Schematic diagram of the Depth-Averaged Flow Module (FLOMOD) showing the logical flow of data through the module and the major functions that are performed.

nodal variable to an equation number is constructed, and solution parameters (number of equations, and maximum and root-mean-square frontwidth) are computed.

A steady-state solution is performed next. The intermediate solution results are written to a restart/recovery file (if requested) after every iteration. If a run terminates abnormally, the run can be restarted using the intermediate results stored in a restart/recovery file. The computed flow data (depth-averaged velocities and the flow depth at node points) can be printed after every iteration or at the completion of selected iterations. The results of the final iteration of a steady-state solution are written to a flow data file. A flow check is made and element continuity norms are computed.

A time-dependent solution is performed next. Preparations are made and boundary conditions are updated to the current time. Time-dependent data can be read from data records or from data files. If time-dependent data records are read, they need to be arranged in chronological order at the end of the input stream. Results can be printed at the end of every iteration or only at the end of selected iterations for every time step or only for selected time steps. Results are written to a flow data file at the end of each time step. Intermediate results are written to a restart/recovery file (if requested) at the end of every iteration. A flow check is made and element continuity norms are computed at the end of every time step.

Equation Solution

The depth-averaged flow equations are, in their complete form, a coupled system of nonlinear partial-differential equations. The many alternative methods for solving the system of nonlinear algebraic equations that results from a finite element discretization of the governing partial-differential equations present such a wide choice that it is difficult to know

which method is best. A solution method that works well for one problem may not work at all for another problem.

Solving the nonlinear equation system is the most costly aspect of a finite element solution. Computational efficiency in terms of both time and storage space dictates that a symmetric equation system be solved if possible. However, the coefficient matrix that is formed is nonsymmetric because of nonlinear inertia and bottom friction terms that appear in the governing equations.

A finite element solution of the governing equations produces a set of global discretized equations in the form

$$K(a)a = f , \quad (7-3)$$

where K is a matrix of assembled element coefficients; a is a vector of unknown nodal values; and f is a global load vector. The simultaneous nonlinear system of equations is solved using a strategy that combines full-Newton iteration, quasi-Newton iteration, and a frontal solution scheme.

Solution strategy

Full-Newton iteration is written as

$$a_{i+1} = a_i - J(a_i)^{-1}R(a_i) , \quad (7-4)$$

where a_{i+1} is a newly computed solution vector; a_i is a known solution vector at the i th iteration; $J(a_i)$ is a Jacobian (tangent) matrix computed from a_i ; and $R(a_i) = K(a_i)a_i - f$ is a residual load vector. In practice, the iteration is performed as

$$J(a_i)\Delta a = -R(a_i) , \quad (7-5)$$

where Δa_i is the incremental solution value. An updated solution is computed as

$$a_{i+1} = a_i + \omega_r \Delta a_i, \quad (7-6)$$

where ω_r is a relaxation factor that is allowed to range from 0.0 to 2.0 (the default value is 1.0). A solution usually converges rapidly when the initial estimate is close to the true solution.

During a full-Newton solution the Jacobian matrix is decomposed into lower and upper triangular matrices. These factorized matrices can be updated in a relatively simple manner rather than being recomputed completely at each iteration. Broyden's update procedure in inverse form (Engleman and others, 1981) is used to update the factored Jacobian matrix. This quasi-Newton method can reduce significantly the computation time needed to obtain a solution.

Given an initial solution estimate, a , the LU factorization of its Jacobian, J , and the initial search direction, W_a , the quasi-Newton algorithm proceeds as follows:

```

For i = 1 to i
  1. Form  $\delta_i = \Delta a_{i-1}$ 
  2. Compute  $\Delta a_i = -J_o^{-1} R(a_i)$ 
  3. For j = 1 to i-1
      Compute  $\Delta a_i = \Delta a_i - \rho_j (\delta_j + r_j) \delta_i' \Delta a_i$ 
  Next j
  4. Form  $r_i = \Delta a_i - \delta_i$ 
       $\rho = \frac{1}{\delta_i' r_i}$ 
  5. Compute  $\Delta a_i = \Delta a_i - \rho_i (\delta_i + r_i) \delta_i' \Delta a_i$ 
Next i

```

where δ' is the transpose of the vector δ .

Each iteration requires solution of a single linear system for which the triangular factors of the coefficient matrix are already known, plus the vector operations that are needed to update the matrix. Two updating vectors (w and r) are created at each iteration and are kept and used again in subsequent iterations, up to a limit imposed by the user. When the upper limit is reached, the updating vectors are shifted one position downward (thus losing the first pair) and computations continue. If the limiting number of updates is set to zero, the coefficient matrix is not updated and modified-Newton iteration results.

FLOMOD provides a choice of solution strategies. A typical solution will combine both full-Newton and quasi-Newton iterations in an attempt to achieve the fastest solution possible. Usually, at least two or three full-Newton iterations are required when starting cold (that is, when initial velocities are set to zero and a constant water-surface elevation is assigned), or after making substantial changes to boundary conditions or the geometry of a finite element network. Initial iterations can be followed by one or more quasi-Newton iterations, or by a combination of full-Newton and quasi-Newton iterations.

The optimal number of update vectors to use in a quasi-Newton iteration will vary from solution to solution. Beyond a limit, the updating procedure becomes uneconomical. Maintaining more than about five sets of update vectors in memory has been found to result in wasted computational effort. The number of update vectors used in FLOMOD is limited to a maximum of five.

Frontal solution scheme

The frontal solution technique is a direct solution scheme that is closely related to the finite element method. The solution scheme is designed to minimize core-storage requirements as well as the number of arithmetic operations needed to solve a system of linear algebraic equations. The main idea of the frontal method is to assemble and eliminate element equations at

the same time. As soon as an equation is formed completely from the contributions of all relevant elements, it is reduced and eliminated from the "active" coefficient matrix. An equation that has been eliminated from the active coefficient matrix is written to a buffer contained in core memory. When the buffer is full, it is written to an auxiliary storage device. A coefficient matrix usually is never formed in its entirety. An active matrix contains at any given instant only those equations that have been partly assembled or are complete but not yet eliminated.

The number of unknowns in a wavefront at any given time is called the frontwidth and generally will change continually during a solution. The maximum frontwidth reached during a solution determines the required maximum size of an active coefficient matrix. When element assembly is complete, the upper triangular matrix of the LU decomposition will have been formed and will be ready for back substitution.

A modified version of the frontal solution scheme presented by Hood (1976, 1977) is used in FLOMOD. Modifications were made to eliminate unnecessary computations and to save both the upper and lower triangular matrix decompositions if a quasi-Newton solution is to be performed. Also, eliminated equations are stored in a buffer (the size of which depends on available computer storage and storage-device limitations), which is written to an off-line storage device when full or nearly full. Data transfer time decreases as the size of the equation buffer is increased. A diagonal-pivoting strategy is used whereby equations that are complete in the active coefficient matrix and ready for elimination are scanned, and the equation that has the largest diagonal element is eliminated next. A minimum number of completed equations can be maintained in an active coefficient matrix, thus ensuring a choice of pivotal coefficients.

The frontal solution algorithm contained in FLOMOD has been tested on small to extremely large problems and has been proven quite successful in all cases.

Continuity Norm

A potential problem caused by the use of mixed interpolation (that is, linear interpolation of depth and quadratic interpolation of velocities) is that mass conservation is not well enforced because the ratio of discrete continuity constraints to discrete momentum equations is much smaller than the continuum ratio of 0.5.

Computing the mass flux at model cross sections in steady-state simulations is one method for determining whether mass conservation errors are within acceptable limits. At cross sections where the mass flux differs substantially from the inflow, the finite element network can be refined to reduce the errors. An even better way to determine which parts of a network can be refined to improve mass conservation is to compute a continuity norm for each element in the network.

Letting R denote the continuity equation residual,

$$R_H = \frac{\partial H}{\partial t} + H \frac{\partial U}{\partial x} + U \frac{\partial H}{\partial x} + H \frac{\partial V}{\partial y} + V \frac{\partial H}{\partial y}, \quad (7-7)$$

the continuity norm for an element, N , is computed as

$$N_e = \frac{1}{A_e} \left(\int_{A_e} R_H^2 dA_e \right)^{1/2}, \quad (7-8)$$

where A_e is the area of the element. A continuity norm will be large for an element in which continuity equation residuals are large. Computation of continuity norms is optional. Continuity norms greater than a specified value are flagged with an asterisk in the printed output so that elements that need to be refined to obtain a more accurate solution can be identified easily.

Automatic Boundary Adjustment

An automatic boundary adjustment feature in FLOMOD allows elements that are not covered completely by water to exist in a finite element network. If the boundary is not allowed to adjust automatically then all nodes in a network need to be "wet" at all times (that is, water depth always needs to be greater than zero at a node point) otherwise computational problems will result when a node becomes "dry" (that is, when water depth becomes less than or equal to zero). A conceptually simple algorithm is used to determine automatically the boundary of a finite element network so that no dry nodes exist. The procedure excludes from an "active" network all elements that are connected to one or more dry node points.

To explain how the algorithm determines whether or not an element is to be included in an active network, several terms need to be defined. An element is said to be "on" if it is included in the active network, and is said to be "off" if it is not included. A "dry" element is an element that is connected to at least one node point that is dry. A "wet" element is an element in which all node points are wet.

At the beginning of each iteration, every element that is currently on is checked to find out if it is dry. If found to be dry, the element is turned off. In addition, each element that is currently off is checked to determine if it can be turned on. The decision to turn on an element is based on the minimum flow depth and maximum ground-surface elevation at the node points connected to the element. If the minimum water-surface elevation is greater than the maximum ground-surface elevation, plus a small depth tolerance, the element is turned on. The need for a depth tolerance is twofold. First, because of energy losses, there probably will be some change in the water-surface elevation across an element when it is turned on. Second, the condition of an element (wet or dry) can oscillate from one iteration to the

next, and cause the solution to converge slowly or not at all. A depth tolerance of 0.5 feet (0.15 meters) has been found to provide good results and is the default value used in FLOMOD. However, the best depth tolerance to use will depend on the size of the elements in a network, and on the flow conditions.

It is possible that an element that would actually be wet is turned off in the final solution. However, the depth of flow in the element would be small, and the effect of not including the element in the active network would be negligible. The possibility of a wet element being turned off in a final solution can be minimized by constructing smaller elements in areas where the active network boundary is expected to occur.

The automatic boundary-adjustment feature allows a finite element network to be designed without too much concern for the location of boundaries. However, ground-surface elevations still need to be assigned carefully. If the automatic boundary-adjustment feature is used and a high node point (located on a channel bank in the middle of a flood plain, for example) becomes dry, all the elements connected to that node point will be turned off for the next iteration. Removal of a large number of elements from an active network could affect significantly the solution unless all the elements turned off were quite small.

Either slip or no-slip conditions (as specified by a user) are applied automatically at all existing or newly created boundary nodes. However, if a velocity, unit flow rate, or depth boundary condition is specified at a node point that is eliminated from an active network, and the node later is re-admitted to the active network, the boundary conditions that were specified for that node will not be applied again. If a velocity, unit flow rate, or water-surface elevation is specified at a node, a user needs to be certain that the node will not be removed from the active network even temporarily.

Initial Conditions and Convergence

Initial conditions (that is, starting values) need to be specified at each node point in an active network at the beginning of a run. Initial conditions consist of depth-averaged velocities in the x and y directions and flow depth (water-surface elevation minus ground-surface elevation) at corner nodes, just depth-averaged velocities at midside and center nodes, and time-derivatives of each of the variables if the simulation is time-dependent (unsteady).

Cold starts

Determining reasonable starting values if a solution has not been obtained previously may at first seem to be a difficult or an impossible task. However, a cold start procedure can be used whereby a constant water-surface elevation and zero velocities are assigned. A cold start is the only practical way of beginning a simulation on a new network.

During a cold start, the water-surface elevation is initially assumed constant and velocities are set to zero at all node points. A constant water-surface elevation is specified in the SWMS data set on the system-parameters record (SWMS.5). Because FLOMOD solves for depth of flow, the ground-surface elevation is subtracted from the constant water-surface elevation to obtain the depth of flow at all corner nodes. A constant water-surface elevation can be overridden at any node point by specifying a water-surface elevation as an initial condition at that node point. Initial velocities at node points are assumed always to equal zero unless an initial condition value is specified.

When starting cold, the initial constant water-surface elevation needs to be greater than the ground-surface elevation at the node points where boundary conditions are to be specified. Usually computations will need to be started using an initial

water surface that is much higher than the final solution water surface at most node points in the finite element network. Dry elements are permitted to exist in a network if the boundary of the network is adjusted automatically. If the boundary of a network is not adjusted automatically to account for dry elements, all active elements in the network always need to be wet (that is, have positive flow depths at all node points).

Hot starts

A solution from a previous run can be used to assign initial conditions for a subsequent run. This way of specifying initial conditions is referred to as a hot start. Because computed velocities and depths usually are written to a data file, a flow data file from a run can thus become an initial condition data file for a subsequent run.

Convergence problems

A solution might not converge if the initial conditions are not sufficiently close to the true solution, regardless of the type of start.

If a cold start is attempted and convergence problems develop, the difference between specified boundary conditions and the initial conditions is probably too great. The relaxation factor, λ , can be decreased so that the solution change from one iteration to the next will be reduced. Try setting λ equal to 0.5 or less for a few iterations if solution convergence is a problem.

It may be necessary to obtain an intermediate solution using boundary conditions that are somewhere between the initial condition values and the desired boundary condition values. The next run will be started hot using the results from the intermediate solution as initial conditions.

Sometimes more than one intermediate solution will be required before the desired boundary conditions can be applied. The number of intermediate solutions needed depends on the difference between the initial conditions at the boundaries and the boundary conditions that are specified. If the difference is small, intermediate solutions are probably not needed at all.

Another way of solving a convergence problem is to temporarily increase kinematic eddy viscosity to a large value. Large eddy viscosities encourage solution convergence because of their dissipative effect when velocity gradients are large. Large values of eddy viscosity probably need to be maintained for two or three iterations before they are reduced to physically appropriate values.

If convergence problems persist and a plot of the finite element network has been inspected carefully for geometric inconsistencies, then the network probably needs to be refined. An entire network can be refined automatically using the network refinement option in DINMOD. If only a part of a network needs to be refined, the refinement needs to be done manually. Areas in need of refinement probably will be located where velocities and variations in velocities are extremely large, such as near constrictions (bridge openings).

Specifying Boundary Conditions

Water-surface elevation, velocity, unit flow rate, or total flow normal to a boundary are the values that can be specified as boundary conditions. Although all boundary conditions except total flow can be specified at any node point in a finite element network, generally boundary conditions will be prescribed only at node points that actually form part of the network boundary. The two types of boundaries (solid and open) and boundary conditions that are appropriate for each type of boundary are described in section 4.

A five-digit code (variable NFIX in the BOUN data set) is used to define the type(s) of boundary condition(s) that is (are) specified at a node point. The boundary condition code is read as XYZ00, where:

X = digit that controls specification of x and tangential direction velocity or unit flow, or total flow normal to an open boundary resulting from flow at a node point.

Y = digit that controls specifications of y and normal direction velocity or unit flow, or total flow normal to a solid boundary resulting from flow at a node point.

Z = digit that controls specification of water-surface elevation at a node point, or specification of a supercritical flow at a boundary node.

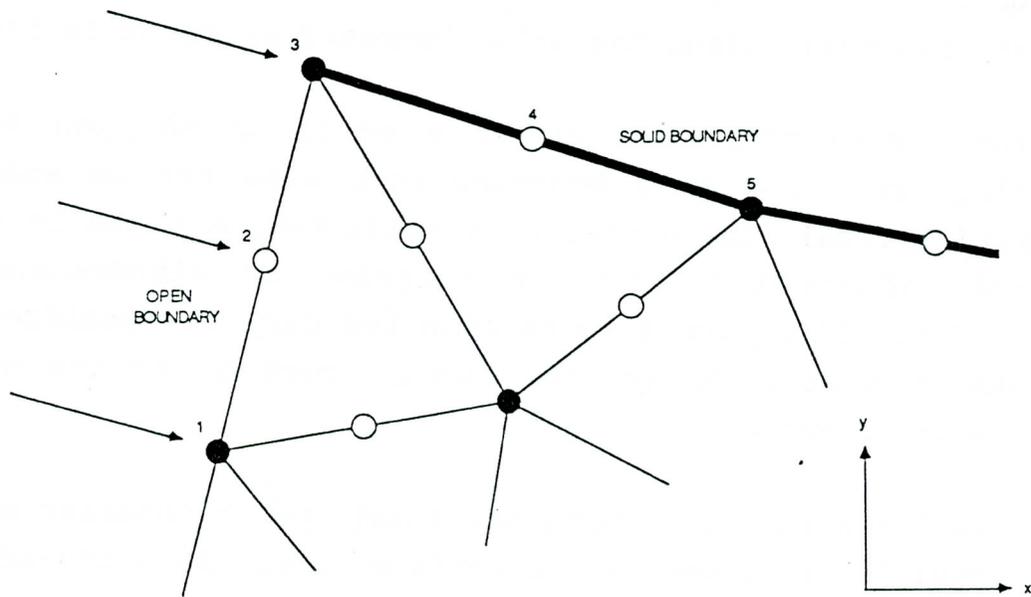
Slip (tangential flow), no-slip (zero flow), or semi-slip (tangential shear stress) conditions can be applied automatically to all solid-boundary node points (see description of the variable ISLIP on the SWMS.2 data record). Slip and no-slip conditions can be overridden by specifications in the boundary condition data set. To apply a no-slip condition at a boundary node, the boundary condition code is set to 11Z00, and the x and y velocities are set to zero. To apply a slip or a semi-slip condition at a boundary node, the boundary condition code 05Z00 is assigned to the node and total flow normal to the boundary at the node is set to zero.

Special consideration needs to be given a node point where a solid boundary and an open boundary meet (a boundary interface node). Boundary type (solid or open) needs to change always at a corner node and never at a midside node. If unit flow or velocity is prescribed at a boundary interface node, the resulting velocity vector needs to be parallel to the solid boundary (see node 3 in figure 7-8 for an example). If a velocity vector at a boundary interface node is not parallel to the solid boundary, flow will "leak" into or out of the network

depending on the direction of the vector. If water-surface elevation is prescribed at a boundary interface node, either a slip condition or a no-slip condition also needs to be specified (see node 3 in figure 7-9 for an example), otherwise accidental "leaks" will exist along the solid boundary adjacent to the node.

Total flow normal to either a solid or an open boundary resulting from flow at a boundary node also can be specified. Prescribing total flow normal to a solid boundary can be used to simulate inflows from small tributaries or withdrawals at a point. Total flow normal to an open boundary is specified where major known inflows or outflows occur, such as at the upstream end of a river reach.

Total flow normal to both solid and open boundaries may need to be specified at a boundary interface node. If a non-zero flow normal to an open boundary is specified at a boundary interface node, and a slip condition is prescribed along the solid boundary (see node 3 in figure 7-8 for an example), the velocity vector at the boundary interface node will be directed into the network parallel to the solid boundary.



EXPLANATION

● CORNER NODE

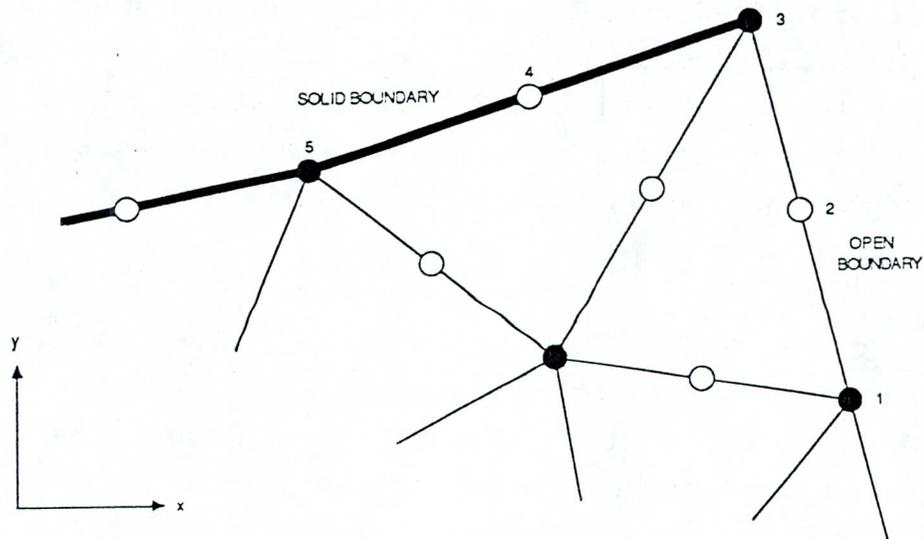
○ MIDSIDE NODE

Node No.	Boundary Condition Code	X-direction Specification	Y-direction Specification	Water-Surface Elevation
1	22000	1.6 cfs/ft	-0.4 cfs/ft	--
2	22000	1.4 cfs/ft	-0.5 cfs/ft	--
3	22000	1.8 cfs/ft	-0.6 cfs/ft	--
4	05000	---	0 cfs	---
5	05000	---	0 cfs	---

Notes:

- (1) Velocity or unit flow vectors need to be parallel to the solid boundary at a node where an open boundary and a solid boundary meet (such as node 3).
- (2) Zero flow normal to a solid boundary (a "slip" condition) is automatically specified at all solid boundary nodes if ISLIP = 0.

