Techniques of Water-Resources Investigations
of the United States Geological Survey

Chapter C3

A MODEL FOR SIMULATION OF FLOW IN SINGULAR AND INTERCONNECTED CHANNELS

By R. W. Schaffranek, R. A. Baltzer, and D. E. Goldberg

Book 7

AUTOMATED DATA PROCESSING AND COMPUTATIONS

Click here to return to USGS Publications

UNITED STATES DEPARTMENT OF THE INTERIOR

JAMES G. WATT, Secretary

GEOLOGICAL SURVEY

Doyle G. Frederick, Acting Director

UNITED STATES GOVERNMENT PRINTING OFFICE, WASHINGTON : 1981

For sale by the Superintendent of Documents, U.S. Government Printing Office
Washington D.C. 20402

PREFACE

The series of manuals on techniques describes procedures for planning and executing specialized work in water-resources investigations. The material is grouped under major headings called books and further subdivided into sections and chapters; section C of Book 7 is on computer programs.

This chapter presents a digital computer model for simulating the unsteady flow regimen occurring in a singular reach or throughout a system of reaches composed of simply or multiply connected one-dimensional-flow channels governed by time-dependent forcing functions and boundary conditions. The model is broadly applicable to a wide range of hydrologic conditions and field situations. Channel geometry need not be prismatic. Reach lengths of the branches and segments used in the model need not be equal. Procedures to be followed in implementing the model to a specific field application are presented in a straightforward, step-by-step manner. Operational modeling capability is achieved by linking the model to a highly efficient storage-and-retrieval routine that accesses a data base containing time series of boundary values. This operational capability is enhanced by optional linkage to an extensive set of digital graphics subroutines. Although the model is well tested and will efficiently produce reliable flow computations for a wide variety of field applications, the user is reminded that achieving successful simulation modeling is not dependent solely on employing a well-formulated model. The user's knowledge and understanding of hydrodynamic principles, his willingness to recognize and abide by the limitations inherent in the model, his imagination and skill – seasoned by experience – in schematizing the prototype for modeling, and his common-sense ability to recognize errant data or results are important attributes, all contributing to successful simulation modeling.

Any use of trade names and trademarks in this publication is for descriptive purposes only and does not constitute endorsement by the U S Geological Survey

TECHNIQUES OF WATER-RESOURCES INVESTIGATIONS OF THE U.S. GEOLOGICAL SURVEY

The U.S. Geological Survey publishes a series of manuals describing procedures for planning and conducting specialized work in water-resources investigations. The manuals published to date are listed below and may be ordered by mail from the **Branch of Distribution, U.S. Geological Survey, 1200 South Eads Street, Arlington, VA 22202** (an authorized agent of the Superintendent of Documents, Government Printing Office).

Prepayment is required. Remittances should be sent by check or money order payable to U.S. Geological Survey. Prices are not included in the listing below as they are subject to change. **Current prices can be obtained by calling the USGS Branch of Distribution, phone (202) 751-6777.** Prices include cost of domestic surface transportation. For transmittal outside the U.S.A. (except to Canada and Mexico) a surcharge of 25 percent of the net bill should be included to cover surface transportation.

When ordering any of these publications, please give the title, book number, chapter number, and "U.S. Geological Survey Techniques of Water-Resources Investigations."

TWI 1-D1. Water temperature–influential factors, field measurement, and data presentation, by H. H. Stevens, Jr., J F. Ficke, and G. F. Smoot. 1975. 65 pages.

TWI 1-D2. Guidelines for collection and field analysis of ground-water samples for selected unstable constituents, by W. W. Wood. 1976. 24 pages.

TWI 2-D1. Application of surface geophysics to ground-water investigations, by A. A. R Zohdy, G. P. Eaton, and D. R. Mabey. 1974. 116 pages.

TWI 2-E1. Application of borehole geophysics to water-resources investigations, by W. S. Keys and L. M. MacCary. 1971. 126 pages.

TWI 3-A1. General field and office procedures for indirect discharge measurements, by M. A. Benson and Tate Dalrymple. 1967. 30 pages.

TWI 3-A2. Measurement of peak discharge by the slope-area method, by Tate Dalrymple and M. A Benson. 1967. 12 pages.

TWI 3-A3. Measurement of peak discharge at culverts by indirect methods, by G. L. Bodhaine. 1968. 60 pages.

TWI 3-A4. Measurement of peak discharge at width contractions by indirect methods, by H. F. Matthai. 1967. 44 pages.

TWI 3-A5. Measurement of peak discharge at dams by indirect methods, by Harry Hulsing. 1967. 29 pages.

TWI 3-A6. General procedure for gaging streams, by R. W. Carter and Jacob Davidian. 1968. 13 pages.

TWI 3-A7. Stage measurements at gaging stations, by T. J. Buchanan and W. P. Somers. 1968. 28 pages.

TWI 3-A8. Discharge measurements at gaging stations, by T. J Buchanan and W. P. Somers. 1969. 65 pages.

TWI 3-A11. Measurement of discharge by moving-boat method, by G. F. Smoot and C. E. Novak. 1969. 22 pages.

TWI 3-B1. Aquifer-test design, observation, and data analysis, by R. W. Stallman. 1971. 26 pages.

TWI 3-B2. Introduction to ground-water hydraulics, a programed text for self-instruction, by G. D. Bennett. 1976. 172 pages.

TWI 3-B3. Type curves for selected problems of flow to wells in confined aquifers, by J. E. Reed.

TWI 3-C1. Fluvial sediment concepts, by H. P. Guy. 1970. 55 pages.

TWI 3-C2. Field methods for measurement of fluvial sediment, by H. P. Guy and V. W. Norman. 1970. 59 pages.

TWI 3-C3. Computation of fluvial-sediment discharge, by George Porterfield. 1972. 66 pages.
TWI 4-A1. Some statistical tools in hydrology, by H. C. Riggs. 1968. 39 pages.
TWI 4-A2. Frequency curves, by H. C Riggs. 1968. 15 pages.
TWI 4-B1. Low-flow investigations, by H. C. Riggs. 1972. 18 pages.
TWI 4-B2. Storage analyses for water supply, by H C. Riggs and C. H. Hardison. 1973. 20 pages.
TWI 4-B3. Regional analyses of streamflow characteristics, by H. C. Riggs. 1973. 15 pages.
TWI 4-D1. Computation of rate and volume of stream depletion by wells, by C. T. Jenkins. 1970. 17 pages.
TWI 5-A1. Methods for determination of inorganic substances in water and fluvial sediments, by M. W. Skougstad and others, editors. 1979. 626 pages.
TWI 5-A2. Determination of minor elements in water by emission spectroscopy, by P. R. Barnett and E. C. Mallory, Jr. 1971. 31 pages.
TWI 5-A3. Methods for analysis of organic substances in water, by D. F. Goerlitz and Eugene Brown. 1972. 40 pages.
TWI 5-A4. Methods for collection and analysis of aquatic biological and microbiological samples, edited by P. E. Greeson, T. A. Ehlke, G. A. Irwin, B. W. Lium, and K. V. Slack. 1977. 332 pages.
TWI 5-A5. Methods for determination of radioactive substances in water and fluvial sediments, by L. L. Thatcher, V. J. Janzer, and K. W. Edwards. 1977. 95 pages.
TWI 5-C1. Laboratory theory and methods for sediment analysis, by H. P. Guy. 1969. 58 pages.
TWI 7-C1. Finite difference model for aquifer simulation in two dimensions with results of numerical experiments, by P. C. Trescott, G. F. Pinder, and S. P. Larson. 1976. 116 pages.
TWI 7-C2. Computer model of two-dimensional solute transport and dispersion in ground water, by L. F. Konikow and J. D. Bredehoeft. 1978. 90 pages.
TWI 7-C3. A model for simulation of flow in singular and interconnected channels, by R. W. Schaffranek, R. A. Baltzer, and D. E. Goldberg.
TWI 8-A1. Methods of measuring water levels in deep wells, by M S. Garber and F. C. Koopman. 1968. 23 pages.
TWI 8-B2. Calibration and maintenance of vertical-axis type current meters, by G. F. Smoot and C. E. Novak. 1968. 15 pages.

CONTENTS

FIGURES

TABLES

INTERNATIONAL SYSTEM OF UNITS (SI) AND INCH-POUND SYSTEM EQUIVALENTS

SI unit	Inch-pound equivalent
Length	
centimeter (cm) =	0.3937 inch (in)
meter (m) =	3.281 feet (ft)
kilometer (km) =	0.6214 mile (mi)
Area	
centimeter^2 (cm^2) =	0.1550 inch^2 (in^2)
meter^2 (m^2) =	10.76 feet^2 (ft^2)
kilometer^2 (km^2) =	0.3861 mile^2 (mi^2)
Volume	
centimeter^3 (cm^3) =	0.06102 inch^3 (in^3)
meter^3 (m^3) =	35.31 feet^3 (ft^3)
=	8.107×10^{-4} acre-foot (acre-ft)
Volume per unit time	
meter^3 per second (m^3/s) =	35.31 feet^3 per second (ft^3/s)
=	1.585×10^4 gallons per minute (gal/min)
Mass per unit volume	
kilogram per meter^3 (kg/m^3) =	0.06243 pound per foot^3 (lb/ft^3)
gram per centimeter^3 (g/cm^3) =	6.243×10^{-5} pound per foot^3 (lb/ft^3)
Temperature	
degree Celsius (°C) =	(degree Fahrenheit − 32)/1.8 (°F)

SYMBOLS AND UNITS

Symbol	Definition	Unit
A	cross-sectional area	L^2
$\mathbf{A}$	coefficient matrix	
B	channel top width	L
$\mathbf{B}$	vector of constants	
c	wave celerity ($\sqrt{gH}$)	LT^{-1}
C_d	water-surface drag coefficient	
dA	a finite elemental area	L^2
f	a function	
$f(I)$	functional representation of a dependent variable	
g	acceleration of gravity	LT^{-2}
H	hydraulic depth (A/B)	L
i	subscript index which denotes a function's spatial location	
j	superscript index which denotes a function's temporal location	
k	function of the flow-resistance coefficient $\left(\frac{\eta}{1.49}\right)^2$ in inch-pound system; η^2 in metric system)	$T^2L^{-2/3}$
K	channel conveyance $\left(\frac{1}{\sqrt{k}} AR^{2/3}\right)$	L^3T^{-1}
n	Manning's flow-resistance coefficient	$TL^{-1/3}$
Q	flow discharge	L^3T^{-1}

Symbol	Definition	Unit
$Q_1^{I}, \ldots, Q_4^{III}$	elements of the vector of unknowns, $\boldsymbol{X}$, representing the flow discharge of a branch (I,II,III) at a junction (1,2,3,4)	L^3T^{-1}
Q_m	flow discharge of mth branch at a junction	L^3T^{-1}
R	hydraulic radius of cross section	L
S	slope of energy line	
$\boldsymbol{S}$	vector of state	
t	time	T
Δt	time increment	T
u	flow velocity at a point	LT^{-1}
$\boldsymbol{u}$	transformation matrix	
$u_1^{I}, \ldots, u_2^{III}$	elements of the vector of constants, $\boldsymbol{B}$, which define, in part, the branch-transformation equation of a branch (I,II,III)	
$\boldsymbol{u}_{(i)}$	transformation matrix of ith segment	
$\boldsymbol{u}_n$	transformation matrix of nth branch	
U	mean velocity in cross section	LT^{-1}
$\boldsymbol{U}$	transformation matrix	
$U_{11}^{I}, \ldots, U_{22}^{III}$	elements of the coefficient matrix, $\boldsymbol{A}$ which define, in part, the branch-transformation equation of a branch (I,II,III)	
$\boldsymbol{U}_{(i)}$	transformation matrix of ith segment	
$\boldsymbol{U}_n$	transformation matrix of nth branch	
U_a	wind velocity	LT^{-1}
W	nodal flow	L^3T^{-1}
W_k	nodal flow at kth junction	L^3T^{-1}
x	longitudinal distance along channel	L
Δx	length	L
Δx_i	length of ith segment	L
$\boldsymbol{X}$	vector of unknowns	
$z_1(t), \ldots, z_4(t)$	elements of the vector of constants, $\boldsymbol{B}$, representing the time-dependent water-surface elevation at a junction (1,2,3,4)	L
Z	water-surface elevation	L
$Z_1^{I}, \ldots, Z_4^{III}$	elements of the vector of unknowns, $\boldsymbol{X}$, representing the water-surface elevation of a branch (I,II,III) at a junction (1,2,3,4)	L
Z_m	water-surface elevation of mth branch at a junction	L
α	angle between wind direction and x-axis	(deg)
β	momentum coefficient	
γ	flow-equation coefficient	
δ	flow-equation coefficient	
ϵ	flow-equation coefficient	
ζ	flow-equation coefficient	
η	flow-resistance coefficient similar to Manning's n	$TL^{-1/3}$
θ	spatial-derivative weighting factor	
λ	flow-equation coefficient	
μ	flow-equation coefficient	
ξ	wind-resistance coefficient $\left(C_d \frac{\rho_a}{\rho}\right)$	
ρ	water density	ML^{-3}
ρ_a	atmospheric density	ML^{-3}
σ	flow-equation coefficient	
χ	weighting factor for function values	
ω	flow-equation coefficient	
$\sim$	superscript notation used to signify function values derived from the previous iteration	

A MODEL FOR SIMULATION OF FLOW IN SINGULAR AND INTERCONNECTED CHANNELS

By R. W. Schaffranek, R. A. Baltzer, and D. E. Goldberg

Abstract

A one-dimensional numerical model is presented for simulating the unsteady flow in singular riverine or estuarine reaches and in networks of reaches composed of interconnected channels. The model is both general and flexible in that it can be used to simulate a wide range of flow conditions for various channel configurations. The channel geometry of the network to be modeled should be sufficiently simple so as to lend itself to characterization in one spatial dimension. The flow must be substantially homogenous in density, and hydrostatic pressure must prevail everywhere in the network channels. The slope of each channel bottom ought to be mild and reasonably constant over its length so that the flow remains subcritical. The model accommodates tributary inflows and diversions and includes the effects of wind shear on the water surface as a forcing function in the flow equations. Water-surface elevations and flow discharges are computed at channel junctions, as well as at specified intermediate locations within the network channels.

The one-dimensional branch-network flow model uses a four-point, implicit, finite-difference approximation of the unsteady-flow equations. The flow equations are linearized over a time step, and branch transformations are formulated that describe the relationship between the unknowns at the end points of the channels. The resultant matrix of branch-transformation equations and required boundary-condition equations is solved by Gaussian elimination using maximum pivot strategy.

Five example applications of the flow model are illustrated. The applications cover such diverse conditions as a singular upland river reach in which unsteady flow results from hydropower regulations, coastal rivers composed of sequentially connected reaches subject to unsteady tide-driven flow, and a multiply connected network of channels whose flow is principally governed by wind tides and seiches in adjoining lakes.

The report includes a listing of the FORTRAN IV computer program and a description of the input data requirements. Model supporting programs for the processing and input of initial and boundary-value data are identified, various model output formats are illustrated, and instructions are given to permit the production of graphical output using the line printer, electromechanical pen plotters, cathode-ray-tube display units, or microfilm recorders.

Introduction

Advent of efficient, economical electronic computation during the past two decades has had a profound effect on our means for conducting water-resources assessments. The two principal disciplines underpinning water-resources science – theoretical hydromechanics and experimental hydraulics – have been joined by the new and still emerging discipline of *computational hydromechanics.* This new discipline, while lying essentially at the intersection of theoretical hydromechanics, numerical analysis, and computer science and drawing upon the developmental progress in each of the others, is being recognized as an independent branch of knowledge in its own right. It makes use of the parametric information derived from hydraulic experimentation as well. In effect, computational hydromechanics is providing the means for transforming the theoretical knowledge of hydromechanics into useful and practical tools for water-resources study. The numerical model presented in this report is an example of just such a tool.

Research of flow simulation modeling in riverine and estuarine systems began in the U.S. Geological Survey in the late 1950's. The objective was to provide a strong physical basis for the development of methods with which to determine unsteady flows in channels affected by tides, flood waves, or hydropower regulation or where flow inertial effects were appreciable. Various numerical methods for treating the complete Saint Venant wave-propagation equations were studied, and various models were constructed and reported in the literature (Baltzer and Shen, 1961; Lai, 1965a, b). The earliest models were designed to treat only a single reach of channel since the numerical methods were primitive and the computational capabilities of the day rather limiting. Models capable of representing systems composed of two or more sequentially connected reaches (Baltzer and Lai, 1968) were quickly followed by models capable of depicting dendritic channel systems comprising many connected subreaches. However, these early models lacked the support of a comprehensive modeling system with which to easily effect their implementation and a computer data base with which to broaden the scope of their use. Consequently, each new model implementation was done in an ad hoc manner. In the early 1970's work was begun on a general purpose computerized system, including a boundary-value data base and other supportive files, designed specifically for modular use in simulation modeling. Use of this comprehensive modeling system in conjunction with the branch-network flow model – the name given to the numerical simulation model described in this report – is demonstrated subsequently. Whether or not to use the model with the supportive modeling system is optional.

The branch-network flow model is based upon the one-dimensional, nonlinear partial-differential equations governing unsteady flow in channels for which the dependent variables are the flow rate, Q, and the water-surface elevation, Z. The application of the model is subject to the basic assumptions and limitations inherent in the equations' formulation as described in the report. The equations include terms accommodating the shear-stress effect of wind on the water surface and the Boussinesq momentum-correction coefficient permitting adjustment for nonuniformity of flow in the channel cross section. The partial-differential equations are discretized and replaced by the appropriate finite-difference equations according to a weighted, four-point scheme. Weighting factors governing the discretized quantities of functional values and space derivatives in the finite-difference equations are specifiable, thus providing the model user the flexibility to vary the implicit-solution technique from a box-centered scheme on the one hand to a fully forward scheme on the other. The model is unconditionally linearly stable throughout this range.

The branch-network flow model, as described and documented in this report, is a broadly applicable, proven model. It is intended for operational use and is applicable to any channel (branch) or system of channels (network of branches) subject to backwater flow, unsteady flow, or both, whether caused by the ocean tides, flood waves, seiches, wind, or man-induced regulation. It may be implemented after data for the appropriate channel geometry and initial conditions descriptive of a prototype are obtained and when sequences of synchronous, precisely timed, boundary-value data are provided at its boundary extremities. The model is designed to efficiently compute unsteady flow and water-surface elevation of either singular or interconnected channels. In general, a prototype waterway may be as simple as one channel with an appropriate set of boundary-value data defined at its extremities or as complex as a system of interconnected channels offering multiple flow paths and requiring boundary-condition definition at several external locations. A typical network composed of branches (reaches) and segments (subreaches) is illustrated in figure 1. Although the flow rates and water-surface elevations that occur at the end points of each of the segments could be computed directly, an important feature of this model is the incorporation of a tranformation technique that results in a very significant savings in required computer time and storage. The transformation is accomplished by grouping the segments into branches, forming a transformation equation to relate the unknowns between consecutive segments within a branch, and using the resultant branch-transformation equations to form a coefficient matrix much reduced in size over what would otherwise be the case. Model flexibility permits the user to define segments and branches as may be appropriate. Moreover, the user may designate tributary inflows at internal junctions since boundary compatibility conditions at all such junctions are resolved automatically.

Aspects of the model and its implementation are presented in a thorough yet concise manner. In this regard it is assumed that the model user has an elementary knowledge of the hydromechanics of open-channel flow, of finite-difference methods for solving partial-differential equations, and of matrix algebra. Moreover, a basic familiarity with

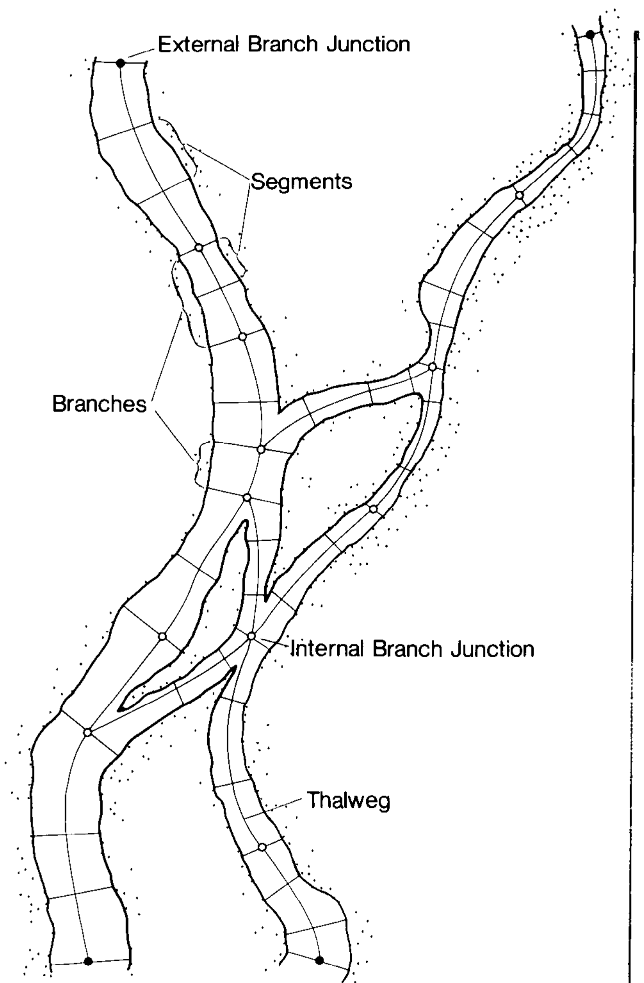


FIGURE 1. – Definition schematic of a hypothetical network.

modern computers and their operating systems and with the FORTRAN IV programing language is presumed.

The discussions of channel properties, cross-sectional geometry, initial and boundary-value data, computational control parameters, and model calibration and verification are intended to be reasonably self-contained. The effects of selecting different computational weighting factors and of inadvertently using various types of errant data are illustrated for a calibrated model of one particular prototype flow system. The structure, manner of operation, and input of data to the branch-network-model program – replete with operational examples – are fully documented. A listing of the current version of the computer program is presented in Appendix IV of this report. Inherent program limitations, resulting primarily as a consequence of a priori selection of array dimensions do exist, necessarily, and are identified. However, the program may be easily modified to accommodate prototype networks having unique dimension requirements. The model user is specifically informed what variables must be adjusted to effect such changes.

The ultimate usefulness of a simulation model depends in large measure on two factors: first, its adaptability to a broad variety of prototype conditions and, second, the ease with which it can be implemented, modified to reflect changes to the prototype, and used to generate results that can be comprehended easily and quickly. Tabular listings and digital-graphic plots, illustrating some of the available output formats, are presented for five different applications of the model depicting a broad scope of prototype conditions.

Acknowledgments

The authors hereby express their appreciation for the assistance of their colleagues at U.S. Geological Survey field offices in California, Connecticut, Idaho, Michigan, and Washington, who were responsible in large measure for the collection of the prototype data used in this report. The authors are grateful, also, to the many Federal, State, and local governmental agencies in each of those States who, through their cooperation with the U.S. Geological Survey, have contributed financially to the collection of these data.

The water-level data used in the development of the flow model of the Detroit River were provided by the National Ocean Survey, National Oceanic and Atmospheric Administration. The cooperation of our colleagues in the National Ocean Survey in making these data available to us is gratefully appreciated.

Unsteady-Flow Equations for Networks of Open Channels

Flow equations

The one-dimensional partial-differential equations governing transient flow in open channels have been reported previously in the literature (Baltzer and Lai, 1968; Dronkers, 1964, 1969; Strelkoff, 1969; Yen, 1973). The system of differential equations presented by Baltzer and Lai constitutes the basis for the open-channel-network flow equations. Using the water-surface elevation, Z, and the channel discharge, Q, as dependent variables, the equation of continuity can be written as

$$B\frac{\partial Z}{\partial t}+\frac{\partial Q}{\partial x}=0,$$

in which B is the channel top width, as shown in figure 2. The distance, x, in the longitudinal direction and the elapsed time, t, are the independent variables. The equation of motion for one-dimensional open-channel flow can be obtained as

$$\frac{\partial Q}{\partial t}+\frac{Q}{A}\frac{\partial Q}{\partial x}+Q\frac{\partial (Q/A)}{\partial x}+qA\frac{\partial Z}{\partial x}+\frac{gk}{AR^{4/3}}Q|Q|=0,$$

in which g is the acceleration of gravity, A is the cross-sectional area, R is the hydraulic radius, and k is a function of the flow-resistance coefficient, η (similar to Manning's n), which can be expressed in the inch-pound system of units as

$$k=\left(\frac{\eta}{1.49}\right)^2 \text{ (or in the metric system as } k=\eta^2\text{).}$$

When wind effect is taken into consideration the equation of motion becomes

$$\frac{\partial Q}{\partial t}+\frac{Q}{A}\frac{\partial Q}{\partial x}+Q\frac{\partial (Q/A)}{\partial x}+gA\frac{\partial Z}{\partial x}+\frac{gk}{AR^{4/3}}Q|Q|-\xi BU_a^2\cos\alpha=0,$$

in which U_a is the wind velocity vector making an angle α with the positive x-axis and ξ is a dimensionless wind-resistance coefficient, which can be expressed as a function of the water-surface drag coefficient, C_d, the water density, ρ, and the air density, ρ_a, as

$$\xi=C_d\frac{\rho_a}{\rho}.$$

The applicability of these equations is governed by several underlying assumptions that arise in the derivation process. Specifically, the slope of the channel bottom must be mild and

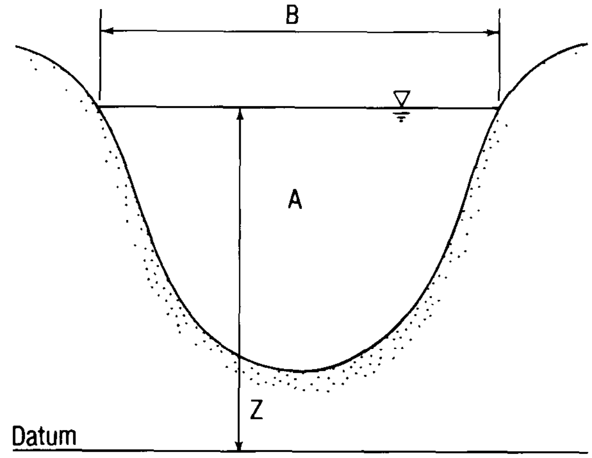


FIGURE 2.—Definition sketch of a channel cross section.

reasonably constant over the reach length, so that the flow remains subcritical. Lateral flow into or out of the channel must be negligible between channel junctions. The Manning formula is assumed to provide an accurate approximation of the frictional-resistance force for unsteady as well as steady flow. Furthermore, it is assumed that flow in the channel is substantially homogenous in density, that hydrostatic pressure exists everywhere in the channel, and that a uniform velocity distribution prevails throughout any cross section.

Since the flow velocity in most natural rivers and waterbodies typically varies throughout the cross section, a more realistic set of flow equations can be obtained by re-examining the equation of motion, thereby seeking to relax the uniform-velocity-distribution assumption. From a statistical analysis of turbulent flow behavior, one finds that the instantaneous flow velocity at a point consists of the mean velocity for the cross section plus the local component of deviation from the mean. When the velocity distribution over the channel cross section becomes highly nonuniform, it becomes necessary to account for these velocity deviations in the equation of motion. Taking into account these velocity fluctuations, one derives the following form of the equation of motion (Schaffranek, 1976):

$$\frac{\partial Q}{\partial t}+\beta\frac{Q}{A}\frac{\partial Q}{\partial x}+Q\frac{\partial (\beta Q/A)}{\partial x}+gA\frac{\partial Z}{\partial x}+\frac{gk}{AR^{4/3}}Q|Q|-\xi BU_a^2\cos\alpha=0,$$

where β, known as the momentum or Boussinesq coefficient, can be expressed as

$$\beta=\frac{\int u^2\,dA}{U^2A}.$$

In this relationship, derived from the conservation of momentum principle, u represents the velocity of water passing through some finite elemental area, dA, U is the mean velocity in the cross section, and A is the cross-sectional area as previously defined. In the equation the variation of β with respect to longitudinal distance is assumed to be negligible.

For channels of regular cross section and fairly straight channel alinement, it may be reasonable to neglect the minor effect of the nonuniform velocity distribution by setting the momentum coefficient equal to one. That this is possible for such channels is due primarily to the diminished significance of this effect as opposed to the effects of inaccuracies in determining the channel schematization, as well as the bottom and surface friction coefficients. In reality, however, the momentum coefficient for flows in natural rivers and waterbodies will always be greater than unity. It is generally found that the value of β for fairly straight prismatic channels ranges approximately from 1.01 to 1.12 (see Chow, 1959). Generally, the coefficient is larger for small channels and smaller for large channels of considerable depth. Consequently, the above equation of motion for a nonuniform velocity distribution is utilized to broaden the scope of applicability of the flow model.

Further reduction of the equation of motion results in a form more amenable to finite-difference approximation. Factoring the equation and separating the derivative of the quotient in the third term, one can obtain the following form:

$$\frac{1}{gA}\frac{\partial Q}{\partial t}+\frac{2\beta Q}{gA^2}\frac{\partial Q}{\partial x}-\frac{\beta Q^2}{gA^3}\frac{\partial A}{\partial x}+\frac{\partial Z}{\partial x}+\frac{k}{A^2R^{4/3}}Q|Q|-\frac{\xi B}{gA}U_a^2\cos\alpha=0.$$

This equation and the equation of continuity, restated for convenience,

$$B\frac{\partial Z}{\partial t}+\frac{\partial Q}{\partial x}=0,$$

represent the flow equations utilized in the branch-network flow model.

Boundary conditions

Solution of the flow equations requires specification of boundary conditions throughout the duration of the simulation at the physical extremities of the network, as well as at branch junctions within the network. Equations describing the boundary conditions at branch junctions are automatically generated by the branch-network-model program, whereas boundary-condition equations for the network extremities are derived from user-supplied time histories of boundary-value data or formulated from user-specified functions.

Compatibility conditions at internal junctions

The most common boundary condition encountered in networks of interconnected channels occurs at junctions where two or more branches join. This situation typically occurs where a channel is joined by a tributary or where a channel is divided by the presence of an island. At such internal junctions, stage (water-surface elevation) and discharge compatibility conditions must be satisfied. By neglecting velocity differences and energy losses due to turbulence at the junctions, appropriate compatibility conditions can be specified. For a junction composed of n branches, discharge continuity requires that

$$\sum_{m=1}^{n} Q_m=W_k,$$

where W_k is some specified external flow at junction k. Since the stage at a junction is single valued, stage compatibility equations are applied as follows:

$$Z_m=Z_{m+1},\ m=1,2,\ldots,(n-1).$$

Therefore, at an internal junction of n branches there are one discharge continuity and $n-1$ stage compatibility conditions that must be satisfied.

Boundary conditions at external junctions

In addition to the required boundary conditions at internal junctions, boundary conditions must be specified at all external junctions, that is, junctions with a singular connecting branch as identified in figure 1. Various combinations of boundary conditions can be specified at the external junctions of a network. External boundary conditions can consist of a zero discharge (as, for example, at a dead-end branch), known discharge as a function of time, known stage as a function of time, or a known unique stage-discharge relationship. Boundary conditions, defined by time-sequences of discrete boundary-value data, can be made available to the branch-network flow model via punched computer cards or from computer data files of the direct-access type.

Initial conditions

In order to initiate a solution of the system of equations with the specified boundary conditions, initial values of the unknown quantities are required. These values may be obtained from measurements, computed from some other source, such as steady-state approximations, or computed from previous simulations. Successive use of the newly computed values as initial values permits the computation to proceed step-by-step until the boundary-value data are exhausted or the simulation is otherwise terminated. Successful convergence of the computation to the correct solution requires that the initial values be reasonably accurate; the less accurate the initial values, the longer the computation takes to dissipate the initialization error and converge to the true solution.

Solution Technique for Open-Channel Network Flow Simulation

The set of nonlinear partial-differential equations describing unsteady flow in open-channel networks defies analytical solution. Approximate solutions can be obtained by replacing the partial-differential equations by appropriate finite-difference expressions. In the branch-network flow model a weighted, four-point, finite-difference scheme is employed, and the resultant system of algebraic equations is solved simultaneously. This weighted, four-point, implicit-solution technique is used because of its inherent computational efficiency, stability, and versatility with respect to the application of boundary conditions.

Finite-difference formulation

Formulation of the finite-difference equations consists of treating time derivatives of the dependent variables, stage and discharge, as centered both in space and in time and of treating spatial derivatives of the dependent variables as centered in space and positioned in time according to a user-defined weighting factor. The lone exception is the spatial derivative of the cross-sectional area in the equation of motion, which is approximated by a forward-difference technique. The geometric properties of area, top width, and hydraulic radius, as well as the discharges in nonderivative form in the equation of motion, are treated as weighted, four-point, difference quantities in a fashion similar to the approximation of the spatial derivatives of the dependent variables. (Discrete values of the hydraulic radius are approximated by the hydraulic depth, which is the cross-sectional area divided by the channel top width.) Thus, these functional values can be approximated at the time level at which the spatial derivatives are defined or at any other different level within the time interval.

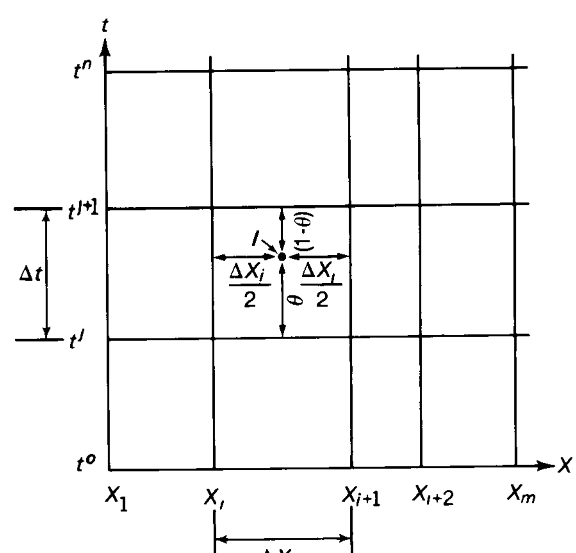


FIGURE 3.—Space-time-grid system for finite-difference approximations.

As can be seen from the space-time-grid system shown in figure 3, the four points used in the approximatios are identified by the intersections of the vertical lines at distances x_i and x_{i+1} with the horizontal lines representing the time levels at t^j and t^{j+1}. The computational scheme uses a fixed time step, Δt, but permits the subdivision of branches into segments of equal or unequal lengths. In the finite-difference approximations, Δx_i represents the length of the ith segment of any given branch. From the space-time-grid system of figure 3, time and space derivatives of the functional value, $f(I)$, are approximated, respectively, as follows:

$$\frac{\partial f(I)}{\partial t} \approx \frac{f_{i+1}^{j+1}+f_i^{j+1}-f_{i+1}^j-f_i^j}{2\Delta t},$$

and

$$\frac{\partial f(I)}{\partial x} \approx \theta \frac{f_{i+1}^{j+1}-f_i^{j+1}}{\Delta x_i}+(1-\theta)\frac{f_{i+1}^j-f_i^j}{\Delta x_i}.$$

In the approximation of the spatial derivatives, $f(I)$ represents the dependent variables, stage and discharge, and θ is a weighting factor determining the time between the t^j and t^{j+1} time lines at which the spatial derivatives are evaluated. The spatial derivative of the cross-sectional area in the equation of motion is approximated by a forward-difference technique as

$$\frac{\partial A}{\partial x} \approx \frac{A_{i+1}^{j+1}-A_i^{j+1}}{\Delta x_i}.$$

In a manner similar to the treatment of the spatial derivatives, quantities such as the cross-sectional area, top width, hydraulic radius, and the discharge in nonderivative form in the equation of motion, represented by $f(I)$, are approximated by

$$f(I) \approx \chi \frac{f_{i+1}^{j+1}+f_i^{j+1}}{2}+(1-\chi)\frac{f_{i+1}^j+f_i^j}{2}.$$

In this expression χ is a weighting factor, similar to θ, specifying the time at which these functional quantities are evaluated between the t^j and t^{j+1} adjacent time levels at the midpoint of the ith segment.

In the four-point, finite-difference scheme, θ is a real constant, generally thought of as lying in the interval $0 \leq \theta \leq 1$. If θ is not zero, one must solve a set of simultaneous linear equations, and therefore it is called an implicit system. When θ is less than 0.5 the four-point, implicit, finite-difference equations are found to be conditionally linearly stable. The equations are found to be unconditionally linearly stable when θ is greater than or equal to 0.5 and less than or equal to 1.0. A weighting factor value of 0.5 yields the traditional box scheme used by Preissman (1960) and by Amein and Fang (1970), whereas a θ value of 1.0 represents the fully forward scheme presented by Baltzer and Lai (1968).

The weighting factor χ is similarly taken as lying in the interval $0 \leq \chi \leq 1$. If χ is zero, function values are determined exclusively from previous time step quantities, whereas a χ weighting factor of one produces a fully forward approximation of the applicable functions.

Computational effects of various values for the θ and χ weighting factors are discussed in subsequent sections, and appropriate value ranges are suggested. Stability and other numerical properties of the four-point, implicit, finite-difference scheme are also discussed in detail by Fread (1974).

Utilizing these finite-difference approximations and the notation $\tilde{f}(I)$ to signify function values derived from the previous iteration, one can transform the partial-differential flow equations into the following finite-difference expressions for the ith segment: for the continuity equation,

$$\tilde{B}\left[\frac{Z_{i+1}^{j+1}+Z_i^{j+1}}{2\Delta t}-\frac{Z_{i+1}^j+Z_i^j}{2\Delta t}\right]+\theta\frac{Q_{i+1}^{j+1}-Q_i^{j+1}}{\Delta x_i}+(1-\theta)\frac{Q_{i+1}^j-Q_i^j}{\Delta x_i}=0,$$

and for the equation of motion,

$$\frac{1}{g\tilde{A}}\left[\frac{Q_{i+1}^{j+1}+Q_i^{j+1}}{2\Delta t}-\frac{Q_{i+1}^j+Q_i^j}{2\Delta t}\right]+\frac{2\beta\tilde{Q}}{g\tilde{A}^2}\left[\theta\frac{Q_{i+1}^{j+1}-Q_i^{j+1}}{\Delta x_i}+(1-\theta)\frac{Q_{i+1}^j-Q_i^j}{\Delta x_i}\right]$$
$$-\frac{\beta\tilde{Q}^2}{g\tilde{A}^3}\frac{\tilde{A}_{i+1}^{j+1}-\tilde{A}_i^{j+1}}{\Delta x_i}+\theta\frac{Z_{i+1}^{j+1}-Z_i^{j+1}}{\Delta x_i}+(1-\theta)\frac{Z_{i+1}^j-Z_i^j}{\Delta x_i}$$
$$+\frac{k|\tilde{Q}|}{\tilde{A}^2\tilde{R}^{4/3}}\left[\chi\frac{Q_{i+1}^{j+1}+Q_i^{j+1}}{2}+(1-\chi)\frac{Q_{i+1}^j+Q_i^j}{2}\right]-\frac{\xi\tilde{B}}{g\tilde{A}}U_a^2\cos\alpha=0.$$

In this solution technique stage and discharge are computed at the ends of the segments identified by the i and $i+1$ locations. The equations consist of four unknown quantities represented by Z_{i+1}^{j+1}, Z_i^{j+1}, Q_{i+1}^{j+1}, and Q_i^{j+1}. Therefore, as it exists, the equation set is indeterminant since the two equations have four unknowns. However, with suitable boundary conditions specified, the number of equations can be increased in order that a solution can be effected by implicit means.

Coefficient-matrix formulation

Solution of the flow equations is conveniently accomplished by matrix methods after appropriate coefficient matrices are constructed. Rewriting the continuity equation as

$$Q_{i+1}^{j+1} - Q_i^{j+1} + \frac{(1-\theta)}{\theta}(Q_{i+1}^{j} - Q_i^{j}) + \frac{\tilde{B}\Delta x_i}{2\Delta t\theta}(Z_{i+1}^{j+1} + Z_i^{j+1} - Z_{i+1}^{j} - Z_i^{j}) = 0,$$

and letting

$$\gamma = \frac{\tilde{B}\Delta x_i}{2\Delta t\theta},$$

one derives the desired coefficient form of the continuity equation in terms of the four unknowns for the ith segment,

$$Q_{i+1}^{j+1} + \gamma Z_{i+1}^{j+1} - Q_i^{j+1} + \gamma Z_i^{j+1} = \delta,$$

wherein

$$\delta = -\frac{(1-\theta)}{\theta}(Q_{i+1}^{j} - Q_i^{j}) + \gamma(Z_{i+1}^{j} + Z_i^{j}).$$

In a like manner, the coefficient form of the equation of motion can be derived from the previously formulated finite-difference equation. After it is factored, the equation of motion for the ith segment can be written,

$$\frac{\Delta x_i}{2\Delta t\theta g\tilde{A}}[Q_{i+1}^{j+1} + Q_i^{j+1} - Q_{i+1}^{j} - Q_i^{j}] + \frac{2\beta\tilde{Q}}{g\tilde{A}^2}\left[Q_{i+1}^{j+1} - Q_i^{j+1} + \frac{(1-\theta)}{\theta}(Q_{i+1}^{j} - Q_i^{j})\right] + Z_{i+1}^{j+1} - Z_i^{j+1} + \frac{(1-\theta)}{\theta}(Z_{i+1}^{j} - Z_i^{j}) + \frac{\chi\Delta x_i k|\tilde{Q}|}{2\theta\tilde{A}^2\tilde{R}^{4/3}}\left[Q_{i+1}^{j+1} + Q_i^{j+1} + \frac{(1-\chi)}{\chi}(Q_{i+1}^{j} + Q_i^{j})\right]$$

$$= \frac{\beta\tilde{Q}^2}{\theta g\tilde{A}^3}(\tilde{A}_{i+1}^{j+1} - \tilde{A}_i^{j+1}) + \frac{\xi\Delta x_i\tilde{B}}{\theta g\tilde{A}}U_a^2\cos\alpha.$$

With the definition of the coefficients,

$$\lambda = \frac{\Delta x_i}{2\Delta t\theta g\tilde{A}}, \quad \mu = \frac{2\beta\tilde{Q}}{g\tilde{A}^2}, \quad \text{and} \quad \sigma = \frac{\chi\Delta x_i k|\tilde{Q}|}{2\theta\tilde{A}^2\tilde{R}^{4/3}},$$

the equation of motion, after substitution and collection of terms, becomes

$$(\lambda+\sigma)[Q_{i+1}^{j+1} + Q_i^{j+1}] + \mu[Q_{i+1}^{j+1} - Q_i^{j+1}] + [Z_{i+1}^{j+1} - Z_i^{j+1}] = \epsilon,$$

wherein

$$\epsilon = \left(\lambda - \sigma\frac{(1-\chi)}{\chi}\right)[Q_{i+1}^{j} + Q_i^{j}] - \mu\frac{(1-\theta)}{\theta}[Q_{i+1}^{j} - Q_i^{j}] - \frac{(1-\theta)}{\theta}[Z_{i+1}^{j} - Z_i^{j}] + \frac{\beta\tilde{Q}^2}{\theta g\tilde{A}^3}[\tilde{A}_{i+1}^{j+1} - \tilde{A}_i^{j+1}] + \frac{\xi\Delta x_i\tilde{B}}{\theta g\tilde{A}}U_a^2\cos\alpha.$$

Finally, with the substitutions $\zeta = \lambda + \sigma + \mu$ and $\omega = \lambda + \sigma - \mu$, the coefficient form of the equation of motion in the four unknown quantities for the ith segment can be written as

$$\zeta Q_{i+1}^{j+1} + Z_{i+1}^{j+1} + \omega Q_i^{j+1} - Z_i^{j+1} = \epsilon.$$

Together with the continuity equation derived previously as

$$Q_{i+1}^{j+1} + \gamma Z_{i+1}^{j+1} - Q_i^{j+1} + \gamma Z_i^{j+1} = \delta,$$

the flow equations for the ith segment can be expressed in the following matrix form

$$\begin{bmatrix} 1 & \zeta \\ \gamma & 1 \end{bmatrix}\begin{bmatrix} Z_{i+1}^{j+1} \\ Q_{i+1}^{j+1} \end{bmatrix} - \begin{bmatrix} 1 & -\omega \\ -\gamma & 1 \end{bmatrix}\begin{bmatrix} Z_i^{j+1} \\ Q_i^{j+1} \end{bmatrix} = \begin{bmatrix} \epsilon \\ \delta \end{bmatrix}.$$

Branch-transformation equation

Using appropriate internal and external boundary conditions and initial values, one may effect a matrix solution directly for the set of flow equations for all segments within the network, a segment being the primary subdivision of a branch as shown in figure 1. The resultant solution set would consist of computed values of stage and discharge at all cross sections delineating the segments. However, the equation set of a network consisting of M segments would form a coefficient matrix of minimum order

$2M+2$ requiring solution at each time step. For instance, a network composed of 10 sequentially connected branches, each subdivided into 5 segments (50 segments in all) would require a 102×102 size coefficient matrix of 10,404 computer-word locations to hold the equation set. Since the computer costs necessary to perform a solution for an equation set of this magnitude could be substantial, it is desirable to examine alternate means of formulating the coefficient matrix.

From the coefficient matrices of the finite-difference equations a transformation equation can be obtained that defines the relationship between the unknowns at consecutive cross sections that delimit a branch segment. By coupling all segment-transformation equations for a branch, a transformation equation results that relates the unknowns at the termini of the branch. By using these branch-transformation equations instead of segment-flow equations, the size of the coefficient matrix is reduced to order $4N$ for a network of N branches. The above-mentioned, ten-branch system, having five segments per branch, would then require only a 40×40 size coefficient matrix of 1,600 computer-word locations. Obviously, the result is a significant savings in computer costs because of reduced computational time and computer storage demands. Values of water-level and discharge at the cross sections delineating the segments within each branch are subsequently derived by back substitution.

Defining a two-component vector of state at the ith cross section,

$$\boldsymbol{S}_i^{j+1} = \begin{bmatrix} Z_i^{j+1} \\ Q_i^{j+1} \end{bmatrix},$$

one may write the following transformation equation for the ith segment from the vector of state for the cross section at the $i+1$ location,

$$\boldsymbol{S}_{i+1}^{j+1} = \boldsymbol{U}_{(i)} \boldsymbol{S}_i^{j+1} + \boldsymbol{u}_{(i)}.$$

The transformation matrices of the ith segment, $\boldsymbol{U}_{(i)}$ and $\boldsymbol{u}_{(i)}$, follow from the previously defined coefficient matrices thusly:

$$\boldsymbol{U}_{(i)} = \begin{bmatrix} 1 & \zeta_{(i)} \\ \gamma_{(i)} & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & -\omega_{(i)} \\ -\gamma_{(i)} & 1 \end{bmatrix}$$

and

$$\boldsymbol{u}_{(i)} = \begin{bmatrix} 1 & \zeta_{(i)} \\ \gamma_{(i)} & 1 \end{bmatrix}^{-1} \begin{bmatrix} \epsilon_{(i)} \\ \delta_{(i)} \end{bmatrix}.$$

A branch-transformation equation can now be obtained through successive application of the segment-transformation equation. The resulting equation relates the unknowns at cross sections 1 and m of the nth branch,

$$\boldsymbol{S}_m^{j+1} = \boldsymbol{U}_n \boldsymbol{S}_1^{j+1} + \boldsymbol{u}_n,$$

wherein the transformation matrices of the nth branch, $\boldsymbol{U}_n$ and $\boldsymbol{u}_n$, are obtained through successive substitution of the segment-transformation equation from the $m-1$ segment down to the first segment. These branch-transformation matrices,

$$\boldsymbol{U}_n = \boldsymbol{U}_{(m-1)} \boldsymbol{U}_{(m-2)} \ldots \boldsymbol{U}_{(1)}$$

and

$$\boldsymbol{u}_n = \boldsymbol{u}_{(m-1)} + \boldsymbol{U}_{(m-1)}(\boldsymbol{u}_{(m-2)} + \boldsymbol{U}_{(m-2)}(\boldsymbol{u}_{(m-3)} \ldots + \boldsymbol{U}_{(3)}(\boldsymbol{u}_{(2)} + \boldsymbol{U}_{(2)}\boldsymbol{u}_{(1)}) \ldots)$$

describe the relationship between the vectors of state, $\boldsymbol{S}_1^{j+1}$ and $\boldsymbol{S}_m^{j+1}$, at the ends of the branch, that is, at the junctions. After a matrix solution is effected producing the stages and discharges at junctions, intermediate values of the unknowns at additional cross sections that delimit the branch segments are successively computed using the segment-transformation equation for the particular branch.

Matrix solution

For a network of N branches, the branch-transformation equations, internal boundary conditions, and external boundary conditions form a linear system of $4N$ equations in $4N$ unknowns. Branch-transformation equations appear first in the matrix followed immediately by internal boundary equations and finally by the external boundary conditions expressed in equation form. The system of equations may be expressed in matrix notation as $\boldsymbol{AX}=\boldsymbol{B}$, where the coefficient matrix $\boldsymbol{A}$ is $4N \times 4N$, $\boldsymbol{X}$ is the vector of $4N$ unknowns, and $\boldsymbol{B}$ is the right-hand column vector of $4N$ constants.

For the hypothetical Y network illustrated in figure 4, the contents of the coefficient matrices are shown in figure 5. The network consists of three branches (branches are identified by superscripts) joining at a single junction (junctions are indicated by subscripts). Boundary conditions at the three external junctions consist of known stage as a function of time. No external flow exists at the internal junction; stage and discharge compatibility conditions at this junction are as specified in the figure.

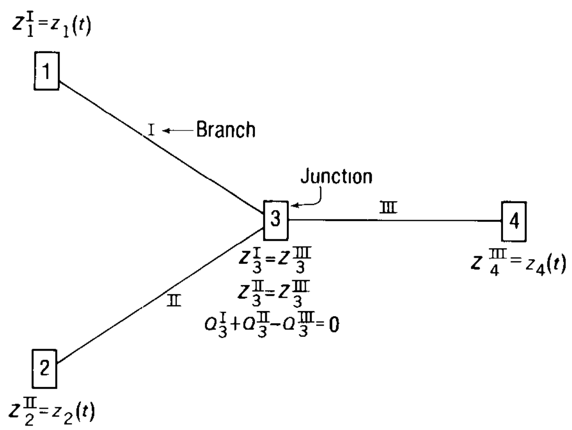


FIGURE 4.—A simple hypothetical network. (Superscripts identify branches; subscripts identify junctions.)

Numerous techniques exist for numerical solution of systems of linear equations such as illustrated in figure 5. One of the most widely used methods for solving simultaneous equations is Gaussian elimination. (The term "elimination" is derived from the process whereby unknowns are successively eliminated by combining equations.) Gaussian elimination is a two-step method. By successive combinations of equations, all of the coefficients in the *A* matrix below the diagonal are eliminated to form an upper triangular matrix. After the upper triangular matrix has been formed, the unknowns are determined by back substitution of lower equations into those higher in the matrix.

The equation used to eliminate the unknowns in the other equations is called the pivot equation. Roundoff errors in the elimination process can be minimized by choosing the equation having the largest coefficient in the column as the pivot equation. This technique, known as Gaussian elimination using maximum pivot strategy, is utilized to effect the matrix solution in the branch-network flow model.

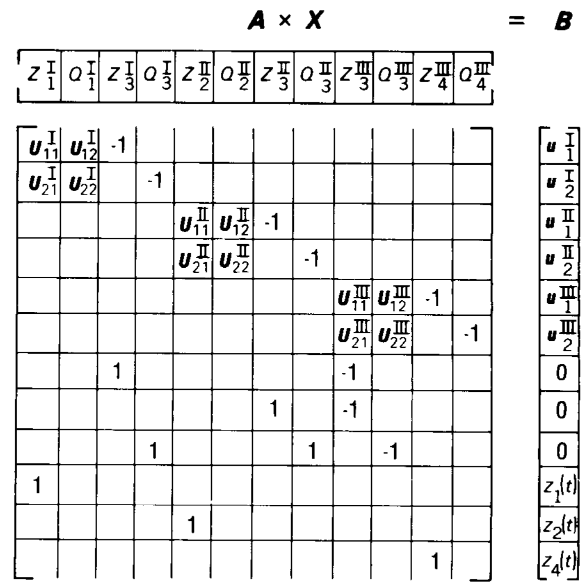


FIGURE 5.—Coefficient matrices for the hypothetical network shown in figure 4.