Figure 7-8. Example of boundary condition specifications where a solid boundary and an open boundary meet and unit flows are specified.



EXPLANATION

- CORNER NODE
- MIDSIDE NODE

Node No.	Boundary Condition Code	X-direction Specification	Y-direction Specification	Water-Surface Elevation
1	00200	---	---	7.2 ft
2	00200	---	---	7.3 ft
3	05200	---	0 cfs	7.4 ft
4	05000	---	0 cfs	---
5	05000	---	0 cfs	---

Notes:

- (1) Both a slip condition (zero flow across a solid boundary resulting from flow at a node) and water-surface elevation are specified at node 3. Computed flow at the node will be parallel to the solid boundary.
- (2) Zero flow normal to a solid boundary (a "slip" condition) is automatically specified at all solid boundary nodes if ISLIP = 0.

Figure 7-9. Example of boundary condition specifications where a solid boundary and an open boundary meet and water-surface elevations are specified.

Total flow at a cross section

Total flow at a cross section that forms part of an open boundary of a finite element network can be specified as a boundary condition. An open boundary where total flow at a cross section is specified usually will represent the upstream end of a river/flood plain model where total flow has been measured or where a relation between total flow and stage (a rating curve) exists. The specified total flow at a cross section is distributed among the nodes that define the cross section on the basis of conveyance.

Each cross section at which total flow is specified is identified by a number from 1 to 10. Thus, total flow can be specified at up to 10 cross sections in a model. Up to 79 node points may be used to define each cross section. Positive flows represent inflows to a network, and negative flows represent outflows from a network. The specified total flow at a cross section is distributed among the nodes that define the cross section on the basis of conveyance.

Water-surface elevation at a cross section

A level or sloping water-surface elevation at a cross section that forms part of an open boundary of a network can be specified as a boundary condition. An open boundary where water-surface elevation at a cross section is specified usually will represent the downstream end of a river/flood plain model where high-water marks have been collected, or a channel in a model of a bay or estuary where stage has been recorded. The water-surface elevation may be specified directly or calculated using the slope-area method. If the slope-area method is used to determine the water-surface elevation, the calculated elevation is applied to all node points of the cross sections (that is, a level water surface is assigned).

Each cross section at which water-surface elevation is specified is identified by a number from 1 to 10. Thus, water-

surface elevation can be specified at up to 10 cross sections in a model. Up to 79 node points may be used to define each cross section. If a level water-surface is specified, a single water-surface elevation and a boundary condition code need to be prescribed. Specification of a sloping water-surface requires two elevations, one at each end node of the cross section. Intermediate elevations along the cross section are interpolated on the basis of a linear variation. If the slope-area method is used, the total flow rate through the section and the total energy slope need to be specified along with the boundary condition code that will be assigned to each node point of the cross section.

Output Analysis Module: ANOMOD

ANOMOD presents the results of flow simulations in the form of reports and plots, and acts as a data post-processor in the modeling system. Specifically, ANOMOD does the following:

- Prints finite element network data (node and element data).
- Prints flow data (velocities in the x and y directions, water depth, and water-surface elevation).
- Plots a finite element network.
- Plots velocity or unit flow vectors.
- Plots ground-surface elevation contours.
- Plots water-surface-elevation contours.
- Plots flow-check lines.
- Prints a time-history report of velocity, unit flow, and stage (water-surface elevation).
- Plots time-history graphs of velocity, unit flow, or water-surface elevation at a node point.

- Plots contours of the difference between water-surface elevations produced by two different simulations. This function can be used to plot contours of backwater caused by a bridge.
- Plots ground-surface elevation, water-surface elevation, velocity, or unit flow at a cross section.
- Checks all node and element numbers, time-history node numbers, the number of flow-check lines, and the number of element sides for compatibility with appropriate array dimensions and other program limits.

Reports present finite element network data and/or flow simulation results (flow data) in a printed format. Although much of this information can be presented by other FESWMS-2DH programs, ANOMOD provides yet another means of generating data reports. Analysis of results in report form will be sufficient for only the smallest of networks. It is suggested that extensive use be made of the plotting capabilities of ANOMOD to insure a satisfactory simulation.

The logical flow of information through ANOMOD from the entry of input data to the generation of output, and the major software elements of the model are illustrated in figure 7-10.

ANOMOD begins by reading job control data records. Input data files are read next if requested. Network data records and flow data records are read next and will override information previously read from an input data file.

Next, data sets that control network plots, vector plots, contour plots, flow-check line plots, and time-history reports and graphs are read, and the requested operations are performed immediately. Plots may be combined by entering the appropriate code on the data set identification record. Plots that combine vectors, contours, and flow-check lines can be used to present and analyze the results of simulations in a variety of ways.

Time-history graphs plot velocity, unit flow, or stage versus time and are used to present the results of a time-dependent (unsteady) flow simulation. Time-history graphs are drawn within an 8.5-in by 11.0-in border. Velocity and unit flow graphs illustrate the magnitude and direction of velocity and unit flow vectors, respectively. Time-history reports consist of a printed summary of the velocity, unit flow, or stage versus time at a specified node point.

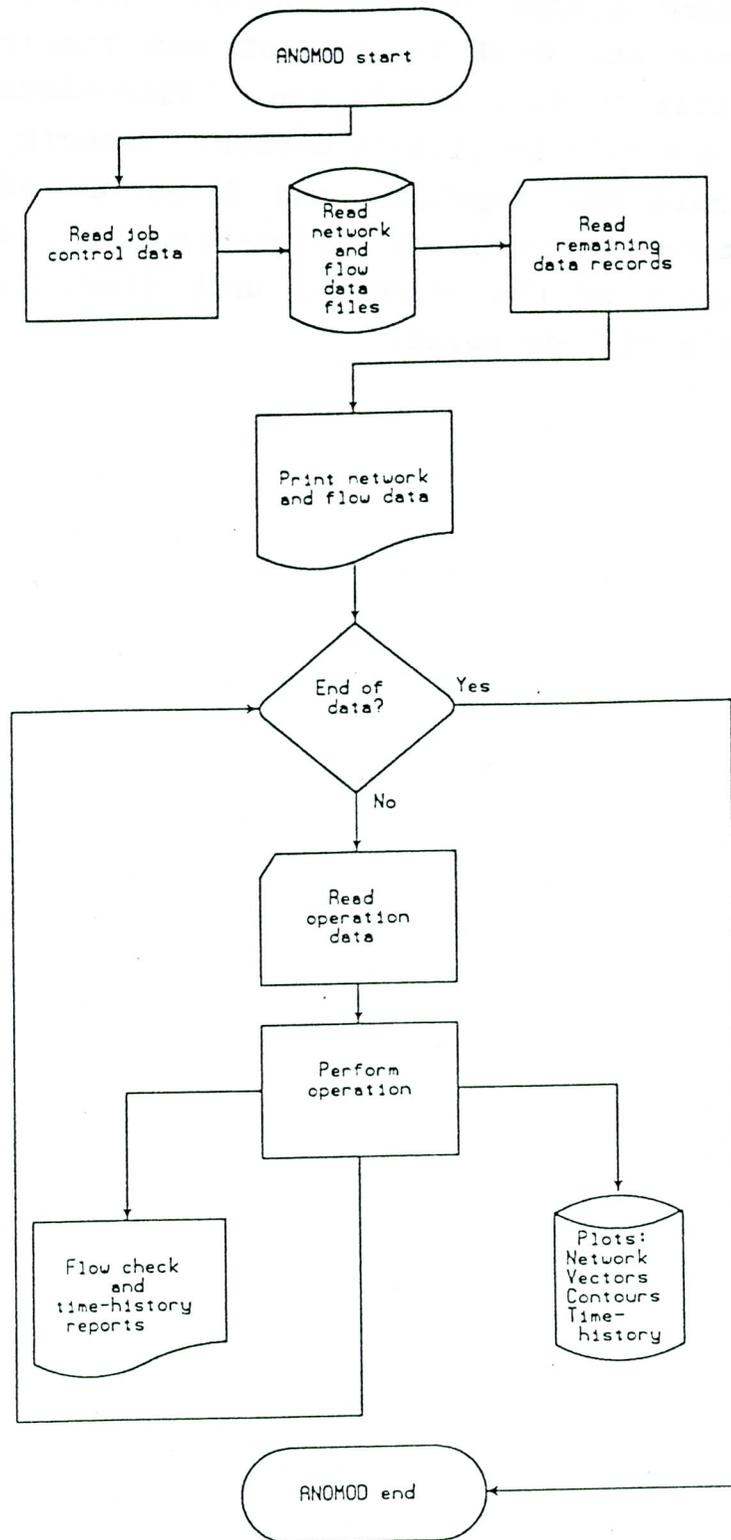


Figure 7-10. Schematic diagram of the Analysis of Output Module (ANOMOD) showing the logical flow of data through the module and the major functions that are performed.

Section 8

INPUT DATA OVERVIEW

Input data needed to run FESWMS-2DH are described in detail in this section. The material in this section and the remaining sections of this manual will serve as a reference for anyone who uses the modeling system. Input data files for several examples are provided in Appendix C.

Input to FESWMS-2DH is read from data records and data files. Information read from a data record always will supersede information read from a data file.

Input for each module of the modeling system is organized into groups, or sets, of related data. Data sets consist of one or more data records. For some data sets, the same data record may be repeated many times. Each data set is preceded by a data set identification record which indicates the type of data to follow. Input of each data set is controlled by a data set code contained in the first four or five columns of a data set identification record.

The first data group in an input stream needs to be the program control data set, which is preceded by an SWMS identification record. Other data sets can follow in any order unless noted otherwise. A LAST identification record signals the end of all input data, and needs to be the last record in an input data stream. Each data set identification record needs to be followed immediately by the appropriate input data. Some data sets need to be terminated by one or more blank data records.

The related groups of data that establish and define a data set are described in the following sections for each of the FESWMS-2DH modules. Each section contains an overview of the contents and purpose of a data set, followed by a description of each variable in the data set including the variable's location on a data record, name, allowable values, and definition. If applicable, the unit of measurement, the default value (that is, the value assumed when the entry is left blank), and the relation of a variable to other data items also are described.

The locations of variables on each data record are indicated by column numbers. Each data record is divided into fields of either 5 or 10 columns each. Integer variable names begin with the letters I through N, and real variable names begin with the letters A through H and O through Z, unless indicated otherwise.

The values a variable may assume, and the conditions for each value are described. Some data items simply call for use of program options, and other data items contain numerical values that express the magnitude of a variable. For the latter type of data item, a plus (+) or minus (-) sign indicates the admissible numerical values for that variable. An entry left blank is read as zero, and a default value will be assigned if appropriate.

Input Data Files

Input data files that contain finite element network data, initial condition (flow) data, boundary condition data, and wind data can be read by FESWMS-2DH programs. Input data files may be either formatted or unformatted. Examples of Fortran statements that can be used to read the data files are presented in this section.

Network Data File

A network data file contains node and element data that define a finite element network. An unformatted network data file can be read using the following Fortran statement:

```
READ(IUNIT) NP,NE,(XCORD(N),YCORD(N),GRND(N),CEIL(N),  
# N=1,NP),((LNODES(K,L),K=1,9),LTYPE(L),LASEQS(L),L=1,NE)
```

A formatted network data file can be read using the following Fortran statements:

```
READ(IUNIT,1001) NP,NE  
READ(IUNIT,1002) (N,(XCORD(N),YCORD(N),GRND(N),CEIL(N),  
# N=1,NP)  
READ(IUNIT,1003) (L,(LNODES(K,L),K=1,9),LTYPE(L),LASEQS(L),  
# L=1,NE)  
1001 FORMAT(2I10)  
1002 FORMAT(I10,4E10.0)  
1003 FORMAT(12I5)
```

Variables contained in the preceding Fortran statements used to read a network data file are defined below:

IUNIT	= Fortran unit number of the network data file.
NP	= largest node point number.
NE	= largest element number.
N	= node point number.
K	= variable array index.
L	= element number.
XCORD	= an array that contains the x coordinate of each node point.
YCORD	= an array that contains the y coordinate of each node point.

GRND = an array that contains the ground-surface elevation of each node point.
 CEIL = an array that contains the ceiling elevation of each node point.
 LNODES = an array that contains the connectivity list of each element.
 LTYPES = an array that contains the property type code of each element.
 LASEQS = an array that contains the assembly sequence of each element

Initial Condition (Flow) Data File

An initial condition (flow) data file contains values of depth-averaged velocity in the x and y directions, water depth, and time derivatives of these quantities at each node point. These values are used as initial conditions for a run. An initial condition data file is usually a flow data file that has been generated by a previous run. An unformatted initial condition data file can be read using the following Fortran statement:

```

    READ(IUNIT) TIME, NP, (XVEL(N), YVEL(N), DEPTH(N),
    # UDOT(N), VDOT(N), HDOT(N), N=1, NP)
  
```

A formatted initial condition data file can be read using the following Fortran statements:

```

    READ(IUNIT, 1001) TIME, NP
    READ(IUNIT, 1002) (N, (XVEL(N), YVEL(N), DEPTH(N),
    # UDOT(N), VDOT(N), HDOT(N), N=1, NP)
  1001  FORMAT(E10.0, I10)
  1002  FORMAT(I10, 6E10.0)
  
```

Variables contained in the preceding Fortran statements used to read an initial condition (flow) data file are defined below:

IUNIT = Fortran unit number of the initial condition data file.
 TIME = simulation time in hours.
 NP = largest node point number in the network.
 N = node point number.
 XVEL = an array that contains the value of U (x velocity) at each node point.

YVEL = an array that contains the value of V (y velocity) at each node point.
 DEPTH = an array that contains the value of H (water depth) at each node point.
 UDOT = an array that contains the value of $\delta U/\delta t$ at each node point.
 VDOT = an array that contains the value of $\delta V/\delta t$ at each node point.
 HDOT = an array that contains the value of $\delta H/\delta t$ at each node point.

Boundary Condition Data File

A boundary condition data file contains values of boundary condition codes and boundary condition specifications for node points where boundary conditions are prescribed. During a time-dependent simulation, boundary condition data for a node point remain unchanged until new specifications are read. An unformatted boundary condition data file can be read using the following Fortran statement:

```

    READ(IUNIT) TIME,NODES,
    # (N,NFIXES(N),XSPEC(N),YSPEC(N),ZSPEC(N),I=1,NODES)
  
```

A formatted boundary condition file can be read using the following Fortran statements:

```

    READ(IUNIT,1001) TIME,NODES
    READ(IUNIT,1002) (N,NFIXES(N),XSPEC(N),YSPEC(N),ZSPEC(N),
    # I=1,NODES)
    1001 FORMAT(E10.0,I10)
    1002 FORMAT(2I10,3E10.0)
  
```

Variables contained in the preceding Fortran statements used to read a boundary condition data file are defined below:

IUNIT = Fortran unit number of the boundary condition data file.
 TIME = simulation time in hours.
 NODES = number of node points for which boundary conditions (which apply at the specified simulation time) are to be read.
 N = node point number.
 NFIXES = an array that contains the boundary condition code at each node point.
 XSPEC = an array that contains the boundary specification

for flow in the x or tangential direction at each node point.

YSPEC = an array that contains the boundary specification for flow in the y or normal direction at each node point.

ZSPEC = an array that contains the water-surface elevation specification at each node point.

I = variable index.

Wind Data File

A wind data file contains wind direction angle and velocity at node points. During a time-dependent simulation, wind data for a node point remain unchanged until new specifications are read. An unformatted wind data file can be read using the following Fortran statement:

```
READ(IUNIT) TIME, NODES, (N, WINDA(N), WINDV(N), I=1, NODES)
```

A formatted wind data file can be read using the following Fortran statements:

```
READ(IUNIT, 1001) TIME, NODES
READ(IUNIT, 1002) (N, WINDA(N), WINDV(N), I=1, NODES)
1001 FORMAT(E10.0, I10)
1002 FORMAT(I10, 2E10.0)
```

Variables contained in the preceding Fortran statements used to read a wind data file are defined below:

IUNIT = Fortran unit number of the wind data file.

TIME = simulation time in hours.

NODES = number of node points for which wind data (that apply at the specified simulation time) are to be read.

N = node point number.

WINDA = an array that contains the wind direction angle at each node point.

WINDV = an array that contains the wind velocity at each node point.

I = variable index.

Interactive Data File Editors

Entering data is a time-consuming task that is prone to error. The interactive data file editing programs DINDFE, FLODFE, and ANODFE will simplify entry of input data and reduce the risk of making errors when entering data. The data file editing programs allow a user to interactively edit new or existing input data for DINMOD, FLOMOD, and ANOMOD, respectively. The operation of the data file editors is controlled by menus that are displayed on a terminal screen. A user is prompted for each data item that is required by a program. The default or current value of a data item is displayed, and the allowable range of the data item is indicated. The data file editing programs write the entered data to input data files in the required format, and greatly reduce the chance of data-entry error.

The data file editors are easy to use and almost eliminate the need for the users manual. However, a thorough check of network geometry still is needed because elimination of data-entry errors does not insure geometric consistency. Also, a plot of the network still needs to be examined closely for poor design and incorrect ground-surface elevations.

This page is blank.

Section 9

INPUT DATA DESCRIPTION: DINMOD

Data sets read by the Data Input Module (DINMOD) are preceded by the following identification records:

DATA SET IDENTIFICATION RECORD - FORMAT(A4,6X,7E10.0)

<u>Variable</u>	<u>IDS</u>	<u>Data description</u>
SWMS	¹	Program control data
PLOT		Plot control data
ELEM		Element data
NODE		Node data
INTE		Node interpolation data
AUTO		Automatic grid generation data
RESE		Element resequencing data
LAST	²	End of data

¹Required first data set. All other data sets may follow in any order.

²Needs to be the last record in the data input stream.

SWMS

Program Control Data Set

Program control data records immediately follow an SWMS data set identification record. The identification record contains the data set identification code, a code that controls the printed output format, and a code that controls printing of messages on the screen. The remaining two records contain information that controls the overall operation of the program. Record 1 contains the job title which is used in printed output headings. Record 2 contains job option codes that control printing, plotting, other program operations, and input and output file specifications.

SWMS IDENTIFICATION RECORD - FORMAT(A4,6X,2E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..4	IDS	SWMS	Program control data set identification code.
5..10	--	--	Not used.
11..20	WIDE	0	132-column format will be used for all printed output.
		1	80-column format will be used for all printed output.
21..30	SCREEN	0	Messages that describe program operations will not be printed on the terminal screen.
		1	Messages that describe program operations will be printed on the terminal screen.

SWMS RECORD 1 - FORMAT(A80)

1..80	JTITLE	a/n	Alphanumeric characters to be used in printed output headings that identify the job. Usually the name of the water body being modeled.
-------	--------	-----	--

SWMS RECORD 2 - FORMAT(8I5)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..5	IPRNT	0..3	The sum of the following printed output options that are desired: <u>Options</u>
		0	Control data, error messages, and job completion statements will be printed.
		1	Input data read from records will be echo printed.
		2	Processed element and node data will be printed.
6..10	IPLOT	0..3	The sum of the following plot options that are desired: <u>Options</u>
		0	Nothing will be plotted.
		1	A network plot will be created using information contained in the PLOT data set.
		2	A contour plot of ground-surface elevations will be created using information contained in the PLOT data set.
11..15	ICHEK	0	A geometric data check will not be performed.
		1	A geometric data check will be performed. The finite element network will be thoroughly checked for geometric consistency and completeness.
16..20	IAUTO	0	Automatic grid generation will not be performed.
		1	Automatic grid generation will be performed. All or part of the network will be generated automatically.

SWMS

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
21..25	IREFN	0	The finite element network will not be refined.
		1	The finite element network will be refined. All elements in the network will be subdivided into four similar elements. Data for newly created node points and element connectivity lists will be generated automatically.
26..30	IRESQ	0	Element resequencing will not be performed.
		1	Element resequencing will be performed. A more efficient element assembly sequence will be sought using information entered in the RESE data set.
31..35	IGIN	0	A finite element network (grid) data file will not be read. All network data will be entered on data records.
		1	Finite element network (grid) data will be read from a data file. Additional network data may be entered on data records. The default network data file name is "GRID.DAT".
36..40	IGOUT	0	A finite element network (grid) data file will not be written at the end of the run.
		±	Finite element network (grid) data will be written to a file. If IGOUT > 0, a formatted file will be written; if IGOUT < 0, an unformatted file will be written. The default network data file name is "GRID.DAT".