Solution of the matrix results in determination of the stage and discharge at the termini of the branches (at the junction cross sections). Intermediate values of the unknowns at the termini of the segments (at cross sections located between junctions) are determined through successive application of the segment-transformation equation defined previously.

By use of the initial conditions, that is the values given at t^0, to initiate the simulation, the computation proceeds step-by-step to the end of the simulation at t^n through successive solution of the coefficient matrices. The matrix coefficients contain known quantities at the present time level, t^j, as well as unknowns at the forward time level, t^{j+1}. Iterative solutions may be performed within the time step to refine the computed results and thereby satisfy the user-specified accuracy requirements. Present time-step quantities are obtained either as initial values (at the t^0 time level) or as the last solution values derived in the previous time step (at all subsequent time levels). Forward time-step quantities are obtained initially either from present values or by extrapolation and thereafter from the most recent values derived through iteration within the time step.

Branch-Network Model Implementation

In addition to the hydrodynamic factors governing the applicability of the one-dimensional flow equations, several fundamental considerations constrain flow analysis by one-dimensional methods of approximation. The most obvious constraint is that the individual channel geometry be sufficiently simple so as to lend itself to one-dimensional characterization. Specifically, the channels should be sufficiently "straight" in the longitudinal sense so that the flow may be simulated by flow in straight channels. Channels with bends of significant curvature may also be treated if determination of the flow field in the bend is not important. In this situation the influence of the curvature on the flow field may either be neglected, if deemed minor, or accounted for by means of appropriate values for the momentum and (or) flow-resistance coefficients. All artificial or natural section controls within the channels must be identified and taken into account in the model schematization.

Other constraints related to the water-surface profile are also significant in determining the suitability of a one-dimensional analysis. The water-surface profile at the boundary-value, data-collection locations must be horizontal in the transverse direction to insure proper determination of the water-surface elevations. Furthermore, the overall channel distance between external junctions must be sufficient to permit accurate determination of the longitudinal water-level variation so that adverse effects of measurement errors are minimized. This is particularly important if these measured data are to be used as boundary values or for calibration.

Certainly, many other factors may also enter into determination of the applicability of one-dimensional flow analyses. Some of these have been reported elsewhere by Schaffranek and Baltzer (1978). To attempt to list all the necessary considerations is not possible. However, certain implementation constraints of major concern will be identified as they arise subsequently.

Channel and Cross-Sectional Geometry

Implementation of the branch-network flow model necessitates determination of several physical and hydraulic properties of the prototype system under investigation. Certain of these properties are made readily available through direct field measurements. Others defy direct field quantification, thereby necessitating initial approximation and subsequent refinement throughout the model calibration and verification processes. Nevertheless, accurate flow computation requires proper definition of these properties in order to maintain compatibility with the one-dimensional analytical methods employed.

In order to implement the branch-network flow model it is necessary to accurately describe the prototype system under investigation. In this schematization process, it is essential to precisely identify the branch and junction locations, the branch and segment lengths, the cross-sectional geometry, and all other geometrical and physical properties that affect the flow. As a first step toward fulfilling these data requirements, a review of existing hydrologic and hydrographic information combined with a map and field reconnaissance of the prototype is an essential preliminary activity. This helps to define data collection requirements and field constraints and to determine the scope of the entire modeling effort.

Schematization of a network for flow simulation requires depicting the system by branches delimited by, and possibly connected at, junctions. Junctions are either internal or external to the network. (See fig. 1.) Locations at which two or more channels join or where nodal flow must be accommodated are internal junctions; model-defined boundary conditions supplemented by specified nodal flows are applied at these locations. Junctions at which a single branch is defined are external junctions; user-supplied boundary conditions are required at these locations. External junctions define the extremities of the network being simulated. In deciding upon the location of external junctions, it is, of course, economically advantageous to establish their position so as to utilize data collected at existing field-station locations to satisfy boundary-condition requirements. However, where such data are not readily available and a location and means must be decided upon for acquiring the necessary boundary values, logistic and economic, as well as hydraulic considerations, play a significant role in the decision process. After junction locations are assigned, and the branches are thereby defined, a decision must be made whether or not to subdivide the branches into segments. Geometric and hydraulic factors, as well as computational considerations, are the basis upon which the subdivision of branches into segments is determined. Once the branches and segments have been delineated, their lengths can be determined by field surveys or by measuring along the channel thalweg as depicted on topographic maps or marine charts. (In one-dimensional analyses the x-axis is taken as either the thalweg or map centerline of the channel, and the y-axis is the cross-channel coordinate.)

Cross-sectional geometry must be defined at the termini of all segments. Cross-sectional information consists of stage-area and stage-width relationships supplied in tabular form. The required cross-sectional geometry can be approximated from hydrographic survey charts, such as may be available from the National Ocean Survey, NOAA, or from the U.S. Army, Corps of Engineers, or measured directly by standard hydrographic survey techniques. Direct field measurement is recommended to insure that the model accurately depicts the current prototype conditions.

Channel Conveyance Parameters

In addition to the channel and cross-sectional properties required to conduct flow simulations, definition of certain other channel parameters is critical. Accurate definition of the flow-resistance coefficient is always necessary. Proper specification of the velocity distribution (momentum) coefficient and the wind-shear, water-surface drag coefficient may also be required depending on the particular flow conditions. While it is often difficult to quantify these parameters precisely, reasonable approximations are obtainable.

Of the channel parameters identified, perhaps the most difficult to quantify is the flow-resistance coefficient. This is particularly true since the flow-resistance coefficient is typically a compound function of the physical and hydraulic properties of the channel. Thus, the flow-resistance coefficient, although principally dependent on the channel roughness, may be affected by inherent minor inaccuracies in the chosen schematization of the prototype. Complex networks also compound the problems associated with determination of the flow-resistance coefficient.

As stated in the section Flow equations, the Manning equation for steady uniform flow is assumed to provide a reasonable approximation of the frictional resistance expected for unsteady flow. In the Manning equation,

$$U=\frac{1.49}{\eta}R^{2/3}S^{1/2}$$

or, in the metric system,

$$U=\frac{1}{\eta}R^{2/3}S^{1/2},$$

R is the hydraulic radius, S is the slope of the energy gradient, U is the velocity and η is the flow-resistance coefficient. The flow-resistance coefficient notation, η, is used in place of Manning's n to indicate, first, that the coefficient represents an unsteady flow situation and, second, that it may also be accounting for schematization inaccuracies. However, the flow-resistance coefficient should never exceed reasonable bounds. In fact, it should never vary greatly from its corresponding steady-flow approximation determined from the Manning equation. Such deviation, should it occur, must be interpreted as a signal of trouble and is very probably indicative of inappropriate use of one-dimensional techniques, schematization and (or) data inaccuracies, or excessive distortion of the prototype in the chosen schematization.

Accurate determination of the flow-resistance coefficient is often difficult. For unsteady-flow computation in a channel or network in which approximately steady-flow conditions occur, an η value equivalent to Manning's n, determined by the foregoing equation, may be used. For the more difficult situation where steady-flow or nearly steady-flow conditions never prevail, the flow-resistance coefficient may be obtainable only by a trial-and-error process beginning with an initial estimate and successively adjusting the value until satisfactory flow results are achieved. If measured discharges are available, it is also possible to numerically compute η by methods described by Baltzer and Lai (1968).

Often flow resistance varies under changing flow conditions. It may be necessary, therefore, to treat the flow-resistance coefficient as a function of time. For example, it has been shown that the flow-resistance coefficient frequently varies with the Reynolds number (Baltzer and Lai, 1968). The variable behavior of the flow-resistance coefficient has also been linked to changes in the bed regime of the channel bottom or to extreme changes in the water temperature. Similar correlations of the flow-resistance coefficient have been detected and reported by others. Therefore, in the branch-network flow model the flow-resistance coefficient can be defined as a linear or quadratic function of the water temperature, discharge, flow depth, Froude number, or Reynolds number.

Accurate simulation of the flow conditions may also necessitate evaluation of the momentum coefficient as defined in the section Flow equations. This may be required for channels characterized by contractions and (or) expansions, meandering paths of travel, or cross-sectional irregularities, such as islands, sand bars, or gullies. In such highly nonprismatic or curved channels a uniform velocity distribution does not exist.

Evaluation of the momentum coefficient may be accomplished directly from field-measured horizontal and vertical velocity profiles by use of the integral definition. However, if available data are insufficient to determine the coefficient directly, other velocity distribution profiles may be examined and used to approximate the coefficient and thereby produce more accurate flow results than would otherwise be possible under the uniform-velocity-distribution assumption. In order to effect an approximation it is important to note that the momentum coefficient for the cross-sectional area is separable into two components accounting for the transverse and vertical deviations of the velocity from their respective mean values. The momentum coefficient for the cross-sectional area is the product of these individual components (Schaffranek, 1976). Various semiempirically based theories have been proposed that provide accurate, realistic, laboratory- and field-substantiated approximations to the vertical velocity profile of natural channels and waterways; examples include Prandtl's mixing length theory, G. I. Taylor's vorticity transfer theory, and Von Kármán's similarity hypothesis. Similarly, more accurate approximation methods, such as exponential or logarithmic distribution forms, are available to depict the transverse distribution of flow in natural channels. Accurate evaluation of the momentum coefficient by realistic definition of the transverse and vertical flow distribution will obviously serve to improve the accuracy of the flow computations. The momentum coefficient is always greater than or equal to one; a value of one implies a uniform velocity distribution. For the turbulent flows typical of most natural channels, the momentum coefficient is on the order of 1.06.

Under some conditions it may also be necessary to account for wind-induced currents caused by wind stress acting on the water surface within the network. Thus, wind has been included as a forcing function in the branch-network flow equations. Evaluation of the wind effect requires specification of the air and water densities, the wind speed and direction (usually measured at standard anemometer height), and a water-surface drag coefficient. The difficulty in accurately simulating wind effect rests with selection of the most suitable value for the water-surface drag coefficient. Experimentation has shown that the value of this coefficient depends not only on the flow depth but also on the height, steepness, and celerity of the wind-generated surface waves. Representative values of the water-surface drag coefficient appear to range between 1.5×10^{-3} for light winds and 2.6×10^{-3} for strong winds (Wilson, 1960). Determination of the appropriate value may require analysis of the flow under various wind conditions. Plots of the water-surface drag coefficient versus wind speed are illustrated in Neumann and Pierson (1966).

Initial and boundary-value data

Three comprehensive sets of data are required to carry out flow simulations by mathematical-numerical models such as the branch-network flow model. Initial-condition data, channel geometry, and boundary-value data (usually water-surface elevations precisely timed and synchronized) constitute mandatory data requirements for flow computation by the branch-network flow model.

In addition to the required channel and cross-sectional data, which were described in the section Channel and cross-sectional geometry, initial values of the unknown quantities must be determined and supplied as initial conditions. As mentioned in the section Initial conditions, these values consisting of water-surface elevations and discharges at the termini of all segments can be obtained directly by field measurement, computed from previous simulations, approximated from some other source (such as assumed steady-state conditions), or simply estimated. By use of unsteady-flow data for the Sacramento River, model convergence for various deviations of the initial discharge from its true value is illustrated in figure 6. For initial values that are in error by as much as 100 percent the convergence time of the model is found to be roughly 2 hours (eight time steps) with a maximum deviation of 12.5 percent after five time steps (1¼ hours). At 0200 hours, 2 hours after the start of the simulation, all computed discharges are within 2 percent of the true value. Thus, while reasonable initial conditions are desirable, estimates can be used for starting conditions if a sufficient amount of "warmup" time is provided for the model to dissipate the errors and converge to the true solution. It is noteworthy that models of flow systems having high rates of energy dissipation will converge more rapidly than those having low rates of energy dissipation (Lai, 1965a, b).

Boundary conditions for the branch-network flow model consist of two types. First, stage and discharge compatibility conditions must be satisfied at junctions where two or more branches join internal to the network. Assignment of internal boundary conditions is accomplished automatically by the branch-network-model program. Using branch and junction identification information provided as input parameters, the program constructs the internal boundary-condition equations and fills the appropriate elements of the coefficient matrices with values describing these boundary conditions.

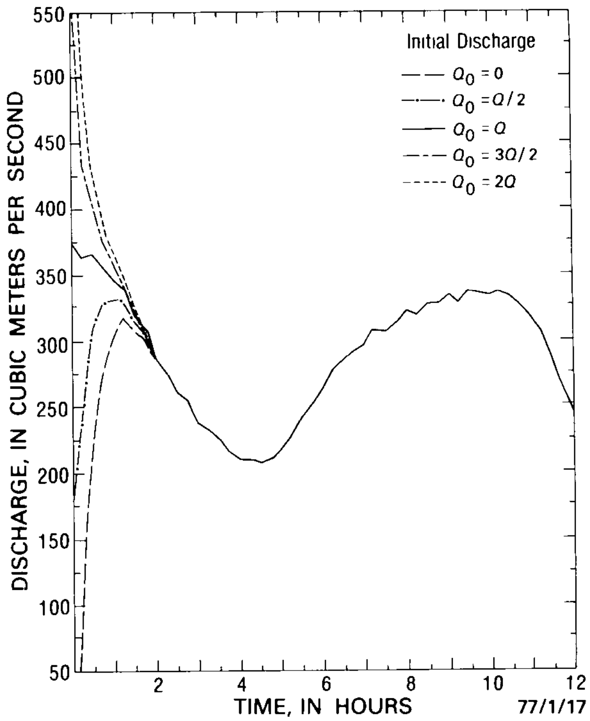


FIGURE 6.—Discharges computed for the Sacramento River near Freeport, Calif., by use of various initial values in the branch-network flow model.

The second type of boundary condition must be defined by the model user. Such boundary conditions occur at the external junctions of the network, that is, at a junction consisting of a singular connecting branch. External boundary values may be specified either as a stage hydrograph or a discharge hydrograph or be prescribed by a unique stage-discharge relationship. The most commonly used external boundary condition is the stage hydrograph, since precise synchronous recording of the water-surface elevations can be accomplished automatically. Digital recorders actuated by a float or by a gas back-pressure servomanometer and timed by precision crystal timers encode these data on 16-channel punched-paper tapes at preselected time intervals. Subsequently, these punched-paper tapes of field-recorded, time-dependent data are read by optical means, and the data are telemetered for processing on the U.S. Geological Survey computer system. Transmitted data are concurrently translated to extended binary coded decimal (EBCDIC) characters and temporarily recorded on digital magnetic tape. The magnetic tape containing the transmitted time-dependent data, identified by field-station number, data recording frequency, and beginning and ending dates and times, is then ready for subsequent translation, editing, and filing. An array of computer programs, referred to as the time-dependent data processing system, is available for use to edit and file such time-dependent data in order to provide boundary conditions for one-, two-, and three-dimensional models of flow and (or) transport. After processing, these data are made readily available for direct inclusion in the branch-network flow model. They can be retrieved directly by the model from the edited time-dependent data base by identifying the field-station number, data type, recording frequency, dates, and times of interest on data-request program-control cards input to the model. For data recorded at frequencies different from the computational time increment used in the model, parabolic interpolation is performed by the model to determine values consistent with the chosen time step.

All significant external inflows and outflows of the network must also be determined and identified in the model implementation. Constant inflows and (or) outflows in the network are presently treated as occurring at branch junctions. Therefore, it may be necessary to "lump" lateral flow between junctions, such as bank seepage, and define it as a point source occurring at one or more appropriate junctions. In the model schematization, inflow is typically taken to be positive in sign.

Computation-control parameters

Flow simulation by the branch-network flow model requires specification of several parameters that principally control the numerical computation process. Determination of appropriate values for these computational-control parameters is important because they have an effect on the accuracy, convergence, and stability of the model. Three primary considerations, critical with regard to controlling the numerical computation, are determination of the simulation time increment, definition of the segment lengths, and selection of appropriate finite-difference weighting factors. Other considerations, such as the accuracy of the initial conditions and the value of the tolerance limits, are also important aspects affecting the numerical computation process.

Numerical solution of the flow equations on a rectangular x–t grid system, whether by explicit finite-difference techniques or by the method-of-characteristics technique, imposes a constraint on determination of the computational time increment based on definition of the segment lengths. This constraint is not applicable, however, in a rigorous mathematical sense, to the implicit solution technique, such as is employed in the branch-network flow model. Characteristic and explicit schemes are subject to the Courant restriction, which imposes the following constraint upon the time-increment to segment-length ratio

$$\frac{\Delta t}{\Delta x} \leq \frac{1}{|U \pm \sqrt{gH}|} .$$

In this relationship Δt is the time increment, Δx is the distance increment (or segment length), U is the mean flow velocity, g is the acceleration of gravity, and H is the depth of flow. Adherence to this restriction is necessary in order to assure stable computational conditions when utilizing method-of-characteristics or explicit solution techniques. This restriction on the time increment can frequently cause characteristic and explicit solution schemes to require excessive amounts of computer time; for this reason implicit solution techniques offer distinct economical advantages. Nevertheless, the Courant condition, which is a function of the wave celerity ($c=\sqrt{gH}$) and the flow velocity, remains a valuable index when selecting a time increment for implicit solutions, as well. For a fixed length, the Courant condition restricts the time increment of characteristic and explicit solutions accordingly,

$$\Delta t \leq \frac{\Delta x}{|U \pm \sqrt{gH}|}$$

On the other hand, various numerical simulations using the branch-network flow model have remained stable for large time steps appreciably exceeding the value imposed by the Courant restriction. One must be aware, however, that increasing the time increment may also degrade the accuracy of the simulation and thereby render the results useless. The time step used in the branch-network flow model may safely exceed the Courant value by a factor of two to five without undue degradation of the computed results. However, each model implementation is unique; thus, the appropriate time increment should be judiciously determined. The amount by which the Courant condition can be safely exceeded is a function of the weighting factor, θ, and the ratio of the critical segment length to the primary, translatory wave length.

Several simulation results using various time steps are illustrated in figure 7. The flow shown in the illustration is for the Sacramento River near Freeport, Calif. This application of the branch-network flow model treats the Sacramento River between Sacramento and Freeport, Calif., as one branch. The total branch length is 17.4 km, and it is treated as a single segment in the model simulations. Stages recorded at Sacramento (station number 11–4475.00) and near Freeport (station number 11–4476.50) on January 17, 1977, and used as boundary-value data for the model, are shown in figure 8. For the given flow conditions – maximum flow velocity of 0.45 m/s and wave celerity of 7.0 m/s – the time increment imposed by the Courant condition is approximately 40 minutes. The model was run with time steps of 15 minutes, 1 hour, 2 hours, and 4 hours. The 15-minute time step abides by the Courant condition whereas the 1-hour, 2-hour, and 4-hour time steps are, respectively, 1.5, 3, and 6 times the Courant value. As can be seen from figure 7 the computation remains stable for all time steps tested. However, the simulation results obtained using the larger time steps may not be very usable and, in fact, present a rather crude approximation of the flow profile. Therefore, it is always important to select a time step that produces the best approximation of the flow conditions and, hence, the most usable results.

Oftentimes model output results are used for special purposes, such as to drive a transport model. This may require computation at some predetermined time increment. For this situation the segment lengths can also be selected using the Courant condition as an index. Usually in one-dimensional flow analyses the segment lengths are set on the order of 10 times the width. However, the segment length specification is primarily a function of the flow accuracy requirements.

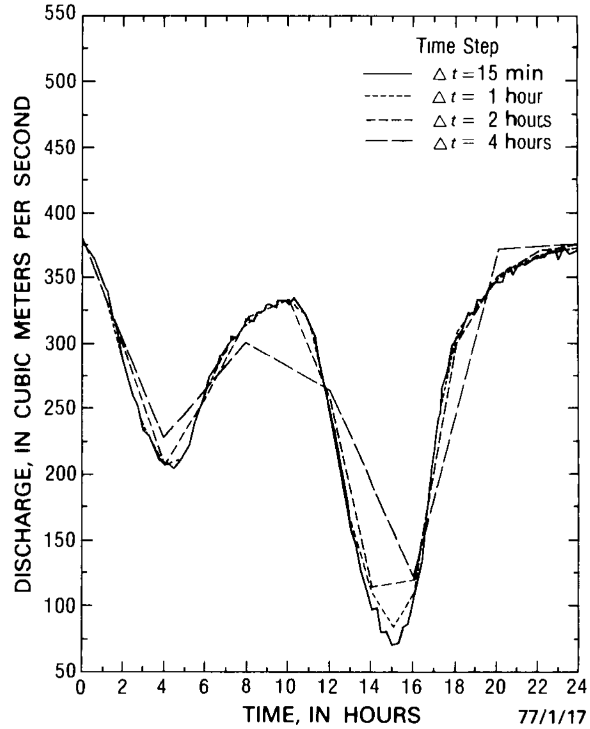


FIGURE 7.—Discharges computed for the Sacramento River near Freeport, Calif., by use of various time increments in the branch-network flow model.

Thus, although the time-step to segment-length ratio presented by the Courant condition need not be strictly upheld, the conservative properties of the model, and hence the accuracy of the results, are best for values close to the Courant criterion. Therefore, the Courant criterion is a valuable index for selecting the time step to be used in the branch-network flow model.

Another equally important consideration in implementation of the branch-network flow model is selection of an appropriate value of the finite-difference weighting factor for the spatial derivatives. Utilizing the weighted, implicit,

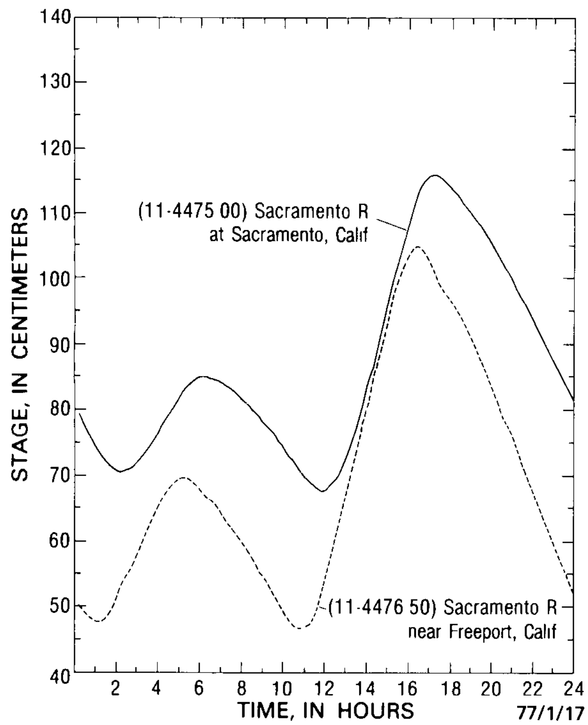


FIGURE 8.—Water-surface elevations for the Sacramento River at Sacramento, Calif., and near Freeport, Calif., used as boundary-value data for the branch-network flow-model simulation shown in figure 7.

four-point difference scheme affords considerable flexibility in simulating various transient-flow conditions. Consequently, it is very important to carefully analyze the flow simulations using various weighting factors for the range of flow conditions expected and to judiciously select the most appropriate value for θ. The effect of the weighting factor on the stability of the solution is illustrated in figure 9. This analysis was accomplished by using the model to simulate perturbated steady flow in the Sacramento River at Sacramento, Calif. In these simulations the stages at Sacramento and near Freeport, used as boundary conditions to drive the model, were fixed at 6.15 m and 4.89 m, respectively, for which the steady flow discharge amounted to 1903.3 m^3/s. The initial discharge conditions were set slightly higher at 1903.7 m^3/s in order to introduce a perturbation on the flow for conducting the analysis and to illustrate the convergence properties of the model. As figure 9 shows, the model exhibits oscillations in the flow computation for θ values less than 0.6. These oscillations are small, symetrical about the true solution, and after their initial development, neither grow nor decay with time. This phenomenon, referred to as psuedoinstability or computational mode, can be negated by taking θ greater than or equal to 0.6, as shown in figure 9. The selection of appropriate values for θ is, however, largely dependent on the particular flow conditions being simulated and the solution time increment. For most transient-flow conditions a reasonable value for the spatial-derivative weighting factor appears to be $0.6 \leq \theta \leq 1.0$. The branch-network flow model defaults to the fully forward scheme specified by θ equal to 1.0; this value of θ produces the most stable computational mode.

Although the direct (noniterative) solution technique in the branch-network flow model produces acceptable results when the selected time step and the chosen schematization are appropriate for the prototype, iteration within the time step is available and can be used to produce results of a higher order accuracy. Such iterative improvement of computed results is optional. Two methods are provided for controlling the iterative refinement process. These controls are intended to be used in tandem. First, the maximum allowable number of iterations can be user specified. Secondly, convergence criteria – the maximum acceptable difference between successive results computed through iteration within the time step – can be specified for both stage and discharge. Default conditions are a maximum of five iterations per time step with the discharge convergence criterion established from the initial values and the stage convergence criterion set at 0.01 ft for data input in the inch-pound system and 1 cm for metric data input. The default discharge convergence criterion is set at one-half of one percent of the minimum (absolute value greater than zero) initial discharge specified. If all initial discharges are zero the discharge convergence criterion is assigned a value of one by default. For the initial convergence test at each new time step the unknowns are automatically set by the model to current time-step values or to model-extrapolated values. The model default is no extrapolation. These parameters provide complete flexibility in controlling the iteration process, thus allowing the model user to tailor

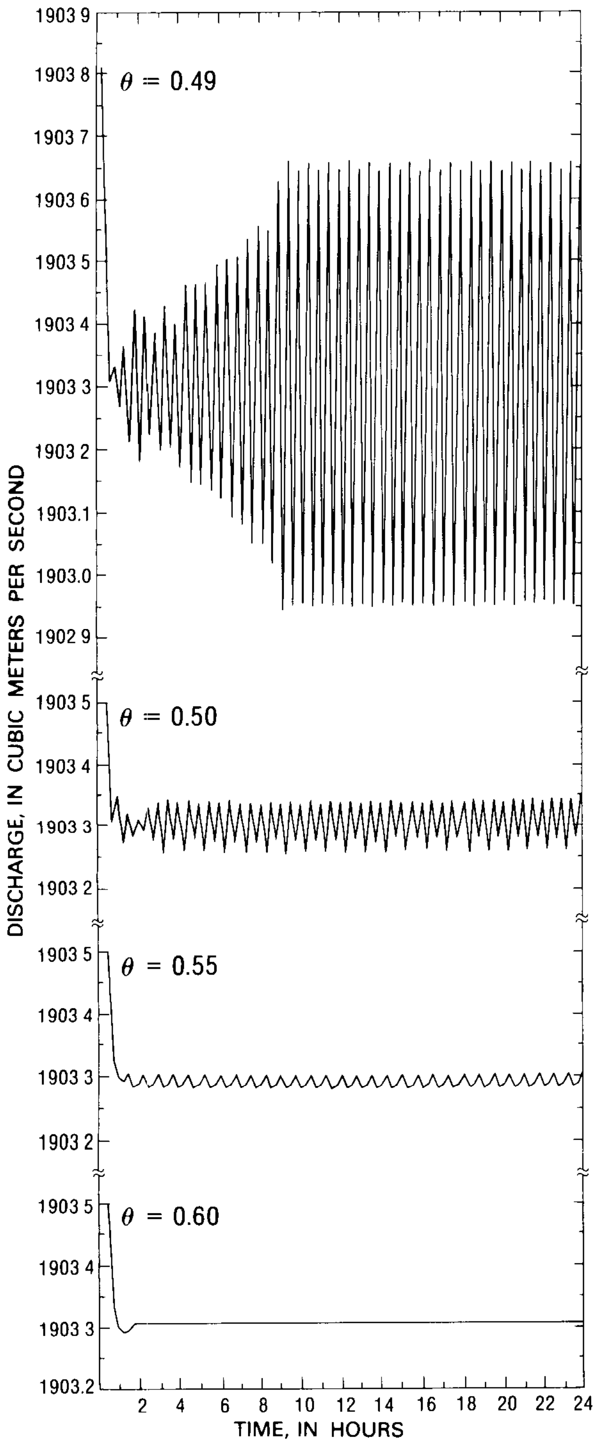


FIGURE 9.—Steady flow computations for the Sacramento River at Sacramento, Calif., by use of various finite-difference weighting factors (θ) in the branch-network flow model.

the computation to suit the accuracy requirements of his particular application. Several applications of the model for various flow conditions have shown that default convergence criteria are generally satisfied in three or less iterations per time step.

Calibration and verification

The success of any flow model application depends greatly on the availability of accurate prototype data. Prototype data are needed for flow-model calibration, that is, determination or refinement of the least quantifiable parameters, such as the flow-resistance coefficient and the water-surface drag coefficient; flow simulations can be properly assessed only through verification with good-quality observed data. In the calibration process the objective is to adjust these parameters to accurately replicate the prototype system for the range of flow conditions expected. Although model reproduction of a prototype water-level hydrograph is of some verification value, a more rigorous verification is achieved by model reproduction of the prototype discharge hydrograph in both phase and amplitude. Once this level of calibration is achieved it is possible to extend the utility of the model to simulate flow conditions beyond the calibration range, if sufficient checks are made to insure the validity of the calibration for such extreme flow conditions.

Data for model calibration and verification consist of time series of measured discharges, together with concurrently recorded water-surface elevations. The length and frequency of required discharge measurements are largely dependent on the unsteady nature of the flow. However, economic and operational constraints often control the duration and frequency of such measurements. In general, for tide-driven systems, one or more measurements, usually of tidal-cycle duration, are scheduled each year. The location of such measurements is based principally on model needs as determined by the prototype flow conditions. Oftentimes, selection of the location at which discharge measurements are to be made is a matter of determining where the flow may be most accurately measured, given the physical and economical constraints on field operations.

All aspects of a particular model schematization are subject to adjustment during the calibration process. However, aspects that are physically well defined and that can be measured, and thus determined with reasonable accuracy, are considerably less subject to adjustments than are those for which measured values are lacking or for which direct determination is impossible. For instance, reach lengths are definable and can be accurately measured; they should not be subject to alteration during calibration. Likewise, channel geometry data are generally not altered during calibration, since such data can be measured with reasonable accuracy. Moreover, water-level boundary-value data, presumably recorded synoptically with precise time synchrony and properly referenced to a common datum, should not require alteration. Yet it is frequently just in this regard that errors occur and model calibration difficulties do arise. Occasionally, one or more water-level recorders may be incorrectly referenced to datum or correctly referenced to an incorrect datum, with the result that not only are the time-sequences of boundary-value data not correct but perhaps the channel geometry may become improperly referenced to stage as well. Recorded boundary-value data that lack time synchrony infer distorted wave-propagation rates within the model and result in phase and amplitude calibration problems. Thus, failure to maintain synchronous operation of the water-level recorders or the presence of errors in the datum reference of these recorders may result in an improperly schematized model and erroneous, often confusing, model results. (One must be particularly sensitive to "force fitting" a model calibration using data of questionable validity.) In order to avoid an unnecessarily complicated and prolonged model-calibration process, it is important to identify and correct errors in the directly measurable quantities as early in the calibration process as possible. It is best, of course, if by thorough planning and diligent field operation, such errors can be avoided entirely.

Once the accuracy of the directly measurable quantities has been verified, the principal aspect of model calibration, namely, determination of the channel conveyance properties, remains to be accomplished. Initially, these properties can only be approximated, estimated, or otherwise inferred. Thus, the conveyance properties, principally the conveyance due to channel resistance expressed in terms of the Manning equation as

$$K=\frac{1.49}{\eta}AR^{2/3},$$

or, in the metric system, as

$$K=\frac{1}{\eta}AR^{2/3},$$

may legitimately require modification during the calibration. However, the value of the coefficient, η, must always be physically realistic.

Under certain flow conditions the conveyance produced by wind shear acting on the water surface and that resulting from the nonuniform distribution of the flow velocity may also be significant and necessarily reauire consideration in the calibration process.

The difficulty in evaluating the flow-resistance coefficient for calibrating the flow model stems principally from the fact that the energy-dissipation relationship is an approximation borrowed from the realm of steady, uniform flow. Little factual information is known regarding the effect of boundary-shear resistance under unsteady-flow conditions. Knowledge of the effect of boundary friction is mainly empirical, a fact that complicates model calibration and verification but is not an insurmountable problem. The applicable flow-resistance coefficient is, as noted in the section, Channel conveyance parameters, not only a function of such hydraulic properties as the flow depth, turbulence, and water temperature but also of the schematization of the prototype. Therefore, it is necessary to use the model to arrive at appropriate values, since for different schematizations, different values of η may be found.

In order to conduct the model calibration several accurate sets of discharge data are necessary. It is desirable to have such discharge information for a range of flow conditions. If possible, the model calibration process should begin with a set of data gathered during steady or nearly steady flow conditions. The initial estimate of the flow-resistance coefficient can be determined from the Manning equation or can be simply estimated. The calibration process then involves successive adjustment of the conveyance factors until satisfactory agreement between the computed and measured flow data is achieved. The model calibration must then be verified by comparative testing with other independent sets of observed-flow data. The model is successfully verified if the computed results agree well with the field-observed data. If agreement is not good, the model calibration parameters must be further adjusted until such agreement is achieved or the cause of the deviation is identified.

By use of the calibrated model of the Sacramento River, several simulations have been conducted to illustrate the effect of changes in the flow-resistance coefficient. In figure 10 the results of these simulations are compared with both measured discharges and with computed discharges from the calibrated model.

A too small value for η reduces flow resistance, thus increasing momentum or inertia. The simulation performed with a decreased flow-resistance coefficient not only results in a discharge hydrograph having peaks and troughs of larger magnitude (fig. 10) but also results in a phase shift, which is not easily discernible in figure 10. On the other hand, an excessive value for η shows the reverse effect. Thus, the model adjustment critically depends on the use of proper η values.

Additional simulations were conducted to illustrate the effects of changes in boundary-value data and cross-sectional geometry. The results of these simulations are compared with both measured discharges and with computed discharges taken from the calibrated model as shown in figures 11 and 12.

Recorded water-surface elevations may sometimes be incorrectly defined because of datum errors, survey inaccuracies, or vertical displacement (settling) of the gaging structure. The result of such errors is an increase or a decrease in the water-surface slope throughout the channel reach. Increasing the water-surface slope increases the discharge in the downstream direction. Reducing the water-surface slope produces the reverse effect, as can be seen from figure 11. To accommodate such detected errors, it is possible, during the flow simulation, to apply a datum correction to the recorded boundary-value data via an input parameter of the branch-network flow model.

An excessive cross-sectional area, derived by entering a table of stage versus cross-sectional area that represents the schematized channel with too large a value for stage, results in magnification of both positive and negative flows. A too small cross-sectional area, the result of entering a table of stage versus cross-sectional area with too small a value for stage, produces the reverse effect, as can be seen from figure 12. An adjustment to correct the error can be made in the model by using the datum correction input parameter to equally decrease or increase the stage value at the ends of the reach.

Figures 13 and 14 illustrate the computational effects of the weighting factors for the function values and their spatial derivatives in the branch-network flow equations. Selection of appropriate values for these weighting factors is primarily dependent upon the flow conditions and the geometric properties of the prototype. Through use of such variable weighting factors, complete flexibility is provided to accommodate implementation of the model for prototypes having widely varying flow conditions and physical characteristics.

The weighting factor, θ, positions the finite-difference approximation of the spatial derivatives between two adjacent time lines of the space-time grid system. (See fig. 3.) Appropriate values for the spatial derivative weighting factor appear to be in the range $0.6 \leq \theta \leq 1.0$, as noted in the section Computational control parameters. Computational experience has shown that values of θ less than 0.6 have consistently generated unacceptable pseudoinstability (computational mode). The fully forward technique specified by θ equal to 1.0 provides the greatest computational stability. However, it also has the sometimes undesirable characteristic of damping the computed wave. Such numerical damping tends to increase in proportion to increases in the computational time step. Other factors, including the wave celerity, also influence determination of the appropriate θ value. In evaluating θ, an optimal value must be determined so as to minimize numerical damping of the computed transient while at the same time minimizing the

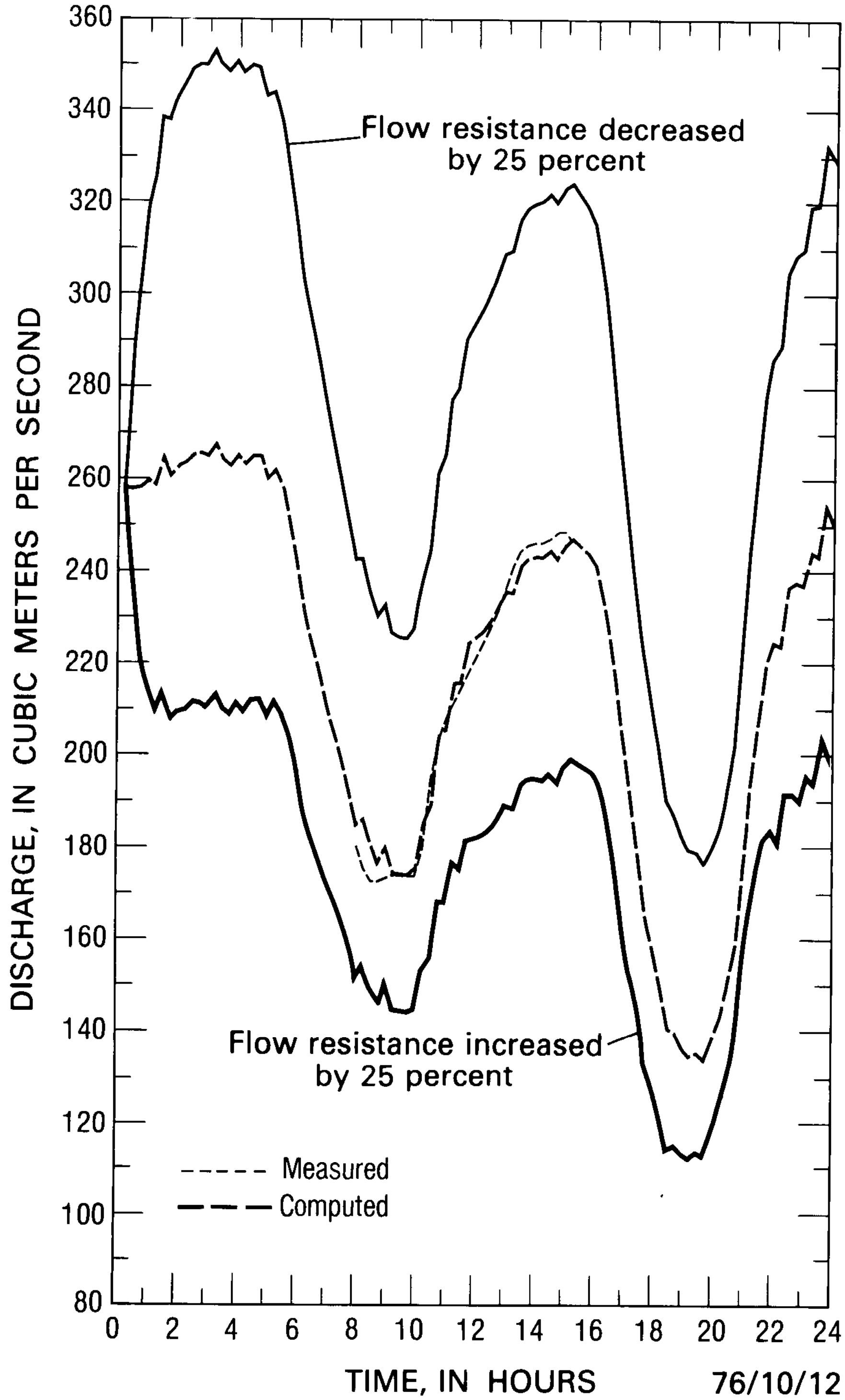


FIGURE 10.—Discharges computed for the Sacramento River at Sacramento, Calif., with flow resistance increased and decreased by 25 percent.

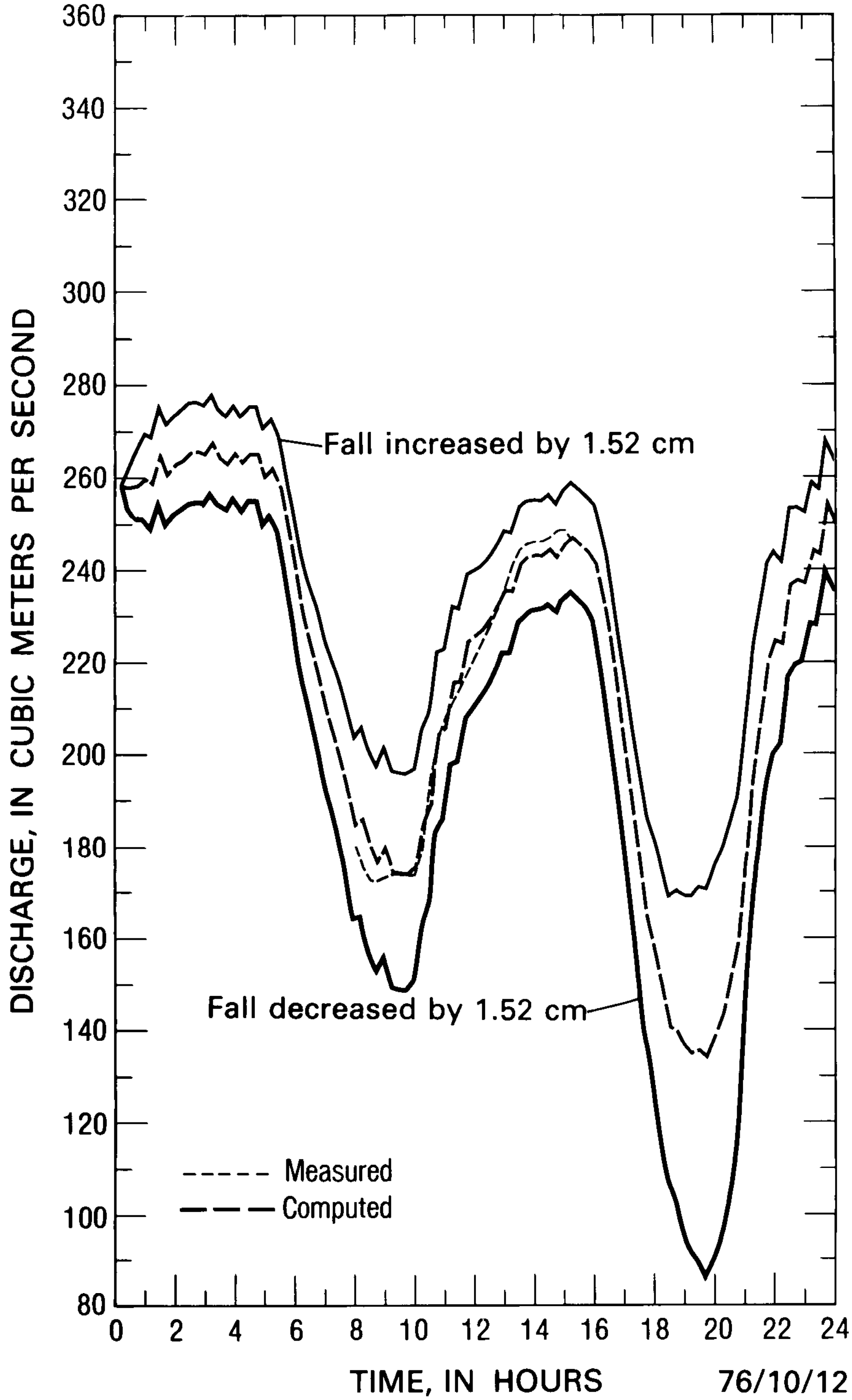


FIGURE 11.—Discharges computed for the Sacramento River at Sacramento, Calif., with the fall increased and decreased by 1.52 cm.

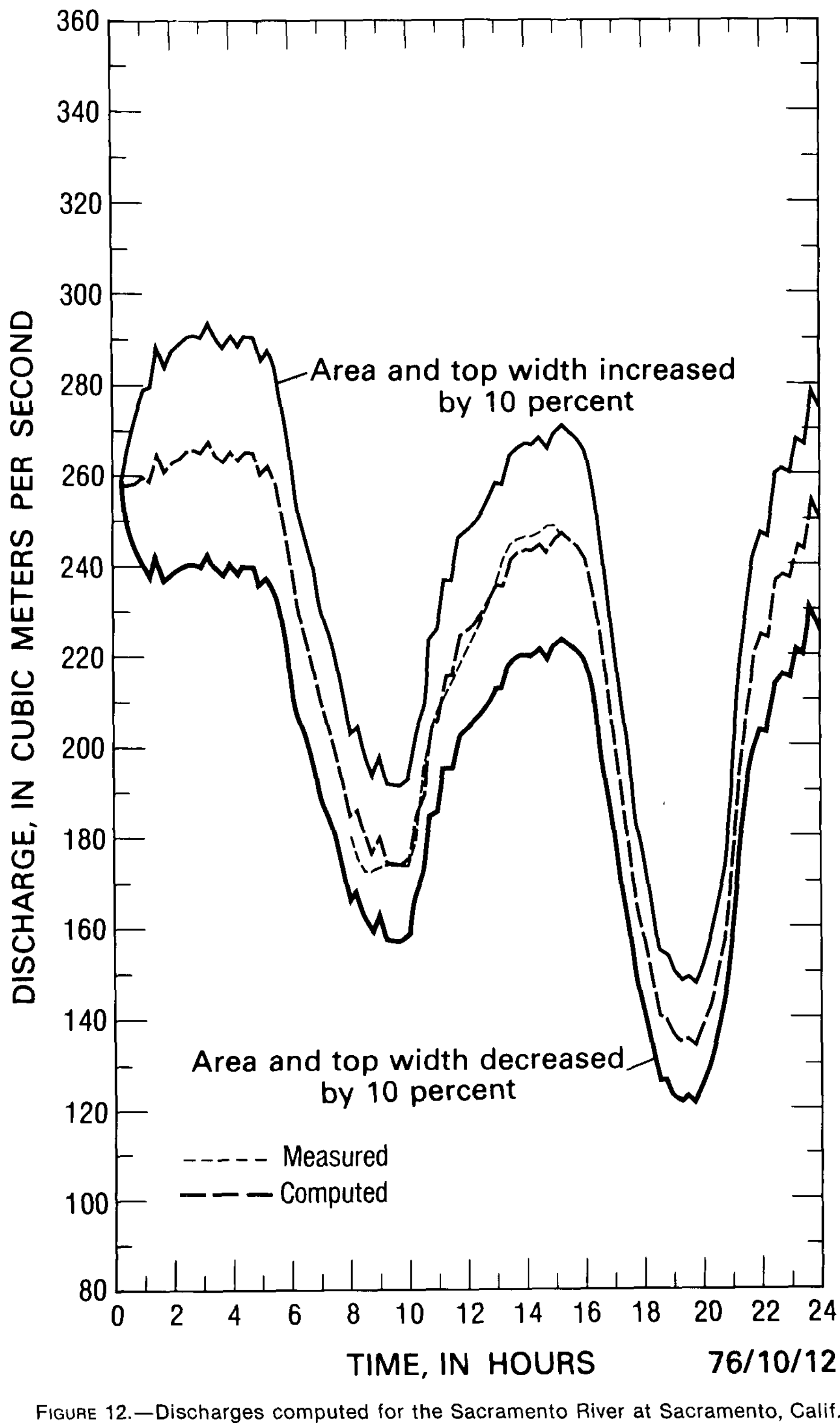


FIGURE 12.—Discharges computed for the Sacramento River at Sacramento, Calif., with the cross-sectional area and top width increased and decreased by 10 percent.

computational mode, as exemplified in figure 9. Figure 13 indicates a weak pseudoinstability condition manifesting itself in the computed discharge hydrograph for the Sacramento River when a value of 0.6 is used for the spatial-derivative weighting factor. Whereas the discharge hydrograph using a value of 1.0 for θ does not exhibit such pseudoinstabilities, the computed transient is somewhat attenuated, thereby implying a compromise θ value is appropriate for this application.

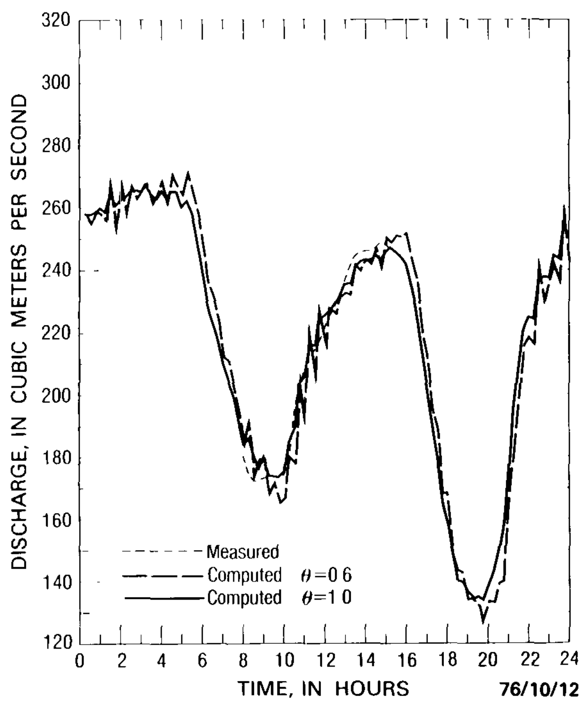


FIGURE 13.—Discharges computed for the Sacramento River at Sacramento, Calif., with various weighting factors (θ) for the spatial derivatives in the branch-network flow model.

In the computation process the geometric properties of area, top width, and hydraulic radius and the discharges in nonderivative form in the equation of motion can be accurately approximated over the time interval by using an appropriate value for the weighting factor, χ. A cursory review of the finite-difference equations for θ and χ, as defined in the section Finite-difference formulation, reveals the similarity between the two weighting factors. Reasonable values for χ fall in the range $0.5 \leq \chi \leq 1.0$. Figure 14 illustrates the effect on the computed discharges using values of 0.5 and 1.0 for χ. The discharge hydrograph computed using a χ value of 0.5 shows a phase lag compared with that computed using values based totally on the advanced time line, that is, χ equal to 1.0. Subtle differences also exist in the computed minimum discharges. The significance of the weighting factor for these quantities appears to increase for channels having increasingly variable geometric properties, such as, narrowing or widening cross sections. Typically, the weighting factor χ is initially set equal to θ and ultimately adjusted as required during the model calibration process.

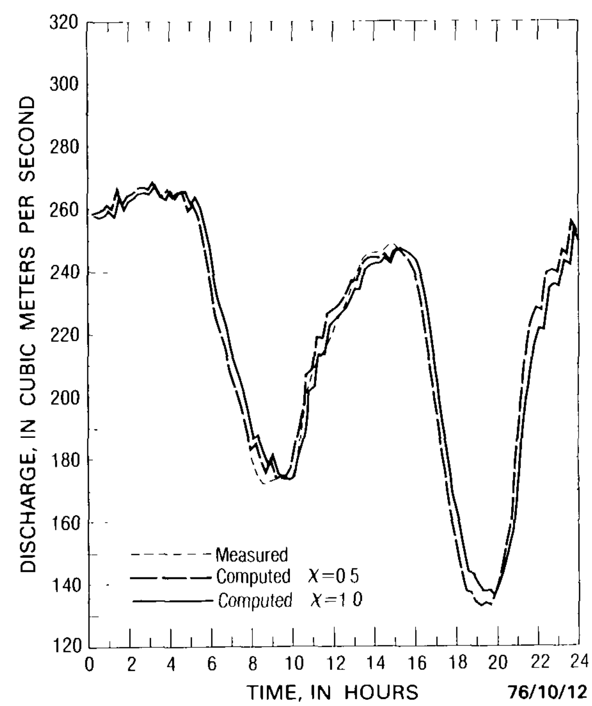


FIGURE 14.—Discharge computed for the Sacramento River at Sacramento, Calif., with various weighting factors (χ) for the function values in the branch-network flow model.

Calibration of network-type flow models may be difficult depending upon the complexity of the flow regime and the interconnection of channels within the prototype. Networks of interconnected open channels generally fall into two categories. In a network of simply connected branches the flow has only one path-of-travel between any two locations in the system. Models of this type of network tend to be easier to calibrate than models of networks of multiply connected branches, wherein it is possible for the flow to travel by more than one route between various locations. If the calibration of a system of multiply connected branches is not correct, erroneous circulations may appear internally in the network rendering the model results useless. Calibration of network models is best conducted by first subdividing the network into simpler, single- or multiple-branch models keeping intact the principal natural circulation loops of the overall system. After successful calibration of these individual models, evaluation of the complete network model can be undertaken by combining the various smaller subset models. By this technique the network model calibration may be accomplished more systematically and economically.

Branch-Network Program Description

The FORTRAN IV code for the branch-network flow model is listed in Appendix IV. The model is composed of a MAIN program and eight primary subroutines, namely, OUT, PRTPLT, ARBIN, SETA, GEMXPI, OPLOT, DADIO, and DTCODE. The subroutines DADIO and DTCODE, which are referenced by the MAIN program but which are not included in the source listing in Appendix IV, are resident in the time-dependent-data, storage and retrieval system library SCHAF.DADIO.-LOADMOD. This library is cataloged on the U.S. Geological Survey computer system. The special system function, MOVE, not identified in the list but used by the MAIN program for fast, efficient array manipulation, is stored in the cataloged FORTRAN library XTENT.LIB. Other lower-level subroutines, aside from the common intrinsic FORTRAN functions, specifically required for the graphical display of computed results and directly referenced via subroutine OPLOT are available in the SYS1.DISSPLA.LIBRARY and SYS1.FLAT-BEDC system program libraries.

For reference purposes and to permit the cross-referencing of program variables and arrays with the mathematical equations, the MAIN program variables and arrays are defined in Appendix II. Commonality of variable and array names among the MAIN program and the various subroutines has been preserved to the extent practicable.

The bulk of the work of the simulation is performed in the MAIN program module of the model. The primary functions of the MAIN program are numerous:

1. to control the model input and output,
2. to initiate and terminate the simulation,
3. to allocate and appropriately initialize variables and arrays,
4. to retrieve the required boundary-value data and generate the boundary-condition equations,
5. to construct the coefficient matrices and perform the necessary matrix transformations, and
6. to generally supervise the various subprograms and the overall computation process.

The eight primary subroutines of the model perform various functions in support of the simulation conducted by the MAIN program. A matrix solution for the set of branch-transformation equations is effected by Gaussian elimination using maximum pivot strategy in the subroutine GEMXPI. The subroutine SETA computes the flow-resistance coefficient as a linear or quadratic function of water temperature, flow depth, discharge, Froude number, or Reynolds number, as designated by the user according to the prototype flow properties. The ARBIN subroutine interpolates cross-sectional area and top-width properties at a specified stage and approximates the hydraulic radius from the input stage-area-width geometry tables. The subroutine OUT prints computed results in tabular form following each iteration or time step, prints daily summaries of flow results, prints cumulative flow volumes, which it stores and subsequently retrieves from a direct-access file, and optionally punches initial condition cards (for subsequent input to follow-on executions of the model). Line-printer plots of computed results, optionally including plots of measured data, are produced by the PRTPLT subroutine. Similarly, the OPLOT subroutine prepares a daily computed and, optionally, a measured discharge or stage hydrograph for subsequent offline plotting on a Tektronix interactive cathode-ray-tube graphical terminal, a CalComp drum or flatbed electromechanical pen plotter, or an Information International, Inc., FR80 microfilm recorder. Subroutines DADIO and DTCODE, which are part of the time-dependent-data storage-and-retrieval system, perform various functions in response to the model's need for boundary-value data. The DTCODE subroutine verifies the beginning and ending dates and times for requested boundary-value data, and the DADIO subroutine effects the data retrieval from the time-dependent data base.

Program restrictions

In addition to the limitations previously mentioned that result from the mathematical formulation, from the numerical technique, or from the computed data and (or) given boundary-value data-accuracy requirements, certain restrictions are imposed by the computer program itself. Although the aforementioned limitations may necessarily preclude implementation of the model, restrictions resulting from the computer code can, in general, be overcome. These restrictions are primarily a consequence of the dimensionality of arrays as currently established in the computer program. Appendix III provides a table in which are listed those arrays that may need to be expanded to accommodate networks with unique dimensional requirements. The version of the program as listed in Appendix IV accommodates networks composed of as many as 15 branches, 15 junctions, and 5 external-boundary conditions. At most 60 sets of cross-sectional data can be input to this version. Any given branch can be subdivided into two, three, or more segments as long as the total number of sets of cross-sectional data is not exceeded. (This is a benefit derived from the array addressing technique used in the model.) For each required cross section a maximum of 20 values may be input to the program to define the stage-area-width relationship representing the geometry at that location. As many as five sets of measured time-series data can be input to the current version of the program for plotting purposes. The maximum number of measured data, per set, stored in memory for plotting purposes is 288, thus providing sufficient capacity to retain one day of continuous data recorded at 5-minute intervals. The maximum number of daily computed results held in core for plotting purposes is 288, thus accommodating results computed at intervals of five minutes or longer. As described, these limitations are not imposed as rigid constraints to be strictly adhered to; they merely represent limitations for this particular version of the model as programed. The model is programed in a manner which makes it easily possible, using the information provided in Appendix III, to change the code to accommodate specific, unique, model-application requirements. Certain other restrictions, particularly as pertain to the input of program control cards described in Appendix I, will be identified subsequently. Other obvious limitations resulting from the input format for program variables are apparent from Appendix I.

Model application

To apply the branch-network flow model to a network, it is first necessary to schematize the channel geometry in accord with the prototype conditions, model requirements, and model output expectations. Using available topographic quadrangle maps or maps of comparable quality, one must first visualize an appropriate network schematization. In the schematization process it is necessary to delineate the branches and segments so as to account for variations in the channel cross-sectional geometry, roughness, and velocity distribution, as well as tributary flows, nodal flows, and other hydraulic factors peculiar to the particular prototype under investigation. Channels with linearly varying cross-sectional properties should be schematized in a way that accounts for changes in the rate of cross-sectional variation and bottom profile changes, as well as other channel properties. Abrupt, substantial changes in the hydraulic and geometric properties of the various channels composing the prototype must be accounted for, either by subdividing such channels into multiple segments or by treating them as two or more branches. Channels of appreciable length should also be subdivided into multiple segments or branches for computational considerations. Experience with one-dimensional flow simulations indicates that branch lengths of a few kilometers up to a maximum of about 25 km may be used; this guideline appears to be appropriate for the branch-network flow model as well. However, the most ideal branch lengths are approximately 8–17 km.

Once a visual conception of the network schematization has been formulated, it is necessary to delineate the branches and identify the junction locations. The branch and segment lengths can be determined by direct field-survey techniques or by measuring along the channel thalweg as depicted on topographic maps, marine charts, or aerial-survey photographs. The chosen network schematization is conveyed to the model program by assigning numbers to the branches and junctions composing the system. The branch-network flow model is formulated in a manner that imposes only one constraint upon the numbering scheme employed. Specifically, there are no restrictions such as downstream numbering that must be adhered to in the branch and junction identification process. The only limitation on the numbering system adopted is that for a network composed of N branches and K junctions, the branches must be independently assigned successive integer values from 1 to N and junctions must be independently defined by successive integer values from 1 to K. (Branch numbers are assigned according to the order of input of the branches to the model program.) For each branch a positive flow direction must be identified (or arbitrarily selected) in order to preserve the correct algebraic signs of the computed unknowns. This is accomplished by assigning the appropriate junction numbers in the flow-direction arrays. In order to further identify the branches of the network, a name can be assigned to each branch.

After the branches to be treated in the network have been identified and all significant tributaries have been considered, cross-sectional geometry must be defined. Cross-sectional geometry, in the form of stage-area-width tables, must be supplied to the model at the beginning and end of all branches. If simulation results are needed at intermediate locations within a branch or if variations in the physical and hydraulic properties of the branch require more accurate schematization, one or more additional cross-sectional geometry tables can be input, thereby subdividing the branch into two or more segments. Each set of cross-sectional data within a branch describes the local channel geometry. Simulation output is available at all locations for which cross sections are defined; thus, as a minimum, computed results are produced at the termini of all branches within the network. The number of cross-sectional geometry tables needed to define the geometry of each branch must be designated. These tables must be input to the model in downstream order.

Cross-sectional data can be obtained by direct soundings or from existing navigation charts. After the geometry is determined, it can be manually prepared for input to the model or processed through a special computer program specifically designed to aid in the analysis and preparation of cross-sectional geometry for input to one-dimensional mathematical-numerical models. This program computes area, width, wetted perimeter, and hydraulic radius of channel cross sections at successive increments of stage from survey data consisting of point measurements of the channel-bottom elevation referenced by cross-channel stationing. After the field-recorded data are digitally encoded, compiled, and punched on computer cards or otherwise directly entered into a computer file, they are input to the cross-sectional geometry program. The output of the program consists of stage-area and stage-width tables prepared in punched-card format, as shown in table 1, which is compatible for direct inclusion in the branch-network flow model.

The required, tabular, cross-sectional data can be input to the model directly via cards or via accessing files of card-image data previously stored on an intermediate magnetic disk by the cross-sectional geometry program. Disk input of cross-sectional geometry is particularly advantageous once the schematization is verified and not subject to frequent alteration.

Other physical and hydraulic properties such as the momentum coefficient, the water temperature, and the wind speed and direction may also need to be evaluated and supplied to the model via initial-condition cards.

Implementation of the branch-network flow

TABLE 1.—Stage-area-width relationships for the Sacramento River at Sacramento, Calif.

Stage (ft)	Area (ft^2)	Width (ft)
-10.00	2,541	411
-8.00	3,364	468
-6.00	4,345	506
-4.00	5,567	528
-2.00	6,640	544
0.0	7,743	566
2.00	8,892	582
4.00	10,069	594
6.00	11,268	605
8.00	12,486	615
10.00	13,728	625
12.00	14,988	635
14.00	16,271	647
16.00	17,583	667
18.00	18,929	680
20.00	20,314	703
22.00	21,731	713
24.00	23,175	732
26.00	24,650	743
28.00	26,148	754

model also requires the preparation and input of boundary-value data if boundary conditions are not described by unique stage-discharge relationships. Boundary conditions must be defined at all external junction locations, that is, junctions having a singular connecting branch, such as 1, 2, and 4 of figure 4. As discussed in the section Initial and boundary-value data, boundary-value data can be specified in the form of stage or discharge hydrographs. The null discharge at a dead-end branch can also be used as a boundary condition. When necessary parabolic interpolation is employed with the given time series of input boundary-value data to produce values consistent with the time step being used in the simulation. A limited amount of boundary-value data—as many as 720 values per location in the current version of the model—can be manually punched on cards for input to the program. Operational use of the model, wherein larger sequences of boundary-value data are required, is most efficiently and conveniently accomplished using data stored and retrieved through the time-dependent data-processing system. The operational advantages of automatic data storage and retrieval through use of this system are obvious. Mixed modes of specifying boundary conditions can also be employed. One or more required boundary conditions can be defined by data retrieved from the data base of time-dependent data, whereas others are described by card input data or by equations. Although no specific order of input of boundary-value data-definition cards is required, if any boundary conditions are to be defined by data retrieved from the data base of time-dependent data, the first boundary-value data-definition card input must be one requesting data from disk, preferably the one specifying the boundary-value data recorded at the highest frequency. The remaining boundary-value data-definition cards can be input in any sequence. Similar considerations apply to the input of measured data.

Boundary-value data and cross-sectional geometry data input to the model must, of course, be referenced to a common datum. Other flow information in the form of inflows (or outflows) at internal junctions—also identified as nodal flows—and initial values of unknowns at all cross sections must be furnished to the model in card image form.

Having determined a computational time step, iteration and convergence criteria, and selected weighting factors for the spatial derivatives and geometric properties, one can begin model execution after the required input and job-control cards are prepared.

Program run preparation

Data input

The first step in preparing the program for execution on the computer is to punch the required input control cards according to the format given in Appendix I. A schematic diagram showing the order of input cards for a model setup is given in figure 15. Sample execution decks are illustrated subsequently (figs. 18, 35). As these figures suggest, orderly input of control cards is required. The basic sequence of data input is unchanged regardless of the particular model application.

Figure 15 shows that all branch-identity cards, initial-condition cards, and cross-sectional-geometry cards are input immediately

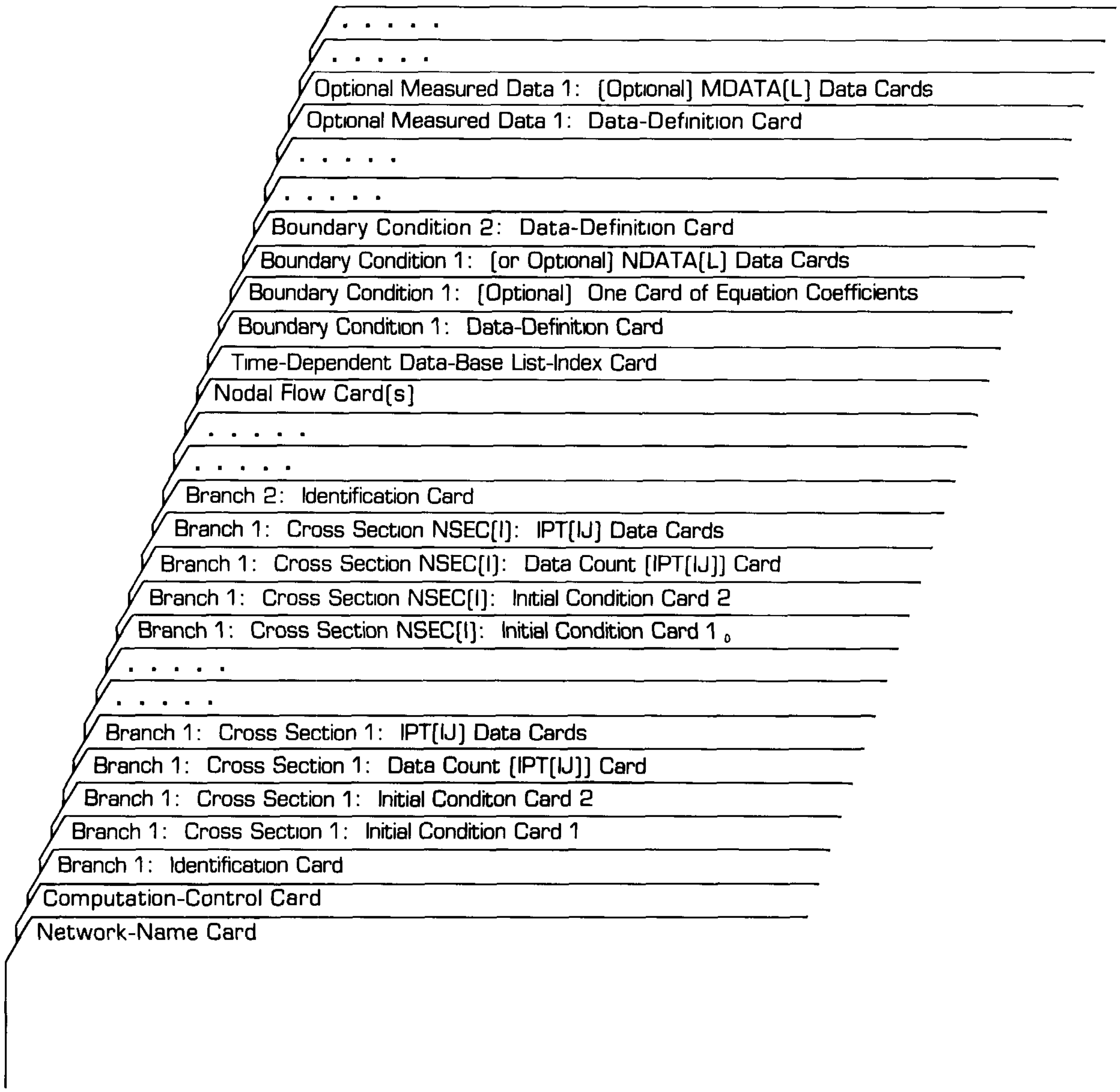


FIGURE 15. – Input-card order for the branch-network flow model.

following the network-name and computation-control cards and immediately preceding the nodal-flow card(s). The branch-identity cards, describing the branches of the network, must be input in sequences according to their designated branch numbers. This is necessary in order to retain the branch number assignments for output identification purposes. Cross-sectional-geometry cards, used to describe the channel geometry, must be input in their downstream (positive flow) order of appearance within each respective branch. (The cross section located at the flow source for the particular branch is input first; the one at the outlet, last.) All sets of cross-sectional-geometry cards defining the geometry of a branch are input following the identity card of that particular branch. Each set of cross-sectional-geometry cards is immediately preceded by two initial-condition cards that define initial values at the cross section. If cross-sectional geometry is input from a disk file, initial-condition cards are contiguous. The stage-area-width relationships defining the cross-sectional geometry are coded on individual cards that must be input in sequence beginning with the lowest stage value for the particular cross section. Each set of cards defining the relationships at a particular cross section is preceded by a card that identifies the number of such relationships (cards) input.

Nodal flows, defined via nodal-flow cards, must be coded on the required card(s) in sequence according to their assigned junction numbers. One nodal-flow card is sufficient to define the inflow (or outflow) at each of ten junctions. Constant nodal flow with respect to time is assumed and is maintained throughout the extent of the simulation. Sufficient nodal-flow cards must be input to account for all junctions within the particular network schematization, even though the nodal flow at all or some junctions is negligible. At a minimum one nodal-flow card is required.

Following the nodal-flow card(s), the list-index card, which signals the production of time-dependent, data-base, summary information, must be input. A blank card must be inserted in place of the list-index card when running the model using boundary-value data input exclusively via cards. Following the list-index card, boundary-value data-definition cards are input. When a boundary condition is defined by a stage-discharge equation or when boundary-value data are supplied via cards, one card – in addition to the required boundary-value, data-definition card – is input to define the equation coefficients (up to and including a cubic equation) or multiple cards are input to provide the data (one boundary-value datum per card). In either case, the card(s) must immediately follow the respective data-definition card. After all boundary-value, data-definition cards and appropriate coefficient or data cards (if any) are input, optional measured-data-definition cards follow. (Aside from boundary-value data, measured time-series data are required only for optional plotting of model-prototype comparisons.) Similarly when measured data are supplied via cards, they are input one value per card immediately following the respective data-definition card. All boundary-value and measured data must be input in proper time sequence, with no intervening gaps in time permitted.

Whereas boundary-condition locations are always identified by their junction number, measured-data locations can be specified either by junction or branch cross-sectional numbers. For a measured-data location specified by a junction number, computed flow results are always derived from the first branch connecting at that junction. For example (refer to fig. 4), measured data identified by junction number 3 would be associated with flow results computed at branch cross-sectional number I–2, as opposed to branch cross-sections II–2 or III–1. Therefore, care must be exercised in identifying measured-data locations by junction number, particularly at junctions joining more than two branches or junctions having nodal inflows or outflows. No such ambiguities exist for measured-data locations identified by branch cross-sectional numbers.

Job control

In order to facilitate use of the branch-network flow model from remote terminals connected to the Amdahl computer system operated by the U.S. Geological Survey, the model program has been compiled and loaded in an online library. To simplify the job-control requirements, a cataloged procedure, called BRANCH, which is composed of job-control statements, has been written and is available to execute the program from this library. The next step in preparing the model for execution is to set up the job-control cards necessary to invoke the cataloged procedure. Subsequent illustrations (figs. 18, 35) show the job-control statements required for two sample executions of the program using this cataloged procedure. Table 2 lists and defines the symbolic parameters in the cataloged procedure, BRANCH, which is available through the private procedure library SCHAF.PROCLIB. Appropriate symbolic parameter assignments can be determined by referring to table 2.

TABLE 2.—Symbolic parameters of the BRANCH cataloged procedure

Symbolic Parameter	Default	Description
PROG	BRANCH	Version of the branch-network-model program to be executed (BRANCH: complete version; BRANOP: version without OPLOT routine; BRANOD: version without OPLOT and DADIO routines).
ECORE	550K	Region size (K bytes) required to execute the program (BRANCH: 550K, BRANOP: 360K; BRANOD: 250K).
ETIME	1	Execution time for the program, where time is specified in minutes.
GIUNIT	3330	Unit type of the device containing the data-station reference file for the data base of time-dependent, boundary-value data.
GIVOL	(None)	Volume serial number of the device containing the data-station reference file for the data base of time-dependent, boundary-value data.
GINAME	NULLFILE	Data set name of the data-station reference file for the data base of time-dependent, boundary-value data.
DAUNIT	3330	Unit type of the device containing the data base of time-dependent, boundary-value data.
DAVOL	(None)	Volume serial number of the device containing the data base of time-dependent, boundary-value data.
DANAME	(None)	Data set name of the data base of time-dependent, boundary-value data.
XSUNIT	3330	Unit type of the device containing the cross-sectional geometry data file.
XSVOL	(None)	Volume serial number of the device containing the cross-sectional geometry data file.
XSNAME	NULLFILE	Data set name of the cross-sectional geometry data file.
FVUNIT	3330	Unit type of the device containing the cumulative flow-volume output file.
FVVOL	(None)	Volume serial number of the device containing the cumulative flow-volume output file.
FVNAME	NULLFILE	Data set name of the cumulative flow-volume output file.
FVDISP	OLD	The current disposition of the cumulative flow-volume output file; code NEW if a new flow-volume file is to be created.
TTUNIT	3330	Unit type of the device to contain the DISSPLA compressed file for Tektronix plotting.
TTVOL	(None)	Volume serial number of the device to contain the DISSPLA compressed file for Tektronix plotting.
TTNAME	NULLFILE	Data set name of the DISSPLA compressed file.
TTDISP	OLD	The current disposition of the DISSPLA compressed file; code NEW if a new file is to be created.
CCVOL	(None)	Volume serial number of the standard-labeled, magnetic tape to contain the CalComp plot file.
CCNAME	NULLFILE	Data set name of the CalComp plot file.
FRVOL	(None)	Volume serial number of the standard-labeled, magnetic tape to contain the FR80 plot file.
FRNAME	NULLFILE	Data set name of the FR80 plot file.

There are seven primary data files identified in table 2. The data-station reference and time-dependent data files[1] (identified by the symbolic parameter prefixes GI and DA, respectively) are necessary if boundary-value data are stored on a direct-access disk by the time-dependent data processing system for subsequent inclusion in the model. The cross-sectional geometry file[1] (identified by the symbolic parameter prefix XS and generated via the cross-sectional geometry program) contains stage-area-width tables for input to the model. The remaining four files identified in table 2 are intended for model output purposes. The flow-volume output file[1] (identified by the symbolic parameter prefix FV) retains cumulative flow volumes for the purpose of printing monthly summaries. The DISSPLA compressed direct-access-type file[1] (identified by the symbolic parameter prefix TT) is generated if subsequent plotting is to be accomplished on a Tektronix cathode-ray-tube graphical terminal. Likewise, the CalComp and FR80 magnetic-tape plot files[1] (identified by the symbolic parameter prefixes CC and FR, respectively) are generated if plots are to be produced using a CalComp drum or flatbed, electromechanical

[1]Details on the file structure and allocation techniques can be obtained from the authors

pen plotter or an Information International, Inc., FR80 microfilm recorder.

Program versions

The branch-network flow model is programed in modular fashion. This modular structure facilitates the exclusion of computer code at the expense of reduced model capabilities. If, for instance, the computer generation of plots via graphical terminals, electromechanical pen plotters, or microfilm recorders is not desired, the software instructions to create such plot files can be negated from the model source code of Appendix IV by replacing the OPLOT subroutine with the FORTRAN IV code listed in figure 16. (Computer generated plotting designed specifically for the line printer is unaffected by this exclusion.) Furthermore, if boundary conditions are always to be defined via equations or card-input data, the time-dependent-data storage-and-retrieval subroutine, DADIO, which provides the capability to retrieve boundary-value data automatically from a data base during the simulation, can be overridden by inserting the FORTRAN IV code listed in figure 17. In addition to the full model version, as listed in Appendix IV, two such reduced versions of the branch-network flow model have been compiled and loaded in the cataloged programs library for access via the BRANCH cataloged procedure, the symbolic parameters of which are defined in table 2. The complete version of the branch-network flow model, assigned the program name BRANCH in the library, requires 550K bytes of storage for execution. The version of the model with reduced plot capabilities, assigned the program name BRANOP, requires 360K bytes of storage. The model version with reduced plot options and no time-dependent data base link, called BRANOD, requires 250K bytes. To invoke the model version with the required capabilities one need only make the appropriate assignment of the PROG and ECORE symbolic parameters of the BRANCH cataloged procedure.

Storage and time requirements

Execution storage requirements for the three versions of the branch-network flow model are, as given in the section Program versions, 550K bytes for BRANCH, 360K bytes for BRANOP, and 250K bytes for BRANOD. The BRANCH program version listed in Appendix IV requires 400K bytes of storage for compilation when using the IBM (International Business Machines) FORTRAN IV H-level compiler and requesting a cross-reference listing of variables and labels. The time needed for compilation on the IBM 370/155 computer system is approximately 2 minutes, whereas on the Amdahl 470 V/7 computer the CPU time required for compilation is approximately 10 seconds. Execution time requirements vary greatly and are dependent upon the computational-time step, network schematization, convergence criteria, and the type of output desired, as well as other less significant factors. A month-long simulation of a river system comprising five single-segment branches required 6 minutes (approximately 12 seconds per day) of IBM 370/155 CPU time computing at a 15-minute time step and printing daily summaries of flow results. Another single-branch, single-segment model simulation took 8.3 minutes of IBM 370/155 CPU time to compute nine months at a 15-minute time step and produce flow-volume summaries. Yet another simulation of a large estuarine system composed of 25 multiple-segment branches required 28

```
      SUBROUTINE OPLOT(NSEC,XSKT,BRNAME,IJF,IJT,ISTATN,ZQCOMP,IBJNC,       OP   0
     1MBCH,MSEC,MDATA,ZQMEAS,MAXCZQ,MAXS,MAXMZQ,PRINTR)                    OP   0
      INTEGER PRINTR                                                       OP   1
      INTEGER BRNAME(10,1),ISTATN(1)                                       OP   2
      INTEGER *2NSEC(1),XSKT(1),IJF(1),IJT(1),IBJNC(1)                     OP   3
      INTEGER *2MBCH(1),MSEC(1),MDATA(1),MIYR,MIMO,MIDA,MIHR,MIMN          OP   4
      DIMENSION ZQCOMP(MAXCZQ,1), ZQMEAS(MAXMZQ,1)                         OP   5
      RETURN                                                               OP   6
      ENTRY ZQPLOT                                                         OP   7
      RETURN                                                               OP   8
      END                                                                  OP   9
```

FIGURE 16.—FORTRAN IV code used to negate the DISSPLA coded plot subroutine, OPLOT, from the branch-network flow model.

```
      SUBROUTINE DADIO(PRINTR,PUNCH,DSREF,TDDATA,LIST,RTCODE)           DA   0
      INTEGER PRINTR,PUNCH,DSREF,TDDATA                                 DA   1
      INTEGER STANUM                                                    DA   2
      INTEGER *2LIST,RTCODE,STRIP                                       DA   3
      INTEGER *2TYPE,BY,BMO,BD,BH,BMN,EY,EMO,ED,EH,EMN,STAGE(1),RDPDY   DA   4
      LOGICAL PRTMSG                                                    DA   5
      EXTERNAL CREATE                                                   DA   5
      RETURN                                                            DA   6
      ENTRY DATYPE(I,J,K,L,M)                                           DA   7
      RETURN                                                            DA   8
      ENTRY DADI(STANUM,TYPE,BY,BMO,BD,BH,BMN,EY,EMO,ED,EH,EMN,STAGE,RDP DA   9
     1DY,STRIP,PRTMSG,RTCODE)                                           DA  10
      RETURN                                                            DA  11
      ENTRY DADO(STANUM,TYPE,BY,BMO,BD,BH,BMN,EY,EMO,ED,EH,EMN,STAGE,RDP DA  12
     1DY,STRIP,PRTMSG,RTCODE)                                           DA  13
      RETURN                                                            DA  14
      END                                                               DA  15
      SUBROUTINE DTCODE(YR,MO,DA,HR,MN,ITIME,ETIME,*)                   DT   0
      INTEGER ETIME,ITIME                                               DT   1
      INTEGER *2YR,MO,DA,HR,MN,JDAYN                                    DT   2
      INTEGER *2DPERM(12)/31,28,31,30,31,30,31,31,30,31,30,31/          DT   3
      DPERM(2)=28+(4-(YR-YR/4*4))/4                                     DT   4
      IF (YR.LT.O.OR YR.GT.99.OR.MO.LT.1.OR.MO GT 12.OR.DA.LT.1.OR.DA.GT DT   5
     1.DPERM(MO).OR.HR LT.O.OR.HR.GT.24.OR.MN.LT.O OR.MN.GT.59) RETURN 1 DT   6
      JDAYN=DA                                                          DT   7
      N=MO-1                                                            DT   8
      IF (N.EQ.O) GO TO 20                                              DT   9
      DO 10 I=1,N                                                       DT  10
   10 JDAYN=JDAYN+DPERM(I)                                              DT  11
   20 ITIME=YR*10000000+JDAYN*10000+HR*100+MN                           DT  12
      ETIME=(JDAYN-1)*1440+HR*60+MN                                     DT  13
      RETURN                                                            DT  14
      END                                                               DT  15
      SUBROUTINE CREATE(IRETCO,IERNO,ILOGUN)                            CR   1
      COMMON /IHC236/ EMPTY,LOGUN                                       CR   2
      LOGICAL EMPTY                                                     CR   3
      LOGUN=ILOGUN                                                      CR   4
      IRETCO=O                                                          CR   5
      EMPTY= TRUE.                                                      CR   6
      RETURN                                                            CR   7
      END                                                               CR   8
```

FIGURE 17.—FORTRAN IV code used to negate the time-dependent boundary-value-data storage-and-retrieval subroutine, DADIO, from the branch-network flow model.

seconds of Amdahl 470 V/7 CPU time. Flow results (computed at 15-minute intervals) were printed at every time step for all 76 cross sections and computed-versus-measured discharges were plotted on the CalComp plotter for the duration of the simulation (1.1 days). The time required to conduct a given simulation is a function of many variables, thus no simple rule for estimating execution time can be stated. Execution time is assigned to the model simulation by the ETIME symbolic parameter of the BRANCH cataloged procedure.

Diagnostic messages

In addition to standard FORTRAN IV and system-supplied diagnostic messages, error and other pertinent messages are printed by the DISSPLA plot software, by the time-dependent-data storage-and-retrieval system, as well as by the branch-network flow model itself. Diagnostic messages from the DISSPLA plot software and the time-dependent-data storage-and-retrieval system can be suppressed by setting the respective input parameters, IPLMSG and IPRMSG, to zero.

The following diagnostic messages are generated by the MAIN program and the subroutines of the branch-network flow model. Additional comments on the possible reasons for the error are given below the message:

INITIAL STAGE VALUE UNSPECIFIED IN BRANCH (I) SECTION (J)

Initial values of stage and discharge must be supplied at all cross sections.

INITIAL STAGE XXXXX.XX OUT OF DEFINED RANGE OF CHANNEL GEOMETRY FOR BRANCH (I) SECTION (J)

The initial value of stage is out-of-range of the stage-area-width geometry table for the specified cross section.

IMPROPER NUMBER OF CROSS-SECTIONAL DATA (2< =IPT< =20)

More than one and 20 or fewer stage-area-width relationships must be input to define the geometry at each cross section.

DUPLICATE, OR OUT-OF-ORDER, STAGES IN CHANNEL-GEOMETRY TABLE FOR BRANCH (I) SECTION (J)

Unique stage-area-width relationships defining the cross-sectional geometry must be input in sequential order beginning with the lowest stage value.

MATRIX NOT SQUARE

This condition can be caused by improper schematization and (or) input parameter errors.

INVALID BOUNDARY-VALUE DATA PARAMETER(S)

Information on a boundary-value data-definition card is invalid or inconsistent.

INVALID MEASURED DATA PARAMETER(S)

Information on a measured data-definition card is invalid or inconsistent.

MATRIX IS SINGULAR

The matrix has no inverse. This condition can be caused by improper boundary conditions or by schematization errors.

TOO MANY MEASURED DATA LOCATIONS (MXMD=5)

Up to 5 sets of measured data can be input in this version of the model.

JUNCTION (J) OF BOUNDARY-VALUE DATA IMPROPERLY SPECIFIED (0<IBJNC<15)

The junction number must be greater than 0 and less than 15.

IMPROPER NUMBER OF MEASURED DATA SPECIFIED (1< =MDATA< =288)

Up to 288 measured data can be input at each location.

INVALID BRANCH (I) SECTION (J) SPECIFIED FOR MEASURED DATA

The branch and cross-section numbers identifying a measured data location are errant or inconsistent with the network schematization.

JUNCTION (J) OF MEASURED DATA NOT FOUND

The specified junction number of a measured data location is in error.

INITIAL VALUE(S) OUT OF DEFINED RANGE OF CHANNEL GEOMETRY

One or more initial stage values are out-of-range of the respective stage-area-width geometry tables.

INITIAL OR COMPUTED STAGE RESULTS IN ZERO OR NEGATIVE AREA AND (OR) TOP WIDTH

The initial or computed stage value is inconsistent with the stage-area-width geometry table.

COMPUTED STAGE OUT OF DEFINED RANGE OF CHANNEL GEOMETRY

The computed stage is out-of-range of the stage-area-width geometry table.

INVALID COMPUTATION CONTROL PARAMETER(S)

One or more input parameter values on the computation-control card are invalid.

MAXIMUM ITERATIONS EXCEEDED AT (HR:MN) ON (YR/MO/DY) Z-ZP (I,J)=XXX.XXXX Q-QP (I,J)=XXXXXX.X
Convergence conditions were not satisfied during the specified time step. The stage and discharge printed represent the maximum difference between the last successive solutions. Computation continues using the last computed values.

EXECUTION TERMINATED DUE TO INCORRECT ATTEMPT TO UPDATE CUMULATIVE FLOW-VOLUME FILE
The cumulative flow-volume file is allocated to contain flow volumes computed for a given calendar year and a specific network. An attempt to add data from a different network or calendar year produces the above error message.

STEP SIZE IN PLOT SCALE ALGORITHM EXCEEDS MAXIMUM LIMIT
The magnitude of the quantity to be plotted is prohibitively large.

Input/output description

The ultimate success of any simulation effort depends in large measure on the capacity to easily manipulate both model and prototype data in order to analytically interpret model results. This aspect of modeling is not only valuable for purposes of model calibration and verification but is important in providing the facility with which to appraise, compare, and comprehend the significance of various water-management alternative plans. Because an efficient, economical, data storage-and-retrieval technique is an integral element of the branch-network flow model and because model results can be presented in diverse yet easily and quickly comprehensible formats by its numerous graphical display options, it is possible to achieve a broadly applicable, operationally usable, numerical simulation capacity.

The program-control cards and job-control statements required to execute the branch-network flow model can be set up according to the instructions given previously. A sample setup applying the model to the Sacramento-Freeport reach of the Sacramento River is illustrated in figure 18. In this example model setup, all cross-sectional geometry, boundary-value, and measured data are input via computer cards. (Figure 35 illustrates an execution deck setup with data input from disk files.) Some of the boundary-value and measured data in figure 18 are shown coded on the same card, that is, those delimited by four dashes. This is not meant to imply that–contrary to the instructions in the section Data input–more than one boundary-value or measured datum can be input per card, rather this was merely done to conserve space in the illustration. This complete deck setup will permit the user to experiment with the model. A line-printer plot of computed-versus-measured discharge and other associated printout from the simulation is shown in figure 19*A–C*.

Model input and output can be either in the International System of Units (metric) or the inch-pound system of units. The simulation is performed in the units of measure of the input data. However, the computed results can be converted for output upon request by the user.

The first page of printout (fig. 19*A*) from all simulations is a list of the control-card parameters as assigned by card or default. In addition, the boundary-value datum corrections–if any are to be applied–and the stage computation datum are printed (fig. 19*A*). The boundary-value datum corrections are input on boundary-value data-definition cards, whereas the stage computation datum is determined by the model. The stage computation datum is an arbitrary datum taken as the average of the absolute maximum and minimum stages specified on all cross-sectional-geometry tables. The simulation is performed with all stages automatically referenced to this datum for reasons of computational accuracy.

Listed on the second page of printout (fig. 19*B*) are the branch identification parameters, the cross-sectional-geometry tables, and the initial values. This information is always printed for each simulation regardless of the specific output type chosen.

The third and final page of simulation output illustrated in figure 19*C*, is the desired line-

```
//COMPUTER JOB CARD
/*PROCLIB  SCHAF.PROCLIB
//BRANCH EXEC BRANCH,PROG=BRANOD,ECORE=250K
//SYSIN DD *
SACRAMENTO R. : SACRAMENTO-FREEPORT REACH
    1 2 2    ME    33    4                                   .75
 1 2 2SACRAMENTO R   SACRAMENTO-FREEPORT REACH
      2.48    6373.0            57024 00        0.2620E-010 1283E-06- 4167E-12

 3
0.         7460.    550
2.         8602.    580
4.         9780.    595.
      2 37     4107 0

 3
0          8050     595
2          9251.    607
4         10476.    618.


 Z 1 32  11447500 FROM= 76/10/12 08 00  TO= 76/10/12 15.45     96
2 48
NOTE: AS SHOWN BY PREVIOUS CARD, PUNCH DATA OF NEXT TWO CARDS ONE VALUE PER CARD
 ---- 2.55 2.62 2.67 2.74 2 79 2.84 2.88 2.91 2.92 2.92 2.92 2.90 2 88 2.85 2.83
 2.80 2 77 2.73 2.70 2.67 2 63 2.60 2.56 2.52 2 48 2.44 2.40 2.37 2.34 2.31 2.29
 Z 2 32  11447650 FROM= 76/10/12 08:00  TO= 76/10/12 15:45     96
2 37
NOTE: AS SHOWN BY PREVIOUS CARD, PUNCH DATA OF NEXT TWO CARDS ONE VALUE PER CARD
 ---- 2.43 2 49 2.55 2 60 2.64 2 66 2.65 2.63 2.60 2 57 2.53 2 50 2.47 2.43 2 39
 2.35 2.30 2 26 2.21 2 16 2.11 2.06 2.02 1.93 1.94 1.90 1.88 1 86 1.85 1 88 1.91
 Q 1 30          FROM= 76/10/12 08 00  TO= 76/10/12 15·15     96
6360
NOTE· AS SHOWN BY PREVIOUS CARD, PUNCH DATA OF NEXT TWO CARDS ONE VALUE PER CARD
 ---- 6180 6090 6090 6120 6140 6140 6130 6140 6340 6860 7160 7340 7470 7580 7700
 7810 7940 8060 8180 8340 8530 8640 8680 8690 8690 8720 8780 8780 8690
/*
//
```

FIGURE 18.—Sample deck setup to execute the branch-network flow model of the Sacramento-Freeport reach of the Sacramento River. (See Appendix I for proper card input format of boundary-value and measured data.)

printer plot (IOTOPT=3 and IPLDEV=0) of computed-versus-measured discharge (IPLOPT=3) for the Sacramento River at Sacramento, Calif. Other options permit plotting computed-versus-measured stage (IPLOPT=4), plotting computed stage hydrographs (IPLOPT=2), or plotting computed discharge hydrographs (IPLOPT=1). These options are also available for plotting on a line printer (IPLDEV=0), a Tektronix cathode-ray-tube graphical terminal (IPLDEV=1), a CalComp drum or flatbed electromechanical pen plotter (IPLDEV=2), or an Information International, Inc., FR80 microfilm recorder (IPLDEV=3).

In addition to the generation of plots (IOTOPT=3), the model will print flow results in tabular form at every time step (IOTOPT=0), print flow results in similar tabular form at every iteration (IOTOPT=1), print only daily summaries of flow results (IOTOPT=2), or print monthly flow-volume summaries (IOTOPT=4).

The various output formats available with the branch-network flow model are illustrated in figures presented throughout this report. Table 3 is a composite list of these figures. Appropriate values of the program control variables used to produce the output illustrated in these figures are also given.

The output derived by IOTOPT option 0 and illustrated in figure 20 is the computed flow results printed in tabular form at each computa-

```
UNSTEADY FLOW COMPUTATION IN A NETWORK OF OPEN CHANNELS

          BRANCH-NETWORK MODEL (VERSION 79/04/19)
               A FOUR-POINT IMPLICIT SCHEME

                  LINEAR MATRIX SOLUTION
   BY GAUSS ELIMINATION USING MAXIMUM PIVOT STRATEGY

===========================================================

        UNITS OF INPUT (EN/ME)          =       EN
        UNITS OF OUTPUT (EN/ME)         =       ME
        BRANCHES (1<=N<=15)             =        1
        JUNCTIONS (2<=N<=15)            =        2
        BOUNDARY VALUES (1<=N<= 5)      =        2
        GEOMETRY INPUT UNIT (5/10)      =        5
        PRINTOUT OPTION (0<=N<=4)       =        3
        PLOT OPTION (0<=N<=4)           =        3
        PLOTTER DEVICE (0<=N<=3)        =        0
        PRINT MESSAGE OPTION (0/1)      =        0
        PLOT MESSAGE OPTION (0/1)       =        0
        EXTRAPOLATION OPTION (0/1)      =        0
        PUNCH INITIAL COND  (0/1)       =        0
        FRICTION TYPE (1<=N<=6)         =        4
        MAXIMUM ITERATIONS              =        5
        NUMBER OF STEPS                 =       32
        DERIVATIVE FACTOR (0<=N<=1)     =     1 00
        GEOMETRY FACTOR (0<=N<=1)       =     0 75
        TIME INCREMENT(MINUTES)         =       15
        DISCHARGE CONVERGENCE           =     20 5
        STAGE CONVERGENCE               =   0 0100
        WIND SPEED(MPH/KPH)             =      0 0
        SURFACE DRAG COEFFICIENT        =   0 0026
        WATER DENSITY                   =   1 9617
        STAGE COMPUTATION DATUM         =    2 000
        BVD( 1) DATUM CORRECTION        =    0 0
        BVD( 2) DATUM CORRECTION        =    0 0

===========================================================
```

A

```
CHANNEL GEOMETRY FOR SACRAMENTO R     SACRAMENTO-FREEPORT REACH

BRANCH  1 FROM JUNCTION  1 TO  2   SACRAMENTO R   SACRAMENTO-FREEPORT REACH

 CROSS SECTION  1          STAGE             AREA                WIDTH
                            (FT)            (FT**2)               (FT)
                            0 0             7460 0               550 0
                            2 00            8602 0               580 0
                            4 00            9780 0               595 0

 INITIAL VALUES   STAGE=  2 48  DISCHARGE=   6373 0         BETA= 1 000
 ----------------------------------------------------------------------
 LENGTH= 57024 0 FT;          TEMP= 59 0 DEG F,        WIND=   0 0 DEG

 ETA= 0 262000E-01 + ( 0 128300E-06)*Q + (-0 416700E-12)*Q**2
 ----------------------------------------------------------------------

 CROSS SECTION  2          STAGE             AREA                WIDTH
                            (FT)            (FT**2)               (FT)
                            0 0             8050 0               595 0
                            2 00            9251 0               607 0
                            4 00           10476 0               618 0

 INITIAL VALUES   STAGE=  2 37  DISCHARGE=   4107 0         BETA= 1.000
```

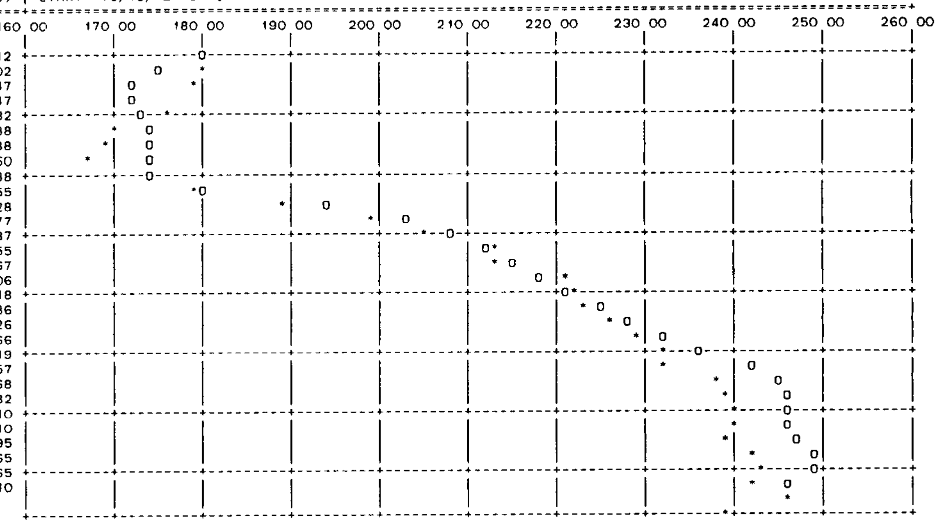
B

SACRAMENTO R SACRAMENTO-FREEPORT REACH — FLOW COMPUTED BY THE BRANCH-NETWORK MODEL

\# \# \# \#

SACRAMENTO R SACRAMENTO-FREEPORT REACH — START= 76/10/12 8 0 — END= 76/10/12 15 45 — (O) MEASURED (*) COMPUTED AT BRANCH SECTION 1 1

TIME HR MN	DISCHARGE (M**3/S) (*)	DISCHARGE (M**3/S) (O)
8 0	180 48	180 12
8 15	180 28	175 02
8 30	179 34	172 47
8 45	172 16	172 47
9 0	175 89	173 32
9 15	169 69	173 88
9 30	169 06	173 88
9·45	167 31	173 60
10 0	174 08	173 88
10 15	178 99	179 55
10 30	188 66	194 28
10 45	198 62	202 77
11 0	205 38	207 87
11 15	212 58	211 55
11 30	212 55	214 67
11 45	220 58	218 06
12 0	221 97	221 18
12 15	223 42	224 86
12 30	225 70	228 26
12 45	228 71	231 66
13 0	232 13	236 19
13 15	232 00	241 57
13 30	237 84	244 68
13 45	239 37	245 82
14 0	239 81	246 10
14 15	239 70	246 10
14 30	239 24	246 95
14 45	242 47	248 65
15 0	242 81	248 65
15 15	242 18	246 10
15 30	245 60	
15 45	239 22	



C

FIGURE 19. — Sample output from the simulation-deck setup of figure 18, with IOTOPT option 3 and IPLOPT option 3. *A*, List of control-card parameters assigned by card or default. *B*, Branch-identification parameters, cross-sectional geometry tables, and initial values. *C*, Desired line-printer plot of computed versus measured discharge for Sacramento River at Sacramento, Calif.

FLOW RESULTS FOR SACRAMENTO R SACRAMENTO-FREEPORT REACH

DATE YR/MO/DY	TIME HR MN	STAGE (M)	VELOCITY (M/SEC)	DISCHARGE (M**3/S)	AREA (M**2)	WIDTH (M)	FALL (M)	BRANCH SECTION (ITERATIONS)	ETA	STAGE (M)	VELOCITY (M/SEC)	DISCHARGE (M**3/S)	AREA (M**2)	WIDTH (M)
76/10/12	0 15	0 99	0 30	256 6	867 5	179 6	0 17	1 (0) 1 2	0 0275	0 82	0 35	315 8	898 1	186 2
	0 30	0 98	0 30	256 5	865 1	179 5	0 18	1 (3) 1 2	0 0275	0 79	0 36	321 1	893 8	186 0
	0 45	0 96	0 30	257 4	861 8	179 4	0 18	1 (3) 1 2	0 0275	0 78	0 36	322 0	890 4	185 9
76/10/12	1 0	0 94	0 30	259 2	859 0	179 3	0 19	1 (3) 1 2	0 0275	0 75	0 37	323 7	886 4	185 8
	1 15	0 93	0 30	257 6	855 8	179 1	0 19	1 (3) 1 2	0 0275	0 73	0 37	327 5	882 5	185 7
	1 30	0 91	0 31	264 2	853 0	179 0	0 20	1 (3) 1 2	0 0275	0 71	0 37	323 3	879 0	185 6
	1 45	0 89	0 31	259 5	849 7	178 9	0 20	1 (3) 1 2	0 0275	0 69	0 38	329 2	875 1	185 5
76/10/1?	2 0	0 87	0 31	261 9	846 5	178 8	0 20	1 (3) 1 2	0 0275	0 68	0 37	326 3	871 7	185 4
	2 15	0 86	0 31	262 9	843 7	178 6	0 20	1 (3) 1 2	0 0275	0 65	0 38	327 2	867 7	185 3
	2 30	0 84	0 31	264 6	841 0	178 5	0 21	1 (3) 1 2	0 0275	0 63	0 38	328 8	863 7	185 1
	2 45	0 82	0 32	264 8	837 7	178 4	0 21	1 (3) 1 2	0 0275	0 61	0 38	329 0	860 3	185 0
76/10/12	3 0	0 81	0 32	264 1	834 4	178 3	0 21	1 (3) 1 2	0 0275	0 60	0 38	328 3	856 9	184 9
	3 15	0 79	0 32	266 8	831 7	178 1	0 21	1 (3) 1 2	0 0275	0 58	0 38	325 6	853 6	184 8
	3 30	0 77	0 32	263 2	828 4	178 0	0 21	1 (3) 1 2	0 0275	0 56	0 38	327 3	850 2	184 7
	3 45	0 75	0 32	261 9	825 1	177 9	0 21	1 (2) 1 2	0 0275	0 54	0 38	325 9	846 9	184 6
76/10/12	4 0	0 74	0 32	264 4	821 8	177 7	0 21	1 (2) 1 2	0 0275	0 53	0 38	317 8	844 6	184 5
	4 15	0 72	0 32	262 2	819 1	177 6	0 21	1 (2) 1 2	0 0275	0 51	0 37	315 5	841 8	184 4
	4 30	0 71	0 32	264 4	816 4	177 5	0 20	1 (2) 1 2	0 0275	0 50	0 37	307 0	840 2	184 4
	4 45	0 69	0 32	264 2	813 6	177 4	0 19	1 (3) 1 2	0 0274	0 50	0 35	296 1	839 6	184 4
76/10/12	5 0	0 68	0 32	259 1	810 9	177 3	0 17	1 (3) 1 2	0 0274	0 50	0 34	285 7	839 6	184 4
	5 15	0 67	0 32	261 2	809 2	177 2	0 16	1 (3) 1 2	0 0274	0 51	0 32	266 5	840 7	184 4
	5 30	0 66	0 32	257 3	807 6	177 1	0 13	1 (3) 1 2	0 0273	0 52	0 29	246 7	843 5	184 5
	5 45	0 65	0 31	248 5	806 5	177 1	0 11	1 (3) 1 2	0 0272	0 54	0 27	227 2	846 9	184 6
76/10/12	6 0	0 65	0 30	239 7	806 5	177 1	0 09	1 (3) 1 2	0 0272	0 56	0 24	207 7	850 2	184 7
	6 15	0 65	0 28	228 7	807 1	177 1	0 08	1 (3) 1 2	0 0271	0 58	0 22	191 4	853 6	184 8
	6 30	0 66	0 27	222 4	808 7	177 2	0 07	1 (3) 1 2	0 0271	0 60	0 20	174 4	856 9	184 9
	6 45	0 68	0 27	217 2	810 9	177 3	0 06	1 (3) 1 2	0 0270	0 62	0 18	158 6	860 8	185 1
76/10/12	7 0	0 69	0 26	209 1	813 6	177 4	0 05	1 (3) 1 2	0 0270	0 64	0.17	150 4	864 3	185 2
	7 15	0 71	0 25	204 2	816 4	177 5	0 05	1 (3) 1 2	0 0270	0 66	0 16	140 1	868 2	185 3
	7 30	0 72	0 24	198 6	819 1	177 6	0 04	1 (3) 1 2	0 0269	0 68	0 15	129 1	872 8	185 4
	7 45	0 74	0 23	191 6	822 4	177 8	0 04	1 (3) 1 2	0 0269	0 70	0 14	122.1	876 8	185 5
76/10/12	8 0	0 76	0 22	183 7	825 7	177 9	0 04	1 (3) 1 2	0 0269	0 72	0 14	119 5	880.2	185.6
	8 15	0 78	0 22	185 5	829 5	178 1	0 04	1 (3) 1 2	0 0269	0 74	0 13	110.5	884 2	185 7
	8 30	0 80	0 22	179 6	833 3	178 2	0 04	1 (2) 1 2	0 0269	0 76	0 13	115.4	887 0	185 8
	8 45	0 82	0 21	175 6	836 1	178 3	0 04	1 (2) 1 2	0 0269	0 78	0 13	116 6	890 4	185 9
76/10/12	9 0	0 84	0 21	179 7	839 9	178 5	0 05	1 (2) 1 2	0 0269	0 79	0 13	115 3	893 3	186 0
	9 15	0 85	0 21	173 8	842 6	178 6	0 05	1 (2) 1 2	0 0269	0 80	0 14	125 5	895.5	186 1
	9 30	0 87	0 21	173 3	845 4	178 7	0 06	1 (2) 1 2	0 0269	0 81	0 15	135 7	896 7	186 1
	9 45	0 88	0 20	173 3	847 6	178 8	0 07	1 (3) 1 2	0 0269	0 81	0 17	151 8	896 7	186 1
76/10/12	10 0	0 89	0 21	174 9	849 2	178 9	0 09	1 (3) 1 2	0 0270	0 80	0 20	174 9	895 0	186 1
	10 15	0 89	0 22	184 3	849 7	178 9	0 10	1 (3) 1 2	0 0270	0 79	0 21	189 7	893 8	186 0
	10 30	0 89	0 22	188 8	849 7	178 9	0 11	1 (3) 1 2	0 0271	0 78	0 24	210 3	891 6	186 0
	10 45	0 89	0 24	203 4	849 7	178 9	0 12	1 (3) 1 2	0 0271	0 77	0 25	219 5	889 9	185 9
76/10/12	11 0	0 89	0 24	205 0	848 6	178 9	0 12	1 (3) 1 2	0 0272	0 76	0 27	237 2	887 6	185 8
	11 15	0 88	0 25	215 2	847 6	178 8	0 13	1 (3) 1 2	0 0272	0 75	0 27	242 0	885 9	185 8
	11 30	0 87	0 25	215 2	845 9	178 7	0 13	1 (3) 1 2	0 0272	0 74	0 29	252 8	883 6	185 7
	11 45	0 86	0 26	223 3	844 8	178 7	0 14	1 (3) 1 2	0 0273	0 73	0 29	255 5	881 3	185 7
76/10/12	12 0	0 85	0 27	224 7	843 2	178 6	0 14	1 (3) 1 2	0 0273	0 71	0 30	262 2	879 0	185 6

	DISCHARGE (upstream)			DISCHARGE (downstream)	
MINIMUM	MEAN	MAXIMUM	MINIMUM	MEAN	MAXIMUM
173 3	228 6	266 8	110 5	238 6	329 2

FIGURE 20.—Computed flow results at each time step printed in tabular form, produced using IOTOPT option 0.