Plot Data Set

Plot data records immediately follow a PLOT data set identification card. The four records in the data set contain information that control the network plot. Input data include a title to be plotted, codes that control the values to be plotted at node points and element centroids, and plot scale and transformation data.

PLOT RECORD 1 - FORMAT(A80)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..80	PTITLE	a/n	Alphanumeric characters to be used as the plot title.

PLOT RECORD 2 - FORMAT(2I5)

1..5	IPLTN	0..7	The sum of the following plot options that are desired:
------	-------	------	---

Options

0	Nothing will be plotted at node points.
---	---

1	Node numbers will be plotted at node points.
---	--

2	Ground-surface elevations will be plotted at node points.
---	---

4	Ceiling elevations will be plotted at node points.
---	--

6..10	IPLTE	0..7	The sum of the following plot options that are desired:
-------	-------	------	---

Options

0	Nothing is to be plotted at element centroids.
---	--

1	Element numbers will be plotted at element centroids.
---	---

2	Property type codes will be plotted at element centroids.
---	---

4	Element sequences will be plotted at element centroids.
---	---

PLOT

PLOT RECORD 3 - FORMAT(5E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..10	HTT	+	Height of the title to be plotted along the y axis, in inches. The default value is 0.14 inches.
11..20	HTN	+	Height of numbers to be plotted at network node points, in inches. The default value is 0.07 inches.
21..30	HTE	+	Height of numbers to be plotted at element centroids, in inches. The default value is 0.105 inches.
31..40	HTC	+	Height of contour line labels, in inches. The default value is 0.105 inches. Contour line labels are plotted along network boundaries where the boundary is intersected by a contour line. The label is the value represented by the contour line.
41..50	CVINC	+	Contour interval in feet (meters) that is used if a ground-surface elevation contour plot is requested.

PLOT RECORD 4 - FORMAT(7E10.0)

1..10	XLOWER	+	The x coordinate in feet (meters), before rotation, of the point that will be the lower lefthand corner of the network plot after it is rotated ROTATE degrees.
11..20	YLOWER	+	The y coordinate in feet (meters), before rotation, of the point that will be the lower lefthand corner of the network plot after it is rotated ROTATE degrees.

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
21..30	XOFFSET	+	The x direction offset in feet (meters) used to determine the upper righthand corner of the network plot. The default value is 1.0E+35 (that is, everything to the right of the lower lefthand corner point will be plotted).
31..40	YOFFSET	+	The y direction offset in feet (meters) used to determine the upper righthand corner of the network plot. The default value is 1.0E+35 (that is, everything above the lower lefthand corner point will be plotted.)
41..50	XSCALE	+	The x coordinate plot scale, in feet (meters) per inch.
51..60	YSCALE	+	The y coordinate plot scale, in feet (meters) per inch.
61..70	ROTATE	+	Angle of plot rotation, in degrees measured counterclockwise.

ELEM

Element Data Set

Element data records immediately follow an ELEM data set identification card. One record is required for each element. An element data record contains the element number, the sequence of nodes connected to the element (the element connectivity list), the element property type code, and the element assembly sequence. The data set is terminated with one or more blank data records.

ELEM RECORD - FORMAT(12I5)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..5	L	+	Element number; needs to be less than or equal to the element array dimension.
6..50	LNODES(K,L)	+	Element connectivity list. The sequence of nodes connected to the element. Enter 6 node numbers for a triangular element, 8 node numbers for a "serendipity" quadrangular element, or 9 node numbers for a "Lagrangian" quadrangular element. Begin the list at any corner node and proceed in a counterclockwise direction around the element. For a 9-node quadrangular element, the center node is entered last.
51..55	LTYPES(L)	±	Element type code. The type code corresponds to a set of element parameters (coefficients) that are entered in the PROP data set in FLOMOD. If LTYPES(L) < 1, the element will be turned off (that is, it will not be used).
56..60	LASEQS(L)	+	Element assembly sequence. The sequence number of the element for processing by the frontal method. An efficient element sequence can be generated using the resequencing option.

Terminate the ELEM data set with one or more blank data records.

Node Data Set

Node data records immediately follow an NODE data set identification record. The identification record contains the data set identification code and factors used to convert node point coordinates and ground and ceiling elevations read from data records to the desired units (either feet or meters). A node data record is required for a corner node. Curved element sides are specified by also entering coordinate data for the midside node of the element side. A node data record contains the node number, values of the measured coordinates, and ground-surface and ceiling elevations at the node point. The data set is terminated with one or more blank data records.

NODE IDENTIFICATION RECORD -FORMAT(A4,6X,6E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..4	IDS	NODE	Node data set identification code.
5..10	--	--	Not used.
11..20	XFACT	+	Multiplication factor used to convert x coordinates read from data records to feet (meters). The default value is 1.0.
21..30	YFACT	+	Multiplication factor used to convert y coordinates read from data records to feet (meters). The default value is 1.0.
31..40	ZFACT	+	Multiplication factor used to convert ground and ceiling elevations read from data records to feet (meters). The default value is 1.0.
41..50	XZERO	+	Feet (meters) to be added to all x coordinates read from data records after they are multiplied by XFACT. The default value is 0.0 ft (0.0 m).
51..60	YZERO	+	Feet (meters) to be added to all y coordinates read from data records after they are multiplied by YFACT. The default value is 0.0 ft (0.0 m).

NODE

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
61..70	ZZERO	+	Feet (meters) to be added to all ground and ceiling elevations read from data records after they are multiplied by ZFACT. The default value is 0.0 ft (0.0 m).

NODE RECORD - FORMAT(I10,4E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..10	N	+	Node number; needs to be less than or equal to the nodal array dimension.
11..20	XYZZ(1,N)	+	X coordinate of node N.
21..30	XYZZ(2,N)	+	Y coordinate of node N.
31..40	XYZZ(3,N)	+	Ground-surface elevation at node N.
41..50	XYZZ(4,N)	+	Ceiling elevation at node N.

Note -- Coordinates and elevations are converted to feet (meters) by the factors specified on the data set identification record. Thus, any system of units can be used to record coordinates and elevations.

Terminate the NODE data set with one or more blank data records.

Node Interpolation Data Set

Node interpolation data records immediately follow an INTE data set identification record. One set of data records needs to be prepared for each straight line along which X or Y coordinates are to be interpolated. Node interpolation data records contain a code that identifies the type of interpolation to be performed, and a list of corner nodes for which coordinates are to be interpolated. The first node in the list is one endpoint of the line and the last node in the list is the other endpoint. Terminate the data set with one or more blank data records.

INTE RECORD 1 - FORMAT(I5)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..5	INTYP	1	Y coordinates will be interpolated at each node in the following list using the specified X coordinate of the node.
		2	X coordinates will be interpolated at each node in the following list using the specified Y coordinate of the node.
		3	X and Y coordinates of corner nodes spaced an equal distance apart will be interpolated at each node in the following list.

INTE RECORD 2 - FORMAT(16I5)

1..80	LINE(K)	+	List of straight-line corner nodes beginning at one endpoint and ending at the other endpoint entered in fields of five columns each. The list is terminated with a -1 entry. The first entry for each interpolation line needs to be placed in the first field. Use as many records as necessary to complete the list.
-------	---------	---	---

Note -- The only limit placed on the number of node point interpolation lines and the number of points contained in a single line is that the total number of points in all lines may not exceed the LINE array dimension.

AUTO

Automatic Grid Generation Data Set

Automatic grid generation data immediately follow an AUTO data set identification card. Automatic grid generation data consist of control information, and a list of corner nodes that defines a simply connected region which is to be subdivided into a network of straight-sided 6-node triangular elements. The data set is terminated with one or more blank data records.

AUTO IDENTIFICATION RECORD - FORMAT(A4,6X,E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..4	IDS	AUTO	Data set identification code.
5..10	--	--	Not used.
11..20	OMEGA	+	Value of the weighting coefficient ω used in equation 7-1. The default value is 0.333.

AUTO RECORD 1 - FORMAT(3I5)

1..5	NPTS	+	Number of corner node points in the following list.
6..10	LTYPE	+	Element type code to be assigned to each automatically generated element in the region defined by the following list.
11..15	NITSM	+	Number of smoothing iterations to be performed after the elements have been generated. The default value is 5.

AUTO RECORD 2 - FORMAT(16I5)

1..80	LNGEN(K)	+	List of corner node points that define a simply connected region that is to be subdivided into a network of triangular elements, entered in fields of five columns each. Corner nodes are entered in a counter-clockwise direction starting at any corner node. Use as many records as needed to complete the list.
-------	----------	---	---

Terminate the AUTO data set with one or more blank data records.

Resequencing Data Set

Resequencing data immediately follow an RESE data set identification record. Element resequencing data consist of control instructions, and a list of elements that are used to begin the ordering of elements for a more efficient solution by the frontal method. As a general rule, at least two starting locations should be tried, one at either end of a finite element network. Up to 10 sets of resequencing data may be entered. The data set is terminated with one or more blank data records.

RESE RECORD 1 - FORMAT(2I5)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..5	IRTYP	1	Resequencing will be performed using the minimum frontgrowth method.
		2	Resequencing will be performed using the level structure method.
6..10	MXHLD	+	Maximum length of stay in the element wavefront used by the minimum frontgrowth method. The default value is 100.

RESE RECORD 2 - FORMAT(16I5)

1..80	ISEQ(J)	+	List of element numbers used to begin resequencing, entered in fields of five columns each. The resequencing list is terminated with a -1 entry. The first entry for each list needs to be placed in the first field. A maximum of 80 elements can be contained in each resequencing list.
-------	---------	---	--

Terminate the RESE data set with one or more blank data records.

This page is blank.

Section 10

INPUT DATA DESCRIPTION: FLOMOD

Data sets read by the Depth-Averaged Flow Module (FLOMOD) are preceded by the following identification records:

DATA SET IDENTIFICATION RECORD - FORMAT(A4,6X,7E10.0)

<u>Variable IDS</u>	<u>Data description</u>
SWMS ¹	Program control data
ELEM	Element data
NODE	Node data
PROP	Element properties set data
INIT	Initial condition data
BOUN	Boundary condition data
QSEC	Total flow cross-section data
ZSEC	Water-surface elevation cross-section data
WIND	Wind data
FLUX	Flow check data
WEIR	Weir data
CULV	Culvert data
TIME ²	Time-dependent data
LAST ³	End of data

¹Required first data set.

²Time-dependent data sets need to appear in chronological order at the end of the input data stream.

³Needs to be the last record in the input data stream.

Program Control Data Set

Program control data records immediately follow an SWMS data set identification record. The identification record contains the data set identification code and a code that controls the format of printed output. The other five records of the data set contain information that controls the overall operation of the program. Record 1 contains the job title, which is used in printed output headings. Record 2 contains job option codes. Record 3 contains input/output file specifications. Record 4 contains iteration control data and time-dependent solution parameters. Record 5 contains general system specifications.

SWMS IDENTIFICATION RECORD - FORMAT(A4,6X,2E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..4	IDS	SWMS	Program control data set identification code.
5..10	--	--	Not used.
11..20	WIDE	0	132-column format will be used for all printed output.
		+	80-column format will be used for all printed output.
21..30	SCREEN	0	Messages that describe program operations will not be written to the screen during program execution.
		1	Messages that describe program operations will be written to the screen during program execution.

SWMS RECORD 1 - FORMAT(A80)

1..80	TITLE	a/n	Alphanumeric characters to be used in printed output headings that identify the job. Usually the name of the water body being modeled.
-------	-------	-----	--

SWMS RECORD 2 - FORMAT(11I5)

1..5	IDRUN	0	A normal steady-state or time-dependent solution is to be performed.
------	-------	---	--

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
		1	A restart/recovery run is to be performed. Initial conditions will be read from a restart/recovery file "RSRC.DAT".
6..10	IPRNT	0..127	Sum of the following printed output options that are desired: <u>Options:</u>
		0	Control data, error messages, and solution results will be printed.
		1	All input data read from data records will be echo printed.
		2	Element and node data will be printed.
		4	Initial condition data will be printed.
		8	Element assembly sequence will be printed.
		16	Degree-of-freedom array that contains equation numbers that correspond to each nodal variable will be printed.
		32	Froude number at each node point will be calculated and printed at the end of a steady-state solution and at the end of each time step.
		64	Energy head at each node point will be calculated and printed at the end of a steady-state solution and at the end of each time-step.
11..15	IUNIT	0	U.S. customary units will be used in all computations and for printed output.
		1	International System (SI) units will be used in all computations and for printed output.

SWMS

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
16..20	IWIND	0	Wind-induced surface stresses will not be considered.
		1	Wind-induced surface stresses will be considered.
21..25	IFRIC	0	Bottom stresses will be computed using the Manning equation.
		1	Bottom stresses will be computed using the Chézy equation.
26..30	ISLIP	0	Slip (tangential flow/zero tangential shear) conditions will be applied automatically to all solid boundary nodes.
		1	No-slip (zero flow) conditions will be applied automatically to all solid boundary nodes.
		2	Semi-slip (tangential flow/tangential wall shear) conditions will be applied automatically to all solid boundary nodes.
31..35	IHINT	0	Low-order numerical integration will be used on all elements.
		1	High-order numerical integration will be used on all curve-sided elements.
		2	High-order numerical integration will be used on all elements.
36..40	INORM	0..3	Sum of the following continuity norm options that are desired: <u>Options:</u>
		0	Continuity norms will not be computed.
		1	Continuity norms will be computed at the end of a steady-state solution.
		2	Continuity norms will be computed at the end of every time-step of a time-dependent solution.

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
41..45	IONOFF	0	Elements will not be turned on and off during a run.
		1	Elements will be turned on and off during a run.
			Note -- Use of the automatic boundary adjustment option allows elements that are not covered entirely by water to be included in a network. If IONOFF = 1, all "dry" elements will be excluded from the "active" network.
46..50	ISAVE	0	Files that contain the upper and lower decompositions of the coefficient matrix will be deleted at the end of a run.
		1	Files that contain the upper and lower decompositions of the coefficient matrix will be saved at the end of a run.
51..55	NPVMIN	+	Minimum number of completed equations retained in the active matrix during a frontal solution. NPVMIN > 1 provides a choice of pivotal coefficients but will increase the number of computations and the frontwidth of the system of equations. The default value is 1.
SWMS RECORD 3 - FORMAT(8I5)			
1..5	IGRID	0	All finite element network data will be entered on data records.
		1	Finite element network data will be read from a network data file. Additional network data may be entered on data records. The default name of the network data file is "GRID.DAT".
6..10	INITC	0	All initial condition data will be entered on data records.

SWMS

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
		1	Initial condition data will be read from an initial condition data file. Additional initial condition data may be entered on data records. The default name of the initial condition data file is "INIT.DAT".
			Note -- An initial condition data file usually will be a flow data (solution output) file created by a previous run.
11..15	ISOUT	0	Solution output will not be written to a flow data file.
		±	Solution output will be written to a flow data file at the end of a steady-state run and at the end of every time-step of a time-dependent run. If ISOUT > 0, a formatted file will be written; if ISOUT < 0, an unformatted file will be written. Default name of the flow data file is "FLOW.DAT".
16..20	IRSRC	0	A restart/recovery file will not be used.
		±	Intermediate results will be written to a restart/recovery file after every iteration to allow a run to be restarted from the last successful iteration if the run terminates abnormally. If IRSRC > 0, a formatted file will be written; if IRSRC < 0, an unformatted file will be written. The default name of the restart/recovery file is "RSRC.DAT".
21..25	IBCIN	0	All boundary condition data will be entered on data records.
		+	Boundary condition data will be read from a boundary condition data file. Additional boundary condition data may be entered on data records. The default name of the boundary condition data file is "BOUN.DAT".

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
26..30	IWDIN	0	All wind data will be entered on data records.
		+	Wind data will be read from a data file. Additional wind data may be entered on data records. The default name of the wind data file is "WIND.DAT".
31..35	ILCOF	+	Fortran unit number of the unformatted file that will contain the lower triangular decomposition of the coefficient matrix. The default unit number is 98. The file is written only if quasi-Newton iterations are to be performed, or if the upper and lower triangular matrices are to be saved at the end of a run. The default name of the lower coefficient matrix file is "LOWER.DAT".
36..40	IUCOF	+	Unit number of the unformatted file that will contain the upper triangular decomposition of the coefficient matrix. The default unit number is 99. The file is always written. The default name of the upper coefficient matrix data file is "UPPER.DAT".

Note -- Specifying a unit numbers other than the defaults for the lower and upper coefficient matrices may be useful if you are writing the files to a magnetic tape drive. For example, on a PR1ME computer system, Fortran units 21 to 24 are reserved for 9-track magnetic tape units 0 to 3, respectively.

SWMS

SWMS RECORD 4 - FORMAT(4I10,4E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..10	NITS	0	A steady-state solution will not be performed.
		+	Steady-state solution iteration code read as KKJJII where: II = the number of initial full-Newton iterations to be performed (usually 2 or 3). JJ = the number of quasi-Newton iterations to be performed after all the initial full-Newton iterations, and after each additional full-Newton iteration. KK = the number of additional full-Newton iterations to be performed.
			Note -- The maximum allowable number of steady-state iterations is 99. Hence, $II + JJ*(1 + KK)$ needs to be less than or equal to 99.
11..20	NITD	0	A time-dependent solution will not be performed.
		+	Time-dependent solution iteration code read as KKJJII, where II, JJ, and KK for each time step are the same as for a steady-state solution. The maximum total number of iterations at a time step is 99.
21..30	NUPV	0..5	Number of update vectors to be used in a quasi-Newton solution. If NUPV is 0, no update vectors are used and modified-Newton iteration results. The maximum value is 5. The default value is 0 (that is, a modified-Newton iteration is the default).

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
31..40	NITP	+	Iteration print code read as KKJJII where: II = the number of iterations skipped between printed output during a steady-state solution. JJ = the number of iterations skipped between printed output during each time-step of a time-dependent solution. KK = the number of time-steps skipped between printed output during a time-dependent solution. Note -- The default value for II, JJ, and KK is 0 (that is, output will be printed at the end of every iteration of both steady and time-dependent solutions, and and for every time step of a time-dependent solution.
41..50	TSTRT	+	Starting simulation time, in hours.
51..60	TMAX	0	A time-dependent solution will not be performed.
		+	Maximum simulation time of for a time-dependent solution, in hours.
61..70	DELT	+	Length of each time step used in a time-dependent solution, in hours.
71..80	THETA	+	Time integration factor (dimensionless). The minimum value is 0.5; the maximum value is 1.

SWMS

SWMS RECORD 5 - FORMAT(8E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..10	WSEL	+	Water-surface elevation, in feet (meters). WSEL is assigned to each node point in a network that has not been assigned an initial water-surface elevation. WSEL can be used to assign a constant water-surface elevation for a cold start.
11..20	OMEGA	0	Effect of the Coriolis force will not be considered.
		±	Average local latitude of the surface-water system, in degrees. OMEGA is positive in the Northern hemisphere and negative in the Southern hemisphere.
21..30	ROWAT	+	Average water density, in slugs per cubic foot (kilograms per cubic meter). The default value is 1.937 slug/ft ³ (999.0 kg/m ³).
31..40	BETA0	+	Coefficient β_0 in equation 4-8 used to compute the momentum correction coefficient. The default value is 1.0.
41..50	CBETA	+	Coefficient c_β in equation 4-8 used to compute the momentum correction coefficient. The default value is 0.0.
51..60	CFLAG	+	Continuity norm flag value. Continuity norms greater than CFLAG will be denoted by an asterisk. Appropriate values are problem dependent. The default value is 1.0E+35 (that is, continuity norms will not be flagged).
61..70	DEPTOL	+	Depth tolerance, in feet (meters), used during automatic boundary adjustment to decide whether or not to turn on an element. The default value is 0.5 ft (0.15 m).

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
71..80	RELAX	0 to 2	Relaxation factor ω_r in equation 7-6 used in equation solution. The default value is 1.0.
SWMS RECORD 6 - FORMAT(6E10.0)			
1..10	WVEL	+	Wind velocity, in feet per second (meters per second). This value is assigned to each node in the network unless overridden.
11..20	WDIR	+	Wind direction angle, in degrees measured counterclockwise from the positive x-axis. This value is assigned to each node in the network unless overridden.
21..30	ROAIR	+	Air density, in slugs per cubic foot (kilograms per cubic meter) ₃ . The default is 0.00237 slug/ft ₃ (1.225 kg/m ₃).
31..40	CSURF1	+	Coefficient c_{s1} in equation 4-14 used to compute the wind stress coefficient. The default value is 1.0.
41..50	CSURF2	+	Coefficient c_{s2} in equation 4-14 used to compute the wind stress coefficient. The default value is 0.0.
51..60	WVMIN	+	Minimum wind velocity W_{min} , in meters per second, used in equation 4-14 to compute the wind stress coefficient. The default value is 0.0 m/s.