TABLE 3.—Composite list of output examples and appropriate values of the output-control variables

Output option-control variables			Number of figure illustrating output
IOTOPT	IPLOPT	IPLDEV	
0	-----	-----	20.
1	-----	-----	([1])
2	-----	-----	29.
3	3	0	19C.
3	1	[2]1, 2, or 3	27, 32.
3	2	[2]1, 2, or 3	([3])
3	3	[2]1, 2, or 3	25.
3	4	[2]1, 2, or 3	31, 36.
4	-----	-----	21.

[1] Not illustrated, but similar in format to figure 20

[2] For this output option, printed output as produced for IOTOPT=0 is also generated This output option is not available in the BRANOP model-program version

[3] Not illustrated, but similar in format to figure 32

tion time interval, which is 15 minutes in this simulation for the Sacramento River. Each line of printout gives the computed flow at the upstream and downstream ends of the segment at each successive time interval. The flow results printed on the left side of the page are for the upstream cross section, which in this simulation, is located at the gage site in the city of Sacramento whereas those printed on the right side are for the downstream cross section located near Freeport, Calif. Also identified on the printout is the fall computed through the segment and the η value used to compute the flow. The number enclosed in parentheses in the center of the listing is the number of iterations required during the time step to satisfy the convergence criteria. The number of iterations at the initial time step is always zero since these quantities are specified as initial values. Flow

OCTOBER 1977 DISCHARGE AT STATION 11447650

DAY	VOLUME(S) IN MILLIONS OF CUBIC FEET				
1	209 (0700)	-10 (0945)	186 (1715)	-46 (2115)	71
2	240 (0800)	-9 (1030)	168 (1745)	-45 (2145)	51
3	271 (0845)	-12 (1130)	146 (1830)	-41 (2230)	28
4	289 (0930)	-19 (1245)	141 (1930)	-41 (2315)	7
5	319 (1030)	-9 (1330)	170 (2100)	-23 (2400)	0
6	316 (1130)	-18 (1445)	170 (2215)	-16	
7	-13 (0130)	286 (1200)	-38 (1600)	168 (2315)	-4
8	-33 (0300)	258 (1230)	-42 (1645)	185	
9	0 (0015)	-41 (0400)	244 (1300)	-47 (1715)	193
10	9 (0100)	-41 (0445)	237 (1330)	-49 (1745)	191
11	32 (0200)	-39 (0545)	226 (1400)	-49 (1815)	178
12	62 (0245)	-40 (0645)	206 (1430)	-59 (1845)	162
13	94 (0345)	-41 (0730)	191 (1445)	-68 (1915)	140
14	127 (0430)	-38 (0815)	176 (1530)	-77 (2000)	119
15	160 (0530)	-35 (0915)	162 (1600)	-79 (2045)	93
16	195 (0645)	-29 (1015)	154 (1645)	-81 (2130)	64
17	240 (0745)	-31 (1115)	145 (1730)	-81 (2230)	35
18	267 (0830)	-38 (1230)	148 (1845)	-68 (2315)	6
19	291 (0930)	-44 (1345)	160 (2030)	-42	
20	-1 (0015)	315 (1045)	-33 (1430)	191 (2200)	-24
21	-19 (0145)	270 (1130)	-46 (1545)	198 (2330)	-4
22	-35 (0300)	245 (1215)	-50 (1630)	215	
23	3 (0030)	-37 (0415)	223 (1245)	-54 (1700)	208
24	23 (0130)	-34 (0515)	207 (1315)	-58 (1745)	195
25	51 (0230)	-30 (0600)	190 (1345)	-65 (1815)	182
26	78 (0315)	-35 (0700)	176 (1415)	-72 (1845)	165
27	113 (0415)	-24 (0730)	190 (1500)	-49 (1900)	162
28	142 (0500)	-20 (0815)	177 (1530)	-51 (1930)	144
29	175 (0600)	-13 (0845)	179 (1600)	-41 (1945)	128
30	203 (0700)	-9 (0930)	164 (1630)	-46 (2015)	102
31	228 (0800)	-7 (1015)	161 (1730)	-33 (2045)	85

FIGURE 21 —Sample printout of monthly accumulated flow-volume summary produced using IOTOPT option 4.

results can be printed in similar tabular form at every iteration by assigning IOTOPT a value of 1; however, large volumes of printout should be expected from this option. If information such as the minimum, mean, and maximum discharges for the day is needed, it can be derived for each cross section by specifying IOTOPT option 2. The times of occurrence of the minimum and maximum discharges, as well as the concurrent stage, average cross-sectional velocity, and cross-sectional area, are also printed.

In appraising the flow conditions in tide-affected systems one may often be interested in the volume interchange of the network. Volumes of discharges for downstream and upstream flows at specific locations in a network can be obtained by IOTOPT option 4 as is illustrated in figure 21. This option allows for the accumulation of computed flow volumes at all gaging-station locations in the network. The flow discharge in one direction is accumulated by the model until the flow reverses. The discharge for this reversed flow is then accumulated until the next change-of-flow direction. Thus, both upstream and downstream flow volumes are tabulated, and the approximate time of the flow reversal is identified in the monthly flow-volume summaries. Accumulated flow volumes are stored in a direct-access file and are printed in tabular form, as illustrated by figure 21, at the conclusion of each simulation. Knowledge of the flow interchange of the riverine or estuarine system, such as is available via this output option, can be useful in appraising the flushing capacity or mixing capability of the system.

During simulations in which plots are produced for devices other than the line printer, computed results are also printed in tabular form at every time step according to IOTOPT option 0. Actual plot production (other than by the line printer) is accomplished in auxiliary, offline operations from intermediate computer files generated by the model. The CalComp and

FR80 plot files are generated on magnetic tape, whereas Tektronix plot files are stored on a temporary disk file. If a CalComp plot is desired, a plot request form must be submitted to the Production Control Unit of the Computer Center Division in order to have the plot tape processed on the Reston, Va., CalComp plotter.[2] In order to plot FR80 files, follow the installation guidelines where the actual plotting is to be performed. In the case where plotting is to be performed on a Tektronix cathode-ray-tube graphical terminal, the model creates a compressed direct-access-type file for access by the DISSPLA Post Processor program. The required time-sharing option (TSO) commands for allocation of files and execution of the DISSPLA Post Processor routine to access the compressed file and output to a Tektronix graphics terminal of the 4010 series appear in figure 22. After the required TSO commands are entered the DISSPLA Post Processor will pause for input of appropriate control directives, whereupon the terminal RETURN key must be depressed to initiate plotting. Likewise, the terminal pauses at the end of each plot; the next plot can be requested by again depressing the RETURN key.

Any general operational simulation capacity must fulfill a diversity of requirements. For instance one model user might be interested in a graph showing the time-sequential variation of the discharge or water-surface elevation at a particular cross section. On the other hand, a tabular summary of the volume of flow through a particular cross section may be desired by another user. Yet another requirement might be a comparative plot of the before-and-after conditions illustrating the hydrodynamic changes resulting from some alteration of the channel or network configuration. Whatever the particular requirement, branch-network flow model results are made readily available in a variety of formats via several output devices, as identified in the section Input/output description. In particular, the conversational, remote-job-entry facility of the aforementioned interactive display terminal offers unique capabilities to model users. By using the terminal keyboard a model user can enter control parameters for a model execution, initiate the simulation on the host computer, and have alphanumeric or graphical results returned via the display screen – all in a single remote-terminal session. The model-computed discharges at the boundary-value data collection sites for this simulation of the Sacramento River are graphically displayed on the Tektronix terminal, as illustrated in figure 23. Time-dependent water-surface elevations at Sacramento and near Freeport, Calif., processed and stored using the time-dependent data-processing system, provide the boundary-value data necessary to actuate the model simulation. This interactive capability significantly hastens the model calibration and verification operations.

```
ATTR TERM RECFM(F) LRECL(137) BLKSIZE(137)
ALLØC FI(FT05F001) DA(*) USING(TERM)
ALLØC FI(FT06F001) DA(*) USING(TERM)
TERM LINESIZE(136) INPUT(***)
ALLØC FI(CØMPIN) DA(your plot file name)
CALL 'SYS1.DISSPLA.STEPLIB(TEKPØP)'
```

FIGURE 22.—TSO allocation and execution commands for Tektronix plotting by use of the DISSPLA Post Processor.

Branch-Network Model Applications

The branch-network flow model is presently being utilized to simulate the time-varying flows in several coastal and upland river systems. These implementations of the model represent a broad sampling of the hydrologic field conditions one might typically expect to accommodate when contemplating model use. Other implementations of the model – including singular and simply or multiply connected channels – are in process, are being actively planned, or are under consideration.

In order to demonstrate the general applicability of the model, five specific implementa-

[2] This may be accomplished, by registered users of the U S. Geological Survey computer system, by telephoning the Automatic Data Processing Unit, Water Resources Divison (telephone (non-FTS) 703-860-7131 or (FTS) 928-7131), or the Production Control Unit, Computer Center Division (telephone (non-FTS) 703-860-7171 or (FTS) 928-7171), and providing the necessary plot instructions The finished plots will be mailed to the address specified by the requestor

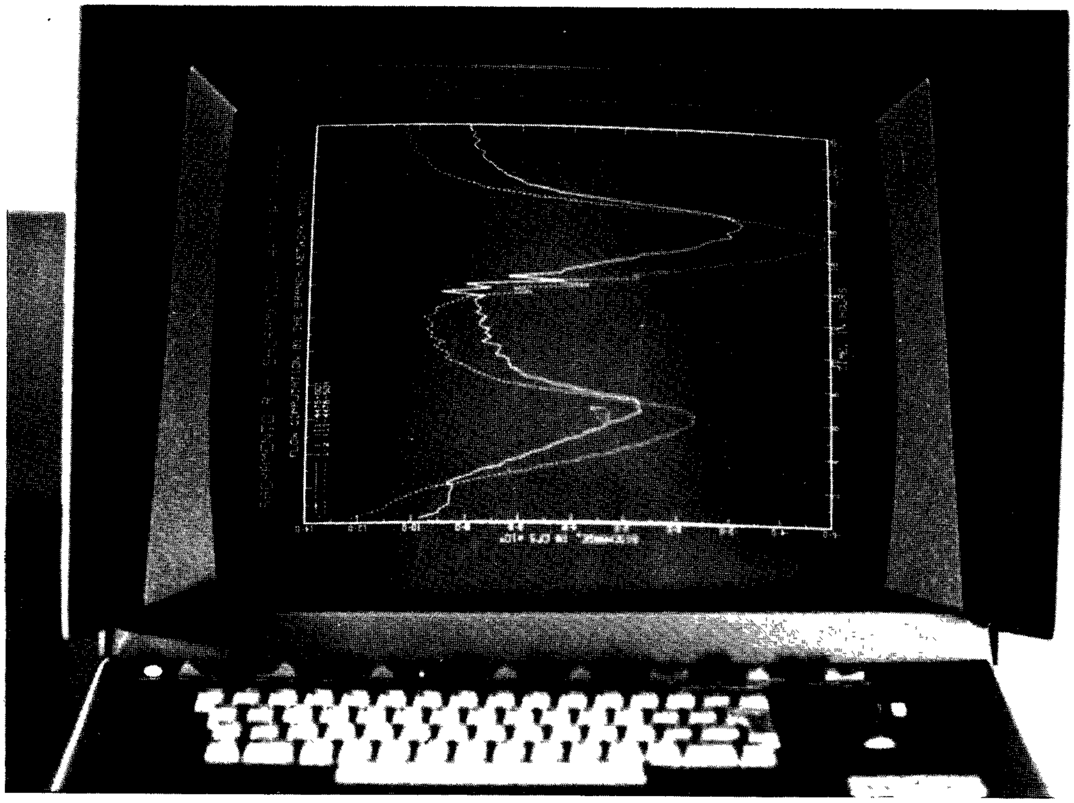


FIGURE 23.—Closeup view of Tektronix cathode-ray-tube display unit.

tions are presented and discussed briefly herein. These particular applications were selected to illustrate not only the means for model implementation but also to portray the adaptability of the model to a wide range of hydrologic conditions and field situations. The implementations depict such diverse field situations as hydropower-plant regulated flows in a singular upland-river reach, tide-induced flows in both riverine and estuarine reaches, and meteorologically generated seiches and wind tides in a multiply connected network of channels joining two large lakes. These example model implementations are in various phases of calibration and verification and are presented merely as illustrations of the model's capabilities. In presenting the findings of these illustrative applications, only those details pertinent to an understanding of the overall implementation process are introduced and discussed.

It is important to recognize that the diversity of applications illustrated is not due solely to the development and subsequent existence of the branch-network flow model. Rather, what is presented in these applications is a product of a systematic, broadly applicable, and operationally usable means for conducting numerical simulation modeling. In order to achieve this capacity for modeling, the branch-network flow model is integrally linked with an efficient, versatile, and highly economical data storage-and-retrieval technique having access to a data base of time-dependent boundary values. Thus, the storage-and-retrieval technique incorporates an automatic mechanism within the model by which to selectively acquire and manipulate the large volumes of boundary-value data typically required for operational simulation modeling. Furthermore, the model is supported by a computer software system whose function is to process field-recorded channel geometry and to produce stage-area-width tables which depict the channel geometry. Yet another distinct feature of the branch-network flow model is its inherent graphics capability, which not only significantly hastens the model calibration and verification operations but also provides a unique, rapid, economical mechanism for evaluating solution alternatives to water-management problems.

Sacramento-Freeport reach of the Sacramento River

Data from the tide-affected reach of the Sacramento River extending downstream from the city of Sacramento to a location near the town of Freeport, Calif., have been collected and used for model research and development purposes for many years. Data from this 17.4-km reach of the Sacramento River have been used extensively throughout this report for simulations illustrating the computational behavior of the branch-network flow model. The modeled reach, as illustrated in figure 24, has recently been extended downstream to the town of Hood, Calif. The unsteady flow is caused by tide-induced translatory waves propagating upstream through San Francisco Bay from the Pacific Ocean. During periods of high upland discharge, which generally occur during the winter and spring months, tide effects are negligible if not completely absent from the reach, and steady flow prevails. During periods of extreme low upland river runoff, tide-induced reversals in the flow direction have been detected as far upstream as the city of Sacramento. However, during these low-flow periods, no salt-water intrusion has been detected in the reach, and the flow remains well mixed vertically. Because of this extensive range of flow conditions, the Sacramento-Freeport reach provides an excellent source of prototype data for evaluating experimental modeling techniques and for conducting model-sensitivity analyses.

As the Sacramento River traverses the lowlands of the Central Valley of California it is almost entirely confined within levees, exhibiting only a slight expansion in cross-sectional area throughout the reach. Inflow and diversions are negligible within the Sacramento-to-Freeport reach. A comprehensive field survey was conducted in order to establish the overall channel properties of the reach being modeled. After the field data were digitally encoded, compiled, and punched on computer cards, they were input to the cross-sectional geometry program, which subsequently produced the stage-area-width tabulations of channel geometry in the format shown in table 1.

The boundary-value data used for simulation of the flow in the Sacramento River consist of water-surface elevations digitally recorded at the upstream and downstream ends of the modeled reach. Such boundary-value data are continuously recorded at the city of Sacramento, near the town of Freeport, and at Hood, Calif., at stations numbered 11–4475.00, 11–4476.50, and 11–4476.52, respectively, in figure 24. For the reach being simulated in this example application, boundary-value data recorded at the city of Sacramento and near the town of Freeport are used. The digitally recorded water-surface elevations are prepared for input to the model through use of the time-dependent data-processing system.

After the model was set up and an initial batch of required boundary-value data were processed and filed, a flow-resistance coefficient (η) had to be determined before the actual simulations could commence. In the branch-network flow model η can be expressed as a constant or as a polynomial function of the water temperature, the flow depth, the discharge, the Reynolds number, or the Froude number. The determination of a functional relationship and the definition of its appropriate coefficients are derived during the model-calibration process by the model user. In general, if a functional relationship is to be used, then definition of suitable coefficient values requires prototype data representing a range of flow and other hydrologic conditions sufficient to permit accurate determination of this relationship.

For calibration of the Sacramento–Freeport model, a constant η was used initially; it produced reasonable results. However, as the flow computation was extended to longer periods of time and to a wider range of flow and other hydrologic conditions, it was found necessary to use a variable η to produce satisfactory results. Consequently, η has been expressed in terms of a quadratic function of the discharge, with the result that the model now produces more accurate results for a wider range of flow regimes.

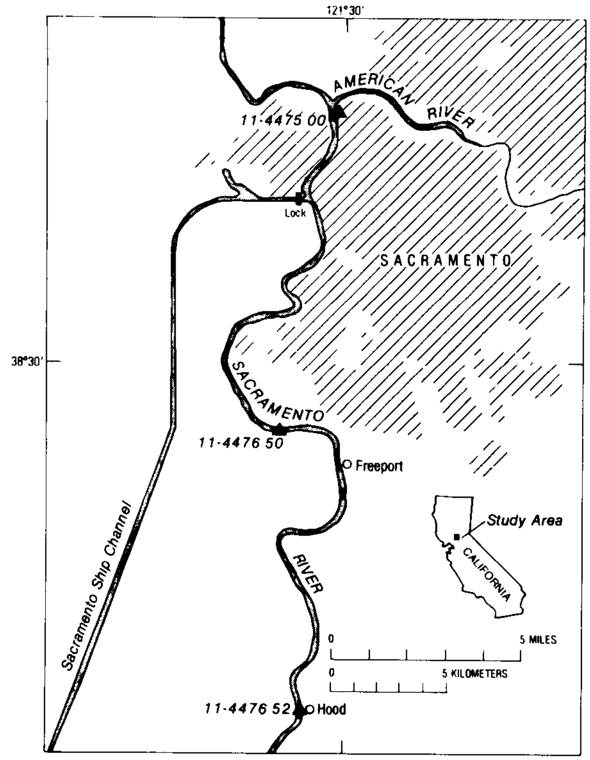


FIGURE 24.—Sacramento River reach near Sacramento, Calif.

The functional relationship determined to be applicable to the schematization of the Sacramento–Freeport reach (Oltmann, 1979) is $\eta = 2.620 \times 10^{-2} + 1.283 \times 10^{-7} Q - 4.167 \times 10^{-13} Q^2$.

Several sets of measured discharges and recorded stages ranging from extremely low to extremely high flows have been used to calibrate the model. One such sample simulation for calibration purposes is illustrated in figure 25. Stages concurrently recorded at the Sacramento and Freeport gages on October 12, 1976, were used as boundary values for this model simulation. This comparative plot of computed versus measured discharges for the Sacramento River at the city of Sacramento was computer produced during the flow simulation and subsequently digitally plotted in an offline operation. Through the use of a range of such sets of discharge data and concurrently recorded stages, the flow-resistance coefficient used in the model has been adjusted, and the flow computations have gradually improved. Thus, the model itself has been used to deduce a functionally dependent resistance coefficient that suits the model schematization employed.

Columbia River reach at Rocky Reach Dam near Wenatchee, Wash.

The branch-network flow model has been used to compute the flow of the Columbia River immediately downstream from Rocky Reach Dam near Wenatchee, Wash. This relatively short reach (3.1 km) is treated as a single-segment branch in the model schematization. Flow in the reach is highly unsteady owing to regulation created by the combined operation of turbines and gates at the dam for the purpose of optimal hydroelectric power generation.

Channel geometry data for the model were abstracted from detailed field surveys, processed by the cross-sectional geometry program, and prepared for input to the model. The branch-network flow model treats the reach as a single segment; therefore, stage-area-width tables were produced that define the upstream and downstream cross sections at the boundary-value data locations.

Water-surface elevations are used as boundary conditions for the model application. These data are collected on a continuous basis at the field-station locations (stations numbered 12–4537.00 and 12–4537.01) identified in figure 26 near river miles 471 and 473, respectively. The close proximity of the boundary-value stations underscores the importance of precise synchronized recording of the water-surface elevations. The boundary-value data are extracted from the time-dependent data base during the simulation as required to define the boundary conditions.

The highly unsteady nature of the flow is illustrated in the model-generated plot of computed discharges in figure 27. As this figure illustrates, the unsteady discharge can vary as much as 2,000 m^3/s in less than 2 hours elapsed time. In fact, the discharge has been observed to vary as much as 1,000 m^3/s in less than 0.5 hour. This application amply demonstrates the ability of the branch-network flow model to simulate highly varying flow conditions, as may be encountered in regulated upland rivers.

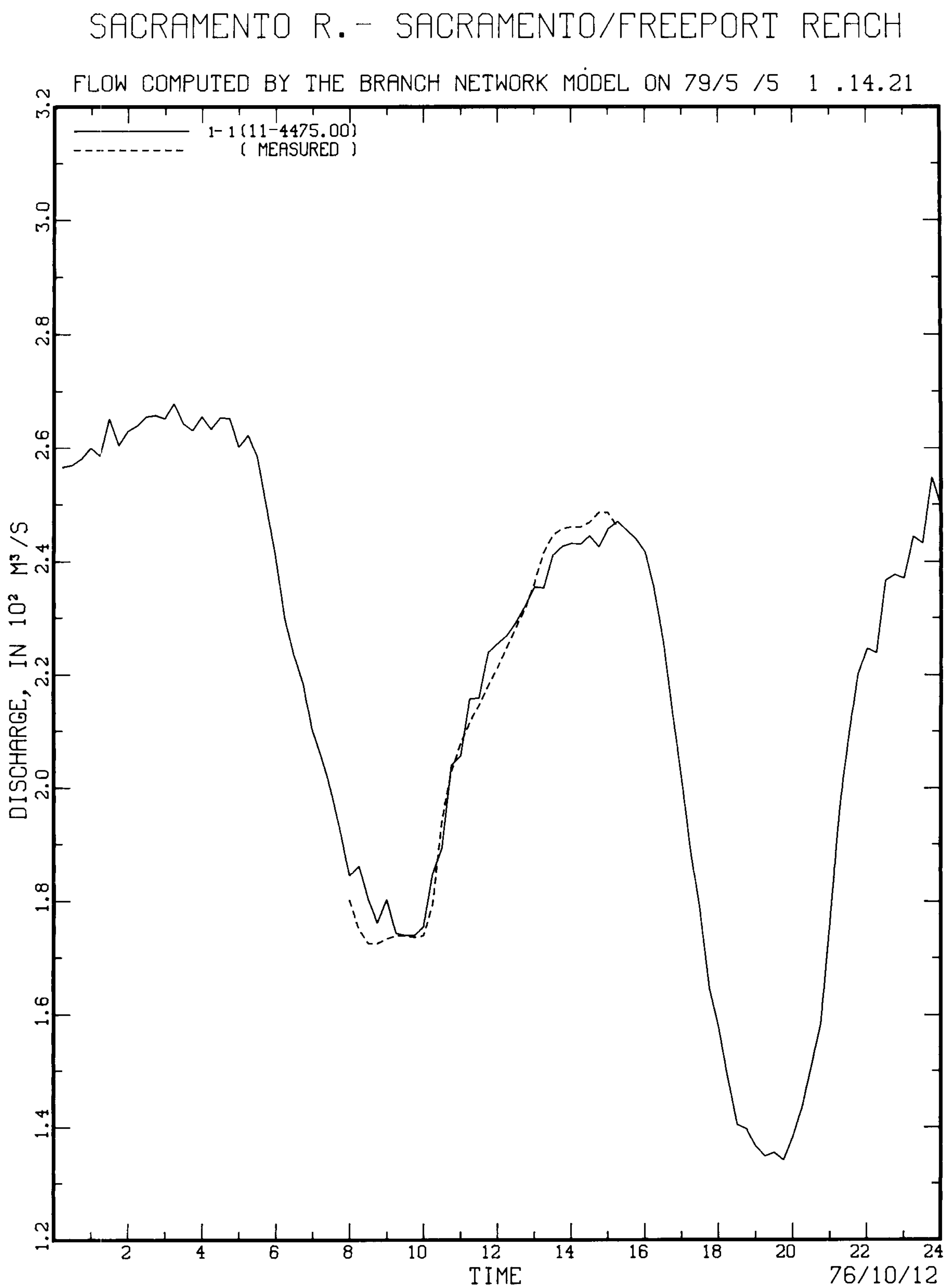


FIGURE 25.—Model-generated plot of computed-versus-measured discharges for the Sacramento River, produced using IOTOPT option 3, IPLOPT option 3, and IPLDEV option 2.

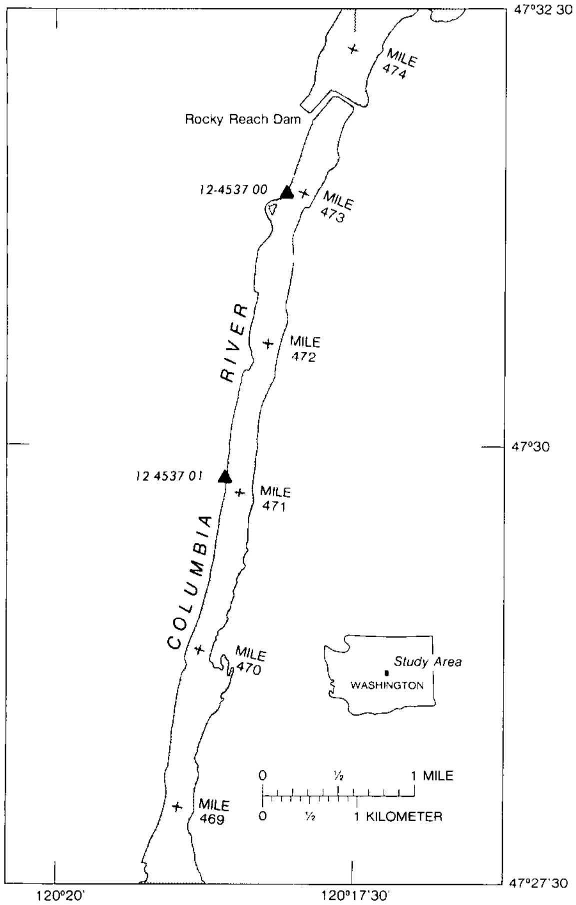


FIGURE 26.—Columbia River reach near Rocky Reach Dam in the State of Washington.

Kootenai River reach near Porthill, Idaho

The branch-network flow model has been used to simulate the flows in a 54.75-km reach of the Kootenai River near Porthill, Idaho. The reach being modeled is delineated by the gaging stations located at Klockman Ranch and at the town of Porthill, Idaho (stations numbered 12–3140.00 and 12–3220.00, respectively), as illustrated in figure 28. Water-surface elevations were also previously recorded at Copeland, Idaho (station number 12–3185.00), which is located near midreach. As figure 28 illustrates, the Kootenai River is characterized by a rather sinuous, meandering path of travel. The river flows in a northerly direction through a more-or-less diked channel traversing a rather narrow mountain-valley flood plain before crossing the international boundary and discharging into Kootenay Lake in Canada. Depending on the regulated level of this lake, varying backwater effects occur at the Porthill gaging location. The reach is also subject to regulated flows propagating downstream from nearby Libby Dam in Montana. It is the combination of these regulated flows and backwater effects that necessitates the use of an unsteady-flow model in order to produce accurate flow information at the Porthill international-boundary gaging station.

The reach between the Klockman Ranch gage (station number 12–3140.00) and the Porthill gage (station number 12–3220.00) is subdivided into five branches for purposes of model implementation. Four internal junctions are located at the Trout Creek tributary, at the Copeland gage, at the Parker Creek tributary, and at the Smith Creek tributary (fig. 28). These junctions are located in order to account for the tributary inflow. Tributary discharges are treated as nodal inflows occurring at the junction locations.

From survey data collected at 15 cross sections between the boundary-value-data gaging stations, a set of geometry tables for the branch-network flow model was constructed for purposes of conducting the simulations. The cross-sectional data were analyzed, and tables were prepared for use in the model via the cross-sectional geometry program.

Simulations are conducted using simultaneous, digitally recorded, water-surface elevations from the Klockman Ranch and Porthill gage locations as boundary-value data for establishing the required boundary conditions in the model. These data are routinely translated, processed, and filed in the time-dependent data base for direct inclusion in the flow model.

Water-surface elevations previously recorded at the Copeland gage location at midreach, as well as several approximately steady-flow discharge measurements, were used to conduct the preliminary flow-model calibration. Initial calibration of the model using a total of eight such discharge measurements resulted in a maximum error of 10 percent in the computed momentary discharge. The eight discharge measurements represented a range of flow conditions from a low of 250 to a maximum of 800 m^3/s. Additional calibration and verification of

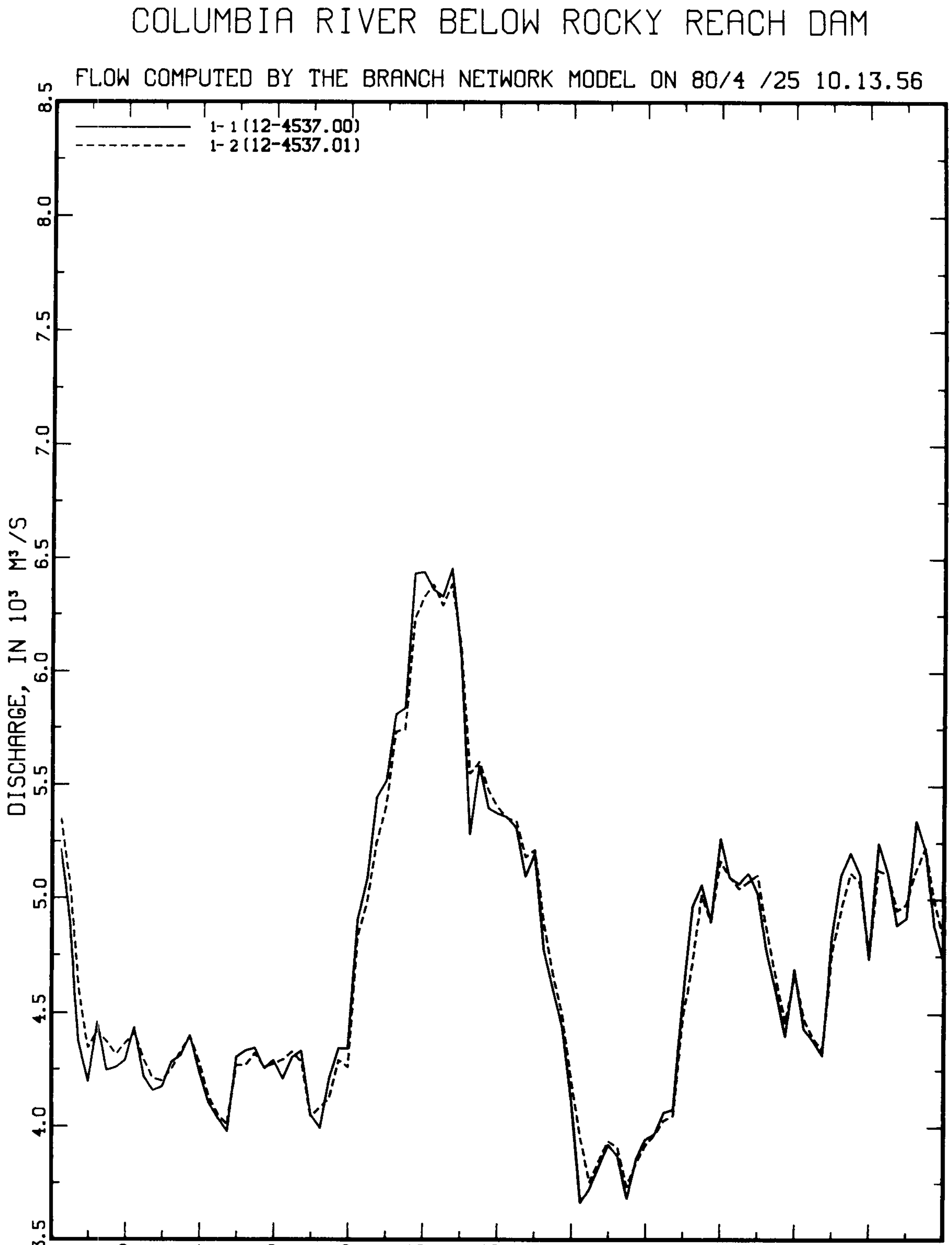


FIGURE 27.—Model-generated plot of computed discharges for the Columbia River, produced using IOTOPT option 3, IPLOPT option 1, and IPLDEV option 2.

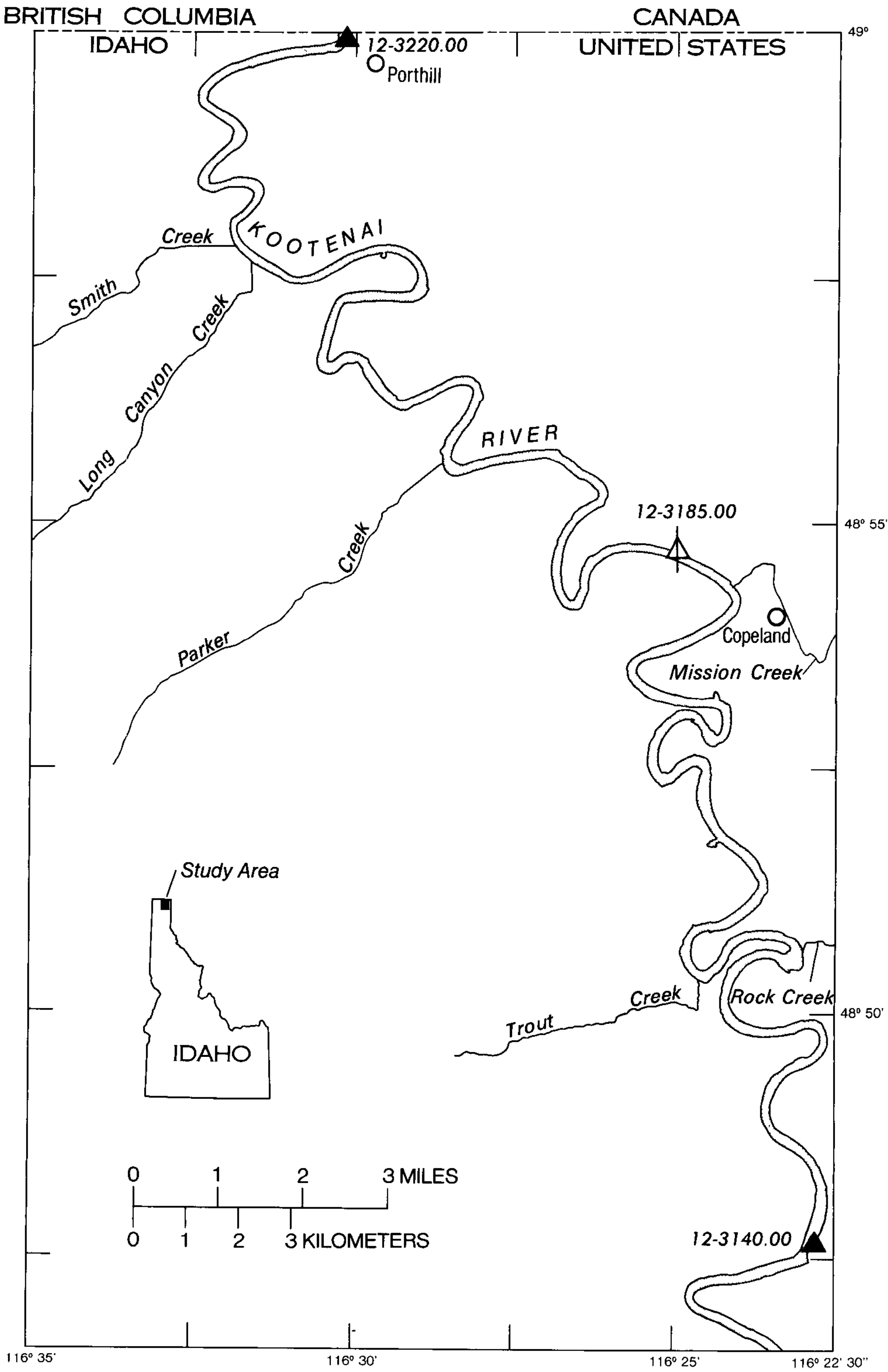


FIGURE 28. — Kootenai River reach near Porthill, Idaho.

the Kootenai River flow model are being conducted.

Figure 29 illustrates another output format available from the branch-network flow model. This output is a tabular summary, on a daily basis, of the computed flows for the Kootenai River. The minimum, mean, and maximum discharges computed at each junction (cross-section) location are identified on successive lines of printout. The daily results printed for August 31, 1978, are indicated as representing a partial day of record. They are in fact, the initial values used to actuate the flow-model simulation. Looking at the computed flow results for a complete day, for example September 2, 1978, one can see that the minimum discharge for the day at each cross section occurred at or near the last computational time for the day (2400 hours). The times of the maximum discharge for the day are similarly identified for each cross section; these occurred between 0515 and 0730 hours at all locations. Note that cross section one of branch one is at the Klockman Ranch gage location and cross section two of branch five is at the Porthill gage location.

By virtue of the schematization, the internal cross sections at common junction locations are, of course, duplicates of one another. Thus, as can be seen from the results illustrated in figure 29, the computed flows at these common junction locations are—as well they should be—nearly identical. A slight difference may be detected and can be attributed to the tributary inflow and

FLOW RESULTS FOR KOOTENAI RIVER BETWEEN KLOCKMANN RANCH AND PORTHILL

	MINIMUM							MEAN	MAXIMUM				
DATE (YR/MO/DY)	TIME (HR MN)	STAGE (M)	VELOCITY (M/S)	DISCHARGE (M**3/S)	AREA (M**2)	BRANCH	SECTION	DISCHARGE (M**3/S)	TIME (HR MN)	STAGE (M)	VELOCITY (M/S)	DISCHARGE (M**3/S)	AREA (M**2)
*78/ 8/31	24 0	13 79	0 16	201 2	1224 4	1	1	201 2	24 0	13 79	0 16	201 2	1224 4
	24· 0	13 75	0 17	206 8	1231 7	1	2	206 8	24 0	13 75	0 17	206 8	1231 7
	24 0	13 75	0 17	206 8	1231 7	2	1	206 8	24. 0	13 75	0.17	206 8	1231 7
	24 0	13 68	0 21	220 7	1060 4	2	2	220 7	24 0	13 68	0 21	220 7	1060.4
	24 0	13 68	0 21	220 7	1060 4	3	1	220 7	24 0	13 68	0 21	220 7	1060 4
	24 0	13 63	0 19	228 9	1190 7	3	2	228 9	24 0	13 63	0 19	228.9	1190 7
	24 0	13 63	0 19	228 9	1190 7	4	1	228 9	24 0	13 63	0 19	228 9	1190 7
	24 0	13 59	0 21	234 4	1112 0	4	2	234 4	24 0	13 59	0 21	234 4	1112 0
	24 0	13 59	0 21	234 4	1112 0	5	1	234 4	24 0	13 59	0 21	234 4	1112 0
	24 0	13 55	0 19	238 6	1268 6	5	2	238 6	24 0	13 55	0 19	238 6	1268 6
78/ 9/ 1	0 15	13 79	0 17	207 5	1224 4	1	1	334 6	9 0	14 19	0 32	405 0	1280 3
	0 15	13 75	0 17	209 3	1231 4	1	2	327 9	9 45	14 10	0 31	393 5	1288 9
	0 15	13 75	0 17	210 3	1231 4	2	1	329 0	9.45	14 10	0 31	394.5	1288.9
	0 30	13 68	0 20	216 6	1059 6	2	2	321 8	10 45	13 93	0 35	383 0	1096 3
	0 30	13 68	0 21	217 7	1059 6	3	1	322 9	10 45	13 93	0 35	384 1	1096 3
	0 45	13 62	0 19	222 9	1189 1	3	2	319 1	12 0	13 81	0 31	379.2	1217 0
	0 45	13 62	0 19	222 9	1189 1	4	1	319 1	12 0	13 81	0.31	379 2	1217.0
	1 0	13 57	0 20	226 8	1109 8	4	2	316 9	11 45	13 70	0.33	376 3	1129 5
	1 0	13 57	0 20	226 8	1109 8	5	1	316 9	11 45	13 70	0 33	376 3	1129 5
	1 45	13 53	0 18	230 0	1265 8	5	2	315 5	11·30	13 61	0 29	373 8	1278 1
78/ 9/ 2	23 45	13 77	0 13	163 0	1221 9	1	1	290 6	5·15	14 29	0 32	409 3	1295.4
	24 0	13 75	0 14	167 7	1231 0	1	2	297 4	5 45	14 19	0.31	401 4	1304 8
	24 0	13 75	0 14	168 7	1231 0	2	1	298 5	5 45	14 19	0 31	402 4	1304 8
	24 0	13 70	0 16	174 3	1063 1	2	2	305 5	6 45	14 02	0 36	394 9	1109 2
	24 0	13 70	0 17	175 5	1063 1	3	1	306 6	6 45	14 02	0 36	396.0	1109 2
	24 0	13 67	0 15	178 8	1196 1	3	2	309 8	7 30	13 89	0 32	392 1	1229 0
	24 0	13 67	0 15	178 8	1196 1	4	1	309 8	7 30	13 89	0 32	392 1	1229.0
	24 0	13 64	0 16	180 3	1120 5	4	2	311 3	7·30	13 78	0 34	390.1	1142.2
	24 0	13 64	0 16	180 3	1120 5	5	1	311 3	7 30	13 78	0 34	390 1	1142 2
	24 0	13 62	0 14	180 7	1279 5	5	2	311 8	7 15	13 69	0 30	388.3	1289 9
78/ 9/ 3	18 15	13 75	0 12	144 8	1218 4	1	1	159 2	13 45	13 76	0 14	172.6	1221 0
	18 45	13 73	0 12	148 4	1228 0	1	2	159 8	14 0	13 74	0 14	168 2	1230 0
	18 45	13 73	0 12	149 5	1228 0	2	1	160 9	14 0	13 74	0 14	169 2	1230 0
	19 0	13 69	0 14	152 9	1061 6	2	2	161 5	0 15	13 70	0 16	173 0	1062 9
	19 0	13 69	0 15	154 0	1061 6	3	1	162 6	0 15	13 70	0 16	174.2	1062 9
	21 15	13 66	0 13	156 9	1194 4	3	2	163 0	0 15	13 67	0 15	176 8	1196 0
	21 15	13 66	0 13	156 9	1194 4	4	1	163 0	0·15	13 67	0 15	176 8	1196 0
	12 15	13 64	0 14	157 2	1119 5	4	2	163 2	0 15	13 64	0 16	177 9	1120 4
	12 15	13 64	0 14	157 2	1119 5	5	1	163 2	0 15	13 64	0 16	177 9	1120 4
	12 0	13 62	0 12	154 8	1279 0	5	2	163 3	0 15	13 62	0 14	178 2	1279 5
78/ 9/ 4	14 30	13 72	0 12	140 2	1214 1	1	1	153 1	4 30	13 75	0 14	165 4	1219 3
	18 30	13 70	0 12	141 0	1223 9	1	2	153 0	6 45	13.73	0 13	162 6	1228 2
	18 30	13 70	0 12	142 0	1223 9	2	1	154 0	6 45	13 73	0 13	163 6	1228 2
	17 15	13 67	0 13	141 4	1058 5	2	2	153 9	7 0	13 69	0 15	161 9	1061 1
	17 15	13 67	0 13	142 5	1058 5	3	1	155 0	7 0	13 69	0.15	163 1	1061.1
	17 15	13 65	0 12	142 6	1192 8	3	2	155 0	7 45	13 66	0 14	163 0	1194 3
	17 15	13 65	0 12	142 6	1192 8	4	1	155 0	7 45	13 66	0.14	163 0	1194 3
	16 30	13 63	0 13	141 4	1118 1	4	2	155 0	7 45	13 63	0 15	164 4	1118 9
	16 30	13 63	0 13	141 4	1118 1	5	1	155 0	7 45	13 63	0 15	164.4	1118 9
	16 15	13 61	0 11	140 5	1278 1	5	2	155 0	7 45	13 61	0 13	164 7	1278 1

* IDENTIFIES A PARTIAL DAY OF RECORD

FIGURE 29.—Sample output of the daily summary of computed flow for the Kootenai River, produced using IOTOPT option 2.

(or) the specified computational convergence criteria.

Connecticut River reach near Hartford, Conn.

The Connecticut River, the largest river in New England, bisects the State of Connecticut as it flows southward to Long Island Sound. The river is tide affected from its mouth on Long Island Sound northward almost to the Connecticut-Massachusetts State line. The extent of tidal influence is greatly controlled by the amount of fresh-water inflow and, therefore, varies considerably from season to season. Although the river near Hartford, Conn., is frequently influenced by tidal effects, the direction of flow is always downstream. Flow reversals, however, often occur between Middletown and Hartford, even though much of the tide effect is suppressed by fresh-water inflow during periods of high upland runoff. Thus, within the 41.2-km reach being modeled between the Bulkeley Bridge gage at Hartford and the CANEL Pier gage, which is 9.9 river kilometers downstream of Middletown (stations numbered 01–1900.70 and 01–1930.50, respectively, in fig. 30), the flow conditions vary from steady to unsteady flow.

In order to conduct the simulations, boundary conditions are specified via water-surface elevation data continuously recorded at the Bulkeley Bridge and CANEL Pier gage locations. These continuously recorded, water-surface elevations are processed and prepared for input to the model through the time-dependent data-processing system.

For simulation purposes the modeled reach of the Connecticut River is treated as two branches each consisting of four unequal segments, therefore requiring five stage-area-width tables per branch to delineate the respective segments. The internal junction is located at approximately the midpoint of the overall reach, 21.4 km downstream of the Bulkeley Bridge gage near Cromwell, Conn. Cross-sectional data, obtained by depth soundings, from topographic maps, or from marine charts, were used to defined the average cross-sectional properties of the channel at the segment ends. The cross-sectional geometry program reduced the profile data to the required stage-area-width tables for direct input to the branch-network flow model.

Data for calibration and verification of the model consist of several discharge measurements made over at least a tidal cycle. Water-surface elevations were also observed during the periods of these discharge measurements at several sites along the reach selected in order to assist in the overall model calibration.

Figure 31 shows computed versus measured water-surface elevations for the Connecticut River at the Bodkin Rock gage location (field station number 01–1930.00), identified in the figure as branch cross-sectional number 2–4, which is 6.2 river kilometers upstream of the CANEL Pier gage. As can be seen from the figure, the agreement appears satisfactory exhibiting mainly a slight difference in phasing. The maximum discrepancy between measured and computed stages is approximately 2.1 cm, occurring at 1230 hours.

Figure 32*A*, *B* illustrates plots of discharge hydrographs available via the branch-network flow model. Each curve in these plots depicts the computed discharge at a specific cross section, in other words, at the ends of each segment. The results computed at all nine cross-sectional locations are illustrated in these figures. The curves are identified by branch cross-sectional numbers. Discharge hydrographs for branch 1 are plotted in figure 32*A*, whereas hydrographs plotted in figure 32*B* are for branch 2. Of course, at the common internal junction location near Cromwell, Conn., the computed results identified by branch cross-sectional numbers 1–5 and 2–1 are identical by virtue of the model schematization. From these plots one can perceive the attenuating effect on the discharge of the tide wave as it propagates upstream to the Bulkeley Bridge gage site.

Detroit River between Lake St. Clair and Lake Erie

The Detroit River is a navigable international waterway. More specifically it comprises a series of interconnected channels joining Lake St. Clair with Lake Erie, as shown in figure 33.

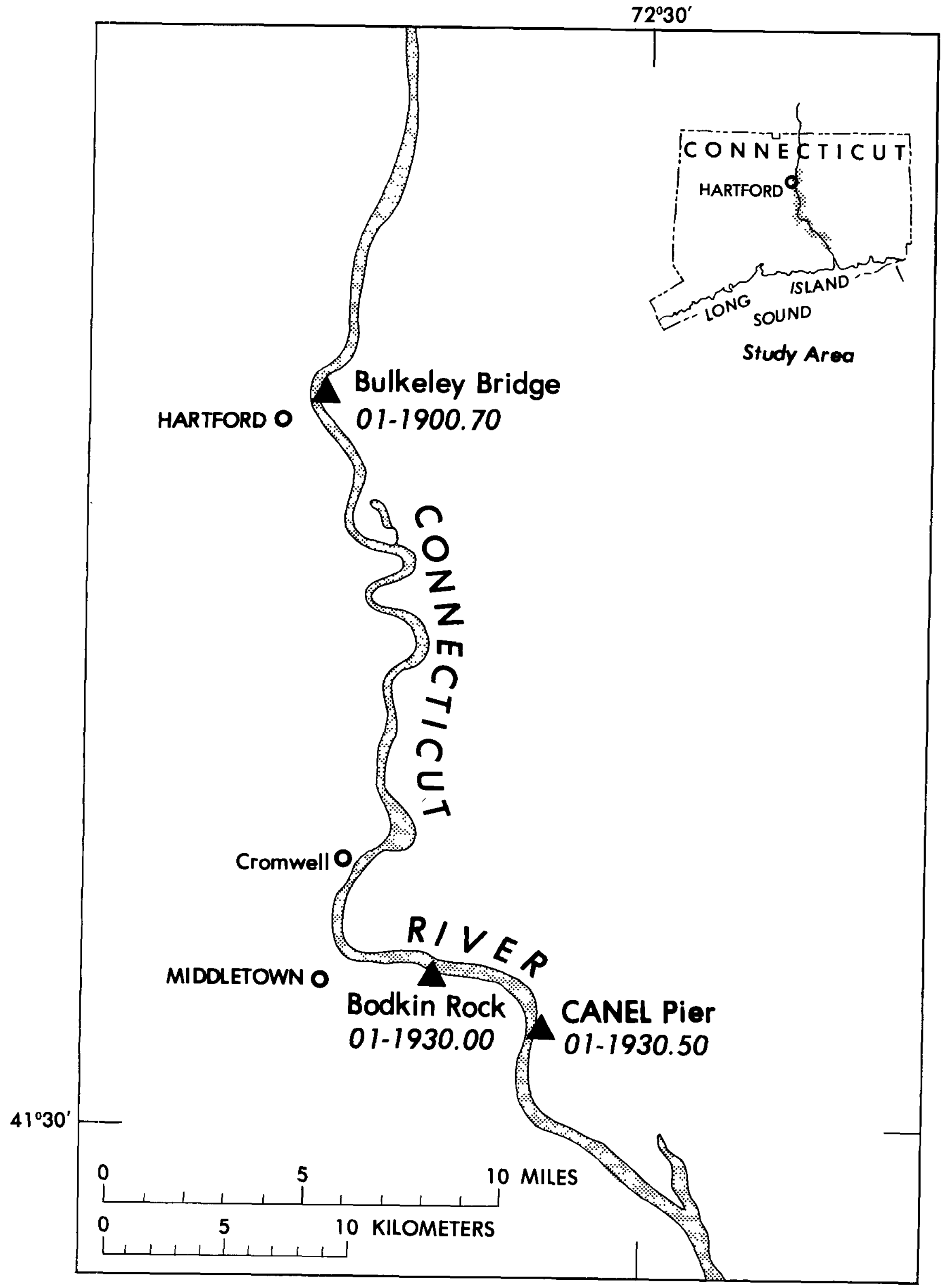


FIGURE 30. – Connecticut River reach near Hartford, Conn.

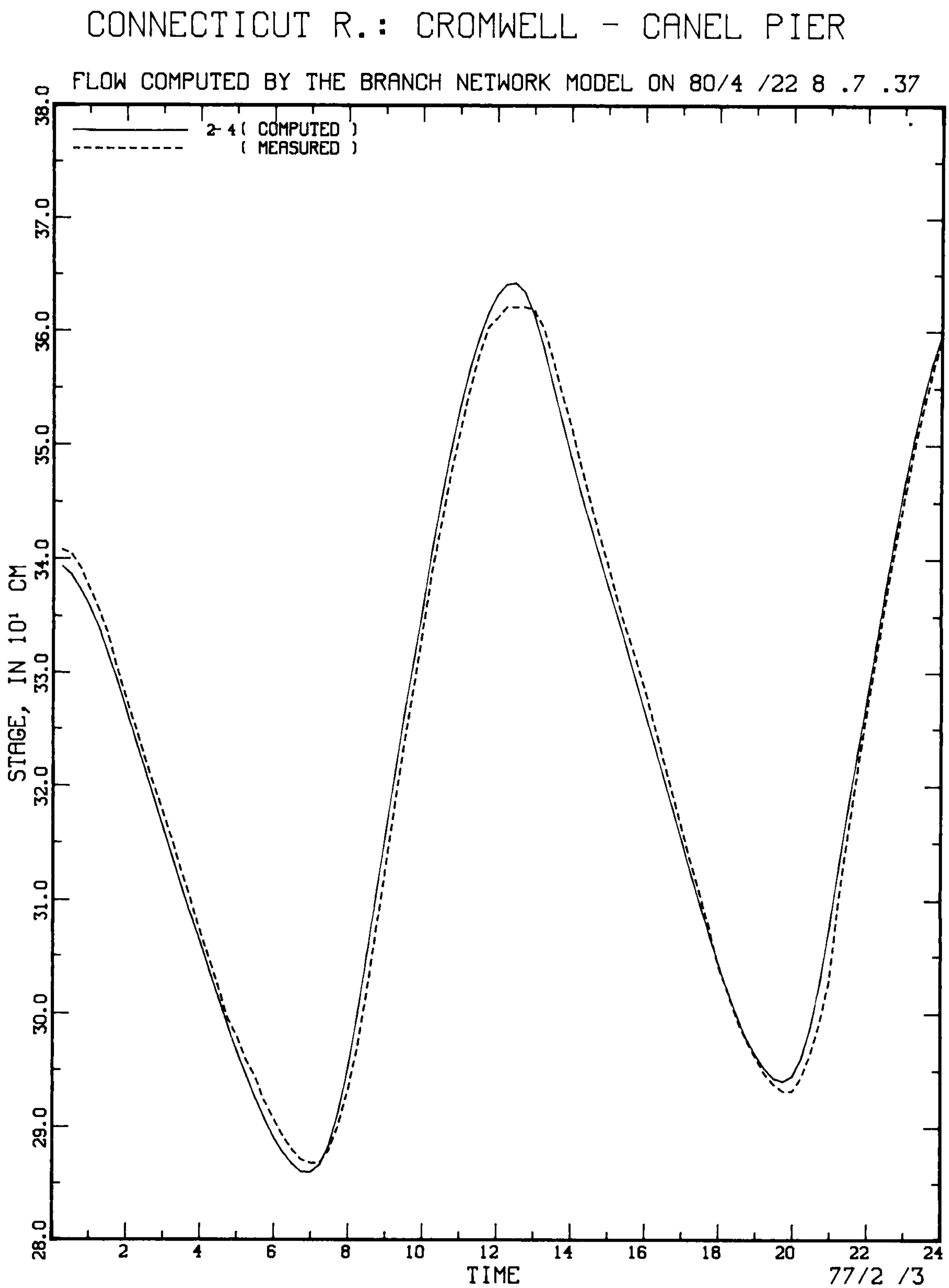


FIGURE 31.—Model-generated plot of computed-versus-measured water-surface elevations for the Connecticut River, produced using IOTOPT option 3, IPLOPT option 4, and IPLDEV option 2.

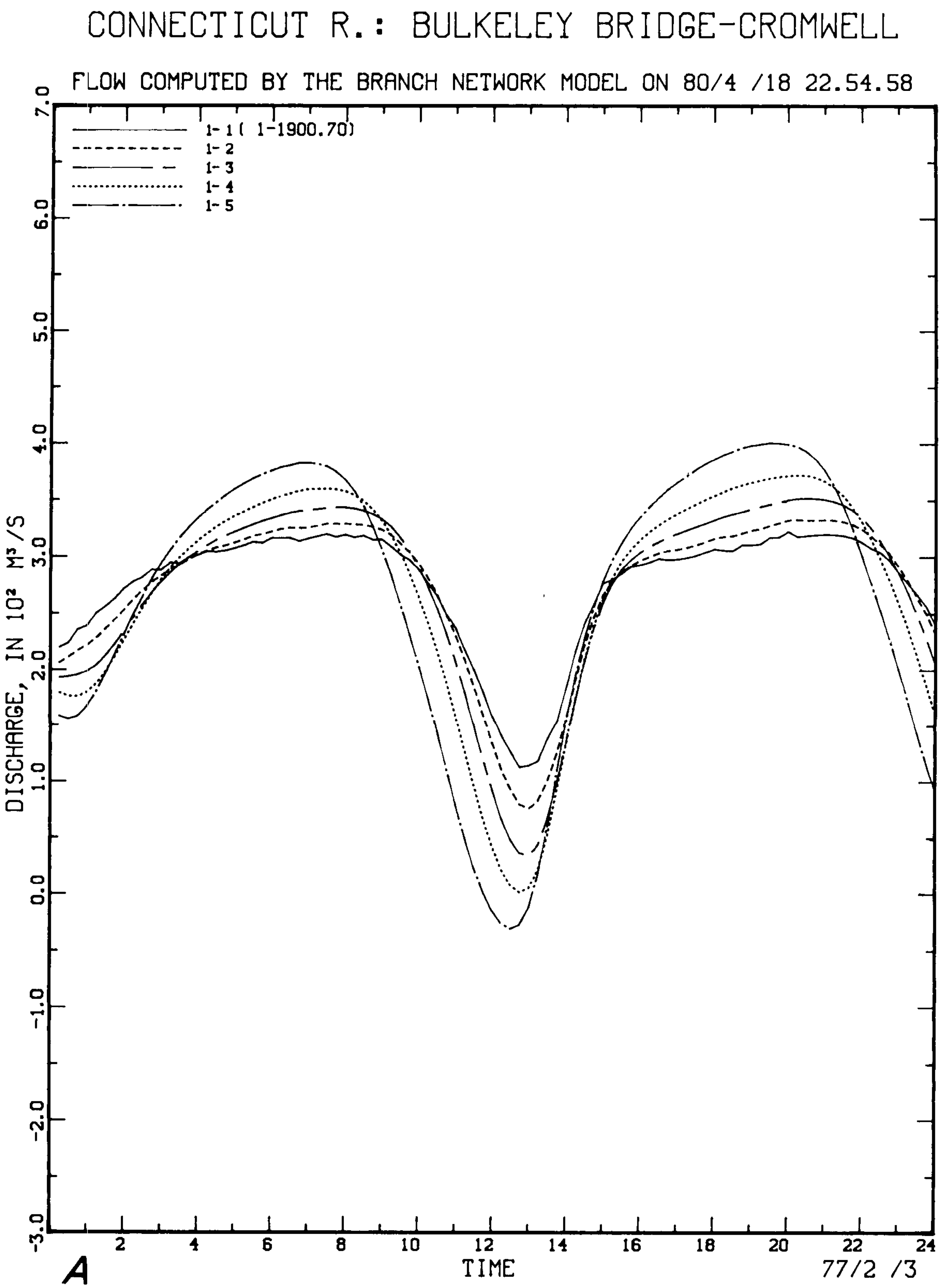


FIGURE 32.—Model-generated hydrographs of computed discharges for the Connecticut River, produced using IOTOPT option 3, IPLOPT option 1, and IPLDEV option 2. *A*, Branch 1. *B*, Branch 2.

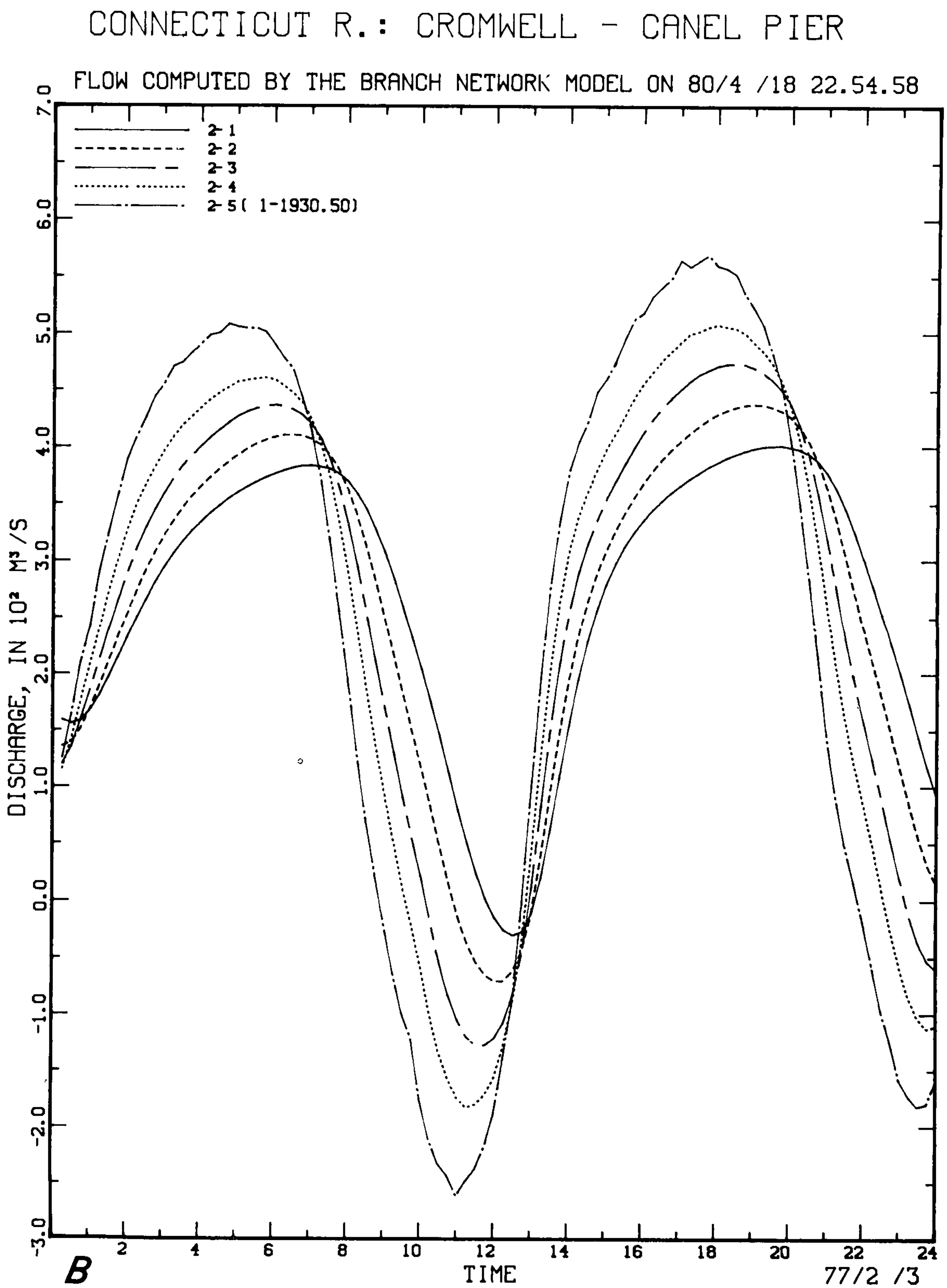


FIGURE 32.—Continued.

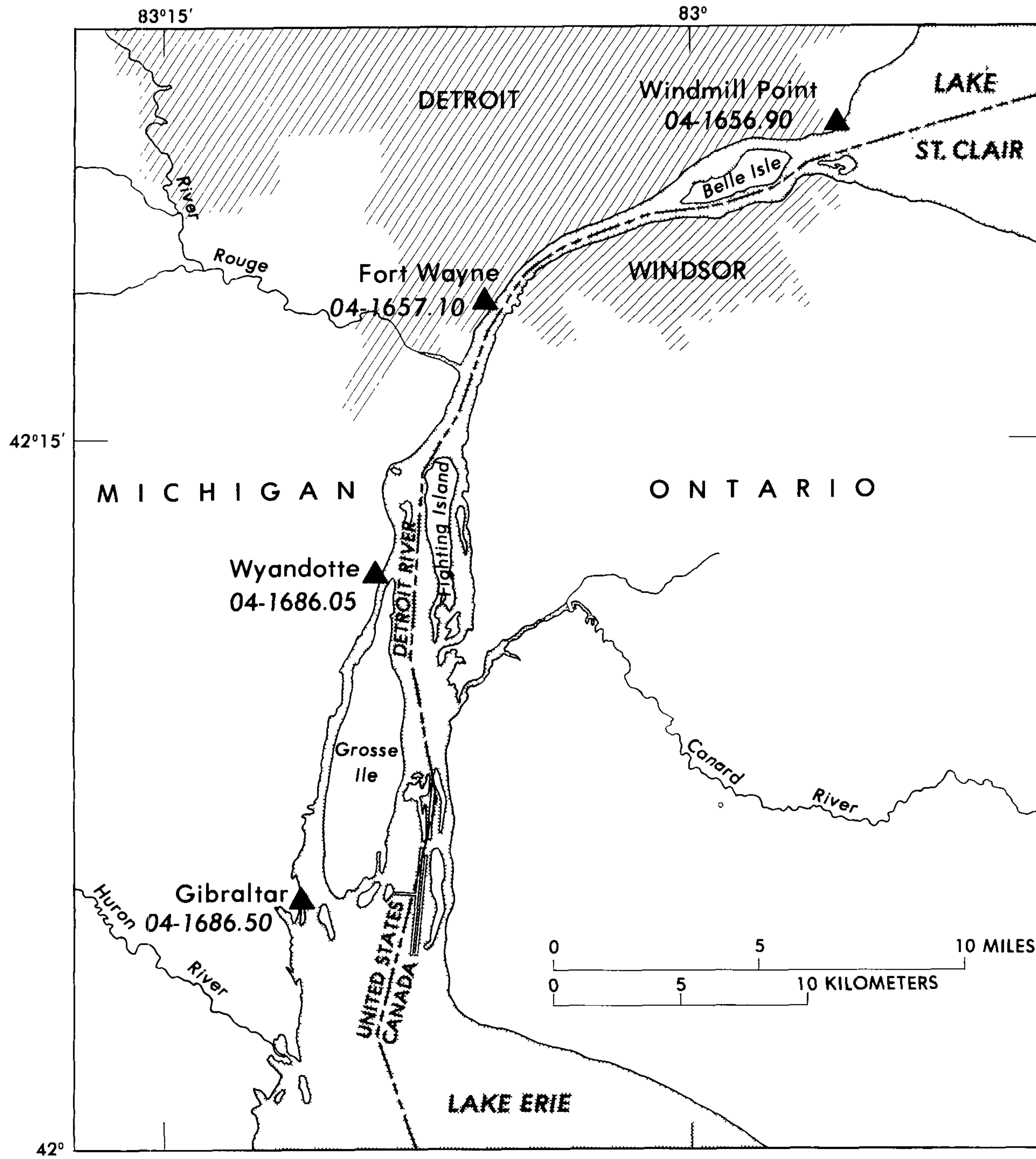


FIGURE 33.—Detroit River near Detroit, Mich.

Significant fluctuations in its flow result from wind tides and meteorologically induced seiches occurring in both Lakes St. Clair and Erie. During severe wind conditions the elevation of the water surface of Lake Erie at the mouth of the Detroit River has been observed to fluctuate as much as 2.5 m over a period of less than one day. Similar, but less pronounced, fluctuations of the water-surface elevation have been observed in Lake St. Clair at the headwaters of the river.

Significant, also, is the fact that the flow of the Detroit River is affected not only by the wind setup and seiching occurring in the adjoining lakes but that the flow can be appreciably affected, because of the river's length, breadth, and orientation, by wind shear acting on its surface.

The flow in the Detroit River and particularly fluctuations in the flow are of interest to those concerned with navigation, water supply, water quality, and industry. Knowledge of the flow is of international importance as well. Thus, the interconnecting channels of the Detroit River have been schematized, and the branch-network flow model is being used to simulate flows at selected locations. Boundary conditions for these simulations consist of stage hydrographs for Lake St. Clair (recorded at the Windmill Point gage location) and Lake Erie (recorded at the Gibraltar gage location).

The Detroit River application is truly a network simulation in the sense that the model schematization specifically accounts for the presence of one minor and three major islands within the river. The major islands of Belle Isle, Fighting Island, and Grosse Ile separate the Detroit River into various channels–and therefore, multiple flow paths–as can be seen in figure 33. A minor island at the headwaters of the river at Lake St. Clair is also accounted for in model schematization. The model schematization of the Detroit River is indicated in figure 34. As this figure indicates the full length of the river extending from a water-level gage at Windmill Point, Mich., to another near Gibraltar, Mich. (stations numbered 04–1656.90 and 04–1686.50, respectively) is being modeled. The overall length of the principal reach is 38.9 km. Water-surface elevations are also monitored and digitally recorded at intermediate locations at Fort Wayne, Mich., and Wyandotte, Mich. (stations numbered 04–1657.10 and 04–1686.05, respectively). These water-surface elevations constitute the boundary-value data, as well as data for model calibration and (or) verification.

As figure 34 illustrates, the Detroit River is treated as a network of 12 single-segment branches. Cross-sectional geometry data depicting the various channels were derived from hydrographic charts. Stage-area-width tables were prepared which describe the cross section at the beginning and end of each branch identified by a Roman numeral in figure 34. The 11 junctions of the model are identified by the numbered squares in figure 34. External boundary conditions are required at junctions 1, 2, 10, and 11. Recorded water-surface elevations at Windmill Point are used to define the boundary conditions at junctions 1 and 2. Recorded water-surface elevations at the Gibraltar gage location are used to define the boundary conditions at junctions 10 and 11. The required boundary conditions at internal junctions are established automatically by the model program.

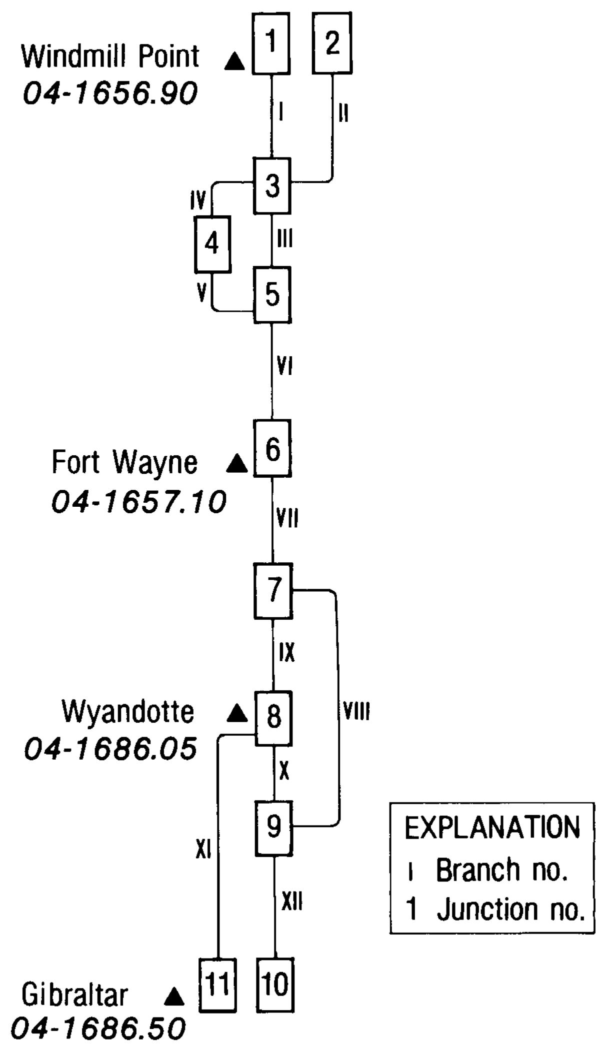


FIGURE 34.—Schematization of the Detroit River for the branch-network flow model.

```
//COMPUTER JOB CARD
/*PROCLIB  SCHAF PROCLIB
//BRANCH EXEC BRANCH,PROG=BRANOP,ECORE=360K,
// XSNAME='SCHAF BRANCH DETROIT GEOMETRY',
// GINAME='BALTZ GPHINDX1 DETROIT',
// DANAME='SCHAF TIMEDPDT DACCDATA'
//SYSIN DD *
DETROIT RIVER BETWEEN LAKE ST CLAIR AND LAKE ERIE
  1211 4     ME10  34 1 11       15 60500 0
 1 3 2DETROIT R    WINDMILL POINT - BELLE ISLE
     4 46 151531 00              5500 00      59 000 2235E-010 0        0 0
   135 00     1 0000
     4 34 151162 00

 2 3 2DETROIT R    PEACH ISLAND - BELLE ISLE
     4 46  50766 00              5500 00      59 000 2385E-010 0        0 0
   106 00     1 0000
     4 34  50618 00

 3 5 2DETROIT R    FLEMING CHANNEL
     4 34 142830 00             19500 00      59 000 2235E-010 0        0 0
   137 00     1 0000
     4 08 142239 00

 3 4 2DETROIT R    NORTH BELLE ISLE CHANNEL
     4 34  58949 00             10500 00      59 000 2485E-010 0        0 0
   108 00     1 0000
     4 20  58635 00

 4 5 2DETROIT R    SOUTH BELLE ISLE CHANNEL
     4 20  58635 00             10500 00      59 000 2485E-010 0        0 0
   157 00     1 0000
     4 08  58353 00

 5 6 2DETROIT R    BELLE ISLE - FORT WAYNE
     4 08 200592 00             25000 00      59 000 2235E-010 0        0 0
   147 00     1 0000
     3 81 200404 00

 6 7 2DETROIT R    FORT WAYNE - FIGHTING IS
     3 81 200404 00             19000 00      59 000.2235E-010 0        0 0
   170 00     1 0000
     3 59 200654 00

 7 9 2DETROIT R    EAST FIGHTING IS  CHANNEL
     3 59  45602 00             35500 00      59 000 2585E-010 0        0 0
   149 00     1 0000
     3 16  46271 00

 7 8 2DETROIT R    FIGHTING ISLAND CHANNEL
     3 59 155052 00             18500 00      59 000 2285E-010 0        0 0
   147 00     1 0000
     3 32 155845 00

 8 9 2DETROIT R    WYANDOTTE - GROSSE ILE
     3 32 102613 00             16500 00      59 000 2285E-010 0        0 0
   147 00     1 0000
     3 16 103066 00

 811 2DETROIT R    TRENTON CHANNEL
     3 32  53232 00             41500 00      59 000 2335E-010 0        0 0
   162 00     1 0000
     2 19  54287 00

 910 2DETROIT R    EAST GROSSE ILE CHANNEL
     3 16 149339 00             23500 00      59 000 2785E-010 0        0 0
   148 00     1 0000
     2 19 151521 00


3330                LIST_INDEX BEFORE= 0AFTER= 0
 Z 1       4165690       78/ 7/25 00 15      78/ 7/25 24 00     288  -70 0
 Z 2     5 4165690       78/ 7/25 00 15      78/ 7/25 24 00          -70 0
 Z10     5 4168650       78/ 7/25 00 15      78/ 7/25 24 00     288  -70 0
 Z11       4168650       78/ 7/25 00 15      78/ 7/25 24 00     288  -70 0
 Z 8       4168605       78/ 7/25 00 15      78/ 7/25 24 00      96  -70 0
/*
//
```

FIGURE 35.—Sample deck setup to execute the branch-network flow model of the Detroit River.

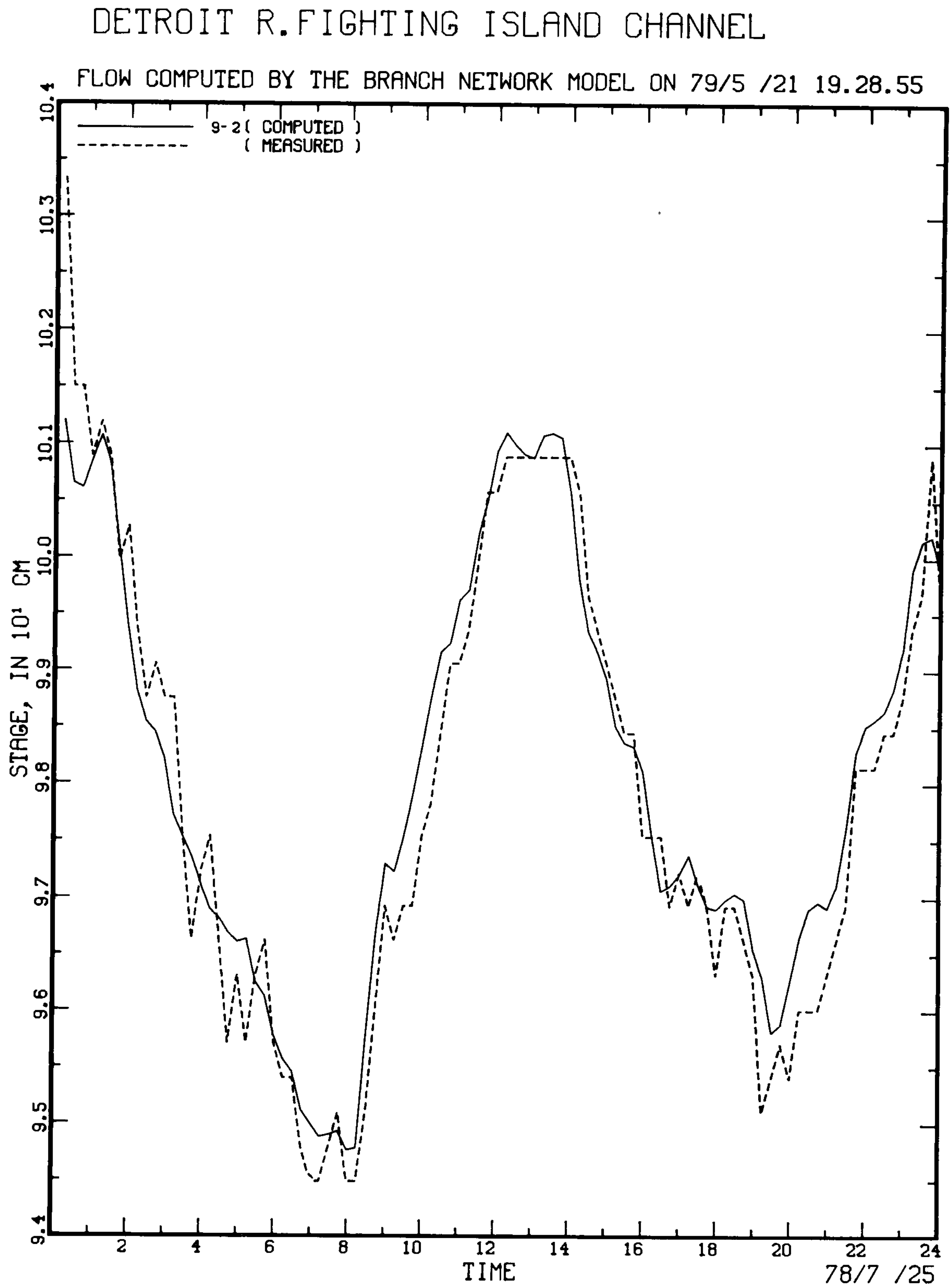


FIGURE 36. – Model-generated plot of computed-versus-measured water-surface elevations for the Detroit River, produced using IOTOPT option 3, IPLOPT option 4, and IPLDEV option 2.

A sample card deck setup to execute the branch-network flow model of the Detroit River is shown in figure 35. This figure is included to illustrate the relative ease and operational simplicity with which a complete flow simulation model may be initiated by the model user. In this particular execution setup, cross-sectional geometry, as well as boundary-value data, are retrieved from computer files. Cross-sectional geometry tables are retrieved from a computer file established by the cross-sectional geometry program, whereas boundary-value data are retrieved from a data base of time-dependent data. The initial conditions for the simulation depicted in figure 35 were computed and subsequently punched from a previous simulation. The model is set up to execute on a 15-minute time step using a value of 0.6 for θ and χ as defined in the section Finite-difference formulation.

The sample deck setup illustrated in figure 35 is intended to produce a line-printer plot of computed versus measured water-surface elevations. The model-generated graph, plotted via a Tektronix interactive terminal and illustrated in figure 36, was derived from a similar deck setup. This output represents a plot of the computed versus measured water-surface elevations at the Wyandotte gage location (fig. 33). In general, the agreement between computed and measured stages appears to be satisfactory; however, additional calibration and verification of this particular model are required. More conclusive tests of the model must await collection of synoptic sets of measured discharges, wind-vector data, and, of course, boundary-value water-level data for various flow and meteorological conditions. Computed discharges were within 3.5 percent of the measured discharges for one such set of synoptic data collected near the Fort Wayne gage location. Consequently, the Detroit River schematization appears to be appropriate for the flow model implementation and simulation; however, additional flow simulations are necessary to verify this assumption.

Summary

The branch-network flow model has been successfully used to simulate flow in singular reaches and in networks of interconnected open channels. The results of several applications illustrate the flexibility and accuracy of the flow model in simulating a wide range of flow conditions. The various model implementations were efficiently carried out using a computer program for analyzing channel cross-sectional geometry, a computerized system for editing, transcribing, storing, and retrieving time-dependent boundary-value data, and specific model-generated graphical outputs for evaluating computed results. These capabilities, which significantly hasten the model calibration and verification operations, also constitute an operational system for implementing and using the branch-network flow model.

The branch-network flow equations include wind shear on the water surface as a forcing function and are formulated to account for nonuniform velocity distributions through the momentum or Boussinesq coefficient. The four-point, finite-difference technique, with weighting factors for function values and their spatial derivatives in the flow equations, provides a high degree of flexibility in simulating diverse flow conditions in channels of variable cross-sectional properties. A unique branch-transformation technique is utilized in the model, resulting in a significant savings in computational time and computer storage. The implicit solution technique employed permits computations at large time steps. The subdivision of branches into segments of equal or unequal lengths is possible, thereby providing for the computation of water-surface elevations and flow discharges at any desired location.

References Cited

Amein, Michael, and Fang, C. S., 1970, Implicit flood routing in natural channels: American Society of Civil Engineers Proceedings, Journal of the Hydraulics Division, v. 96, no. HY12, p. 2481–2500.

Baltzer, R. A., and Lai, Chintu, 1968, Computer simulation of unsteady flows in waterways: American Society of Civil Engineers Proceedings, Journal of the Hydraulics Division, v. 94, no. HY4, p. 1083–1117.

Baltzer, R. A., and Shen, John, 1961, Computation of flows in tidal reaches by finite-difference technique: Proceedings of the First National Coastal and Shallow-Water Research Conference, The National Science Foundation and The Office of Naval Research, Tallahassee, Fla., p. 258–264.

Chow, V. T., 1959, Open-channel hydraulics: New York, McGraw-Hill, 680 p.

Dronkers, J. J., 1964, Tidal computations in rivers and coastal waters: Amsterdam, The Netherlands, North-Holland Publishing Co., 518 p.

———1969, Tidal computations for rivers, coastal areas, and seas: American Society of Civil Engineers Proceedings, Journal of the Hydraulics Division, v. 95, no. HY1, p. 29–77.

Fread, D. L., 1974, Numerical properties of implicit four-point finite-difference equations of unsteady flow: National Oceanic and Atmospheric Administration Technical Memorandum, NWS, HYDRO-18, 38 p.

Lai, Chintu, 1965a, Flows of homogeneous density in tidal reaches, solution by the method of characteristics: U.S. Geological Survey open-file report, 58 p.

———1965b, Flows of homogeneous density in tidal reaches, solution by the implicit method: U.S. Geological Survey open-file report, 43 p.

Neumann, Gerhard, and Pierson, W. J., Jr., 1966, Principles of physical oceanography: Englewood Cliffs, N. Y., Prentice-Hall, 545 p.

Oltman, R. N., 1979, Application of transient-flow model to the Sacramento River at Sacramento, Calif.: U.S. Geological Survey water-resources investigations 78–119, 23 p.

Preissmann, Alexander, 1960, Propagation des intumescences dans les canaux et riviers (Propagation of translatory waves in channels and rivers): First Congress of the French Association for Computation (1er Congrès d'association Française de calcul), Grenoble, France, p. 433–442.

Schaffranek, R. W., 1976, Some observations on the open-channel flow equations for turbulent surface-water bodies: George Washington University, Department of Civil Engineering, Masters Thesis, 67 p.

Schaffranek, R. W., and Baltzer, R. A., 1978, Fulfilling model time-dependent data requirements; Vol. III: Coastal Zone '78, Symposium on Technical, Environmental, Socioeconomic, and Regulatory Aspects of Coastal Zone Management: American Society of Civil Engineers, p. 2069–2084.

Strelkoff, Theodor, 1969, One-dimensional equations of open-channel flow: American Society of Civil Engineers Proceedings, Journal of the Hydraulics Division, v. 95, HY3, p. 861–876.

Yen, Ben Chie, 1973, Open-channel flow equations revisited: American Society of Civil Engineers Proceedings, Journal of the Engineering Mechanics Division, v. 91, no. EM5, p. 979–1009.

Wilson, B. W., 1960, Note on surface wind stress over water at low and high wind speeds: Journal of Geophysical Research, v. 65, no. 10, p. 3377–3382.

APPENDIXES I–IV

Appendix I, Program Control-Card Format

There are nine basic card types used for input to the branch-network flow model. The order of card input is illustrated in figure 15. The functional purpose of each card is given as follows:

Network-name card identifies the network being simulated.

Computation-control card defines the network dimensions, assigns the computation time increment, specifies the iteration and convergence criteria, signifies the choice of input/output units, assigns various constants and coefficients, and selects the type of output desired.

Branch-identity card identifies each branch by name and number and indicates the positive flow direction, as well as the number of cross sections to be input to define the channel segments and their geometry (one such card for each branch in the network).

Initial-condition cards (two cards for each of the cross sections in the identified branch) assign the segment lengths, water tempterature, flow-resistance coefficients, wind direction, and momentum coefficient, in addition to the initial values of stage and discharge.

Cross-sectional geometry cards constitute a set of data cards (preceded by one card identifying the number of data cards to input) defining the particular cross-sectional geometry relationships (one set for each cross section in the identified branch).

Nodal-flow card(s) assigns the external inflows (outflows, if negative) at each internal junction.

List-index card controls identification of data stored in the time-dependent data base, and thereby available as boundary-value data.

Boundary-value data cards consist of one card identifying the boundary-value data (required at each external junction) by type, station number, external junction number, recording frequency, and beginning and ending dates and times and are optionally followed by one card (containing functional boundary-condition coefficients) or by multiple cards (containing actual boundary-value data, if such data are to be input manually from cards).

Measured data cards consist of an initial card identifying the measured data (used for plotting versus computed results) by type, station number, junction or branch and cross-section numbers, recording frequency, and beginning and ending dates and times and are optionally followed by cards containing the measured values.

As indicated the first eight card types are required with measured data cards being optionally required. All available parameter defaults can be taken simply by having the appropriate card column(s) blank. If all parameters on a particular card have acceptable defaults, the defaults can be exercised by inserting a blank card. As is identified in the following table, both metric and inch-pound equivalent default parameter values are available.

Variable	Columns	Format	Default	Definition
Network-name card (one required per execution)				
NETNAM	1-80	20A4	blanks	Name of the network of open channels.
Computation-control card (one required per execution)				
IUNIT	1- 2	A2	EN	System of units of input data (EN: in/lbs; ME: metric).
NBCH	3- 4	I2	(None)	Total number of branches in the network (0<NBCH<16).
NJNC	5- 6	I2	(None)	Total number of junctions (both internal and external) in the network (1<NJNC<16).
NBND	7- 8	I2	(None)	Number of external boundary conditions, and internal station locations if any, to be user defined (1<NBND<6).
NSTEPS[1]	9-12	I4	--------	Number of time steps to be computed.
OUNIT	13-14	A2	EN	System of units of output results (EN: in/lbs; ME: metric).
LUGEOM	15-16	I2	5	Logical unit number of the device containing the cross-sectional geometry data (5: card reader; 10: other).
NIT[2]	17-18	I2	5	Maximum number of iterations permitted per time step (usually 3≤NIT≤5).
IOTOPT	19	I1	0	Output option (0: print results at every time step; 1: print results at every iteration; 2: print daily summary of results; 3: plot results at every time step; 4: print monthly flow-volume summaries).
IPLOPT[3]	20	I1	0	Plot option (0: do not plot; 1: plot computed discharge; 2: plot computed stage; 3: plot measured versus computed discharge; 4: plot measured versus computed stage).
IPLDEV[3 4]	21	I1	0	Plotter device (0: line printer; 1: Tektronix; 2: CalComp; 3: FR80).

See footnotes at end of table.

Variable	Columns	Format	Default	Definition
Computation-control card (one required per execution)—Continued				
IPRMSG	22	I1	0	Option to permit the time-dependent-data storage-and-retrieval system to print messages (0: do not print message; 1: print message).
IPLMSG	23	I1	0	Option to permit the plotter software to print messages (0: do not print messages; 1: print messages).
IEXOPT	24	I1	0	Option to extrapolate initial values for unknowns from present time step values (0: do not extrapolate; 1: extrapolate).
TYPETA	25	I1	1	Type of functional flow-resistance relationship, (1: constant; 2: temperature; 3: depth; 4: discharge; 5: Froude number; 6: Reynolds number).
INHR[5]	26-27	I2	________	Hour of initial-value data.
INMN[5]	28-29	I2	________	Minute of initial-value data.
IDTM[6]	30-33	I4	________	Simulation time increment in minutes.
THETA	34-36	F3.2	1.0	Finite-difference weighting factor (θ) for the spatial derivatives (usually $0.6 \leq \text{THETA} \leq 1.0$)
QQTOL[7]	37-41	F5.1	________	Discharge convergence criterion.
ZZTOL	42-46	F5.3	0.01/0.003048	Stage convergence criterion in feet or meters.
WSPEED	47-51	F5.2	0.0	Wind speed in miles or kilometers per hour.
WSDRAG	52-56	F5.4	0.0026	Water-surface drag coefficient.
H2ODEN	57-61	F5.4	1.9617/1.011	Water density in slugs/ft^3 or g/cm^3.
CHI[8]	62-64	F3.2	1.0	Weighting factor (χ) for function values in the flow equations (usually $0.5 \leq \text{CHI} \leq 1.0$).
IPUNIN	65	I1	0	Option to punch out initial condition cards at the end of the simulation (0: do not punch; 1: punch).
Branch-identification cards (one required per branch)				
IJF	1- 2	I2	(None)	Junction number identifying the source of positive flow for the branch ($0 < \text{IJF} \leq \text{NJNC}$).
IJT	3- 4	I2	(None)	Junction number identifying the outlet of positive flow for the branch ($0 < \text{IJT} \leq \text{NJNC}$).
NSEC[9]	5- 6	I2	(None)	Number of cross sections input to define the geometry of the branch.
NAME	7-46	10A4	Blanks	Name of branch.
Initial-condition cards (two required per cross section)				
First initial-condition card for cross section:				
Z[10]	1-10	F10.3	(None)	Initial stage value.
Q	11-20	F10.3	(None)	Initial discharge value.
DX	31-40	F10.2	(None)	Segment length.
T	41-50	F10.2	59.0/15.0	Water temperature, in degrees Fahrenheit or Celsius.
RN	51-80	3E10.4	(None)	Coefficients of flow-resistance relationship, i.e., $\eta(x) = \text{RN}(1) + \text{RN}(2)*x + \text{RN}(3)*x**2$.
Second initial-condition card for cross section:				
WANGLE	1-10	F10.3	0.0	Wind direction measured from the positive x-axis which lies along the centerline of the channel.
BETVEL	11-20	F10.3	1.0	Momentum coefficient.

See footnotes at end of table

Variable	Columns	Format	Default	Definition
Cross-sectional geometry cards (one set required per cross section)				
First card of cross-sectional geometry identifies the number of input data cards:				
IPT	1- 2	I2	(None)	Number of cross-sectional geometry data cards ($1 < IPT \leq 20$).
IPT number of cross-sectional geometry data cards:				
ZA[11]	1-10	F10.3	(None)	Stage at which corresponding area and top width were measured.
AA	11-20	F10.3	(None)	Cross-sectional area at specified stage.
BB	21-30	F10.3	(None)	Top width of cross section at specified stage.
Nodal-flow card(s) (one value per junction; 10 junctions per card)				
W[12]	1-80	10F8.2	0.0	External inflow (or outflow) at junction (constant nodal flow for duration of simulation assumed).
List-index card for time-dependent data base (one required per execution)				
DATYPE[13]	1- 4	I4	3330	Type of magnetic disk used to hold time-dependent data base (usually 2314 or 3330).
LISTB	38-39	I2	0	Option to list the time-dependent data base index before computation (1: print only the directory list; −1: print the directory list and the chronological summary; 0: do not print).
LISTA	46-47	I2	0	Option to list the time-dependent data base index after computation (1: print only the directory list; −1: print the directory list and the chronological summary; 0: do not print).
Boundary-value data cards (one set required per external boundary condition)				
First card of each boundary-value data set is a data-definition card:				
ITYPE[14]	1- 2	A2	' Z'	Type of boundary-value data specified (' Z': stage, ' Q': discharge).
IBJNC	3- 4	I2	(None)	Junction number of external boundary location ($0 < IBJNC \leq NJNC$).
NDATA	5- 7	I3	0	Number of boundary-value data input (0: implies data are to be retrieved from the time-dependent data base; 1: boundary condition is specified by an equation; >1: identifies the number of boundary-value data cards to be read).
DTT[15]	8- 9	F2.0	(None)	Recording interval of boundary-value data in minutes.
ISTATN	10-17	I8	(None)	Station identification number of boundary-value data specified.
ITIME	25-39	5(I2,1X)	(None)	Beginning data and time of boundary-value data (YR/MO/DY HR:MN).
NTIME	45-59	5(I2,1X)	(None)	Ending data and time of boundary-value data (YR/MO/DY HR:MN).
IDREAD[15]	62-65	I4	(None)	Number of boundary-value data recorded per day.
DATUM[16]	66-72	F7.3	0.0	Datum correction for stage boundary-value data.
IDONLY[17]	80	I1	(None)	Flag to indicate whether or not the boundary-value data-definition card is input to describe boundary-value data or only to identify station information (0: implies inclusion for data input; 1: implies inclusion for station identification only).
NDATA number of boundary-value data cards if data are input via cards:				
ZQ	1-10	F10.3	(None)	Stage or discharge boundary value.
One card containing coefficients if boundary condition is specified by an equation:				
ZQBVCO	1-40	4E10.4	0.0	Coefficients of the boundary-value equation, i.e., $Z(Q) = ZQBVCO(1) + ZQBVCO(2)*Q + ZQBVCO(3)*Q**2 + ZQBVCO(4)*Q**3$.

See footnotes at end of table

Variable	Columns	Format	Default	Definition
Measured-data cards (one set optionally required when plotting computed versus measured data):				
First card of each measured-data set is a data-definition card:				
MTYPE[18]	1- 2	A2	' Z'	Type of measured data supplied (' Z': stage; ' Q': discharge).
MJNC[19]	3- 4	I2	--------	Junction number of measured data location (0<MJNC ≤NJNC).
MDATA	5- 7	I3	0	Number of measured data input (0: implies data are to be retrieved from the time-dependent data base; >1: identifies the number of measured-data cards to be read).
CDTT[15]	8- 9	F2.0	(None)	Input interval of measured data in minutes.
MSTATN	10-17	I8	(None)	Station identification number of measured data specified.
MITIME[20]	25-39	5(I2,1X)	(None)	Beginning date and time of measured data (YR/MO/DY HR:MN).
MNTIME[20]	45-59	5(I2,1X)	(None)	Ending date and time of measured data (YR/MO/DY HR:MN).
MDREAD[15 21]	62-65	I4	(None)	Number of measured data input per day.
CDATUM	66-72	F7.3	0.0	Adjustment factor for measured data.
MBCH[19]	78-79	I2	--------	Branch number of measured data location (0<MBCH ≤NBCH).
MSEC[19]	80	I1	--------	Cross-section number of measured data location (0<MSEC≤NSEC).
MDATA number of measured-data cards if data are input via cards:				
ZQMEAS	1-10	F10.3	(None)	Measured stage or discharge value.

[1] If not specified, the number of time steps to be computed is determined from the time span specified on the *first* boundary-value data definition card.

[2] The computation is permitted to continue using the previous computed values whenever the maximum number of iterations is exceeded A message is printed, however, identifying the maximum stage and discharge deviations and the location(s) of their occurrence.

[3] These variables are only applicable for IOTOPT=3

[4] Tektronix, CalComp, and FR80 plots are produced in auxiliary operations from files of plotter instructions generated during the simulation.

[5] If not specified, the time of initial-value data is taken as the time of the first boundary-value datum.

[6] If not specified, the simulation time step is set to the data recording interval on the *first* boundary-value data definition card

[7] The default discharge convergence criterion is taken as 0 5 percent of the minimum (absolute value greater than zero) initial-value discharge. If all initial discharges are zero the default discharge convergence criterion is set to one

[8] If not specified, the weighting factor χ is set equal to the weighting factor for the spatial derivatives, θ.

[9] The total number of cross sections used to define the geometry of all branches composing the network must not exceed the maximum number of cross sections allocated (NBSEC ≥ Σ NSEC(I); I = 1,NBCH) for the particular version of the model program (see section Program restrictions). In general, it is recommended not to exceed the maximum number of cross sections allocated per branch, which is 4 in this model-program version

[10] Initial values at external boundary locations default to the first boundary-value datum input

[11] Stage-area-width relationships must be input in sequence starting with the values at the lowest stage.

[12] Code nodal-flow values in sequence according to the junction numbering scheme.

[13] Other direct-access devices can be accommodated as required.

[14] If boundary-value data sets are input from both disk and cards, put disk boundary-value data definition cards first beginning with the boundary-value data recorded at the greatest frequency (smallest time interval.)

[15] The data interval and the number of data per day need not both be specified, either is sufficient.

[16] Appropriate uses of the DATUM adjustment factor are to change datum references or to correct for known or suspected recorder elevation shifts.

[17] The IDONLY flag permits the accumulation and compilation of flow volumes at internal station locations of the network. The station identification number must be provided to accommodate filing flow volumes at a particular location.

[18] Only one set of measured data can be input per branch of the network.

[19] The location of measured data may be defined either by junction number or by branch and cross-sectional numbers.

[20] All sets of measured data must begin and end at a common date and time in the same calendar day. This data and time must be within the time span of the simulation.

[21] All measured data must be supplied at the computation time step frequency

Appendix II, Definition of MAIN Program Variables and Arrays

The ability to relate program variables and arrays to the mathematical formulation of the flow equations may be necessary or desirable on occasion. The following table defining the program variables and arrays in the MAIN program is provided for this purpose. It may also be useful if it is necessary to modify the program to accommodate large network systems or other unique flow conditions. Variables and arrays used similarly in the subprograms of the model are also defined accordingly. However, no commonality of definitions is intended or should be inferred between the model source code, as presented herein, and the time-dependent-data storage-and-retrieval or the graphical display software systems as utilized.

Array (size) or Variable	Definition
A(60)	cross-sectional area at present time step.
AA(20,60)	cross-sectional area array of stage-area-width geometry tables.
AM(3600)	coefficient matrix of unknowns.
AP(60)	cross-sectional area at future time step.
AAVG	four-point weighted-average, cross-sectional area.
AQMAX(60)	cross-sectional area at time of maximum discharge for the day.
AQMIN(60)	cross-sectional area at time of minimum discharge for the day.
AAVGCU	cube of four-point weighted-average, cross-sectional area.
AAVGSQ	square of four-point weighted-average, cross-sectional area.
AIRDEN	air density, used to calculate the wind-resistance coefficient.
B(60)	cross-sectional top width at present time step.
BB(20,60)	cross-sectional top width array of stage-area-width geometry tables.
BP(60)	cross-sectional top width at future time step.
BU(30)	branch transformation vector.
BMX(60)	right-hand-side vector of unknowns.
BUU(60)	branch transformation matrix.
BAVG	four-point weighted-average, cross-sectional top width.
BIGQ	maximum difference in computed discharges over the time step.
BIGZ	maximum difference in computed stages over the time step.
BETCOR	average momentum coefficient for the segment.
BETVEL(60)	momentum coefficient for the cross section.
BRNAME(10,15)	name of branches in the network.
CN	conversion factor for the flow-resistance function.
CW	factor of wind forcing function.
C1	temporary branch transformation coefficient.
C2	Do.
C3	Do.
C4	Do.
CHI	finite-difference weighting factor for function values in the equation of motion.
CDTT	data recording frequency for measured data.
CDATUM	temporary variable used as adjustment factor for time-dependent data.
DC	units of temperature data.
DT	computational time step in seconds.
DX(60)	branch-segment length.
DET	inverse of coefficient matrix.
DTT(5)	data recording frequency for boundary-value data.
DCHI	form of the finite-difference weighting factor for function values.
DXIJ	length of the *J*th segment of the *I*th branch.
DATUM(5)	adjustment factor for stage boundary-value data.
DELTA	matrix coefficient.
DPERM(12)	number of days per month.
DTPRT	logical variable controlling printout at time step.
DTYPE	boundary-value data type.
DAYSUM	logical variable controlling daily summary printout.
DTHETA	form of the finite-difference weighting factor for spatial derivatives.
DTZERO	flow-volume interpolation variable.
EN	units identifier for inch-pound system.
ERROR	logical variable signalling an error in the initial values.
EPSLON	matrix coefficient.
FOUND	logical variable identifying missing initial values.
G	gravitational acceleration.
GAMMA	matrix coefficient.
GINDEX	logical unit variable for data-station reference file.
H2ODEN	water density used to calculate the wind-resistance coefficient.

Array (size) or Variable	Definition
I	DO-loop variable most frequently used as branch index.
II	total number of equations for the network.
IJ	cross-section index.
IS	flag signalling a singular matrix.
I2	branch transformation index.
I4	Do.
IAR	statement function for coefficient-matrix addressing.
ICT(15)	counter for number of branches at each junction.
IDA	beginning day of boundary-value data.
IDX(15,15)	list of branches at each junction.
IHR	beginning hour of boundary-value data.
IJF(15)	junction identifying flow source of branch.
IJT(15)	junction identifying flow outlet of branch.
IJ2	segment transformation index.
IJ4	Do.
IMN	beginning minute of boundary-value data.
IMO	beginning month of boundary-value data.
IPT(60)	number of stage-area-width relationships for cross section.
IYR	beginning year of boundary-value data.
IBCH	branch number.
IBLK	test variable for default, boundary-value data type.
ICHK	flag signalling matrix solver to check for maximum pivots.
IDTM	computation time step in minutes.
IISQ	square of the number of equations to be solved.
IJP1	cross-section index.
INHR	initial hour of simulation.
INMN	initial minute of simulation.
IREM	temporary variable used to hold the remainder in various arithmetic operations.
I2P1	branch transformation index.
I2P2	Do.
I4P1	Do.
I4P2	Do.
I4P3	Do.
I4P4	Do.
IBIGQ	branch with maximum difference in computed discharges.
IBIGZ	branch with maximum difference in computed stages.
IBJNC(5)	junction number of boundary-value data location.
IDETA(6)	letter indicating the type of "η" relationship specified.
IFVOL(8,31,5)	accumulated flow volume.
IJVKT(5)	number of flow reversals within the day.
IJVOL(5)	cross section at which flow volumes are accumulated.
IJ2P1	segment transformation index.
IJ2P2	Do.
IJ4P1	Do.