ELEM

Element Data Set

Element data records immediately follow an ELEM data set identification card. One record is required for each element. An element data record contains the element number, the sequence of nodes connected to the element (the element connectivity list), the element property type code, and the element assembly sequence. The data set is terminated with one or more blank data records.

ELEM RECORD - FORMAT(12I5)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..5	L	+	Element number. Needs to be less than or equal to the element array dimension.
6..50	LNODES(K,L)	+	Element connectivity list. The sequence of nodes connected to the element. Enter 6 node numbers for a triangular element, 8 node numbers for a "Serendipity" quadrangular element, or 9 node numbers for a "Lagrangian" quadrangular element. Begin the list at any corner node and proceed in a counterclockwise direction around the element. For a 9-node quadrangular element, the center node is entered last.
51..55	LTYPES(L)	±	Element type code. The type code corresponds to a set of element parameters (coefficients) that are entered in the PROP data set. If LTYPES(L) < 1, the element will be turned off (that is, the element will not be used in computations).
56..60	LASEQS(L)	+	Element assembly sequence. The sequence number of the element for processing by the frontal method. An efficient element sequence can be generated using the resequencing option.

Terminate the ELEM data set with one or more blank data records.

Node Data Set

Node data records immediately follow an NODE data set identification record. The identification record contains the data set identification code and factors used to convert node point coordinates and ground and ceiling elevations read from data records to the desired units (either feet or meters). A node data record is required for a corner node. Curved element sides are specified by also entering coordinate data for the midside node of the element side. A node data record contains the node number, values of the measured coordinates, and ground-surface and ceiling elevations at the node point. The data set is terminated with one or more blank data records.

NODE IDENTIFICATION RECORD - FORMAT(A4,6X,6E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..4	IDS	NODE	Node data set identification code.
5..10	--	--	Not used.
11..20	XFACT	+	Multiplication factor used to convert x coordinates read from data records to feet (meters). The default value is 1.0.
21..30	YFACT	+	Multiplication factor used to convert y coordinates read from data records to feet (meters). The default value is 1.0.
31..40	ZFACT	+	Multiplication factor used to convert ground and ceiling elevations read from data records to feet (meters). The default value is 1.0.
41..50	XZERO	+	Feet (meters) to be added to all x coordinates read from data records after they are multiplied by XFACT. The default value is 0.0 ft (0.0 m).
51..60	YZERO	+	Feet (meters) to be added to all y coordinates read from data records after they are multiplied by YFACT. The default value is 0.0 ft (0.0 m).

NODE

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
61..70	ZZERO	+	Feet (meters) to be added to all ground and ceiling elevations read from data records after the they are multiplied by ZFACT. The default value is 0.0 ft (0.0 m).

NODE RECORD - FORMAT(I10,4E10.0)

1..10	N	+	Node number; needs to be less than or equal to the nodal array dimension.
11..20	XYZZ(1,N)	+	X coordinate of node N.
21..30	XYZZ(2,N)	+	Y coordinate of node N.
31..40	XYZZ(3,N)	+	Ground-surface elevation at node N.
41..50	XYZZ(4,N)	+	Ceiling elevation at node N.

Note -- Coordinates and elevations are converted to feet (meters) by the factors specified on the data set identification record. Thus, any system of units can be used to record coordinates and elevations.

Terminate the NODE data set with one or more blank data records.

Element Properties Data Set

Element properties data records immediately follow a PROP data set identification record. One record is required for each set of element parameters. An element properties data set record contains the element type code, Manning roughness coefficient as a function of depth, a Chézy discharge coefficient, and turbulence model parameters. The coefficients contained on a data record are applied to an element that has been assigned the type code of that record. The data set is terminated with one or more blank data records.

PROP RECORD - FORMAT(I10,7E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..10	M	0..99	Element type code. The parameters contained on this record are applied to an element that has been assigned this type code.
11..20	LPROPS(1,M) +		Manning roughness coefficient applied to all water depths less than or equal to the depth entered in the next field.
21..30	LPROPS(2,M) +		Water depth, in feet (meters), below which the roughness coefficient entered in the previous field is applied.
31..40	LPROPS(3,M) +		Manning roughness coefficient applied to all depths greater than or equal to the depth in the next field.
41..50	LPROPS(4,M) +		Water depth, in feet (meters), above which the roughness coefficient entered in the previous field is applied.

Note -- For depths greater than the first depth and less than the second depth entered above, the roughness coefficient is interpolated linearly. If the second coefficient is zero or blank, the first coefficient is applied to all depths.

PROP

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
51..60	LPROPS(5,M)	+	Dimensionless Chézy discharge coefficient (C/\sqrt{g}) used for all water depths.
61..70	LPROPS(6,M)	+	Turbulence model base kinematic eddy viscosity, ν_o , in square feet per second (square meters per second), used in equation 4-18.
71..80	LPROPS(7,M)	+	Turbulence model coefficient, c_μ (dimensionless), used in equation 4-18.

Terminate the PROP data set with one or more blank data records.

Initial Condition Data Set

Initial condition data records immediately follow an INIT data set identification record. One record is prepared for each node at which initial conditions are specified. Initial condition data consist of the node number, the initial velocities in the x and y directions, depth of flow, and time derivatives of the x and y velocities and the depth of flow. Values entered on data records will override those read from an initial condition data file. The data set is terminated with one or more blank data records.

INIT RECORD - FORMAT(I10,6E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..10	N	+	Node number; needs to be less than or equal to the nodal array dimension.
11..20	VNEWS(1,N)	±	Initial x velocity at node N, in feet per second (meters per second)
21..30	VNEWS(2,N)	±	Initial y velocity at node N, in feet per second (meters per second).
31..40	VNEWS(3,N)	+	Initial depth of flow at node N, in feet (meters).
41..50	VDOTS(1,N)	±	Initial x velocity rate of change at node N, in feet per second per second (meters per second per second).
51..60	VDOTS(2,N)	±	Initial y velocity rate of change at node N, in feet per second per second (meters per second per second).
61..70	VDOTS(3,N)	±	Initial flow depth rate of change at node N, in feet per second per second (meters per second per second).

Terminate the INIT data set with one or more blank data records.

BOUN

Boundary Condition Data Set

Boundary condition data records immediately follow a BOUN data set identification record. One record is prepared for each boundary node at which conditions other than slip/no-slip/semi-slip are specified. For a time-dependent (unsteady) run, only values that change from the previous time step need to be specified. Boundary condition data consist of the number of the node to which the data apply, a boundary condition code, and specified conditions. Either tangential flow (slip) or zero flow (no-slip) conditions (as determined by variable ISLIP on the SWMS.2 data record) are applied automatically at all boundary nodes unless specified otherwise. Values entered on data records will override those read from a boundary condition data file. The data set is terminated with one or more blank data records.

BOUN RECORD - FORMAT(I10,5X,I5,3E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..10	N	+	Node number; needs to be less than or equal to the nodal array dimension.
11..15	--	--	Not used.
16	NFIXES(N)	1	Velocity in the x direction at node N is specified.
		2	Unit flow in the x-direction at node N is specified.
		3	Velocity <i>tangent</i> to the boundary at node N is specified.
		4	Unit flow <i>tangent</i> to the boundary at node N is specified.
		5	Total flow <i>normal</i> to the open boundary resulting from flow at node N is specified.
17	NFIXES(N)	1	Velocity in the y direction at node N is specified.
		2	Unit flow in the y direction at node N is specified.
		3	Velocity <i>normal</i> to the boundary at node N is specified.
		4	Unit flow <i>normal</i> to the boundary at node N is specified.

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
		5	Total flow normal to the solid boundary resulting from flow at node N is specified.
		8	Velocity tangent to the open boundary at node N is specified.
			Note(1) -- To apply a zero flow (no-slip) condition at a boundary node, enter a "1" in columns 16 and 17, and specify the x and y velocities to be zero. To apply a tangential flow (slip) condition at a boundary node, enter a "0" in column 16, a "5" in column 17, and specify the total flow normal to the boundary to be zero.
			Note(2) -- Special consideration needs to be given nodes where solid and open boundaries meet. Prescribed unit flow or velocity needs to be parallel to the solid boundary. If water-surface is prescribed, tangential or zero flow needs to be specified as described in Note(1) above.
18	NFIXES(N)	1	Water-surface elevation at node N is specified as an essential boundary condition.
		2	Water-surface elevation at node N is specified as a natural boundary condition.
		3	Supercritical flow exists at the outflow boundary node. Water-surface elevation is not specified.
		5	A source (inflow) or sink (withdrawal) is specified at node N. The node does not have to lie on the network boundary.
19	NFIXES(N)	--	Not used; leave blank or enter zero.
20	NFIXES(N)	--	Not used; leave blank or enter zero.

BOUN

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
21..30	BSPECS(1,N)	+	Specified velocity in feet (meters) per second, or unit flow in square feet (meters) per second, in the x direction or tangent to the boundary at node N.
31..40	BSPECS(2,N)	+	Specified velocity, in feet per second (meters per second); unit flow, in square feet per second (square meters per second); or total flow, in cubic feet per second (cubic meters per second), in the y direction or normal to the boundary at node N. Or, if total flow normal to an open boundary is specified, the velocity tangent to the open boundary at node N.
			<p>Note -- If total flow is specified, a positive value indicates flow into a network and a negative value indicates flow out of a network resulting from flow at node N.</p>
41..50	BSPECS(3,N)	+	Specified water-surface elevation, in feet (meters), or total flow, in cubic feet per second (cubic meters per second) if a source (inflow) or sink (withdrawal) is specified, at node N. Total flow is positive for a source, and negative for a sink.

Terminate the BOUN data set with one or more blank data records.

Total Flow Cross-Section Data Set

Total flow cross-section data records immediately follow a QSEC data set identification record. Total flow cross-section data consist of the total flow rate at a cross section that forms an open boundary in a network, and a list of nodes that define the cross section. Each cross section requires one record that contains the section identification number and the total flow rate, followed by up to five additional records that contain a list of node points that define the cross section. Up to 10 cross sections may be specified. The data set is terminated with one or more blank data records.

QSEC RECORD 1 - FORMAT(I10,E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..10	IXSQ	+	Cross section identification number from 1 to 10.
11..20	XSQ(IXSQ)	±	Total flow that is to be distributed between the node points that define the cross section in cubic feet per second (cubic meters per second). A positive value denotes an inflow and a negative value denotes an outflow.

QSEC RECORD 2 - FORMAT(16I5)

1..80	LXSQN(K)	+	A list of node numbers that define a connected series of element sides that represent the cross section. Enter nodes in fields of five columns each. The list is terminated by a -1 entry. Up to 5 records (79 node points plus the -1 entry) may be used to define a cross section. The first entry for each record needs to be placed in the first field.
-------	----------	---	---

Note -- During a time-dependent simulation, total flow is not changed until a new QSEC data set is read. Node points that define the cross section can be changed at any time during a time-dependent run by entering a new list of nodes. If nodes do not change, just enter a -1 in the first field.

ZSEC

Water-Surface Elevation Cross-Section Data Set

Water-surface elevation cross-section data records immediately follow a ZSEC data set identification record. Water-surface elevation cross-section data consist of the water-surface elevation at a cross section that forms an open boundary in a network, and a list of node points that define the cross section. Each cross section requires one record that contains the section identification number, the water-surface elevation or total flow and energy slope, and a boundary condition code, followed by up to five additional records that contain a list of nodes that define the cross section. Up to 10 cross sections may be specified. The data set is terminated with one or more blank data records.

ZSEC RECORD 1 - FORMAT(I10,E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..10	IXSZ	+	Cross section identification number from 1 to 10.
11..20	XSZ1(IXSZ)	±	Water-surface elevation, in feet (meters), to be prescribed at every node point in the cross section if XSZ2 is zero; or water-surface elevation, in feet (meters), at the first node point of the cross section if XSZ2 is not equal to zero; or total flow through the cross section, in cubic feet per second (cubic meters per second), if the water-surface elevation is to be determined by the slope-area method.
21..30	XSZ2(IXSZ)	±	Water-surface elevation, in feet (meters), at the last node point in the cross section list, if the water surface slopes from one end of the cross section to the other; or the energy slope, in feet per foot (meter per meter) if the water-surface elevation is to be determined by the slope-area method is. Enter zero or leave blank if the water-surface elevation is constant across the section.

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
			Note -- If XSZ2 \neq 0, water-surface elevation at node points between the two end nodes are interpolated on the basis of distance between the two end nodes.
31..40	IBCZ(IXSZ)	± 1	Water-surface elevation will be specified as an essential boundary condition at each node of the cross section. A negative value indicates that the water-surface elevation will be determined by the slope area method and that flow rate and energy slope have been specified.
		± 2	Water-surface elevation will be specified as a natural boundary condition at each node of the cross section. A negative value indicates that the water-surface elevation will be determined by the slope area method and that flow rate and energy slope have been specified.
		3	Supercritical flow exists at each node of the cross section. The cross section is assumed to form an outflow boundary, and water-surface elevation is not specified.

ZSEC RECORD 2 - FORMAT(16I5)

1..80	LXSZN(K)	+	A list of node numbers that define a connected series of element sides that represent the cross section. Nodes are entered in fields of five columns each. The list is terminated by a -1 entry. Up to 5 records (79 node points plus the -1 entry) may be used to define a cross section. The first entry for each record needs to be placed in the first field.
-------	----------	---	---

ZSEC

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
----------------	-----------------	--------------	--------------------

			Note -- During a time-dependent simulation, water-surface elevation at a cross section is not changed until a new ZSEC data set is read. Node points that define the cross section can be changed at any time during a time-dependent run by entering a new list of nodes. If nodes do not change, just enter a -1 in the first field.
--	--	--	--

Terminate the ZSEC data set with one or more blank data records.

Wind Data Set

Wind data records immediately follow a WIND data set identification record. One record is prepared for each node at which conditions other than general wind specifications are desired. For a time-dependent (unsteady) run, only values that change from the previous time step need to be specified. Wind condition data consist of the node number and the specified wind direction and velocity. The data set is terminated with one or more blank data records.

WIND RECORD - FORMAT(I10,2E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..10	N	+	Node number; needs to be less than or equal to the nodal array dimension.
11..20	WINDS(1,N)	+	Wind velocity at node N, in feet per second (meters per second).
21..30	WINDS(2,N)	+	Wind direction angle at node N, in degrees measured counterclockwise from the positive x-direction.

Terminate the WIND data set with one or more blank data records.

FLUX

Flow Check Data Set

Flow check data immediately follow a FLUX data set identification record. Flow check data consist of a list of node numbers that define a line of element sides across which total flow is to be computed. Flow across the first line is used as a base flow against which other calculated flows are compared. Flow-check lines may be composed of either straight or curved element sides. The data set is terminated with one or more blank data records.

FLUX RECORD - FORMAT(16I5)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..80	LFLUXN(K)	+	A list of node numbers that define a connected series of straight or curved element sides across which total flow is to be computed, entered in fields of five columns each. The line is terminated by a -1 entry. The first entry for each line needs to be placed in the first field. Use as many records as necessary to complete the line.

Note -- The only limit placed on the number of flow-check lines and the number of points that define a single line is that the total number of points in all lines may not exceed the LFLUXN array dimension.

Terminate the FLUX data set with one or more blank data records.

Weir Data Set

Weir data records immediately follow a WEIR data set identification record. One record is required for each weir segment. A weir data record contains the numbers of node points on the upstream and downstream sides of a weir segment, a discharge coefficient, the length of the weir segment, and the crest elevation of the weir segment. The data set is terminated with one or more blank data records.

WEIR RECORD - FORMAT(2I5,3E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..5	NOPW(1,J)	+	Number of the boundary node point on one side of the weir segment.
6..10	NOPW(2,J)	0	Flow is allowed to leave the system at the previously specified boundary node.
		+	Number of the boundary node point on the opposite side of the weir segment.
11..20	WCOF(J)	+	Dimensionless discharge coefficient for free-flow conditions at the weir segment (C_w in equation 4-20); usually about 0.53.
21..30	WLEN(J)	+	Length of the weir segment (L_w in equation 4-20) in feet (meters).
31..40	WCEL(J)	+	Crest elevation of the weir segment (z_c in equation 4-19) in feet (meters).

Terminate the WEIR data set with one or more blank data records.

CULV

Culvert Data Set

The culvert data record immediately follow a CULV data set identification record. One record is required for each culvert. A culvert data record contains the numbers of node points on the upstream and downstream ends of the culvert; a discharge coefficient; and the cross-section area, hydraulic radius, length, roughness coefficient, and the entrance invert elevation of the culvert. The data set is terminated with one or more blank data records.

CULV RECORD - FORMAT(2I5,6E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..5	NOPC(1,J)	+	Number of the boundary node at one end of the culvert. If the node number is negative, water only will be allowed to leave the network at this node point as if a flap-gate was installed on the other end of the culvert.
6..10	NOPC(2,J)	0	Water is allowed to leave the network at the previously specified boundary node.
		+	Number of the boundary node point at the other end of the culvert.
11..20	CCOF(J)	+	Dimensionless discharge coefficient for the culvert (C_c in equation 4-22 or equation 4-23).
21..30	CARE(J)	+	Cross-section area of the culvert (A_c in equation 4-22 or equation 4-23), in square feet (square meters).
31..40	CHYR(J)	+	Hydraulic radius of the culvert (R_c in equation 4-22), in feet (meters).
41..50	CLEN(J)	+	Length of the culvert (L_c in equation 4-22), in feet (meters).
51..60	CMAN(J)	+	Manning roughness coefficient for the culvert (n_c in equation 4-22).

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
61..70	CELV(J)	0	Type 4 flow is assumed.
		+	Entrance invert elevation of the culvert (z_{inv} in equation 4-21), in feet (meters). Type 5 flow is assumed.

Terminate the CULV data set with one or more blank data records.

TIME

Time-Dependent Data Set

The time-dependent data record immediately follow a TIME data set identification record. A time-dependent data set immediately precedes sets of boundary condition, total flow cross-section, water-surface elevation cross section, and wind data used in time-dependent (unsteady) simulations. A time-dependent data record contains the simulation time in hours at which the following data become effective. Time-dependent data sets and their associated boundary condition and/or wind data sets need to appear in chronological order at the end of the input data stream.

TIME RECORD - FORMAT(E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..10	TSNEW	+	Simulation time, in hours, at which the following boundary condition, total flow cross section, water-surface elevation cross section, or wind data specifications become effective.

Analysis of Output Module: ANOMOD

Data sets read by ANOMOD are preceded by the following identification records:

DATA SET IDENTIFICATION RECORD - FORMAT(A10,7E10.0)

<u>Variable IDS</u>	<u>Data description</u>
SWMS ¹	Program control data
ELEM	Element data
NODE or NODEA	Node data (primary)
NODEB	Node data (secondary)
FLOW or FLOWA	Flow-field data (primary)
FLOWB	Flow-field data (secondary)
	<u>Operation data:</u> ²
GRID	Network plot data
VECT	Vector plot data
CONT	Contour plot data
FLUX	Flow-check plot data
HIST	Time-history report and plot data
LAST ³	End of data

¹Required first data set.

²Operation data sets follow all other data sets.

³Needs to be the last record in the input data stream.

SWMS

Program Control Data Set

Program control data records immediately follow an SWMS data set identification record. The identification record contains the data set identification code and a code that controls the printed output format. The remaining record in the data set contains a code that controls printed output, input file specifications, and a code that identifies the units that are used.

SWMS IDENTIFICATION RECORD - FORMAT(A4,6X,2E10.0,30X,2E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..4	IDS	SWMS	Program control data set identification code.
5..10	--	--	Not used.
11..20	WIDE	0	132-column format will be used for all printed output.
		+	80-column format will be used for all printed output.
21..30	SCREEN	0	Messages that describe program operations will not be written to the screen as the program is executing.
		1	Messages that describe program operations will be printed to the screen as the program is executing.
31..60	--	--	Not used.
61..70	TFIRST	0 or +	Simulation time corresponding to the first time-step (for time-dependent solution data) for which flow data is to be printed. Enter 0 for steady-state solution data.
71..80	TLAST	0 or +	Simulation time corresponding to the last time-step (for time-dependent solution data) for which flow data is to be printed. TLAST needs to be greater than or equal to TFIRST. Enter 0 for steady-state solution data.

SWMS RECORD 1 - FORMAT(6I5)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..5	IPRNT	0..15	The sum of the following printed output options that are desired:
			<u>Options</u>
		0	Control data and error messages will be printed.
		1	Input data read from records will be echo printed.
		2	Network (node and element) data will be printed.
		4	Primary flow data will be printed.
		8	Secondary flow data will be printed.
		16	Froude number at nodes will be calculated and reported if flow data are printed.
		32	Energy heads at nodes will be calculated and reported if flow data are printed.
			Note -- Time-dependent flow data will be printed for all time steps between simulation times TFIRST and TLAST entered on the SWMS data set identification record.
6..10	IGRID	0	A primary network data file will not be read. All primary network data will be entered on data records.
		+	A primary network data file will be read. The default name of the primary network data file is "GRID.DAT".