Array (size) or Variable	Definition
IJ4P2	Do.
IJ4P3	Do.
IJ4P4	Do.
INTER	data recording frequency.
ITVOL(8,31,5)	time of flow reversal.
ITYPE(5)	boundary-value data type.
IUNIT	units of measure of input data.
IDONLY(5)	flag indicating data definition is for station identification only.
IDTPDY	number of time steps per day.
IDTYPE	type of disk containing time-dependent data base.
IETIME	elapsed minutes in the calendar year to the beginning of boundary-value data.
IETIYR	total elapsed minutes in calendar year of boundary-value data.
IETJYR	total elapsed minutes in next consecutive calendar year of boundary-value data.
IEXOPT	option to extrapolate unknowns.
IITIME	time of first boundary-value data.
INDATA(360)	array of data retrieved from time-dependent data base.
IOTOPT	output option.
IPLDEV	type of device used for plotting.
IPLMSG	flag controlling the printout of messages generated by the plotter software.
IPLOPT	plot output option.
IPRMSG	flag controlling the printout of messages generated by the time-dependent-data storage-and-retrieval routine.
IPUNIN	option to punch initial condition cards.
IQDATA(360)	array of discharge boundary-value data.
IRDPDY	readings per day of boundary-value data.
ISTATN(5)	station identification number of boundary-value data.
ITQMAX(60)	time of maximum discharge for the day.
ITQMIN(60)	time of minimum discharge for the day.
IZDATA(720)	array of stage boundary-value data.
IZQBVE(5)	flag signalling that boundary condition is to be specified by an equation.
J	DO-loop variable used as segment, cross-section, and junction index.
JDA	beginning day of partial boundary-value data retrieval.
JHR	beginning hour of partial boundary-value data retrieval.
JMN	beginning minute of partial boundary-value data retrieval.
JMO	beginning month of partial boundary-value data retrieval.
JP1	segment index.
JYR	beginning year of partial boundary-value data retrieval.
JBIGQ	cross section with maximum difference in computed discharges.
JBIGZ	cross section with maximum difference in computed stages.
JDAYN	Julian day number.

Array (size) or Variable	Definition
JETIME	elapsed minutes in the calendar year to the beginning of boundary-value data retrieved.
JITIME	time of first boundary-value data retrieved.
K	DO-loop variable used for various indexing.
KT	time-step counter.
KDA	day at current time step.
KHR	hour at current time step.
KMN	minute at current time step.
KMO	month at current time step.
KYR	year at current time step.
KETIME	elapsed minutes in the calendar year to current time step.
KTMATS	matrix solution counter.
KTMEAS	measured data set counter.
L	DO-loop variable used as boundary-value and measured data index.
LASTN	iterations required for last time step.
LISTA	option to list time-dependent data base index after simulation.
LISTB	option to list time-dependent data base index before simulation.
LAMBDA	matrix coefficient.
LEAPDY	leap-day indicator.
LETIME	elapsed minutes in the calendar year to time of last plot.
LUGEOM	logical unit variable for cross-sectional geometry data file.
M	DO-loop variable for time step.
ME	units identifier for metric system.
MM	coefficient matrix index.
MT	units of metric data.
MU	matrix coefficient.
M0	coefficient matrix index.
MDA	ending day of partial boundary-value data retrieval.
MDT	data recording frequency for measured data.
MHR	ending hour of partial boundary-value data retrieval.
MMN	ending minute of partial boundary-value data retrieval.
MMO	ending month of partial boundary-value data retrieval.
MYR	ending year of partial boundary-value data retrieval.
MAXS	maximum number of cross sections accommodated in the network.
MBCH(5)	branch identifying measured data location.
MIDA	beginning day of measured data.
MIHR	beginning hour of measured data.
MIMN	beginning minute of measured data.
MIMO	beginning month of measured data.
MIYR	beginning year of measured data.

Array (size) or Variable	Definition
MJNC	junction identifying measured data location.
MKDA	ending day of measured data.
MKHR	ending hour of measured data.
MKMN	ending minute of measured data.
MKMO	ending month of measured data.
MKYR	ending year of measured data.
MSEC(5)	cross section identifying measured data location.
MXBH	maximum number of branches accommodated in the network.
MXBY	maximum number of external boundary and flow-volume locations accommodated in the network.
MXJN	maximum number of junctions accommodated in the network.
MXMD	maximum number of measured data locations accommodated in the network.
MXPT	maximum number of stage-area-width relationships accommodated per cross section.
MAXBD	maximum number of boundary-value data accommodated per retrieval.
MDATA(5)	number of measured data input.
MTYPE(5)	measured data type.
MAXCZQ	maximum number of computed results per day.
MAXMZQ	maximum number of measured data accommodated.
MAXQBD	maximum number of discharge boundary-value data accommodated per retrieval.
MAXZBD	maximum number of stage boundary-value data accommodated per retrieval.
MDREAD	readings per day of measured data.
MEITIM	elapsed minutes in the calendar year to the beginning of measured data.
MEKTIM	elapsed minutes in the calendar year to the end of measured data.
METIME	elapsed minutes in the calendar year to the end of boundary-value data retrieved.
MITIME	time of last boundary-value data retrieved.
MOREBD	logical variable signalling the need to retrieve additional boundary-value data.
MSTATN(5)	station identification number of measured data.
N	DO-loop variable for iteration.
ND	number of data.
NN	coefficient matrix index.
NS	number of cross sections.
NDA	ending day of boundary-value data.
NHR	ending hour of boundary-value data.
NIT	number of iterations permitted per time step.
NMN	ending minute of boundary-value data.
NMO	ending month of boundary-value data.
NNN	coefficient matrix index.
NYR	ending year of boundary-value data.

Array (size) or Variable	Definition
NBCH	number of branches in the network.
NBND	number of external boundary condition and flow-volume locations in the network.
NBPJ	number of branches joining at a junction.
NJNC	number of junctions in the network.
NSEC(15)	number of cross sections in the branch.
NSM1	number of segments in a branch.
NDATA(5)	number of boundary-value data input.
NDFIRT	total number of boundary-value data to be retrieved.
NDPART	number of data in partial boundary-value data retrieval.
NETIME	elapsed minutes in the calendar year to the end of boundary-value data.
NETNAM(20)	name of network.
NITIME	time of last boundary-value data.
NOCONV	logical variable signalling conversion of units.
NOEXTP	logical variable controlling extrapolation.
NOPRIT	logical variables controlling printout.
NSTEPS	total number of time steps to be computed.
OMEGA	matrix coefficient.
OUNIT	units of measure of output results.
ONECHI	form of the geometry finite-difference weighting factor.
OPLOTS	logical variable controlling plot generation.
PSI	matrix coefficient.
PTPLT	logical variable controlling printer-plot generation.
PUNCH	logical unit variable for card punch.
PRINTR	logical unit variable for line printer.
PRTMSG	logical variable controlling the printout of messages generated by the time-dependent-data storage-and-retrieval system.
Q(60)	discharge at present time step.
QP(60)	discharge at future time step.
QIJ	discharge at the *J*th cross section of the *I*th branch at present time step.
QAVG	four-point weighted-average discharge.
QMAX	maximum discharge for the day.
QMIN	minimum discharge for the day.
QSUM	cumulative discharge for the day.
QTOL	discharge difference for tolerance check.
QIJP1	discharge at the $J+1$st cross section of the *I*th branch at the present time step.
QQTOL	discharge convergence criterion.
QTEMP	temporary discharge variable.
QTYPE	code identifying discharge data.
QZCONV	discharge or stage conversion factor.
R(60)	hydraulic radius at present time step.
RN(4,60)	coefficients of flow-resistance equation.
RP(60)	hydraulic radius at future time step.
ROW(60)	pointers to rows containing maximum pivot elements.
RAVG	four-point weighted-average hydraulic radius.
RNIJ	flow-resistance coefficient of the *J*th segment of the *I*th branch.
READER	logical unit variable for card reader.
RTCODE	error code returned from time-dependent data storage-and-retrieval routine.
SIGMA	matrix coefficient.
STRIP	option to strip error codes from data retrieved from time-dependent data base.
STAGES	logical variable signalling the plotting of stages.
T(60)	water temperature.
TH	factor of parabolic interpolation for boundary-value data.
TWOG	twice the gravitational acceleration.
THETA	finite-difference weighting factor for the spatial derivatives.
THPSI	flow equation factor.
TUNIT	units identifier for temperature data.
TDDATA	logical unit variable for the time-dependent data base.
TWOCSQ	twice the square of the conversion factor for the flow-resistance function.
TYPETA	option identifying the type of flow-resistance relationship specified.
U(120)	segment transformation vector.
UU(240)	segment transformation matrix.
UNIT	units identifier for initial-value data.
UUIJP1	temporary variable used in branch transformation computation.
UUIJP2	Do.
UUIJP3	Do.
UUIJP4	Do.
W(15)	nodal flow at junction.
WANGLE(60)	angle of wind direction with respect to positive flow direction.
WSDRAG	water-surface drag coefficient.
WSPEED	wind speed.
XSKT(15)	cross-section counter.
Z(60)	stage at present time step.
ZA(20,60)	stage array of stage-area-width geometry tables.
ZP(60)	stage at future time step.
ZQ(720,5)	stage and (or) discharge boundary-value data.
ZIJ	stage at the *J*th cross section of the *I*th branch at present time step.
ZETA	matrix coefficient.
ZTOL	stage difference for tolerance check.

Array (size) or Variable	Definition
ZIJP1	stage at the $J+1$st cross section of the Ith branch at present time step.
ZQMAX(60)	stage at time of maximum discharge for the day.
ZQMIN(60)	stage at time of minimum discharge for the day.
ZQPIJ	stage or discharge at the Jth cross section of the Ith branch at future time step.
ZTEMP	temporary stage variable.
ZTMAX	maximum stage specified in stage-area-width geometry tables.
ZTMIN	minimum stage specified in stage-area-width geometry tables.
ZTYPE	code identifying stage data.
ZZTOL	stage convergence criterion.
ZDATUM	stage computation datum.
ZQBVCO(4,5)	coefficients of stage-discharge rating curves.
ZQCOMP (288,60)	computed stages or discharges for the day.
ZQMEAS (192,5)	measured stage or discharge data.

Appendix III, Adjustable Arrays

Object-time dimensioning of arrays is utilized in the branch-network flow model. This technique facilitates the expansion of arrays to accommodate networks with unique dimension requirements. This table identifies those arrays whose dimensions may require modification dependent upon the characteristics of the network being simulated. Because object-time dimensioning is employed, a change in the dimension of an array is directly accomplished by declaring its new dimension in the MAIN program only, with no modifications required in the subroutines. To facilitate the expansion of arrays, the following table identifies the variables controlling the dimensions, the current (default) dimensions, and the array type. Knowing the variables controlling the array dimensions and the array type it is a simple matter to expand the array capacities and to compute the model's new machine storage requirements. Dimension variables are defined in the table footnotes.

Array	Type	Variable dimension	Current dimension
A	REAL*4	(NBSEC)[1]	(60)
AP	do	do	(60)
AQMAX	do	do	(60)
AQMIN	do	do	(60)
B	do	do	(60)
BP	do	do	(60)
BETVEL	do	do	(60)
DX	do	do	(60)
IPT	INTEGER*2	do	(60)
ITQMAX	do	do	(60)
ITQMIN	do	do	(60)
Q	REAL*4	do	(60)
QP	do	do	(60)
QMAX	do	do	(60)
QMIN	do	do	(60)
QSUM	do	do	(60)
R	do	do	(60)
RP	do	do	(60)
T	do	do	(60)
WANGLE	do	do	(60)
Z	do	do	(60)
ZP	do	do	(60)
ZQMAX	do	do	(60)
ZQMIN	do	do	(60)
U	REAL*8	(2*NBSEC)	(120)
UU	do	(4*NBSEC)	(240)
RN	REAL*4	do	(4,60)
AA	do	(MXPT,NBSEC)[2]	(20,60)
BB	do	do	(20,60)
ZA	do	do	(20,60)
ZQCOMP	do	(MAXCZQ,NBSEC)[3]	(288,60)
IJF	INTEGER*2	(MXBH)[4]	(15)
IJT	do	do	(15)
NSEC	do	do	(15)
XSKT	do	do	(15)
BRNAME	INTEGER*4	(10,MXBH)	(10,15)
AM	REAL*4	((4*MXBH)**2)	(3600)
BU	REAL*8	(2*MXBH)	(30)
BMX	REAL*4	(4*MXBH)	(60)
BUU	REAL*8	do	(60)
ROW	INTEGER*2	do	(60)
W	REAL*4	(MXJN)[5]	(15)
ICT	INTEGER*2	do	(15)
IDX	do	(MXJN,MXBH)	(15,15)
DTT	REAL*4	(MXBY)[6]	(5)
DATUM	do	do	(5)
IBJNC	INTEGER*2	do	(5)
ITYPE	do	do	(5)
NDATA	do	do	(5)
ISTATN	INTEGER*4	do	(5)
IZQBVE	INTEGER*2	do	(5)
ZQBVCO	REAL*4	(4,MXBY)	(4,5)
ZQ	do	(MAXZBD,MXBY)[7]	(720,5)
IZDATA	INTEGER*2	(MAXZBD)	(720)
INDATA	INTEGER*4	(MAXZBD/2)	(360)
IQDATA	do	do	(360)
MBCH	INTEGER*2	(MXMD)[8]	(5)
MSEC	do	do	(5)
MDATA	do	do	(5)
MTYPE	do	do	(5)
MSTATN	INTEGER*4	do	(5)
ZQMEAS	REAL*4	(MAXMZQ,MXMD)[9]	(288,5)

[1] NBSEC is the total number of cross sections used to define the channel geometry of the network. (Computed results are produced at these locations.)

[2] MXPT is the maximum number of stage-area-width relationships used to define the channel geometry at a given cross section.

[3] MAXCZQ is the maximum number of daily computed results held in storage for plotting purposes.

[4] MXBH is the maximum number of branches accommodated within the network.

[5] MXJN is the maximum number of junctions accommodated within the network.

[6] MXBY is the maximum number of external boundary locations and internal flow-volume locations accommodated within the network.

[7] MAXZBD is the maximum number of boundary-value data held in storage for computation purposes. (The boundary-value data arrays are automatically refreshed with data from the time-dependent data base as required during the simulation.)

[8] MXMD is the maximum number of measured data locations accommodated within the network.

[9] MAXMZQ is the maximum number of measured data held in storage for plotting purposes.