SWMS

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
11..15	IFLOW	0	A primary flow data file will not be read. All primary flow data data will be entered on data records.
		+	A primary flow data file will be read. The default name of the primary flow data file is "FLOW.DAT".
16..20	GRIDB	0	A secondary network data file will not be read. All secondary network data will be entered on data records.
		+	A secondary network data file will be read. The default name of the secondary network data file is "GRID_B.DAT".
21..25	IFLOWB	0	A secondary flow data file will not be read. All secondary flow data will be entered on data records.
		+	A secondary flow data file will be read. The default name of the secondary flow data file is "FLOW_B.DAT".
26..30	IUNITS	0	U.S. customary units will be used.
		1	International System (S.I.) units will be used.

Note -- The value of IUNITS determines what units to use in printed report headings and on graph axis labels.

Element Data Set

Element data records immediately follow an ELEM data set identification card. One record is required for each element. An element data record contains the element number, the sequence of nodes connected to the element (the element connectivity list), the element type code, and the element assembly sequence. The data set is terminated with one or more blank data records.

ELEM RECORD -FORMAT(12I5)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..5	L	+	Element number; needs to be less than or equal to the element array dimension.
6..50	LNODES(K,L)	+	Element connectivity list (that is, the sequence of nodes connected to the element. Enter 6 node numbers for a triangular element, 8 node numbers for a "serendipity" quadrangular element, or 9 node numbers for a "Lagrangian" quadrangular element. Begin the list at any corner node and proceed in a counterclockwise direction around the element. For a 9-node quadrangular element, the center node is entered last.
51..55	LTYPES(L)	±	Element type code. The property type code corresponds to a set of element parameters (coefficients) that are entered in the PROP data set in FLOMOD. If LTYPES(L) < 1, the element will be turned off (that is, it will not be used).
56..60	LASEQS(L)	+	Element assembly sequence. The sequence number of the element for processing by the frontal method. An efficient element sequence can be generated using the resequencing option.

Terminate the ELEM data set with one or more blank data records.

NODE

Node Data Set

Node data records immediately follow an NODE, NODEA or NODEB data set identification record. Primary node data are preceded by an NODE or an NODEA data set identification record and secondary node data (used to contour differences in water-surface elevation between two simulations when ground elevations have changed) are preceded by an NODEB data set identification record. The identification record contains the data set identification code and factors that are used to convert node point coordinates and ground and ceiling elevations read from records to the desired units (either feet or meters). One node data record is required for each corner node. Curved element sides may be specified by also entering coordinate data for the midside node of an element side. A node data record contains the node number, values of measured coordinates, and the ground-surface and ceiling elevations at the node point. The data set is terminated by one or more blank records.

NODE IDENTIFICATION RECORD - FORMAT(A10,6E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..4	IDS	NODE	Node data set identification code.
5..10	--	--	Not used.
11..20	XFACT	+	Multiplication factor used to convert x coordinates read from data records to feet (meters). The default value is 1.0.
21..30	YFACT	+	Multiplication factor used to convert y coordinates read from data records to feet (meters). The default value is 1.0.
31..40	ZFACT	+	Multiplication factor used to convert ground and ceiling elevations read from data records to feet (meters). The default value is 1.0.
41..50	XZERO	+	Feet (meters) to be added to all x coordinates read from data records after they are multiplied by XFACT. The default value is 0 ft (0 m).

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
51..60	YZERO	+	Feet (meters) to be added to all y coordinates read from data records after they are multiplied by YFACT. The default value is 0 ft (0 m).
61..70	ZZERO	+	Feet (meters) to be added to all ground and ceiling elevations read from data records after they are multiplied by ZFACT. The default value is 0 ft (0 m).

NODE RECORD - FORMAT(I10,4E10.0)

1..10	N	+	Node number; needs to be less than or equal to the nodal array dimension.
11..20	XYZZ(1,N)	+	X coordinate of node N.
21..30	XYZZ(2,N)	+	Y coordinate of node N.
31..40	XYZZ(3,N)	+	Ground-surface elevation at node N.
41..50	XYZZ(4,N)	+	Ceiling elevation at node N (zero if there is no ceiling at the node).

Note -- Coordinates and elevations are converted to feet (meters) using the factors specified on the data set identification record. Thus, any system of units can be used to record coordinates and elevations.

Terminate the NODE data set with one or more blank data records.

FLOW

Flow Data Set(s)

Flow field data records immediately follow an FLOW, FLOWA, or FLOWB data set identification record. Primary flow data are preceded by an FLOW or an FLOWA data set identification record and secondary flow data (used to plot differences in water-surface elevation between two simulations) are preceded by a FLOWB data set identification record. One record is required for each node point for which values are to be entered. A flow data record contains the node number, the x and y direction depth-averaged velocities, and the flow depth at the node point. These data represent either a steady-state solution or the solution at the end of one time-step of a time-dependent (unsteady) simulation. Only one set of flow data may be entered on records in a single run. Flow data will have to be read from a file for operations that require solutions at the end of more than one time-step. The data set is terminated with one or more blank data records.

FLOW RECORD - FORMAT(I10,3E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1 to 10	N	+	Node number; needs to be less than or equal to the nodal array dimension.
11..20	VNEWS(1,N)	+	Depth-averaged velocity in the x direction at node N, in feet per second (meters per second).
21..30	VNEWS(2,N)	+	Depth-averaged velocity in the y direction at node N, in feet per second (meters per second).
31..40	VNEWS(3,N)	+	Water depth at node N, in feet (meters).

Terminate the FLOW data set with one or more blank data records.

Network Plot Data Set

Network plot data immediately follow a GRID data set identification record. Network plot data control plotting of the finite element grid. Entered data include a plot title; codes that control the values to be plotted at node points and element centroids; and plot scale and transformation data.

GRID IDENTIFICATION RECORD - FORMAT(A10,7E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..10	IDS	GRID	Data set identification code.
11..70	--	--	Not used.
71..80	NEWPLT	0	A new origin will be established for the plot.
		1	The current origin will be retained. A combined plot will result if a plot was drawn previously.

GRID RECORD 1 - FORMAT(A80)

1..80	PTITLE	a/n	Alphanumeric characters to be used as the plot title.
-------	--------	-----	---

GRID RECORD 2 - FORMAT(2I5)

1..5	IPLTN	0..7	The sum of the following plot options that are desired:
------	-------	------	---

Options

0	Nothing will be plotted at node points.
1	Node numbers will be plotted at node points.
2	Ground-surface elevations will be plotted at node points.
4	Ceiling elevations will be plotted at node points.

GRID

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
6..10	IPLTE	0..7	The sum of the following plot options that are desired: <u>Options</u>
		0	Nothing is to be plotted at element centroids.
		1	Element numbers will be plotted at element centroids.
		2	Element type codes will be plotted at element centroids.
		4	Element sequences will be plotted at element centroids.

GRID RECORD 3 - FORMAT(3E10.0)

1..10	HTT	+	Height in inches of the title to be plotted along the y-axis. The default value is 0.14 inches.
11..20	HTN	+	Height in inches of numbers to be plotted at network node points. The default value is 0.07 inches.
21..30	HTE	+	Height in inches of numbers to be plotted at element centroids. The default value is 0.105 inches.

GRID RECORD 4 - FORMAT(7E10.0)

1..10	XLOWER	+	The x coordinate in feet (meters) before rotation of the point that will be the lower lefthand corner of the network plot after it is rotated ROTATE degrees.
11..20	YLOWER	+	The y coordinate in feet (meters) before rotation of the point that will be the lower lefthand corner of the network plot after it is rotated ROTATE degrees.

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
21..30	XOFFSET	+	The x direction offset in feet (meters) used to determine the upper righthand corner of the network plot. The default value is 1.0E+35 (that is, everything to the right of the lower lefthand corner will be plotted)
31..40	YOFFSET	+	The y direction offset in feet (meters) used to determine the upper righthand corner of the network plot. The default value is 1.0E+35 (that is, everything above the lower lefthand corner will be plotted).
41..50	XSCALE	+	The x coordinate plot scale in feet (meters) per inch.
51..60	YSCALE	+	The y coordinate plot scale in feet (meters) per inch.
61..70	ROTATE	+	Angle of plot rotation in degrees measured counterclockwise.

VECT

Vector Plot Data Set

Vector plot data immediately follow a VECT data set identification record. Vector plot data control plotting of velocity and unit flow vectors. Entered data include a title to be plotted, codes that control the vector and scalar values to be plotted at node points, character size specifications, and plot scale and transformation data.

VECT IDENTIFICATION RECORD - FORMAT(A10,7E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..10	IDS	VECT	Data set identification code.
11..20	STIME	+	The simulation time in hours of the desired time-step in the primary flow data file. The default value is 0.0 hr.
21..30	--	--	Not used.
71..80	NEWPLT	0	A new origin will be established for the plot.
		1	The current origin will be retained. If a plot was drawn previously, a combined plot will result.

VECT RECORD 1 - FORMAT(A80)

1..80	PTITLE	a/n	Alphanumeric characters to be used as the plot title.
-------	--------	-----	---

VECT RECORD 2 - FORMAT(2I5)

1..5	IPLTV	0	Vectors will not be plotted.
		1	Velocity vectors will be plotted.
		2	Unit flow vectors will be plotted.

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
6..10	IPLTS	0..7	The sum of the following scalar values to be plotted at node points:
			<u>Options</u>
		0	No scalar values will be plotted.
		1	Node numbers.
		2	Velocity magnitude in feet (meters) per second.
		4	Unit flow magnitude in square feet (meters) per second.

VECT RECORD 3 - FORMAT(4E10.0)

1..10	VSCALE	+	Vector scale; feet (meters) per second per inch for velocity, or square feet (meters) per second for unit flow.
		-	Vector scale; negative value of the length, in inches, of the longest vector. All other vectors will be scaled accordingly.
			■ Note -- The default value is -1.0.
11..20	VLMIN	+	Minimum length vector to be plotted, in inches. The default value is 0 inches (that is, all vectors will be plotted).
21..30	AHTYP	+	Vector type code. This is a single digit code used to specify the type of vector as shown below:

Vector type

1	—————▶	(default)
2	—————>	
3	●—————	

VECT

VECT RECORD 4 - FORMAT(3E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
31..40	AHLEN	+	Arrowhead length or dot diameter, in inches, for the vectors shown above. The default value is 0.07 in.
1..10	HTT	+	Height in inches of the title to be plotted along the y-axis. The default value is 0.14 in.
11..20	HTN	+	Height in inches of numbers to be plotted at network node points. The default value is 0.07 in.

VECT RECORD 5 - FORMAT(7E10.0)

1..10	XLOWER	+	The x coordinate in feet (meters) before rotation of the point that will be the lower lefthand corner of the network plot after it is rotated ROTATE degrees.
11..20	YLOWER	+	The y coordinate in feet (meters) before rotation of the point that will be the lower lefthand corner of the network plot after it is rotated ROTATE degrees.
21..30	XOFFSET	+	The x direction offset in feet (meters) used to determine the upper righthand corner of the network plot. The default value is 1.0E+35 (that is, everything to the right of the lower lefthand corner will be plotted).
31..40	YOFFSET	+	The y direction offset in feet (meters) used to determine the upper righthand corner of the network plot. The default value is 1.0E+35 (that is, everything above the lower lefthand corner will be plotted).
41..50	XSCALE	+	The x coordinate plot scale in feet (meters) per inch.
51..60	YSCALE	+	The y coordinate plot scale in feet (meters) per inch.

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
61..70	ROTATE	+	Angle of plot rotation in degrees measured counterclockwise.

CONT

Contour Plot Data Set

Contour plot data immediately follow a CONT data set identification record. Contour plot data control plotting of ground-surface and water-surface elevation contours. Input data include a title to be plotted, codes that control values for which contour lines are to be drawn, character size specifications, and plot scale and transformation data.

CONT IDENTIFICATION RECORD - FORMAT(A10,E10.0,50X,E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..10	IDS	CONT	Data set identification code.
11..20	STIME	+	Simulation time, in hours, of the desired time-step in the primary flow data file. The default value is 0.0 hr.
21..30	--	--	Not used.
71..80	NEWPLT	0	A new origin will be established for the plot.
		1	The current origin will be retained. If a plot was drawn previously, a combined plot will result.

CONT RECORD 1 - FORMAT(A80)

1..80	PTITLE	a/n	Alphanumeric characters to be used as the plot title.
-------	--------	-----	---

CONT RECORD 2 - FORMAT(3I5)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..5	IPLTC	1..7	The sum of the following plot options that are desired:
			<u>Options</u>
		1	A contour plot of the value specified by the variable ICVAL will be drawn.
		2	Node numbers will be plotted at all node points.
		4	Contour values will be plotted at all node points.
6..10	ICVAL	±1	Ground-surface elevation will be contoured.
		±2	Water-surface elevation or difference in water-surface elevation will be contoured.
		±3	Velocity magnitude or difference in velocity magnitude will be contoured.
		±4	Froude number or difference in Froude number will be contoured.
		±5	Total energy head or difference in total energy head will be contoured.
			Note -- Differences are contoured if ICVAL is negative. Differences are computed by subtracting secondary flow values from primary flow values.
11..15	NDECV	+	Number of decimal places to be included in contour line labels. A contour line is labeled where the line intersects the network boundary.
		0	Contour lines will be labeled with an integer value.
		-1	Contour lines will not be labeled.

CONT

CONT RECORD 3 - FORMAT(3E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..10	CVMIN	+	Minimum value to be contoured. Any multiple of the contour interval that is less than this value will be ignored.
11..20	CVMAX	+	Maximum value to be contoured. Any multiple of the contour interval that is greater than this value will be ignored.
			Note -- If CVMIN and CVMAX are both left blank, or if CVMAX is less than or equal to CVMIN, then CVMAX is set to 99999, and CVMIN is set to -99999.
21..30	CVINC	+	Contour interval in feet (meters). If CVINC is not specified, a value will be calculated on the basis of the range of the values to be contoured.

CONT RECORD 4 - FORMAT(3E10.0)

1..10	HTT	+	Height in inches of the title to be plotted along the y-axis. The default value is 0.14 in.
11..20	HTN	+	Height in inches of numbers to be plotted at network node points. The default value is 0.07 in.
21..30	HTC	+	Height in inches of contour line labels. The default value is 0.105 in. Labels are plotted alongside the network boundary where the boundary is intersected by a contour line. The label is the value that the contour line represents. The number of decimal places to be included in the label is controlled by the variable NDECV on the CONT.2 record.

CONT RECORD 5 - FORMAT(7E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..10	XLOWER	+	The x coordinate in feet (meters) before rotation of the point that will be the lower lefthand corner of the network plot after it is rotated ROTATE degrees.
11..20	YLOWER	+	The y coordinate in feet (meters) before rotation of the point that will be the lower lefthand corner of the network plot after it is rotated ROTATE degrees.
21..30	XOFFSET	+	The x direction offset in feet (meters) used to determine the upper righthand corner of the network plot. The default value is 1.0E+35 (that is, everything to the right of the lower lefthand corner will be plotted).
31..40	YOFFSET	+	The y direction offset in feet (meters) used to determine the upper righthand corner of the network plot. The default value is 1.0E+35 (that is, everything above the lower lefthand corner will be plotted).
41..50	XSCALE	+	The x coordinate plot scale in feet (meters) per inch.
51..60	YSCALE	+	The y coordinate plot scale in feet (meters) per inch.
61..70	ROTATE	+	Angle of plot rotation in degrees measured counterclockwise.

FLUX

Flow Check Report and Plot Data Set

Flow check report and plot data immediately follow a FLUX data set identification record. Flow check data control the reporting and plotting of flow-check lines and the total flow across them. Input data include codes that control the reporting and plotting of flow checks, plot scale and transformation data, and lists of node numbers that define lines of element sides across which total flow is to be computed. In the printed report, flow across the first line is used as a base discharge against which other calculated flows are compared. Flow-checklines may be composed of either straight or curved element sides. The data set is terminated with one or more blank data records.

FLUX IDENTIFICATION RECORD - FORMAT(A10,7E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..10	IDS	FLUX	Data set identification code.
11..20	STIME	+	Simulation time in hours of the desired time-step in the flow data file. The default value is 0.0 hours.
21..30	--	--	Not used.
71..80	NEWPLT	0	A new origin will be established for the plot.
		1	The current origin will be retained. If a plot was drawn previously, a combined plot will result.

FLUX RECORD 1 - FORMAT(A80)

1..80	PTITLE	a/n	Alphanumeric characters to be used as the plot title.
-------	--------	-----	---

FLUX RECORD 2 - FORMAT(3I5)

1..5	IPRTF	0	A flow check report will not be printed.
		1	A flow check report will be printed.

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
6..10	IPLTF	0	Flow-check lines will not be plotted.
		1	Flow-check lines will be plotted.
11..15	NDECF	±	Number of decimal places to be included in flow-check line labels. Flow-check lines are labeled with the computed total flow across the line. Enter a -1 to plot only integer values.

FLUX RECORD 3 - FORMAT(2E10.0)

1..10	HTT	+	Height in inches of the title to be plotted along the y-axis. The default value is 0.14 inches.
11..20	HTN	+	Height in inches of numbers to be plotted at network node points. The default value is 0.07 inches.
21..30	HTF	+	Height in inches of flow-check line labels. The default value is 0.105 inches.

FLUX RECORD 4 - FORMAT(7E10.0)

1..10	XLOWER	+	The x coordinate in feet (meters) before rotation of the point that will be the lower lefthand corner of the network plot after it is rotated ROTATE degrees.
11..20	YLOWER	+	The y coordinate in feet (meters) before rotation of the point that will be the lower lefthand corner of the network plot after it is rotated ROTATE degrees.
21..30	XOFFSET	+	The x direction offset in feet (meters) used to determine the upper righthand corner of the network plot. The default value is 1.0E+35 (that is, everything to the right of the lower lefthand corner will be plotted).

FLUX

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
31..40	YOFFSET	+	The y direction offset in feet (meters) used to determine the upper righthand corner of the network plot. The default value is 1.0E+35 (that is, everything above the lower lefthand corner will be plotted).
41..50	XSCALE	+	The x coordinate plot scale in feet (meters) per inch.
51..60	YSCALE	+	The y coordinate plot scale in feet (meters) per inch.
61..70	ROTATE	+	Angle of plot rotation in degrees measured counterclockwise.

FLUX RECORD 5 - FORMAT(16I5)

1..80	LINE(K)	+	A list of node numbers defining a connected series of straight or curved element sides across which total flow is to be computed. The line is terminated with a -1 entry. Use as many records as needed to complete the line. Repeat the listing for each flow-check line to be plotted and/or reported.
-------	---------	---	--

Note -- The only limit placed on the number of flow-check lines and the number of points in each line is that the total number of points in all lines cannot exceed the LINE array dimension.

Terminate the FLUX data set with one or more blank data records.

Time-History Report and Plot Data Set

Time-history report and plot data immediately follow a HIST DATA set identification record. Time-history report and plot data control reporting and plotting of time-dependent values of velocity, unit flow, and stage (water-surface elevation) at a node point. Time-history data are plotted in the form of 8.5-in by 11.0-in graphs of the value versus time. For the vector quantities velocity and unit flow, both the magnitude and direction are plotted. Direction is defined as the angle between the vector and the positive x axis. Time-history data consist of values that control reporting and plotting, a list of node numbers for which data are desired, and a code that specifies the data to be reported or plotted. The data set is terminated with one or more blank data records.

HIST IDENTIFICATION RECORD - FORMAT(A10,2E10.0)

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
1..10	IDS	HIST	Data set identification code.
11..20	TFIRST	+	Simulation time, in hours, of the first time-step in the primary flow data file to be reported or plotted.
21..30	TLAST	+	Simulation time, in hours, of the last time-step in the primary flow data file to be reported or plotted.

HIST RECORD - FORMAT(2I10)

1..10	NTHP(K)	+	Number of the node point for which a time-history report or plot is desired.
11..16	--	--	Not used.
17	KTHP(K)	0	A time-history report will not be printed.
		1	A time-history report from time TFIRST to TLAST will be printed.
18	KTHP(K)	0	A time-history plot of velocity will not be drawn.

TIME

<u>Columns</u>	<u>Variable</u>	<u>Value</u>	<u>Description</u>
		1..2	A time-history plot of velocity from time TFIRST to time TLAST will be drawn. If the entry is 2, a square will be drawn around each data point that is plotted.
19	KTHP(K)	0	A time-history plot of unit flow will not be drawn.
		1..2	A time-history plot of unit flow from time TFIRST to time TLAST will be drawn. If the entry is 2, a square will be drawn around each data point that is plotted.
20	KTHP(K)	0	A time-history plot of stage (water-surface elevation) will not be drawn.
		1..2	A time-history plot of stage (water-surface elevation) from time TFIRST to time TLAST will be drawn. If the entry is 2, a square will be drawn around each data point that is plotted.

Note -- TFIRST and TLAST are the simulation times corresponding to the first and last time-steps for which data is to be reported or plotted. TFIRST and TLAST are entered on the HIST data set identification record.

Terminate the TIME data set with one or more blank data records.

Section 12

OUTPUT DATA DESCRIPTION

This section describes in detail the output data produced by each module including the general output structure, the quantity of output (both mandatory and optional output), and the structure of output data files.

Printed output in each module is directed to Fortran input/output (I/O) unit number IO6 which is set equal to 6 in the BLOCK DATA subprogram of each module. The I/O unit number can be easily redefined if necessary. Printed output can also be requested in a 132-column wide format (the default) or an 80-column wide format. The same information is printed in both formats, but the 132-column wide output is condensed and will cause less paper to be used.

Data Input Module: DINMOD

The Data Input Module provides a variety of output control options. A network data file that contains node and element information can be created, and the processed network data can be plotted.

Printed Output

A title block that contains the program name and version number is printed at the start of each run followed by the program control data. Remaining input data records will be echo printed (that is, printed immediately after being read) only if requested.

Computations are performed only during element resequencing. The resequencing parameters maximum-frontwidth and sequence-sum (sum of frontwidth-squared) are printed for each resequencing.

The final element assembly sequence also is printed. Elements are listed in the order in which they are to be processed by the frontal solver in FLOMOD.

Because node, element, and property type data are stored in arrays according to the node, element, and property type number, the numbers of these items cannot exceed the dimensions of the associated arrays. As data are read from records, the node, element, and property type numbers are checked to insure compliance with this requirement. When a number is found that exceeds the associated array dimension, an error message is printed.

When a network is constructed and debugged, it is good practice to have all input data echo printed because any error messages concerning exceedance of array dimensions will appear in the printed output immediately after the data item that caused the error. Correcting a problem data item is then a simple matter.

Messages that describe other errors detected during data processing will be printed. Usually the appropriate action to take to rectify the problem will be obvious. Further explanation of errors and suggested corrective action can be found in the section titled "Trouble Shooting."

The final processed element and node data are printed in report form. Element data consist of the element number, connectivity list, property type code, and the computed area for each element in the system. Node data consist of the node number, x and y coordinates, ground-surface elevation, and ceiling elevation for each node point.

Output Data Files

Processed node and element data can be stored in a network data file that can be used in future runs of DINMOD and by other FESWMS-2DH programs. Network data can be written to either a formatted or an unformatted file. Unformatted data is read much faster than formatted data, but a formatted data file can be printed and edited easily. Unformatted data files are most useful when a finite element network is large and the network data file is read repeatedly. Formatted data files are most useful when a network is being constructed and debugged. It is a simple matter for DINMOD to read a formatted network data file and write an unformatted network data file, or vice versa.

Graphic Output

A finite element network plot can be created that includes node numbers, ground and ceiling elevations at node points, element numbers, element property type codes, and element assembly sequences. Although extensive error checks are conducted by DINMOD, a network cannot be accepted as error-free until a close visual inspection of the plotted network has been performed. Graphic output is written to a plotfile in the form of plotting instructions. Instruction descriptions are presented in a following section.

Depth-averaged Flow Module: FLOMOD

The Depth-Averaged Flow Module provides a variety of output options. A flow data file that contains computed depth-averaged horizontal velocities, water-surface elevation, and time-derivatives of these quantities at all node points in a network can be written. A restart/recovery file that contains intermediate output that can be used to restart a run if the run has been terminated abnormally can also be written.

Printed Output

A title block that contains the program name and version number is printed at the start of each run followed by the program control data. Remaining input data read from records will be echo printed only if requested.

After data for a steady-state solution or an initial time-step for a time-dependent solution have been read, general system parameters are printed. General system parameters include the largest node number in a network, the largest element number in a network, largest property type code, the number of weir segments, the number of culverts, water density, Coriolis parameter, air density, and wind stresses in the x and y directions.

Network data, boundary condition data, wind data, and initial condition data are printed next if requested. Network data consist of element data (the element number, connectivity list, property type code, assembly sequence, and area) and node data (the node number, x and y coordinates, ground-surface elevation, and ceiling elevation). Boundary condition data include the node number, boundary condition code, and boundary specifications (velocity or unit flow in the x and y directions or the tangential and normal directions, and water-surface elevation). Wind data consist of wind stresses in the x and y directions at each node point. Initial condition data consist of the node point number and the x and y velocities, depth and corresponding water-surface elevation, and time derivatives of these quantities that are assigned at each node at the beginning of computations.

An element assembly sequence (that is, the order in which elements are to be processed by the frontal solver) and a degree-of-freedom array are printed next if requested. The degree-of-freedom array relates an equation number to each nodal degree-of-freedom (variable). Degree-of-freedom array information usually is not needed and will take up a lot up storage space for a large

network, but it can be used to help identify subtle problems in network geometry, or boundary or initial conditions that have been specified incorrectly.

A flow check based on initial conditions is printed next if flux line data have been read. The flow check consists of computed flows across a line of connected element sides defined by a list of node numbers. For each flux line, the total flow across the section in the x and y directions, and the total net flow across the section are printed. Total flow across the first flux line is used as a base flow against which the remaining flows are compared as a percentage. The percentage can be used to check quickly how well mass is conserved (continuity) in a steady-state solution.

Solution parameters are printed next. Solution parameters include the number of elements (that is, elements for which a property type code has been specified), the number of equations to be formed if all elements are active, the maximum (MAX) frontwidth, the root-mean-square (RMS) frontwidth, and the average (AVG) frontwidth. Maximum frontwidth determines the size of the active array required during equation solution.

The variable MAVAIL (available active array memory) in FLOMOD needs to be greater than or equal to MAX frontwidth squared, otherwise an error message will be printed and the run terminated. Computation time required for each iteration is proportional to the value of RMS frontwidth times the number of equations.

At the conclusion of an iteration, the number of buffer blocks written during the frontal solution is printed followed by convergence parameters. Convergence parameters include the average change of each of the solution variables, the single largest change for each of the solution variables, and the node points at which the largest changes occurred. When starting

cold, or after major changes in network geometry or boundary conditions, two or three full-Newton iterations may be needed before a user can determine whether or not the solution is converging. Maximum changes in any of the variables that occur repeatedly at the same node point or even at node points in the same area of a network may indicate that the network is poorly defined in the area of the node point(s), or that boundary conditions need to be adjusted.

A solution is printed after selected iterations and at the last iteration of a steady-state run, and only for selected time steps of a time-dependent run. Nodal solution values of depth-averaged velocities, depth of flow, and water-surface elevation at each node point in a network are printed. Water-surface elevation is determined by adding the computed flow depth to the ground-surface elevation at a node point.

One-dimensional weir segment and culvert information is printed immediately following the nodal solution values. For each weir segment, the segment number, connecting nodes and corresponding total energy heads and water-surface elevations, the total flow over the segment, and the weir submergence factor are printed. For each culvert, the culvert number, connecting nodes and corresponding water-surface elevations, and the total flow through the culvert are printed.

Following the final iteration of a steady-state solution and after the final iteration of each printed time step in a time-dependent solution, Froude numbers at node points, energy heads at node points, and element continuity norms are printed if requested. Froude numbers greater than one indicate supercritical flow conditions and are flagged by printing an asterisk after the value. In both Froude number and energy head reports, nodes at which pressure flow exist are flagged by an asterisk after the value of flow depth. Continuity norms greater than a user-defined value are flagged by printing an asterisk

next to the value. A large continuity norm indicates an element where the absolute value of the continuity equation residual is much larger than zero. Groups of elements that have large continuity norms indicate areas of the network that may need to be refined by adding more elements and node points.

Another flow check is printed after the last iteration of a steady-state run and at the end of each printed time step in a time-dependent run.

Output Data Files

Two output data files can be generated by FLOMOD: (1) A restart/recovery file, and (2) a flow data file.

Restart/recovery data file

A restart/recovery data file contains intermediate solution results that can be used to restart a run that has been terminated abnormally. A restart/recovery file can be either a formatted file or an unformatted file.

Flow data file

A flow data file contains the results from the last iteration of a steady-state solution and from the last iteration of every time step of a time-dependent solution. Flow data can be written to either a formatted or an unformatted file.

Analysis of Output Module: ANOMOD

The Analysis of Output Module provides printed reports and plots that allow a user to analyze the results from a simulation.

Printed Output

A title block that contains the program name and version number is printed at the start of each run followed by program control data. Remaining input data read from records will be echo printed if requested.

A network data report consisting of all node and element data is printed next if requested. A flow data report consisting of the depth-averaged velocities in the x and y directions, flow depth, and water-surface elevation at all node points is then printed if requested. Time-history data reports are generated when an HIST data set is read and a printed report is requested. Time-history data reports consist of a summary of the desired quantity (velocity, unit flow, or depth) at the end of each time step during a specified simulation time period.

Graphic Output

Graphic output consists of plots of a finite element network (elements, element values, and node point values), velocity vectors, unit flow vectors, contours of ground-surface and water-surface elevations, flow check lines and the computed total flow across them, and time-history plots of velocity, unit flow, and stage (water-surface elevation). Network, vector, contour, and flow check plots can be combined by plotting them on top of one another. When a combined plot is requested, the boundary of a network is drawn only for the first plot and only the title of the first plot is drawn. Graphic output in the form of plotting instructions is written to a plotfile. Plotting instructions are described in a following section.

Plotfile Format

Graphic output is written to a plotfile in the form of plotting instructions. The plotting instructions conform to the Hewlette-Packard Graphics Language (HPGL) instruction set. An HPGL instruction consists of a two-letter mnemonic followed by its parameter field, if any, and a character that terminates the instruction. Parameters that follow an instruction are separated from each other by a comma. A semicolon is used to terminate all instructions except the label instruction, which defines a character string to be plotted. ASCII character ETX (decimal value 3) is used to terminate a label instruction.

Instruction Description

Plotting instructions that are written to FESWMS-2DH plotfiles are described in this section. Each instruction is listed with its syntax, purpose, and parameters. Parameters contained within parentheses are optional.

The Initialize Instruction, IN

Syntax: IN;

Purpose: The first instruction written to a plotfile. It indicates that the plotting device needs to be initialized.

Parameters: None.

The Input P1 and P2 Instruction, IP

Syntax: IP;

Purpose: Sets plot scaling points. This instruction is used to denote the start of a new plot contained in the plot file.

Parameters: None.

The Pen Down Instruction, PD

Syntax: PD (X, Y);

Purpose: Sets the pen status to down and moves to the coordinates X,Y (if they are given) with the pen in a down position.

Parameters: X, Y - a pair of decimal numbers that represent coordinates in plotter units (there are 1021 plotter units per inch).

The Pen Up Instruction, PU

Syntax: PU (X, Y);

Purpose: Sets the pen status to up and moves to the coordinates X,Y (if they are given) with the pen in an up position.

Parameters: X, Y - a pair of decimal numbers that represent coordinates in plotter units (there are 1021 plotter units per inch).

The Plot Absolute Instruction, PA

Syntax: PA X, Y;

Purpose: Plots to the X,Y coordinates given using the current pen up/down status.

Parameters: X, Y - a pair of decimal numbers that represent coordinates in plotter units (there are 1021 plotter units per inch).

The Absolute Direction Instruction, DI

Syntax: DI Run, Rise;

Purpose: Sets the direction in which labels are drawn.

Parameters: Run, Rise - decimal values where

$$\begin{aligned} \text{Run} &= \sin r , \\ \text{Rise} &= \cos r , \end{aligned}$$

and r is the label angle measured counter-clockwise from the horizontal x axis.

The Absolute Character Size Instruction, SI

Syntax: SI Width, Height;
Purpose: Sets character width and height for labels.
Parameters: Width, Height - decimal values that represent the size of a character in centimeters.

The Label Instruction, LB

Syntax: LB Characters ETX
Purpose: Draws a character string.
Parameters: Characters - an ASCII character string that is terminated with ASCII character ETX (decimal value 3).

The Pen Select Instruction, SP

Syntax: SP Pen Number;
Purpose: Selects or stores a pen. This instruction is used to separate plots contained in the same plotfile. The pen is stored at the end of each plot by sending the instruction "SP0;" (that is, a pen number of 0 is specified).
Parameters: Pen Number - the number of the pen to be used.

The Shade Wedge Instruction, WG

Syntax: WG Radius, Start Angle, Sweep Angle;
Purpose: Defines and shades an arc segment of a circle of a specified radius centered at the current pen position. This instruction is used to define a solid filled dot that denotes the base of type 3 vectors.
Parameters: Radius - radius of the wedge (dot) in inches.
Start Angle - start angle of the wedge in degrees (always 0).
Sweep Angle - arc angle of the wedge in degrees (always 360).

This page is blank.

Section 13

RUN-PREPARATION INSTRUCTIONS

Procedures for organizing input data in preparation to execute FESWMS-2DH modules are described in this section.

Run-Stream Description

A tabular representation of the data records that constitute an input run-stream for each FESWMS-2DH module is provided in tables 13-1 to 13-3. An input run-stream includes all control records and data records ordered in a proper sequence. The interactive data file editing program that accompanies each FESWMS-2DH module can be used to enter all input data which is then automatically ordered in a proper sequence and stored in an input data file.

Resource Requirements

Computer memory and mass storage required to run FESWMS-2DH depend on the size of dimensioned arrays, which dictate the maximum size finite element network that can be modeled. FLOMOD has the largest memory and storage requirements of any of the FESWMS-2DH modules. A network that consists of up to 400 elements and 1600 nodes, and has a maximum frontwidth of up to 150 degrees-of-freedom (about 60 nodes) can be modeled using a microcomputer with approximately 570 kilobytes of random access memory available for program execution. Larger networks will require larger amounts of computer memory and disk storage space. The modules can be easily redimensioned to allow networks that have more than 500 elements and 1500 nodes to be modeled.

Table 13-1. Run-stream representation for the Data Input Module:
DINMOD.

Data set ID and record number	Data items entered on the record
SWMS.ID	IDS, WIDE, SCREEN
SWMS.1	JTITLE
SWMS.2	IPRNT, IPLOT, ICHEK, IAUTO, IREFN, IRESQ, IGIN, IGOUT
PLOT.ID	IDS
PLOT.1	PTITLE
PLOT.2	IPLTN, IPLTE
PLOT.3	HTT, HTN, HTE, HTC, CVINC
PLOT.4	XLOWER, YLOWER, XOFFSET, YOFFSET, XSCALE, YSCALE, AROT
ELEM.ID	IDS
ELEM.1	L, LNODES(1,L), LNODES(2,L), ..., LNODES(9,L), LTYPES(L), LASEQS(L)
.	One ELEM.1 record for each element
.	
(BLANK)	
NODE.ID	IDS, XFACT, YFACT, ZFACT, XZERO, YZERO, ZZERO
NODE.1	N, XYZZ(1,N), XYZZ(2,N), XYZZ(3,N), XYZZ(4,N)
.	One NODE.1 record for each node
.	
(BLANK)	
INTE.ID	IDS
INTE.1	INTYP
INTE.2	LINE(1), LINE(2), ..., LINE(16)
.	Repeat INTE.1-INTE.2 record series for each interpolation line
.	
(BLANK)	
AUTO.ID	IDS
AUTO.1	NPTS, LMAT, NITSM
AUTO.2	LNGEN(1), LNGEN(2), ..., LNGEN(16)
.	Repeat AUTO.1-AUTO.2 record series for each region
.	
(BLANK)	
RESE.ID	IDS
RESE.1	IRTP, MXHLD
RESE.2	ISEQ(1), ISEQ(2), ..., ISEQ(16)
.	Repeat RESE.1-RESE.2 record series for each starting element sequence
.	
(BLANK)	
LAST.ID	IDS

Table 13-2. Run-stream representation for the Depth-Averaged
Flow Model: FLOMOD

Data set ID and record number	Data items entered on the record
SWMS.ID	IDS, WIDE, SCREEN
SWMS.1	TITLE
SWMS.2	IDRUN, IPRNT, IUNIT, IWIND, IFRIC, ISLIP, IHINT, INORM, IONOFF, ISAVE, NPVMIN
SWMS.3	IGRID, INITC, ISOUT, IRSRC, IBCIN, IWDIN, ILCOF, IUOCF
SWMS.4	NITS, NITD, NUPV, NITP, TSTRT, TMAX, DELT, THETA
SWMS.5	WSEL, OMEGA, ROWAT, BETA0, CBETA, CFLAG, DEPTOL, RELAX
SWMS.6	WVEL, WDIR, ROAIR, CSURF1, CSURF2, WVMIN
ELEM.ID	IDS
ELEM.1	L, LNODES(1,L), LNODES(2,L), ..., LNODES(9,L), LTYPES(L), LASEQS(L)
.	One ELEM.1 record for each element
(BLANK)	
NODE.ID	IDS, XFACT, YFACT, ZFACT, XZERO, YZERO, ZZERO
NODE.1	N, XYZZ(1,N), XYZZ(2,N), XYZZ(3,N), XYZZ(4,N)
.	One NODE.1 record for each node
(BLANK)	
PROP.ID	IDS
PROP.1	M, LPROPS(1,M), LPROPS(2,M), ..., LPROPS(7,M)
.	One PROP.1 record for each element type.
(BLANK)	
INIT.ID	IDS
INIT.1	N, VNEWS(1,N), VNEWS(2,N), VNEWS(3,N), VDOTS(1,N), VDOTS(2,N), VDOTS(3,N)
.	One INIT.1 record for each node
(BLANK)	
BOUN.ID	IDS
BOUN.1	N, NFIXES(N), BSPECS(1,N), BSPECS(2,N), BSPECS(3,N)
.	One BOUN.1 record for each node
(BLANK)	

Table 13-2. Run-stream representation for the Depth-Averaged Flow Module: FLOMOD (continued).

Data set ID and record number	Data items entered on the record
QSEC.ID	IDS
QSEC.1	IXSQ, XSQ(IXSQ)
QSEC.2	LXSQN(1), LXSQN(2), ..., LXSQN(16)
.	
.	Repeat QSEC.1 - QSEC.2 record series for up to 10 cross sections
.	
(BLANK)	
ZSEC.ID	IDS
ZSEC.1	IXSZ, XSZ1(IXSZ), XSZ2(IXSZ)
ZSEC.2	LXSZN(1), LXSZN(2), ..., LXSZN(16)
.	
.	Repeat ZSEC.1 - ZSEC.2 record series for up to 10 cross sections
.	
(BLANK)	
WIND.ID	IDS
WIND.1	N, SIGMA(N,1), SIGMA(N,2)
.	
.	One WIND.1 record for each node
.	
(BLANK)	
FLUX.ID	IDS
FLUX.1	LFLUXN(1), LFLUXN(2), ..., LFLUXN(16)
.	
.	Repeat FLUX.1 record series for each flow-check line
.	
(BLANK)	
WEIR.ID	IDS
WEIR.1	NOPW(J,1), NOPW(J,2), WCOF(J), WLEN(J), WCEL(J)
.	
.	One WEIR.1 record for each weir segment
.	
(BLANK)	
CULV.ID	IDS
CULV.1	NOPC(J,1), NOPC(J,2), CCOF(J), CARE(J), CHYR(J), CLEN(J), CMAN(J), CELV(J)
.	
.	One CULV.1 record for each culvert
.	
(BLANK)	

Table 13-2. Run-stream representation for the Depth-Averaged Flow Module: FLOMOD (continued).

Data set ID and record number	Data items entered on the record
TIME.ID	IDS
TIME.1	TSNEW
.	Followed by appropriate data sets. Repeat each time values are to be changed during a time-dependent run.
.	
.	
LAST.ID	IDS

Table 13-3. Run-stream representation for the Analysis of Output
Module: ANOMOD.

Data set ID and record number	Data items entered on the record
SWMS.ID	IDS, WIDE, TFIRST, TLAST
SWMS.1	IPRNT, IGRID, IFLOW, IGRIDB, IFLOWB, IUNITS
ELEM.ID	IDS
ELEM.1	L, LNODES(1,L), LNODES(2,L), ..., LNODES(9,L), LTYPES(L), LASEQS(L)
.	One ELEM.1 record for each element
(BLANK)	
NODE.ID	IDS, XFACT, YFACT, ZFACT, XZERO, YZERO, ZZERO
NODE.1	N, XYZZ(1,N), XYZZ(2,N), XYZZ(3,N), XYZZ(4,N)
.	One NODE.1 record for each node
(BLANK)	
FLOW.ID	IDS
FLOW.1	N, XVEL(1,N), XVEL(2,N), XVEL(3,N)
.	One FLOW.1 record for each node
(BLANK)	
GRID.ID	IDS, NEWPLT
GRID.1	PTITLE
GRID.2	IPLTN, IPLTE
GRID.3	HTT, HTN, HTE
GRID.4	XLOWER, YLOWER, XOFFSET, YOFFSET, XSCALE, YSCALE, AROT
VECT.ID	IDS, STIME, NEWPLT
VECT.1	PTITLE
VECT.2	IPLTV, IPLTS
VECT.3	VSCALE, VLMIN, AHTYP, AHLEN
VECT.4	HTT, HTN
VECT.5	XLOWER, YLOWER, XOFFSET, YOFFSET, XSCALE, YSCALE, AROT
CONT.ID	IDS, STIME, NEWPLT
CONT.1	PTITLE
CONT.2	IPLTC, ICVAL, NDECV
CONT.3	CVMIN, CVMAX, CVINC
CONT.4	HTT, HTN, HTC
CONT.5	XLOWER, YLOWER, XOFFSET, YOFFSET, XSCALE, YSCALE, AROT

Table 13-3. Run-stream representation for the Analysis of Output
Module: ANOMOD (continued).