Appendix IV, FORTRAN IV Program Listing

```
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #    BRANCH-NETWORK FLOW MODEL USING A LINEAR-IMPLICIT TECHNIQUE     #
C #                          VERSION 79/04/19                          #
C #      BY R. W. SCHAFFRANEK, R. A. BALTZER, AND D. E. GOLDBERG       #
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #                                                                   #
C #   THIS PROGRAM CALCULATES TRANSIENT FLOW IN A NETWORK OF          #
C #   INTERCONNECTED OPEN CHANNELS.  TIME DERIVATIVES ARE APPROXIMATED#
C #   AS CENTERED IN SPACE AND TIME; SPATIAL DERIVATIVES ARE TREATED  #
C #   AS CENTERED IN SPACE AND WEIGHTED IN TIME ACCORDING TO A USER   #
C #   DEFINED WEIGHTING FACTOR.  A LINEAR MATRIX SOLUTION IS EFFECTED #
C #   WITH ITERATIVE IMPROVEMENT OF RESULTS OPTIONALLY SPECIFIABLE.   #
C #   THE 4*N BY 4*N MATRIX (N IS THE NUMBER OF BRANCHES) IS SOLVED   #
C #   BY GAUSS ELIMINATION USING MAXIMUM PIVOT STRATEGY.              #
C #                                                                   #
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #                          ARRAY DIMENSIONS                          #
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #                                                                   #
C #   MAXIMUM BRANCHES:   MXBH=15      MAXIMUM JUNCTIONS:   MXJN=15   #
C #   X-SECTIONS PER BRANCH: NSEC=4    SEGMENTS PER BRANCH: NSEG=3    #
C #   MAX CROSS SECTIONS:   NBSEC=60   MAX POINTS PER X-SECT:  MXPT=20#
C #   MAX BOUNDARY LOCATIONS: MXBY=5   MAX MEASURED LOCATIONS: MXMD=5 #
C #   MAXIMUM Z(T) B.V.D : MAXZBD=720  MAXIMUM Q(T) B.V.D.: MAXQBD=360#
C #   MAX COMPUTED PER DAY: MAXCZQ=288 MAX MEASURED DATA: MAXMZQ=288  #
C #                                                                   #
C #   DIMENSION MAXQBD ONE-HALF OF MAXZBD                             #
C #                                                                   #
C #   BRNAME(10,MXBH),IJF(MXBH),IJT(MXBH),NSEC(MXBH),XSKT(MXBH),      #
C #   BU(2*MXBH),BUU(4*MXBH),BMX(4*MXBH),AM((4*MXBH)**2),ROW(4*MXBH), #
C #   QMAX(NBSEC),QMIN(NBSEC),                                        #
C #   QSUM(NBSEC),IPT(NBSEC),ZQMIN(NBSEC),AQMIN(NBSEC)                #
C #   ZQMAX(NBSEC),AQMAX(NBSEC),ITQMAX(NBSEC),ITQMIN(NBSEC),          #
C #   Z(NBSEC),Q(NBSEC),A(NBSEC),B(NBSEC),R(NBSEC)                    #
C #   ZP(NBSEC),QP(NBSEC),AP(NBSEC),BP(NBSEC),RP(NBSEC),              #
C #   WANGLE(NBSEC),BETVEL(NBSEC),ZA(MXPT,NBSEC),ZQCOMP(MAXCZQ,NBSEC),#
C #   AA(MXPT,NBSEC),BB(MXPT,NBSEC),IDX(MXJN,MXBH),DATUM(MXBY),       #
C #   ICT(MXJN),W(MXJN),U(2*NBSEC),UU(4*NBSEC),T(NBSEC),DX(NBSEC),    #
C #   RN(4,NBSEC),IJVOL(MXBY),IDONLY(MXBY),IJVKT(MXBY),ISTAPR(MXBY),  #
C #   IFVOL(8,31,MXBY),ITVOL(8,31,MXBY),FVSTAT(31,MXBY),ISTATN(MXBY), #
C #   ITYPE(MXBY),IBJNC(MXBY),NDATA(MXBY),DTT(MXBY),ZQ(MAXZBD,MXBY),  #
C #   MSTATN(MXMD),MTYPE(MXMD),MSEC(MXMD),MDATA(MXMD),MBCH(MXMD),     #
C #   ZQMEAS(MAXMZQ,MXMD),IZDATA(MAXZBD),IQDATA(MAXQBD),              #
C #   INDATA(MAXZBD/2),ZQBVCO(4,MXBY),IZQBVE(MXBY)                    #
C #                                                                   #
C #   SET MAXIMUM RECORD SIZE OF DEFILE FILE 50 STATEMENT OF OUT      #
C #   SUBROUTINE TO 52 TIMES MXBY.                                    #
C #                                                                   #
      REAL ZQCOMP(288,60),ZQ(720,5),ZQMEAS(288,5)
      REAL QMAX(60),QMIN(60),QSUM(60),ZQMIN(60),AQMIN(60),ZQMAX(60),AQMA
     1X(60)
      REAL A(60),B(60),Z(60),Q(60),DX(60),T(60),RN(4,60),ZP(60),QP(60),A
     1P(60),BP(60),RP(60),WANGLE(60),BETVEL(60),R(60),ZA(20,60),AA(20,60
     2),BB(20,60)
      REAL W(15),DTT(5),DATUM(5),ZQBVCO(4,5)
      REAL LAMBDA,MU
      REAL AM(3600),BMX(60)
      REAL *8C1,C2,C3,C4,UUIJP1,UUIJP2,UUIJP3,UUIJP4,U(120),UU(240),BU(3
     10),BUU(60)
      INTEGER BRNAME(10,15),ISTATN(5),MSTATN(5)
```

```
      INTEGER *2IJF(15),IJT(15),NSEC(15),IPT(60),ITYPE(5),IBJNC(5),NDATA
     1(5),IDX(15,15),ICT(15),XSKT(15),MTYPE(5),MSEC(5),MDATA(5),MBCH(5),
     2IZQBVE(5)/5*0/,ITQMIN(60),ITQMAX(60),ROW(60)
      INTEGER IFVOL(8,31,5),FVSTAT(31,5),ISTAPR(5)
      INTEGER*2 IJVOL(5),IDONLY(5),IJVKT(5),ITVOL(8,31,5)
      INTEGER *2IZDATA(720)
      INTEGER IQDATA(360),INDATA(360)
      EQUIVALENCE (INDATA(1),IQDATA(1),IZDATA(1))
      INTEGER READER/ 5/,PRINTR/ 6/,PUNCH/ 7/,GINDEX/ 8/,TDDATA/98/
      INTEGER *2IRDPDY,IYR,IMO,IDA,IHR,IMN,NYR,NMO,NDA,NHR,NMN,LISTB,LIS
     1TA,MDREAD,MIYR,MIMO,MIDA,MIHR,MIMN,MKYR,MKMO,MKDA,MKHR,MKMN
      INTEGER *2JYR,JMO,JDA,JHR,JMN,MYR,MMO,MDA,MHR,MMN
      INTEGER *2RTCODE/0/,DTYPE/' Z'/,STRIP/-1/,ZTYPE/' Z'/,QTYPE/' Q'/
      INTEGER *2IDETA(6)/'  ','T ','D ','Q ','F ','R '/,TYPETA
      INTEGER *2EN/'EN'/,ME/'ME'/,IUNIT,OUNIT,UNIT/'FT'/,MT/' M'/,TUNIT/
     1' F'/,DC/' C'/,IBLK/'  '/
      INTEGER *2DPERM(12)/31,28,31,30,31,30,31,31,30,31,30,31/
      LOGICAL PRTMSG/.FALSE./,NOCONV/.TRUE./,ERROR/.FALSE./,OPLOTS,FOUND
     1,STAGES,NOEXTP,NOPRIT,DAYSUM/.FALSE./,MOREBD/.FALSE./,DTPRT,PTPLT
      COMMON /DATIME/ KYR,KMO,KDA,KHR,KMN,IDTM,M,NSTEPS,INHR,INMN,IDTPDY
     1,LASTN
      COMMON /OUTPUT/ NETNAM(20),NBCH,NBND,IOTOPT,IPLOPT,IPLDEV,STAGES,Z
     1DATUM,IUNIT,OUNIT
      COMMON /MEDATA/ MDT,KTMEAS,LETIME,KETIME,MEITIM,MEKTIM,IPLMSG,MIYR
     1,MIMO,MIDA,MIHR,MIMN
      DATA MXPT/20/,MAXCZQ/288/,MAXS/60/,MAXQBD/360/,MAXZBD/720/,MXBH/15
     1/,MXJN/15/,MXBY/5/,MXMD/5/,MAXMZQ/288/
      DATA AIRDEN/0.002509/,QZCONV/1.0/,ZTMIN/9999999./,ZTMAX/-9999999./
C     STATEMENT FUNCTION FOR LOCATING ELEMENTS IN COEFFICIENT MATRIX
      IAR(I,J)=I+II*(J-1)
C
C     READ PROGRAM CONTROL PARAMETERS AND ASSIGN DEFAULTS
C
      XSKT(1)=MAXS
      QMAX(1)=ZTMAX
      CALL MOVE(QMAX(1),QMAX(2),MAXS-1,4)
      QMIN(1)=ZTMIN
      CALL MOVE(QMIN(1),QMIN(2),MAXS-1,4)
      QSUM(1)=0.0
      CALL MOVE(QSUM(1),QSUM(2),MAXS-1,4)
      BMX(1)=0.0
      CALL MOVE(BMX(1),BMX(2),MAXS-1,4)
      READ (READER,1390) NETNAM
      WRITE (PRINTR,1400)
      READ (READER,1410) IUNIT,NBCH,NJNC,NBND,NSTEPS,OUNIT,LUGEOM,NIT,IO
     1TOPT,IPLOPT,IPLDEV,IPRMSG,IPLMSG,IEXOPT,TYPETA,INHR,INMN,IDTM,THET
     2A,QQTOL,ZZTOL,WSPEED,WSDRAG,H2ODEN,CHI,IPUNIN
      IF (IUNIT.NE.ME) IUNIT=EN
      IF (NBCH.LE.0.OR.NBCH.GT.MXBH) GO TO 1380
      IF (NJNC.LE.0.OR.NJNC.GT.MXJN) GO TO 1380
      IF (NBND.LE.0.OR.NBND.GT.MXBY) GO TO 1380
      II=4*NBCH
      IISQ=II*II
      AM(1)=0.0
      CALL MOVE(AM(1),AM(2),IISQ-1,4)
      IF (IUNIT.EQ.EN) GO TO 10
      UNIT=MT
      TUNIT=DC
      AIRDEN=0.001293
      IF (ZZTOL.LE.0.0) ZZTOL=0.003048
      IF (H2ODEN.LE.0.0) H2ODEN=1.011
      GO TO 20
   10 IF (H2ODEN.LE.0.0) H2ODEN=1.9617
      IF (ZZTOL.LE.0.0) ZZTOL=0.01
   20 IF (WSDRAG.LE.0.0) WSDRAG=0.0026
      QTOL=999999.
      IF (OUNIT.NE.ME) OUNIT=EN
      IF (LUGEOM.NE.10) LUGEOM=5
```

```
      IF (NIT.LE.O) NIT=5
      IF (IOTOPT.GT.4) IOTOPT=O
      IF (IPLOPT.GT.4.OR.IPLDEV.GT.3) IPLOPT=O
      PTPLT=IOTOPT.EQ.3
      DAYSUM=IOTOPT.EQ.2.OR.IOTOPT.EQ.4
      NOPRIT=IOTOPT.NE.1
      DTPRT=IOTOPT.EQ.O.OR.IOTOPT.EQ.1
      PRTMSG=IPRMSG.NE.O
      NOEXTP=IEXOPT.EQ.O
      IF (TYPETA.EQ.O.OR.TYPETA.GT.6) TYPETA=1
      IF (INHR.LT.O.OR.INHR.GT.24.OR.INMN.LT.O.OR.INMN.GT.59) GO TO 1380
      IF (THETA.LE.O.O.OR.THETA.GT.1.O) THETA=1.O
      DTHETA=(1.O-THETA)/THETA
      IF (CHI.LE.O.O.OR.CHI.GT.1.O) CHI=THETA
      ONECHI=1.O-CHI
      DCHI=ONECHI/CHI
      OPLOTS=IPLOPT.NE.O.AND.IPLDEV.NE.O.AND.IOTOPT.NE.3
      STAGES=IPLOPT.EQ.2.OR.IPLOPT.EQ.4
      IF (IUNIT.EQ.OUNIT) GO TO 40
      NOCONV=.FALSE.
      IF (IUNIT.EQ.ME) GO TO 30
      QZCONV=O.02832
      IF (STAGES) QZCONV=30.48
      GO TO 40
   30 QZCONV=35.31
      IF (STAGES) QZCONV=3.281
C
C     READ BRANCH IDENTIFICATION PARAMETERS, INITIAL-VALUE DATA, AND
C     CROSS-SECTION DATA
C
   40 CALL ARBIN(AA,BB,ZA,IPT,XSKT,MXPT,MAXS)
      DO 90 I=1,NBCH
      READ (READER,1420) IJF(I),IJT(I),NSEC(I),(BRNAME(K,I),K=1,10)
      IF (I.GT.1) XSKT(I)=XSKT(I-1)-NSEC(I-1)
      NS=NSEC(I)
      IJ=MAXS-XSKT(I)
      DO 90 J=1,NS
      IJ=IJ+1
      IF (J.NE.NS) GO TO 50
      READ (READER,1440) Z(IJ),Q(IJ)
      GO TO 60
   50 READ (READER,1430) Z(IJ),Q(IJ),DX(IJ),T(IJ),(RN(K,IJ),K=1,3)
      RN(4,IJ)=RN(1,IJ)
      IF (T(IJ).EQ.O.O.AND.IUNIT.EQ.EN) T(IJ)=59.O
      IF (T(IJ).EQ.O.O.AND.IUNIT.EQ.ME) T(IJ)=15.O
   60 IF (Q(IJ).EQ.O.O) GO TO 70
      QIJ=ABS(Q(IJ)*O.005)
      IF (QTOL.GT.QIJ) QTOL=QIJ
   70 READ (READER,1470) WANGLE(IJ),BETVEL(IJ)
      IF (BETVEL(IJ).LT.1.O) BETVEL(IJ)=1.O
      READ (LUGEOM,1460) IPT(IJ)
C     INITIALIZE FIRST FORWARD VALUES
      ZP(IJ)=Z(IJ)
      QP(IJ)=Q(IJ)
      ND=IPT(IJ)
      IF (ND.LT.2.OR.ND.GT.MXPT) GO TO 1240
      READ (LUGEOM,1470) (ZA(K,IJ),AA(K,IJ),BB(K,IJ),K=1,ND)
      DO 80 K=2,ND
      IF (ZA(K-1,IJ).GE.ZA(K,IJ)) GO TO 1250
   80 CONTINUE
      IF (Z(IJ).NE.O.O)
     1CALL ARB(ZP(IJ),I,J,AP(IJ),BP(IJ),RP(IJ),&1340,&1350)
      IF (ZA(1,IJ).LT.ZTMIN) ZTMIN=ZA(1,IJ)
      IF (ZA(ND,IJ).GT.ZTMAX) ZTMAX=ZA(ND,IJ)
   90 CONTINUE
      ZDATUM=(ZTMAX+ZTMIN)*O.5
      IF (QTOL.EQ.999999.) QTOL=1.O
      IF (QQTOL.LE.O.O) QQTOL=QTOL
```

```
C
C     READ EXTERNAL INFLOW/OUTFLOW AT INTERNAL JUNCTIONS
C
      READ (READER,1480) (W(J),J=1,NJUNC)
C
C     READ BOUNDARY-VALUE DATA FOR EXTERNAL JUNCTIONS
C
      READ (READER,1490) IDTYPE,LISTB,LISTA
      IF (IDTYPE.EQ.2314) TDDATA=97
      DO 340 L=1,NBND
      READ (READER,1500) ITYPE(L),IBJNC(L),NDATA(L),DTT(L),ISTATN(L),IYR
     1,IMO,IDA,IHR,IMN,NYR,NMO,NDA,NHR,NMN,IRDPDY,DATUM(L),IDONLY(L)
      IF (IBJNC(L).LE.O.OR.IBJNC(L).GT.MXJN) GO TO 1300
      IF (IDONLY(L).EQ.1) GO TO 320
C     THE BOUNDARY-VALUE DATA RECORDED AT THE GREATEST FREQUENCY MUST
C     BE THE FIRST DATA SET SPECIFIED FOR RETRIEVAL FROM DIRECT-ACCESS
C     STORAGE
      ND=NDATA(L)
      CDATUM=DATUM(L)
      IF (ITYPE(L).EQ.IBLK) ITYPE(L)=ZTYPE
      IF (ND.EQ.1) GO TO 100
      IF (DTT(L).EQ.0.0.AND.IRDPDY.EQ.0) GO TO 1270
      IF (IRDPDY.EQ.0) IRDPDY=1440./DTT(L)
      IF (DTT(L).EQ.0.0) DTT(L)=1440./IRDPDY
      IF (IRDPDY.NE.1440./DTT(L)) GO TO 1270
      DTT(L)=DTT(L)*60.
      INTER=1440/IRDPDY
  100 IF (L.NE.1) GO TO 110
      CALL DTCODE(IYR,IMO,IDA,IHR,IMN,IITIME,IETIME,&1270)
      CALL DTCODE(NYR,NMO,NDA,NHR,NMN,NITIME,NETIME,&1270)
  110 IF (ND.NE.0) GO TO 280
C     READ BOUNDARY-VALUE DATA FROM DIRECT-ACCESS STORAGE
      DTYPE=ZTYPE
      IF (ITYPE(L).EQ.QTYPE) DTYPE=QTYPE
      IF (L.NE.1) GO TO 220
      CALL DADIO(PRINTR,PUNCH,GINDEX,TDDATA,LISTB,RTCODE)
      IF (RTCODE.NE.0) GO TO 1270
      IREM=NYR-IYR
      IF (IREM) 1270,130,120
  120 IF (IREM.GT.1) GO TO 1270
      LEAPDY=(4-(IYR-IYR/4*4))/4
      NETIME=NETIME+(365+LEAPDY)*1440
  130 ND=(NETIME-IETIME)/INTER+1
      NDFIRT=ND
      NDPART=ND
      IF (IDTM.EQ.0) IDTM=1440/IRDPDY
      IF (NSTEPS.EQ.0) NSTEPS=((ND-1)*INTER)/IDTM+1
      IF (DTYPE.EQ.QTYPE) GO TO 140
C     CHECK IF NUMBER OF BOUNDARY-VALUE DATA REQUESTED EXCEEDS ARRAY
C     DIMENSIONS
      IF (ND.LE.MAXZBD) GO TO 230
      MAXBD=MAXZBD
      GO TO 150
  140 IF (ND.LE.MAXQBD) GO TO 230
      MAXBD=MAXQBD
  150 MOREBD=.TRUE.
      JETIME=IETIME+(MAXBD-1)*INTER
      LEAPDY=(4-(IYR-IYR/4*4))/4
      IETIYR=(365+LEAPDY)*1440
      JYR=IYR
      IF (JETIME.LE.IETIYR) GO TO 160
      JETIME=JETIME-IETIYR
      JYR=JYR+1
      IF (JYR.GT.99) JYR=0
  160 JDAYN=(JETIME-1)/1440+1
      IREM=JDAYN
      DPERM(2)=28+(4-(JYR-JYR/4*4))/4
      DO 170 K=1,12
```

```
      IF (IREM.LE.DPERM(K)) GO TO 180
  170 IREM=IREM-DPERM(K)
  180 JMO=K
      JDA=IREM
      IREM=JETIME-(JDAYN-1)*1440
      JHR=IREM/60
      JMN=0
      IF (JHR.NE.0) GO TO 200
      JHR=24
      JDA=JDA-1
      IF (JDA.NE.0) GO TO 200
      JMO=JMO-1
      IF (JMO.NE.0) GO TO 190
      JYR=JYR-1
      IF (JYR.LT.0) JYR=99
      JMO=12
  190 JDA=DPERM(JMO)
  200 CALL DTCODE(JYR,JMO,JDA,JHR,JMN,JITIME,JETIME,&1270)
      IF (JYR.EQ.IYR) GO TO 210
      LEAPDY=(4-(IYR-IYR/4*4))/4
      JETIME=JETIME+(365+LEAPDY)*1440
  210 NDPART=(JETIME-IETIME)/INTER+1
  220 IF (.NOT.MOREBD) GO TO 230
      CALL DADI(ISTATN(L),DTYPE,IYR,IMO,IDA,IHR,IMN,JYR,JMO,JDA,JHR,JMN,
     1INDATA(1),IRDPDY,STRIP,PRTMSG,RTCODE)
      ND=(JETIME-IETIME)/INTER+1
      GO TO 240
  230 CALL DADI(ISTATN(L),DTYPE,IYR,IMO,IDA,IHR,IMN,NYR,NMO,NDA,NHR,NMN,
     1INDATA(1),IRDPDY,STRIP,PRTMSG,RTCODE)
      ND=(NETIME-IETIME)/INTER+1
  240 IF (RTCODE.NE.0.AND.(RTCODE.NE.4.OR.STRIP.GE.0).AND.(RTCODE.NE.10.
     1OR.STRIP.GE.0)) GO TO 1270
      IF (ITYPE(L).EQ.QTYPE) GO TO 260
      DO 250 K=1,ND
  250 ZQ(K,L)=IZDATA(K)*0.01+CDATUM-ZDATUM
      GO TO 320
  260 DO 270 K=1,ND
  270 ZQ(K,L)=IQDATA(K)
      GO TO 320
C     READ STAGE/DISCHARGE RATING CURVE COEFFICIENTS
  280 IF (ND.NE.1) GO TO 290
      READ (READER,1450) (ZQBVCO(K,L),K=1,4)
      IF (ITYPE(L).NE.QTYPE) ZQBVCO(1,L)=ZQBVCO(1,L)-ZDATUM
      IZQBVE(L)=1
      GO TO 320
C     READ BOUNDARY-VALUE DATA FROM CARDS
  290 IF (L.NE.1) GO TO 300
      NDFIRT=ND
      NDPART=ND
      IF (IDTM.EQ.0) IDTM=1440/IRDPDY
      IF (NSTEPS.EQ.0) NSTEPS=((ND-1)*INTER)/IDTM+1
  300 READ (READER,1510) (ZQ(K,L),K=1,ND)
      IF (ITYPE(L).EQ.QTYPE) GO TO 320
      DO 310 K=1,ND
  310 ZQ(K,L)=ZQ(K,L)+CDATUM-ZDATUM
  320 DO 330 I=1,NBCH
      IF (IBJNC(L).EQ.IJT(I)) IJVOL(L)=MAXS-XSKT(I)+NSEC(I)
      IF (IBJNC(L).EQ.IJF(I)) IJVOL(L)=MAXS-XSKT(I)+1
  330 CONTINUE
  340 CONTINUE
      IDTPDY=1440/IDTM
      DT=IDTM*60.
      IF (INHR.NE.0.OR.INMN.NE.0) GO TO 350
      INHR=IHR
      INMN=IMN
C
C     READ MEASURED DATA; BEGIN DATE, BEGIN TIME, AND DATA FREQUENCY ARE
C     ASSUMED CONSTANT FOR ALL MEASURED DATA SETS INPUT AS SPECIFIED ON
```

```
C     FIRST MEASURED-DATA DEFINITION CARD.  ALL MEASURED DATA INPUT
C     MUST BE WHOLLY CONTAINED WITHIN A SINGLE DAY.  ALL MEASURED DATA
C     INPUT FOR LINE-PRINTER PLOTTING MUST BE AT THE COMPUTATION TIME-
C     STEP FREQUENCY.
C
  350 L=0
  360 L=L+1
      IF (L.GT.MXMD) GO TO 540
      IF (L.GT.1) GO TO 430
      READ (READER,1500,END=550) MTYPE(L),MJNC,MDATA(L),CDTT,MSTATN(L),M
     1IYR,MIMO,MIDA,MIHR,MIMN,MKYR,MKMO,MKDA,MKHR,MKMN,MDREAD,CDATUM,MSE
     2C(L),MBCH(L)
      IF (CDTT.EQ.0.0.AND.MDREAD.EQ.0) GO TO 1280
      IF (MDREAD.EQ.0) MDREAD=1440./CDTT
      IF (CDTT.EQ.0.0) CDTT=1440./MDREAD
      IF (MDREAD.NE.1440./CDTT) GO TO 1280
      MDT=CDTT
      CALL DTCODE(MIYR,MIMO,MIDA,MIHR,MIMN,IITIME,MEITIM,&1280)
      CALL DTCODE(MKYR,MKMO,MKDA,MKHR,MKMN,NITIME,MEKTIM,&1280)
      IREM=MKYR-MIYR
      IF (IREM) 1280,380,370
  370 IF (IREM.GT.1) GO TO 1280
      LEAPDY=(4-(MIYR-MIYR/4*4))/4
      MEKTIM=MEKTIM+(365+LEAPDY)*1440
  380 IF (MDATA(L).NE.0) GO TO 460
      ND=(MEKTIM-MEITIM)/MDT+1
      IF (ND.LT.1.OR.ND.GT.MAXMZQ) GO TO 1310
  390 MDATA(L)=ND
      DTYPE=ZTYPE
      IF (MTYPE(L).EQ.QTYPE) DTYPE=QTYPE
      CALL DADI(MSTATN(L),DTYPE,MIYR,MIMO,MIDA,MIHR,MIMN,MKYR,MKMO,MKDA,
     1MKHR,MKMN,INDATA(1),MDREAD,STRIP,PRTMSG,RTCODE)
      IF (RTCODE.NE.0.AND.(RTCODE.NE.4.OR.STRIP.GE.0).AND.(RTCODE.NE.10.
     1OR.STRIP.GE.0)) GO TO 1280
      IF (MTYPE(L).EQ.QTYPE) GO TO 410
      DO 400 K=1,ND
  400 ZQMEAS(K,L)=IZDATA(K)*0.01+CDATUM
      GO TO 480
  410 DO 420 K=1,ND
  420 ZQMEAS(K,L)=IQDATA(K)
      GO TO 480
  430 READ (READER,1520,END=550) MTYPE(L),MJNC,MDATA(L),CDTT,MSTATN(L),M
     1DREAD,CDATUM,MBCH(L),MSEC(L)
      IF (CDTT.NE.0.0.OR.MDREAD.NE.0.0) GO TO 440
      MDREAD=1440/MDT
      GO TO 450
  440 IF (CDTT.EQ.0.0.AND.MDREAD.NE.1440/MDT) GO TO 1280
      IF (MDREAD.EQ.0.AND.CDTT.NE.MDT) GO TO 1280
  450 IF (MDATA(L).EQ.0) GO TO 390
  460 ND=MDATA(L)
      IF (ND.LT.1.OR.ND.GT.MAXMZQ) GO TO 1310
      READ (READER,1510) (ZQMEAS(K,L),K=1,ND)
      IF (MTYPE(L).EQ.QTYPE) GO TO 480
      DO 470 K=1,ND
  470 ZQMEAS(K,L)=ZQMEAS(K,L)+CDATUM
  480 IF (NOCONV) GO TO 500
      DO 490 K=1,ND
  490 ZQMEAS(K,L)=ZQMEAS(K,L)*QZCONV
C     DETERMINE BRANCH NUMBER OF MEASURED DATA: FIRST BRANCH ASSOCIATED
C     WITH JUNCTION WILL INDICATE LOCATION.
  500 IF (MBCH(L).EQ.0.AND.MSEC(L).EQ.0) GO TO 510
      IF (MBCH(L).LE.0.OR.MBCH(L).GT.MXBH) GO TO 1320
      IF (MSEC(L).LE.0.OR.MSEC(L).GT.NSEC(MBCH(L))) GO TO 1320
      GO TO 360
  510 DO 530 I=1,NBCH
      MSEC(L)=NSEC(I)
      IF (IJT(I).EQ.MJNC) GO TO 520
      MSEC(L)=1
```

```
      IF (IJF(I).NE.MJUNC) GO TO 530
  520 MBCH(L)=I
      GO TO 360
  530 CONTINUE
      GO TO 1330
  540 READ (READER,1520,END=550)
      WRITE (PRINTR,1720) MXMD
C
C     ASSIGN UNINITIALIZED STAGE VALUE AT BOUNDARY-VALUE-DATA LOCATION
C     TO FIRST STAGE VALUE OF BOUNDARY-VALUE-DATA INPUT
C
  550 KTMEAS=L-1
      DO 600 I=1,NBCH
      NS=NSEC(I)
      IJ=MAXS-XSKT(I)
      DO 600 J=1,NS
      IJ=IJ+1
      IF (Z(IJ).NE.0.0) GO TO 600
      IF (J.NE.1.AND.J.NE.NS) GO TO 590
      FOUND=.FALSE.
      DO 580 L=1,NBND
      IF (IZQBVE(L).EQ.1) GO TO 580
      IF (ITYPE(L).NE.ZTYPE) GO TO 580
      IF (J.EQ.NS) GO TO 560
      IF (IBJNC(L).EQ.IJF(I)) GO TO 570
      GO TO 580
  560 IF (IBJNC(L).NE.IJT(I)) GO TO 580
  570 FOUND=.TRUE.
      Z(IJ)=ZQ(1,L)+ZDATUM
      ZP(IJ)=Z(IJ)
      CALL ARB(ZP(IJ),I,J,AP(IJ),BP(IJ),RP(IJ),&1340,&1350)
  580 CONTINUE
      IF (FOUND) GO TO 600
  590 ERROR=.TRUE.
      WRITE (PRINTR,1640) I,J
  600 CONTINUE
      IF (ERROR) STOP
C
C     PRINT OUT COMPUTATION CONTROL CARD INFORMATION
C
      WRITE (PRINTR,1530) IUNIT,OUNIT,MXBH,NBCH,MXJN,NJNC,MXBY,NBND,LUGE
     1OM,IOTOPT,IPLOPT,IPLDEV,IPRMSG,IPLMSG,IEXOPT,IPUNIN,TYPETA,NIT,NST
     2EPS,THETA,CHI,IDTM,QQTOL,ZZTOL,WSPEED,WSDRAG,H2ODEN,ZDATUM
      DO 610 L=1,NBND
      IF (IZQBVE(L).EQ.1) GO TO 610
      WRITE (PRINTR,1540) IBJNC(L),DATUM(L)
  610 CONTINUE
      WRITE (PRINTR,1550)
C
C     PRINT OUT CROSS-SECTION DATA
C
      DO 650 I=1,NBCH
      NS=NSEC(I)
      IJ=MAXS-XSKT(I)
      WRITE (PRINTR,1560) NETNAM,I,IJF(I),IJT(I),(BRNAME(K,I),K=1,10)
      DO 650 J=1,NS
      IJ=IJ+1
      ND=IPT(IJ)
      WRITE (PRINTR,1570) J,UNIT,UNIT,UNIT
      WRITE (PRINTR,1580) (ZA(K,IJ),AA(K,IJ),BB(K,IJ),K=1,ND)
      WRITE (PRINTR,1590) Z(IJ),Q(IJ),BETVEL(IJ)
      IF (Z(IJ).LE.ZA(ND,IJ).AND.Z(IJ).GE.ZA(1,IJ)) GO TO 620
      ERROR=.TRUE.
      WRITE (PRINTR,1650) Z(IJ),I,J
  620 IF (J.EQ.NS) GO TO 650
      WRITE (PRINTR,1600)
      WRITE (PRINTR,1610) DX(IJ),UNIT,T(IJ),TUNIT,WANGLE(IJ),RN(1,IJ)
      IF (RN(3,IJ).NE.0.0) GO TO 630
```

```
      IF (RN(2,IJ).EQ.0.0) GO TO 640
      WRITE (PRINTR,1620) RN(2,IJ),IDETA(TYPETA)
      GO TO 640
  630 WRITE (PRINTR,1630) RN(2,IJ),IDETA(TYPETA),RN(3,IJ),IDETA(TYPETA)
  640 WRITE (PRINTR,1600)
      IF (J.NE.NS) WANGLE(IJ)=COS(0.01745329*WANGLE(IJ))
  650 CONTINUE
      IF (ERROR) STOP
      IF (IUNIT.EQ.EN) GO TO 660
      WSPEED=WSPEED*1000./3600.
      G=9.806
      CN=1.
      GO TO 670
  660 G=32.174
      WSPEED=WSPEED*5280./3600.
      CN=1.486
  670 TWOG=2.0*G
      TWOCSQ=2.0*CN*CN
      CW=WSDRAG*AIRDEN/(H2ODEN*G)*WSPEED*WSPEED
C
C     APPLY STAGE COMPUTATION DATUM
C
      IF (ZDATUM.EQ.0.0) GO TO 690
      DO 680 I=1,NBCH
      NS=NSEC(I)
      IJ=MAXS-XSKT(I)
      DO 680 J=1,NS
      IJ=IJ+1
      Z(IJ)=Z(IJ)-ZDATUM
      ZP(IJ)=Z(IJ)
      ND=IPT(IJ)
      DO 680 K=1,ND
  680 ZA(K,IJ)=ZA(K,IJ)-ZDATUM
C
C     CALCULATE NUMBER OF BRANCHES AT EACH JUNCTION AND ASSIGN INDICES
C
  690 DO 710 J=1,NJNC
      ICT(J)=0
      DO 710 I=1,NBCH
      IF (IJF(I).NE.J) GO TO 700
      ICT(J)=ICT(J)+1
      IDX(J,ICT(J))=-I
  700 IF (IJT(I).NE.J) GO TO 710
      ICT(J)=ICT(J)+1
      IDX(J,ICT(J))=I
  710 CONTINUE
C
C     BEGIN COMPUTATION LOOP
C
      KYR=IYR
      KMO=IMO
      KDA=IDA
      KHR=INHR
      KMN=INMN
      LETIME=IETIME
      KT=(INHR*60+INMN-1)/IDTM
      IF (PTPLT) KT=0
      ICHK=0
      CALL GEMXPI(AM,BMX,ROW,II,IS,ICHK)
      DPERM(2)=28+(4-MOD(KYR,4))/4
      CALL OUT(NSEC,XSKT,QMIN,QSUM,QMAX,ITQMIN,ZQMIN,AQMIN,ITQMAX,ZQMAX,
     1AQMAX,RN,MAXS,ISTATN,IJVKT,IFVOL,ITVOL,FVSTAT,ISTAPR,MXBY,
     2PRINTR,PUNCH)
      IF (OPLOTS) CALL OPLOT(NSEC,XSKT,BRNAME,IJF,IJT,ISTATN,
     1ZQCOMP,IBJNC,MBCH,MSEC,MDATA,ZQMEAS,MAXCZQ,MAXS,MAXMZQ,PRINTR)
      IF (PTPLT) CALL PRTPLT(NSEC,XSKT,BRNAME,IJF,IJT,ISTATN,ZQCOMP,IBJN
     1C,MBCH,MSEC,MDATA,ZQMEAS,MAXCZQ,MAXS,MAXMZQ,IYR,IMO,IDA,IHR,IMN,PR
     2INTR)
```

```
      KTMATS=0
      ND=1
      N=0
      DO 1230 M=1,NSTEPS
      LASTN=N
      KT=KT+1
C
C     PREPARATION FOR NEXT TIME STEP
C
      DO 790 I=1,NBCH
      NS=NSEC(I)
      IJ=MAXS-XSKT(I)
      DO 790 J=1,NS
      IJ=IJ+1
      IF (IOTOPT.NE.4) GO TO 740
      DO 730 L=1,NBND
      IF (IJ.NE.IJVOL(L)) GO TO 730
      IF ((Q(IJ).GT.0.0.AND.QP(IJ).GT.0.0).OR.(Q(IJ).LT.0.0.AND.QP(IJ).L
     1T.0.0).OR.(Q(IJ).EQ.0.0.AND.QP(IJ).EQ.0.0)) GO TO 720
      DTZERO=(-Q(IJ)*DT)/(QP(IJ)-Q(IJ))
      IFVOL(IJVKT(L),1,L)=IFVOL(IJVKT(L),1,L)+Q(IJ)*0.01*DTZERO
      ITVOL(IJVKT(L),1,L)=KHR*100+KMN
      IJVKT(L)=IJVKT(L)+1
      IFVOL(IJVKT(L),1,L)=IFVOL(IJVKT(L),1,L)+QP(IJ)*0.01*(DT-DTZERO)
      GO TO 730
  720 IFVOL(IJVKT(L),1,L)=IFVOL(IJVKT(L),1,L)+Q(IJ)*0.01*DT
  730 CONTINUE
  740 ZTEMP=Z(IJ)
      QTEMP=Q(IJ)
      Z(IJ)=ZP(IJ)
      Q(IJ)=QP(IJ)
      QIJ=Q(IJ)
      IF (STAGES) GO TO 750
      ZQCOMP(KT,IJ)=QIJ*QZCONV
      GO TO 760
  750 ZQCOMP(KT,IJ)=(Z(IJ)+ZDATUM)*QZCONV
  760 QSUM(IJ)=QSUM(IJ)+QIJ
      IF (QIJ.LE.QMAX(IJ)) GO TO 770
      QMAX(IJ)=QIJ
      ZQMAX(IJ)=ZP(IJ)+ZDATUM
      AQMAX(IJ)=AP(IJ)
      ITQMAX(IJ)=KHR*100+KMN
  770 IF (QIJ.GE.QMIN(IJ)) GO TO 780
      QMIN(IJ)=QIJ
      ZQMIN(IJ)=ZP(IJ)+ZDATUM
      AQMIN(IJ)=AP(IJ)
      ITQMIN(IJ)=KHR*100+KMN
  780 IF (NOEXTP) GO TO 790
C     (IEXOPT=0) USE CURRENT VALUES AS INITIAL VALUES FOR UNKNOWNS
C     (IEXOPT=1) EXTRAPOLATE INITIAL VALUES FOR UNKNOWNS FROM CURRENT
      ZP(IJ)=2.*ZP(IJ)-ZTEMP
      QP(IJ)=2.*QP(IJ)-QTEMP
  790 CONTINUE
C
C     BEGIN ITERATIVE IMPROVEMENT LOOP
C
      DO 1190 N=1,NIT
C
C     CALCULATE BRANCH MATRICES
C
      DO 820 I=1,NBCH
      IJ=MAXS-XSKT(I)
      IJP1=IJ+1
      NSM1=NSEC(I)-1
      CALL ARB(Z(IJP1),I,1,A(IJP1),B(IJP1),R(IJP1),&1360,&1350)
      IF (M.EQ.NSTEPS) GO TO 800
      CALL ARB(ZP(IJP1),I,1,AP(IJP1),BP(IJP1),RP(IJP1),&1360,&1350)
  800 DO 820 J=1,NSM1
```

```
      IJ=IJ+1
      DXIJ=DX(IJ)
      QIJ=Q(IJ)
      ZIJ=Z(IJ)
      JP1=J+1
      IJP1=IJ+1
      QIJP1=Q(IJP1)
      ZIJP1=Z(IJP1)
      CALL ARB(ZIJP1,I,JP1,A(IJP1),B(IJP1),R(IJP1),&1360,&1350)
      IF (M.EQ.NSTEPS) GO TO 820
      CALL ARB(ZP(IJP1),I,JP1,AP(IJP1),BP(IJP1),RP(IJP1),&1360,&1350)
      BAVG=CHI*((BP(IJ)+BP(IJP1))*0.5)+ONECHI*((B(IJ)+B(IJP1))*0.5)
      AAVG=CHI*((AP(IJ)+AP(IJP1))*0.5)+ONECHI*((A(IJ)+A(IJP1))*0.5)
      RAVG=CHI*((RP(IJ)+RP(IJP1))*0.5)+ONECHI*((R(IJ)+R(IJP1))*0.5)
      QAVG=CHI*((QP(IJ)+QP(IJP1))*0.5)+ONECHI*((QIJ+QIJP1)*0.5)
      BETCOR=(BETVEL(IJ)+BETVEL(IJP1))*0.5
      IF (TYPETA.NE.1) RN(4,IJ)=SETA(TYPETA,IUNIT,G,QAVG,AAVG,RAVG,T(IJ)
     1,RN(1,IJ))
      RNIJ=RN(4,IJ)
      AAVGSQ=AAVG*AAVG
      AAVGCU=AAVGSQ*AAVG
      LAMBDA=DXIJ/(TWOG*AAVG*DT*THETA)
      SIGMA=ABS(QAVG)*RNIJ*RNIJ*DXIJ*CHI/(TWOCSQ*AAVGSQ*RAVG**1.3333333*
     1THETA)
      MU=2.0*BETCOR*QAVG/(G*AAVGSQ)
      EPSLON=(LAMBDA-DCHI*SIGMA)*(QIJ+QIJP1)-MU*DTHETA*(QIJP1-QIJ)-DTHET
     1A*(ZIJP1-ZIJ)+BETCOR*QAVG*QAVG/(G*THETA*AAVGCU)*(AP(IJP1)-AP(IJ))+
     2CW*BAVG*WANGLE(IJ)*DXIJ/(THETA*AAVG)
      ZETA=LAMBDA+SIGMA+MU
      OMEGA=LAMBDA+SIGMA-MU
      GAMMA=DXIJ*BAVG/(2.*DT*THETA)
      DELTA=GAMMA*(ZIJ+ZIJP1)-DTHETA*(QIJP1-QIJ)
      DET=1./(1.-ZETA*GAMMA)
C     SEGMENT MATRIX COMPUTATION
      IJ4=(IJ-1)*4
      IJ4P1=IJ4+1
      IJ4P2=IJ4+2
      IJ4P3=IJ4+3
      IJ4P4=IJ4+4
      UU(IJ4P1)=(1.+ZETA*GAMMA)*DET
      UU(IJ4P2)=(-OMEGA-ZETA)*DET
      UU(IJ4P3)=(-2.*GAMMA)*DET
      UU(IJ4P4)=(1.+OMEGA*GAMMA)*DET
      IJ2=(IJ-1)*2
      IJ2P1=IJ2+1
      IJ2P2=IJ2+2
      U(IJ2P1)=(EPSLON-ZETA*DELTA)*DET
      U(IJ2P2)=(DELTA-EPSLON*GAMMA)*DET
      I4=(I-1)*4
      I4P1=I4+1
      I4P2=I4+2
      I4P3=I4+3
      I4P4=I4+4
      I2=(I-1)*2
      I2P1=I2+1
      I2P2=I2+2
      IF (J.GT.1) GO TO 810
      BUU(I4P1)=UU(IJ4P1)
      BUU(I4P2)=UU(IJ4P2)
      BUU(I4P3)=UU(IJ4P3)
      BUU(I4P4)=UU(IJ4P4)
      BU(I2P1)=U(IJ2P1)
      BU(I2P2)=U(IJ2P2)
      GO TO 820
C     BRANCH MATRIX COMPUTATION
  810 C1=BUU(I4P1)
      C2=BUU(I4P2)
      C3=BUU(I4P3)
```

```
      C4=BUU(I4P4)
      UUIJP1=UU(IJ4P1)
      UUIJP2=UU(IJ4P2)
      UUIJP3=UU(IJ4P3)
      UUIJP4=UU(IJ4P4)
      BUU(I4P1)=UUIJP1*C1+UUIJP2*C3
      BUU(I4P2)=UUIJP1*C2+UUIJP2*C4
      BUU(I4P3)=UUIJP3*C1+UUIJP4*C3
      BUU(I4P4)=UUIJP3*C2+UUIJP4*C4
      C1=BU(I2P1)
      C2=BU(I2P2)
      BU(I2P1)=UUIJP1*C1+UUIJP2*C2+U(IJ2P1)
      BU(I2P2)=UUIJP3*C1+UUIJP4*C2+U(IJ2P2)
  820 CONTINUE
C
C     IS THIS THE FIRST ITERATION (N=1) OF THIS TIME STEP (M) ?
C
      IF (N.NE.1) GO TO 880
      IF (DTPRT) CALL DTOUT(Q,Z,A,B)
      IF (.NOT.PTPLT) GO TO 830
      IF (KT.LT.MAXCZQ.AND.KHR.NE.24) GO TO 840
      KETIME=IETIME+(M-1)*IDTM
      CALL LNPLOT
      KT=0
      GO TO 840
  830 IF (KT.LT.IDTPDY) GO TO 840
      IF (DAYSUM) CALL DAILY
      KT=0
      IF (.NOT.OPLOTS) GO TO 840
      KETIME=IETIME+(M-1)*IDTM
      CALL ZQPLOT
  840 IF (M.EQ.NSTEPS) GO TO 1370
      KMN=KMN+IDTM
      IF (KMN.LT.60) GO TO 850
      KHR=KHR+KMN/60
      KMN=MOD(KMN,60)
  850 IF (KHR.LT.24.OR.(KHR.EQ.24.AND.KMN.EQ.0)) GO TO 860
      KHR=KHR-24
      KDA=KDA+1
  860 IF (KDA.LE.DPERM(KMO)) GO TO 870
      KDA=1
      KMO=KMO+1
  870 IF (KMO.LT.13) GO TO 880
      KMO=1
      KYR=KYR+1
      IF (KYR.GT.99) KYR=0
      DPERM(2)=28+(4-MOD(KYR,4))/4
  880 IF (NOPRIT) GO TO 890
      LASTN=N-1
      CALL DTOUT(QP,ZP,AP,BP)
C
C     SET UP NETWORK MATRIX AND VECTOR
C
  890 NN=1
      MM=1
      DO 900 I=1,NBCH
C     INSERT BRANCH MATRICES
      NNN=IAR(NN,MM)
      I4=(I-1)*4
      I2=(I-1)*2
      AM(NNN)=BUU(I4+1)
      NNN=NNN+II
      AM(NNN)=BUU(I4+2)
      NNN=NNN+II
      AM(NNN)=-1.
      NNN=IAR(NN+1,MM)
      AM(NNN)=BUU(I4+3)
      NNN=NNN+II
```

```
      AM(NNN)=BUU(I4+4)
      NNN=NNN+II+II
      AM(NNN)=-1.
C     CONSTRUCT RIGHT SIDE VECTOR
      BMX(NN)=-BU(I2+1)
      BMX(NN+1)=-BU(I2+2)
      NN=NN+2
      MM=MM+4
  900 CONTINUE
C
C     INSERT BOUNDARY CONDITIONS FOR INTERNAL JUNCTIONS
C
      DO 930 J=1,NJUNC
      IF (ICT(J).EQ.1) GO TO 930
      NBPJ=ICT(J)
C     INSERT DISCHARGE CONTINUITY
      DO 910 I=1,NBPJ
      IBCH=IDX(J,I)
      MM=4+(IABS(IBCH)-1)*4
      IF (IBCH.LT.O) MM=MM-2
      NNN=IAR(NN,MM)
      AM(NNN)=IBCH/IABS(IBCH)
  910 CONTINUE
      BMX(NN)=-W(J)
      NN=NN+1
C     INSERT STAGE COMPATIBILITY
      IBCH=IDX(J,1)
      MO=3+(IABS(IBCH)-1)*4
      IF (IBCH.LT.O) MO=MO-2
      DO 920 I=2,NBPJ
      IBCH=IDX(J,I)
      MM=3+(IABS(IBCH)-1)*4
      NNN=IAR(NN,MO)
      AM(NNN)=1.
      IF (IBCH.LT.O) MM=MM-2
      NNN=IAR(NN,MM)
      AM(NNN)=-1.
      BMX(NN)=O.
      NN=NN+1
  920 CONTINUE
  930 CONTINUE
C
C     RETRIEVE ADDITIONAL BOUNDARY-VALUE DATA FROM DIRECT-ACCESS STORAGE
C
      IF (N.NE.1) GO TO 1080
      IF (.NOT.MOREBD) GO TO 1080
      K=ND*DT/DTT(1)+1.
      IF (K.LE.NDPART) GO TO 1080
      DO 1070 L=1,NBND
      IF (IDONLY(L).EQ.1.OR.NDATA(L).NE.O) GO TO 1070
      INTER=DTT(L)/60.
      IRDPDY=1440/INTER
      DTYPE=ZTYPE
      IF (ITYPE(L).EQ.QTYPE) DTYPE=QTYPE
      IF (L.NE.1) GO TO 1010
      ND=NDFIRT-NDPART+1
      NDFIRT=ND
      NDPART=ND
      IF (ND.GT.MAXBD) GO TO 940
      MOREBD=.FALSE.
      GO TO 1010
  940 METIME=JETIME+(MAXBD-1)*INTER
      LEAPDY=(4-(JYR-JYR/4*4))/4
      IETJYR=(365+LEAPDY)*1440
      MYR=JYR
      IF (METIME.LE.IETJYR) GO TO 950
      METIME=METIME-IETJYR
      MYR=MYR+1
```

```
      IF (MYR.GT.99) MYR=0
  950 JDAYN=(METIME-1)/1440+1
      IREM=JDAYN
      DPERM(2)=28+(4-(MYR-MYR/4*4))/4
      DO 960 K=1,12
      IF (IREM.LE.DPERM(K)) GO TO 970
  960 IREM=IREM-DPERM(K)
  970 MMO=K
      MDA=IREM
      IREM=METIME-(JDAYN-1)*1440
      MHR=IREM/60
      MMN=0
      IF (MHR.NE.0) GO TO 990
      MHR=24
      MDA=MDA-1
      IF (MDA.NE.0) GO TO 990
      MMO=MMO-1
      IF (MMO.NE.0) GO TO 980
      MYR=MYR-1
      IF (MYR.LT.0) MYR=99
      MMO=12
  980 MDA=DPERM(MMO)
  990 CALL DTCODE(MYR,MMO,MDA,MHR,MMN,MITIME,METIME,&1270)
      IF (MYR.EQ.JYR) GO TO 1000
      LEAPDY=(4-(JYR-JYR/4*4))/4
      METIME=METIME+(365+LEAPDY)*1440
 1000 NDPART=(METIME-JETIME)/INTER+1
 1010 IF (.NOT.MOREBD) GO TO 1020
      CALL DADI(ISTATN(L),DTYPE,JYR,JMO,JDA,JHR,JMN,MYR,MMO,MDA,MHR,MMN,
     1INDATA(1),IRDPDY,STRIP,PRTMSG,RTCODE)
      ND=(METIME-JETIME)/INTER+1
      GO TO 1030
 1020 CALL DADI(ISTATN(L),DTYPE,JYR,JMO,JDA,JHR,JMN,NYR,NMO,NDA,NHR,NMN,
     1INDATA(1),IRDPDY,STRIP,PRTMSG,RTCODE)
      ND=(NETIME-JETIME)/INTER+1
 1030 IF (RTCODE.NE.0.AND.(RTCODE.NE.4.OR.STRIP.GE.0).AND.(RTCODE.NE.10.
     1OR.STRIP.GE.0)) GO TO 1270
      IF (ITYPE(L).EQ.QTYPE) GO TO 1050
      CDATUM=DATUM(L)
      DO 1040 K=1,ND
 1040 ZQ(K,L)=IZDATA(K)*0.01+CDATUM-ZDATUM
      GO TO 1070
 1050 DO 1060 K=1,ND
 1060 ZQ(K,L)=IQDATA(K)
 1070 CONTINUE
      JETIME=METIME
      JYR=MYR
      JMO=MMO
      JDA=MDA
      JHR=MHR
      JMN=MMN
      ND=1
C
C     INSERT BOUNDARY CONDITIONS FOR EXTERNAL JUNCTIONS
C
 1080 DO 1140 L=1,NBND
      IF (IDONLY(L).EQ.1) GO TO 1140
      IBCH=IDX(IBJNC(L),1)
      MM=1+(IABS(IBCH)-1)*4
      IF (ITYPE(L).EQ.QTYPE) MM=MM+1
      IF (IBCH.GT.0) MM=MM+2
      NNN=IAR(NN,MM)
      AM(NNN)=1.
C     DETERMINE BOUNDARY CONDITION FROM STAGE/DISCHARGE BOUNDARY-
C     CONDITION EQUATION
      IF (IZQBVE(L).EQ.0) GO TO 1110
      IF (IBCH.LT.0) GO TO 1090
      IJ=MAXS-XSKT(IBCH)+NSEC(IBCH)
```

```
      GO TO 1100
 1090 IBCH=-IBCH
      IJ=MAXS-XSKT(IBCH)+1
 1100 ZQPIJ=ZP(IJ)
      IF (ITYPE(L).EQ.QTYPE) ZQPIJ=QP(IJ)
      BMX(NN)=ZQBVCO(1,L)+(ZQBVCO(2,L)+(ZQBVCO(3,L)+ZQBVCO(4,L)*ZQPIJ)*Z
     1QPIJ)*ZQPIJ
      GO TO 1130
C     PARABOLIC INTERPOLATION FOR BOUNDARY CONDITION FROM BOUNDARY-VALUE
C     DATA
 1110 K=ND*DT/DTT(L)+1.
      TH=(ND*DT-(K-1)*DTT(L))/DTT(L)
      IF (NDATA(L).EQ.0) GO TO 1115
      K=M*DT/DTT(L)+1.
      TH=(M*DT-(K-1)*DTT(L))/DTT(L)
 1115 IF (K.NE.1) GO TO 1120
      K=2
      TH=TH-1.
 1120 BMX(NN)=.5*TH*(ZQ(K+1,L)-ZQ(K-1,L)+TH*(ZQ(K+1,L)+ZQ(K-1,L)-2.*ZQ(K
     1,L)))+ZQ(K,L)
 1130 NN=NN+1
 1140 CONTINUE
C
C     SOLVE MATRIX OF LINEAR EQUATIONS
C
      IF (II.NE.NN-1) GO TO 1260
      CALL GEMXP(IS,ICHK)
      KTMATS=KTMATS+1
      AM(1)=0.0
      CALL MOVE(AM(1),AM(2),IISQ-1,4)
      IF (IS.EQ.1) GO TO 1290
C
C     CALCULATE INTERMEDIATE VALUES
C
      NN=1
      BIGQ=0.0
      BIGZ=0.0
      DO 1180 I=1,NBCH
      IJ=MAXS-XSKT(I)
      IJP1=IJ+1
      ZTEMP=ZP(IJP1)
      QTEMP=QP(IJP1)
      ZP(IJP1)=BMX(NN)
      QP(IJP1)=BMX(NN+1)
      ZTOL=ABS(ZTEMP-ZP(IJP1))
      QTOL=ABS(QTEMP-QP(IJP1))
      IF (ZTOL.LE.BIGZ) GO TO 1150
      BIGZ=ZTOL
      IBIGZ=I
      JBIGZ=1
 1150 IF (QTOL.LE.BIGQ) GO TO 1160
      BIGQ=QTOL
      IBIGQ=I
      JBIGQ=1
 1160 CONTINUE
      NN=NN+4
      NSM1=NSEC(I)-1
      DO 1180 J=1,NSM1
      IJ=IJ+1
      IJP1=IJ+1
      IJ2=(IJ-1)*2
      IJ4=(IJ-1)*4
      ZTEMP=ZP(IJP1)
      QTEMP=QP(IJP1)
      ZP(IJP1)=UU(IJ4+1)*ZP(IJ)+UU(IJ4+2)*QP(IJ)+U(IJ2+1)
      QP(IJP1)=UU(IJ4+3)*ZP(IJ)+UU(IJ4+4)*QP(IJ)+U(IJ2+2)
      ZTOL=ABS(ZTEMP-ZP(IJP1))
      QTOL=ABS(QTEMP-QP(IJP1))
```

```
      IF (ZTOL.LE.BIGZ) GO TO 1170
      BIGZ=ZTOL
      IBIGZ=I
      JBIGZ=J+1
 1170 IF (QTOL.LE.BIGQ) GO TO 1180
      BIGQ=QTOL
      IBIGQ=I
      JBIGQ=J+1
 1180 CONTINUE
      IF (BIGZ.LE.ZZTOL.AND.BIGQ.LE.QQTOL) GO TO 1220
 1190 CONTINUE
      IF (NOCONV) GO TO 1210
      IF (IUNIT.EQ.ME) GO TO 1200
      BIGZ=BIGZ*0.3048
      BIGQ=BIGQ*0.02832
      GO TO 1210
 1200 BIGZ=BIGZ*3.281
      BIGQ=BIGQ*35.31
 1210 WRITE (PRINTR,1810) KHR,KMN,KYR,KMO,KDA,IBIGZ,JBIGZ,BIGZ,IBIGQ,JBI
     1GQ,BIGQ
 1220 ND=ND+1
 1230 CONTINUE
C
C     TERMINATE EXECUTION
C
      GO TO 1370
 1240 WRITE (PRINTR,1660) MXPT
      STOP
 1250 WRITE (PRINTR,1670) I,J
      STOP
 1260 WRITE (PRINTR,1680)
      GO TO 1370
 1270 WRITE (PRINTR,1690)
      STOP
 1280 WRITE (PRINTR,1700)
      STOP
 1290 WRITE (PRINTR,1710)
      GO TO 1370
 1300 WRITE (PRINTR,1730) IBJNC(L),MXJN
      STOP
 1310 WRITE (PRINTR,1740) MAXMZQ
      STOP
 1320 WRITE (PRINTR,1750) MBCH(L),MSEC(L)
      STOP
 1330 WRITE (PRINTR,1760) MJNC
      STOP
 1340 WRITE (PRINTR,1770)
      STOP
 1350 WRITE (PRINTR,1780)
      STOP
 1360 WRITE (PRINTR,1790)
      IF (DAYSUM) CALL DAILY
      IF (DTPRT) CALL DTOUT(QP,ZP,AP,BP)
 1370 CONTINUE
      IF (M.EQ.NSTEPS.AND.IPUNIN.EQ.1) CALL PINCO(Z,Q,DX,T,WANGLE,BETVEL
     1)
      KETIME=NETIME
      IF (PTPLT.AND.KT.NE.0) CALL LNPLOT
      IF (LISTA.NE.0) CALL DADIO(PRINTR,PUNCH,GINDEX,TDDATA,LISTA,RTCODE
     1)
      WRITE (PRINTR,1820) KTMATS
      IF (.NOT.OPLOTS) STOP
      IF (KT.EQ.0) STOP
      CALL ZQPLOT
      STOP
 1380 WRITE (PRINTR,1800)
      STOP
C
```

```
C     INPUT/OUTPUT FORMAT STATEMENTS                                    BR1013
C                                                                       BR1014
C                                                                       BR1015
 1390 FORMAT (20A4)                                                     BR1016
 1400 FORMAT ('1',38X,' UNSTEADY FLOW COMPUTATION IN A NETWORK OF OPEN C BR1017
     1HANNELS '//' ',46X,'BRANCH-NETWORK MODEL (VERSION 79/04/19)'/' ',5 BR1018
     22X,'A FOUR-POINT IMPLICIT SCHEME'//' ',55X,'LINEAR MATRIX SOLUTION BR1019
     3'/' ',42X,'BY GAUSS ELIMINATION USING MAXIMUM PIVOT STRATEGY'/)   BR1020
 1410 FORMAT (A2,3I2,I4,A2,2I2,7I1,2I2,I4,F3.2,F5.1,F5.3,F5.2,2F5.4,F3.2 BR1021
     1,I1)                                                              BR1022
 1420 FORMAT (3I2,10A4)                                                 BR1023
 1430 FORMAT (2F10.3,10X,2F10.2,3E10.4)                                 BR1024
 1440 FORMAT (2F10.3)                                                   BR1025
 1450 FORMAT (4E10.4)                                                   BR1026
 1460 FORMAT (I2)                                                       BR1027
 1470 FORMAT (3F10.3)                                                   BR1028
 1480 FORMAT (10F8.2)                                                   BR1029
 1490 FORMAT (I4,33X,I2,6X,I2)                                          BR1030
 1500 FORMAT (A2,I2,I3,F2.0,I8,7X,5(I2,1X),5X,5(I2,1X),2X,I4,F7.3,7X,I1, BR1031
     1T78,I2)                                                           BR1032
 1510 FORMAT (F10.3)                                                    BR1033
 1520 FORMAT (A2,I2,I3,F2.0,I8,44X,I4,F7.3,5X,I2,I1)                    BR1034
 1530 FORMAT (/39X,'==================================================== BR1035
     1===='//49X,'UNITS OF INPUT (EN/ME)      =',A9/49X,'UNITS OF OUTPUT  BR1036
     2(EN/ME)     =',A9/49X,'BRANCHES (1<=N<=',I2,')          =',I9/49X,'JU BR1037
     3NCTIONS (2<=N<=',I2,')         =',I9/49X,'BOUNDARY VALUES (1<=N<=',I BR1038
     42,') =',I9/49X,'GEOMETRY INPUT UNIT (5/10) =',I9/49X,'PRINTOUT OPT BR1039
     5ION (0<=N<=4)  =',I9/49X,'PLOT OPTION (0<=N<=4)        =',I9/49X,'PL BR1040
     6OTTER DEVICE (0<=N<=3)   =',I9/49X,'PRINT MESSAGE OPTION (0/1) =', BR1041
     7I9/49X,'PLOT MESSAGE OPTION (0/1)  =',I9/49X,'EXTRAPOLATION OPTION BR1042
     8 (0/1) =',I9/49X,'PUNCH INITIAL COND. (0/1)  =',I9/49X,'FRICTION T BR1043
     9YPE (1<=N<=6)     =',I9/49X,'MAXIMUM ITERATIONS          =',I9/49X,' BR1044
     $NUMBER OF STEPS            =',I9/49X,'DERIVATIVE FACTOR (0<=N<=1)= BR1045
     $',F9.2/49X,'GEOMETRY FACTOR (0<=N<=1)  =',F9.2/49X,'TIME INCREMENT BR1046
     $(MINUTES)     =',I9/49X,'DISCHARGE CONVERGENCE       =',F9.1/49X,'ST BR1047
     $AGE CONVERGENCE            =',F9.4/49X,'WIND SPEED(MPH/KPH)        = BR1048
     $',F9.1/49X,'SURFACE DRAG COEFFICIENT   =',F9.4/49X,'WATER DENSITY  BR1049
     $               =',F9.4/49X,'STAGE COMPUTATION DATUM     =',F9.3)    BR1050
 1540 FORMAT (49X,'BVD(',I2,') DATUM CORRECTION   =',F9.3)              BR1051
 1550 FORMAT (/39X,'==================================================== BR1052
     1====')                                                            BR1053
 1560 FORMAT ('1CHANNEL GEOMETRY FOR ',20A4//' BRANCH ',I2,' FROM JUNCTI BR1054
     1ON ',I2,' TO ',I2,' : ',10A4)                                     BR1055
 1570 FORMAT (/'  CROSS SECTION',I3,' :',6X,'STAGE',16X,'AREA',17X,'WIDT BR1056
     1H'/27X,'(',A2,')',14X,'(',A2,'**2)',17X,'(',A2,')')               BR1057
 1580 FORMAT (21X,F10.2,11X,F10.1,11X,F10.1)                            BR1058
 1590 FORMAT (/'  INITIAL VALUES:  STAGE=',F6.2,'  DISCHARGE=',F9.1,10X, BR1059
     1'BETA=',F6.3)                                                     BR1060
 1600 FORMAT (2X,'------------------------------------------------------ BR1061
     1-----------------')                                               BR1062
 1610 FORMAT (2X,'LENGTH=',F8.1,' ',A2,';',9X,' TEMP= ',F4.1,' DEG',A2,' BR1063
     1;',9X,' WIND= ',F5.1,' DEG'//2X,'ETA=',E13.6)                     BR1064
 1620 FORMAT ('+',18X,' + (',E13.6,')*',A1)                             BR1065
 1630 FORMAT ('+',18X,2(' + (',E13.6,')*',A1),'**2')                    BR1066
 1640 FORMAT (' ERROR,     INITIAL STAGE VALUE UNSPECIFIED IN BRANCH ',I BR1067
     12,' SECTION ',I2)                                                 BR1068
 1650 FORMAT (' ERROR,     INITIAL STAGE',F7.2,' OUT OF DEFINED RANGE OF BR1069
     1 CHANNEL GEOMETRY FOR BRANCH ',I2,' SECTION ',I2)                 BR1070
 1660 FORMAT (' ERROR,     IMPROPER NUMBER OF CROSS-SECTIONAL DATA (2<=I BR1071
     1PT<=',I2,')')                                                     BR1072
 1670 FORMAT (' ERROR,     DUPLICATE, OR OUT-OF-ORDER, STAGES IN CHANNEL BR1073
     1-GEOMETRY TABLE FOR BRANCH ',I2,' SECTION ',I2)                   BR1074
 1680 FORMAT (' ERROR,     MATRIX NOT SQUARE')                          BR1075
 1690 FORMAT (' ERROR,     INVALID BOUNDARY-VALUE DATA PARAMETER(S)')   BR1076
 1700 FORMAT (' ERROR,     INVALID MEASURED DATA PARAMETER(S)')         BR1077
 1710 FORMAT (' ERROR,     MATRIX IS SINGULAR')                         BR1078
 1720 FORMAT (' ERROR,     TOO MANY MEASURED DATA LOCATIONS (MXMD=',I2,' BR1079
     1)')                                                               BR1080
```

```
 1730 FORMAT (' ERROR,     JUNCTION ',I2,' OF BOUNDARY-VALUE DATA IMPROP
     1ERLY SPECIFIED (O<IBJNC<=',I2,')')
 1740 FORMAT (' ERROR,     IMPROPER NUMBER OF MEASURED DATA SPECIFIED (1
     1<=MDATA<=',I3,')')
 1750 FORMAT (' ERROR,     INVALID BRANCH ',I2,' SECTION ',I2,' SPECIFIE
     1D FOR MEASURED DATA')
 1760 FORMAT (' ERROR,     JUNCTION ',I2,' OF MEASURED DATA NOT FOUND')
 1770 FORMAT (' ERROR,     INITIAL VALUE(S) OUT OF DEFINED RANGE OF CHAN
     1NEL GEOMETRY')
 1780 FORMAT (' ERROR,     INITIAL OR COMPUTED STAGE RESULTS IN ZERO OR
     1NEGATIVE AREA AND/OR TOP WIDTH')
 1790 FORMAT (' ERROR,     COMPUTED STAGE OUT OF DEFINED RANGE OF CHANNE
     1L GEOMETRY')
 1800 FORMAT (' ERROR,     INVALID COMPUTATION CONTROL PARAMETER(S)')
 1810 FORMAT (' WARNING,   MAXIMUM ITERATIONS EXCEEDED AT ',I2,':',I2,'
     1ON ',I2,'/',I2,'/',I2,'     Z-ZP(',I2,',',I2,')=',F7.4,' Q-QP(',I2
     2,',',I2,')=',F7.1)
 1820 FORMAT (/' NUMBER OF SOLUTIONS = ',I4/)
      END


      SUBROUTINE OUT(NSEC,XSKT,QMIN,QSUM,QMAX,ITQMIN,ZQMIN,AQMIN,ITQMAX,
     1ZQMAX,AQMAX,RN,MAXS,ISTATN,IJVKT,IFVOL,ITVOL,FVSTAT,ISTAPR,
     2MXBY,PRINTR,PUNCH)
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #   THIS SUBROUTINE PRINTS FLOW COMPUTATION RESULTS IN TABULAR FORM #
C #   AND OPTIONALLY PUNCHES INITIAL CONDITIONS AT THE END OF A RUN.  #
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #                                                                   #
      INTEGER PRINTR,PUNCH
      INTEGER *2EN/'EN'/,MT/' M'/,IUNIT,OUNIT,UNIT/'FT'/,IBLK/'  '/,ASTK
     1/' *'/,IPART,FVUNIT
      INTEGER *2NSEC(1),XSKT(1),ITQMIN(1),ITQMAX(1)
      REAL A(1),B(1),Z(1),Q(1),DX(1),T(1),RN(4,1),QMIN(1),QSUM(1),QMAX(1
     1),ZQMIN(1),AQMIN(1),ZQMAX(1),AQMAX(1),WANGLE(1),BETVEL(1)
      LOGICAL IFIRST/.TRUE./,ILAST,PRTDAY,PRTIME,PARTDY/.FALSE./,STAGES
      COMMON /DATIME/ KYR,KMO,KDA,KHR,KMN,IDTM,M,NSTEPS,INHR,INMN,IDTPDY
     1,LASTN
      COMMON /OUTPUT/ NETNAM(20),NBCH,NBND,IOTOPT,IPLOPT,IPLDEV,STAGES,Z
     1DATUM,IUNIT,OUNIT
      INTEGER ISTATN(1),BLANK/'    '/
      INTEGER FVNAME(20),FVYEAR,FVSTAT(31,1),ISTAPR(1)
      INTEGER*2 MONTHS(5,12)/' J','AN','UA','RY',' ',' F','EB','RU','AR
     1','Y ',' ','MA','RC','H ',' ',' ','AP','RI','L ',' ',' ',' M'
     2,'AY',' ',' ',' ',' ',' J','UN','E ',' ',' ',' J','UL','Y ',' ',
     3' ','AU','GU','ST',' ',' SE','PT','EM','BE','R ',' O','CT','OB','
     4ER',' ',' N','OV','EM','BE','R ',' D','EC','EM','BE','R '/
      INTEGER IFVOL(8,31,1)
      INTEGER*2 IJVKT(1),ITVOL(8,31,1),FVDATA(12)
      INTEGER *2IDTTHR(8),IDTUHR(8),IDTTMN(8),IDTUMN(8)
      INTEGER *2DPERM(12)/31,28,31,30,31,30,31,31,30,31,30,31/
      LOGICAL EMPTY
      REAL *8ERRVAL
      EXTERNAL CREATE
      COMMON /IHC236/ EMPTY,LOGUN
      REAL REPEAT(7)/4H,1(',4H,2(',4H,3(',4H,4(',4H,5(',4H,6(',4H,7('/
      REAL PRTFMT(11)/4H(' ',4H,T12,4H7,I6,4H,T5,,4HI2,4,4HX,I6,4H,7(',
     14H (',,4H4I1,,4H')',,4HI6))/
C     SET DEFINE FILE 50 MAXIMUM RECORD SIZE TO 52 TIMES MXBY
      DEFINE FILE 50(367,260,L,NEXREC)
      QCONVT=1.0
      FVFACT=0.0001
      LASTM=0
      KTLINE=60
      NUMXS=MAXS-XSKT(NBCH)+NSEC(NBCH)
      IF (IOTOPT.NE.4) GO TO 80
      DPERM(2)=28+(4-(KYR-KYR/4*4))/4
```

```
      EMPTY=.FALSE.
      CALL ERRSAV(236,ERRVAL)
      CALL ERRSET(236,2,-1,1,CREATE,236)
      READ (50'367) FVNAME,FVYEAR,FVDATA,FVUNIT
      IF (EMPTY) GO TO 30
      DO 10 K=1,20
      IF (FVNAME(K).NE.NETNAM(K)) GO TO 20
   10 CONTINUE
      IF (FVUNIT.NE.OUNIT) GO TO 20
      IF (FVYEAR.EQ.KYR) GO TO 60
      IF (FVYEAR.EQ.KYR+1.AND.KMO.EQ.12.AND.KDA.EQ.31) GO TO 60
   20 WRITE (PRINTR,570)
      STOP
   30 DO 40 K=1,20
   40 FVNAME(K)=NETNAM(K)
      FVUNIT=OUNIT
      FVYEAR=KYR
      IF (KMO.EQ.12.AND.KDA.EQ.31) FVYEAR=FVYEAR+1
      DO 50 MO=1,12
   50 FVDATA(MO)=0
      WRITE (50'367) FVNAME,FVYEAR,FVDATA,FVUNIT
      CALL ERRSTR(236,ERRVAL)
   60 DO 70 L=1,MXBY
      IJVKT(L)=1
      DO 70 K=1,8
      ITVOL(K,1,L)=0
   70 IFVOL(K,1,L)=0
   80 CONTINUE
      ASSIGN 100 TO ICONVT
      ASSIGN 240 TO NCONVT
      IF (OUNIT.EQ.EN) GO TO 85
      UNIT=MT
      FVFACT=0.1
   85 IF (IUNIT.EQ.OUNIT) RETURN
      QCONVT=0.02832
      ASSIGN 110 TO ICONVT
      ASSIGN 230 TO NCONVT
      IF (IUNIT.EQ.EN) RETURN
      QCONVT=35.31
      ASSIGN 120 TO ICONVT
      ASSIGN 220 TO NCONVT
      RETURN
      ENTRY DTOUT(Q,Z,A,B)
C
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #   SECONDARY ENTRY POINT TO OUTPUT FLOW RESULTS AT EACH TIME STEP  #
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C
      PRTDAY=KMN.EQ.0.OR.M.EQ.NSTEPS
      IF (.NOT.IFIRST) GO TO 90
      IFIRST=.FALSE.
      PRTDAY=.TRUE.
      WRITE (PRINTR,620) NETNAM
      WRITE (PRINTR,640)
      WRITE (PRINTR,630) UNIT,UNIT,UNIT,UNIT,UNIT,UNIT,UNIT,UNIT,UNIT,UN
     1IT,UNIT
      WRITE (PRINTR,640)
   90 CONTINUE
      PRTIME=.TRUE.
      DO 170 I=1,NBCH
      IBXS=0
      NSM1=NSEC(I)-1
      IJ=MAXS-XSKT(I)
      DO 170 J=1,NSM1
      IJ=IJ+1
      IJP1=IJ+1
      GO TO ICONVT, (100,110,120)
C     NO CONVERSION REQUIRED
```

```
  100 ZIJ=Z(IJ)+ZDATUM
      ZIJP1=Z(IJP1)+ZDATUM
      QIJ=Q(IJ)
      QIJP1=Q(IJP1)
      AIJ=A(IJ)
      AIJP1=A(IJP1)
      BIJ=B(IJ)
      BIJP1=B(IJP1)
      GO TO 130
C     CONVERT FROM ENGLISH TO METRIC
  110 ZIJ=(Z(IJ)+ZDATUM)*0.3048
      ZIJP1=(Z(IJP1)+ZDATUM)*0.3048
      QIJ=Q(IJ)*0.02832
      QIJP1=Q(IJP1)*0.02832
      AIJ=A(IJ)*0.0929
      AIJP1=A(IJP1)*0.0929
      BIJ=B(IJ)*0.3048
      BIJP1=B(IJP1)*0.3048
      GO TO 130
C     CONVERT FROM METRIC TO ENGLISH
  120 ZIJ=(Z(IJ)+ZDATUM)*3.281
      ZIJP1=(Z(IJP1)+ZDATUM)*3.281
      QIJ=Q(IJ)*35.31
      QIJP1=Q(IJP1)*35.31
      AIJ=A(IJ)*10.76
      AIJP1=A(IJP1)*10.76
      BIJ=B(IJ)*3.281
      BIJP1=B(IJP1)*3.281
  130 VIJ=QIJ/AIJ
      VIJP1=QIJP1/AIJP1
      DZ=ZIJ-ZIJP1
      IBXS=IBXS+1
      IEXS=IBXS+1
      IF (PRTDAY) GO TO 150
      IF (.NOT.PRTIME) GO TO 140
      WRITE (PRINTR,750) KHR,KMN,ZIJ,VIJ,QIJ,AIJ,BIJ,DZ,I,LASTN,IBXS,IEX
     1S,RN(4,IJ),ZIJP1,VIJP1,QIJP1,AIJP1,BIJP1
      GO TO 160
  140 WRITE (PRINTR,650) ZIJ,VIJ,QIJ,AIJ,BIJ,DZ,I,IBXS,IEXS,RN(4,IJ),ZIJ
     1P1,VIJP1,QIJP1,AIJP1,BIJP1
      GO TO 170
  150 WRITE (PRINTR,660) KYR,KMO,KDA,KHR,KMN,ZIJ,VIJ,QIJ,AIJ,BIJ,DZ,I,LA
     1STN,IBXS,IEXS,RN(4,IJ),ZIJP1,VIJP1,QIJP1,AIJP1,BIJP1
      PRTDAY=.FALSE.
  160 PRTIME=.FALSE.
  170 CONTINUE
      IF (KHR.NE.24.AND.M.NE.NSTEPS) RETURN
      WRITE (PRINTR,640)
      WRITE (PRINTR,670)
      KT=M-LASTM
      DO 190 I=1,NBCH
      NSM1=NSEC(I)-1
      IJ=MAXS-XSKT(I)
      DO 180 J=1,NSM1
      IJ=IJ+1
      IJP1=IJ+1
      QMINIJ=QMIN(IJ)*QCONVT
      QBARIJ=QSUM(IJ)*QCONVT/KT
      QMAXIJ=QMAX(IJ)*QCONVT
      QMNJP1=QMIN(IJP1)*QCONVT
      QBRJP1=QSUM(IJP1)*QCONVT/KT
      QMXJP1=QMAX(IJP1)*QCONVT
      WRITE (PRINTR,680) QMINIJ,QBARIJ,QMAXIJ,QMNJP1,QBRJP1,QMXJP1
      QMIN(IJ)=+9999999.
      QMAX(IJ)=-9999999.
      QSUM(IJ)=0.0
  180 CONTINUE
      QMIN(IJP1)=+9999999.
```

```
      QMAX(IJP1)=-9999999.
      QSUM(IJP1)=0.0
  190 CONTINUE
      WRITE (PRINTR,690)
      IFIRST=.TRUE.
      LASTM=M
      RETURN
      ENTRY DAILY
C
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #   SECONDARY ENTRY POINT TO OUTPUT DAILY SUMMARIES OF FLOW RESULTS #
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C
      IF (KTLINE+NUMXS.LT.60) GO TO 200
      WRITE (PRINTR,620) NETNAM
      WRITE (PRINTR,700)
      WRITE (PRINTR,710) UNIT,UNIT,UNIT,UNIT,UNIT,UNIT,UNIT,UNIT,UNIT
      WRITE (PRINTR,700)
      KTLINE=9
  200 ILAST=M.EQ.NSTEPS
      IPART=IBLK
      KT=M-LASTM
      IF (KT.EQ.IDTPDY) GO TO 210
      IPART=ASTK
      PARTDY=.TRUE.
  210 DO 270 I=1,NBCH
      IJ=MAXS-XSKT(I)
      NS=NSEC(I)
      DO 270 J=1,NS
      IJ=IJ+1
      GO TO NCONVT, (220,230,240)
C     CONVERT FROM METRIC TO ENGLISH
  220 ZQMIN(IJ)=ZQMIN(IJ)*3.281
      ZQMAX(IJ)=ZQMAX(IJ)*3.281
      QMIN(IJ)=QMIN(IJ)*35.31
      QSUM(IJ)=QSUM(IJ)*35.31
      QMAX(IJ)=QMAX(IJ)*35.31
      AQMIN(IJ)=AQMIN(IJ)*10.76
      AQMAX(IJ)=AQMAX(IJ)*10.76
      GO TO 240
C     CONVERT FROM ENGLISH TO METRIC
  230 ZQMIN(IJ)=ZQMIN(IJ)*0.3048
      ZQMAX(IJ)=ZQMAX(IJ)*0.3048
      QMIN(IJ)=QMIN(IJ)*0.02832
      QSUM(IJ)=QSUM(IJ)*0.02832
      QMAX(IJ)=QMAX(IJ)*0.02832
      AQMIN(IJ)=AQMIN(IJ)*0.0929
      AQMAX(IJ)=AQMAX(IJ)*0.0929
      GO TO 240
C     NO CONVERSION REQUIRED
  240 VQMIN=QMIN(IJ)/AQMIN(IJ)
      VQMAX=QMAX(IJ)/AQMAX(IJ)
      QBARIJ=QSUM(IJ)/KT
      MINHR=ITQMIN(IJ)/100
      MINMN=ITQMIN(IJ)-MINHR*100
      MAXHR=ITQMAX(IJ)/100
      MAXMN=ITQMAX(IJ)-MAXHR*100
      IF (I.EQ.1.AND.J.EQ.1) GO TO 250
      WRITE (PRINTR,730) MINHR,MINMN,ZQMIN(IJ),VQMIN,QMIN(IJ),AQMIN(IJ),
     1I,J,QBARIJ,MAXHR,MAXMN,ZQMAX(IJ),VQMAX,QMAX(IJ),AQMAX(IJ)
      GO TO 260
  250 WRITE (PRINTR,720) IPART,KYR,KMO,KDA,MINHR,MINMN,ZQMIN(IJ),VQMIN,Q
     1MIN(IJ),AQMIN(IJ),I,J,QBARIJ,MAXHR,MAXMN,ZQMAX(IJ),VQMAX,QMAX(IJ),
     2AQMAX(IJ)
  260 KTLINE=KTLINE+1
      QMIN(IJ)=+9999999.
      QMAX(IJ)=-9999999.
      QSUM(IJ)=0.0
```

```
  270 CONTINUE
      LASTM=M
      IF (IOTOPT.NE.4) GO TO 330
      IF (KYR.NE.FVYEAR.OR.KT.NE.IDTPDY) GO TO 310
      JDAYN=KDA
      KM1=KMO-1
      IF (KM1.EQ.O) GO TO 290
      DO 280 K=1,KM1
  280 JDAYN=JDAYN+DPERM(K)
  290 CONTINUE
      DO 300 L=1,NBND
      ND=IJVKT(L)
      DO 300 K=1,ND
      REALFV=IFVOL(K,1,L)
  300 IFVOL(K,1,L)=REALFV*FVFACT*QCONVT+SIGN(0.5,REALFV)
      WRITE (50'JDAYN) (ISTATN(L),(ITVOL(K,1,L),IFVOL(K,1,L),K=1,8),L=1,
     1NBND)
      FVDATA(KMO)=1
C     WRITE IDENTITY RECORD TO DIRECT-ACCESS FLOW VOLUME FILE
      WRITE (50'367) FVNAME,FVYEAR,FVDATA,FVUNIT
  310 DO 320 L=1,NBND
      ND=IJVKT(L)
      IJVKT(L)=1
      DO 320 K=1,ND
      IFVOL(K,1,L)=0
  320 ITVOL(K,1,L)=0
  330 IF (ILAST) GO TO 340
      IF (KTLINE+NUMXS.LT.60) RETURN
  340 WRITE (PRINTR,700)
      IF (.NOT.PARTDY) GO TO 350
      WRITE (PRINTR,740)
      PARTDY=.FALSE.
  350 IF (.NOT.ILAST) RETURN
      IF (IOTOPT.NE.4) RETURN
C
C     PRINT MONTHLY SUMMARIES OF FLOW VOLUMES
      DO 540 MO=1,12
      IF (FVDATA(MO).EQ.O) GO TO 540
      ND=DPERM(MO)
      KTSTAP=0
      DO 360 L=1,MXBY
  360 ISTAPR(L)=0
      JDAYNB=1
      KM1=MO-1
      IF (KM1.EQ.O) GO TO 380
      DO 370 K=1,KM1
  370 JDAYNB=JDAYNB+DPERM(K)
  380 JDAYNE=JDAYNB+DPERM(MO)-1
      K=1
C     READ FLOW VOLUMES FROM DIRECT-ACCESS FILE
      DO 390 JDAYN=JDAYNB,JDAYNE
      READ (50'JDAYN) (FVSTAT(K,L),(ITVOL(J,K,L),IFVOL(J,K,L),J=1,8),L=1
     1,MXBY)
  390 K=K+1
C     LOCATE NEXT FLOW VOLUME DATA STATION TO BE PRINTED
  400 DO 440 K=1,ND
      DO 430 L=1,MXBY
      IF (FVSTAT(K,L).EQ.BLANK.OR.FVSTAT(K,L).EQ.O) GO TO 440
      IF (KTSTAP.EQ.O) GO TO 420
      DO 410 ISTA=1,KTSTAP
      IF (FVSTAT(K,L).EQ.ISTAPR(ISTA)) GO TO 430
  410 CONTINUE
  420 KTSTAP=KTSTAP+1
      ISTPRO=FVSTAT(K,L)
      ISTAPR(KTSTAP)=ISTPRO
      IF (OUNIT.EQ.EN) GO TO 424
      WRITE (PRINTR,585) MONTHS(1,MO),MONTHS(2,MO),MONTHS(3,MO),MONTHS(4
     1,MO),MONTHS(5,MO),KYR,ISTPRO
```

```
      WRITE (PRINTR,590)
      GO TO 450
  424 WRITE (PRINTR,580) MONTHS(1,MO),MONTHS(2,MO),MONTHS(3,MO),MONTHS(4
     1,MO),MONTHS(5,MO),KYR,ISTPRO
      WRITE (PRINTR,590)
      GO TO 450
  430 CONTINUE
  440 CONTINUE
      GO TO 540
C     PRINT MONTHLY FLOW VOLUME SUMMARY FOR STATION ISTPRO
  450 NOBLNK=0
      DO 530 K=1,ND
      DO 480 L=1,MXBY
      IF (FVSTAT(K,L).EQ.BLANK.OR.FVSTAT(K,L).EQ.0) GO TO 490
      IF (FVSTAT(K,L).NE.ISTPRO) GO TO 480
      IFVSUM=IFVOL(1,K,L)
      DO 460 J=2,8
      ITIME=ITVOL(J-1,K,L)
      IF (ITIME.EQ.0) GO TO 470
      IDTHR=ITIME/100
      IDTMN=ITIME-IDTHR*100
      IDTTHR(J-1)=IDTHR/10
      IDTUHR(J-1)=IDTHR-IDTTHR(J-1)*10
      IDTTMN(J-1)=IDTMN/10
      IDTUMN(J-1)=IDTMN-IDTTMN(J-1)*10
      IFVSUM=IFVSUM+IFVOL(J,K,L)
  460 CONTINUE
      JVOL=J
      GO TO 510
  470 JVOL=J-1
      GO TO 500
  480 CONTINUE
  490 NOBLNK=NOBLNK+1
      IF (NOBLNK.GT.1) GO TO 530
      WRITE (PRINTR,600)
      GO TO 530
  500 IF (JVOL.GT.1) GO TO 510
      WRITE (PRINTR,610) IFVSUM,K,IFVOL(1,K,L)
      GO TO 520
  510 PRTFMT(7)=REPEAT(JVOL-1)
      WRITE (PRINTR,PRTFMT) IFVSUM,K,IFVOL(1,K,L),(IDTTHR(J-1),IDTUHR(J-
     11),IDTTMN(J-1),IDTUMN(J-1),IFVOL(J,K,L),J=2,JVOL)
  520 NOBLNK=0
  530 CONTINUE
      GO TO 400
  540 CONTINUE
      RETURN
      ENTRY PINCO(Z,Q,DX,T,WANGLE,BETVEL)
C
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #   SECONDARY ENTRY POINT TO PUNCH INITIAL CONDITIONS AT END OF RUN #
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C
      DO 560 I=1,NBCH
      IJ=MAXS-XSKT(I)
      NSM1=NSEC(I)-1
      DO 550 J=1,NSM1
      IJ=IJ+1
      ZIJ=Z(IJ)+ZDATUM
      WRITE (PUNCH,760) ZIJ,Q(IJ),DX(IJ),T(IJ),(RN(K,IJ),K=1,3)
      WINDIJ=ARCOS(WANGLE(IJ))/0.01745329
  550 WRITE (PUNCH,770) WINDIJ,BETVEL(IJ)
      ZIJ=Z(IJ+1)+ZDATUM
  560 WRITE (PUNCH,780) ZIJ,Q(IJ+1)
      RETURN
C
C     OUTPUT FORMAT STATEMENTS
C
```

```
C
  570 FORMAT (' EXECUTION TERMINATED DUE TO INCORRECT ATTEMPT TO UPDATE
     1CUMULATIVE FLOW-VOLUME FILE')
  580 FORMAT ('1',5A2,'19',I2,' FLOW IN MILLIONS OF CUBIC FEET AT STATIO
     1N ',I8/)
  585 FORMAT ('1',5A2,'19',I2,' FLOW IN THOUSANDS OF CUBIC METERS AT STA
     1TION ',I8/)
  590 FORMAT ('    DAY    ',7('  FLOW (REV.)'),'  FLOW'/'    NO.    ',7('
     1 VOL. (TIME)'),'  VOL.'/)
  600 FORMAT (' ')
  610 FORMAT (' ',T127,I6,T5,I2,4X,I6)
  620 FORMAT (1H1,T34,'FLOW RESULTS FOR ',20A4/)
  630 FORMAT ('    DATE    TIME| STAGE |VELOCITY|DISCHARGE|  AREA | WIDT
     1H| FALL |BRANCH SECTION|  ETA | STAGE |VELOCITY|DISCHARGE|  AREA |
     2 WIDTH|'/'  YR/MO/DY HR:MN|  (',A2,') |(',A2,'/SEC)|(',A2,'**3/S)|
     3(',A2,'**2)|  (',A2,')| (',A2,') | (ITERATIONS) |',6X,'|  (',A2,')
     4 |(',A2,'/SEC)|(',A2,'**3/S)|(',A2,'**2)|  (',A2,')|')
  640 FORMAT ('  --------------+-------+--------+---------+-------+-----
     1-+------+--------------+------+-------+--------+---------+-------+
     2------+')
  650 FORMAT (16X,'|',F6.2,F8.2,F11.1,F9.1,F7.1,'|',F5.2,4X,I2,4X,I2,':'
     1,I2,2X,F6.4,'|',F6.2,F8.2,F11.1,F9.1,F7.1,'|')
  660 FORMAT ('  ',2(I2,'/'),I2,I3,':',I2,'|',F6.2,F8.2,F11.1,F9 1,F7.1,
     1'|',F5.2,4X,I2,1X,'(',I1,')',I2,':',I2,2X,F6.4,'|',F6.2,F8.2,F11.1
     2,F9.1,F7.1,'|')
  670 FORMAT (16X,'|',16X,'DISCHARGE',16X,'|',28X,'|',16X,'DISCHARGE',16
     1X,'|'/16X,'|     MINIMUM',7X,'MEAN',9X,'MAXIMUM  |',28X,'|     MIN
     2IMUM',7X,'MEAN',9X,'MAXIMUM  |')
  680 FORMAT (16X,'|   ',2(F9.1,4X),1X,F9.1,'   |',28X,'|    ',2(F9.1,4X),
     11X,F9.1,'   |')
  690 FORMAT (16X,'+',41('-'),'+',28X,'+',41('-'),'+')
  700 FORMAT ('  ---------+-------+-------+--------+---------+-------+-
     1------------+---------+-------+-------+--------+---------+-------
     22+')
  710 FORMAT (12X,'|',26X,'MINIMUM',9X,'|',18X,'MEAN',2X,'|',26X,'MAXIMU
     1M',9X,'|'/'     DATE     |  TIME   STAGE  VELOCITY DISCHARGE   AREA
     2|BRANCH SECTION DISCHARGE|  TIME    STAGE  VELOCITY DISCHARGE   ARE
     3A |'/'   (YR/MO/DY)|(HR:MN)    (',A2,')    (',A2,'/S)  (',A2,'**3/S)
     4(',A2,'**2)|',15X,'(',A2,'**3/S)|(HR:MN)    (',A2,')    (',A2,'/S)
     5(',A2,'**3/S) (',A2,'**2)|')
  720 FORMAT (1X,A2,2(I2,'/'),I2,' | ',I2,':',I2,'  ',F6.2,F8.2,F11.1,F9
     1.1,'|  ',I2,6X,I2,'   ',F8.1,' | ',I2,':',I2,'  ',F6.2,F8.2,F11.1,
     2F9.1,'|')
  730 FORMAT (11X,' | ',I2,':',I2,'  ',F6.2,F8.2,F11.1,F9.1,'|  ',I2,6X,
     1I2,'   ',F8.1,' | ',I2,':',I2,'  ',F6.2,F8.2,F11.1,F9.1,'|')
  740 FORMAT ('  * IDENTIFIES A PARTIAL DAY OF RECORD')
  750 FORMAT (10X,I3,':',I2,'|',F6.2,F8.2,F11.1,F9.1,F7.1,'|',F5.2,4X,I2
     1,1X,'(',I1,')',I2,':',I2,2X,F6.4,'|',F6.2,F8.2,F11.1,F9.1,F7.1,'|'
     2)
  760 FORMAT (2F10.2,10X,2F10.2,3E10.4)
  770 FORMAT (F10.2,F10.4)
  780 FORMAT (2F10.2/)
      END


      SUBROUTINE OPLOT(NSEC,XSKT,BRNAME,IJF,IJT,ISTATN,ZQCOMP,
     1IBJNC,MBCH,MSEC,MDATA,ZQMEAS,MAXCZQ,MAXS,MAXMZQ,PRINTR)
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #                        SUBROUTINE OPLOT                           #
C #   THIS FORTRAN AND DISSPLA CODED SUBROUTINE PREPARES A DAILY      #
C #   COMPUTED AND, OPTIONALLY, MEASURED DISCHARGE/STAGE HYDROGRAPH   #
C #   FOR PLOTTING ON A TEKTRONIX CRT GRAPHICS TERMINAL USING THE     #
C #   DISSPLA POSTPROCESSOR, A CALCOMP 936 (36 INCH DRUM) PLOTTER, OR #
C #   A III FR80 MICROFILM PLOTTER.                                   #
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #                                                                   #
      REAL LEGEND(4,2)
```

```
      INTEGER BRNAME(10,1),ISTATN(1)
      INTEGER YR,MO,DA,HR,MN,SE
C     INITIALIZE CPP FOR THE NUMBER OF CURVES PER PLOT (MAXIMUM OF 5)
      INTEGER CPP/5/,IQUAD/0/
C     TO OVERRIDE PROGRAM DEFAULTS FOR LEGEND LOCATION, SET IQUAD TO
C     DESIRED QUADRANT.
      INTEGER PRINTR
      INTEGER *2IUNIT,OUNIT,ME/'ME'/
      INTEGER *2NSEC(1),XSKT(1),IJF(1),IJT(1),IBJNC(1)
      INTEGER *2MBCH(1),MSEC(1),MDATA(1),MIYR,MIMO,MIDA,MIHR,MIMN
      DIMENSION ZQCOMP(MAXCZQ,1), ZQMEAS(MAXMZQ,1), ZQTIME(288), RMTIME(
     1288)
      LOGICAL LOCLEG/.TRUE./,MEPLOT/.FALSE./,IFLAG,LABSTA,STAGES
      COMMON /DATIME/ KYR,KMO,KDA,KHR,KMN,IDTM,M,NSTEPS,INHR,INMN,IDTPDY
     1,LASTN
      COMMON /OUTPUT/ NETNAM(20),NBCH,NBND,IOTOPT,IPLOPT,IPLDEV,STAGES,Z
     1DATUM,IUNIT,OUNIT
      COMMON /MEDATA/ MDT,KTMEAS,LETIME,KETIME,MEITIM,MEKTIM,IPLMSG,MIYR
     1,MIMO,MIDA,MIHR,MIMN
      DATA ZQCMAX/-999.0/,TCMAX/2400.0/
      DATA SIZE/0.14/,XLEN/7.80/,YLEN/10.0/,KTPLOT/0/,HSIZ/0.105/,SIZ7/0
     1.07/,CHARWH/0.857/,HLEG/0.087/,YLEGSP/0.080/,XLCURV/1.00/,IONE/-1/
C     DISSPLA CHARACTER WIDTH/HEIGHT RATIO IS 6/7 (CHARWH = 0.857)
C
C     INITIALIZE PLOT CONTROL PARAMETERS
C
      CALL JULDAT(YR,MO,DA)
      CALL TIME(SE)
      SE=(SE+49)/100
      HR=SE/3600
      SE=SE-HR*3600
      MN=SE/60
      SE=SE-MN*60
      HSIZWH=CHARWH*HSIZ
      SIZEWH=CHARWH*SIZE
      ID=(INHR*60+INMN)/IDTM
      ND=1440/IDTM
      DO 10 K=1,ND
   10 ZQTIME(K)=IDTM*K
      XORGIN=0.0
      XSTEP=1440./XLEN
      XSPACE=XLEN/12.0
      GO TO (20,30,40), IPLDEV
   20 CALL COMPRS
      GO TO 50
   30 CALL CC936
      GO TO 50
   40 CALL FR80(1)
   50 IF (IPLMSG.EQ.0) GO TO 60
      IONE=1
      IPLMSG=1
   60 IF (IPLOPT.LT.3) RETURN
      CPP=1
      MID=1
      MEPLOT=.TRUE.
      IDM=(MIHR*60+MIMN)/MDT
      MND=1440/MDT
      DO 70 K=1,MND
   70 RMTIME(K)=MDT*K
      RETURN
      ENTRY ZQPLOT
C
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #   SECONDARY ENTRY POINT TO PRODUCE GRAPHICAL PLOTS                #
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C
      IF (M.EQ.NSTEPS) ND=(KHR*60+KMN)/IDTM
      KD=ND-ID+1
```

```
      IF (KD.EQ.1) GO TO 460
      IF (.NOT.MEPLOT) GO TO 80
      MND=(KHR*60+KMN)/MDT
      IF (KETIME.LT.MEITIM.OR.LETIME.GT.MEKTIM) GO TO 460
C
C     DETERMINE POSSIBLE COORDINATES FOR LOCATION OF LEGEND(P,Q)
C     P = DESIRED QUADRANT; Q = 1 STORES X-COORD.; Q = 2 STORES Y-COORD.
C
   80 XPOS=XLEN-(XLCURV+.175+4.0*HSIZWH)
      YPOS=YLEN-(.175+HLEG)
      LEGEND(1,1)=XPOS
      LEGEND(1,2)=YPOS
      LEGEND(2,1)=.175
      LEGEND(2,2)=YPOS
      IF (IQUAD.LE.2) GO TO 90
      LEGEND(3,1)=.175
      LEGEND(3,2)=.175
      LEGEND(4,1)=XPOS
      LEGEND(4,2)=.175
C
C     BEGIN BRANCH CURVE PLOTTING LOOP
C
   90 DO 450 I=1,NBCH
      IF (.NOT.MEPLOT) GO TO 110
      DO 100 MM=1,KTMEAS
      IF (MBCH(MM).NE.I) GO TO 100
      IF (MND.GT.MDATA(MM)) MND=MDATA(MM)
      MDATA(MM)=MDATA(MM)-MND
      MKD=MND-MID+1
      GO TO 110
  100 CONTINUE
      GO TO 450
  110 NPLOT=1
      KK=1
      IBND=1
      NS=NSEC(I)
      IN=MAXS-XSKT(I)
      IF (NS.LE.CPP) GO TO 120
      NPLOT=NS/CPP
      KK=NPLOT*CPP
      IF (KK.NE.NS) NPLOT=NPLOT+1
C
C     BEGIN LOOP CONTROLLING NUMBER OF PLOTS
C
  120 DO 440 IPLOT=1,NPLOT
      IFLAG=.FALSE.
      KTPLOT=KTPLOT+1
      CALL BGNPL(IONE*KTPLOT)
      CALL FLATBD
      CALL BLOWUP(1.25)
      CALL PAGE(11.0,14.0)
      CALL BANGLE(90.)
      CALL BSHIFT(7.475,0.125)
      CALL NOBRDR
      CALL TITLE(BRNAME(1,I),-40,0,0,0,0,XLEN,YLEN)
C
C     DETERMINE MINIMUM AND MAXIMUM DISCHARGE/STAGE FOR PLOT SCALING
C
      ZQMIN=+9999999.
      ZQMAX=-9999999.
      INS=(IPLOT-1)*CPP+1
      JNS=NS
      IF (IPLOT.NE.NPLOT) JNS=IPLOT*CPP
      IJ=IN+INS-1
      DO 130 J=INS,JNS
      IJ=IJ+1
      DO 130 K=ID,ND
      ZQIJK=ZQCOMP(K,IJ)
```

```
      IF (ZQIJK.LT.ZQMIN) ZQMIN=ZQIJK
      IF (ZQIJK.LE.ZQMAX) GO TO 130
      ZQMAX=ZQIJK
      TMAX=ZQTIME(K)
  130 CONTINUE
      IF (.NOT.MEPLOT) GO TO 150
      IF (IPLOT.NE.MSEC(MM)) GO TO 150
      ZQCMIN=ZQMIN
      ZQCMAX=ZQMAX
      TCMAX=TMAX
      IFLAG=.TRUE.
      DO 140 K=MID,MND
      ZQIJK=ZQMEAS(K,MM)
      IF (ZQIJK.LT.ZQMIN) ZQMIN=ZQIJK
      IF (ZQIJK.LE.ZQMAX) GO TO 140
      ZQMAX=ZQIJK
      TMAX=RMTIME(K)
  140 CONTINUE
      IF (ZQCMIN.LT.ZQMIN) ZQMIN=ZQCMIN
      IF (ZQCMAX.GT.ZQMAX) ZQMAX=ZQCMAX
C
C     ESTABLISH PLOT ORIGIN AND SCALE
C
  150 RANGE=ZQMAX-ZQMIN
      IPOWER=ALOG10(RANGE)-1.0
      IF (STAGES) IPOWER=IPOWER-1
  160 YSTEP=10.0**IPOWER
      ZQNUM=AMOD(ZQMIN,YSTEP)
      YORGIN=ZQMIN-ZQNUM
      IF (ZQMIN.LT.0.0.AND.ZQNUM.NE.0.0) YORGIN=YORGIN-YSTEP
      ZQNUM=AMOD(ZQMAX,YSTEP)
      YUPPER=ZQMAX-ZQNUM
      IF (ZQNUM.NE.0.0) YUPPER=YUPPER+YSTEP
      ZQSTEP=(YUPPER-YORGIN)/YLEN
      IF (ZQSTEP.GT.YSTEP) YSTEP=2.0*YSTEP
      IF (ZQSTEP.GT.YSTEP) YSTEP=2.5*YSTEP
      IF (ZQSTEP.LE.YSTEP) GO TO 180
  170 IPOWER=IPOWER+1
      IF (IPOWER.LT.6) GO TO 160
      WRITE (PRINTR,470)
      RETURN
  180 ZQNUM=AMOD(ZQMIN,YSTEP)
      YORGIN=ZQMIN-ZQNUM
      IF (ZQMIN.LT.0.0.AND.ZQNUM.NE.0.0) YORGIN=YORGIN-YSTEP
      YUPPER=YLEN*YSTEP+YORGIN
      IF (ZQMAX.GT.YUPPER) GO TO 170
      CALL GRAPH(XORGIN,XSTEP,YORGIN,YSTEP)
C     IF NECESSARY INCREASE PLOT SCALE EXPONENT OF VERTICAL AXIS
      IF (YSTEP.GT.0.5) IPOWER=ALOG10(YSTEP)
      ZQBASE=10.0**IPOWER
      IF (ABS(YORGIN/ZQBASE).LT.9.8.OR.ABS((YLEN*YSTEP+YORGIN)/ZQBASE).L
     1T.9.8) GO TO 190
      IPOWER=IPOWER+1
      ZQBASE=10.0**IPOWER
C
C     DRAW TIME AXIS
C
  190 CALL FRAME
      XBACK=0.5
      YPOS=-1.5*HSIZ
      YNEW=YLEN-0.14
      CALL HEIGHT(HSIZ)
      HX=XSPACE*0.5
      YNEWW=YLEN-0.07
      DO 200 K=2,24,2
      XPOS=K/2*XSPACE
      HXPOS=XPOS-HX
      CALL STRTPT(HXPOS,0.07)
```

```
      CALL CONNPT(HXPOS,0.00)
      CALL STRTPT(XPOS,0.14)
      CALL CONNPT(XPOS,0.00)
      IF (K.GE.10) XBACK=1.5
      XPOS=XPOS-XBACK*HSIZWH
      CALL INTNO(K,XPOS,YPOS)
  200 CONTINUE
      DO 210 K=2,24,2
      XPOS=K/2*XSPACE
      HXPOS=XPOS-HX
      CALL STRTPT(HXPOS,YLEN)
      CALL CONNPT(HXPOS,YNEWW)
      CALL STRTPT(XPOS,YLEN)
      CALL CONNPT(XPOS,YNEW)
  210 CONTINUE
C
C     DRAW DISCHARGE/STAGE AXIS
C
      CALL ANGLE(90.)
      XPOS=-0.5*HSIZ
      ZQNUM=YORGIN/ZQBASE
      ISPACE=2
      AZQNUM=ABS(ZQNUM)
      IF (AZQNUM.GE.9.8) ISPACE=ALOG10(AZQNUM)+2.05
      CALL REALNO(ZQNUM,1,XPOS,-ISPACE*HSIZWH)
      XNEW=XLEN-0.14
      XNEWW=XLEN-0.07
      DO 220 K=1,9
      YPOS=K
      HYPOS=YPOS-0.5
      ZQNUM=(YPOS*YSTEP+YORGIN)/ZQBASE
      ISPACE=2
      AZQNUM=ABS(ZQNUM)
      IF (AZQNUM.GE.9.8) ISPACE=ALOG10(AZQNUM)+2.05
      CALL REALNO(ZQNUM,1,XPOS,YPOS-ISPACE*HSIZWH)
      CALL STRTPT(0.00,HYPOS)
      CALL CONNPT(0.07,HYPOS)
      CALL STRTPT(0.00,YPOS)
      CALL CONNPT(0.14,YPOS)
  220 CONTINUE
      CALL STRTPT(0.00,YLEN-0.5)
      CALL CONNPT(0.07,YLEN-0.5)
      DO 230 K=1,9
      YPOS=K
      HYPOS=YPOS-0.5
      CALL STRTPT(XNEWW,HYPOS)
      CALL CONNPT(XLEN,HYPOS)
      CALL STRTPT(XNEW,YPOS)
      CALL CONNPT(XLEN,YPOS)
  230 CONTINUE
      CALL STRTPT(XNEWW,YLEN-0.5)
      CALL CONNPT(XLEN,YLEN-0.5)
      ZQNUM=(YLEN*YSTEP+YORGIN)/ZQBASE
      ISPACE=2
      AZQNUM=ABS(ZQNUM)
      IF (AZQNUM.GE.9.8) ISPACE=ALOG10(AZQNUM)+2.05
      CALL REALNO(ZQNUM,1,XPOS,YLEN-ISPACE*HSIZWH)
      CALL RESET('ANGLE')
      CALL RESET('HEIGHT')
C
C     DETERMINE LOCATION OF LEGEND
C
      IF (IQUAD.NE.0.AND.LOCLEG) GO TO 240
      LOCLEG=.FALSE.
      ICON=MOD(NS,CPP)
      IF (ICON.EQ.0.AND.CPP.LE.5) ICON=CPP
      IF ((CPP.GT.5.AND.NPLOT.GT.1).OR.ICON.GT.5) ICON=5
      IF (IFLAG) ICON=ICON+1
```

```
      YLEG=ICON*(YLEGSP+HLEG)+.200
      YTEST=(ZQNUM-YLEG*(ZQNUM-YORGIN/ZQBASE)/YLEN)*10.**IPOWER
      IQUAD=2
      IF (ZQMAX.GE.YTEST.AND.TMAX.LT.540.0) IQUAD=1
      IF (ZQCMAX.GE.YTEST.AND.TCMAX.LT.540.0) IQUAD=1
C
C     LABEL PLOT AND AXES
C
  240 XPOS=XLEN*0.5-31.*SIZEWH
      YPOS=YLEN+SIZE
      CALL MESSAG('FLOW COMPUTED BY THE BRANCH NETWORK MODEL ON   /  /
     1   .   .   ',62,XPOS,YPOS)
      XPOS=XPOS+45.0*SIZEWH
      CALL INTNO(YR,XPOS,YPOS)
      XPOS=XPOS+3.0*SIZEWH
      CALL INTNO(MO,XPOS,YPOS)
      XPOS=XPOS+3.0*SIZEWH
      CALL INTNO(DA,XPOS,YPOS)
      XPOS=XPOS+3.0*SIZEWH
      CALL INTNO(HR,XPOS,YPOS)
      XPOS=XPOS+3.0*SIZEWH
      CALL INTNO(MN,XPOS,YPOS)
      XPOS=XPOS+3.0*SIZEWH
      CALL INTNO(SE,XPOS,YPOS)
      XPOS=XLEN-8.*SIZEWH
      YPOS=-2.75*SIZE
      CALL MESSAG('  /  /  ',8,XPOS,YPOS)
      CALL INTNO(KYR,XPOS,YPOS)
      XPOS=XPOS+3.0*SIZEWH
      CALL INTNO(KMO,XPOS,YPOS)
      XPOS=XPOS+3.0*SIZEWH
      CALL INTNO(KDA,XPOS,YPOS)
      XPOS=XLEN*0.5-2.*SIZEWH
      CALL MESSAG('TIME',4,XPOS,YPOS)
      CALL ANGLE(90.)
      XPOS=-1.75*SIZE
      NOCHAR=21
      IF (STAGES) NOCHAR=16
      YPOS=(YLEN-NOCHAR*SIZEWH)*0.5
      IF (OUNIT.EQ.ME) GO TO 260
      IF (STAGES) GO TO 250
      CALL MESSAG('DISCHARGE, IN 10  CFS',NOCHAR,XPOS,YPOS)
      NOCHAR=NOCHAR-5
      GO TO 280
  250 CALL MESSAG('STAGE, IN 10  FT',NOCHAR,XPOS,YPOS)
      NOCHAR=NOCHAR-4
      GO TO 280
  260 IF (STAGES) GO TO 270
      NOCHAR=NOCHAR+1
      CALL MESSAG('DISCHARGE, IN 10  M /S',NOCHAR,XPOS,YPOS)
      XPOS1=XPOS-0.5*SIZE
      YPOS1=YPOS+19.0*SIZEWH
      CALL HEIGHT(SIZ7)
      CALL INTNO(3,XPOS1,YPOS1)
      NOCHAR=NOCHAR-6
      GO TO 280
  270 CALL MESSAG('STAGE, IN 10  CM',NOCHAR,XPOS,YPOS)
      NOCHAR=NOCHAR-4
  280 XPOS=XPOS-0.5*SIZE
      YPOS=YPOS+NOCHAR*SIZEWH
      CALL HEIGHT(SIZ7)
      CALL INTNO(IPOWER,XPOS,YPOS)
      CALL RESET('ANGLE')
C
C     INITIALIZE LOCATION OF LEGEND
C
      CALL HEIGHT(HLEG)
      XPOS=LEGEND(IQUAD,1)
```

```
      YPOS=LEGEND(IQUAD,2)+.5*HLEG
      XPOS1=XPOS+XLCURV+HSIZWH
      YPOS1=YPOS-.5*HLEG
      XPOS2=XPOS1+3.0*HSIZWH
      XPOS3=XPOS1
      IF (I.LT.10) XPOS3=XPOS3+HSIZWH
      HSPACE=HLEG+YLEGSP
      JNS=CPP
      IF (MEPLOT) GO TO 300
      IF (NS.GT.CPP) GO TO 290
      JNS=NS
      GO TO 300
  290 IF (IPLOT.LT.NPLOT-1) GO TO 300
      INS=NS-KK+CPP
      JNS=INS/2
      IF (IPLOT.NE.NPLOT) JNS=JNS+MOD(INS,2)
C
C     BEGIN CURVE PLOTTING LOOP
C
  300 DO 390 J=1,JNS
      JLINE=MOD(J,5)
      ISEC=J+(IPLOT-1)*CPP
      GO TO (310,320,330,340), JLINE
      CALL CHNDOT
      IF (J-5) 380,350,380
  310 CALL RESET('DASH')
      IF (J-1) 380,370,380
  320 CALL DASH
      IF (J-2) 380,350,380
  330 CALL CHNDSH
      IF (J-3) 380,350,380
  340 CALL DOT
      IF (J.NE.4) GO TO 380
  350 IF (IQUAD.LE.2) GO TO 360
      YPOS1=YPOS1+HSPACE
      YPOS=YPOS+HSPACE
      GO TO 370
  360 YPOS1=YPOS1-HSPACE
      YPOS=YPOS-HSPACE
  370 CALL STRTPT(XPOS,YPOS)
      CALL CONNPT(XPOS+XLCURV,YPOS)
      CALL MESSAG('  -',3,XPOS1,YPOS1)
      CALL INTNO(I,XPOS3,YPOS1)
      CALL INTNO(ISEC,XPOS2,YPOS1)
C     PLOT DISCHARGE/STAGE DATA CURVES
  380 IJ=IN+ISEC
      CALL CURVE(ZQTIME(ID),ZQCOMP(ID,IJ),KD,0)
      IF (.NOT.IFLAG) GO TO 390
      CALL DASH
      IF (IQUAD.LE.2) YPOS=YPOS-HSPACE
      IF (IQUAD.GT.2) YPOS=YPOS+HSPACE
      CALL STRTPT(XPOS,YPOS)
      CALL CONNPT(XPOS+XLCURV,YPOS)
      CALL CURVE(RMTIME(IDM),ZQMEAS(MID,MM),MKD,0)
  390 CONTINUE
C
C     PLOT FIELD STATION NUMBERS
C
      CALL HEIGHT(HSIZ)
      LABSTA=.FALSE.
      XPOS=LEGEND(IQUAD,1)-14.0*HSIZWH
      IF (IQUAD.EQ.2.OR.IQUAD.EQ.3) XPOS=XPOS+XLCURV+18.0*HSIZWH
      YSTNO=LEGEND(IQUAD,2)
      DO 420 L=IBND,NBND
      IF (ISTATN(L).EQ.0) GO TO 420
      IF (IBJNC(L).NE.IJF(I)) GO TO 400
      ISEC=1
      IF (IJF(I).GT.IJT(I)) ISEC=NSEC(I)
```

```
      GO TO 410
  400 IF (IBJNC(L).NE.IJT(I)) GO TO 420
      ISEC=1
      IF (IJT(I).GT.IJF(I)) ISEC=NSEC(I)
  410 IF (ISEC.GT.CPP*IPLOT) GO TO 420
      ICON=MOD(ISEC,CPP)
      IF (ICON.GT.5) GO TO 420
      IF (ICON.EQ.O.AND.CPP.LE.5) ICON=CPP
      LABSTA=.TRUE.
      YPOS=YSTNO-(ICON-1)*HSPACE
      IF (IQUAD.GT.2) YPOS=YSTNO+(ICON-1)*HSPACE
      ISTA=ISTATN(L)
      ISTA1=ISTA/1000000
      STA2=(ISTA-ISTA1*1000000)*0.01+0.005
      CALL MESSAG(' (   -         )',13,XPOS,YPOS)
      XPOS1=XPOS+2.0*HSIZWH
      IF (ISTA1.LT.10) XPOS1=XPOS1+HSIZWH
      CALL INTNO(ISTA1,XPOS1,YPOS)
      CALL REALNO(STA2,2,XPOS+4.0*HSIZWH,YPOS)
      IF (.NOT.IFLAG) GO TO 420
      IF (IQUAD.LE.2) YPOS=YPOS-HSPACE
      IF (IQUAD.GT.2) YPOS=YPOS+HSPACE
      CALL MESSAG(' ( MEASURED )',13,XPOS,YPOS)
      IFLAG=.FALSE.
  420 CONTINUE
      IF (.NOT.IFLAG) GO TO 430
      IF (LABSTA) GO TO 430
      YPOS=YSTNO
      IF (IQUAD.GT.2) YPOS=YSTNO+HSPACE
      CALL MESSAG(' ( COMPUTED )',13,XPOS,YPOS)
      IF (IQUAD.LE.2) YPOS=YPOS-HSPACE
      IF (IQUAD.GT.2) YPOS=YPOS+HSPACE
      CALL MESSAG(' ( MEASURED )',13,XPOS,YPOS)
  430 IBND=L
      CALL ENDPL(IPLMSG*KTPLOT)
  440 CONTINUE
  450 CONTINUE
C
C     TERMINATE PLOT AND CLOSE OUT PLOT FILE
C
  460 IF (M.EQ.NSTEPS) CALL DONEPL
      ID=1
      LETIME=KETIME+IDTM
      IF (.NOT.MEPLOT) RETURN
      IF (KETIME.LT.MEITIM) RETURN
      IDM=1
      MID=MND+1
      RETURN
C
C     OUTPUT FORMAT STATEMENT
C
C
  470 FORMAT (/' ERROR,     STEP SIZE IN PLOT SCALE ALGORITHM EXCEEDS MA
     1XIMUM LIMIT')
      END


      SUBROUTINE PRTPLT(NSEC,XSKT,BRNAME,IJF,IJT,ISTATN,ZQCOMP,IBJNC,MBC
     1H,MSEC,MDATA,ZQMEAS,MAXCZQ,MAXS,MAXMZQ,IYR,IMO,IDA,IHR,IMN,PRINTR)
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #   THIS SUBROUTINE PRODUCES A LINE-PRINTER PLOT OF STAGE OR        #
C #   DISCHARGE DATA COMPUTED BY A FLOW MODEL. THE MAXIMUM NUMBER OF  #
C #   POINTS PLOTTED PER PAGE OF OUTPUT IS SPECIFIABLE.               #
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #                                                                   #
      REAL SREAL(11)
      INTEGER BRNAME(10,1),ISTATN(1)
```

```
      INTEGER YR,MO,DA,HR,MN
C     INITIALIZE PAGE SIZE FOR THE NUMBER OF DATA PLOTTED PER PAGE
      INTEGER PAGESZ/48/
      INTEGER *2IUNIT,OUNIT,ME/'ME'/,MT/' M'/,CM/'CM'/
      INTEGER *2VBAR/'| '/,ASTK/'* '/,OH/'O '/,SYMBOL(6)/'A ','B ','C ',
     1'D ','E ','F '/,SYMBEG,SYMEND
      INTEGER *2IYR,IMO,IDA,IHR,IMN,CHARA,CHARB
      INTEGER *2NSEC(1),XSKT(1),IJF(1),IJT(1),IBJNC(1)
      INTEGER *2MBCH(1),MSEC(1),MDATA(1),MIYR,MIMO,MIDA,MIHR,MIMN
      DIMENSION ZQCOMP(MAXCZQ,1), ZQMEAS(MAXMZQ,1)
      INTEGER STANO1,STANO2,PRINTR
      INTEGER ZTITLE(3)/'    ','STAG','E   '/
      INTEGER QTITLE(3)/'  DI','SCHA','RGE '/
      INTEGER *2ZUNIT(5)/'  ',' (','FT',') ','  '/
      INTEGER *2QUNIT(5)/' (','FT','**','3/','S)'/
      INTEGER DPERM(12)/31,28,31,30,31,30,31,31,30,31,30,31/
      INTEGER *2B(101)/'| ',9*'  ','| ',9*'  ','| ',9*'  ','| ',9*'  ','
     1| ',9*'  ','| ',9*'  ','| ',9*'  ','| ',9*'  ','| ',9*'  ','| ',9*
     2'  ','| '/
      INTEGER *2D(101)/'+ ',9*'- ','+ ',9*'- ','+ ',9*'- ','+ ',9*'- ','
     1+ ',9*'- ','+ ',9*'- ','+ ',9*'- ','+ ',9*'- ','+ ',9*'- ','+ ',9*
     2'- ','+ '/
      LOGICAL MEPLOT/.FALSE./,STAGES
      COMMON /DATIME/ KYR,KMO,KDA,KHR,KMN,IDTM,M,NSTEPS,INHR,INMN,IDTPDY
     1,LASTN
      COMMON /OUTPUT/ NETNAM(20),NBCH,NBND,IOTOPT,IPLOPT,IPLDEV,STAGES,
     1ZDATUM,IUNIT,OUNIT
      COMMON /MEDATA/ MDT,KTMEAS,LETIME,KETIME,MEITIM,MEKTIM,IPLMSG,MIYR
     1,MIMO,MIDA,MIHR,MIMN
C
C     INITIALIZE PLOT CONTROL PARAMETERS
C
      LASTM=0
      IF (STAGES) GO TO 30
      DO 10 K=1,3
   10 ZTITLE(K)=QTITLE(K)
      DO 20 K=1,5
   20 ZUNIT(K)=QUNIT(K)
      IF (OUNIT.EQ.ME) ZUNIT(2)=MT
      GO TO 40
   30 IF (OUNIT.EQ.ME) ZUNIT(3)=CM
   40 YR=IYR
      MO=IMO
      DA=IDA
      HR=IHR
      MN=IMN
      DPERM(2)=28+(4-MOD(YR,4))/4
      IF (IPLOPT.LT.3) RETURN
      MEPLOT=.TRUE.
      MID=1
      ID=(INHR*60+INMN)/IDTM
      RETURN
C
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #   SECONDARY ENTRY POINT TO PLOT FLOW RESULTS                      #
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C
      ENTRY LNPLOT
      KT=M-LASTM
      IF (.NOT.MEPLOT) GO TO 50
      IF (KETIME.LT.MEITIM.OR.LETIME.GT.MEKTIM) GO TO 420
      IDM=(MIHR*60+MIMN)/MDT-ID+1
      MND=(KHR*60+KMN)/MDT
C
C     BEGIN BRANCH LOOP
C
   50 DO 410 I=1,NBCH
      IF (.NOT.MEPLOT) GO TO 80
```

```
      DO 70 MM=1,KTMEAS
      IF (MBCH(MM).NE.I) GO TO 70
      IF (MDATA(MM).EQ.0) GO TO 410
      IF (MND.GT.MDATA(MM)) MND=MDATA(MM)
      NDM=IDM+MND-1
      IF (NDM.LE.1440/MDT) GO TO 60
      NDM=1440/MDT
      MND=NDM-IDM+1
   60 MDATA(MM)=MDATA(MM)-MND+MID-1
      IDMM1=IDM-1
      GO TO 80
   70 CONTINUE
      GO TO 410
   80 NS=NSEC(I)
      IJ=MAXS-XSKT(I)
C
C     BEGIN CROSS-SECTION LOOP
C
      JBEG=1
      JEND=2
      NPLOTS=(NS+1)/2
      IF (MEPLOT) NPLOTS=1
      DO 400 IPLOT=1,NPLOTS
      IF (IPLOT.EQ.1) GO TO 90
      JBEG=JEND+1
      JEND=JEND+2
      IJ=IJP1
      IF (JEND.LE.NS) GO TO 90
      JEND=JBEG
      JBEG=JEND-1
      GO TO 100
   90 IJ=IJ+1
  100 IJP1=IJ+1
      SYMBEG=SYMBOL(JBEG)
      SYMEND=SYMBOL(JEND)
C
C     DETERMINE FIELD-STATION NUMBER AT CROSS-SECTION LOCATION
C
      IF (MEPLOT) GO TO 140
      IF (JBEG.NE.1.AND.JEND.NE.NS) GO TO 170
      ISEC=0
      DO 130 L=1,NBND
      IF (ISTATN(L).EQ.0) GO TO 130
      IF (IBJNC(L).NE.IJF(I)) GO TO 110
      ISEC=1
      IF (IJF(I).GT.IJT(I)) ISEC=NS
      GO TO 120
  110 IF (IBJNC(L).NE.IJT(I)) GO TO 130
      ISEC=1
      IF (IJT(I).GT.IJF(I)) ISEC=NS
  120 STANO1=ISTATN(L)/1000000
      STANO2=ISTATN(L)-(STANO1*1000000)
      STANO3=STANO2/100.+.005
      GO TO 170
  130 CONTINUE
      GO TO 170
C
C     DETERMINE DATA RANGE FOR COMPUTED VRS. MEASURED DATA PLOT
C
  140 IJ=IJ+MSEC(MM)-1
      ZQMIN=+9999999.
      ZQMAX=-9999999.
      DO 150 K=1,KT
      ZQIJK=ZQCOMP(K,IJ)
      IF (ZQIJK.LT.ZQMIN) ZQMIN=ZQIJK
      IF (ZQIJK.GT.ZQMAX) ZQMAX=ZQIJK
  150 CONTINUE
      DO 160 K=MID,MND
```

```
      ZQIJK=ZQMEAS(K,MM)
      IF (ZQIJK.LT.ZQMIN) ZQMIN=ZQIJK
      IF (ZQIJK.GT.ZQMAX) ZQMAX=ZQIJK
  160 CONTINUE
      GO TO 200
C
C     DETERMINE DATA RANGE FOR PLOTTING COMPUTED RESULTS
C
  170 ZQMIN=ZQCOMP(1,IJ)
      ZQMAX=ZQMIN
      ZQIJK=ZQCOMP(1,IJP1)
      IF (ZQIJK.LT.ZQMIN) ZQMIN=ZQIJK
      IF (ZQIJK.GT.ZQMAX) ZQMAX=ZQIJK
      IF (KT.EQ.1) GO TO 190
      DO 180 IJK=IJ,IJP1
      DO 180 K=2,KT
      ZQIJK=ZQCOMP(K,IJK)
      IF (ZQIJK.LT.ZQMIN) ZQMIN=ZQIJK
      IF (ZQIJK.GT.ZQMAX) ZQMAX=ZQIJK
  180 CONTINUE
      GO TO 200
  190 ZQMAX=ZQMIN+ABS(ZQMIN*0.1)
C
C     DETERMINE PLOT SCALE
C
  200 RANGE=ZQMAX-ZQMIN
      IPOWER=ALOG10(RANGE)-1.0
  210 YSTEP=10.0**IPOWER
      ZQREM=AMOD(ZQMIN,YSTEP)
      YORGIN=ZQMIN-ZQREM
      IF (ZQMIN.LT.0.0.AND.ZQREM.NE.0.0) YORGIN=YORGIN-YSTEP
      ZQREM=AMOD(ZQMAX,YSTEP)
      YUPPER=ZQMAX-ZQREM
      IF (ZQREM.NE.0.0) YUPPER=YUPPER+YSTEP
      ZQSTEP=(YUPPER-YORGIN)*0.1
      IF (ZQSTEP.GT.YSTEP) YSTEP=2.0*YSTEP
      IF (ZQSTEP.GT.YSTEP) YSTEP=2.5*YSTEP
      IF (ZQSTEP.LE.YSTEP) GO TO 220
      IPOWER=IPOWER+1
      IF (IPOWER.LT.6) GO TO 210
      WRITE (PRINTR,550)
      RETURN
  220 ZQREM=AMOD(ZQMIN,YSTEP)
      YORGIN=ZQMIN-ZQREM
      IF (ZQMIN.LT.0.0.AND.ZQREM.NE.0.0) YORGIN=YORGIN-YSTEP
      DO 230 K=1,11
  230 SREAL(K)=YORGIN+(K-1)*YSTEP
      YSTEP=YSTEP*10.0
      HSTEP=0.5*YSTEP
      YORGIN=YORGIN*100.0
      NPAGE=KT/PAGESZ+1
      IF (MOD(KT,PAGESZ).EQ.0) NPAGE=NPAGE-1
      KEND=0
C
C     BEGIN PAGE LOOP
C
      DO 400 IPAGE=1,NPAGE
      KBEG=KEND+1
      KEND=IPAGE*PAGESZ
      IF (IPAGE.EQ.NPAGE) KEND=KT
      WRITE (PRINTR,470) NETNAM
      IF (.NOT.MEPLOT) GO TO 240
      WRITE (PRINTR,540) ZTITLE,(BRNAME(K,I),K=1,10),OH,ASTK,MBCH(MM),MS
     1EC(MM)
      WRITE (PRINTR,510) ZUNIT,ZUNIT,YR,MO,DA,HR,MN,KYR,KMO,KDA,KHR,KMN
      WRITE (PRINTR,500) SREAL,ASTK,OH,(VBAR,K=1,11)
      GO TO 270
  240 WRITE (PRINTR,480) ZTITLE,(BRNAME(K,I),K=1,10),SYMBEG,SYMEND,JBEG,
```

```
     1JEND
      IF (ISEC.NE.JBEG.AND.ISEC.NE.JEND) GO TO 250
      WRITE (PRINTR,490) ZUNIT,ZUNIT,YR,MO,DA,HR,MN,STANO1,STANO3,SYMBOL
     1(ISEC),KYR,KMO,KDA,KHR,KMN
      GO TO 260
  250 WRITE (PRINTR,510) ZUNIT,ZUNIT,YR,MO,DA,HR,MN,KYR,KMO,KDA,KHR,KMN
  260 WRITE (PRINTR,500) SREAL,SYMBEG,SYMEND,(VBAR,K=1,11)
  270 IF (IPAGE.NE.1) GO TO 280
      NHOUR=HR
      NMINIT=MN
C
C     BEGIN LOOP FOR PLOTTING COMPUTED VRS. MEASURED DATA
C
  280 IF (.NOT.MEPLOT) GO TO 350
      DO 340 K=KBEG,KEND
      ZQIJK=ZQCOMP(K,IJ)*100.+HSTEP
      LOCA=1+(ZQIJK-YORGIN)/YSTEP
      IF (NMINIT.EQ.O.OR.K.EQ.KBEG.OR.K.EQ.KEND) GO TO 300
      CHARA=B(LOCA)
      IF (K.LT.IDM.OR.K.GT.NDM) GO TO 290
      ZQIJK=ZQMEAS(K-IDMM1,MM)*100.+HSTEP
      LOCB=1+(ZQIJK-YORGIN)/YSTEP
      CHARB=B(LOCB)
      B(LOCA)=ASTK
      B(LOCB)=OH
      WRITE (PRINTR,520) NHOUR,NMINIT,ZQCOMP(K,IJ),ZQMEAS(K-IDMM1,MM),B
      B(LOCA)=CHARA
      B(LOCB)=CHARB
      GO TO 320
  290 B(LOCA)=ASTK
      WRITE (PRINTR,530) NHOUR,NMINIT,ZQCOMP(K,IJ),B
      B(LOCA)=CHARA
      GO TO 320
  300 CHARA=D(LOCA)
      IF (K.LT.IDM.OR.K.GT.NDM) GO TO 310
      ZQIJK=ZQMEAS(K-IDMM1,MM)*100.+HSTEP
      LOCB=1+(ZQIJK-YORGIN)/YSTEP
      CHARB=D(LOCB)
      D(LOCA)=ASTK
      D(LOCB)=OH
      WRITE (PRINTR,520) NHOUR,NMINIT,ZQCOMP(K,IJ),ZQMEAS(K-IDMM1,MM),D
      D(LOCA)=CHARA
      D(LOCB)=CHARB
      GO TO 320
  310 D(LOCA)=ASTK
      WRITE (PRINTR,530) NHOUR,NMINIT,ZQCOMP(K,IJ),D
      D(LOCA)=CHARA
  320 NMINIT=NMINIT+IDTM
      IF (NMINIT.LT.60) GO TO 330
      NHOUR=NHOUR+NMINIT/60
      NMINIT=MOD(NMINIT,60)
  330 IF (NHOUR.LT.24.OR.(NHOUR.EQ.24.AND.NMINIT.EQ.O)) GO TO 340
      NHOUR=NHOUR-24
  340 CONTINUE
      GO TO 400
C
C     BEGIN LOOP FOR PLOTTING COMPUTED RESULTS
C
  350 DO 390 K=KBEG,KEND
      ZQIJK=ZQCOMP(K,IJ)*100.+HSTEP
      LOCA=1+(ZQIJK-YORGIN)/YSTEP
      ZQIJK=ZQCOMP(K,IJP1)*100.+HSTEP
      LOCB=1+(ZQIJK-YORGIN)/YSTEP
      IF (NMINIT.EQ.O.OR.K.EQ.KBEG.OR.K.EQ.KEND) GO TO 360
      CHARA=B(LOCA)
      CHARB=B(LOCB)
      B(LOCA)=SYMBEG
      B(LOCB)=SYMEND
```

```
      WRITE (PRINTR,520) NHOUR,NMINIT,ZQCOMP(K,IJ),ZQCOMP(K,IJP1),B
      B(LOCA)=CHARA
      B(LOCB)=CHARB
      GO TO 370
  360 CHARA=D(LOCA)
      CHARB=D(LOCB)
      D(LOCA)=SYMBEG
      D(LOCB)=SYMEND
      WRITE (PRINTR,520) NHOUR,NMINIT,ZQCOMP(K,IJ),ZQCOMP(K,IJP1),D
      D(LOCA)=CHARA
      D(LOCB)=CHARB
  370 NMINIT=NMINIT+IDTM
      IF (NMINIT.LT.60) GO TO 380
      NHOUR=NHOUR+NMINIT/60
      NMINIT=MOD(NMINIT,60)
  380 IF (NHOUR.LT.24.OR.(NHOUR.EQ.24.AND.NMINIT.EQ.0)) GO TO 390
      NHOUR=NHOUR-24
  390 CONTINUE
  400 CONTINUE
  410 CONTINUE
C
C     SET PARAMETERS FOR NEXT PLOT
C
  420 ID=1
      YR=KYR
      MO=KMO
      DA=KDA
      HR=KHR
      MN=KMN
      MN=MN+IDTM
      IF (MN.LT.60) GO TO 430
      HR=HR+MN/60
      MN=MOD(MN,60)
  430 IF (HR.LT.24.OR.(HR.EQ.24.AND.MN.EQ.0)) GO TO 440
      HR=HR-24
      DA=DA+1
  440 IF (DA.LE.DPERM(MO)) GO TO 450
      DA=1
      MO=MO+1
  450 IF (MO.LT.13) GO TO 460
      MO=1
      YR=YR+1
      IF (YR.GT.99) YR=0
      DPERM(2)=28+(4-MOD(YR,4))/4
  460 LASTM=M
      IF (.NOT.MEPLOT) RETURN
      LETIME=KETIME+IDTM
      IF (KETIME.LT.MEITIM) RETURN
      MIHR=HR
      MIMN=MN
      MID=NDM-IDM+2
      RETURN
C
C     OUTPUT FORMAT STATEMENTS
C
C
C
  470 FORMAT ('1',20A4,09X,'FLOW COMPUTED BY THE BRANCH-NETWORK MODEL'//
     1' ',T63,'# # # #'//' ======+========================+',99('='),'+')
  480 FORMAT ('  TIME |',4X,3A4,6X,'|',T33,10A4,T106,'(',A1,'-',A1,') CR
     1OSS SECTIONS ',I1,'-',I1,T131,'|')
  490 FORMAT (' HR:MN |',5A2,1X,5A2,1X,'| START= ',I2,2('/',I2),I3,':',I
     12,T64,'FIELD-STATION NUMBER ',I2,'-',F7.2,' (',A1,')',T111,'END= '
     2,I2,2('/',I2),I3,':',I2,' |')
  500 FORMAT (' ','======+========================+',99('='),'+'/' ',T24,1
     11F10.2/T13,'(',A1,')',T24,'(',A1,')',T31,A1,10(9X,A1))
  510 FORMAT (' HR:MN |',5A2,1X,5A2,1X,'| START= ',I2,2('/',I2),I3,':',I
     12,T111,'END= ',I2,2('/',I2),I3,':',I2,' |')
```

```
  520 FORMAT (1X,I2,':',I2,' |',F10.2,1X,F10.2,1X,101A1)
  530 FORMAT (1X,I2,':',I2,' |',F10.2,12X,101A1)
  540 FORMAT ('  TIME |',4X,3A4,6X,'|',T33,10A4,T82,'(',A1,') MEASURED (
     1',A1,') COMPUTED AT BRANCH:SECTION ',I2,':',I1,T131,'|')
  550 FORMAT (/' ERROR,     STEP SIZE IN PLOT SCALE ALGORITHM EXCEEDS MA
     1XIMUM LIMIT')
      END
```

```
      SUBROUTINE ARBIN(AA,BB,ZA,IPT,XSKT,MXPT,MAXS)
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #   THIS SUBROUTINE INTERPOLATES AREA AND TOP-WIDTH PROPERTIES.     #
C #   THE HYDRAULIC RADIUS RETURNED IS AN APPROXIMATION (AREA/WIDTH). #
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #                                                                   #
      INTEGER *2IPT(1),XSKT(1)
      DIMENSION AA(MXPT,1), BB(MXPT,1), ZA(MXPT,1)
      RETURN
      ENTRY ARB(ZIJ,I,J,A,B,R,*,*)
C
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #   SECONDARY ENTRY POINT CALLED TO PERFORM INTERPOLATION           #
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C
      IJ=MAXS-XSKT(I)+J
      IF (ZIJ.LT.ZA(1,IJ).OR.ZIJ.GT.ZA(IPT(IJ),IJ)) RETURN 1
      ND=IPT(IJ)
      DO 10 K=2,ND
      IF (ZIJ.GT.ZA(K,IJ)) GO TO 10
      ZIJK=(ZA(K,IJ)-ZIJ)/(ZA(K,IJ)-ZA(K-1,IJ))
      DA=AA(K,IJ)-AA(K-1,IJ)
      DB=BB(K,IJ)-BB(K-1,IJ)
      A=AA(K,IJ)-DA*ZIJK
      B=BB(K,IJ)-DB*ZIJK
      IF (A.LE.0.0.OR.B.LE.0.0) RETURN 2
      R=A/B
      RETURN
   10 CONTINUE
      RETURN 1
      END
```

```
      SUBROUTINE GEMXPI(AM,BMX,ROW,II,IS,ICHK)
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #   THIS SUBROUTINE SOLVES A SYSTEM OF EQUATIONS BY GAUSSIAN        #
C #   ELIMINATION USING MAXIMUM COLUMN PIVOT STRATEGY.
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
      DIMENSION AM(1), BMX(1)
      INTEGER*2 ROW(1)
C
C        INITIALIZE
C
      DO 10 K=1,II
      ROW(K)=K
   10 CONTINUE
      IF (ICHK.NE.0.AND.ICHK.NE.1) ICHK=0
      RETURN
C
C        DECOMPOSE AND SOLVE ENTRY POINT
C
      ENTRY GEMXP(IS,ICHK)
C
C        FORWARD SOLUTION
C
      TOL=.1E-50
      IS=0
```

```
      JJ=-II
      DO 90 JROW=1,II
      JJ=JJ+II
C
C        IF ICHK=1 SKIP OVER SEARCH FOR MAXIMUM PIVOT
C
      IF (ICHK.NE.1) GO TO 20
      MROW=ROW(JROW)
      BIGA=AM(MROW+JJ)
      IF (ABS(BIGA).GT.TOL) GO TO 60
   20 ABBIGA=0.0
      IMAX=0
C
C        SEARCH FOR MAXIMUM COEFFICIENT IN COLUMN
C
      DO 30 JCOL=JROW,II
      K=ROW(JCOL)+JJ
      IF (AM(K).EQ.0.0) GO TO 30
      ABSA=ABS(AM(K))
      IF (ABBIGA.GE.ABSA) GO TO 30
      BIGA=AM(K)
      ABBIGA=ABSA
      IMAX=JCOL
   30 CONTINUE
      IF (IMAX.EQ.0) GO TO 50
C
C        SWAP ROW POINTERS
C
      MROW=ROW(IMAX)
      IF (IMAX.EQ.JROW) GO TO 40
      ROW(IMAX)=ROW(JROW)
      ROW(JROW)=MROW
C
C        TEST FOR PIVOT LESS THAN TOLERANCE (SINGULAR MATRIX)
C
   40 IF (ABBIGA.GT.TOL) GO TO 60
   50 IS=1
      RETURN
C
C        DIVIDE EQUATION BY LEADING COEFFICIENT
C
   60 REBIGA=1.0/BIGA
      NBEG=JJ+MROW
      NEND=NBEG+II*(II-JROW)
      DO 70 K=NBEG,NEND,II
      IF (AM(K).NE.0.0) AM(K)=AM(K)*REBIGA
   70 CONTINUE
      BMX(MROW)=BMX(MROW)*REBIGA
C
C        ELIMINATE THE ITH COLUMN BELOW THE DIAGONAL
C
      IF (JROW.EQ.II) GO TO 100
      NBEG=NBEG+II
      JP=JROW+1
      DO 90 K1=JP,II
      KR=ROW(K1)
      KXJ=JJ+KR
      AVAL=AM(KXJ)
      IF (AVAL.EQ.0.0) GO TO 90
      DO 80 K2=NBEG,NEND,II
      KXJ=KXJ+II
      IF (AM(K2).EQ.0.0) GO TO 80
      AM(KXJ)=AM(KXJ)-AVAL*AM(K2)
   80 CONTINUE
      BMX(KR)=BMX(KR)-AVAL*BMX(MROW)
   90 CONTINUE
C
C        BACK SUBSTITUTE, PLACING THE SOLUTION IN
```

```
C         AN UNOCCUPIED COLUMN OF A
C
  100 IIN=II-1
      II2=II*IIN
      AM(II)=BMX(MROW)
      DO 110 K1=1,IIN
      KB=II-K1
      KR=ROW(KB)
      AM(KB)=BMX(KR)
      IA=II2+KR
      IC=II
      DO 110 K2=1,K1
      IF (AM(IA).NE.0.0) AM(KB)=AM(KB)-AM(IA)*AM(IC)
      IA=IA-II
      IC=IC-1
  110 CONTINUE
C
C        RESTORE THE SOLUTION TO THE B VECTOR
C
      DO 120 K=1,II
      BMX(K)=AM(K)
  120 CONTINUE
      RETURN
      END


      FUNCTION SETA(TYPETA,IUNIT,G,Q,A,R,T,RN)
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #   THIS SUBPROGRAM COMPUTES BOTTOM FRICTION AS A FUNCTION OF WATER #
C #   TEMPERATURE, FLOW DEPTH, DISCHARGE, FROUDE NUMBER, OR REYNOLDS  #
C #   NUMBER ACCORDING TO A USER DEFINED LINEAR OR QUADRATIC          #
C #   RELATIONSHIP FOR A BRANCH SEGMENT.  CONSTANT BOTTOM FRICTION IS #
C #   ALSO ACCOMMODATED.  A QUADRATIC RELATIONSHIP IS DEFINED FOR THE #
C #   KINEMATIC VISCOSITY AS A FUNCTION OF TEMPERATURE WHICH IS VALID #
C #   IN THE FAHRENHEIT TEMPERATURE RANGE 32<=T<=95 AND THE CELSIUS   #
C #   TEMPERATURE RANGE 0<=T<=35.  THE KINEMATIC VISCOSITY (NU) IS    #
C #   USED AS 10**6NU = 1 M**2/SEC IN THE METRIC SYSTEM AND AS        #
C #   10**5NU = 1 FT**2/SEC IN THE ENGLISH SYSTEM.  SET TYPETA=1 FOR  #
C #   CONSTANT BOTTOM FRICTION (ETA), 2 FOR ETA AS A FUNCTION OF      #
C #   TEMPERATURE, 3 FOR DEPTH, 4 FOR DISCHARGE, 5 FOR FROUDE NUMBER, #
C #   OR 6 FOR REYNOLDS NUMBER.                                       #
C # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
C #                                                                   #
      REAL NU
      INTEGER *2TYPETA,IUNIT,EN/'EN'/,ME/'ME'/
      DIMENSION RN(4)
      SETA=RN(1)
      GO TO (10,20,30,40,50,60), TYPETA
   10 RETURN
   20 X=T
      GO TO 70
   30 X=R
      GO TO 70
   40 X=ABS(Q)
      GO TO 70
   50 V=ABS(Q)/A
      X=V/SQRT(G*R)
      GO TO 70
   60 V=ABS(Q)/A
      IF (IUNIT.EQ.EN) NU=3.165276-(0.0454095-0.00021370*T)*T
      IF (IUNIT.EQ.ME) NU=1.794000-(0.0530429-0.00064286*T)*T
      X=V*R/NU
   70 SETA=SETA+(RN(2)+RN(3)*X)*X
      RETURN
      END
```

☆U.S. GOVERNMENT PRINTING OFFICE· 1981 — 341-614/173