Data set ID and record number	Data items entered on the record
FLUX.ID	IDS, STIME, NEWPLT
FLUX.1	PTITLE
FLUX.2	IPRTF, IPLTF, NDECF
FLUX.3	HTT, HTN, HTF
FLUX.4	XLOWER, YLOWER, XOFFSET, YOFFSET, XSCALE, YSCALE, AROT
FLUX.5	LINE(1), LINE(2), ..., LINE(16)
.	Repeat FLUX.5 record series for each flow-check line.
HIST.ID	IDS, TFIRST, TLAST
HIST.1	NTHP, KTHP
.	One HIST.1 record for each time-history plot
LAST.ID	IDS

Printed output can be stored in a data file, but eventually the data may need to be presented in hard copy format. A printer will be needed to obtain listings of printed output files.

Graphic output needs to be displayed on the screen of a graphics terminal or on some other plotting device such as an incremental digital plotter, dot-matrix printer, or a laser printer. Plotters need to recognize Hewlette-Packard Graphics Language (HPGL) commands. A variety of printers are supported which and identified during system setup. However, the format of a plot file can be interpreted easily (it is just a list of HPGL commands) and sent to whatever plotting device is available. A number of computer graphics programs can directly import an HPGL command file.

Restart/Recovery Procedure

Because FLOMOD requires substantial computer resources to solve the depth-averaged flow equations, a provision has been made to enable a run that has terminated abnormally to be restarted. At the end of every iteration FLOMOD writes the solution results to a restart/recovery data file. If a run terminates abnormally (for example, as the result of a power failure), the run can be restarted from the last completed iteration simply by setting the value of the variable IDRUN to 1 on the SWMS.1 data record and executing the program again.

Input/Output File Name Assignment

Input and output file names are read from files named DINMOD.FIL, FLOMOD.FIL, and ANOMOD.FIL by the programs DINMOD, FLOMOD, and ANOMOD, respectively. When a FESWMS-2DH module (DINMOD, FLOMOD, or ANOMOD) is run, the program looks for the appropriate input/output name file in the directory from which the program was executed. If the name file is not found, default input/output file names are assumed by the program.

Section 14

TROUBLE SHOOTING

Warning and error messages produced by the modeling system are listed and the required corrective action for each warning or error is described.

Warning messages point out possible sources of trouble although the program execution will not be terminated when a warning is given. Error messages indicate definite sources of trouble and program execution will be terminated at a convenient point after an error is encountered. Most warnings and errors are caused by a poorly constructed finite element network or data that has been entered incorrectly. Other error messages are produced when array dimensions are too small or when troubles arise during equation solution.

Warning Messages

The following warning messages may be printed. Possible corrective actions are described.

<u>Warning #</u>	<u>Warning Message, Remarks, and Corrective Action</u>
110	Node X violates middle-third rule in element X. <i>The midside node needs to be located within the middle-third of an element side or the possibility of a vanishing determinant will exist. Reposition the midside node.</i>
120	Element X is unattached. <i>The element is not connected to any other element. Problems may result in the equation solution. Delete the element or attach it to the rest of the network.</i>

Warning # Warning Message, Remarks, and Corrective Action

130 Determinant ≤ 0 in element X at integration point X. A negative or zero determinant may lead to unacceptable solutions. The problem is caused by a poorly formed element. Reconstruct the element.

140 Midside node between corner nodes X and X was not found.

While interpolating node point coordinates along a straight line, an element side containing the two corner node points was not found. Check the list of nodes defining the interpolation line.

150 $MAXF*MAXF > MAVAIL$. Available memory may be exceeded.

The solution storage array dimensioned as A(MAVAIL) may not be large enough. Either reduce the maximum frontwidth of the system or increase MAVAIL and redimension the working array A(MAVAIL).

160 Flow is SUPERcritical at node X ($Fr = X$) and SUBcritical flow boundary conditions have been specified.

The type of boundary conditions need to correspond to the state of the flow. For SUPERcritical flow, both depth and velocity or unit flow need to be specified at an inflow boundary, nothing is specified at an outflow boundary. For SUBcritical flow, usually velocity or unit flow will be specified at an inflow boundary, and depth (water-surface elevation) will be specified at an outflow boundary. Misspecification of boundary conditions may cause the solution to have convergence problems.

170 Flow is SUBcritical at node X ($Fr = X$) and SUPERcritical flow boundary conditions have been specified.

See the corrective actions given for Warning #160.

Error Messages

The following error messages may be printed. Possible corrective actions are described.

<u>Error #</u>	<u>Error Message, Remarks, and Corrective Action</u>
110	<p>Coordinates are not specified for node X in line X.</p> <p><i>Coordinates need to be specified for the node if it is to be included in the node interpolation line. Specify the coordinates of the node point.</i></p>
111	<p>X coordinate for node X in line X is invalid.</p> <p><i>The x coordinate of the node point needs to be between the x coordinates of the two endpoints of the interpolation line. Check the coordinates of the node and the endpoints.</i></p>
112	<p>Y coordinate for node X in line X is invalid.</p> <p><i>The y coordinate of the node point needs to be between the y coordinate of the two endpoints of the interpolation line. Check the coordinates of the node and the endpoints.</i></p>
140	<p>Side X, X, X in element X disagrees with side X, X, X in element X.</p> <p><i>A geometric inconsistency has been detected. Check the element connectivity lists to make sure they are correct. Visually inspect a plot of the finite element network in the areas around the two elements.</i></p>
141	<p>Center node X appears in elements X and X.</p> <p><i>The center node in a nine-node quadrangular element should appear in the connectivity list of only that element. Correct the connectivity lists of one or both of the elements.</i></p>
142	<p>Node mixup. Node X appears in elements X and X.</p> <p><i>The node is specified as the center node of a 9-node quadrangular element and is in the connectivity list of another element. Correct one or both of the element connectivity lists.</i></p>

<u>Error #</u>	<u>Error Message, Remarks, and Corrective Action</u>
145	Node X in element X is invalid. <i>Node numbers need to be greater than zero and less than MAXP. Correct the node number.</i>
150	Maximum number of element sides exceeded. <i>Increase the maximum number of elements sides, MAXS, and redimension associated arrays in DINMOD.</i>
151	Error in element side list. <i>Carefully inspect a plot of the finite element network for geometric inconsistencies.</i>
160	Error in adjacent element list for node X in element X. <i>Make sure that a geometric data check has been performed. If a geometric data check has been performed, then carefully inspect a plot of the network in the vicinity of the node.</i>
163	Sequence X for element X > MAXE. <i>The computed element sequence is greater than the element array dimension MAXE. Make sure that a geometric data check has been performed. If it has, then carefully inspect a plot of the network for geometric inconsistencies.</i>
170	Coordinates have not been specified for corner node X. <i>Coordinates need to be specified or interpolated for all corner node points. Specify the coordinates.</i>
180	Generated element number exceeds element array dimension. <i>The generated element number is greater than MAXE. Either increase the element array dimension or reduce the number of elements in the network.</i>
181	Generated node number exceeds nodal array dimension. <i>The generated node number is greater than MAXP. Either increase the nodal array dimension or reduce the number of nodes in the network.</i>

<u>Error #</u>	<u>Error Message, Remarks, and Corrective Action</u>
182	<p>Polygon number X has no internal angles < 180 degrees.</p> <p><i>A simply connected region defined by a series of node points will have at least three interior angles less than 180 degrees. Check the list of node points that form the polygon. Make sure the nodes have been entered in a counterclockwise direction around the boundary of the region. Make sure that coordinates have been specified for all the node points.</i></p>
185	<p>NP to exceed MAXP upon subdivision of element X.</p> <p><i>During element refinement, the maximum allowable node point number will be exceeded. Either reduce the number of nodes in the network or increase the nodal array dimension (MAXP) and redimension all associated arrays.</i></p>
186	<p>NE to exceed MAXE upon subdivision of element X.</p> <p><i>During element refinement, the maximum allowable element number will be exceeded. Either reduce the number of elements in the network or increase the element array dimension (MAXE) and redimension all associated arrays.</i></p>
190	<p>Nodes X and X have identical coordinates.</p> <p><i>Two nodes cannot have identical coordinates. Change the coordinates of one or both of the node points.</i></p>
210	<p>Indeterminate boundary angle at node X.</p> <p><i>A problem was encountered when computing an approximate boundary angle at the node. Visually inspect a plot of the finite element network around the node.</i></p>
213	<p>Total flow cross section node X is not a boundary node.</p> <p><i>Nodes defining a cross section at which total flow is specified need to lie on an open boundary of the finite element network. Either relocate the cross section or specify an open boundary node.</i></p>

<u>Error #</u>	<u>Error Message, Remarks, and Corrective Action</u>
220	<p>Element number X in the XXXX data set is not valid.</p> <p>The element number exceeds the element array dimension, MAXE. Either make sure all element numbers are less than or equal to MAXE, or increase MAXE and the dimension of all element arrays.</p>
230	<p>Node number X in the XXXX data set is out of range.</p> <p>The node number exceeds the node point array dimension, MAXP. Either make sure all node point numbers are less than or equal to MAXP, or increase MAXP and the dimension of all node point arrays.</p>
235	<p>Property set number X in the XXXX data set is out of range.</p> <p>The property set number exceeds the property set array dimension, MAXM. MAXM is set equal to 99 and cannot be changed. Make sure that property type codes do not exceed 99 when constructing the finite element network.</p>
240	<p>Property set number is zero for element X in sequence X.</p> <p>The property set number for the element in the specified element resequencing starting list is zero. If the property type code is zero, the element will not be used in computations. Either assign the appropriate element type code or remove the element from the starting sequence.</p>
245	<p>Adjacent element list for node X in element X is in error.</p> <p>The adjacent element list computed during element resequencing is not correct. Visually inspect the finite element network in the vicinity of the element for geometric inconsistencies that have not been detected previously.</p>
250	<p>Frontwidth does not vanish upon assembly of the last element.</p> <p>The frontwidth (defined in terms of node points) does not vanish upon assembly of the last element in the computed element assembly sequence. Visually inspect the finite element network for geometric inconsistencies that have not been detected previously detected.</p>

<u>Error #</u>	<u>Error Message, Remarks, and Corrective Action</u>
260	<p>Element sequence X at element X is not valid.</p> <p>The element assembly sequence exceeds the element array dimension, MAXE. Visually inspect the finite element network for geometric inconsistencies that have not been previously detected.</p>
270	<p>Number of elements adjacent to node X exceeds 10.</p> <p>The maximum allowable number of elements adjacent (connected) to a node point is 10 if element resequencing is performed. Modify the finite element network so that no node point is connected to more than 10 elements.</p>
280	<p>Number of elements adjacent to element X exceeds 20.</p> <p>The maximum allowable number of elements adjacent (connected) to an element is 20 if element resequencing is performed. Modify the finite element network so that no element is connected to more than 20 elements.</p>
300	<p>Chezy C for property set X is ≤ 0.</p> <p>If Chézy discharge coefficients are to be used to define the bed friction coefficient, values in the element property sets need to be greater than zero. Check the property set.</p>
310	<p>Manning n for property set X is ≤ 0.</p> <p>If Manning roughness coefficients are to be used to define the bed friction coefficient, values in the element property sets need to be greater than zero. Check the property set.</p>
320	<p>Max allowable number of flux-line points exceeded.</p> <p>The number of flux-line points has exceeded the flux-line array dimension, MAXLP. Either reduce the number of flux-line points or increase MAXLP and the dimension of the flux line array.</p>
330	<p>Max allowable number of weir segments exceeded.</p> <p>The number of weir segments has exceeded the weir array dimension, MAXW. Either reduce the number of weir segments in the finite element network, or increase MAXW and the dimension of all weir arrays.</p>

<u>Error #</u>	<u>Error Message, Remarks, and Corrective Action</u>
340	<p>Max allowable number of culverts exceeded.</p> <p>The number of culverts has exceeded the culvert array dimension, MAXC. Either reduce the number of culverts in the finite element network or increase MAXC and the dimension of all culvert arrays.</p>
345	<p>Max allowable section number exceeded.</p> <p>The identification number of the total flow cross-section can be between 1 and 10. Change the total flow cross-section identification number.</p>
346	<p>Max allowable number of section points exceeded.</p> <p>Up to 79 node points may be used to define a total flow cross section. If you need more than 79 nodes, divide the section into two or more cross sections and divide the total flow between them.</p>
347	<p>Boundary code for section X is out of range.</p> <p>The boundary condition code for the cross section is not within the allowable range. Check the code and respecify.</p>
350	<p>TSNEW <= current simulation time.</p> <p>The simulation time at which the next set of time-dependent data becomes effective is less than or equal to the current simulation time. Check the specified time.</p>
360	<p>No iterations have been specified.</p> <p>At least one iteration (either a steady-state or an unsteady-state iteration) needs to be specified. Check the program control data.</p>
370	<p>Number of steady-state iterations exceeds max.</p> <p>The maximum total number of steady-state iterations exceeds 99. Reduce the number of specified iterations.</p>
380	<p>Number of iterations/time-step exceeds max.</p> <p>The maximum number of iterations per time-step exceeds 99. Reduce the number of specified iterations.</p>

<u>Error #</u>	<u>Error Message, Remarks, and Corrective Action</u>
390	Time integration factor out of allowable range. The time integration factor needs to be greater than or equal to 0.5 and less than or equal to 1.0. Check the program control data.
391	Relaxation factor out of allowable range. The relaxation factor needs to be greater than zero and less than or equal to two. Check the value and respecify.
395	Time-dependent data sets are out of order. Time dependent data sets need to appear in chronological order at the end of the input data stream after all other data sets.
400	FLOMOD input data file XXXX not found. The named FLOMOD input data file was to be read but was not found. Either the input data file has a different name or it has not been created. Create the data file or supply the correct file name.
401	Network data file XXXX cannot be found. The named network data file was to be read but was not found. Either enter all network data on data records or supply the correct network data file name.
402	Restart/recovery file XXXX cannot be found. The named restart/recovery file was to be read but was not found. Either supply the correct file name or do not ask for a restart/recovery run.
403	Initial condition file XXXX cannot be found. The named initial condition data file was to be read but was not found. Either enter all initial condition data on data records or supply the correct initial condition file name.
404	Boundary condition file XXXX cannot be found. The named boundary condition data file was to be read but was not found. Either read all boundary condition data from data records or supply the correct boundary condition file name.

Error #

Error Message, Remarks, and Corrective Action

- 405 Wind data file XXXX cannot be found.
The named wind data file was to be read but was not found. Either read all wind data from data records or supply the correct wind data file name.
- 406 DINMOD input data file XXXX cannot be found.
The named DINMOD input data file was to be read but was not found. Either the input data file has a different name or it has not yet been created. Create the data file or supply the correct file name.
- 407 ANOMOD input data file XXXX cannot be found.
The named ANOMOD input data file was to be read but was not found. Either the input data file has a different name or it has not yet been created. Create the data file or supply the correct file name.
- 408 Flow data file XXXX cannot be found.
The named flow data file was to be read but was not found. Either read all flow data from data records or supply the correct file name.
- 420 No node data have been input.
Node data can be read from data records or a network data file. Apparently no node data at all have been entered. Check the NODE data set and the network data file specification in the SWMS data set.
- 421 No element data have been input.
Element data can be read from data records or a network data file. Apparently no element data at all have been entered. Check the ELEM data set and the network data file specification in the SWMS data set.
- 422 No element property set data have been input.
Element property data are read from data records. Apparently no element data at all have been entered. Check the PROP data set.

<u>Error #</u>	<u>Error Message, Remarks, and Corrective Action</u>
440	<p>Specified depth < 0 at node X.</p> <p>The specified water-surface elevation results in a negative depth at the node point. Check the specified water-surface elevation and the ground elevation at the node point.</p>
445	<p>Ceiling elev less than ground elev at node X in element X.</p> <p>The ceiling elevation needs to be greater than the ground elevation. Correct either the ceiling or ground elevation at the node point.</p>
450	<p>Assembly sequence X is assigned to elements X and X.</p> <p>An error exists in the element assembly sequence. Recompute the element assembly sequence using the resequencing option in DINMOD.</p>
460	<p>Element side X, X, X in cross-section X was not found.</p> <p>The element side consisting of a corner node, midside node, and a corner node in the cross-section was not found when searching the element connectivity lists. Check the list of nodes that define the total flow cross section.</p>
461	<p>Depth < 0 at node X: Depth = X: Ground elev = X.</p> <p>A negative depth has been encountered in a cross section. Check the section to make sure that it has been specified correctly.</p>
470	<p>Maximum number of equations has been exceeded.</p> <p>The number of equations to be formed exceeds the solution array dimension, MAXQ. Either reduce the number of nodes in the network or increase MAXQ and the dimension of the solution array.</p>
490	<p>Maximum frontwidth exceeded.</p> <p>The frontwidth is larger than the maximum allowed and array dimensions will be exceeded. Either reduce the maximum frontwidth of the equation system by computing a new element assembly sequence, or increase the size of the frontal solution working array (array A(MAVAIL)) and the corresponding maximum allowable frontwidth.</p>

<u>Error #</u>	<u>Error Message, Remarks, and Corrective Action</u>
510	<p>Available memory too small to allow assembly of element X.</p> <p><i>Either reduce the maximum frontwidth of the equation system by computing a new element assembly sequence, or increase the size of the frontal solution working array (array A(MAVAIL)) and the corresponding maximum allowable frontwidth (MAXF). Possibly NPVMIN is too large, try setting this value to 1 (that is, the default value).</i></p>
511	<p>Error reading upper coefficient matrix file (IBUFF = X).</p> <p><i>Somehow the upper coefficient matrix file has been corrupted. IBUFF is the number of the buffer block that is being read.</i></p>
515	<p>Unexpected end-of-buffer encountered (IBUFF = X).</p> <p><i>The end of a buffer block (block number IBUFF) filled by part of the LOWER coefficient-matrix was encountered unexpectedly. An error exists in the lower coefficient-matrix file (ILCOF.DAT). Make sure that a geometric data check has been performed.</i></p>
516	<p>Unexpected end-of-buffer encountered (IBUFF = X).</p> <p><i>The end of a buffer block (block number IBUFF) filled by part of the UPPER coefficient-matrix was encountered unexpectedly. An error exists in the upper coefficient-matrix file (IUCOF.DAT). Make sure that a geometric data check has been performed.</i></p>
518	<p>All fully-assembled equations have a zero diagonal coefficient. Contents of IHED are as follows:</p> <p><i>The number of completed equations equals or exceeds NPVMIN but all of these equations have zero pivotal coefficients. The array IHED contains the numbers of the partially and fully assembled equations. Fully assembled equations have negative numbers. Try increasing NPVMIN to allow more fully assembled equations to remain in the working matrix.</i></p>

Section 15

REFERENCES

- American National Standards Institute, American National Standard Programming Language FORTRAN, ANSI X3.9-1978, 1978.
- Arcement, G. J., and Schneider, V. R., Guide for Selecting Manning's Roughness Coefficients for Natural Channels and Flood Plains, Federal Highway Administration Report No. FHWA-TS-84-204, 1984, 62 p.
- Barnes, H. H., Jr., Roughness Characteristics of Natural Channels, U.S. Geological Survey Water-Supply Paper 1849, 1967, 213 p.
- Bodhaine, G. L., Measurement of Peak Discharge at Culverts by Indirect Measurements, U.S. Geological Survey Techniques of Water Resources Investigations, Book 3, Chapter A3, 1968, 60 p.
- Bradley, J. N., Hydraulics of Bridge Waterways (2nd ed.), Federal Highway Administration Hydraulic Design Series, No. 1, 1978, 111 p.
- Buell, W. R., and Bush, B. A., "Mesh Generation -- A Survey," Transactions of the American Society of Mechanical Engineers, Journal of Engineering for Industry, ser. B., v. 95, no. 1, 1973, p. 332-338.
- Chow, V. T., Open-Channel Hydraulics, New York, McGraw-Hill, 1959, 680 p.
- Engelman, M. S., Strang, G., and Bathe, K.-J., "The Application of Quasi-Newton Methods in Fluid Mechanics," International Journal for Numerical Methods in Engineering, v. 17, no. 5, 1981, p. 707-718.
- Fischer, H. B., List, E. J., Koh, R. C. Y., Imberger, J., and Brooks, N. H., Mixing in Inland and Coastal Waters, New York, Academic Press, 1979, 483 p.
- Garratt, J. R., "Review of Drag Coefficients over Oceans and Continents," Monthly Weather Review, v. 105, no. 7, 1977, p. 915-929.
- Hicks, B. B., "Some Evaluations of Drag and Bulk Transfer Coefficients over Water Bodies of Different Sizes," Boundary-Layer Meteorology, v. 3, 1972, p. 201-213.

- Hicks, B. B., Drinkrow, R. L., and Grauze, G., "Drag and Bulk Transfer Coefficients Associated with a Shallow Water Surface," *Boundary-Layer Meteorology*, v. 6, no. 1/2, 1974, p. 287-297.
- Hood, P., "Frontal Solution Program for Unsymmetric Matrices," *International Journal for Numerical Methods in Engineering*, v. 10, no. 2, 1976, p. 379-399.
- , "Note on Frontal Solution Program for Unsymmetric Matrices," *International Journal for Numerical Methods in Engineering*, v. 11, no. 6, 1977, p. 1055.
- Jansen, P. P., van Bendegom, L., van den Berg, J., deVries, M., and Zanen, A., *Principles of River Engineering -- The Non-Tidal Alluvial River*, London, Pittman, 1979, 509 p.
- King, I. P., and Norton, W. R., "Recent Applications of RMA's Finite Element Models for Two-Dimensional Hydrodynamics and Water Quality," in Brebbia, C. A., Gray, W. G., and Pinder, G. F., eds., *Finite Elements in Water Resources*, International Conference, 2nd, London, 1978, Proceedings, London, Pentech Press, 1978, p. 2.81-2.99.
- Lee, J. K. and Froehlich, D. C., Review of Literature on the Finite-Element Solution of the Equations of Two-Dimensional Surface-Water Flow in the Horizontal Plane, U.S. Geological Survey Circular 1009, 1986, 61 p.
- Rodi, Wolfgang, "Hydraulics Computations with the k-e Turbulence Model," in Smith, P. E., ed., *Applying Research to Hydraulic Practice*, Conference of the Hydraulics Division of the American Society of Civil Engineers, Jackson, Miss., 1982, Proceedings, New York, American Society of Civil Engineers, 1982, p. 44-54.
- Sokolnikoff, I. S., and Redheffer, R. M., *Mathematics of Physics and Modern Engineering* (2nd ed.), New York, McGraw-Hill, 1966, 752 p.
- Strang, G., and Fix, G. J., *An Analysis of the Finite Element Method*, Englewood Cliffs, N.J., Prentice-Hall, 1973, 306 p.
- Wang, J. D., and Connor, J. J., *Mathematical Modeling of Near Coastal Circulation*, Cambridge, Mass., Massachusetts Institute of Technology, Department of Civil Engineering, Ralph M. Parsons Laboratory for Water Resources and Hydrodynamics, Report No. 200, 1975, 272 p.
- Zienkiewicz, O. C., *The Finite Element Method* (3rd ed.), London, McGraw-Hill, 1977, 787 p.

Appendix A

INPUT DATA WORKSHEETS

This appendix contains worksheets that will simplify recording of data that will be entered into input data files. Worksheets are provided for many of the FESWMS-2DH data sets. One copy of each worksheet is provided. These worksheets can be reproduced as often as needed.

Appendix B

MICROCOMPUTER IMPLEMENTATION

This appendix describes implementation of FESWMS-2DH on a microcomputer running under the Microsoft Disk Operating System (MS-DOS).

Hardware Requirements

At least 570 kilobytes of free RAM (random access memory) are needed to run FESWMS-2DH. An 80x87 math coprocessor is also required.

Sufficient disk space is needed, either on a high-density floppy disk drive or a fixed disk drive, to store all input data files, output data files, and all temporary data files created during program execution. All but the smallest of finite element networks will require more storage space than is available on one or two high-density floppy diskettes. At least 1 megabyte of disk space is needed to hold all the FESWMS-2DH files contained on the floppy diskettes that accompany this appendix. Additional storage space will be needed for input and output data files as well as temporary files that are created during program execution. Therefore, it is suggested that FESWMS-2DH be installed on a fixed disk drive that has sufficient storage space.

Finite Element Network Size Limits

The standard microcomputer version of FESWMS-2DH can accommodate a finite element network that contains up to 400 elements, 1600 node points, a maximum frontwidth of 150 degrees-of-freedom (about 60nodes), 50 culverts, and 50 weir segments. The extended version that operates on microcomputers that have an 80386 or an 80486 processor can accommodate networks

containing up to 4000 elements, 16000 node points, and a maximum frontwidth of 350 degrees-of-freedom if at least 4 megabytes of RAM are installed. Larger networks can be modeled if more RAM is available.

Modeling System File Descriptions

FESWMS-2DH files on the high-density floppy diskette that accompanies this appendix are described below:

INSTALL.EXE - Installation program, just type "INSTALL".
FESWMS.BAT - Batch file that runs FESWMS-2DH shell.

PROGRAMS Directory

FESWMS.DOC - Some additional documentation for the microcomputer version of FESWMS-2DH.
FESWMS.EXE - The system control programs that provides a menu-driven operational environment for the modeling system.
FESWMS.OVR - Overlays for the system control program.
DINMOD.EXE - Data Input Module (DINMOD) executable code.
DINDFE.EXE - DINMOD Data File Editor program.
DINDFE.MES - DINDFE message file.
FLOMOD.EXE - Depth-Averaged Flow Module (FLOMOD) executable code.
FLODFE.EXE - FLOMOD Data File Editor program.
FLODFE.MES - FLODFE message file.
ANOMOD.EXE - Analysis of Output Module (ANOMOD) executable code.
ANODFE.EXE - ANOMOD Data File Editor program.
ANODFE.MES - ANODFE message file.

Modeling System Installation

Before installing FESWMS-2DH it is suggested that the floppy diskettes be copied for backup purposes and stored in a safe place.

The MS-DOS default value for the number of files that can be simultaneously open is eight. Some of the FESWMS-2DH programs may need to have open more than eight files at the same time. It is suggested that the statement

FILES = 20

be included in a CONFIG.SYS file on your system (see the section in your MS-DOS reference guide on configuring your system). This statement increases the number of files that can be open simultaneously to 20.

Buffer space is provided in a microcomputer's memory for the purpose of data input and output. Each buffer contains 528 bytes of RAM. The speed of a read/write operation will increase as more buffers are allocated to disk file input/output. The MS-DOS default is two disk buffers. Because FESWMS-2DH programs do a lot of disk input/output, it is suggested that the statement

```
BUFFERS = 10
```

also be included in the CONFIG.SYS file. This statement increases the number of disk buffers from two to 10.

Remember that if you modify an existing CONFIG.SYS file or create a new CONFIG.SYS file you will have to restart your microcomputer for the commands to take effect.

Installation of FESWMS-2DH files can be done automatically by placing the first FESWMS-2DH diskette in your floppy diskette drive and typing "INSTALL" (do not type the quote marks). You will then be prompted for the letter designation of the drive where the system is to be installed and a main directory named FESWMS will be created on this drive. It is a good idea to keep all of your FESWMS-2DH files in subdirectories under FESWMS, using one subdirectory for each modeling project. All of the FESWMS-2DH programs and ancillary files are kept in a subdirectory named PROGRAMS. A message will be written to the screen as each file is copied from the floppy to your destination directory. If a file of the same name already exists in your destination directory you will be asked if you want to overwrite that file.

System Control Program

The FESWMS-2DH System Control Program (SCP) provides a integrated menu-driven operational environment for the modeling system. The SCP enables convenient access to all of the FESWMS-2DH modules, allows files and directories to be managed easily, and contains a number of utilities that greatly simplify a modeling project. Utilities include graphic display of output on a screen, printer, or plotter; a calculator for simple arithmetic; and a calendar so you always know what day it is. To run the SCP after the modeling system has been installed on your computer, just type "FESWMS" (do not type the quotes ") from any directory on your disk and the screen shown below will appear on your monitor.. If you were not already in the directory where you would like to do your work (that is, the working directory), you can move about your disk quite easily while in the operating environment provided by the SCP.

Menu System

Operation of FESWMS-2DH is controlled through a system of multi-level menus and windows. The main screen contains a horizontal scrolling menu that provides access to all the major features of the modeling system. Using just the cursor arrow keys <Left>, <Right>, <Up>, and <Down>, along with the <Enter> key, all of the menu items and submenus can be accessed. Commands needed to navigate the system of menus are described in table B-1.

Table B-1. Menu System Key Commands

Function	Key Commands	
	Single	Multiple
Move highlight bar left one item on the horizontal menu. If the bar is on the left-most item it wraps to the right-most item.	<Left>	<Ctrl-S>
Move highlight bar right one item on the horizontal menu. If the bar is on the right-most item it wraps to the left-most item.	<Right>	<Ctrl-D>
Move highlight bar up one row on a vertical menu. If the bar is on the top-most item, it wraps to the bottom-most item.	<Up>	<Ctrl-E> or <Ctrl-W>
Move highlight bar down one row on a vertical menu. If the bar is on the bottom-most item, it wraps to the top-most item.	<Down>	<Ctrl-X> or <Ctrl-Z>
Move highlight bar to the left-most item of a horizontal menu or the top-most item of a vertical menu.	<Home>	<Ctrl-Q><S>
Move highlight bar to right-most item of a horizontal menu or the bottom-most item of a vertical menu.	<End>	<Ctrl-Q><D>
Move to the first item on the menu that matches the character pressed. The matching character of an item is highlighted and is usually the first character in the item name.	any alpha-numeric character	-- ^a
Select the item that is highlighted.	<Enter>	--
Close the current submenu or window and return to the previous level.	<Esc>	--
Display a help screen showing available hot keys that allow immediate access to menu functions from any point within the menu system.	<F1>	--

^aA command does not exist.

File Management

A large number of files may be created during a modeling project. A number of functions are available within the SCP that provide an easy means of managing your files. These functions consist of the following:

- Listing files in a directory.
- Browsing through a file (that is, viewing the contents of a file without editing).
- Editing a file.
- Renaming a file.
- Copying a file.
- Deleting a file.
- Moving a file from one directory to another
- Printing a file.
- Canceling a file print instruction.

Full-Screen Editor Commands

The full-screen text editor that is built into the system control programs allows entry and modification of information in a file that can be stored for later use and/or further modification. Text refers to a sequence of characters and/or lines that are being edited. Depending on the context, the term *file* may refer to a physical file stored on a disk or to the text of a file that is being edited. Text is entered in much the same way as text is typed on a typewriter. The cursor always indicates where new text will be placed in a text file, and you can move the cursor around a text file in several ways. Text can be changed quickly and easily using delete and insert commands. A particular string of text in a file can be located quickly using a simple search command and, optionally, replaced by another text string. The full-screen editor offers an extensive set of text editing commands that are described by category in the following sections.

Basic Cursor Movement Commands

Basic movement commands allow the cursor to be moved through a text file without altering any of the text. The cursor can be moved in either of two ways: (1) By the press of a single key, or (2) by first pressing a control character, and then pressing one or more keys that define a command. For example, the cursor can be moved one character to the right by pressing either the right arrow key, or by first pressing the control character key (the key labelled "Ctrl" on your keyboard), and then pressing the letter D (that is, by pressing the key combination Ctrl-D). Some commands operate on words (a word is defined as a sequence of characters delimited by a space, tab, carriage return, line feed, or one of the following characters: < > , . / ? ; : [] { } - = \ + | () * % @ & ^ \$ # ! ~). Key commands that cause basic cursor movement functions are defined in table B-2. A description of each function is followed by the single and multiple key commands assigned to that function.

Insertion and Deletion Commands

Commands that allow characters, words, and lines of text to be inserted and deleted are described in table B-3.

Terminate and Save File Commands

Commands that cause the full-screen editing program to terminate execution, and that cause the current text file to be saved are described in the table B-4.

Find-and-Replace Commands

Commands that cause a search for patterns of text and, optionally, the text to be replaced by a new pattern are described in table B-5. After the search string is entered, search options can be specified. The options used last (if any) are displayed first. New options can be entered (canceling the old ones), the current options may be edited, or the displayed options can be selected by pressing *Enter*.

Table B-2. Basic Cursor Movement Key Commands

Function	Key Commands	
	Single	Multiple
Cursor left one character.	<Left>	<Ctrl-S>
Cursor right one character.	<Right>	<Ctrl-D>
Cursor left one word.	-- ^a	<Ctrl-Left> or <Ctrl-A>
Cursor right one word.	--	<Ctrl-Right> or <Ctrl-F>
Cursor to beginning of line.	<Home>	<Ctrl-Q><S>
Cursor to end of line.	<End>	<Ctrl-Q><D>
Cursor up one line.	<Up>	<Ctrl-E>
Cursor down one line.	<Dn>	<Ctrl-X>
Scroll display up one line.	--	<Ctrl-W>
Scroll display down one line.	--	<Ctrl-Z>
Scroll display up one page.	<PgUp>	<Ctrl-R>
Scroll display down one page.	<PgDn>	<Ctrl-C>
Move cursor to the top of the edit window.	--	<Ctrl-Home> or <Ctrl-Q><E>
Move cursor to the bottom of the edit window.	--	<Ctrl-End> or <Ctrl-Q><X>
Move cursor to the beginning of the file.	--	<Ctrl-PgUp> or <Ctrl-Q><R>
Move cursor to the end of the file.	--	<Ctrl-PgDn> or <Ctrl-Q><C>
Jump to a specified line.	--	<Ctrl-J><L>

^aA command does not exist.

Table B-3. Insertion and Deletion Key Commands

Function	Key Commands	
	Single	Multiple
Delete character at the cursor.		<Ctrl-G>
Delete character to the left of the cursor.	<Bksp>	<Ctrl-H> or <Ctrl-Bksp>
Delete current line.	-- ^a	<Ctrl-Y>
Delete from cursor to the end of the line.	--	<Ctrl-Q><Y>
Delete word to right of cursor.	--	<Ctrl-T>
Start a new line.	<Enter>	<Ctrl-M>
Insert a new line at the cursor.	--	<Ctrl-N>
Move cursor to the next tab stop	<Tab>	<Ctrl-I>
Insert control character (For example, to insert ^G enter enter <Ctrl-P><Ctrl-G>.)	--	<Ctrl-P>

^aA command does not exist.

Table B-4. Terminate and Save File Commands

Function	Key Commands	
	Single	Multiple
Save the current file then continue editing.	<F2>	<Ctrl-K><S>
Load a new file; if the current file has been modified the user will be prompted to save it first.	<F3>	-- ^a
Quit editing; ignore any changes made to the text.	--	<Ctrl-K><Q>
Save the current file then load a new file.	--	<Ctrl-K><D>
Save the current file under a different name	--	<Ctrl-K><N>
Save the file then quit	<F10>	<Ctrl-K><X>

^aA command does not exist.

Table B-5. Find-and-Replace Commands

Function	Key Commands	
	Single	Multiple
Find a specified text string. Find options are as follows: B Search backwards from the current position G Search globally. The entire file will be scanned. L Search locally within the marked block. U Ignore case of alphabetic characters.	-- ^a	<Ctrl-Q><F>
Replace a specified text string. In addition to the options in the Find command the following option is available: N Replacements are to be made without prompting for confirmation.	--	<Ctrl-Q><A>
Repeat last text-string search.	--	<Ctrl-L>

^aA command does not exist.

Block Commands

A block is an arbitrary contiguous unit of text in a file. A block can be as small as a single character or as large as the entire file. A block can be marked by placing a begin-block marker at the first character of the desired block, and an end-block marker just beyond the last character in the desired block. Once marked, the block can be copied, moved, or deleted using the key commands described in table B-6. Although marked blocks are normally highlighted so the block is visible, the block may be *hidden* (or made invisible) by a simple toggle command.

Table B-6. Block Commands

Function	Key Commands	
	Single	Multiple
Mark the cursor position as the beginning of a block.	<F7>	<Ctrl-K>
Mark the cursor position as the end of a block.	<F8>	<Ctrl-K><K>
Mark the current word as a block	-- ^a	<Ctrl-K><T>
Jump to beginning of the currently marked block.	--	<Ctrl-Q>
Jump to end of the currently marked block.	--	<Ctrl-Q><K>
Toggle the display of marked blocks on/off.	--	<Ctrl-K><H>
Copy the currently marked and displayed block to the position of the cursor.	--	<Ctrl-K><C>
Move the currently marked and displayed block to the position of the cursor.	--	<Ctrl-K><V>
Delete the currently marked and displayed block.	--	<Ctrl-K><Y>
Read a file into the text buffer at the current position of the cursor.	--	<Ctrl--K><R>
Write the currently marked and displayed block to a file.	--	<Ctrl-K><W>
Write currently displayed block to the printer.	--	<Ctrl-K><P>

^aA command does not exist.

Miscellaneous Commands

A few miscellaneous commands that do not fit into one of the the previous command categories are defined table B-7.

Table B-7. Miscellaneous Commands

Function	Key Commands	
	Single	Multiple
Restore the original contents of a line.	-- ^a	<Ctrl-Q><L>
Center the current line between column 1 and the right margin.	--	<Ctrl-O><C>

^aA command does not exist.

File Browsing

The contents of a file can be examined without changing it using the Browse function. A number of commands are available for moving about a file. You can go immediately to the top or the bottom of a file, jump to a specified line, or search for a specified text string. In addition, parts of a file can be marked as a block and sent to the printer.

The file browser screen is shown below. The browse window is 79 columns wide and 24 rows high. Files that contain more than 79 columns per row can be viewed by *scrolling* the window right and left to view different sections of the file. The name of the file being viewed, number of the line at the top of the window, and the number of the column at the left of the window are indicated on the top line or *status line*. The vertical slide bar in the right-most column of the screen shows the relative position of the top-most line within the file. Commands needed to move through the file, search for specified text strings, mark blocks of text, send marked blocks to a printer, and toggle browser options are described in table B-7.

Table B-8. File Browsing Commands

Function	Key Commands	
	Single	Multiple
Scroll window left one column	<Left>	<Ctrl-S>
Scroll window right one column	<Right>	<Ctrl-D>
Scroll window left 10 columns	-- ^a	<Ctrl-Left> or <Ctrl-A>
Scroll window right 10 columns	--	<Ctrl-Right> or <Ctrl-F>
Scroll window to column 1	<Home>	<Ctrl-Q><S>
Scroll to end of the longest line	<End>	<Ctrl-Q><D>
Scroll window up one page	<PgUp>	<Ctrl-R>
Scroll window down one page	<PgDn>	<Ctrl-C>
Scroll to beginning of the file	--	<Ctrl-PgUp> or <Ctrl-Q><R>
Scroll to end of the file	--	<Ctrl-PgDn> or <Ctrl-Q><C>
Scroll to a specified line	--	<Ctrl-J><L>
Scroll window up one line	<Up>	<Ctrl-E> or <Ctrl-W>
Scroll window down one line	<Down>	<Ctrl-X> or <Ctrl-Z>
Load a new file	<F3>	--
Mark the top-most line in the window as the start of a block	<F7>	<Ctrl-K>
Mark the top-most line in the window as the end of a block	<F8>	<Ctrl-K><K>
Mark the bottom-most line in the window as the end of a block	--	<Ctrl-B><K>
Jump to beginning of marked block	--	<Ctrl-Q>
Jump to end of marked block	--	<Ctrl-Q><K>
Toggle the display of marked blocks on/off	--	<Ctrl-K><H>
Write currently displayed block to a file	--	<Ctrl-K><W>
Print the currently displayed	--	<Ctrl-K><P>
Find a specified text string	--	<Ctrl-Q><F>
Repeat last text-string search	--	<Ctrl-L>
Set a text marker at the start of the line at the top of the window	--	<Ctrl-K><#>; # = 1..10
Jump to a specified text marker	--	<Ctrl-Q><#>; # = 1..10
Toggle between ASCII/hex text mode	--	<Ctrl-H> or
Toggle tab expansion on/off	--	<Ctrl-Q><T>
Quit browsing	<Esc>	<Ctrl-Break>

This page is blank.

Appendix C

EXAMPLE INPUT DATA FILES

Input data files for several real and hypothetical example applications of FESWMS-2DH are contained on the floppy diskette labeled EXAMPLES. Data files for an example are present in a subdirectory on the diskette. Example applications found on the EXAMPLES diskette are described briefly in the following table.

<u>Subdirectory</u>	<u>Example Application Description</u>
ALEX	Flow through a bridge over Alexander Creek near St. Francisville, Louisiana.
BEND90	Flow around a 90 degree bend in a trapezoidal channel. Note the superelevation at the outside of the bend.
ISLAND	Divided flow around an island in a hypothetical channel.
CONTRARY	Flow through a bridge over Contrary Creek. The water surface is in contact with the bridge deck (pressure flow).